Package 'dynatopmodel'

May 13, 2014

Type Package

Version 1.0

Date 2014-05-12

Title Implementation of the Dynamic TOPMODEL hydrological model

Author Peter Metcalfe, based on Fortran code by Keith Beven and Jim Freer

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

Description a native R implementation and enhancement of Dynamic TOPMODEL, Beven and Freers' (2001) extension to the semi-distributed hydrological model TOPMODEL (Beven and Kirkby, 1979).

Depends R (>= 2.10), raster, xts

Imports shape, fields, rgeos, maptools, rgdal, sp, spam, hydroGOF,topmodel, intervals, tools

Suggests rgl, igraph

License GPL-2

R topics documented:

`dynatopmodel-package` *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. Includes digital terrain analysis for discretisation of catchments by topographic indexes and other geo-referenced landscape layers.

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. 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 within hydrological similar areas identified by the discretisation procedure.

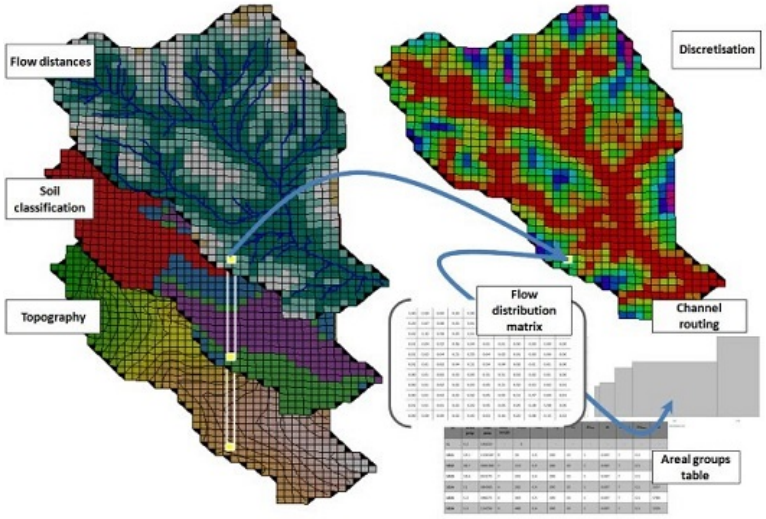


Figure 1: Aggregation of landscape layers into a catchment discretisation and its associated data structures (Metcalfe, Beven and Freer, 2014)

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 aggregated areas, referred to as hydrological response units (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, see figure 1 and the documentation for disc.catch, disc.from.dir and add.disc

Details

Package:	dynatopmodel
Type:	Package
Version:	1.0
Date:	2014-04-28
Depends:	raster, xts,
Imports:	fields, rgeos, maptools, rgdal, sp, spam, hydroGOF, topmodel, intervals, tools
Suggests:	rgl, igraph
License:	GPL (>= 2)

Note

The package has arisen from PhD research undertaken as part of a project funded by the JBA Trust (http://www.jbatrust.org). The Trust is a not-for-profit company to support and promote scientific research, education and training in the fields of environmental risks and resources. Its core funding comes from profits made by the JBA Group. However its objectives are non-commercial and it is independently managed and governed by a board of Trustees.

Due to size limitations just one, stripped-down, catchment project and discretsiation can be supplied with this package. Please contact the author if you would like to arrange access to the full data and examples of other discretisations and catchments.

JBA Trust intend to develop a detailed case study of Dynamic TOMODEL applied to flood and water quality modelling for a medium sized catchment in the UK. Please send Rob Lamb <rob.lamb@jbatrust.org> an e-mail if you are interested in collaborating on this study and / or gaining access to the data.

Thanks go to Professor Keith Beven at Lancaster Environment Centre, Dr Barry Hankin at JBA and Professor Rob Lamb at the JBA Trust for their continued supervision and support.

Author(s)

Peter Metcalfe, based on Fortran code by Keith Beven and Jim Freer

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

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.

Buytaert, W. (2011). topmodel: Implementation of the hydrological model TOPMODEL in R.

Metcalfe, Beven and Freer (2014). A modelling framework, implemented in R, for analysis of spatial heterogeneity, connectivity and complexity of landscape features and dominant rainfall-runoff characteristics. In preparation for ENVSOFT.

add.disc *Add new or replace an existing discretisation*

Description

This function will create a new discretisation according to the catchment layers present in proj, and either append it to the disc list or replace an existing item.

Usage

```
add.disc(proj, cuts = NULL, i.disc = 0, rebuild = F, chan.width= 2, ...)
```

Arguments

proj	Dynamic TOPMODEL project wrapper
cuts	cuts definition, a named list
i.disc	index of discretisation in the disc list of proj that should be replaced by the new values.
chan.width	Channel width (m)
rebuild	If a discretisation with the same parameters exists, should its files be overwritten?
...	If any additional parameters match the names in proj$catch then they will be treated as additional cuts. Otherwise they will be passed as values that can be supplied to disc.catch.

Value

If i.disc is 0 or NULL, the new discretisation. This can then be added or inserted into the project. Otherwise the project with the discretisation inserted at the given index in the list of discretisations.

Note

If i.disc is outside the range of the list index then the new discretisation will be appended.

Author(s)

Peter Metcalfe

Examples

```
## Not run:
# load the Morland Eden DTC test project
require(dynatopmod)
data(mor)

# discretise by slope and aspect
slope <- terrain(mor$dem, opt="slope", unit="degrees")
a <- upslope.area(mor$dem)

# new catchment stack
mor$catch <- stack(list("a"=a, "slope"=slope))
```

```
# new discretisation
disc <- add.disc(mor, a=4, slope=3)
plot(mor$disc$hru)

## End(Not run)
```

---

apply.params *Apply parameters to all sets in a project*

Description

Apply model parameters to all response units within all or a subset of discretisations within in a project.

Usage

```
apply.params(proj, params, which = 1:length(proj$disc))
```

Arguments

proj	
params	List of parameters to apply in form param_name=value
which	The indices of the HRU to which to apply the parameters.

Value

The updated project

Author(s)

Peter Metcalfe

Examples

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

  # load the Morland DTC project
  data(mor)
  data(mor.demo.run)
 mor$obs$qobs <- with(mor.demo.run, cbind("Observed"=ts[,"qobs"], "lnt0=11.8"=ts[,"qsim"]))
  mor$sim.start<-start(mor.demo.run$qobs)
  mor$sim.end<-end(mor.demo.run$qobs)
  # see the effect of reducing the saturated transmissivity
  mor <- apply.params(mor, list(ln_t0=10))

  run.proj(mor)

## End(Not run)
```

`approx.pe.ts` *Estimate potential evapotranspiration*

Description

In temperate, mid latitudes the contribution of evapotranspiration to the water budget can be relatively minor. An approximation derived from the annual variation in insolation is therefore acceptable (ref)

Usage

```
approx.pe.ts(start, end, dt = 1, emin = 0, emax = 10/1000)
```

Arguments

`start`	Start date of simulation in format "YYYY-MM-DD"
`end`	End date of simulated values in format "YYYY-MM-DD"
`dt`	Time interval in hours
`emin`	Minimum daily evapotranspiration (m/day)
`emax`	Maximum daily evapotranspiration (m/day)

Details

The code assumes a direct relationship between evapotranspiration and insolation. Relative humidity and wind speed are neglected, and the effect of cloud cover not considered. Evapotranspiration is assumed to be nil outside daylight hours, and to vary sinusoidal from zero at dawn and sunset to a maximum at noon and sum to the required daily total.

Value

A regular time series (xts) with interval dt whose values give the estimated potential evapotranpiration at each time.

Note

Sunrise and sunset times are for a latitude equivalent to the midlands of the UK, at about 54.5 deg N. Daylight saving time is not taken into account.

Author(s)

Peter Metcalfe, based on code by Keith Beven

Examples

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

# generate a time series of p.e using an annual daily maximum of 3mm
  pe <- approx.pe.ts("2013-01-01", "2013-12-31", emax=3/1000)

  # show a few months
```

```
  plot.zoo(pe["2013-05-01::2013-07-31"])

## End(Not run)
```

`build.calib.set` *Construct Latin hypercube of Dynamic TOPMODEL parameters*

Description

Generate mulitiple parameter sets suitable e.g. for calibration

Usage

```
build.calib.set(groups, params)
```

Arguments

`groups`	table of HRU definitions
`params`	list of parameter values of the form param_name=values.

Details

The function will create a data frame consisting of model parameters containing every permuation of the given parameters. Items generated from params will overwrite the corresponding values found in groups.

Value

A data frame with the columns found in groups but containing every combination of the supplied parameters

Note

Items in params not found in colnames(groups) will be ignored.

Author(s)

Peter Metcalfe

References

A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. M. D. McKay, R. J. Beckman and W. J. Conover.Technometrics, Vol. 21, No. 2 (May, 1979), pp. 239-245

See Also

`run.sets`

Examples

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

# load the Morland DTC project
data(mor)

# initial parameter values
hru <- mor$disc[[1]]$groups

# parameter value ranges
m <-seq(0.007, 0.011, by=0.001)
ln_t0<-seq(13.2, 15.2, by=0.4)

# explicit values for less sensitive parameters
td <- c(0.5, 1, 10)
vchan <- seq(500, 4000, by=500)

# build a calibration set
calib.set <- build.calib.set(groups=hru,
                             list(td=td, vchan=vchan, ln_t0=ln_t0, m=m)))

cat("No. calibration sets = ", nrow(calib.set), "\n")

res <- run.sets(mor, calib.set=calib.set)

## End(Not run)
```

`create.proj` *Create and / or load Dynamic TOPMODEL project*

Description

The function attempts to load a data wrapper (project) for a Dynamic TOPMODEL catchment analysis from the specified set of directories, and returns the newly created project.

Usage

```
create.proj(data.dir, id = data.dir, cuts = NULL, area.thresh = 1,
                chan.width = 1, disc.dir = file.path(data.dir,
                "disc"), obs = NULL, obs.dir = fp(data.dir, "obs"),
                ...)
```

Arguments

`data.dir`	Location of catchment files.
`id`	A character vector identify the project.
`cuts`	If this is supplied a new discretisation with the supplied number and type of cuts is created and added to the project loaded.
`area.thresh`	Minimum size (percent) for HRU groupings
`chan.width`	Channel width (m)

disc.dir	Root directory in which to locate existing and place new catchment discretisations. Defaults to a subdirectory named "disc" of the project directory.
obs	Any observations already loaded
obs.dir	A directory containing rainfall, discharge and potential evapotranspiration data. These should be in ASCII tab-delimited format and the first column should contain the time of the observation in a POSIX compatible format. If observations could not be loaded this operation gives a warning but the function continues. See load.obs for reading individual data sets.
...	Any other attributes that should be added to the project on initialisation.

Details

A minimal catchment project should contain an elevation raster in GEOTiff format named "dem.tif". Any other rasters with the same extent, resolution and coordinate system located in the specified directory will be assummed to hold relevant landscape data and be added to the layers of a multiband raster named "catch" and added to the project. This attribute may then be passed to disc.catch in order to discretise the catchment according to values found one or more of its layers. Any vector data held as ESRI shape files (*.shp) located in the project directory will also be loaded and added as elements of the project. One of these should be named drn and define the channel network. If this is not located then a threshold contributing upslope area will be used to infer the channel locations, the default being the 90th percentile of the log(log) of these values.

Value

A project wrapper suitable to be passed to run.proj

obs	Rainfall, pe and observed discharge data
disc	List of discretisations loaded from the "disc" dir. See add.disc and disc.catch for details of the structure of discretisations.

Note

Note that, in contrast to convention on Windows platforms, the UNIX forward slash is used as path separator .

Author(s)

Peter Metcalfe <p.metcalfe@lancs.ac.uk>

See Also

write.proj add.disc disc.catch run.proj

Examples

```
## Not run:
  require(dynatopmod)
  # Load data for morland catchment
  data(mor)
  # write to a new temporary directory
  dn <- file.path(Sys.getenv("TMP"), "morland")
  write.proj(mor, dn)
  # examine contents
  dir(dn)
```

```
  # reload the project, adding a new discretisation
  new.proj <- create.proj(dn, cuts=c(a=20))

  # show
  dev.new()
  plot(new.proj$dem)
  plot(new.proj$drn, add=T, col="blue", lwd=2)

## End(Not run)
```

`disc.catch` *Discretisation from a multi-band raster of catchment landscape layers*

Description

Discretise the catchment using the given landscape layers and a cuts definition

Usage

```
disc.catch(dem, catch, cuts = NULL, drn = NULL, reaches = NULL,
                cellprops = NULL, chan.width = NULL, area.thresh =
                NULL, groups = NULL, routing = NULL, w = NULL, cm =
                NULL, build.polys = F, ...)
```

Arguments

`dem`	Elevation raster. Cells sizes must be expressed in m and be equal in x and y directions. The DEM should use a projected coordinate system. This will be used to construct the flow distribution matrix w.
`catch`	Multiband raster comprising catchment landscape layers such as soil classifications, vegetation cover and land use suitable for categorising the catchment area into distinct units (a discretisation). For each list item in cuts there should be a layer with the corresponding name.
`cuts`	A list of cuts of the form layer_name=number. Each name should correspond to a layer name in the catchment raster.
`reaches`	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 = max(1, channel.width/xres(dem)
`cellprops`	Not used.
`area.thresh`	Minimum area for response units, expressed as a percentage of the catchment plan area.
`drn`	Optional vector digital river network in ESRI Shapefile format. Should use a projected system and have the same CRS as the DEM and other rasters.
`chan.width`	Channel width in m.

routing	Optionally supply a routing table if one available. Otherwise it will be calculated
cm	Optional HRU spatial distribution raster. Otherwise the calculated raster will be used.
groups	Optionally supply a HRU definition table. Otherwsie this will calculated from the other data.
build.polys	Build a shapefile comprising HRU locations and DRN (time-consuming)?
...	Other arguments not matching any of the abvove wil be treated as part of the cuts definition
w	Supply a flux distribution matrix manually to prevent this being recalculated. If its dimensions conflict with the number of HRUs in the groups table then it will be rebuilt notwithstanding the value supplied in w.

Details

The DEM is typically provided from the parent project, as each discretisation should use the same elevation data. Channel locations may vary between duscretisations depending on the source, level of detail and channel width applied. If the reaches raster is not supplied, the function first tries to create one using the DRN and specified channel width, if that is also missing a threshold contributing area is applied to identify the channel.

The cuts are applied in the order in which they appear in the list.

Value

w	Flux distribution matrix. A (nh+nc) x (nh =nc) 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 comprising the names, size and parameters of each channel and response units
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
ichan	Integer index of the channel response units within the groups table. May be changed in future so that the land units and the channels are returned separately

Note

The channel location raster may extend beyond the bounds of the DEM and catchment layers. If this is the case then it will be cropped to their extent.

Author(s)

Peter Metcalfe <p.metcalfe@lancs.ac.uk>

See Also

disc.from.dir

Examples

```
## Not run:
# load the Morland Eden DTC mitigation catchment
require(dynatopmod.dtm)
data(mor)
attach(mor)

# discretise by slope and and upslope area
slope <- terrain(dem, opt="slope", unit="degrees")
a <- upslope.area(dem)
# create a two-band catchment raster
catch <- stack(list("a"=a, "slope"=slope))
disc <- disc.catch(dem, catch, cuts=c(a=4,slope=3), drn=drn)
# show  results
dev.new()
plot(disc$hru)

## End(Not run)
```

`disc.from.dir` *Create / load catchment discretisation from disk files*

Description

The function attempts to locate and load any raster files found the given directory, if it exists, containing landscape layers. The specified cuts and other parameters will then are applied to the elevation raster and the layers located compatible with it, and a catchment discretisation returned. If an output directory is specified then the resulting files will be written to that location.

Usage

```
disc.from.dir(dn, dem, reaches = NULL, routing = NULL, cuts = NULL,
                  area.thresh = NULL, drn = NULL, chan.width = NULL,
                  rebuild = F, dn.out = dn, catch = NULL, ...)
```

Arguments

`dn`	Directory name. Will be created recursivley if not extant.
`dem`	Elevation raster. Cells sizes must be expressed in m and be equal in x and y directions. The DEM should use a projected coordinate system.
`reaches`	Raster containuing channel reach locations. 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 = max(1, channel.width/xres(dem)
`cuts`	A list of cuts of the form layer_name=number. Each name should correspond to a raster file found in the directory, which will be bound together into a single multi-band raster (stack) comrising the discretisation and channel locations.
`area.thresh`	Minimum area for response units, expressed as a percentage of the catchment plan area.

drn	Optional vector digital river network in ESRI Shapefile format. Should use a projected system and the same CRS as the DEM and other rasters.
chan.width	Channel width in m.
rebuild	Existing files in the directory for a discretisation will be used unless rebuild is set to T. In this case they wil be rebuilt from scratch.
dn.out	A directory in which to place any new discretisation files that result from calling the function. If NULL the files are not saved to disk.
routing	Optional precalculated routing table
catch	Multiband raster of catchment landscape layers e.g. loaded on project creation
...	Any named parmeters not maycing the above will be treated as part of the cuts definition

Details

The DEM is typically provided from the parent project, as each discretisation should use the same elevation data. Channel locations may vary between discretisations depending on the source, level of detail and channel width applied. If the reaches raster is not supplied, the function first tries to create one using the DRN and specified channel width, if that is also missing a threshold contributing area is applied to identify the channel.

Value

A catchment discretisation (list) or NULL if the procedure failed. The structure is described in disc.catch

Author(s)

Peter Metcalfe <p.metcalfe@lancs.ac.uk>

See Also

disc.catch

---

disp.run *Display results of a Dynamic TOPMODEL simulation run*

Description

Pretty display of simulation output plus rainfall and calculated evapotranspiration.

Usage

```
disp.run(run, qmax = NA, legend = F, title = "", disp.par = def.disp.par(), ...)
```

Arguments

run	Results of simulation returned by run.proj
qmax	Maximum discharge shown (m). Useful for setting the y scale. If not supplied then set to (max value of all data)*1.25
legend	Whether to show a legend
title	A title for the graphic
disp.par	Display parameters
...	Any graphical parameters that can be passed to plot.zoo

Details

Display a hydrograph generated by run.proj calculated evapotranspiration and observed rainfall and discharges.

Note

Multiple columns of obsevation data will be plotted separately. All input and discharge data should be specific values given in m/hr

Author(s)

Peter Metcalfe <p.metcalfe@lancs.ac.uk>

See Also

run.proj

Examples

```
## Not run:
 require(dynatopmod)
 # pre run simulation taken from Morland DTC. Manual calibration
 data(mor_demo_run)
 disp.run(mor.demo.run, title="Morland Beck, Eden DTC. 10.1 sq.km")
 title(sub="Discharge data collected and rated by Eden DTC Team http://www.edendtc.org.uk.\n
 Rainfall from Casella TBR AWS. PE estimated. All data at 15 minute intervals", cex.sub=0.75)


## End(Not run)
```

---

gr.off *Switch off graphical output for simulation runs.*

Description

Switches on the graphical output of predicted flows and evapotranspiration whilst simulation is running

Usage

```
gr.off(proj, spatial = F)
```

Arguments

proj	Dynamic TOPMODEL project
spatial	Switch off spatial output as well

Details

Graphical display of model runs is useful but can be time-consuming especially for longer runs and / or during calibration. This suppresses the graphical output.

Value

The project with updated display parameters

Author(s)

Peter Metcalfe

Examples

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

 # load the Morland DTC project
 data(mor)

 # run with console output only
 mor <- graphics.off(mor)
 run.proj(mor)



## End(Not run)
```

---

gr.on *Switch on graphical output*

Description

Switches on the graphical output of predicted flows and evapotranspiration whilst simulation is running

Usage

```
gr.on(proj, spatial = F)
```

Arguments

proj	Dynamic TOPMODEL project
spatial	Switch on spatial output as well

Value

The project with updated display parameters

Note

Spatial output appears in a separate window

Author(s)

Peter Metcalfe

Examples

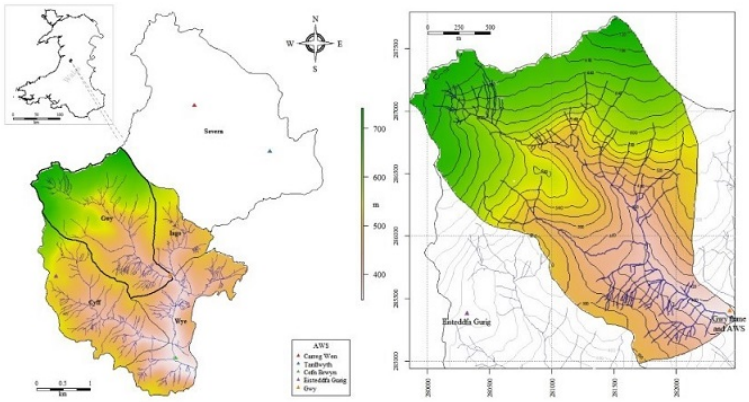
```
## Not run:
 require(dynatopmod)

 # load the Morland DTC project
 data(mor)

 # run with console output only
 mor <- graphics.on(mor)
 run.proj(mor)



## End(Not run)
```

`gwy.demo.run` *Calibration run for Gwy catchment*

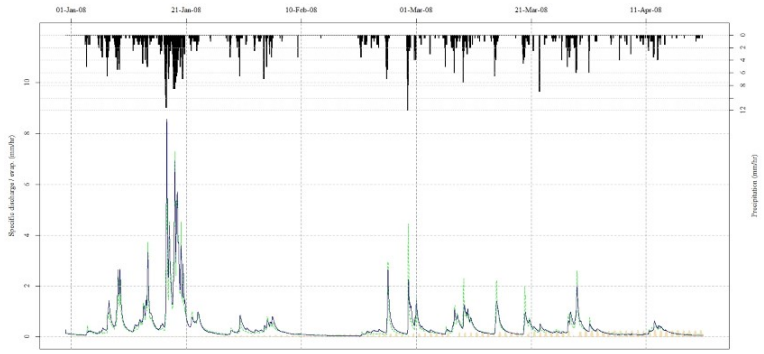
Description

Results of a calibrated run for the Gwy catchment, upper Wye, Plynlimon research catchments, mid Wales, UK.

Details



The project contains just one discretisation of 9 HRUs obtained by splitting the catchment according to upslope area. Time step = 15 minutes with 5 inner time steps giving effective step of 3 minutes; NSE=0.94. The data and calibration procedure to obtain this result is described in Metcalfe, Beven and Freer (2014).



Note

Rainfall and discharge data are under licenced by the Centre for Ecology and Hydrology, UK. For this reason, and to minimise space, only the simulated discharges, potemtial and actual evaportranspiration are included. Please contact the maintainer, Peter Metcalfe <p.metcalfe@lancs.ac.uk>, or Professor Rob Lamb <rob.lamb@jbatrust.org> to arrange accesss to the original data.

Source

Metcalfe, Beven and Freer (2014). A modelling framework, implemented in R, for analysis of spatial heterogeneity, connectivity and complexity of landscape features and dominant rainfall-runoff characteristics. In preparation for ENVSOFT.

References

Marc, V., & Robinson, M. (2007). The long-term water balance (1972-2004) of upland forestry and grassland at Plynlimon, mid-Wales. Hydrology and Earth System Sciences, 11(1), 44-60.

Newson, A. J. (1976). Some aspects of the rainfall of Plynlimon, mid-Wales. Wallingford: Institute of Hydrology.

Newson, M. D. (1976). The physiography, deposits and vegetation of the Plynlimon catchments.(A synthesis of published work and initial findings). Wallingford: Institute of Hydrology.

Newson, M. D., & Gilman, K. (Eds.). (1991). Plynlimon research: the first two decades (p. 188). Wallingford: Institute of Hydrology.

Examples

```
## Not run:
require(dynatopmodel)
data(gwy_demo_run)
# plot all series: simulated discharges, pe and ae
plot.zoo(gwy.demo.run)

## End(Not run)
```

mor *Hydrometric and topographic data for the Morland Beck test catchment, Eden DTC.*

Description

Elevation, DRN, observed flows and rainfall for the Morland subcatchment of the Eden DTC, UK.

Usage

```
data(mor)
```

Details

The Morland is one of the three 10 sq.km test basins located within the Eden Demonstration Test Catchment (DTC) in North West England. Each have been extensively instrumented and supply both discharge and a range of water quality measurements as described by Owen et al. (2012). Rainfall data are collected at 15 minute intervals from two AWS located in the upper part of the catchment. The catchment contains a "control" and a "mitigation" catchment of approximately equal areas. Spatial information for the flow gauging and automated water quality monitoring sites for the subcatchments and the main outlet at Newby Bridge are included (in the elements "sites" of type SpatialPointsDataFrame)

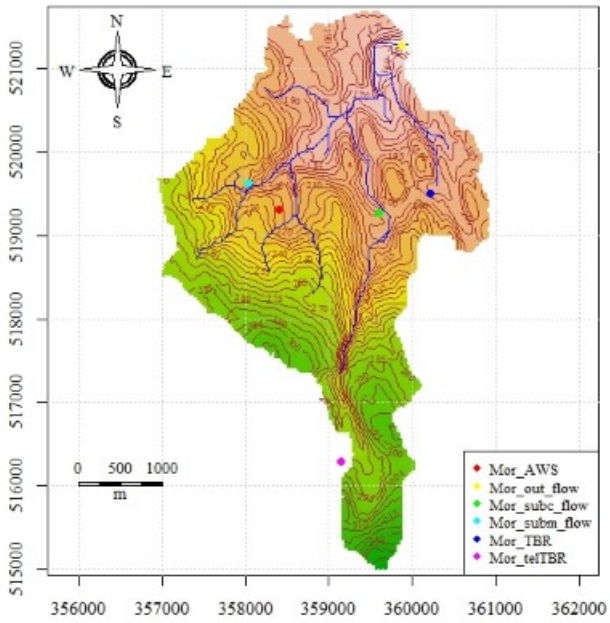


Fig1: Morland Beck, Eden DTC (10.1 sq.km),showing locations of automated weather, flow and water quality monitoring sites and approximate location of channel network.

There is considerable variability in rainfall across the catchment according to elevation and position, with annual totals ranging from around 650mm in the valley bottoms to >2000mm on the high fell tops. Rainfall totals measured by the Morland TBR were 1172 mm in 2012 and 1033 mm in 2013. Potential and actual evapotranspiration are almost identical at ~ 480mm/yr (Allen et al., 2010).

An approximate time series of PE giving an equivalent annual total has been generated using the approx.pe.ts function using a max daily total of 3mm.

Source

DEM: PROFILE DTM [TIFF geospatial data], Scale 1:10000, Tiles: ny52se,ny62sw,ny51ne,ny61nw, Updated: November 2009, Ordnance Survey (GB), Using: EDINA Digimap Ordnance Survey Service, <http://edina.ac.uk/digimap>, Downloaded: Sat Oct 20 11:34:15 BST 2012; DRN derived from Strategi [SHAPE geospatial data], Scale 1:250000, Tiles: GB, Updated: January 2014, Ordnance Survey (GB), Using: EDINA Digimap Ordnance Survey Service, <http://digimap.edina.ac.uk>, Downloaded: Fri Apr 11 21:50:06 BST 2014

Discharge and AWS data: Eden DTC (http://www.edendtc.org)

References

Allen, D. J., Newell, A. J., & Butcher, A. S. (2010). Preliminary review of the geology and hydrogeology of the Eden DTC sub-catchments.

Owen, G. J., Perks, M. T., Benskin, C. M. H., Wilkinson, M. E., Jonczyk, J., & Quinn, P. F. (2012). Monitoring agricultural diffuse pollution through a dense monitoring network in the River Eden Demonstration Test Catchment, Cumbria, UK. Area, 44(4), 443-453.

Examples

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

data(mor)

dev.new()
attach(mor)

par(family="serif")
plot(dem, legend=F)
grid()
contour(dem, add=T, col="brown", nlevels=30)
plot(drn, add=T, col="blue")

# monitoring locations and AWS
plot(sites, add=T, col=rainbow(6), pch=16)
legend(x="bottomright", cex=0.8, legend=sites@data$Site_id, col=rainbow(6), bg="white", pch=16)
compassRose(x="topleft", cex=0.9)
scalebar(d=1000, x=c(356000, 516000), type="bar", cex=0.9, below="m", divs=5)
title(sub="Fig1: Morland Beck, Eden DTC (10.1 sq.km),showing locations of automated weather,\n
  flow and water quality monitoring sites and approx location of channel network.",
  cex.sub=0.8, adj=0)
  detach(mor)



## End(Not run)
```

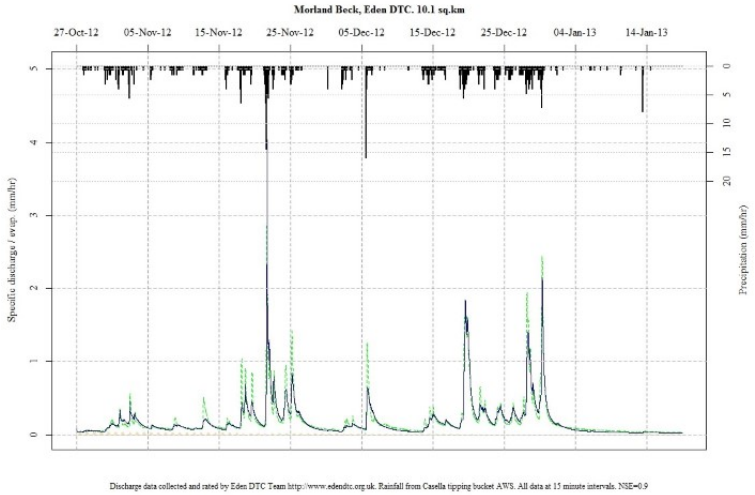
mor.demo.run *Demo run for Morland test catchment*

Description

Manually calibrated run for the 10 sq. km test catchment contained within the Eden DTC, NW England.

Details

This is a run using data for the Newby outlet station and the Tipping Bucket Gauge. 10 HSUs obtained by splitting catchment according to upslope area. Channel width applied = 2m. DRN from topographic analysis using SAGA GIS (http://www.saga-gis.org). NSE using 3 min time step = 0.87.



See documentation for the supplied project for more details and references related to this data set.

Examples

```
## Not run:
data(mor_demo_run)
disp.run(mor.demo.run)

## End(Not run)
```

rebuild.disc *Rebuild an existing discretisation*

Description

Rebuilds all the files for the specified discretisation from scratch and / or adds new cut levels

Usage

```
rebuild.disc(proj, i.disc = 1, disc = NULL, dn = disc$dir, what = "*.dat|*.tif", ...)
```

Arguments

proj	A Dynamic TOPMODEL project
i.disc	The index of the discretisation to rebuild
disc	A discretisation to use if i.disc not supplied
dn	Where to output new files.
what	List of files to delete and rebuild.
...	Any parameters not matching the above will be added to the list of cuts to apply to the discretisation

Value

The project containing the updated discretisation

Author(s)

Peter Metcalfe <p.metcalfe@lancs.ac.uk>

Examples

```
## Not run:
require(dynatopmodel)
data(mor)
# rebuild the test discretisation
rebuild.disc(mor)
# show
dir(mor$disc[[1]]$dir)

## End(Not run)
```

---

run.dtm *Run a Dynamic TOPMODEL simulation*

Description

Run a simulation directly without loading a project.

Usage

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

Arguments

groups	HRU data frame
weights	Flux distribution matrix
rain	Time series (xts or vector) rainfall input (m/hr)
qobs	Optional time series (xts or vector) observed specific discharge (m/hr)
qt0	Initial specific discharge (m/hr)
pe	Optional time series (xts or vector) of potential evapotranspiration specific discharge (m/hr)
dt	Desired time step
ntt	Number of inner time steps
ichan	Channel identifiers
i.out	Channel output reach identifier
vchan	Used only if no values specified in HRU table
vof	Not used
qmax	Maximum discharge (used for display only)
routing	Routing table
reaches	Not currently used
reservoirs	Not currently used
sim.start	Simulation start date-time
sim.end	Simulation start date-time
disp.par	List of output display parameters
run.par	Not used

Details

run.proj is a more convenient way of running simulations as it deals with aggregating the input series to the correct intervals

Value

qobs	time series of observed specific discharges across simulation period (if supplied) (m/hr)
qsim	time series of simulated specific discharges (m/hr)
rain	time series of input rainfall (m/hr)
evap	time series of supplied potential and calculated actual evapotranspiration (m/hr)
fluxes	A list of HRU specific fluxes at each time step (all in m/hr): $qbf: subsurface output (downslope) $qin: subsurface input (upslope) $qof: overland flow $uz: unsaturated gravity drainage into water table
storages	A list of HRU storages / deficits (all in m): $ex: surface excess storage $srz: root zone storage $sd: subsurface deficit
wb	water balance (m)
ovf	total overland flow (m)

Author(s)

Peter Metcalfe <p.metcalfe@lancs.ac.uk>

See Also

run.proj

`run.gof` *Quantatitive goodness of fit measures*

Description

Return a selection of efficiency measures for simulated vs observed values

Usage

```
run.gof(qsim, qobs, what = c("NSE", "R2"), digits = 3)
```

Arguments

`qsim`	simulated values: any series that can be cast to a vector
`qobs`	observed values (same length and units as simulated values)
`digits`	number of decimal places in the results
`what`	A list of character vectors giving the measures to output. Options are: R2: Coefficient of Determination rSD: Ratio of Standard Deviations NSE: Nash-Sutcliffe efficiency mNSE: Modified Nash-Sutcliffe efficiency rNSE: Relative Nash-Sutcliffe efficiency m: modified Index of Agreement d: Index of Agreement me: Mean Error mae: Mean Absolute Error rms: Root Mean Square Error nrms: Normalized Root Mean Square Error r: rPearson product-moment correlation coefficient r.Spearman: Spearman Correlation coefficient rSD: Ratio of Standard Deviations cp: Coefficient of Persistence pbias: Percent Bias KGE: Kling-Gupta efficiency bR2: the coef. of determination multiplied by the slope of the linear regression between 'sim' and 'obs' VE: volumetric efficiency

Value

A list comprising the values of the specifed measures calculated using the observed and simulated values

Author(s)

Peter Metcalfe, using routines by Mauricio Zambrano-Bigiarini

References

Boyle, D. P., H. V. Gupta, and S. Sorooshian (2000), Toward Improved Calibration of Hydrologic Models: Combining the Strengths of Manual and Automatic Methods, Water Resour. Res., 36(12), 3663-3674

Criss, R. E. and Winston, W. E. (2008), Do Nash values have value? Discussion and alternate proposals. Hydrological Processes, 22: 2723-2725. doi: 10.1002/hyp.7072

Gupta, Hoshin V., Harald Kling, Koray K. Yilmaz, Guillermo F. Martinez. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. Journal of Hydrology, Volume 377, Issues 1-2, 20 October 2009, Pages 80-91. DOI: 10.1016/j.jhydrol.2009.08.003. ISSN 0022-1694

Mauricio Zambrano-Bigiarini (2014). hydroGOF: Goodness-of-fit functions for comparison of simulated and observed hydrological time series. R package version 0.3-8. http://CRAN.R-project.org/package=hydroGOF

Yilmaz, K. K., H. V. Gupta, and T. Wagener (2008), A process-based diagnostic approach to model evaluation: Application to the NWS distributed hydrologic model, Water Resour. Res., 44, W09417, doi:10.1029/2007WR006716

See Also

run.sets

Examples

```
## Not run:
  require(dynatopmod)
  data(mor_demo_run)
  with(mor.demo.run, run.gof(qsim, qobs))

## End(Not run)
```

---

run.proj *Run a Dynamic TOPMODEL simulation*

Description

Runs a single discretisation associated with the given project, dealing with aggregation of time series input to a desired time step and checking for missing and invalid input.

Usage

```
run.proj(proj, idisc = 1, disc = NULL, qobs = NA, rain = NULL,
         start = NULL, end = NULL, disp = F, show.stats = T,
         run.tm = F, ...)
```

Arguments

proj	Project wrapper comprising, minimally, rainfall data and at least one discretisation.
idisc	Index of discretisation to run the data against.
disc	Run this discretisation in preference to that specified by idisc.
qobs	Observed specific discharges, used in preference to any loaded with the project. Useful if running simulation repeatedly and using the previous results as comparisons.
rain	Optional rainfall data (m/hr) overwriting project's data
start	Simulation start time. Use project setting if not supplied or infer from supplied input data.
end	Simulation end time. Use project setting if not supplied or infer from supplied input data.
disp	Display the results after run
show.stats	Show efficiences at end of run
run.tm	Show run duration (s) at end of run.
...	Any other parameters that be passed to run.dtm

Details

Dealing with the input to a simulation can be complicated, particularly if input have differing frequencies. The procedudre conveniently runs the simulation given the data contained in a Dynamic TOPMODEL project wrapper and handles aggregating any time series input to a common frequency and checking for missing or invalid data.

Value

qobs	time series of observed specific discharges across simulation period (if supplied) (m/hr)
qsim	time series of simulated specific discharges (m/hr)
rain	time series of input rainfall (m/hr)
evap	time series of supplied potential and calculated actual evapotranspiration (m/hr)
fluxes	A list of HRU specific fluxes at each time step (all in m/hr): $qbf: subsurface output (downslope) $qin: subsurface input (upslope) $qof: overland flow $uz: unsaturated gravity drainage into water table
storages	A list of HRU storages / deficits (all in m): $ex: surface excess storage $srz: root zone storage $sd: subsurface deficit
wb	water balance (m)
ovf	total overland flow (m)

Author(s)

Peter Metcalfe <p.metcalfe@lancs.ac.uk>

See Also

disp.run

Examples

```
## Not run:
# morland demo catchment
data(mor)
run<-run.proj(mor)
disp.run(run)


## End(Not run)
```

---

run.sets *Run model against an ensemble of parameter sets*

Description

Run the model for a particular project against the parameter values taken from a table (data frame), generated e.g by build.calib.sets

Usage

```
run.sets(proj, disc = proj$disc[[1]], calib.set, apply.to = 1:nrow(disc$groups),
  ichan = disc$ichan, fn = NULL)
```

Arguments

proj	A Dynamic TOPMODEL project wrapper
disc	The discretisation of the project to apply parameters to and run.
calib.set	Calibration set (Latin Hypercube) generated by build.calib.set
apply.to	Indicies of HRUs to which to apply values. Other HRU will keep existing values across runs.
ichan	Channel identifiers
fn	optional output filename for results.

Details

The project and discretisation will be run using every set in the input table. If fn is suplied then the results of each run will be appended to that file on completion.

Value

A data frame comprising the results of the multiple runs, including the parameters used, the water balance and efficiency measures. See run.gof for the meaning of these values.

Note

Efficiency measures are generated using the hydroGOF package by Mauricio Zambrano Bigiarini. For details of the output see the documenatation for that package and references within.

Author(s)

Peter Metcalfe <p.metcalfe@lancs.ac.uk>

References

Boyle, D. P., H. V. Gupta, and S. Sorooshian (2000), Toward Improved Calibration of Hydrologic Models: Combining the Strengths of Manual and Automatic Methods, Water Resour. Res., 36(12), 3663-3674

Gupta, Hoshin V., Harald Kling, Koray K. Yilmaz, Guillermo F. Martinez. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. Journal of Hydrology, Volume 377, Issues 1-2, 20 October 2009, Pages 80-91. DOI: 10.1016/j.jhydrol.2009.08.003. ISSN 0022-1694

Criss, R. E. and Winston, W. E. (2008), Do Nash values have value? Discussion and alternate proposals. Hydrological Processes, 22: 2723-2725. doi: 10.1002/hyp.7072

Yilmaz, K. K., H. V. Gupta, and T. Wagener (2008), A process-based diagnostic approach to model evaluation: Application to the NWS distributed hydrologic model, Water Resour. Res., 44, W09417, doi:10.1029/2007WR006716

See Also

build.calib.set run.gof run.proj

Examples

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

 # load the Morland DTC project
 data(mor)

 # initial parameter values
 hru <- mor$disc[[1]]$groups

 # parameter value ranges
 m <-seq(0.005, 0.01, by=0.001)
 ln_t0<-seq(11, 12, by=0.1)

 # explicit values for less sensitive parameters
 td <- c(0.5, 1, 10)
 vchan <- seq(1500, 5500, by=1000)

 # build a calibration set
 calib.set <- build.calib.set(groups=hru,
                              list(td=td, vchan=vchan, ln_t0=ln_t0, m=m))

 cat("No. calibration sets = ", nrow(calib.set), "\n")

 res <- run.sets(mor, calib.set=calib.set)
```

```
## End(Not run)
```

`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 = T, atb = F, deg = 0.1)
```

Arguments

`dem`	elevation raster (m), with equal sized cells and a projected coordinate system
`log`	return log of values.
`atb`	return a two-layer raster comprising both the upslope contributing area and the topographic wetness index ln(a/tan(beta))
`deg`	minimum slope to define a sink (degrees)

Details

For each point in the DEM, this calculates the total uphill area per unit contour draining downslope.

Value

A raster of the same extent and resolution as the input with one or two bands: the first the (log) of the upslope contributing area per unit contour length (m^2/m) and, if specified, the topographic wetness index ln(a)/tan(beta).

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(dynatopmod)
data(mor)
a.atb <- upslope.area(mor$dem, atb=T)
plot(a.atb)

## End(Not run)
```

`write.disc` *Write files from a catchment discretisation to a disk location*

Description

This can be used to save the files for the given discretisation to a new location or to overwrite the files alreday associated. This may be necessary due to e.g. a rebuild due to a changed channel width

Usage

```
write.disc(disc, dn = disc$dir)
```

Arguments

`disc`	Discretisation
`dn`	Optional. New disk location for files. Will be created if not extant.

Value

The discretisation with its attributes updated to reflect its new location.

Author(s)

Peter Metcalfe <p.metcalfe@lancs.ac.uk>

See Also

`rebuild.disc`

`write.proj` *Write a project contents to disk*

Description

Writes a Dynamic TOPMODEL project data to files in a specific disk location

Usage

```
write.proj(proj, dir = proj$dir, disc.dir = file.path(dir,"disc"),
  obs.dir = file.path(dir, "obs"),
  save.disc = T, save.obs = F)
```

Arguments

`proj`	Dynamic TOPMODEL project, a list.
`dir`	Output location. Will be created recursively if not already extant
`save.obs`	Write any observation to a subdirectory "obs" of the output location
`save.disc`	Save each discretisation to the appropriate location
`disc.dir`	Specify an alternative location for discretisations
`obs.dir`	Alternative location for observations

Value

The project with its location (containing in $dir) and those of its discretisations updated.

Author(s)

Peter Metcalfe <p.metcalfe@lancs.ac.uk>

Examples

```
  ## Not run:
  require(dynatopmod)
  # Load data for morland catchment
  data(mor)
  # write to a new temporary directory
  dn <- file.path(Sys.getenv("TMP"), "morland")
  write.proj(mor, dn)
  # examine contents
  dir(dn)

## End(Not run)
```

Index