Package ‘CityWaterBalance’

June 16, 2017

Title Track Flows of Water Through an Urban System

Version 0.1.0

Description Retrieves data and estimates unmeasured flows of water through the urban network. Any city may be modeled with preassembled data, but data for US cities can be gathered via web services using this package and dependencies 'geoknife' and 'dataRetrieval'.

Depends R (>= 3.0.0)

Imports dataRetrieval, dplyr, EcoHydRology, geoknife, graphics, grDevices, lubridate, reshape2, stats, tgp, utils, xts, zoo

License CC0

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Laura Erban [aut, cre]

Maintainer Laura Erban <erban.laura@epa.gov>

Repository CRAN

Date/Publication 2017-06-16 19:45:28 UTC

R topics documented:

`CityWaterBalance` *Tracks flows of water through the urban system*

Description

This function tracks flows of water as they move through pathways and storages within the urban system. Data can be in any self-consistent units.

Usage

```
CityWaterBalance(data, p, print = TRUE)
```

Arguments

`data` xts or zoo object with date index and columns of data for:
precipitation (prcp)
evapotranspiration (et)
streamflow in (inflow)
streamflow out (outflow)
water supply imports (ws_imports)
other imports (etc_imports)
surface water for industrial uses (sw_ind)
surface water withdrawals for potable use (sw_pot)
surface water withdrawals for nonpotable use (sw_npot)
groundwater withdrawals for industrial uses (gw_ind)
groundwater withdrawals for potable use (gw_pot)
groundwater withdrawals for nonpotable use (gw_npot)
deep groundwater recharge (dgr)
combined sewer overflow events (cso)
wastewater treatment plant effluent (wtpe)
runoff estimate (runoff)
baseflow estimate (baseflow)

`p` list of fixed parameter values for:
fraction of pet lost to interception (interc)
multiplier for et (et_mult)
multiplier for outflow (flow_mult)
fraction of area that is open water (open_wat)
multiplier for runoff (run_mult)

fraction of runoff diverted to sewers (run_css)
multiplier for baseflow (bf_mult)
fraction of potable water supply lost to leaks (nonrev)
fraction of industrial water that evaporates (ind_evap)
fraction of potable use that returns to sewers (wast_gen)
fraction of potable use that evaporates (pot_atm)
fraction of nonpotable use that infiltrates (npot_infilt)
fraction of wastewater that evaporates from sludge (slud_evap)
fraction of wastewater effluent from gw infiltration (leak_css)
fraction of groundwater from deep, confined aquifers (dgw)
multiplier for deep pumping replaced by lateral flow (dgw_rep)

`print`	option to print messages

Value

list of dataframes:

`all_flows`	all flows
`state_vars`	state variables
`global_balance`	global water balance
`internal_balance`	internal water balance

Examples

```
p <- list("interc" = 0, "et_mult" = 1, "flow_mult" = 1,
         "open_wat" = 0.02, "run_mult" = 3.378, "run_css" = 0.35,
         "bf_mult" = 1, "nonrev" = 0.08, "ind_evap" = 0.012,
         "wast_gen" = 0.85, "pot_atm" = 0.13, "npot_infilt" = 0.5,
         "slud_evap" = 0, "leak_css" = 0.05,"dgw" = 0.5, "dgw_rep" = 0.5)
m <- CityWaterBalance(cwb_data, p)
```

`combineStreamflow` *Combines streamflow records to estimate total flows*

Description

This function combines streamflow data from multiple gages, averaging records according to a multiplier list

Usage

```
combineStreamflow(flowlist, mult, approx = FALSE)
```

Arguments

`flowlist`	list of flow data, output of getStreamflow or gapfillStreamflow
`mult`	list of multipliers, one for each gage
`approx`	option to interpolate missing values

Value

total flow for each timestep (as xts)

Examples

```
gages <- c('05551540', '05552500')
flow <- getStreamflow('2000-01-01', '2010-12-31', gages)
flow <- gapfillStreamflow(flow, list(c(gages[1], gages[2])))
flow <- combineStreamflow(flow, c(0.5, 0.5))
```

`combineWaterUse` *Combine water use data for urban system into functional flows*

Description

This function takes county-level USGS water use data gathered by getWaterUse and aggregates them for whole urban system according to flows in CityWaterBalance

Usage

```
combineWaterUse(start, end, wu)
```

Arguments

`start`	start date in format 'YYYY-MM-DD'
`end`	end date in format 'YYYY-MM-DD'
`wu`	list of dataframes output by getWaterUse

Value

wu_flows list of xts objects aggregating water withdrawals (MGal) by:

`sw_ind`	surface water for industrial use
`sw_pot`	surface water for potable use
`sw_npot`	surface water for nonpotable use
`gw_ind`	groundwater for industrial use
`gw_pot`	groundwater for potable use
`gw_npot`	groundwater for nonpotable use

Examples

```
## Not run:
wu <- getWaterUse(c('IL'), c('Cook', 'Lake'))
wu_flows <- combineWaterUse('2000-01-01', '2015-01-01', wu)

## End(Not run)
```

cwb_data *Example dataset for CityWaterBalance*

Description

Contains monthly flows data for inputs to CityWaterBalance function

Usage

```
cwb_data
```

Format

A zoo series with 120 rows and 18 variables. All values are fluxes (flow/area) in units of mm/month.

prcp precipitation

et evapotranspiration

pet potential evapotranspiration

inflow streamflow in

outflow streamflow out

sw_ind surface water withdrawals for industrial use

sw_pot surface water withdrawals for potable use

sw_npot surface water withdrawals for nonpotable use

gw_ind groundwater withdrawals for industrial use

gw_pot groundwater withdrawals for potable use

gw_npot groundwater withdrawals for nonpotable use

ws_imports water supply imports

etc_imports other imports

wtpe wastewater treatment plant effluent

dgr deep groundwater recharge

cso combined sewer overflow

runoff runoff to surface waters and sewers

baseflow baseflow from groundwater to surface water

...

gapfillStreamflow *Fill gaps in streamflow records*

Description

This function fills in gappy streamflow records using a median of discharge ratio from master gages.

Usage

```
gapfillStreamflow(flowlist, paired_gages)
```

Arguments

flowlist	list of gage data and information from getStreamflow
paired_gages	list of gage pairs where pair <- c(gappy,master)

Value

list of:

sites	list of gage site names
site_num	list of gage numbers
flows	xts of gap-filled daily average discharge

Examples

```
gages <- c('05551540', '05552500')
flow <- getStreamflow('2000-01-01', '2010-12-31', gages)
flow <- gapfillStreamflow(flow, list(c(gages[1], gages[2])))
```

getAtmoFlows *Gather all atmospheric data time series*

Description

This function gathers precipitation, evapotranspiration and temperature data from the USGS Geo Data Portal (GDP), and calculates potential evapotranspiration. NOTE: Shapefile for geometry must already be uploaded to the GDP.

Usage

```
getAtmoFlows(start, end, geometry, att, val = NA, latitude)
```

Arguments

`start`	start date in format 'YYYY-MM-DD'
`end`	end date in format 'YYYY-MM-DD'
`geometry`	name of geometry as displayed in GDP
`att`	attribute of geometry as displayed in GDP
`val`	values of attribute as displayed in GDP
`latitude`	(degrees)

Value

xts object consisting of:

`prcp`	precipitation (mm)
`et`	evapotranspiration (mm)
`tmax`	max temperature (deg C)
`tmin`	min temperature (deg C)
`pet`	potential evapotranspiration (mm)

Examples

```
## Not run:
atm <- getAtmoFlows('2010-01-01', '2010-12-31', 'sample:Counties', 'STATE','RI',
41.5801)

## End(Not run)
```

`getEvapotranspiration` *Assemble an evapotranspiration times series*

Description

This function retrieves monthly actual evapotranspiration for a given area from SSEBop model output hosted by the USGS Geo Data Portal (GDP)

Usage

```
getEvapotranspiration(start, end, geometry, att, val = NA)
```

Arguments

`start`	start date in format 'YYYY-MM-DD'
`end`	end date in format 'YYYY-MM-DD'
`geometry`	name of geometry as displayed in GDP
`att`	attribute of geometry as displayed in GDP
`val`	values of attribute as displayed in GDP

Value

monthly evapotranspiration, averaged spatially over geometry

Examples

```
## Not run:
et = getEvapotranspiration('2010-01-01', '2010-12-31', 'sample:Counties',
'STATE', 'RI')

## End(Not run)
```

`getPrecipitation` *Assemble a precipitation times series*

Description

This function retrieves monthly precipitation, minimum and maximum temperatures for a given area from PRISM model output hosted by the USGS Geo Data Portal (GDP)

Usage

```
getPrecipitation(start, end, geometry, att, val = NA)
```

Arguments

`start`	start date in format 'YYYY-MM-DD'
`end`	end date in format 'YYYY-MM-DD'
`geometry`	name of geometry as displayed in GDP
`att`	attribute of geometry as displayed in GDP
`val`	values of attribute as displayed in GDP

Value

monthly precipitation, averaged spatially over geometry

Examples

```
## Not run:
prcp <- getPrecipitation('2010-01-01', '2010-12-31', 'sample:Counties', 'STATE',
'RI')

## End(Not run)
```

getSolutions *Evaluate parameter uncertainty*

Description

This function searches for acceptable model solutions within the uncertainty parameters and long-term storage balances using Latin hypercubes.

Usage

```
getSolutions(data, p, n, tol = 0.01, interc = c(0, 0.05), et_mult = c(1,
  1.1), flow_mult = c(1, 1.1), open_wat = c(0.01, 0.1), run_mult = c(1,
  5), run_css = c(0.1, 1), bf_mult = c(0.5, 1.5), nonrev = c(0.05, 0.2),
  ind_evap = c(0.01, 0.02), wast_gen = c(0.75, 0.9), pot_atm = c(0.1,
  0.15), npot_infilt = c(0.25, 0.75), slud_evap = c(0, 0),
  leak_css = c(0.05, 0.25), dgw = c(0.5, 0.5), dgw_rep = c(0, 1),
  global_bal = c(-500, 500), sw_bal = c(-500, 500), css_bal = c(-500,
  500), sgw_bal = c(-500, 500), dgw_bal = c(-500, 500))
```

Arguments

`data`	xts or zoo object. See CityWaterBalance function for details.
`p`	list of initial parameter values. See CityWaterBalance function or the inputs below for descriptions.
`n`	integer number of initial parameter sets to search
`tol`	tolerance acceptable difference mean flow solutions
`interc`	vector of min and max fraction of pet lost to interception
`et_mult`	vector of min and max multiplier for et
`flow_mult`	vector of min and max multiplier for outflow
`open_wat`	vector of min and max fraction of area that is open water
`run_mult`	vector of min and max multiplier for runoff
`run_css`	vector of min and max fraction of runoff diverted to sewers
`bf_mult`	vector of min and max multiplier for baseflow
`nonrev`	vector of min and max fraction of potable water supply lost to leaks
`ind_evap`	vector of min and max fraction of industrial use that evaporates
`wast_gen`	vector of min and max fraction of potable use that returns to sewers
`pot_atm`	vector of min and max fraction of potable use that evaporates
`npot_infilt`	vector of min and max fraction of nonpotable use that infiltrates
`slud_evap`	vector of min and max fraction of wastewater that evaporates from sludge
`leak_css`	vector of min and max fraction of wastewater effluent from gw infiltration
`dgw`	vector of min and max fraction of groundwater from deep, confined aquifers

`dgw_rep` vector of min and max multiplier for deep groundwater pumping replacement

`global_bal` vector of min and max acceptable global water balance values, cumulative over model run

`sw_bal` vector of min and max acceptable surface water balance values, cumulative over model run

`css_bal` vector of min and max acceptable sewer system water balance values, cumulative over model run

`sgw_bal` vector of min and max acceptable shallow groundwater balance values, cumulative over model run

`dgw_bal` vector of min and max acceptable deep groundwater balance values, cumulative over model run

Details

The function creates n parameter sets using a Latin hypercube. It runs ‘CityWaterBalance()‘ with each set, accepting solutions that meet user-defined criteria for storage balances. It then computes the mean of flow solutions, and doubles n until the difference between the means of old and new solutions is less than tol for all flows. Defaults for parameter value ranges are set to reasonable values, but they should be reconsidered for each application. Defaults for storage balances are set high to allow for solution discovery, however, acceptable values must be determined on a case-by-case basis.

Value

out numeric solutions

Examples

```
## Not run:
data <- cwb_data
data$cso <- 0
p <- list("interc" = 0,"et_mult" = 1,"flow_mult" = 1, "open_wat" = 0.02,
          "run_mult" = 3.378, "run_css" = 0.35, "bf_mult" = 1,
          "nonrev" = 0.08, "ind_evap" = 0.012, "wast_gen" = 0.85,
          "pot_atm" = 0.13, "npot_infilt" = 0.5, "slud_evap" = 0,
          "leak_css" = 0.05, "dgw" = 0.5, "dgw_rep" = 0.5)
out <- getSolutions(data, p, 10, 0.1)

## End(Not run)
```

`getStreamflow` *Gather time series of streamflow data*

Description

This function gathers daily average streamgauge data for a group of gauges from USGS NWIS

Usage

```
getStreamflow(start, end, gages)
```

Arguments

`start`	start date in format 'YYYY-MM-DD'
`end`	end date in format 'YYYY-MM-DD'
`gages`	list of USGS gauge numbers

Value

list of:

`sites`	list of gauge site names
`site_num`	list of gauge numbers
`flows`	xts of daily average discharge (cfs)

Examples

```
flow <- getStreamflow('2000-01-01', '2010-12-31', c('05551540', '05552500'))
```

`getWaterUse` *Gather time series of water use data*

Description

This function gathers and summarizes water use data from USGS NWIS by county, year, source (surface or groundwater) and quality (fresh or saline)

Usage

```
getWaterUse(states, counties, years = "ALL")
```

Arguments

`states`	list of state abbreviations
`counties`	list of county name lists for each state
`years`	list of years of available data

Value

list of dataframes of water withdrawals (MGD) by use category, one for each water source and quality:

`swf`	surface water, fresh
`gwf`	groundwater, fresh
`sws`	surface water, saline
`gws`	groundwater, saline

Examples

```
wu <- getWaterUse(c('IL'), c('Cook', 'DeKalb'))
```

mergeData *Merge data sources into input for CityWaterBalance*

Description

This function converts units and merges data needed by CityWaterBalance. All inputs must represent the same time intervals. Outputs are fluxes (mm/month) over study area.

Usage

```
mergeData(area, atm, inflow, outflow, wu, ws_imports = NULL,
  etc_imports = NULL, wweff = NULL, dgr = NULL, cso = NULL,
  runoff = NULL, baseflow = NULL)
```

Arguments

`area`	numeric study area (sq km)
`atm`	xts of atmospheric data, from getAtmoFlows (mm/month)
`inflow`	xts of daily streamflow into area (cfs)
`outflow`	xts of daily streamflow out of area (cfs)
`wu`	xts of water use, from combineWaterUse (MGal/month)
`ws_imports`	xts of imports for water supply (MGal/month)
`etc_imports`	xts of other imports to surface water (MGal/month)
`wweff`	xts of wastewater effluent (MGD)
`dgr`	xts of deep groundwater recharge (mm/month)
`cso`	xts of cso events (mm/month)
`runoff`	xts of runoff (mm/month)
`baseflow`	xts of baseflow (mm/month)

Value

all fluxes (as xts) for each timestep (mm/month)

Examples

```
## Not run:
start <- "2010-01-01"
end <- "2010-12-31"
area <- 2707
atm <- getAtmoFlows(start, end, 'sample:Counties', 'STATE', 'RI', 41.5801)
inflow <- getStreamflow(start, end, c("01112500"))
inflow <- combineStreamflow(inflow, c(1))
outflow <- getStreamflow(start, end, c("01113895", "01114000", "01117000",
"01118500"))
outflow <- combineStreamflow(outflow, c(1, 1, 1, 1))
wu <- getWaterUse(c('RI'), 'ALL')
wu <- combineWaterUse(start, end, wu)
data <- mergeData(area, atm, inflow, outflow, wu)

## End(Not run)
```

`plotStreamflow` *Plot discharge at a set of gages*

Description

This function plots streamflow data

Usage

```
plotStreamflow(flowlist)
```

Arguments

`flowlist`	list object of flow data output from, e.g., getStreamflow

Value

plot

Examples

```
flow <- getStreamflow('2000-01-01', '2010-12-31', c('05551540', '05552500'))
plotStreamflow(flow)
```

plotWaterBalance *Plot components of the urban water balance*

Description

This function plots input to or output from ‘CityWaterBalance‘.

Usage

```
plotWaterBalance(data, yl = "Your y-axis label", annual = FALSE)
```

Arguments

`data`	xts or zoo object
`yl`	y-axis label
`annual`	flag indicating whether to plot annual totals

Value

plot

Examples

```
global_flows <- cwb_data[,c(1,2,4,5)]
plotWaterBalance(global_flows)
p <- list("interc" = 0,"et_mult" = 1,"flow_mult" = 1,"open_wat" = 0.02,
   "run_mult" = 3.378, "run_css" = 0.35, "bf_mult" = 1, "nonrev"=0.08,
   "ind_evap"=0.012,"wast_gen" = 0.85,"pot_atm" = 0.13,"npot_infilt" = 0.5,
   "slud_evap" = 0,"leak_css" = 0.05,"dgw" = 0.5, "dgw_rep" = 0.5)
m <- CityWaterBalance(cwb_data,p)
f <- m$all_flows
css_flows <- f[,c(3,12,27,30,34)]
plotWaterBalance(css_flows)
```

Index