Design Document: Documentation of the wrv R package--Pre- and post-processing program for the groundwater-flow model of the Wood River Valley

By J.C. Fisher, USGS

Design document description and purpose

The U.S. Geological Survey (USGS), in collaboration with the Idaho Department of Water Resources (IDWR) is constructing a MODFLOW numerical groundwater-flow model of the Wood River Valley aquifer system in order to simulate potential anthropogenic and climatic effects on groundwater and surface-water resources. This model will serve as a tool for water-rights administration and water-resource management and planning. The study will be conducted over a 3-year period from late 2012 until model and report completion in 2015.

One of the goals of the modeling study is to develop the model in an open and transparent manner. To this end, a Technical Advisory Committee was formed to provide for transparency in model development and to serve as a vehicle for stakeholder input. Technical representation was solicited by the IDWR and includes such interested parties as water-user groups and current USGS cooperating organizations in the Wood River Valley.

The design, construction, and calibration of a groundwater-flow model requires a number of decisions such as the number of layers, model cell size, or methodologies used to represent processes such as evapotranspiration or pumpage. While these decisions will be documented in a final USGS report, intermediate decision documents will be prepared in order to facilitate technical discussion and ease preparation of the report. These decision documents should be considered preliminary status reports and not final products.

Design decision

Package ‘wrv’

December 2, 2013

Version 0.1-2

Date 2013-12-02

Title Wood River Valley Groundwater Flow Model

Author Jason C. Fisher

Maintainer Jason C. Fisher <jfisher@usgs.gov>

Depends R (>= 3.0.0), sp, rgdal, raster, igraph, rgeos

Suggests RCurl, knitr, xtable, png

SystemRequirements MODFLOW-USG (>= 1.1)

Description Pre- and post-processing program for the groundwater-flow model of the Wood River Valley.

License file LICENSE

Copyright This software is in the public domain because it contains materials that originally came from the United States Geological Survey, an agency of the United States Department of Interior. For more information, see the official USGS copyright policy at http://www.usgs.gov/visual-id/credit_usgs.html#copyright

URL https://github.com/jfisher-usgs/wrv

BugReports https://github.com/jfisher-usgs/wrv/issues

ByteCompile yes

LazyData yes

VignetteBuilder knitr

R topics documented:

alluvium.bottom 2
aquifer.extent 3
aquitard.extent 4
basalt.extent 4
cities 5
CreateModflowInputFiles 6

ExcludeSmallCellChunks . . . 8
ExportRasterStack . . . 9
ExtractAlongTransect . . . 10
FindConnectedCells . . . 11
lakes . . . 12
land.surface . . . 13
major.streams . . . 14
map.labels . . . 15
PlotMap . . . 16
ReadGeoDataFromURL . . . 19
ReadModflowBinaryFile . . . 20
sink.locations . . . 21
source.locations . . . 22
surface.water . . . 23

Index **24**

`alluvium.bottom` *Pre-Quaternary Bedrock Surface and Top of Quaternary Basalt*

Description

Estimated elevation of the pre-Quaternary bedrock surface and top of Quaternary basalt in the the Wood River Valley aquifer system, South-Central Idaho.

Usage

```
alluvium.bottom
```

Format

An object of class `SpatialGridDataFrame-class`. Each cell on the surface grid represents an elevation in meters above the North American Vertical Datum of 1988 (NAVD 88). Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM). The spatial grid is composed of 2822 rows and 2145 columns, and has cell sizes that are constant at 20 meters by 20 meters.

Source

This dataset is a revised version of Plate 1 in Bartolino and Adkins (2012).

References

Bartolino, J.R., and Adkins, C.B., 2012, Hydrogeologic framework of the Wood River Valley aquifer system, south-central Idaho: U.S. Geological Survey Scientific Investigations Report 2012-5053, 46 p., available at http://pubs.usgs.gov/sir/2012/5053/.

See Also

`readGDAL`

Examples

```
image(alluvium.bottom)
summary(alluvium.bottom)

## Not run: # Commands used to create this dataset:
u <- "https://github.com/jfisher-usgs/wrv/raw/master/inst/extdata/alluvium.bottom.tif"
obj <- ReadGeoDataFromURL(u)
names(obj) <- "alluvium.bottom"
identical(obj, alluvium.bottom)
## End(Not run)
```

---

aquifer.extent *Extent of Aquifer System*

Description

Estimated extent of the Wood River Valley aquifer system, south-central Idaho.

Usage

```
aquifer.extent
```

Format

An object of `SpatialPolygonsDataFrame-class` containing 1 `Polygons`. Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM).

Source

Extent defined by Bartollino and Adkins (2012, Plate 1).

References

Bartolino, J.R., and Adkins, C.B., 2012, Hydrogeologic framework of the Wood River Valley aquifer system, south-central Idaho: U.S. Geological Survey Scientific Investigations Report 2012-5053, 46 p., available at http://pubs.usgs.gov/sir/2012/5053/.

Examples

```
plot(aquifer.extent)
str(aquifer.extent)

## Not run: # Commands used to create this dataset:
u <- "https://github.com/jfisher-usgs/wrv/raw/master/inst/extdata/aquifer.extent.zip"
obj <- ReadGeoDataFromURL(u)
obj <- spTransform(obj, alluvium.bottom@proj4string)
all.equal(obj, aquifer.extent)
## End(Not run)
```

aquitard.extent *Extent of Aquitard*

Description

The extent of the confining unit (aquitard) separating the unconfined aquifer from the underlying confined aquifer in the Wood River Valley.

Usage

```
aquitard.extent
```

Format

An object of `SpatialPolygonsDataFrame-class` containing 2 `Polygons`. Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM).

Source

Extent defined by Moreland (1977, fig. 3 in USGS Open-File report). Moreland shows an outlier by Picabo that is assumed to indicate confined conditions in the basalt and not the lake sediments.

References

Moreland, J.A., 1977, Ground water-surface water relations in the Silver Creek area, Blaine County, Idaho: Boise, Idaho Department of Water Resources, Water Information Bulletin 44, 42 p., 5 plates in pocket, accessed January 31, 2012. Also published as U.S. Geological Survey Open-File report 77-456, 66 p., available at http://pubs.er.usgs.gov/pubs/ofr/ofr77456.

Examples

```
plot(aquitard.extent)
str(aquitard.extent)

## Not run: # Commands used to create this dataset:
u <- "https://github.com/jfisher-usgs/wrv/raw/master/inst/extdata/aquitard.extent.zip"
obj <- ReadGeoDataFromURL(u)
obj <- spTransform(obj, alluvium.bottom@proj4string)
all.equal(obj, aquitard.extent)
## End(Not run)
```

basalt.extent *Extent of Basalt*

Description

The extent of the basalt underlying the alluvial Wood River Valley aquifer system.

Usage

```
basalt.extent
```

Format

An object of SpatialPolygonsDataFrame-class containing 1 Polygons. Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM).

Source

Extent defined by Bartollino and Adkins (2012, Plate 1).

References

Bartolino, J.R., and Adkins, C.B., 2012, Hydrogeologic framework of the Wood River Valley aquifer system, south-central Idaho: U.S. Geological Survey Scientific Investigations Report 2012-5053, 46 p., available at http://pubs.usgs.gov/sir/2012/5053/.

Examples

```
plot(basalt.extent)
str(basalt.extent)

## Not run: # Commands used to create this dataset:
u <- "https://github.com/jfisher-usgs/wrv/raw/master/inst/extdata/basalt.extent.zip"
obj <- ReadGeoDataFromURL(u)
obj <- spTransform(obj, alluvium.bottom@proj4string)
all.equal(obj, basalt.extent)
## End(Not run)
```

---

cities — *Cities and Towns*

Description

Cities and towns in the Wood River Valley and surrounding areas.

Usage

```
cities
```

Format

An object of SpatialPointsDataFrame-class containing 9 points. Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM).

Source

Idaho Department of Water Resources (IDWR).

Examples

```
spplot(cities)
str(cities)

## Not run: # Commands used to create this dataset:
u <- "https://github.com/jfisher-usgs/wrv/raw/master/inst/extdata/cities.zip"
obj <- ReadGeoDataFromURL(u)
obj <- spTransform(obj, alluvium.bottom@proj4string)
identical(obj, cities)
## End(Not run)
```

`CreateModflowInputFiles`

Create MODFLOW Input Files

Description

Generate and write input files for a MODFLOW simulation of groundwater flow in the Wood River Valley (WRV) aquifer system.

Usage

```
CreateModflowInputFiles(rs.model, id, p.run, perlen, hk, vani)
```

Arguments

`rs.model`	RasterStack; a collection of `RasterLayer` objects with the same extent and resolution, see 'Details' for required raster layers.
`id`	character; a short identifier for the model run.
`p.run`	character; path name to write model input files.
`perlen`	integer; period length for simulation, in days.
`hk`	numeric; a vector of length 3 giving the horizontal hydraulic conductivity for each hydrogeologic zones (that is alluvium, basalt, and clay), in meters per day.
`vani`	numeric; vertical anisotropy.

Details

Steady-state flow in the WRV aquifer is simulated using the MODFLOW-USG groundwater flow model. This numerical model was chosen for its ability to solve complex unconfined groundwater flow simulations. The solver implemented in MODFLOW-USG incorporates the Newton-Raphson formulation for improving solution convergence and avoiding problems with the drying and rewetting of cells (Niswonger and others, 2011). A structured finite-difference grid was implemented in the model to (1) simplify discretization, (2) keep formats and structures for the MODFLOW-USG packages identical to those of MODFLOW-2005, and (3) allow any MODFLOW post-processor to be used to analyze the results of the MODFLOW-USG simulation (such as Model Viewer).

Model input files are written to path `p.run` and include the Name ('`.nam`'), Basic ('`.ba6`'), Discretization ('`.dis`'), Zone ('`.zon`'), Layer-Property Flow ('`.lpf`'), Flow and Head Boundary ('`.fhb`'), Drain ('`.drn`'), Sparse Matrix Solver ('`.sms`'), and Output Control ('`.oc`') files. See the users guide (*Description of Model Input and Output*) included with the MODFLOW-USG software for details

on input file formats and structures. The Layer-Property Flow file includes options for the calculation of vertical flow in partially dewatered cells. For the WRV model, where there is no indication that perched conditions exist, `CONSTANTCV` and `NOVFC` options can be used to create the most stable solution (Panday and others, 2013, p. 15-16). Options for the Sparse Matrix Solver were set for unconfined simulations by implementing an upstream-weighting scheme with Newton-Raphson linearization, Delta-Bar-Delta under-relaxation, and the χMD solver of Ibaraki (2005).

The required raster layers in the raster stack `rs.model` include:

lay1.ibound The boundary variable in model layer 1 where values `<0` is a constant hydraulic head, `=0` is inactive, and `>0` is active.

lay2.ibound The boundary variable in model layer 2.

lay3.ibound The boundary variable in model layer 3.

lay1.strt The initial (starting) hydraulic head in model layer 1, in meters above the North American Vertical Datum of 1988 (NAVD 88).

lay2.strt The initial hydraulic head in model layer 2.

lay3.strt The initial hydraulic head in model layer 3.

lay1.top The elevation at the top of model layer 1 (land surface), in meters above the NAVD 88.

lay1.bottom The elevation at the bottom of model layer 1, in meters above the NAVD 88.

lay2.bottom The elevation at the bottom of model layer 2.

lay3.bottom The elevation at the bottom of model layer 3.

lay1.zones The hydrogeologic zones in model layer 1 where values `=1` is alluvium, `=2` is basalt, and `=3` is clay (aquitard).

lay2.zones The hydrogeologic zones in model layer 2.

lay3.zones The hydrogeologic zones in model layer 3.

Value

Returns an object of class `NULL`.

Author(s)

J.C. Fisher

References

Ibaraki, M., 2005, χMD User's guide-An efficient sparse matrix solver library, version 1.30: Columbus, Ohio State University School of Earth Sciences.

Niswonger, R.G., Panday, Sorab, and Ibaraki, Motomu, 2011, MODFLOW-NWT, A Newton formulation for MODFLOW-2005: U.S. Geological Survey Techniques and Methods 6-A37, 44 p., available at http://pubs.usgs.gov/tm/tm6a37/.

Panday, Sorab, Langevin, C.D., Niswonger, R.G., Ibaraki, Motomu, and Hughes, J.D., 2013, MODFLOW-USG version 1: An unstructured grid version of MODFLOW for simulating groundwater flow and tightly coupled processes using a control volume finite-difference formulation: U.S. Geological Survey Techniques and Methods, book 6, chap. A45, 66 p., available at http://pubs.usgs.gov/tm/06/a45/.

See Also

`cat`, `write.table`

Examples

```
## Not run:
load(file = file.path(getwd(), "20130926130613", "Model", "raster.stack.rda"))
perlen <- round(15 * 365.242)  # 15 years converted to days
hk <- c(30.48, 86.4, 8.64e-7)  # ave. values for alluvium, basalt, and clay
vani <- 1000                   # educated guess for global value
CreateModflowInputFiles(rs, "wrv_ss_mfusg", tempdir(), perlen, hk, vani)
## End(Not run)
```

ExcludeSmallCellChunks

Exclude Small Cell Chunks

Description

A cell chunk is defined as a group of connected cells with non-missing values. This function identifies cell chunks in a single raster grid layer. The chunk with the largest surface area is preserved and all others removed.

Usage

```
ExcludeSmallCellChunks(r)
```

Arguments

r	RasterLayer; raster grid layer with cell values.

Value

The raster grid layer r with cell values in the smaller cell chunks set to NA.

Author(s)

J.C. Fisher

See Also

clump

Examples

```
r <- raster(ncols = 10, nrows = 10)
set.seed(0)
r[] <- round(runif(ncell(r)) * 0.7)
r <- clump(r)
plot(r)

r.new <- ExcludeSmallCellChunks(r)
plot(r.new, zlim = range(values(r), na.rm = TRUE))
```

ExportRasterStack *Export Raster Stack*

Description

Write raster-stack layers to a local directory in multiple formats (such as Text, PNG, and GeoTIFF).

Usage

```
ExportRasterStack(p, rs, NAflag = -999)
```

Arguments

p	character; path name to write raster stack.
rs	RasterStack; a collection of RasterLayer objects with the same extent and resolution.
NAflag	numeric; to overwrite the default value used to represent NA in a file.

Details

Three directories ('Text', 'PNG', and 'GeoTIFF') are created in the path p and named after the intended data format for files it will contain. A file is written for each raster layer in the raster stack rs and for each format type. Base names for files are a concatination of layer order in the raster stack, layer name, and an extension consistent with its data format ('.txt', '.png', and '.tif'). An external representation of the RasterStack object is also written to path p and given the name 'raster.stack.rda'.

Value

Returns an object of class NULL.

Author(s)

J.C. Fisher

See Also

dir.create, write.table, png, spplot, writeRaster, save

Examples

```
## Not run:
load(file = file.path(getwd(), "20130926130613", "Data", "raster.stack.rda"))
ExportRasterStack(tempdir(), rs)
## End(Not run)
```

ExtractAlongTransect *Extract Raster Values Along Transect*

Description

Extract values from raster layer(s) along a user defined transect line.

Usage

```
ExtractAlongTransect(r, v, rtn.polygon = FALSE)
```

Arguments

r	RasterLayer or RasterStack; raster layer(s).
v	SpatialPoints; transect line vertices.
rtn.polygon	logical; if TRUE a polygon is returned. See 'Value' for additional information.

Details

The transect line is described using a simple polygonal chain. Transect line vertices and raster layer(s) must be specified in a coordinate reference system.

Value

If r is a RasterLayer or rtn.polygon is FALSE, a list is returned with components of class SpatialPointsDataFrame. These components represent continuous line segments along the transect line. The following variables are specified for each coordinate point in the line segment:

d	numeric; distance along the transect line.
z1	numeric; value of raster layer r1.
z2	numeric; value of raster layer r2.

Alternatively, if r is a RasterStack and rtn.polygon is TRUE, a polygon of class gpc.poly is returned . Georeferencing is absent from the polygon with x representing the distance along the transect line and y representing the raster layer value.

Author(s)

J.C. Fisher

See Also

extract, area.poly

Examples

```
coords <- rbind(c(-100, -90), c(80, 90), c(80, 0), c(40, -40))
crs <- CRS("+proj=longlat +datum=WGS84")
v <- SpatialPoints(coords, proj4string = crs)
r1 <- raster(nrows = 10, ncols = 10, ymn = -80, ymx = 80)
set.seed(0)
r1[] <- runif(ncell(r1))
r1[4, 6] <- NA
plot(r1, xlab = "x", ylab = "y")
lines(SpatialLines(list(Lines(list(Line(coords)), ID = "Transect")), proj4string = crs))
points(v, pch = 21, bg = "grey")
segs <- ExtractAlongTransect(r1, v = v)
for (i in 1:length(segs))
  points(segs[[i]], col = "blue")

xlab <- "Distance along transect"
ylab <- "Value"
xlim <- range(vapply(segs, function(seg) range(seg@data[, "d"]), c(0, 0)))
ylim <- range(vapply(segs, function(seg) range(seg@data[, "z1"]), c(0, 0)))
dev.new()
plot(NA, type = "n", xlab = xlab, ylab = ylab, xlim = xlim, ylim = ylim)
for (i in 1:length(segs))
  lines(segs[[i]]@data[, c("d", "z1")], col = rainbow(length(segs))[i])
n <- length(v)
d <- cumsum(c(0, as.matrix(dist((coordinates(v))))[cbind(1:(n - 1), 2:n)]))
abline(v = d, col = "grey", lty = 2)
mtext(paste0("(", paste(head(coordinates(v), 1), collapse = ", "), ")"), adj = 0)
mtext(paste0("(", paste(tail(coordinates(v), 1), collapse = ", "), ")"), adj = 1)

r2 <- sum(r1, 2.0, na.rm = TRUE)
ply <- ExtractAlongTransect(stack(r1, r2), v = v, rtn.polygon = TRUE)
dev.new()
plot(ply, asp = 100, xlab = xlab, ylab = ylab, poly.args = list(col = "lightblue"))
abline(v = d, col = "grey", lty = 2)
mtext(paste("Cross sectional area =", format(area.poly(ply))))
```

FindConnectedCells *Find Connected Cells Using Minimum Thickness*

Description

Remove model cells that violate a minimum vertical overlap between adjacent cells.

Usage

```
FindConnectedCells(r.bot, r.top, min.overlap)
```

Arguments

`r.bot`	RasterLayer; raster grid layer with cell-values describing the bottom surface (e.g. bottom elevations of model layer).
`r.top`	RasterLayer; raster grid layer with cell-values describing the top surface (e.g. top elevations of model layer).
`min.overlap`	numeric; minimum vertical overlap between adjacent cells.

Details

The spatial extent and cell-size dimension must be identical for `r.bot` and `r.top`. Cells with the most egregious violation of `min.overlap` are eliminated first. This function is computationally expensive and should be revisited to optimize performance.

Value

Returns raster layer with logical values indicating whether a cell is connected.

Author(s)

J.C. Fisher

See Also

`ExcludeSmallCellChunks`

Examples

```
set.seed(0)
r.bot <- raster(ncols = 10, nrows = 10)
r.bot[] <- rnorm(ncell(r.bot), mean = 10)
r.top <- raster(ncols = 10, nrows = 10)
r.top[] <- rnorm(ncell(r.top), mean = 12)
summary(r.top - r.bot)

r <- FindConnectedCells(r.bot, r.top, min.overlap = 0.1)
plot(r, col = c("red", "black"))
```

`lakes` *Lakes and Reservoirs*

Description

Lakes and reservoirs of the Wood River Valley and surrounding areas.

Usage

```
lakes
```

Format

An object of `SpatialPolygonsDataFrame-class` containing 55 `Polygons`. Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM).

Source

Idaho Department of Water Resources (IDWR).

Examples

```
plot(lakes)
str(lakes)

## Not run: # Commands used to create this dataset:
u <- "https://github.com/jfisher-usgs/wrv/raw/master/inst/extdata/lakes.zip"
obj <- ReadGeoDataFromURL(u)
obj <- spTransform(obj, alluvium.bottom@proj4string)
identical(obj, lakes)
## End(Not run)
```

`land.surface` *Topography of Land Surface*

Description

The Wood River Valley (WRV) is a geologic feature located in south-central Idaho. This dataset gives the topography of the land surface in the WRV and vicinity.

Usage

```
land.surface
```

Format

An object of class `SpatialGridDataFrame-class`. Each cell on the surface grid represents an elevation in meters above the North American Vertical Datum of 1988 (NAVD 88). Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM). The spatial grid is composed of 2822 rows and 2145 columns, and has cell sizes that are constant at 20 meters by 20 meters.

Source

The National Elevation Dataset (NED) 1/3-arc-second raster (Gesch, 2007; Gesch and others, 2002). This dataset was downloaded on September 22, 2013 in a Esri ArcGRID format using the National Map Viewer. NED data are distributed in geographic coordinates in units of decimal degrees, and in conformance with the NAD 83. Elevation values are in meters above the NAVD 88. The west, east, south, and north bounding coordinates for this dataset are -115, -114, 43, and 44 decimal degrees, respectively. Post-processing includes: (1) project the values of the NED dataset into the `alluvium.bottom` spatial grid using bilinear interpolation, and (2) set values in cells where the elevation of the alluvium bottom is missing to `NA`.

References

Gesch, D.B., 2007, The National Elevation Dataset, in Maune, D., ed., Digital Elevation Model Technologies and Applications: The DEM Users Manual, 2nd Edition: Bethesda, Maryland, American Society for Photogrammetry and Remote Sensing, p. 99-118.

Gesch, D., Oimoen, M., Greenlee, S., Nelson, C., Steuck, M., and Tyler, D., 2002, The National Elevation Dataset: Photogrammetric Engineering and Remote Sensing, v. 68, no. 1, p. 5-11.

See Also

readGDAL, raster, projectRaster

Examples

```
image(land.surface)
summary(land.surface)

## Not run: # Commands used to create this dataset (see Source for data access information):
f <- "C:/n44w115/grdn44w115_13"  # path name for the downloaded grid map (note forward slash)
r <- raster(readGDAL(f, band = 1L))
r <- projectRaster(r, raster(alluvium.bottom), method = "bilinear")
r[is.na(raster(alluvium.bottom))] <- NA
obj <- as(r, "SpatialGridDataFrame")
names(obj) <- "land.surface"
identical(obj, land.surface)
## End(Not run)

## Not run: # Commands used to create hillshade image of land surface:
f <- "C:/n44w115/grdn44w115_13"
r <- raster(readGDAL(f, band = 1L))
ab <- raster(alluvium.bottom)
ext <- extent(ab)
ext <- extent(c(extendrange(c(ext@xmin, ext@xmax), f = 0.05),
             extendrange(c(ext@ymin, ext@ymax), f = 0.05)))  # extend extent by 5 percent
ab <- extend(ab, ext)
r <- projectRaster(r, ab, method = "bilinear")
r <- r * 2.0  # 2x vertical exaggeration
r <- hillShade(slope = terrain(r), aspect = terrain(r, opt = "aspect"),
               angle = 45, direction = 0)
f <- file.path(tempdir(), "hillshade.png")
r.ran <- range(values(r), na.rm = TRUE)
values(r) <- findInterval(values(r), seq(r.ran[1], r.ran[2], length.out = 255)) / 255
png::writePNG(as.matrix(r), target = f, asp = 1, text = c(projargs = proj4string(r)),
              metadata = c(xleft = ext@xmin, ybottom = ext@ymin, xright = ext@xmax,
                           ytop = ext@ymax))
browseURL(f)
## End(Not run)

plot(NA, xlim = c(2460000, 2480000), ylim = c(1360000, 1380000),
     xlab = "", ylab = "", asp = 1)
i <- png::readPNG(system.file("extdata/hillshade.png", package = "wrv"),
                  native = TRUE, info = TRUE)
pos <- attr(i, "metadata")
crs <- sp::CRS(attr(i, "info")$text)
rasterImage(i, pos[1], pos[2], pos[3], pos[4])
box()
```

`major.streams` *Major Tributary Streams*

Description

Major tributary streams of the Wood River Valley and surrounding areas.

Usage

```
major.streams
```

Format

An object of SpatialLinesDataFrame-class containing 581 Lines. Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM).

Source

Idaho Department of Water Resources (IDWR).

Examples

```
spplot(major.streams)
str(major.streams)

## Not run: # Commands used to create this dataset:
u <- "https://github.com/jfisher-usgs/wrv/raw/master/inst/extdata/major.streams.zip"
obj <- ReadGeoDataFromURL(u)
obj <- spTransform(obj, alluvium.bottom@proj4string)
all.equal(obj, major.streams)
## End(Not run)
```

---

map.labels *Map Labels*

Description

Map labels in the Wood River Valley and surrounding areas.

Usage

```
map.labels
```

Format

An object of SpatialPointsDataFrame-class containing 6 points. Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM).

Source

Best estimates of label locations.

Examples

```
plot(map.labels, col = "red")
lab <- cbind(map.labels@coords, map.labels@data)
for (i in 1:nrow(lab)) {
  text(lab$x[i], lab$y[i], labels = lab$label[i], cex = lab$cex[i],
       col = lab$col[i], font = lab$font[i], srt = lab$srt[i])
}

## Not run: # Commands used to create this dataset:
f <- system.file("extdata/map.labels.csv", package = "wrv")
obj <- read.csv(f, strip.white = TRUE, stringsAsFactors = FALSE)
obj$label <- sub("\\\n", "\\n", obj$label)
coordinates(obj) = 1:2
colnames(obj@coords) <- c("x", "y")
proj4string(obj) <- CRS("+init=epsg:4326")
obj <- spTransform(obj, alluvium.bottom@proj4string)
identical(obj, map.labels)
## End(Not run)
```

PlotMap *Plot Map*

Description

Map values of a raster layer. A key showing how the colors map to raster values is shown below the map.

Usage

```
PlotMap(r, layer = 1, att = NULL, n, breaks, xlim = NULL, ylim = NULL,
        zlim = NULL, asp = 1, extend.xy = FALSE, extend.z = FALSE,
        reg.axs = TRUE, trim.r = TRUE, dms.tick = FALSE, bg.lines = FALSE,
        bg.image = NULL, pal = rainbow, col = NULL, max.dev.dim = c(43, 56),
        bw = NULL, scale.loc = "bottomleft", arrow.loc = NULL,
        explanation = NULL, credit = proj4string(r), hill.shade = NULL,
        contour.lines = NULL, rivers = NULL, lakes = NULL, useRaster = TRUE,
        dev.type = c("cur", "new", "pdf", "png"), file = "Rplot")
```

Arguments

`r`	raster, SpatialGridDataFrame, or CRS; a raster layer with values to be plotted or a coordinate reference system (`CRS`).
`layer`	integer; column to use in the `SpatialGridDataFrame`.
`att`	numeric or character; variable identifying the levels attribute to use in the Raster Attribute Table (RAT). This argument requires `r` values that are of class `factor`.
`n`	integer; desired number of intervals to partition the range of raster values (or `zlim` if specified) (optional).
`breaks`	numeric; vector of break points used to partition the colors representing numeric raster values (optional).
`xlim`	numeric; vector of length 2 giving the minimum and maximum values for the *x*-axis.

`ylim`	numeric; vector of length 2 giving the minimum and maximum values for the *y*-axis.
`zlim`	numeric; vector of length 2 giving the minimum and maximum raster values for which colors should be plotted.
`asp`	numeric; *y/x* aspect ratio for spatial axes.
`extend.xy`	logical; if `TRUE` the spatial limits will be extended to the next tick mark on the axes beyond the grid extent.
`extend.z`	logical; if `TRUE` the raster value limits will be extended to the next tick mark on the color key beyond the measured range.
`reg.axs`	logical; if `TRUE` extends the spatial data range.
`trim.r`	logical; if `TRUE` the outer rows and columns that consist of all `NA` values will be removed.
`dms.tick`	logical; if `TRUE` the axes tickmarks are specified in degrees, minutes, and decimal seconds.
`bg.lines`	logical; if `TRUE` the graticule is drawn in back of the raster layer using white lines and a grey background.
`bg.image`	list; specifies whether an image should be drawn in back of the raster layer. The image is described using a list of arguments supplied to `rasterImage`. An additional agrument may also be passed to adjust the opacity of the image (`"alpha"`).
`pal`	function; a color palette to be used to assign colors in the plot.
`col`	character; a vector of colors to be used in the plot. This argument requires `breaks` specification for numeric values of `r` and overrides any palette function specification. For numeric values there should be one less color than breaks. Factors require a color for each level.
`max.dev.dim`	numeric; a vector of length 2 giving the maximum width and height for the graphics device in picas, respectively. Suggested dimensions for single-column, double-column, and sidetitle figures are `c(21, 56)`, `c(43, 56)`, and `c(56, 43)`, respectively.
`bw`	numeric; width of a color key box in picas. This argument requires `r` values that are of class `factor`.
`scale.loc`	character; position of the scale bar: `"bottomleft"`, `"topleft"`, `"topright"`, or `"bottomright"` to denote scale location.
`arrow.loc`	character; position of the north arrow: `"bottomleft"`, `"topleft"`, `"topright"`, or `"bottomright"` to denote arrow location.
`explanation`	character; a label explaining the raster value.
`credit`	character; a label crediting the base map.
`hill.shade`	list; specifies whether a semi-transparent hill shade layer is drawn on top of the raster layer. This layer is described using a list of arguments supplied to `hillShade`. Passed arguments include `"angle"` and `"direction"`. Additional agruments may also be passed that control the vertical aspect ratio (`"z.factor"`) and color opacity (`"alpha"`).
`contour.lines`	list; specifies whether contour lines should be drawn. The contours are described using a list of arguments supplied to `contour`. Passed arguments include `"drawlables"`, `"method"`, and `"col"`.
`rivers`	list; specifies whether lines should be drawn. The lines are described using a list of arguments supplied to the plot method for `SpatialLines`. Passed arguments include `"x"`, `"col"`, and `"lwd"`.

lakes	list; specifies whether polygons should be drawn. The polygons are described using a list of arguments supplied to the plot method for SpatialPolygons. Passed arguments include "x", "col", "border", and "lwd".
useRaster	logical; if TRUE a bitmap raster is used to plot the image instead of polygons. Bitmap images require a regular grid.
dev.type	character; graphics device type. Defaults to the 'active' device ("cur") and if it is unavailable a new device ("new") is opened based on getOption("device"). Specification of a file argument is required to open a new graphic device for a PDF formatted file ("pdf") or PNG formatted bitmap file ("png").
file	character; name of the output file. Requires a dev.type of "pdf" or "png".

Details

The dimensions of a new graphics device is dependent on the argument values of max.dev.dim and asp.

Value

Returns a list with graphical parameters "din" and "usr". See par for details.

Author(s)

J.C. Fisher

See Also

image

Examples

```
r <- raster(system.file("external/test.grd", package="raster"))
PlotMap(r, scale.loc = "topleft", dms.tick = TRUE, trim.r = TRUE)

graphics.off()
r <- raster(nrow = 10, ncol = 10)
r[] <- 1L
r[51:100] <- 2L
r[3:6, 1:5] <- 8L
r <- as.factor(r)
rat <- levels(r)[[1]]
rat$land.cover <- c("Pine", "Oak", "Meadow")
rat$code <- c(12, 25, 30)
levels(r) <- rat
PlotMap(r, att = "land.cover", col = c("grey", "orange", "purple"))
PlotMap(r, att = "code")

graphics.off()
r <- alluvium.bottom
i <- png::readPNG(system.file("extdata/hillshade.png", package = "wrv"),
                  native = TRUE, info = TRUE)
pos <- attr(i, "metadata")
img <- list(image = i, xleft = pos[1], ybottom = pos[2], xright = pos[3], ytop = pos[4])
PlotMap(r@proj4string, bg.image = img, reg.axs = FALSE)
plot(aquifer.extent, border = "red", add = TRUE)
PlotMap(r, bg.image = img, dev.type = "new")
```

```
PlotMap(r, n = 10, extend.xy = TRUE, dev.type = "new")

graphics.off()
PlotMap(r, ylim = c(NA, 1360000), max.dev.dim = c(56, 43), n = 10, extend.z = TRUE,
        contour.lines = list(col = "#A9A9A9"))
plot(aquifer.extent, add = TRUE)
hill.shade <- list(z.factor = 15, alpha = 0.4)
txt <- "Land surface elevation in meters above National Geodetic Vertical Datum of 1929."
ans <- PlotMap(r, ylim = c(NA, 1360000), max.dev.dim = c(56, 43), bg.lines = TRUE,
               hill.shade = hill.shade, arrow.loc = "topright", explanation = txt)

## Not run:
pdf("Rplot.pdf", width = ans$din[1], height = ans$din[2], version = "1.6",
    colormodel = "cmyk")
PlotMap(r, ylim = c(NA, 1360000), max.dev.dim = c(56, 43), bg.lines = TRUE,
        hill.shade = hill.shade, arrow.loc = "topright", explanation = txt,
        useRaster = FALSE)
dev.off()
PlotMap(r, ylim = c(NA, 1360000), max.dev.dim = c(56, 43), bg.lines = TRUE,
        hill.shade = hill.shade, arrow.loc = "topright", explanation = txt,
        dev.type = "png")
## End(Not run)

graphics.off()
```

`ReadGeoDataFromURL` *Read Geographic Data from a URL*

Description

Downloads a geospatially formatted file from a URL (such as `http://` or `https://`) and reads it into a spatial object.

Usage

```
ReadGeoDataFromURL(u)
```

Arguments

u	character; the URL where the file is located.

Details

This function requires the suggested package **RCurl**. Accepted file types include GeoTIFF's (‘`.tif`’) and Esri shapefile's (‘`.zip`’), the set of files that comprise the shapefile must be placed in a *zip* archive.

Value

The returned object for a GeoTIFF is of class `SpatialGridDataFrame-class`. A SpatialPoints*, SpatialLines*, or SpatialPolygons* object is returned for a points, lines, or polygons shapefile, respectively.

Author(s)

J.C. Fisher

See Also

CFILE, curlPerform, unzip, readGDAL, readOGR

Examples

```
## Not run:
u <- "https://github.com/jfisher-usgs/wrv/raw/master/inst/extdata/aquifer.extent.zip"
obj <- ReadGeoDataFromURL(u)
plot(obj)
## End(Not run)
```

ReadModflowBinaryFile *Read MODFLOW Binary File*

Description

Read binary output data from a MODFLOW simulation.

Usage

```
ReadModflowBinaryFile(f, r)
```

Arguments

f	character; name of binary file.
r	RasterLayer; a raster grid template that must match model output.

Details

This function reads binary head ('.hds') and drawdown ('.ddn') files generated from a MODFLOW-2005 or MODFLOW-USG (using a structured grid) simulation.

Value

Returns an object of class RasterStack, a collection of RasterLayer objects representing head or drawdown at each outputted time step.

Author(s)

J.C. Fisher

See Also

readBin

Examples

```
## Not run:
f <- file.path(getwd, "Run", "wrv_ss_mfusg.hds")
r <- rs.model[[1]]
rs <- ReadModflowBinaryFile(f, r)
## End(Not run)
```

---

sink.locations *Polygons Defining Groundwater Sinks*

Description

Polygons are used to define boundary conditions at sink locations in the model domain. The polygons clip the line segments along the aquifer boundary (see `aquifer.extent`), and model cells intersecting these clipped-line segments are defined as boundary cells.

Usage

```
sink.locations
```

Format

An object of `SpatialPolygonsDataFrame-class` containing a set of 2 `Polygons` and a `data.frame` of attribute data. The `data.frame` has the following variables:

Name Identifier for the polygon.

Description Currently not used.

Head The estimated head boundary condition in meters above the North American Vertical Datum of 1988 (NAVD 88).

Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM).

Source

A Keyhole Markup Language (KML) file created in Google Earth with polygons drawn by hand in areas of known sinks.

Examples

```
plot(sink.locations)
str(sink.locations)

## Not run: # Commands used to create this dataset:
f <- system.file("extdata/sink.locations.kml", package = "wrv")
obj <- readOGR(f, basename(f))
obj <- spTransform(obj, alluvium.bottom@proj4string)
f <- system.file("extdata/sink.locations.csv", package = "wrv")
h <- read.csv(f, row.names = 1L)
obj@data <- cbind(obj@data, h)
identical(obj, sink.locations)
## End(Not run)
```

source.locations — *Polygons Defining Groundwater Sources*

Description

Polygons used to define boundary conditions at source locations in the model domain. Model cells intersecting a polygon line segment are defined as source cells. Cells within the body of the polygon but not a source cell are made inactive.

Usage

```
source.locations
```

Format

An object of `SpatialPolygonsDataFrame-class` containing a set of 21 `Polygons` and a `data.frame` of attribute data. The `data.frame` has the following variables:

Name Identifier for the polygon, typically the abbreviated name of the gulch where the source is located.

Description Currently not used.

Flow The estimated volumetric flux at the boundary, in cubic meters per day.

Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM).

Source

A Keyhole Markup Language (KML) file created in Google Earth with polygons drawn by hand in areas of known sources.

Examples

```
plot(source.locations)
str(source.locations)

## Not run: # Commands used to create this dataset:
f <- system.file("extdata/source.locations.kml", package = "wrv")
obj <- readOGR(f, basename(f))
obj <- spTransform(obj, alluvium.bottom@proj4string)
f <- system.file("extdata/source.locations.csv", package = "wrv")
d <- read.csv(f, row.names = 1L)
obj@data <- cbind(obj@data, d)
identical(obj, source.locations)
## End(Not run)
```

surface.water *Traces of Rivers and Canals*

Description

Rivers and canals of the Wood River Valley.

Usage

```
surface.water
```

Format

An object of SpatialLinesDataFrame-class containing 668 Lines. Geographic coordinates are in units of meters, in conformance with the North American Datum of 1983 (NAD 83), and placed in a Idaho Transverse Mercator projection (IDTM).

Source

Idaho Department of Water Resources (IDWR).

Examples

```
spplot(surface.water)
str(surface.water)

## Not run: # Commands used to create this dataset:
u <- "https://github.com/jfisher-usgs/wrv/raw/master/inst/extdata/surface.water.zip"
obj <- ReadGeoDataFromURL(u)
obj <- spTransform(obj, alluvium.bottom@proj4string)
all.equal(obj, surface.water)
## End(Not run)
```

Index

∗Topic **IO**
ExportRasterStack, 9
ReadGeoDataFromURL, 19
ReadModflowBinaryFile, 20
∗Topic **datagen**
CreateModflowInputFiles, 6
∗Topic **datasets**
alluvium.bottom, 2
aquifer.extent, 3
aquitard.extent, 4
basalt.extent, 4
cities, 5
lakes, 12
land.surface, 13
major.streams, 14
map.labels, 15
sink.locations, 21
source.locations, 22
surface.water, 23
∗Topic **hplot**
PlotMap, 16
∗Topic **utilities**
ExcludeSmallCellChunks, 8
ExtractAlongTransect, 10
FindConnectedCells, 11

alluvium.bottom, 2, *13*
aquifer.extent, 3, *21*
aquitard.extent, 4
area.poly, *10*

basalt.extent, 4

cat, *7*
cities, 5
clump, *8*
contour, *17*
CreateModflowInputFiles, 6
CRS, *16*

data.frame, *21, 22*
dir.create, *9*

ExcludeSmallCellChunks, 8, *12*
ExportRasterStack, 9

extract, *10*
ExtractAlongTransect, 10

factor, *16*
FindConnectedCells, 11

getOption, *18*
gpc.poly, *10*

hillShade, *17*

image, *18*

lakes, 12
land.surface, 13
Lines, *15*, *23*
list, *10*

major.streams, 14
map.labels, 15

NA, *8, 9*, *13*, *17*
NULL, *7*, *9*

par, *18*
PlotMap, 16
png, *9*
Polygons, *3–5*, *12*, *21, 22*
projectRaster, *14*

raster, *14*
rasterImage, *17*
RasterLayer, *6*, *9*, *20*
RasterStack, *20*
readBin, *20*
readGDAL, *2*, *14*, *20*
ReadGeoDataFromURL, 19
ReadModflowBinaryFile, 20
readOGR, *20*

save, *9*
sink.locations, 21
source.locations, 22
SpatialLines, *17*
SpatialPointsDataFrame, *10*

SpatialPolygons, *18*
spplot, *9*
surface.water, *23*

unzip, *20*

write.table, *7*, *9*
writeRaster, *9*