00000100000
00001010000
00010001000
00100000100
01000000010
10000000001
Software notes

Ecography 39: 354–360, 2016
doi: 10.1111/ecog.01880

Subject Editor: Michael Borregaard. Editor-in-Chief: Michael Borregaard. Accepted 22 September 2015

geoknife: reproducible web-processing of large gridded datasets

Jordan S. Read, Jordan I. Walker, Alison P. Appling, David L. Blodgett, Emily K. Read and Luke A. Winslow

J. S. Read (jread@usgs.gov), J. I. Walker, A. P. Appling, D. L. Blodgett, E. K. Read and L. A. Winslow, Center for Integrated Data Analytics, U.S. Geological Survey, Middleton, WI, USA. APA also at: Center for Limnology, Univ. of Wisconsin – Madison, Madison, WI, USA.

Geoprocessing of large gridded data according to overlap with irregular landscape features is common to many large-scale ecological analyses. The *geoknife* R package was created to facilitate reproducible analyses of gridded datasets found on the U.S. Geological Survey Geo Data Portal web application or elsewhere, using a web-enabled workflow that eliminates the need to download and store large datasets that are reliably hosted on the Internet. The package provides access to several data subset and summarization algorithms that are available on remote web processing servers. Outputs from *geoknife* include spatial and temporal data subsets, spatially-averaged time series values filtered by user-specified areas of interest, and categorical coverage fractions for various land-use types.

Global and continental-scale datasets are essential resources for addressing many of ecology's most pressing questions (Soranno and Schimel 2014). However, analyzing these datasets requires more computer storage and processing capacity than most individual researchers can readily supply. In particular, many large-scale earth systems analyses require the geoprocessing of gridded data according to overlap with irregular landscape features (Hay and Clark 2003, Heffernan et al. 2014, Read et al. 2014). Such computationally intensive geoprocessing tasks are most efficiently executed on high performance remote servers (Díaz et al. 2007, Reichardt 2010, Zhao et al. 2012, Leonard and Duffy 2014), which can quickly distill large volumes of data into subsets or summaries for local analysis. While remote processing has taken much of the data processing burden off of scientists' desktops (Fig. 1), it also poses challenges to reproducibility because workflows involve data and algorithms stored in several locations, and analysis is often done through multiple interfaces.

Reproducible analyses of large-scale environmental data are certainly possible, but new tools are needed to document workflows that combine remote data processing tasks with smaller-scale local analyses. The list of open-source tools for desktop geospatial data analysis continues to grow (e.g. see <www.qgis.org>, <www.gdal.org>, tools available on <www.cran.r-project.org/web/views/Spatial.html>, <www.pysal.org>), and many resources also exist for accessing data from the Internet (see ropensci.org, Varela et al. 2014, Hirsch and De Cicco 2015). Domain-specific packages in the R open-source programming language (R Development Core Team) promote algorithm transparency and tool reuse (Liaw and Wiener 2002, Oksanen et al. 2007, Pebesma et al. 2012), and can be paired with workflow software to improve the efficiency and reproducibility of local analyses (Kepler; Ludäscher et al. 2006, ProjectTemplate; White 2014). A remaining challenge for data-intensive ecology is capturing critical early-stage processing details for geospatial datasets that are too large to easily share and store locally (Michener and Jones 2012).

Here we present the *geoknife* R package, a multi-purpose tool for web-processing of large gridded spatial datasets that are hosted remotely. *geoknife* creates locally-defined data requests, outsources geoprocessing tasks to professionally maintained web servers, and returns manageably sized datasets for further local analysis (Fig. 1b). By providing scriptable access to these remote processors from a common programming language, *geoknife* makes large-scale analyses reproducible and easier to communicate to other researchers. Moving geoprocessing tasks off the desktop also enhances reproducibility by separating publicly-available data and algorithms from study-specific analyses. Further, *geoknife* provides a means for data and algorithm discovery: while *geoknife* can be used with many remotely-hosted datasets, the package comes already linked to a curated catalog of resources available via the U.S. Geological Survey Geo Data Portal, an application that can access and remotely process hundreds of spatially explicit data sources including land use data and downscaled climate projections. *geoknife* thus joins a growing number of scientific software tools in helping to connect researchers with large, web-based computing and dataset resources.

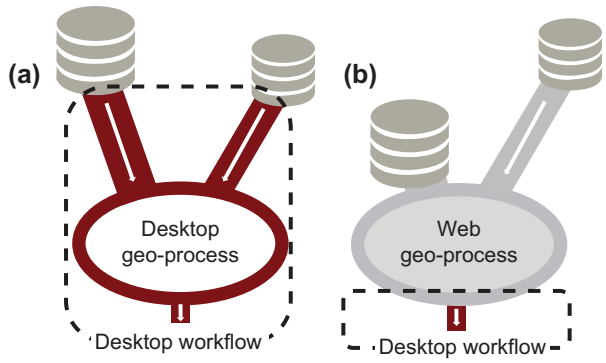

Figure 1. Remote processing within scientific workflows. (a) A common but expensive approach: download large datasets; store and analyze locally. (b) The *geoknife* approach: decrease computational load and enhance reproducibility by outsourcing the data storage and geoprocessing to remote servers. Size of arrows represents data volume.

geoknife R package description

The *geoknife* R package provides intuitive and reproducible access to spatial summaries and subsets of hundreds of regional, national, and global data sources. The package includes methods for finding datasets, defining spatial areas of interest, submitting processing jobs to a remote server, and loading processing results into R (see Example section below). *geoknife* eases reproducibility of web-processing large gridded datasets and includes rich metadata for sharing web-enabled workflows with other researchers.

geoknife outsources processing jobs to the U.S. Geological Survey's Geo Data Portal (GDP; <http://cida.usgs.gov/gdp/>; Blodgett et al. 2012), and the GDP project team is a direct partner in the development of *geoknife*. The GDP has several internet-available components used by *geoknife*: a data archive that catalogs datasets, a server that stores a collection of commonly used geospatial features, and a processing service that accepts and executes processing jobs. The *geoknife* package abstracts the complexity of gridded data, geospatial features, and geoprocessing details into the concepts of 'fabric', 'stencil', and 'knife' (respectively, Fig. 2).

Datasets (*fabrics*) available to *geoknife* include: datasets that are co-located with GDP processing resources, datasets that are hosted external to the GDP but are indexed in the GDP catalog, and other external datasets that are not indexed by the catalog. The *fabric* is a gridded, web-accessible dataset that follows one of two common protocols: Open-source Project for a Network Data Access Protocol (OPeNDAP; Cornillon et al. 2003) or Web Coverage Service (WCS; Evans 2003). Parameters for defining *fabric* include the time dimension for sampling (when applicable), the URL for its location, and the variable(s) of interest. The *geoknife* function `query('webdata')` returns urls and metadata for all datasets in the GDP catalog, while dataset resource pages (e.g. <www.esrl.noaa.gov/psd/thredds/dodsC/Datasets/catalog.html>, <http://apdrc.soest.hawaii.edu/data/data.php>) and web searches can be used to discover other datasets. *geoknife* users thus have access to hundreds of regional, continental, and global datasets.

Geospatial features that delineate specific regions of interest for processing are referred to as *stencil*s. A *stencil* can include point or polygon groupings of various forms and can be supplied as a Web Feature Service (WFS; Vretanos 2002) URL, or directly in R as a data.frame or as spatial polygon objects (sp R package; Pebesma and Bivand 2005, Bivand et al. 2008) These flexible feature types allow users to read shapefiles into R and turn them into stencils, define simpler point or polygon collections in R based on field data, or take advantage of hosted WFS endpoints for defining the spatial scope of data subsets or summarization.

knife defines the way the analysis will be performed, including the algorithm and version used, the URL that receives the processing request, the types of statistics returned, and the format of the results. Processing algorithms available to *knife* include data subsetting and area-weighted (and -unweighted) statistical summaries (additional algorithm details can be found in Blodgett et al. 2011). Once the *geoknife* package is loaded, a list of available algorithms can be

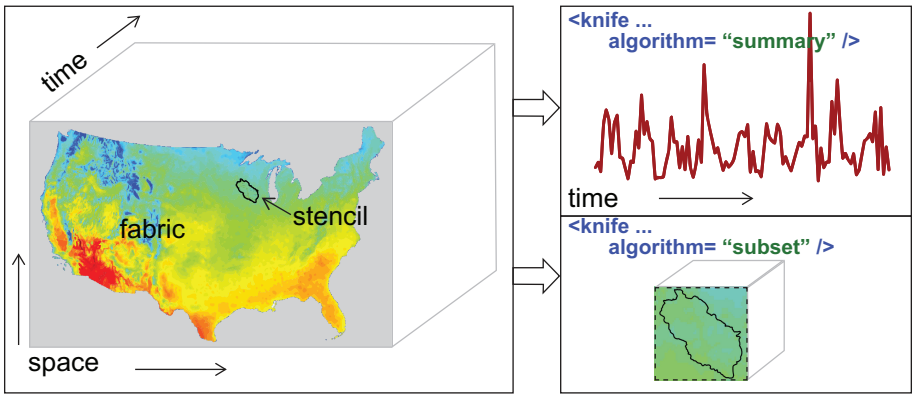

Figure 2. The three components of *geoknife* web-processes are the *stencil*, *fabric*, and *knife*. The *stencil* includes the geospatial features of interest. The *fabric* configures the web data that will be subset or summarized relative to its overlap with *stencil*. The *knife* defines the processing to be used, including algorithms, statistics returned, and email listeners. This example *stencil* includes Driftless Area Ecoregion, with the PRISM dataset as *fabric*. The algorithms defined in *geoknife*'s *knife* will result in various outputs, including text results for time series (top right) or smaller subsets of the data (bottom right). Algorithm names have been simplified for explanatory purposes (Table 1).

obtained using the R command: **`query(webprocess(), 'algorithms')`**. The output formats vary among algorithms. Subsetting algorithms return results in the native format of the dataset (e.g. NetCDF or geotiff formats), while the spatial summary algorithms return spatially-referenced time series data (such as spatially-averaged rainfall or air temperature over a specified region; Table 1; Fig. 2). Subset algorithms return a spatial subset of the data at the full extent of the *stencil*'s bounding box and a temporal subset according to the time range specified. Area-weighted summaries take into account the fraction of individual grid cells in *fabric* covered by *stencil*, while unweighted summaries average each cell that is covered by stencil, regardless of percentage overlap. *geoknife* can be used for both categorical and scalar variables: the 'categorical coverage fraction' algorithm for *knife* processes gridded data where grid cell values represent a categorical type (e.g. land-use type, land-cover type) as opposed to a scalar value (such as temperature). The algorithm returns summarized fractions of these coverage types for each feature in *stencil*.

Metadata captured in *fabric*, *stencil*, *and knife* collectively define the configuration for a geoprocessing request that can be outsourced to the remote web server (the GDP by default). The high-level **`geoknife()`** function takes these three arguments and returns a *geojob*, which includes a reference for an active remote processing job, in addition to other metadata. An ongoing *geojob* has no effect on the user's working environment, as the processing task is carried out elsewhere. The *geojob* can be checked by users or configured to send an email when complete. Processing time varies depending on data configurations, *stencil* complexity, and other factors (see Blodgett et al. 2012 for additional details). For example, a time series of monthly precipitation for 100 yr from the PRISM dataset (Daly et al. 1994) averaged for the state of Oregon is typically calculated and returned to the user within one minute, while other datasets may take much longer for a similar time period. Results of a finished processing job can be loaded into the R environment for local analyses with the **`result()`** function (see Example section for more details).

Web resources for data access and/or processing (such as those leveraged by *geoknife*) can change in time, with negative impacts on reproducibility. While *geoknife* has no system in place to access version information for datasets, *geojobs* store the algorithm version number and the *geoknife* package version number to document the exact processing specifications used (in addition to other user parameters such as specifying file output types). To date, the geoprocessing algorithms used by *geoknife* have remained unchanged since undergoing rigorous review and validation (Blodgett et al. 2011). If future changes are made, users will be able access earlier algorithm versions for processing data, although only one version of each algorithm is hosted and available at the time of this writing.

geoknife processing tasks can be reproduced by retaining the *fabric*, *stencil*, and *knife* objects, or by saving the *geojob*, which contains all necessary information for configuring future processing jobs. Saving these components to file from the R workspace can support re-running the processing routines at a later date or sharing *geoknife*-enabled workflows with others.

Software design

Reproducibility in ecology often begins after data collection or download. This is common in studies with lab or field procedures that are incompletely documented, and for analysis of massive raw datasets. However, for large open datasets, local duplication can be expensive and potentially error prone as local data may become corrupted, lost, or outdated. *geoknife* avoids local duplication while capturing more of the data lifecycle, extending the reproducibility of an analysis. The steps for external data access and processing are stored in reusable, space-efficient configurations (e.g. *geojobs*) that link the analysis to both the original dataset and the algorithms used for subsetting or summarization. The outsourcing of data processing and storage to reliable remote hosts is central to the *geoknife* philosophy (Fig. 1a vs 1b).

geojobs are processed by the U.S. Geological Survey's Geo Data Portal web resources, and the GDP project team has participated actively in the design of *geoknife*. The algorithms available to parameterize a *knife* are thoroughly documented, peer reviewed, and have been tested against other GIS processors (Blodgett et al. 2011). For *geoknife* to execute a *geojob*, it assembles a request, or set of conditions, that is sent to the GDP for execution. The structure and content of the request is defined by web data standards (see below) and GDP algorithm requirements respectively. The details of setting up a new GDP instance are beyond the scope of this paper. However, advanced users can deploy their own GDP, as it is open-source software (under a CC0 license) that is available to download, modify, and setup in processing environments (<https://github.com/USGS-CIDA/geo-data-portal>). *geoknife* can easily direct *geojobs* to other GDP instances through modifying the web processing url of *knife* with the **`url()`** function.

Much of the functionality available from *geoknife* is implemented such that it acts as a web service client to the

Table 1. Three example analyses with corresponding *fabrics*, *stencils*, and *knives*. Demonstrations of these examples can be run in R using the **`demo()`** function, e.g. **`demo('prism_subset', package='geoknife')`**.

fabric	Stencil	knife	result	geoknife demo()[a]
PRISM[b]	Collection of longitude/latitude pairs	OPeNDAP subset	NetCDF file subset to the full bounding box of the stencil	'prism_subset'
PRISM	U.S. states of Oregon, Colorado, and Connecticut	Area Grid Statistics (weighted)	Time series values for each state polygon	'prism_summary'
ICLUS[c]	Level III Ecoregions	Categorical Coverage Fraction	Fractional cover for each polygon	'iclus_categorical'

[a]In geoknife package ver. 1.0.0; [b](Daly et al. 1994); [c](Bierwagen et al. 2010).

GDP. The World Wide Web Consortium (W3C) defines a web service as '...a software system designed to support interoperable machine-to-machine interaction over a network'. (W3C Working Group and others 2004). Web services allow a client to tell a server to execute an action given a set of conditions. For example, when *geoknife* sends a *geojob* to the server for execution, the set of conditions are the *fabric*, *stencil*, and *knife* configuration. In response to this processing request from a *geoknife* user, the server responds with a reference identifier for the newly created processing job. The *geoknife* client can then periodically check this identifier for the status of the ongoing process to see if the job is complete or if some error has occurred. All of *geoknife*'s interactions with remote servers are mediated using web services that conform to published standards for geoprocessing and data access.

geoknife's ability to access a variety of datasets and run user-configured remote processes is made possible through the use of standard web service interfaces and data formats. The Open Geospatial Consortium Web Processing Service (OGC-WPS) specification (Foerster and Stoter 2006) defines how *geoknife* communicates processing configuration details with the server. *fabric* attributes, spatiotemporal coordinates, and variable data are accessed using an OPeNDAP interface. Simple *stencils* can be specified using standard R spatial objects that are sent to the processing server as OGC Geography Markup Language (Cox et al. 2002). Commonly used *stencils* (ecoregions, states, watersheds, etc.), or those that are stored on a public server, can be specified 'by reference' in the *geojob* request that is passed to the processing server. The reference in *stencil* describes a request to an OGC-WFS that will, when dereferenced, provide spatial details for use in a remote process. The state of Oregon in Table 1 is one such example of a publically hosted WFS *stencil*. It is important to note that while these interfaces and data formats form the basis for data access and process specifications for *geoknife*, a user of the package does not need to be aware of their complexities. A foundation of widely adopted standards makes the *geoknife* package easier to extend to the use of other datasets and processing resources in future versions.

A significant advantage of implementing *geoknife* as a client to the GDP is the ability to take advantage of large volumes of data collocated with processing resources. High-throughput data processing can be much faster when processing takes place on the same physical machine or within close network proximity to the data. Many of the remote processing capabilities leveraged by *geoknife* adopt this strategy to avoid the high cost and slow speeds of large-scale data transfer. Interoperability with datasets and *stencils* outside of the GDP's network is a key feature of *geoknife* (see *geoknife* R package description), but longer processing times result from such requests. The collection of generalized processing services and datasets available to users of *geoknife* can be used to extract the data necessary for analyses without the need to store or process high volumes of data with limited computational resources (Fig. 1).

Example usage of geoknife

A common use of *geoknife* is to summarize various gridded time series data at the locations of study sites or larger regions. While applications of the tool differ depending on research questions and variables of interest, a simplified example use case is explained below complete with R code for performing these operations with *geoknife*.

Finding data (fabric)

The **query()** function in *geoknife* can be used to list online datasets that can be processed by the GDP's geo-web processing service. No additional search parameters are used in the following example, so all catalogued datasets are returned:

```
library(geoknife)
webdatasets=query('webdata')
length(webdatasets)
 ## [1] 184
```

Interrogation of datasets can be done by printing the returned dataset list, which displays the title and the url of each dataset by default (this example truncates the 184 datasets to display 5):

```
webdatasets[61:65]
```

```
## An object of class"datagroup":
## [1] Daymet Daily surface weather on a 1km grid for North America, 1980-2012
##    url:    http://thredds.daac.ornl.gov/thredds/dodsC/daymet-agg/daymet-agg.ncml
## [2] Eighth degree-CONUS Daily Downscaled Climate Projections Minimum and Maximum Temperature
##    url:    http://cida.usgs.gov/thredds/dodsC/dcp/conus_t
## [3] Eighth degree-CONUS Daily Downscaled Climate Projections Precipitation
##    url:    http://cida.usgs.gov/thredds/dodsC/dcp/conus_pr
##    [4]    Future    California    Basin Characterization Model Downscaled Climate and Hydrology
##    url:    http://cida.usgs.gov/thredds/dodsC/CA-BCM-2014/future
## [5] GLDAS Version 2.0 Noah 0.25 degree monthly data
##  url:  http://hydro1.sci.gsfc.nasa.gov/dods/GLDAS_NOAH025_M.020
```

Finding additional information about a particular dataset is supported by **title()** and **abstract()**, which return the dataset titles and abstracts respectively:

```
title(webdatasets[87])
```

```
## [1] "Monthly Conterminous U.S. actual evapotranspiration data"
```

```
abstract (webdatasets[87])
```

```
 ## [1] "Actual ET (ETa) is produced using the operational Simplified Surface Energy Balance (SSEBop) model (Senay and others, 2013) for the period 2000
```

```
to present. The SSEBop setup is based
on the Simplified Surface Energy Balance
(SSEB) approach (Senay and others, 2007,
2011) with unique parameterization for
operational applications. It combines
ET fractions generated from remotely
sensed MODIS thermal imagery, acquired
every 8 days, with reference ET using
a thermal index approach. The unique
feature of the SSEBop parameterization
is that it uses pre-defined, seasonally
dynamic, boundary conditions that are
unique to each pixel for the hot/dry
and cold/wet reference points."
```

For this example, the parameter-elevation regressions on independent slopes (PRISM) climate data (Daly et al. 1994) dataset and the Monthly U.S. Evapotranspiration Dataset (Senay et al. 2013) dataset were used. After selecting datasets, **query()** can find the time range and list the different variables in the dataset (note that indexing datasets based on order or title are equivalent):

```
prism <- webdata(webdatasets[99])
evapotran <- webdata(webdatasets['Monthly
Conterminous U.S. actual evapotranspira-
tion data'])
query(prism, 'variables')
 ## [1] "ppt" "tmx" "tmn"

variables(prism) <- 'ppt'
query(prism, 'times')

 ## [1] "1895-01-01 UTC" "2014-12-01 UTC"

query(evapotran, 'variables')
 ## [1] "et"

variables(evapotran) <- 'et'
query(evapotran, 'times')
 ## [1] "2000-01-01 UTC" "2014-12-01 UTC"
```

The processing time range can be modified (or accessed) with **times()**. Because the evapotranspiration dataset is shorter than the monthly prism data, the processing period for prism can be modified to use the same time range as the evapotranspiration dataset:

```
times(prism) <- query(evapotran, 'times')
prism

 ## An object of class "webdata":
 ## times: 2000-01-01 2014-12-01
 ## url: http://cida.usgs.gov/thredds/dodsC/
 prism_v2
 ## variables: ppt
```

Defining spatial features (stencil)

The spatial features (*stencil*) of a *geoknife* processing job can include point collections or polygons, which can be defined from existing web resources or from local data. Web available geometries are specified using the WFS standard (see Software design section), and *geoknife* has many predefined WFS stencils built into the package. For this example, two ecoregions are used. Ecoregions are ecologically coherent spatial units often used for grouping in large environmental analyses (Omernik 1987):

```
eregions = webgeom('ecoregion::Colorado
Plateaus,Driftless Area')
eregions
 ## An object of class "webgeom":
 ## url: http://cida.usgs.gov/gdp/geoserver/
 wfs
 ## geom: derivative:Level_III_Ecoregions
 ## attribute: LEVEL3_NAM
 ## values: Colorado Plateaus, Driftless
 Area
 ## wfs version: 1.1.0
```

Additionally, points can be specified as longitude and latitude values within an R data.frame:

```
plots = simplegeom(data.frame
      ('treatment1' = c(-89, 46),
      'treatment2' = c(-88.32, 45.3),
      'control' = c(-88.6, 45.2)))
```

Submitting processing jobs (geojob)

After defining spatial areas of interest and web datasets, a processing job can be submitted to the Geo Data Portal's web processing service. Parameters that define details of the processing request are defined in the *geojob*'s knife argument, which will not be covered here in detail (use R command ?**webprocess** for more details). To submit various jobs for processing, **geoknife()** is used:

```
prism_job <- geoknife(stencil = eregions,
fabric = prism, wait = TRUE)
evapotran_job <- geoknife(stencil = eregions,
fabric = evapotran, wait = TRUE)
```

Viewing and interpreting results

The results of processing requests can be downloaded (see ?**download()** for a *geojob*) or loaded into R as a `data.frame` with **result()**. For this example, the optional argument `with.units` is used to include units of as an additional column in the resulting data:

```
prism_data <- result(prism_job, with.units = TRUE)
head(prism_data)

## DateTime Colorado Plateaus Driftless Area variable statistic units
## 1 2000-01-01    26.151798    29.06468    ppt    MEAN mm/month
## 2 2000-02-01    32.192220    25.17960    ppt    MEAN mm/month
## 3 2000-03-01    37.505486    24.92028    ppt    MEAN mm/month
## 4 2000-04-01    10.308524    55.29068    ppt    MEAN mm/month
## 5 2000-05-01    13.845586    136.81929   ppt    MEAN mm/month
```

```
evapotran_data <- result(evapotran_job, with.units=TRUE)
head(evapotran_data)
```

```
## DateTime Colorado Plateaus Driftless Area variable statistic units
## 1 2000-01-01    5.299767      0.1899183    et    MEAN    mm
## 2 2000-02-01    8.460567      0.3864346    et    MEAN    mm
## 3 2000-03-01    14.366092     4.7431640    et    MEAN    mm
## 4 2000-04-01    27.275608    19.3804420    et    MEAN    mm
## 5 2000-05-01    43.012160    57.2082800    et    MEAN    mm
```

Further data analysis and plotting of these results can be performed within the R environment.

This example highlights the ease of accessing and subsetting or summarizing high value datasets such as PRISM (Daly et al. 1994). PRISM uses large collections of quality assured station-based meteorological measurements to create spatially and temporally continuous estimates of precipitation, temperature, and other climatic variables that begins in 1895. PRISM's large spatial coverage (the contiguous US) and high spatial resolution (approximately 4 km for monthly outputs) may prove burdensome for download and local storage. Figure 2 details part of this example (note that *stencil* includes only the 'Driftless Area' ecoregion) and two potential processing results that differ according to *knife* parameters: a `data.frame` time series of spatially-weighted precipitation for the *stencil* (shown plotted in the upper right) or a spatial and temporal subset of the original dataset (lower right). Subsetting these datasets into smaller, more manageable files or summarizing results into an R `data.frame` (as in the example above) are tasks supported by *geoknife* (Table 1) for hundreds of gridded datasets.

Software requirements

As noted above, *geoknife* is part of an ecosystem of R packages designed for data access and analysis. Package dependencies for *geoknife* at the time of this writing include *sp* (ver. >= 1.0; Pebesma and Bivand 2005), *XML* (ver. >= 3.98-1.3; Lang and the CRAN Team 2015), *httr* (ver. >= 1.0.0; Wickam 2015), and *methods* (R Development Core Team). Spatial classes from the *sp* package are used in *geoknife*'s *simplegeom* class (one of two classes that can define *stencils*). The *XML* package is used to build and parse XML (EXtensible Markup Language), which is the primary information exchange format for *geoknife* and the Geo Data Portal. Web communication between *geoknife* and the Geo Data Portal is facilitated by the *httr* package. Definitions and methods of *geoknife* classes are supported by the *methods* package. R ver. >= 3.2.0 is required for ver. 1.0.0 of *geoknife*.

Installation

The *geoknife* package is free and open source, and under a CC0 license. Versioned source code is available on <https://github.com/USGS-R/geoknife>. The package can be installed directly into the R environment from the comprehensive R archive network (CRAN) using the R command: `install.packages('geoknife')`.

To cite geoknife or acknowledge its use, cite this Software note as follows, substituting the version of the application that you used for 'version 1.0.0':

Read, J. S., Walker, J. I., Appling, A. P., Blodgett, D. L., Read, E. K. and Winslow, L. A. 2015. geoknife: reproducible web-processing of large gridded datasets. – Ecography 39: 354–360 (ver. 1.0.0).

Acknowledgments – The U.S. Geological Survey National Climate Change and Wildlife Science Center provided funding for the Geo Data Portal and funding from the Dept of the Interior Northeast Climate Science Center and the Center for Integrated Analytics supported geoknife development efforts. We thank co-editors Michael Borregaard and Ted Hart who contributed substantially to the improvement of our original manuscript, in addition to helpful input received from two anonymous reviewers and Jessica Thompson. We also thank the Ecography editorial staff.

References

Bierwagen, B. G. et al. 2010. National housing and impervious surface scenarios for integrated climate impact assessments. – Proc. Natl Acad. Sci. USA 107: 20887–20892.

Bivand, R. S. et al. 2008. Applied spatial data analysis with R. – Springer.

Blodgett, D. L. et al. 2011. Description and testing of the Geo Data Portal: a data integration framework and web processing services for environmental science collaboration. – US Geological Survey, Open-File Report 2011-1157.

Blodgett, D. et al. 2012. Description of the U.S. Geological Survey Geo Data Portal Data Integration Framework. – IEEE J. Selected Topics Appl. Earth Obs. Remote Sens. 5: 1687–1691.

Cornillon, P. et al. 2003. OPeNDAP: accessing data in a distributed, heterogeneous environment. – Data Sci. J. 2: 164–174.

Cox, S. et al. 2002. OpenGIS® Geography Markup Language (GML) implementation specification, version 1.0.0. – Open Geospatial Consortium.

Daly, C. et al. 1994. A statistical-topographic model for mapping climatological precipitation over mountainous terrain. – J. Appl. Meteorol. 33: 140–158.

Díaz, L. et al. 2007. Migrating geoprocessing routines to web services for water resource management applications. – AGILE 2007 Proceedings, pp. 1–9.

Evans, J. D. 2003. Web Coverage Service (WCS), version 1.0.0. – Open Geospatial Consortium.

Foerster, T. and Stoter, J. 2006. Establishing an OGC Web Processing Service for generalization processes. – Workshop of the ICA Commission on Map Generalization and Multiple Representation.

Hay, L. E. and Clark, M. P. 2003. Use of statistically and dynamically downscaled atmospheric model output for hydrologic simulations in three mountainous basins in the western United States. – J. Hydrol. 282: 56–75.

Heffernan, J. B. et al. 2014. Macrosystems ecology: understanding ecological patterns and processes at continental scales. – Front. Ecol. Environ. 12: 5–14.

Hirsch, R. M. and De Cicso, L. A. 2015. User guide to exploration and graphics for RivEr Trends (EGRET) and dataRetrieval: R packages for hydrologic data (version 2.0, February 2015). – US Geological Survey, Techniques and Methods book 4, chapter A10, <http://dx.doi.org/10.3133/tm4A10>.

Lang, D. T. and the CRAN Team 2015. XML: tools for parsing and generating XML within R and S-Plus. – R package ver. 3.98-1.3, <http://CRAN.R-project.org/package=XML>.

Leonard, L. and Duffy, C. J. 2014. Automating data-model workflows at a level 12 HUC scale: watershed modeling in a distributed computing environment. – Environ. Model. Softw. 61: 174–190.

Liaw, A. and Wiener, M. 2002. Classification and regression by randomForest. – R News 2: 18–22.

Ludäscher, B. et al. 2006. Scientific workflow management and the Kepler system. – Concurrency and Computation: Practice and Experience 18: 1039–1065.

Michener, W. K. and Jones, M. B. 2012. Ecoinformatics: supporting ecology as a data-intensive science. – Trends Ecol. Evol. 27: 85–93.

Oksanen, J. et al. 2007. The vegan package. Community ecology package. – R package ver. 2.2-1, <http://CRAN.R-project.org/package=vegan>.

Omernik, J. M. 1987. Ecoregions of the conterminous United States. – Ann. Assoc. Am. Geogr. 77: 118–125.

Pebesma, E. J. and Bivand, R. S. 2005. Classes and methods for spatial data in R. – R News 5: 9–13.

Pebesma, E. et al. 2012. The R software environment in reproducible geoscientific research. – Eos Trans. Am. Geophys. Union 93: 163–163.

Read, J. S. et al. 2014. Simulating 2368 temperate lakes reveals weak coherence in stratification phenology. – Ecol. Model. 291: 142–150.

Reichardt, M. 2010. Open standards-based geoprocessing Web services support the study and management of hazard and risk. – Geomatics Natural Hazards Risk 1: 171–184.

Senay, G. B. et al. 2013. Operational evapotranspiration mapping using remote sensing and weather datasets: a new parameterization for the SSEB approach. – JAWRA J. Am. Water Resour. Assoc. 49: 577–591.

Soranno, P. A. and Schimel, D. S. 2014. Macrosystems ecology: big data, big ecology. – Front. Ecol. Environ. 12: 3.

Varela, S. et al. 2014. paleobioDB: an R package for downloading, visualizing and processing data from the Paleobiology Database. – Ecography 38: 419–425.

Vretanos, P. A. 2002. OGC™ Web Feature Service Implementation Specification (version 1.0. 0). – Open Geospatial Consortium.

W3C Working Group and others 2004. Web services glossary. – W3C Working Group Note, <http://www.w3.org/TR/ws-gloss>.

White, J. M. 2014. ProjectTemplate: automates the creation of new statistical analysis projects. – R package ver. 0.6, <http://CRAN.R-project.org/package=ProjectTemplate>.

Wickam, H. 2015. httr: tools for working with URLs and HTTP. – R package ver. 1.0.0, <http://CRAN.R-project.org/package=httr>.

Zhao, P. et al. 2012. The Geoprocessing Web. – Comput. Geosci. 47: 3–12.