The geoknife package

Jordan Read[1]

[1] *United States Geological Survey*

August 5, 2014

Contents

1 Introduction to geoknife

The geoknife package was created support web-based geoprocessing of large gridded datasets according to their overlap with landscape (or aquatic/ocean) features that are often irregularly shaped. geoknife creates data access and subsequent geoprocessing requests for the USGS's Geo Data Portal to carry out on a web server. The results of these requests are available for download after the process(es) have been completed. This type of workflow has three main advantages: 1) it allows the user to avoid downloading large datasets, and 2) it avoids reinventing the wheel for the creation and optimization of complex geoprocessing algorithms, 3) computing resources are dedicated elsewhere, so geoknife operations do not have much of an impact on a local computer.

geoknife interacts with a remote server to figure out what types of processing capabilities are available, in addition to seeing what types of geospatial features are already available to be used as an area of interest (commonly user-uploaded shapefiles). Because communication with web resources are central to geoknife operations, users must have an active internet connection.

The main elements of setting up and carrying out a geoknife ’job’ are to define the processing algorithm that will be used, choosing an area of interest, filling out the details for the job details (including the dataset to be used; these details are called ”process inputs”), and sending off the job request.

2 General Workflow

```r
library(geoknife)
```

```r
# create geoknife object w/ defaults
geoknife <- geoknife()
# give this geoknife object a linear ring as the
# feature of interest (will be adding multiple rings in the future)
linearRing = bufferPoint(c(-111.48,36.95))
setFeature(gk) <-list(LinearRing=linearRing)

# get a list of available processing algorithms
getAlgorithms(geoknife)

# set processing algorithm to feature weighted grid statistics
# feature weighted:
setAlgorithm(gk) <- getAlgorithms(gk)[4] # feature weighted

# set the post inputs for the processing dataset
setProcessInputs(gk) <- list('DATASET_ID'='ppt',
                             'DATASET_URI'='http://cida.usgs.gov/thredds/dodsC/
                             'TIME_START'='1895-01-01T00:00:00Z',
                             'TIME_END'='1900-01-01T00:00:00Z',
                             'DELIMITER'='TAB')

# printing geoknife object displays the important properties in a readable way
geoknife
# is the same as
print(geoknife)

# kick off your request
geoknife <- startProcess(geoknife)

status.geoknife <- checkProcess(geoknife)

cat('checking status of GDP request.')
cat(' Large complex requests take longer to process.\n')
```

```
repeat{
  if (!is.null(status.geoknife$URL) | status.geoknife$status!=""){
    break
  }
  cat('checking process...\n')
  Sys.sleep(10)
  if (is.null(status.geoknife$URL)){
    status.geoknife <- checkProcess(geoknife)
  }
}

if (status.geoknife$status=='Process successful'){
  cat(paste(status.geoknife$status,
            '\nDownload available at: ',status.geoknife$URL,sep=''))
} else {
  cat(status.geoknife$status)
}
```

2.1 Introduction