lakemorpho: Calculating lake morphometry metrics in R

Jeffrey W. Hollister [1] Joseph Stachelek [2]

[1] *US Environmental Protection Agency, Office of Research and Development, National Health and Environmental Effects Research Laboratory, Atlantic Ecology Division, 27 Tarzwell Drive Narragansett, RI, 02882, USA*

[2] *Michigan State University, Department of Fisheries and Wildlife, Natural Resources Building, 480 Wilson Road, Room 13, East Lansing, MI, 48824, USA*

Metrics describing the shape and size of lakes, known as lake morphometry metrics, are important for any limnological study. In cases where a lake has long been the subject of study these data are often already collected and the data is openly available. Many other lakes have this data collected, but access to the data is challenging as it is often stored on individual computers (or worse in filing cabinets) and is available only to the primary investigators. The vast majority of lakes fall into a third category in which the data is not available. This makes broad scale modelling of lake ecology a challenge as some of the key information about in-lake processes are unavailable. While this valuable *in situ* information may be difficult to obtain, several national datasets exist that may be used to model and estimate lake morphometry. In particular digital elevation models and hydrography have been shown to be predicitive of several lake morphometry metrics. The R package `lakemorpho` has been developed to utilize this data and estimate the following morphometry metrics: surface area, shoreline length, shoreline development, maximum depth, mean depth, volume, maximum lake length, mean lake width, maximum lake width, and fetch. In this software note I describe the motivation behind developing `lakemorpho`, discuss the implementation in R, and describe the use of `lakemorpho` with an example of a typical use case.

Introduction

The study and quantificaiton of lake shape (i.e. lake morphology and morphometry) is one of the foundations of limnology and for students of limnology, some of the first lessons are centered around a typical suite of metrics and how to calculate them [1]. It is also widely accepted that the morphometry of lakes and ponds can impact available nutrients and thus overall productivity. For instance, the widely used Vollenweider input-output models that are used to estimate nutrient concentrations rely on hydraulic residence time and sometimes mean depth, both of which are derived from total lake volume [2,3]. Also, clear water versus turbid water states in lakes have been linked in part to lake morphometry, in particular mean depth [4]. In short, limnologists have long recognized the importance of lake morphology as one factor controlling a variety of ecological processes in lakes.

Traditional methods for calculating lake morphometry metrics have relied upon the use of paper bathymetry maps, planimeters, or unnecessary assumptions [5–8]. In addition, detailed bathymetry is a requirement for most of these methods, but it is not universally available and often only available for a relatively small number of lakes. This is not a problem when the focus of a study is a single lake, a small number of lakes or on well studied lakes. Relying on complete bathymetry becomes a limitation when attempting to conduct regional or national studies of lakes as bathymetry is at best difficult to find or does not exist for all lakes of interest. In these cases alternative approaches for estimating lake morphometry are required.

Recent work has demonstrated the ability to estimate many of these metrics from ubiquitous spatial data. For instance, maximum depth and lake volume may be predicted using the lake polygon and surrounding topography [8,9] provided by the National Hydrography Dataset Plus and the National Elevation Dataset, respectively [10,11]. The initial development of these tools were developed with proprietary tools thus limiting their use. In an effort to reach a broader audience the tools were converted to R, expanded to include a more complete suite of lake morphometry metrics and compiled into an R Package.

Implementation and Use in R

Using R as a Geographic Information System is now possible as several packages provide spatial data handling, geospatial analysis, and visualization. It is becuase of these packages that `lakemorpho` was implemented as an R package. In particular, `lakemorpho` relies on the following packages: `maptools`, `rgdal`, `raster`, `rgeos`, `sp`, `geosphere`[12–18]. In addition to these packages two external libraries, the Geospatial Data Abstraction Library (GDAL) and Geometry Engine, Open Source(GEOS), are needed. Their availability to R and installation varies by operating system [19,20].

Using `lakemorpho`

Included in `lakemorpho` are, one function to create a `lakeMorpho` object, eleven functions to caclulate morphometry metrics, a default plotting function, two example datasets, and an example `lakeMorpho` object.

A typical workflow for using `lakemorpho` to calculate lake metrics would include pulling spatial data into R (e.g. as shapefiles, tiff, etc.), creating a `lakeMorpho` object and calculating the desired lake morphometry metrics. The following sections provide details on the type of input data required and dicsuss the use of the functions, including examples with the provided example data.

The `lakeMorpho` Class and `lakeSurroundTopo`

Many of the lake morphometry metrics rely on the same information about the lake. For instance, the functions to estimate maximum depth, mean depth, and volume rely on statistical summaries of the surrouding topography as well as the maximum in-lake distance to shoreline. [8,9]. To avoid recalculating these values, a `lakeMorpho` class was created to store the information on surrounding topography as well as the original datasets. This object is required input for all of the lake morphometry functions in the the `lakemorpho` package. In addition to this, an object of class `lakeMorpho` also holds the initial datasets and, optionally, can store the spatial objects that result from some of the lake morphometry functions. At a minimum, a `lakeMorpho` object contains:

- "lake" - A `SpatialPolygons` or `SpatialPolygonsDataFrame` object of the original input lake data.
- "elev" - A `RasterLayer` representing the elevation in a suitably large area around the lake.
- "surround" - A `SpatialPolygons` or `SpatialPolygonsDataFrame` object representing the land area defined as the surrounding topography.
- "lakeDistance" - A `RasterLayer` object of the euclidean distance from the shoreline to center of each pixel. Maximum value is equal to the maximum in-lake distance.
- "lakeOnEdge" - A logical value indicating if the `lakeMorpho` value "surround" is on the edge of the value "elev".

The `lakeSurroundTopo` function is the primary mechanism for creating a `lakeMorpho` object. There are two required inputs and one optional input for `lakeSurroundTopo`. The first required input is a `SpatialPolygons` or `SpatialPolygonsDataFrame` of the lake [16]. Only a single lake is accepted as input, although this lake may be composed of multiple polygons (i.e. a lake with islands). If metrics for multiple lakes are required they will need to be passed to the suite of `lakemorpho` functions separately. The second required input is a `RasterLayer` of the elevation surrounding the lake [17]. The default raster size is taken from the resolution of the input elevation data but may be specified separately. The third input specifies the area representing the surrounding topography. By default this is a buffer of the lake shoreline with the buffer width equal to the maximum in-lake distance. An optional `SpatialPolygons` object of any polygon intersecting the lake (e.g. catchements) can be used to define the surrounding topography instead of the default buffer. An object of class `lakeMorpho` is returned from `lakeSurroundTopo`

In addition to providing accepted inputs, users should pay attention to both the extent of the input elevation dataset as well as the coordinate reference systems used. First, the elevation data must be of a large enough extent so that the surrounding topography does not inlcude land area outside that extent (i.e would return NA values). As noted above, the `lakeOnEdge` item indicates if the surrounding topography is on the edge of the input elevation and thus returns NA values. Second, all of the functions of `lakemorpho` assume that projections have been handled prior to creating the `lakemorpho` class or calculating the metrics. If the input data are not of the same projection, `lakeSurroundTopo` will return an error. The data must be reprojected into the same coordinate reference system (CRS). Care must be taken in choosing a CRS as area and length measurments will vary between different CRS.

Usage of `lakeSurroundTopo` and generating a `lakeMorpho` object from the example data included with `lakemorpho` is done as follows:

```
#Load data
data(lakes)


#Create lakeMorpho object, example_lakeMorpho, with required inputs
example_lakeMorpho <- lakeSurroundTopo(exampleLake, exampleElev)
```

The resulting object contains the minimum set of values that are all of the expected class.

```r
lapply(example_lakeMorpho,class)
```

```
## $lake
## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"
##
## $elev
## [1] "RasterLayer"
## attr(,"package")
## [1] "raster"
##
## $surround
## [1] "SpatialPolygons"
## attr(,"package")
## [1] "sp"
##
## $lakeDistance
## [1] "RasterLayer"
## attr(,"package")
## [1] "raster"
##
## $lakeOnEdge
## [1] "logical"
```

Lake Morphometry Functions

Each of the remaining functions all expect a `lakeMorpho` object as input and all return a numeric value. Some of the functions do have a side effect of adding a spatial object to the input `lakeMorpho` object.

calcLakeMetrics Calculate all Lake Morphometry Metrics

lakeFetch Fetch is the maximum open water distance in a given direction and can be used an indicator of mixing as greater fetch implies greater potential for waves[NEED REF]. The `lakeFetch()` function calculates fetch along an input bearing. The input bearing may be any value from 0 to 360 where 0 and 360 both represent north, although the fetch for opposite directions (e.g. east and west) are identical.

To calulcate the fetch of an input lake use:

```
#Fetch for North
lakeFetch(example_lakeMorpho, 0)
```

```
## [1] 6336.798
```

```
lakeFetch(example_lakeMorpho, 360)
```

```
## [1] 6336.798
```

```
#Fetch for West
lakeFetch(example_lakeMorpho, 270)
```

```
## [1] 3129.997
```

lakeMajorAxisLength The major axis of a lake is defined as the longest line intersecting the convex hull formed around its polygon while passing through its center.

```
lakeMajorAxisLength(example_lakeMorpho, addLine = TRUE)
```

```
## [1] 13159.64
```

lakeMaxDepth Maximum lake depth provides information that may be used to, along with flow rates, estimate the residence time of a lake. While there is no substitute for field verifed measurements, maximum lake depth may be estimated with the surrounding topography. The `lakeMaxDepth()` function uses the methods outlined in Hollister *et al* [9] to provide an estimate of the maximum lake depth. It requires only a `lakeMorpho` object as input. Optionally a correction factor based off of verified depth data may be specified is one is known.

The usage for `lakeMaxDepth()` is:

```r
#Maximum Lake Depth
lakeMaxDepth(example_lakeMorpho)
```

```
## [1] 99.17621
```

lakeMaxLength Maximum lake length is the longest open water distance within a lake and, similar to fetch, is a metric that can be used to estimate mixing potential [21]. The current implementation of this in `lakemorpho` places points at equal distances apart along the shoreline of the lake and then finds the longest point-to-point distance that also does not intersect land. This value is returned as the maximum lake length. An optional parameter, with a default value of `TRUE` allows the `SpatialLines` object to be stored on the input `lakeMorpho` object.

To caluclate maximum lake length requires a `lakeMorpho` object and total number of points to use to find the maximum point-to-point distance.

```r
#Max Length with a Point Density of 250
lakeMaxLength(example_lakeMorpho, 250, addLine = FALSE)
```

```
## [1] 9470.766
```

The `pointDens` parameter can have an impact on both the processing time and the resulting value and both of these can vary as a function of the complexity of the shape of the lake with less complex lakes providing more consistent lake length across a range of number of points (Figure **??**). Given this caveat, care must be taken in choosing an appropriate number of points (and thus lines) to use to calculate maximum lake length. Several densities should be tested and the smallest number of points that produce a stable estimate should be used.

lakeMaxWidth Maximum lake width is the maximum shore to shore distance that is perpendicular to the line representing maximum lake length and is another metric related to mixing [21]. The `lakeMaxWidth` function requires a `lakeMorpho` object and `pointDens` value which is used to determine the number of points along the maximum lake length line. The issue with `pointDens` that was discussed above also exists for the use of `pointDens` with `lakeMaxWidth()` and care should be taken to determine an appropriate number of lines to test.

Usage of lakeMaxWidth is:

```r
#Max width with a point density of 250
lakeMaxWidth(example_lakeMorpho, 250)
```

```
## [1] 3194.434
```

lakeMeanDepth Mean depth of a lake is calculated as the volume of the lake divided by the area [21]. This function requires only a `lakeMorpho` object and returns a numeric value of the mean depth. Usage of the function is:

```r
lakeMeanDepth(example_lakeMorpho)
```

```
## [1] 28.94864
```

lakeMeanWidth The mean width of a lake is defined as lake area divided by maximum lake length [21]. Input for this function is a `lakeMorpho` object that has the maximum lake length line added. This requirement is checked and returns an error if the maximim length line is missing.

```r
# Throws an error if maximum lake length is missing
lakeMeanWidth(example_lakeMorpho)
```

```
## [1] 1797.037
```

```r
# Add Maximum Lake Length
lakeMaxLength(example_lakeMorpho, 100, addLine = TRUE)
```

```
## [1] 9025.588
```

```r
lakeMeanWidth(example_lakeMorpho)
```

```
## [1] 1822.948
```

lakeMinorAxisLength The minor axis of a lake is defined as the shortest line intersecting the convex hull formed around the lake polygon while passing through its center.

```r
lakeMinorAxisLength(example_lakeMorpho, addLine = TRUE)
```

```
## [1] 6926.263
```

lakeMinorMajorRatio The ratio of the lake major axis length to the minor axis length is also known as the aspect ratio. Circular lakes have aspect ratios approaching 1 while thin-elongated lakes have aspect ratios approaching 0. If major and minor axis length have not already been added to the `lakeMoropho` object these are calculated. The `addLine` argument adds the lines for the lake's minor and major axes to the `lakeMorpho` object.

```r
lakeMinorMajorRatio(example_lakeMorpho, addLine = TRUE)
```

```
## [1] 0.5263261
```

lakeShorelineDevelopment The shoreline development metric provides a measure of the complexity of the shoreline. It is a ratio the perimeter of the lake to the perimeter of a circle of the same area. Values will be 1 or greater with value of 1 indicating a circular lake. This metric is used as an indicator of potential habitat [21]. It only requires a `lakeMorpho` object as input.

```r
lakeShorelineDevelopment(example_lakeMorpho)
```

```
## [1] 3.198502
```

lakeShorelineLength and lakeSurfaceArea Shoreline length is simply the total perimeter of the lake polygon and as with all other functions requires a `lakeMorpho` object as input. To calculate the shoreline length:

```
lakeShorelineLength(example_lakeMorpho)
```

```
## [1] 45991.38
```

Similarly, surface area for a lake is the total area of the lake polygon. It is calculated via:

```
lakeSurfaceArea(example_lakeMorpho)
```

```
## [1] 16453180
```

lakeVolume The `lakeVolume` function uses maximum lake depth (see lakeMaxDepth) and methods outlined by Hollister *et al.* [8] to estimate lake volume. The method uses the ratio of the maximum depth to the maximum distance…

Future plans

sf pointDens

Software Availability

The `lakemorpho` version 1.1.0 package is currently available directly from the Comprehensive R Archive Network (CRAN) and may simply be installed and loaded in R via:

```
install.packages('lakemorpho')
library('lakemorpho')
```

To access the help pages (including a version of this manuscript) use.

```r
help(package='lakemorpho')
```

There are tentative plans to continue developing new functions for `lakemorpho` and these new features will be available first through the development version on GitHub at http://github.com/usepa/lakemorpho. To install and load the development version requires use of the `devtools` package. This may be done with:

```r
install.packages('devtools')
library('devtools')
install_github('USEPA/lakemorpho')
library(lakemorpho)
```

Figures

References

1. Wetzel R (2001) Limnology, 3 e. lake and river ecosystems. Academic Press, California. 850 p.

2. Vollenweider RA (1975) Input-output models. Schweizerische Zeitschrift für Hydrologie 37: 53–84.

3. Milstead WB, Hollister JW, Moore RB, Walker HA (2013) Estimating summer nutrient concentrations in northeastern lakes from sparrow load predictions and modeled lake depth and volume. PloS one 8: e81457.

4. Genkai-Kato M, Carpenter SR (2005) Eutrophication due to phosphorus recycling in relation to lake

morphometry, temperature, and macrophytes. Ecology 86: 210–219.

5. Kalff J (2002) Limnology: Inland water ecosystems. Prentice Hall New Jersey. 592 p.

6. Welch P (1935) Limnology. McGraw-Hill, New York.

7. Wetzel RG, Likens G (2000) Limnological analyses 3rd editon. Springer Verlag, New York.

8. Hollister J, Milstead WB (2010) Using gis to estimate lake volume from limited data. Lake and Reservoir Management 26: 194–199.

9. Hollister JW, Milstead WB, Urrutia MA (2011) Predicting maximum lake depth from surrounding topography. PLoS ONE 6: e25764. Available: http://dx.doi.org/10.1371/journal.pone.0025764. Accessed 28 Jun 2013.

10. USEPA U (2005) National hydrography dataset plus–NHDPlus.

11. Gesch D, Evans G, Mauck J, Hutchinson J, Carswell Jr W (2009) The national map-elevation: US geological survey fact sheet 2009-3053, 4 p.

12. Bivand R, Lewin-Koh N (2014) Maptools: Tools for reading and handling spatial objects. Available: http://CRAN.R-project.org/package=maptools.

13. Bivand R, Keitt T, Rowlingson B (2014) Rgdal: Bindings for the geospatial data abstraction library. Available: http://CRAN.R-project.org/package=rgdal.

14. Bivand R, Rundel C (2014) Rgeos: Interface to geometry engine - open source (geos). Available: http://CRAN.R-project.org/package=rgeos.

15. Bivand RS, Pebesma EJ, Gómez-Rubio V (2008) Applied spatial data analysis with r. Springer.

16. Pebesma EJ, Bivand RS (2005) Classes and methods for spatial data in r. R news 5: 9–13.

17. Hijmans RJ (2014) Raster: Raster: Geographic data analysis and modeling. Available: http://CRAN.R-project.org/package=raster.

18. Hijmans RJ (2014) Geosphere: Spherical trigonometry. Available: http://CRAN.R-project.org/

package=geosphere.

19. GDAL Development Team (2012) GDAL - geospatial data abstraction library, version 1.9.2. Open Source Geospatial Foundation. Available: http://www.gdal.org.

20. Foundation OSG (2013) GEOS - geometry engine - open source. Open Source Geospatial Foundation. Available: http://trac.osgeo.org/geos/.

21. LAKEWATCH F (2001) Department of fisheries and aquatic sciences, a beginner's guide to water management-lake morphometry.