

Choosing which matrices to work with

2015-11-08

This example introduces some of the different ways to subset the COMADRE database, and adds some other important things you should think about.

The comadre metadata has lots of useful information that can be used to subset the data. Such as: taxonomy (Order, Family, Genus, etc.), time interval of projection (AnnualPeriodicity), spatial replication (NumberPopulations), stage-structure (MatrixCriteria...), location (Lat..., Lon..., Altitude, Country, Continent), Environment (Ecoregion, Season, etc.), matrix properties (MatrixComposite, MatrixFec, etc.) and many more. See the comadre user guide for a list.

Here is an example of a subset of the data. subset the database to the data of interest: only mean matrices for bony fish from studies of 3 years duration or longer, and with a matrix dimension of 3 or greater.

```
tempMetadata <- subset(comadre$metadata,  
                        MatrixComposite == "Mean" &  
                        Class == "Actinopterygii" &  
                        StudyDuration >= 3 &  
                        MatrixDimension > 3)
```

The row names from the subsetting dataframe can now be used to subset the entire comadre database using the function subsetDB which is available via GitHub (using source_url from devtools to download it).

```
library(devtools)  
source_url("https://raw.githubusercontent.com/jonesor/compadreDB/master/  
Functions/subsetDB.R")  
  
id <- as.numeric(rownames(tempMetadata))  
x<-subsetDB(comadre,id)
```

The object x is now a version of the comadre database object that contains only the matrices that match the search criteria.

Matrix size and the study duration are common things you might want to think about when choosing matrices to use. The more stage classes the matrix has, often the more detailed the population dynamics are. On the other hand, if a matrix has many stages then it may mean that the number of data points used to calculate the vital rates for each stage is fewer.

Have a go at subsetting the data according to something you're interested in. For example, only mammals in Asia, or only matrices for populations outside the tropics, or only matrices which describe species which have clonal reproduction.

Another very important thing to think about is the mathematical properties of the matrix. Most analyses assume that matrices are PRIMITIVE, IRREDUCIBLE and ERGODIC. Matrices that are IMPRIMITIVE, REDUCIBLE or NONERGODIC often violate the assumptions of population projection. All nonergodic matrices are reducible and imprimitive and all reducible matrices are imprimitive. If a matrix is reducible it often means that a certain vital rate is missing or has not been recorded. This can cause

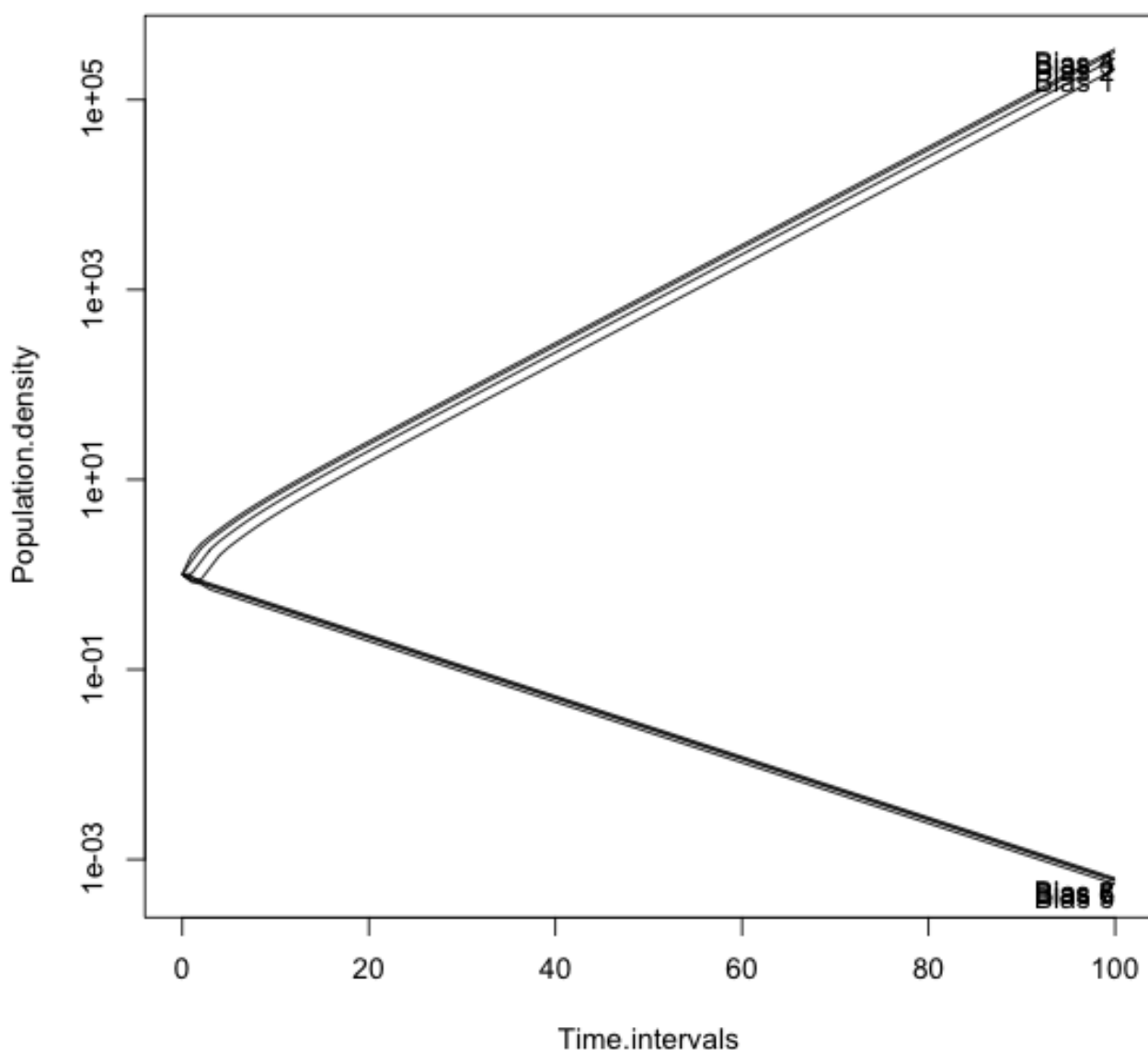
nonergodicity, which is a mathematical problem where the long-term growth depends on the population structure used in the projection. This causes extinction of the population for some population structures.

Let's look at a reducible model.

```
Koaladata <- subset(comadre$metadata,  
                    CommonName == "Koala" &  
                    MatrixPopulation == "Koala Coast" &  
                    MatrixTreatment == "Natural mortality")  
Koalaid <- as.numeric(rownames(Koaladata))  
KoalaMatrix<-subsetDB(comadre,Koalaid)  
KoalaA<-KoalaMatrix$mat[[1]]$matA
```

The package popdemo can project dynamics (you may have to install popdemo)

```
library(popdemo)  
plot(project(KoalaA),log="y")  
## Warning in project(KoalaA): Matrix is reducible
```



The different populations have different long-term growth rates, which means the matrix is nonergodic. This must also mean the matrix is reducible and imprimitive. This is because individuals of stage 5 and above have zero fecundity. We can find out if a matrix is ergodic, irreducible, or primitive using functions in the popdemo package.

```
is.matrix_ergodic(KoalaA)
```

```
## [1] FALSE
```

```
is.matrix_irreducible(KoalaA)
```

```
## [1] FALSE
```

```
is.matrix_primitive(KoalaA)
```

```
## [1] FALSE
```

Generally it is best to recommend NOT to use reducible or nonergodic matrices unless you're sure you know what you're doing!