Getting matrices out of COM(P)ADRE

Owen Jones

20 Nov 2015

This document explains how to get matrices out of the COMADRE and COMPADRE databases.

You will of course need to download the database first. You can get it from the website here: http://www.compadre-db.org/. Click on **Data**, and then on the **COMADRE Animal Matrix Database**, or **COMPADRE Plant Matrix Database** button to find it. Download it to your computer.

The two databases are identical in gross structure, so the following code will apply to both.

Let's imagine you want to get matrices for all *Ursus maritimus* (polar bear) species from the database.

Firstly you will need to load the data file (called `COMADRE_v.x.x.x.RData`) into RStudio. You can do from within RStudio by clicking *File*, *Open file...*, then navigating to find the file, selecting it and clicking *Open*. Alternatively you can use the `load` function:

```
load("~/Downloads/COMADRE_v.1.0.0.RData")
```

Once you have loaded the file, you should see it in your `Environment`. It will be called `comadre`.

You can check the structure of the object by typing the following, whereupon you will see there are four parts:

```
names(comadre)
```

```
> [1] "metadata"    "matrixClass" "mat"         "version"
```

The `metadata` part is a dataframe that contains information about the matrix models contained in the database.

There are lots of columns in the dataframe, and you could subset the data by any of them.

```
names(comadre$metadata)
```

```
>  [1] "SpeciesAuthor"          "SpeciesAccepted"
>  [3] "CommonName"             "CoLCheckOK"
>  [5] "CoLCheckDate"           "Infraspecific"
>  [7] "SpeciesEpithetAccepted" "GenusAccepted"
>  [9] "GenusAuthor"            "Family"
> [11] "Order"                  "Class"
> [13] "Phylum"                 "Kingdom"
> [15] "Authors"                "Journal"
> [17] "YearPublication"        "DOI.ISBN"
> [19] "AdditionalSource"       "StudyDuration"
> [21] "StudyStart"             "StudyEnd"
> [23] "AnnualPeriodicity"      "NumberPopulations"
> [25] "MatrixCriteriaSize"     "MatrixCriteriaOntogeny"
> [27] "MatrixCriteriaAge"      "MatrixPopulation"
> [29] "LatDeg"                 "LatMin"
> [31] "LatSec"                 "LatNS"
> [33] "LonDeg"                 "LonMin"
> [35] "LonSec"                 "LonWE"
> [37] "Altitude"               "Country"
> [39] "Continent"              "Ecoregion"
> [41] "StudiedSex"             "MatrixComposite"
```

```
> [43] "MatrixTreatment"        "MatrixCaptivity"
> [45] "MatrixStartYear"        "MatrixStartSeason"
> [47] "MatrixStartMonth"       "MatrixEndYear"
> [49] "MatrixEndSeason"        "MatrixEndMonth"
> [51] "MatrixSplit"            "MatrixFec"
> [53] "Observation"            "MatrixDimension"
> [55] "SurvivalIssue"
```

You can find the details of what these columns are in the User Guides.

Since we are interested in getting data for a species matching a particular name (*Ursus maritumus*), we'll need to examine the column called `SpeciesAccepted`, which is the currently accepted binomial species name.

You can use the command `grep` to identify where in the database the species is:

```
grep("Ursus_maritimus",comadre$metadata$SpeciesAccepted)
```

```
> [1] 1428 1429 1430 1431 1432 1433
```

```
grep("Rangifer",comadre$metadata$SpeciesAccepted)
```

```
> [1] 1333 1334 1335
```

Note that the database uses an underscore instead of spaces!

So, these row numbers are where the data are: 1428, 1429, 1430, 1431, 1432, 1433.

To get the matrix for any of these we need to access the `mat` part of the database.

Let's look at the first one, at position 1428.

```
comadre$mat[[1428]]
```

```
> $matA
>           A1      A2      A3      A4      A5      A6
> [1,] 0.00000 0.00000 0.00000 0.00000 0.00000 0.45856
> [2,] 0.85454 0.00000 0.00000 0.00000 0.00000 0.00000
> [3,] 0.00000 0.85454 0.00000 0.00000 0.00000 0.00000
> [4,] 0.00000 0.00000 0.85454 0.50112 0.38194 0.90042
> [5,] 0.00000 0.00000 0.00000 0.39930 0.06562 0.00000
> [6,] 0.00000 0.00000 0.00000 0.00000 0.43942 0.00000
>
> $matU
>           U1      U2      U3      U4      U5      U6
> [1,] 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
> [2,] 0.85454 0.00000 0.00000 0.00000 0.00000 0.00000
> [3,] 0.00000 0.85454 0.00000 0.00000 0.00000 0.00000
> [4,] 0.00000 0.00000 0.85454 0.50112 0.38194 0.90042
> [5,] 0.00000 0.00000 0.00000 0.39930 0.06562 0.00000
> [6,] 0.00000 0.00000 0.00000 0.00000 0.43942 0.00000
>
> $matF
>      F1 F2 F3 F4 F5      F6
> [1,]  0  0  0  0  0 0.45856
> [2,]  0  0  0  0  0 0.00000
> [3,]  0  0  0  0  0 0.00000
> [4,]  0  0  0  0  0 0.00000
```

```
> [5,]  0  0  0  0  0 0.00000
> [6,]  0  0  0  0  0 0.00000
>
> $matC
>      C1 C2 C3 C4 C5 C6
> [1,]  0  0  0  0  0  0
> [2,]  0  0  0  0  0  0
> [3,]  0  0  0  0  0  0
> [4,]  0  0  0  0  0  0
> [5,]  0  0  0  0  0  0
> [6,]  0  0  0  0  0  0
```

You will note that there are actually 4 matrices, called `matA`, `matU`, `matF` and `matC`. The matrix most people will want is the **A** matrix. The others are sub-parts of this matrix that represent processes of survival, fertility and clonality. These three matrices sum to the **A** matrix.

You can isolate **just** the **A** matrix like this, renaming it `x`:

```
x <- comadre$mat[[1428]]$matA
x
```

```
>           A1      A2      A3      A4      A5      A6
> [1,] 0.00000 0.00000 0.00000 0.00000 0.00000 0.45856
> [2,] 0.85454 0.00000 0.00000 0.00000 0.00000 0.00000
> [3,] 0.00000 0.85454 0.00000 0.00000 0.00000 0.00000
> [4,] 0.00000 0.00000 0.85454 0.50112 0.38194 0.90042
> [5,] 0.00000 0.00000 0.00000 0.39930 0.06562 0.00000
> [6,] 0.00000 0.00000 0.00000 0.00000 0.43942 0.00000
```

Now you can use this matrix to do the analysis of interest...

But what are the stages in the matrix? That's easy - to obtain the stage information you can consult the relevant index for the `comadre$matrixClass` part of the database object like this:

```
comadre$matrixClass[[1428]]
```

```
>       MatrixClassOrganized    MatrixClassAuthor MatrixClassNumber
> 12079               active              2 years                 1
> 12080               active              3 years                 2
> 12081               active              4 years                 3
> 12082               active      5 + years adult                 4
> 12083               active       Adult with cub                 5
> 12084               active Adult with yearsling                 6
```

Since it's Christmas soon, why not have a look for reindeer (*Rangifer*) in the database?