Analysing the COMPADRE database using R

Mario Vallejo-Marin

March 15 2017

Contents

The COMPADRE database

Is a compilation of plant population matrices. See www.compadre-db.org. Currently it has 7,024 population matrices for thousands of species.

Downloading and reading the data in `R`

You can download the data from the website. The database is in `R` format. Save it to your computer, within the working environment of R, so you can then load it and analyse it.

```
load(file="COMPADRE_v.4.0.1.RData")
```

This will result in a database object called `compadre`. It is a big file of about 45Mb.

The dataframe is arranged in a series of nested levels. At the highest level it has the following components:

```
summary(compadre)
```

```
##             Length Class      Mode
## metadata      47   data.frame list
## matrixClass 7024   -none-     list
## mat         7024   -none-     list
## version        6   -none-     list
```

The `metadata` section contains 47 variables. For a detailed description of the meaning of each of them see the user manual, which is available here.

Accessing the elements of `compadre` can be done using the operator `$`, just as you would do to access the columns of a simple data base.

```
names(compadre$metadata)
```

```
##  [1] "SpeciesAuthor"          "SpeciesAccepted"
##  [3] "CommonName"             "Genus"
##  [5] "Family"                 "Order"
##  [7] "Class"                  "Phylum"
##  [9] "Kingdom"                "OrganismType"
## [11] "DicotMonoc"             "AngioGymno"
## [13] "Authors"                "Journal"
## [15] "YearPublication"        "DOI.ISBN"
## [17] "AdditionalSource"       "StudyDuration"
## [19] "StudyStart"             "StudyEnd"
## [21] "AnnualPeriodicity"      "NumberPopulations"
## [23] "MatrixCriteriaSize"     "MatrixCriteriaOntogeny"
## [25] "MatrixCriteriaAge"      "MatrixPopulation"
## [27] "Lat"                    "Lon"
## [29] "Altitude"               "Country"
## [31] "Continent"              "Ecoregion"
## [33] "StudiedSex"             "MatrixComposite"
## [35] "MatrixTreatment"        "MatrixCaptivity"
## [37] "MatrixStartYear"        "MatrixStartSeason"
## [39] "MatrixStartMonth"       "MatrixEndYear"
## [41] "MatrixEndSeason"        "MatrixEndMonth"
## [43] "MatrixSplit"            "MatrixFec"
## [45] "Observation"            "MatrixDimension"
## [47] "SurvivalIssue"
```

With a bit of tweaking you can interrogate the database to obtain specific information. For example, to find ou how many genera are included in the data base you can do:

```
length(levels(as.factor(compadre$metadata$Genus)))
```

```
## [1] 444
```

And you can print the first ten genera in this list:

```
levels(as.factor(compadre$metadata$Genus))[1:10]
```

```
##  [1] "Abies"        "Abutilon"     "Acacia"       "Acer"
##  [5] "Achillea"     "Aconitum"     "Actaea"       "Actinostemon"
##  [9] "Adenocarpus"  "Adenophora"
```

Data analysis

You can analyse subsets of the data using this general approach:

```
#Subset the dataframe by Genus
mimulus.data<-subset(compadre$metadata, Genus=="Mimulus")
#Row numbers for the data containing Mimulus
keep<-as.numeric(rownames(mimulus.data))
#Subsets the matrices using the "keep" list
mat.mimulus<-compadre$mat[keep]
#This is a list of 36 matrices, which can be accessed using indices
mat.mimulus[[1]]
```

```
## $matA
##                A1           A2           A3           A4
## [1,] 1.972500e-01 938.25000000 9069.0833333 1.322798e+05
## [2,] 3.978583e-05   0.06633333    0.0244500 6.333333e-04
## [3,] 2.722000e-05   0.08722500    0.3366333 2.182167e-01
## [4,] 3.719167e-07   0.01748333    0.1384917 5.829917e-01
##
## $matU
##                U1         U2        U3           U4
## [1,] 1.972500e-01 0.00000000 0.0000000 0.0000000000
## [2,] 3.978583e-05 0.06633333 0.0244500 0.0006333333
## [3,] 2.722000e-05 0.08722500 0.3366333 0.2182166667
## [4,] 3.719167e-07 0.01748333 0.1384917 0.5829916667
```

```
##
## $matF
##      F1     F2       F3       F4
## [1,]  0 938.25 9069.083 132279.8
## [2,]  0   0.00    0.000      0.0
## [3,]  0   0.00    0.000      0.0
## [4,]  0   0.00    0.000      0.0
##
## $matC
##      C1 C2 C3 C4
## [1,]  0  0  0  0
## [2,]  0  0  0  0
## [3,]  0  0  0  0
## [4,]  0  0  0  0
```

```r
#And you can subset them further:
mat.mimulus[[1]]$matC
```

```
##      C1 C2 C3 C4
## [1,]  0  0  0  0
## [2,]  0  0  0  0
## [3,]  0  0  0  0
## [4,]  0  0  0  0
```

```r
sum(mat.mimulus[[1]]$matC)
```

```
## [1] 0
```

Now things more exciting, as we are ready to conduct analyses at a massive scale.

```r
clonal.flag<-numeric()

#Loop to identify which matC entries are >0; i.e., have some measure of clonality
# Excludes indivisible matrices "NA"
n<-length(compadre$metadata$SpeciesAuthor)

for(i in 1:n){
  sum.matC<-sum(compadre$mat[[i]]$matC)
 # print(clonal.flag)
  if(!is.na(sum.matC) & sum.matC > 0) {
    clonal.flag<-c(clonal.flag,i)
    #print(cbind(i, clonal.flag))
    }
}

#List of species with matC > 0
compadre$metadata$SpeciesAccepted[clonal.flag][1:10]
```

```
##  [1] "Murdannia nudiflora"     "Murdannia nudiflora"
##  [3] "Murdannia nudiflora"     "Murdannia nudiflora"
##  [5] "Hylocomium splendens"    "Hylocomium splendens"
##  [7] "Aspasia principissa"     "Heteropsis flexuosa"
##  [9] "Heteropsis macrophylla"  "Heteropsis oblongifolia"
```

```r
length(compadre$metadata$SpeciesAccepted[clonal.flag])
```

```
## [1] 816
```

```
levels(as.factor(compadre$metadata$SpeciesAccepted[clonal.flag]))[1:10]
```

```
##  [1] "Aechmea magdalenae"        "Aechmea nudicaulis"
##  [3] "Allium tricoccum"          "Anthericum liliago"
##  [5] "Anthericum ramosum"        "Arctophila fulva"
##  [7] "Arnica angustifolia"       "Asarum canadense"
##  [9] "Aspasia principissa"       "Astrocaryum aculeatissimum"
```

For example, to calculate lambda:

```
#Calculate lambda for all clonal species:
require(popbio)
```

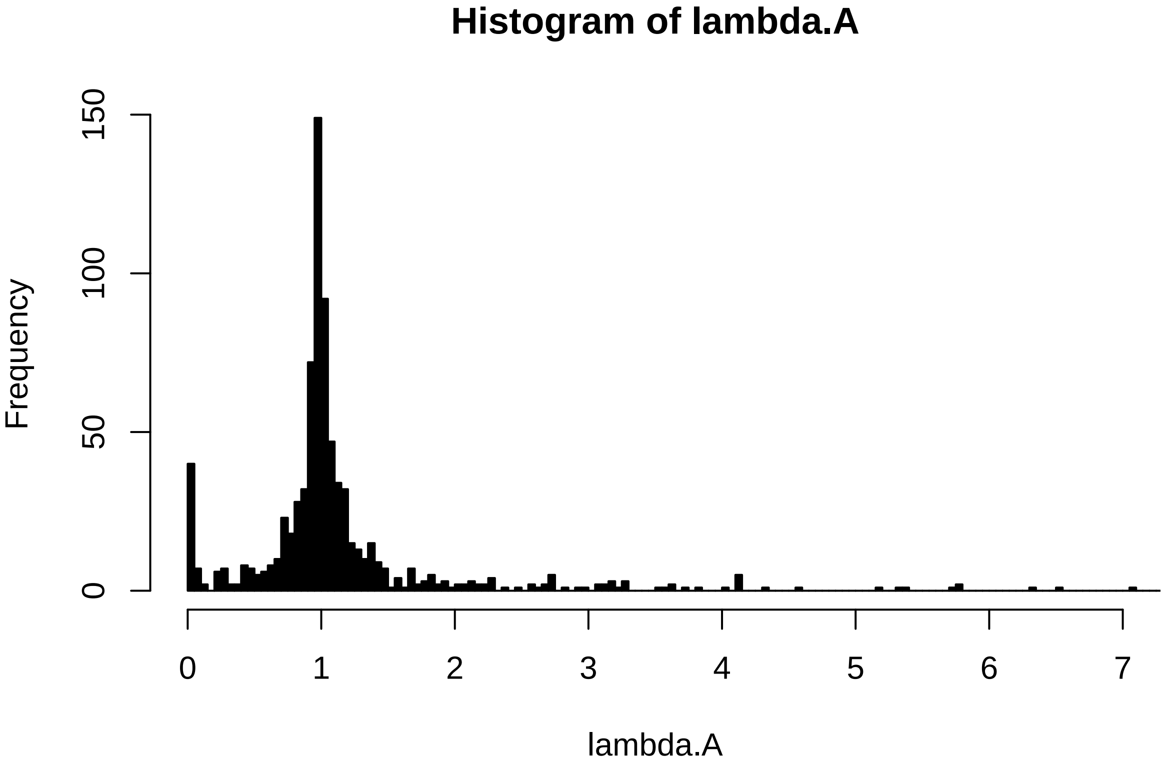
```
## Loading required package: popbio
```

```
n<-length(compadre$metadata$SpeciesAccepted[clonal.flag])
lambda.A<-rep(NA,n)

for(i in 1:n){
  tryCatch({  #Avoids error message due to missing values in x
  lambda.A[i]<-lambda(compadre$mat[[i]]$matA)
  }, error=function(e){})
}
sort(round(lambda.A,3))
```

```
##   [1]   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
##   [9]   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
##  [17]   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
##  [25]   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.010
##  [33]   0.020   0.040   0.042   0.050   0.050   0.050   0.050   0.050
##  [41]   0.055   0.060   0.060   0.060   0.070   0.087   0.088   0.110
##  [49]   0.117   0.211   0.220   0.228   0.230   0.246   0.250   0.266
##  [57]   0.270   0.270   0.276   0.276   0.281   0.294   0.331   0.332
##  [65]   0.351   0.381   0.409   0.410   0.430   0.430   0.430   0.430
##  [73]   0.438   0.443   0.460   0.461   0.462   0.462   0.466   0.471
##  [81]   0.493   0.507   0.510   0.513   0.528   0.537   0.553   0.556
##  [89]   0.565   0.568   0.577   0.600   0.610   0.623   0.625   0.630
##  [97]   0.630   0.630   0.633   0.644   0.674   0.676   0.678   0.680
## [105]   0.687   0.688   0.692   0.694   0.699   0.700   0.708   0.710
## [113]   0.713   0.713   0.714   0.714   0.714   0.714   0.714   0.714
## [121]   0.714   0.714   0.721   0.724   0.729   0.731   0.733   0.736
## [129]   0.740   0.743   0.748   0.749   0.750   0.751   0.753   0.754
## [137]   0.754   0.757   0.758   0.764   0.767   0.769   0.781   0.781
## [145]   0.782   0.784   0.786   0.787   0.787   0.787   0.798   0.802
## [153]   0.803   0.809   0.810   0.810   0.810   0.810   0.813   0.815
## [161]   0.819   0.819   0.822   0.822   0.823   0.823   0.824   0.826
## [169]   0.827   0.830   0.836   0.836   0.838   0.840   0.843   0.845
## [177]   0.847   0.848   0.849   0.852   0.853   0.856   0.858   0.859
## [185]   0.860   0.860   0.861   0.863   0.868   0.868   0.869   0.871
## [193]   0.874   0.875   0.876   0.877   0.877   0.878   0.881   0.882
## [201]   0.882   0.883   0.885   0.888   0.890   0.892   0.893   0.894
## [209]   0.895   0.896   0.897   0.901   0.902   0.902   0.906   0.906
## [217]   0.907   0.909   0.912   0.912   0.913   0.913   0.915   0.918
## [225]   0.920   0.920   0.921   0.922   0.923   0.923   0.926   0.926
## [233]   0.926   0.927   0.927   0.927   0.927   0.927   0.927   0.927
## [241]   0.927   0.927   0.927   0.927   0.927   0.927   0.927   0.927
## [249]   0.927   0.927   0.927   0.928   0.928   0.928   0.929   0.932
## [257]   0.933   0.933   0.933   0.934   0.934   0.934   0.934   0.934
## [265]   0.935   0.936   0.937   0.939   0.941   0.941   0.941   0.942
## [273]   0.944   0.946   0.946   0.947   0.948   0.948   0.949   0.949
## [281]   0.950   0.950   0.950   0.950   0.952   0.953   0.953   0.955
## [289]   0.955   0.956   0.957   0.958   0.959   0.960   0.960   0.960
## [297]   0.961   0.962   0.964   0.964   0.965   0.965   0.965   0.966
## [305]   0.966   0.967   0.967   0.968   0.969   0.969   0.970   0.970
## [313]   0.972   0.973   0.974   0.974   0.974   0.975   0.975   0.976
## [321]   0.976   0.976   0.977   0.977   0.978   0.978   0.979   0.980
## [329]   0.980   0.981   0.981   0.983   0.983   0.984   0.984   0.985
## [337]   0.986   0.986   0.986   0.987   0.987   0.989   0.989   0.989
## [345]   0.990   0.990   0.990   0.990   0.991   0.991   0.991   0.991
## [353]   0.992   0.994   0.994   0.994   0.995   0.996   0.998   0.998
## [361]   0.999   0.999   0.999   0.999   0.999   0.999   0.999   1.000
## [369]   1.000   1.000   1.000   1.000   1.000   1.000   1.000   1.000
## [377]   1.000   1.000   1.000   1.000   1.000   1.000   1.000   1.000
## [385]   1.000   1.000   1.000   1.000   1.000   1.000   1.000   1.000
## [393]   1.000   1.000   1.000   1.000   1.000   1.000   1.000   1.000
## [401]   1.000   1.000   1.000   1.000   1.000   1.000   1.000   1.000
## [409]   1.000   1.000   1.000   1.000   1.000   1.000   1.000   1.000
## [417]   1.000   1.000   1.000   1.000   1.000   1.000   1.000   1.000
## [425]   1.000   1.000   1.000   1.000   1.000   1.000   1.000   1.000
## [433]   1.000   1.000   1.001   1.001   1.002   1.002   1.003   1.003
## [441]   1.004   1.004   1.004   1.005   1.005   1.006   1.007   1.007
## [449]   1.007   1.008   1.009   1.009   1.009   1.010   1.010   1.010
## [457]   1.011   1.011   1.011   1.011   1.014   1.014   1.014   1.016
## [465]   1.017   1.018   1.019   1.019   1.019   1.020   1.021   1.021
## [473]   1.024   1.024   1.025   1.025   1.025   1.026   1.026   1.027
## [481]   1.029   1.029   1.030   1.030   1.031   1.032   1.032   1.033
## [489]   1.033   1.034   1.034   1.034   1.035   1.036   1.036   1.036
## [497]   1.037   1.037   1.038   1.038   1.038   1.038   1.039   1.040
## [505]   1.040   1.040   1.043   1.043   1.043   1.043   1.043   1.043
## [513]   1.044   1.044   1.044   1.045   1.045   1.045   1.046   1.048
## [521]   1.049   1.049   1.049   1.050   1.051   1.051   1.052   1.053
## [529]   1.053   1.054   1.056   1.057   1.060   1.062   1.064   1.064
## [537]   1.065   1.066   1.066   1.066   1.067   1.068   1.068   1.069
## [545]   1.072   1.073   1.075   1.076   1.076   1.078   1.078   1.078
## [553]   1.078   1.080   1.081   1.085   1.086   1.087   1.088   1.090
## [561]   1.091   1.091   1.092   1.093   1.094   1.096   1.096   1.096
## [569]   1.097   1.099   1.100   1.101   1.101   1.102   1.104   1.104
## [577]   1.105   1.105   1.109   1.114   1.114   1.116   1.117   1.118
## [585]   1.120   1.121   1.121   1.122   1.124   1.125   1.126   1.127
## [593]   1.130   1.130   1.132   1.133   1.133   1.134   1.136   1.138
## [601]   1.141   1.144   1.149   1.149   1.150   1.153   1.154   1.155
## [609]   1.156   1.156   1.159   1.160   1.163   1.163   1.164   1.167
## [617]   1.167   1.169   1.169   1.171   1.172   1.173   1.173   1.175
## [625]   1.181   1.181   1.181   1.184   1.184   1.189   1.191   1.192
## [633]   1.194   1.194   1.195   1.199   1.199   1.201   1.215   1.217
## [641]   1.219   1.224   1.225   1.231   1.231   1.232   1.232   1.237
## [649]   1.238   1.243   1.247   1.249   1.252   1.254   1.256   1.258
## [657]   1.259   1.262   1.263   1.270   1.278   1.282   1.284   1.284
## [665]   1.299   1.303   1.305   1.306   1.310   1.310   1.318   1.324
## [673]   1.334   1.339   1.340   1.354   1.357   1.359   1.365   1.366
## [681]   1.372   1.375   1.375   1.375   1.376   1.377   1.381   1.388
## [689]   1.394   1.400   1.404   1.408   1.409   1.410   1.412   1.415
## [697]   1.418   1.425   1.439   1.451   1.471   1.478   1.478   1.480
## [705]   1.489   1.496   1.540   1.553   1.567   1.576   1.588   1.620
## [713]   1.673   1.679   1.683   1.683   1.685   1.686   1.689   1.703
## [721]   1.737   1.761   1.774   1.793   1.804   1.810   1.813   1.823
## [729]   1.832   1.860   1.867   1.906   1.907   1.925   1.976   2.003
## [737]   2.036   2.054   2.085   2.111   2.116   2.116   2.154   2.168
## [745]   2.225   2.227   2.264   2.289   2.293   2.300   2.380   2.494
## [753]   2.558   2.564   2.614   2.658   2.679   2.702   2.723   2.723
## [761]   2.739   2.744   2.810   2.946   2.963   3.058   3.097   3.124
## [769]   3.124   3.161   3.175   3.199   3.220   3.275   3.278   3.293
## [777]   3.509   3.592   3.613   3.631   3.731   3.829   4.028   4.134
## [785]   4.134   4.134   4.134   4.137   4.309   4.581   5.162   5.341
## [793]   5.399   5.748   5.755   5.789   6.329   6.537   7.088   7.682
## [801]   7.774   7.900   8.119  10.112  10.262  16.451  17.242  39.363
## [809]  43.321  46.854  48.744  53.362 238.061 272.359 320.544
```

```
hist(lambda.A, breaks="fd", xlim=c(0,7), col="black") #Omits from the plot all values of lambda>7
```


Histogram of lambda.A
Frequency
lambda.A