

Produce ternary plots of elasticities

2016-06-18

Here we produce a ternary plot *a la* Silvertown & Franco (1993) with various life history traits such as mean life expectancy, population growth rate or reactivity as the “fourth” dimension. We will use Caswell (2001) formulation of mean life expectancy from the fundamental matrix (N), and the package `popdemo` for the reactivity. Other packages we need include `Mage`, `fields`, `vcd` and `scales`:

```
library(popdemo)
library(fields)
library(Mage)
library(vcd)
library(scales)
```

This is the function to calculate mean life expectancy from Caswell (2001):

```
meanLifeExpectancy <- function(matU = matU, startLife = 1){
  uDim=dim(matU)[1]
  N = solve(diag(uDim[startLife])-matU)
  eta = colSums(N)[startLife]
  return(eta)
}
```

As an example for the chosen data, subset COMADRE to studies with a matrix dimension ≥ 3 , that represent mean, unmanipulated conditions duration > 3 years, where sexual reproduction has been modeled explicitly, the matrices are split into U, F and C, and there are no issues with stage-specific survival >1 .

```
x <- subsetDB(comadre, MatrixDimension >= 3
              & MatrixComposite == "Mean" & MatrixTreatment == "Unmanipulated"
              & StudyDuration > 3 & MatrixFec == "Yes"
              & MatrixSplit == "Divided" & SurvivalIssue < 1)
```

This object (`x`) is now a copy of the database that contains ONLY the matrices of interest.

We can ask how many matrices this is by looking at the size of the metadata part.

```
nrow(x$metadata)
```

```
> [1] 163
```

These matrices can now be analyzed by applying functions in a loop, or by using `lapply`.

To calculate elasticities, population growth rate, reactivity and mean life expectancy for the subset matrices, first create an empty `data.frame` to accommodate the output:

```
output <- data.frame(species= rep(NA, nrow(x$metadata)),
                     lambda = rep(NA, nrow(x$metadata)),
                     eta = rep(NA, nrow(x$metadata)),
                     react = rep(NA, nrow(x$metadata)),
                     EStasis = rep(NA, nrow(x$metadata)),
                     EProgression = rep(NA, nrow(x$metadata)),
                     ERetrogression = rep(NA, nrow(x$metadata)),
                     EFecundity = rep(NA, nrow(x$metadata)),
                     EClonality = rep(NA, nrow(x$metadata)))
```

We will use the `Mage` function `matrixElementPerturbation` to calculate element-level perturbations for each matrix in a for loop:

```
for (i in 1:nrow(x$metadata)){
  tryCatch({
    output$species[i] <- x$metadata$SpeciesAuthor[i]
    output$lambda[i] <- max(Re(eigen(x$mat[[i]]$matA)$value))
    output$eta[i] <- meanLifeExpectancy(matU = x$mat[[i]]$matU, startLife = 1)
    output$react[i] <- reactivity(x$mat[[i]]$matA)
    output[i,c("EStasis", "EProgression", "ERetrogression",
              "EFecundity", "EClonality")] <-
      matrixElementPerturbation(matU = x$mat[[i]]$matU, matF = x$mat[[i]]$matF,
                               matC = x$mat[[i]]$matC)[6:10]
  }, error = function(e){})
}
```

Now we can group elasticities of population growth rate to various demographic processes into three axes of our ternary plot:

```
output$S <- output$EStasis + output$ERetrogression
output$G <- output$EProgression
output$R <- output$EFecundity + output$EClonality
```

We need to scale the coordinates of each point between 0 and 1 - this is necessary due to possible rounding issues:

```
output$S <- output$S / rowSums(output[, c("S", "G", "R")])
output$G <- output$G / rowSums(output[, c("S", "G", "R")])
output$R <- output$R / rowSums(output[, c("S", "G", "R")])
```

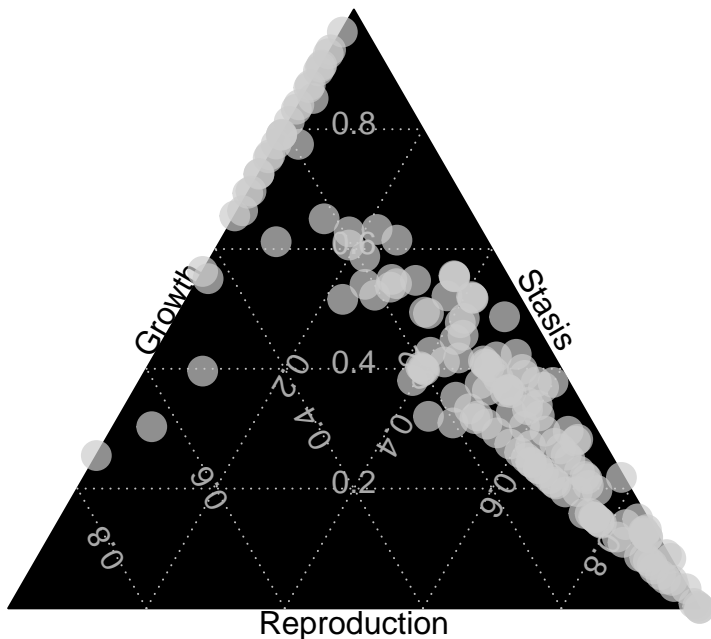
Then we must eliminate the few MPMs where the code did not run correctly, and produced NA values:

```
output <- output[-which(is.na(output$eta)), ]
```

Now we can finally plot the locations of the chosen matrices in a preliminary ternary plot using the function `ternaryplot` from the package `vcd`. The following code also uses the function `alpha` from the `scales` package to modify the colour transparency of the points.

```
ternaryplot(output[,c("R", "S", "G")], scale=1, col=alpha("gray80", 0.7), bg="black",
            dimnames=c("Stasis", "Growth", "Reproduction"), dimnames_position="edge",
            main="Preliminary plot")
```

Preliminary plot



As a use extension to this ternary plot, we can colour-code the points according to the values of λ , η or reactivity. To do that we first need to set up a colour palette for each of these measures, and deduce which colour each point should have.

In every case we can use the same `colorRamp` function to deduce the appropriate colour:

```
col_fun <- colorRamp(c("white", "yellow", "orange", "red", "dark red"))
```

First, λ . For this measure we should first subset out the data with unreasonable values (say > 7):

```
lambdaData <- subset(output, lambda < 7)
```

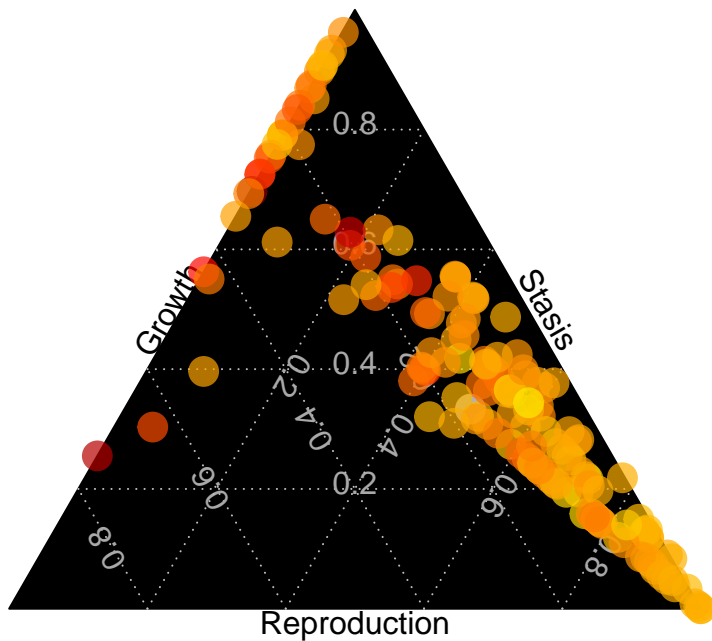
We then log transform and standardize the values, and obtain the hex colour codes for each point using the `rgb` function:

```
Z <- log(lambdaData$lambda)
zNorm <- (Z - min(Z))/(max(Z) - min(Z))
colsLambda <- rgb(col_fun(zNorm), maxColorValue = 256)
```

We can now plot this diagram:

```
ternaryplot(lambdaData[, c("R", "S", "G")], scale = 1,
  col = alpha(colsLambda, 0.7), bg = "black",
  dimnames = c("Stasis", "Growth", "Reproduction"),
  dimnames_position = "edge", main =
    expression(paste("Population growth rate - ", lambda)))
```

Population growth rate – λ



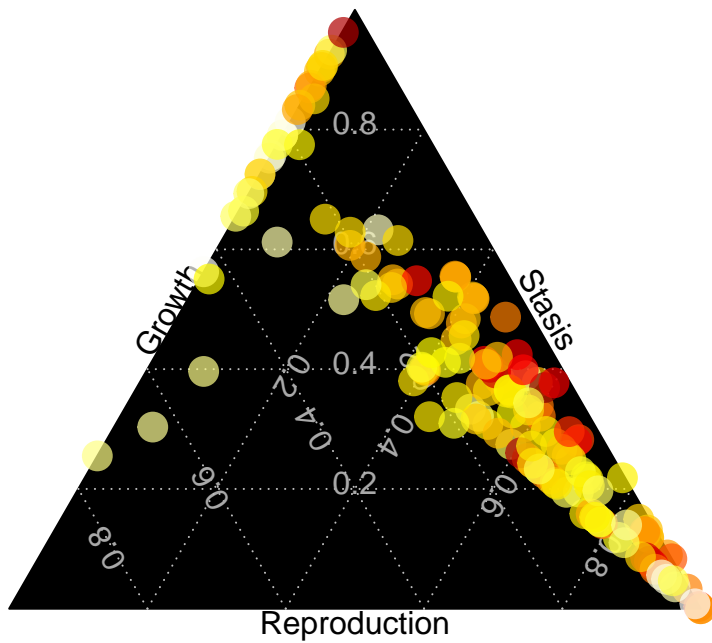
Second, mean life expectancy (eta):

```
etaData <- output
Z <- log(etaData$eta)
zNorm <- (Z - min(Z))/(max(Z) - min(Z))
colsEta <- rgb(col_fun(zNorm), maxColorValue = 256)
```

For which the plot can be generated like this, as before:

```
ternaryplot(etaData[, c("R", "S", "G")], scale = 1,
  col = alpha(colsEta, 0.7), bg = "black",
  dimnames = c("Stasis", "Growth", "Reproduction"),
  dimnames_position = "edge",
  main=expression(paste("Mean life expectancy - ", eta["e"])))
```

Mean life expectancy – η_e



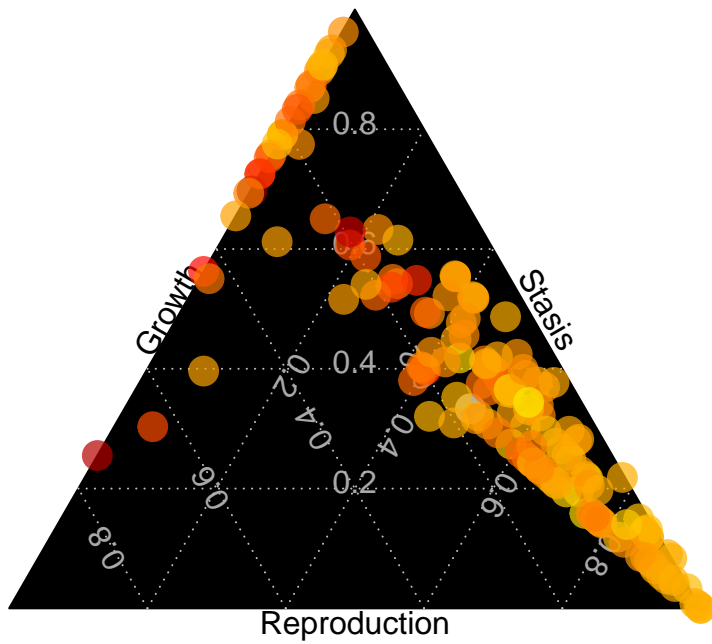
Thirdly, reactivity:

```
reactData <- subset(output, react < 140)
Z <- log(reactData$lambda)
zNorm <- (Z - min(Z))/(max(Z) - min(Z))
colsReact <- rgb(col_fun(zNorm), maxColorValue = 256)
```

And the plot:

```
ternaryplot(reactData[, c("R", "S", "G")], scale = 1,
  col = alpha(colsReact, 0.7), bg = "black",
  dimnames = c("Stasis", "Growth", "Reproduction"),
  dimnames_position = "edge",
  main=expression(paste("Reactivity - ||", hat(A), "||"[1])))
```

Reactivity – $\|\hat{A}\|_1$



One could add the color scale legend using the following code to add to the same plot. Note though, that here the value of the points has been standardised to range from 0 to 1, rather than the original $\lambda/\eta/\text{reactivity}$ values:

```
colCode <- colorRampPalette(c("white", "yellow", "orange", "red", "dark red"))(n = 999)
image.plot(legend.only = TRUE, zlim = range(zNorm), col = colCode,
           smallplot = c(.75, .8, .5, .75), cex.axis=0.2)
```