Characterising centroids of simulated species

Gerardo Martín

17/9/2020

Introduction

These analyses show how traditional characterisation of niche centroids may reduce its power depending on the statistical distribution of the environmental layers used to infer species' fundamental niches. The excercise is based purely on simulated datasets, incuding the environmental raster layers.

Simulating "environmental" layers

The latter ones were simulated using a multivariate normal distribution number generator where the shared variance among generated numbers depends on an exponential kernel (a special case of the Mattérn function). If x_1, x_2 are two data points, the covariance function for those points is:

$$\operatorname{cov}(x_1, x_2) = \sigma^2 \exp\left(-\lambda \cdot \bar{s_{x_1,x_2}}\right)$$

where σ^2 is the variance of the entire layer being simulated, $\bar{s}_{x_1,x_2}$ is the linear euclidean distance pairs of points x_1, x_2, and λ is the decay rate of the expected variance between pixels. Large λ values result in layers with little spatial autocorrelation.

Using this method I generated five sets of ten normally distributed layers. Each of the five sets differred in the settings used to generate each layer. Each layers' mean, σ and λ were generated randomly. Two sets had relatively large means, variances and decay rates, and the remaining three sets had small means. The layers with small means were used to generate the variables with log-normal, beta and gamma distributions, using each the log-normal layers as the shape/rate parameters of the latter distributions.

Simulating species

For this first excercise I generated all the possible combinations of three layers from the final 100 layers. Then randomly sampled 1000 layer combinations and extracted the mode using the estimated kernel density from that layer, thus the centroid represents the highest probability interval for that layer on the basis of the assumed optimal bandwidth of the kernel density estimation.

Then using the extracted modes as a centroid I calculated the covariance between variables to project into a single raster layer the Mahalanobis distance to the estimated centroid. The resulting distance layer was transformed to a probability of presence layer using a logistic transform:

$$P = \frac{\exp(-\text{dist})}{1 + \exp(-\text{dist})}$$

Then I generated random points in the probability surface using the `randomPoints` function of the **dismo** package. The number of points generated was random and was in average 500 with a standard deviation of 90.

Characterising centroids

Traditional approach

I tried to retrieve the centroids of the simulated species occurrences using the **ntbox** package function `ellipsoidfit`. Then from the fitted ellipsoid objects I extracted the estimated centroids and the distance to centroid layer.

Poisson point process model approach

I fitted a Poisson point process model to each generated species using the same model formula for all species:

$$\log(\lambda) = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta'_1 X_1^2 + \beta'_2 X_2^2 + \beta'_3 X_3^2$$

where the X's are the environmental layers used to simulate each species, α is the model intercept, and β's are each layer effects. In this case λ refers to point intensity, in contrast with the exponential covariance function.

To extract the centroids from this model formula after estimating α and β's I obtained the partial derivative of $\log \lambda$ with respect to each X:

$$\frac{\partial \log(\lambda)}{\partial X_i} = \beta_i + 2 \cdot \beta'_i X_i$$

And then solved for $\partial \log \lambda / \partial X_i = 0 = X_i^*$, which corresponds to the X_i value where point intensity is the highest:

$$X_i^* = -\frac{\beta_i}{2 \cdot \beta'_i}$$

This approach works if and only if $\beta'_i < 0$.

Comparing results

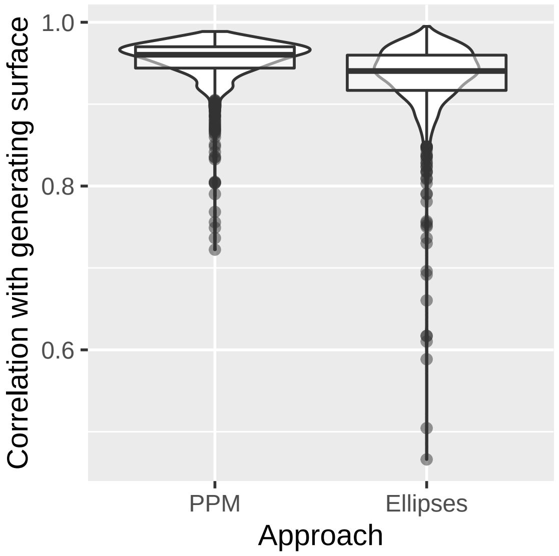
First, with the centroids estimated from each approach (traditional and ppm), I measured the Mahalanobis distance to the real centroid used for simulating species. Then I compared the surfaces generated by projecting the distance to the estimated centroid from the traditional approach and the estimated point intensity from the PPM approach and estimated the correlation between the surfaces.

Then I measured how the distance from the estimated centroids to the real centroids affected the similarity between model-generated surfaces and the surface used to simulate the occurrence records.

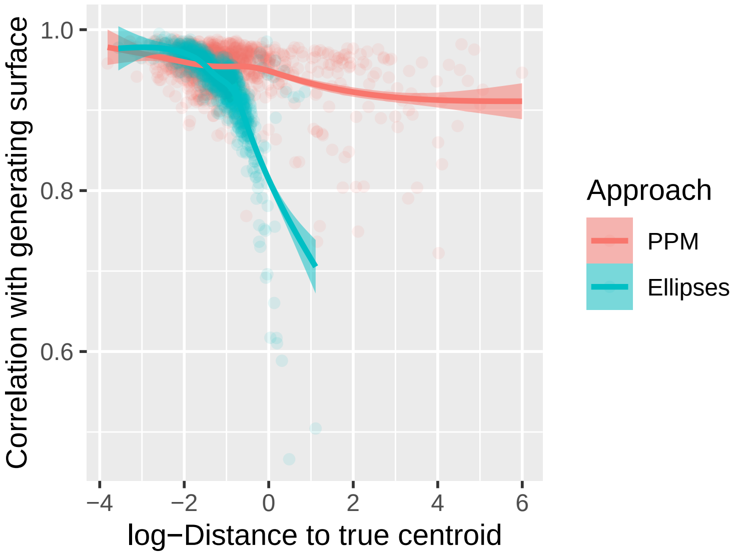
This is still an incomplete analysis as I'm yet to incorporate the type of distributions used for generating each species.

Results

Surfaces generated by PPMs tended to be more similar to the surface that generated the data:



This is because the centroids characterised with PPMs tended to be closer to the real centroid than centroids characterised by ellipses:



A big problem for PPMs however, is that outliers, on the basis of distance to real centroids, are a lot farther away from the real centroids than for ellipses, and this is probably due to the polynomial nature of the formulas (i. e. $\beta_i' > 0$ will throw away any possible estimation of the centroid). By visual inspection of the above plots one would be fooled to think that higher correlation with the generating surface is enough to conclude that the approximation to the real centroid is better, because PPMs are almost always more correlated with the generating surface. But as I mentioned above, when $\beta_i' > 0$, ellipses are a better guess than PPMs.

I calculated the fraction of cases in which PPMs *outperformed* ellipses, on the basis of the correlation with the generating surface and the distance to the real centroid, by substracting the correlation coefficient and the distance to true centroids respectively, from those scores for the PPMs. PPMs in fact usual are twice highly correlated with the generating surface than ellipsoids:

```
corr.dif <- ppms$Corr.surf - ellips$Corr.surf
dist.dif <- ppms$Dist.true.cent - ellips$Dist.true.cent
```

```
p.cor <- length(which(corr.dif >0))/length(corr.dif)
p.dif <- length(which(dist.dif > 0))/length(dist.dif)

p.cor/(1-p.cor)
```

```
## [1] 2.125
```

But regarding distance from the real centroid, the difference only marginally favours PPMs.

```
p.dif/(1-p.dif)
```

```
## [1] 1.123142
```

That is distance from real centroids is only smaller for PPMs in 7% of the cases.

As mentioned above I will explore in greater detail the patterns in these synthetic analyses, first by accounting for the distribution of the variables used in the generating surfaces, and then trying to improve PPM estimates by tweaking model formulas.

Exploring the best and worst cases

For these analyses I used the distance to the real centroid. On this basis it turns out that both methods worked best for exactly the same case:

```
ellip.best <- which.min(ellips$Dist.true.cent)
ppm.best <- which.min(ppms$Dist.true.cent)

ellip.worst <- which.max(ellips$Dist.true.cent)
ppm.worst <- which.max(ppms$Dist.true.cent)
```

```
ellip.best
```

```
## [1] 720
```

```
ppm.best
```

```
## [1] 465
```

The best-performing ellipse model turns out to have been generated with normally distributed variables (symmetric):

```
ellips[ellip.best,]
```

```
##     Corr.surf Dist.true.cent Normal Log.norm Beta Gamma approach
## 720 0.9774509   0.0002760568      1        0    2     0 Ellipses
```

And the best ppm included only one non-normal variable:

```
ppms[ppm.best, ]
```

```
##     Corr.surf Dist.true.cent Normal Log.norm Beta Gamma approach
## 465 0.9579695   0.0001532864      2        1    0     0      PPM
```

The columns with the distribution's names indicate the number of layers of that distribution used for generating the *species*.

And in the worst case, the *species* were

```
ellip.worst
```

```
## [1] 106
```

```
ppm.worst
```

```
## [1] 339
```

And were generated with the distributions, for the case of ellipses:

```
ellips[ellip.worst,]
```

```
##     Corr.surf Dist.true.cent Normal Log.norm Beta Gamma approach
## 106 0.5043596       12.87841      0        1    1     1 Ellipses
```

and for ppms:

```
ppms[ppm.worst, ]
```

```
##     Corr.surf Dist.true.cent Normal Log.norm Beta Gamma approach
## 339 0.9464732       988909.3      0        1    1     1      PPM
```

To no surprise both worst cases were generated with potentially asymmetric distributions.

Details of the models for the worst cases

PPMs I'll explore first the coefficients of the model:

```
coef(ppm.worst.mod)
```

```
##   (Intercept)             a             b             c        I(a^2)
## -1.132844e+01 -2.078113e+00  2.062672e+01  6.053345e-02  1.954668e-03
##        I(b^2)        I(c^2)
## -9.227438e+00 -3.445335e-04
```

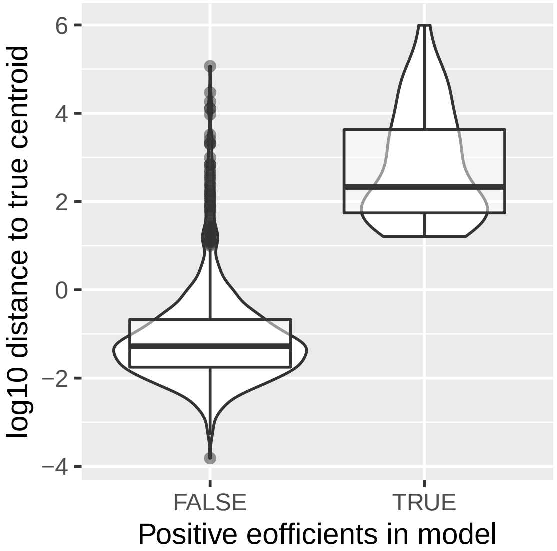
And as it turns out the estimate for $\beta'_a > 0$, which explains why the estimate of the centroid is so far off the real value.

In contrast with this model which performed so poorly on the basis of the estimated centroid, the model that got closest to the real value, estimated negative coefficients for all β'_i's:

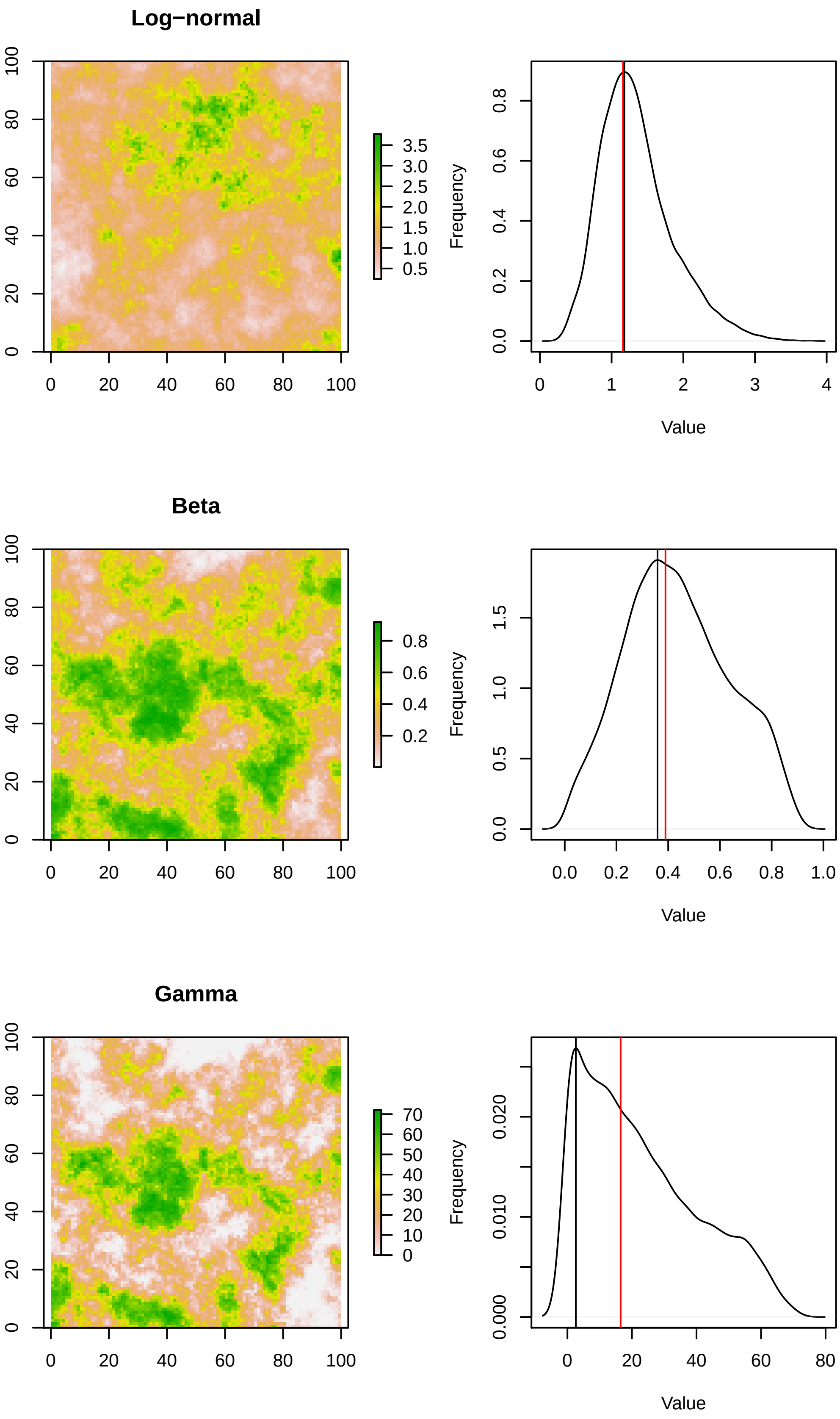
```
coef(ppm.best.mod)
```

```
##  (Intercept)            a            b            c       I(a^2)       I(b^2)
## -274.6297559   16.0387250    1.5901291    1.8035583   -0.2420749   -0.1146402
##       I(c^2)
##   -0.5728335
```

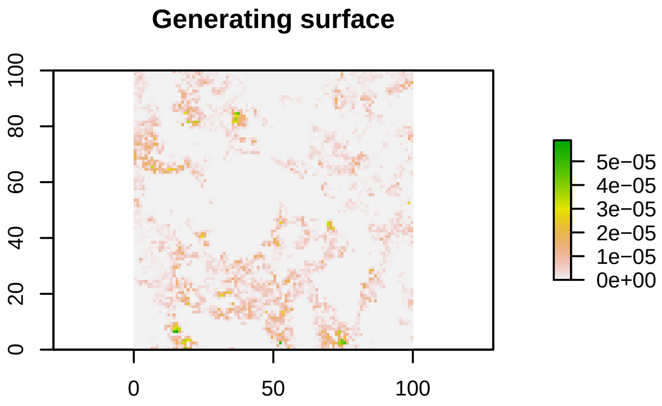
Now by examining all the models, the distance to the true centroid tends to be much larger among the models which fitted at least one positive coefficient, although some of the models with negative coefficients only also struggled a fair bit (outliers):

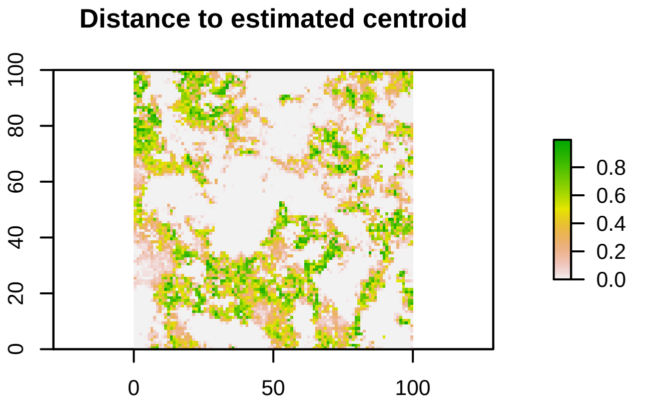


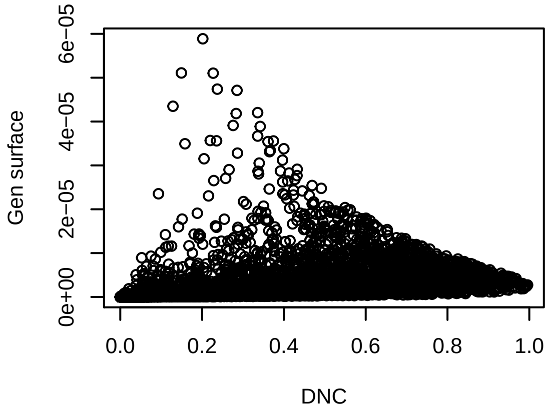
Ellipses For the case of the ellipses there is no alternative than having a close look at the distribution of the specific variables used: *log-normal*, *beta* and *gamma*:



The black vertical lines represent the true centroid values to be retrieved, and the red vertical lines are the retrieved values. The distance between lines is what drove the poorer performance of this model in comparison with the rest. And as can be seen in the plot of “distance vs correlation”, greater distance from the true centroids results in estimating a surface which resembles less the surface used to generate the data, especially with ellipses.







Qualitatively the ellipses generate a similar spatial pattern, but estimated distances do not generally correspond with probability of occurrence