regions

Katrina E Jones

2018-09-20

About Regions

The Regions package encompasses a suite of functions which find regions in serially-homologous structures using segmented regression, based on the method of Head & Polly (2014).

Detecting regions in serially-homologous structures has traditionally been approached with clustering methods. The segmented regression approach implemented here better adresses the integration of serially-homologous biological structures (such as vertebrae), modelling them as morphological gradients along a continuosly varying series. This also constrains `regions` to group only units which are adjacent with one another, reflecting developmental processes i.e., vertebral patterning via colinear *Hox* expression.

Maximum likelihood is used to select regionalization models, up to a maximum of six regions, and produce a regionalization score reflecting the degree of regionalization of each dataset. The best-fit region model is also output.

The main steps are as follows: A. Data Ordination B. Calculate segmented regression models (slowest step) C. Reduce data D. Select best models for each hypotheses (number of regions) E. Compare hypotheses using AICc

Getting started

Preliminaries

Regions code can be loaded using intall packages>from archive, then selecting the bundled file. Here we will use example morphological data from an Alligator to demonstrate use of the package.

```
library(regions)
data("alligator")
```

A typical dataset will contain one positional variable, in this case vertebral number, and multiple dependent variables. These will usually be continuous variables describing the morphology of the various serially homologous vertebrae. In this case it is linear and angular measures taken on vertebrae. Any type of data may be appropriate, though be careful to chose the appropriate distance metric for your data types (see below).

Vertebra	CL	Chpost	Cwpost	Chant
3	22.38	15.15	18.7	14.21
4	22.1	15.82	18.23	15.4
5	22.29	16.39	18.66	16.32
6	21.23	16.93	18.5	16.11
7	21.13	16.63	19.37	16.48

8	20.57	17.07	19.61	16.18

Preparing the Data

A positional variable must be selected as the independent variable for the analysis.

```
Xvar<-alligator[,1]
nvert<-length(Xvar)
Xvar[1:5]
```

```
## [1] 3 4 5 6 7
```

If you have missing data you may wish to fill short strings by interpolating from neighboring elements, up to a maximum of two missing points. Long strings are left as NA.

```
data<-alligator[,2:ncol(alligator)] #rest are dependent variables
data<-Missingval(data)#fill missing data
```

Next, it is recommended that you scale the data prior to analysis, to examine patterns as opposed to magnitudes. This also enables bootstrap selection of PCOs (below).

```
data<-scale(data)
```

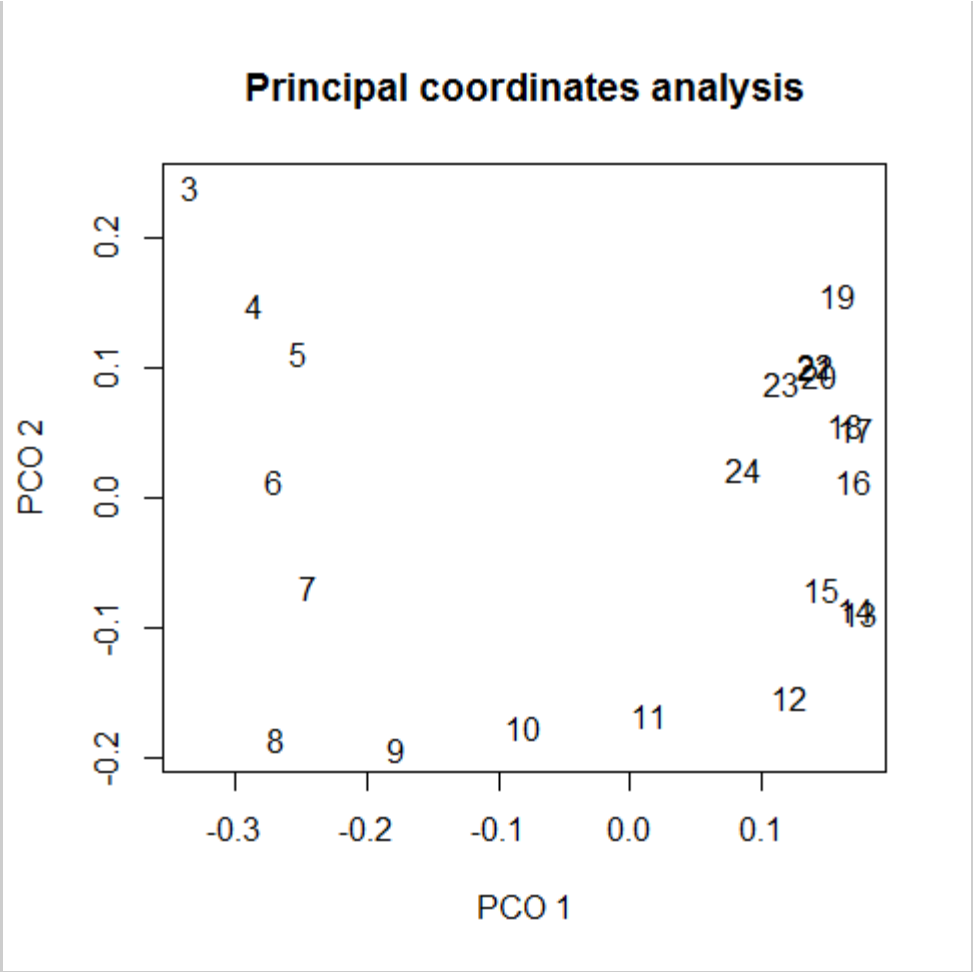
A. Data ordination

To incorporate a wide variety of data types, and potential missing data, you can use a distance-based data ordination, though the analysis should work equally well on PCA where data are appropriate. Principal coordinates analysis (PCO) is used to create axes which maximize the variation. The implementation employed in `svdPCO` uses a distance matrix generated by `cluster::daisy`. It differs from other implementations of PCO (e.g., based on `CMDscale`) as it uses a singular value decomposition (i.e., `svd`) instead of the more generalized `eigen`, thereby avoiding negative eigenvalues.

Three types of distance metric can be used: euclidean, manhatten, or gower. Euclidean should only be used where all variables are similar (e.g., linear measures on the same scale), and is most similar to a PCA. Gower is good for combining different types of continuous data (e.g., angles and linear). Missing data is allowed, as long as there is some overlap in represented variables. For more information see `?(daisy)`.

```
pco.gower<-svdPCO(data, "gower")#PCO using svd
PCOscores<-pco.gower$scores[,1:ncol(pco.gower$scores)]
```

```
axesplot(PCOscores, 1, 2, Xvar)
```



Variation along the column

B. Segmented Regression models

Now we can calculate segmented regression models for all possible combinations of regions on each PCO separately. This is the first step in calculating regionalization and may be very slow depending on the number of variables and regions being analyzed. You may wish to reduce the data first to speed this step up.

You can calculate the segmented regression models using `compileregions`

```
noregions<-3 #Set the maximum number of regions which will be calculated
regiondata<-compileregions(Xvar,PCOscores[,1:10],noregions)
pander::pandoc.table(regiondata[1:5,1:5])
```

```
##
## ----------------------------------------------------------------
##  regions   breakpoint1   breakpoint2   breakpoint3   breakpoint4
## --------- ------------- ------------- ------------- -------------
##     1           0             0             0             0
##
##     2           4             0             0             0
##
##     2           5             0             0             0
##
##     2           6             0             0             0
```

```
##
##     2           7             0             0             0
## ----------------------------------------------------------------
```

C. Data reduction

Most biological data contain significant noise due to measurement error or taphonomy. This noise tends to concentrate on lower PCOs. The AIC selection procedure implemented here assumes that all input variables (PCOs) contain information, and thus penalizes them equally. Therefore, using all the PCOs in the analysis can artificially reduce regionalization score, so it is good to minimize the number of input variables. `Regions` implements multiple options for reducing your dataset.

1. Select a number of PCOs - five PCOs is a common cutoff, see Head and Polly (2014).

```
nopcos<-5
nopcos
```

```
## [1] 5
```

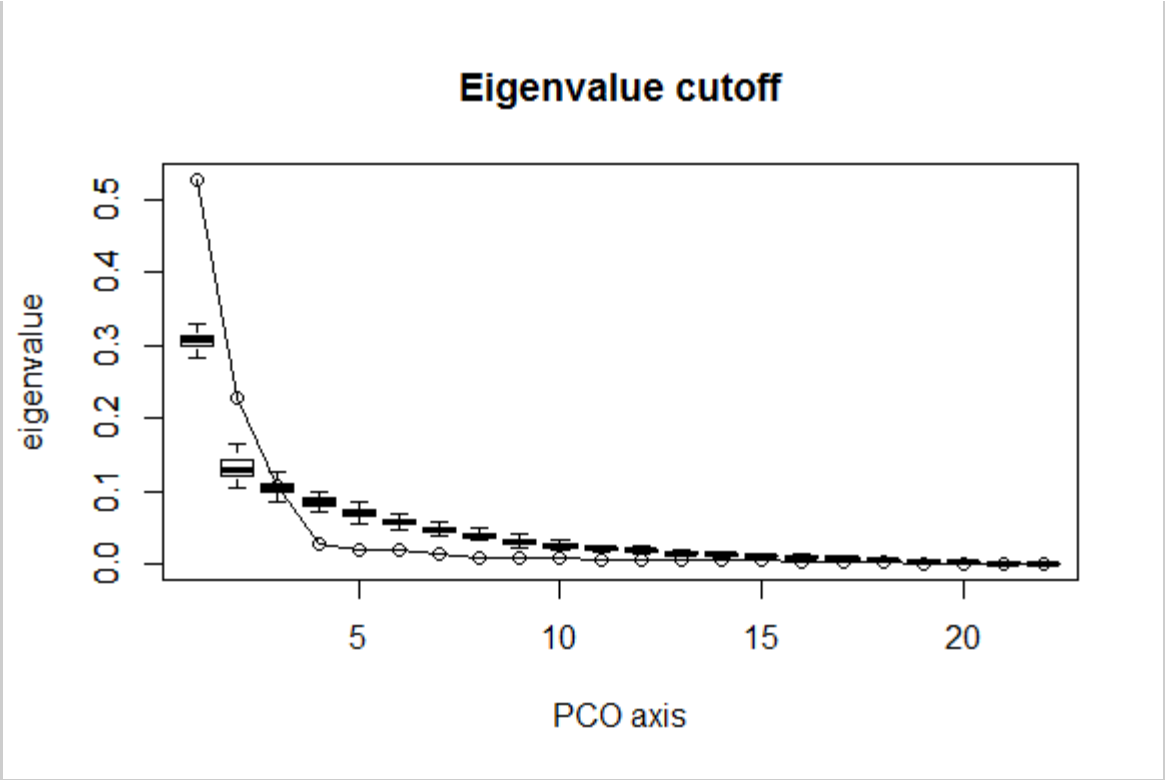
2. Bootstrapping - bootstrapping algorhithm which can help you select the number of PCO's containing significant variation compared to random noise. The is particularly important as noise may be amplified in low variance measures by the scaling step of the PCO.

`PCOcutoff` randomizes the variables for each unit and estimates the mean eigenvalue distribution of this random data. One can select the PCO's to include in the analysis as those with an eigenvalue (% variance explained) greater than the mean eigenvalue of the random data for that PCO. Note that variables must be scaled prior to analysis for this bootstrapping approach to be meaningful.

```
#bootstrapped with 100 itterations
pco.boot<-PCOcutoff(data,100, "gower")
nopcos<-pco.boot$sigpco#Select significant axes
nopcos
```

```
## [1] 3
```

```
#Plot the eigenvalues
eigenplot(pco.boot$eigen.true, pco.boot$eigen.boot)
```



3. Variance Cutoff - Select only the PCOs representing more than five percent of variance.

```
nopcos<-length(which(pco.gower$eigen.val/sum(pco.gower$eigen.val)>0.05))#more than 5% of variance
nopcos
```

```
## [1] 3
```

4. Maximising regionscores - select the number of PCO's which gives the maximum possible region score using `PCOmax`.

If you are concerned that excluding PCOs may be masking regionalization signal, you can optimize the PCO selection to find the maximum number of regions.

```
nopcos<-PCOmax(regiondata, noregions, nvert)$pco.max
nopcos
```

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

D. Select best models

Now we can pick which segmented regression model fits the data best for each hypothesis (1 region, 2 region etc) by simply minimizing the residual sums of squares.

```
models<-modelselect(regiondata,noregions,nopcos)
pander::pandoc.table(models)
```

```
##
## ------------------------------------------------------------------------
##      regions   breakpoint1   breakpoint2   breakpoint3   breakpoint4
## -------- --------- ------------- ------------- ------------- -------------
##  **1**       1            0             0             0             0
##
##  **9**       2           11             0             0             0
##
##  **72**      3            7            12             0             0
## ------------------------------------------------------------------------
##
## Table: Table continues below
##
##
## ---------------------------------
##      breakpoint5    sumRSS
## -------- ------------- ----------
##  **1**          0          0.2213
##
##  **9**          0          0.0271
##
##  **72**         0         0.003684
## ---------------------------------
```

E. Compare hypotheses using AICc

Now we have 4 regionalization hypotheses which we can test using AICc.

The adjusted log likelihood is calculated as

$$n \quad log(RSS/n) + adj$$

where n is $Novertebrae \quad NoPCOscores$ and adj is the AIC adjustment

$$adj = 2p + (2p(p+1))/(n-p-1)$$

where p is the numer of parameters in the mode being fitted

$$p = NoPCOs \quad 2 \quad Noregions + Noregions - 1$$

As each region in each model has a slope and an intercept.

This provides a relative measure of goodness-of-fit for each hypothesis. We can convert that into a continuous regionalization score by calculating Aikake weights from the log likelihood of each hypothesis:

$$Akw = LogL(n)/TotalL$$

Regionalization score is the weighted average of region number,scaled by Akaike weight.

```
support<-model_support(models,nvert, nopcos)
pander::pandoc.table(support$Model_support)
```

```
##
## ------------------------------------------------------------------------
##      regions   breakpoint1   breakpoint2   breakpoint3   breakpoint4
```

```
## -------- --------- ------------- ------------- ------------- -------------
##  **72**       3            7             12              0             0
##
##  **9**        2           11              0              0             0
##
##  **1**        1            0              0              0             0
## ------------------------------------------------------------------------
##
## Table: Table continues below
##
##
## ---------------------------------------------------------------------------
##       breakpoint5     sumRSS      AICc     deltaAIC    model_lik    Ak_weight
## -------- ------------- ---------- -------- ---------- ----------- -----------
##  **72**         0         0.003684   -164.2       0            1            1
##
##  **9**          0          0.0271    -133.6     30.57       2.3e-07      2.3e-07
##
##  **1**          0          0.2213    -96.55     67.66      2.033e-15    2.033e-15
## ---------------------------------------------------------------------------
```

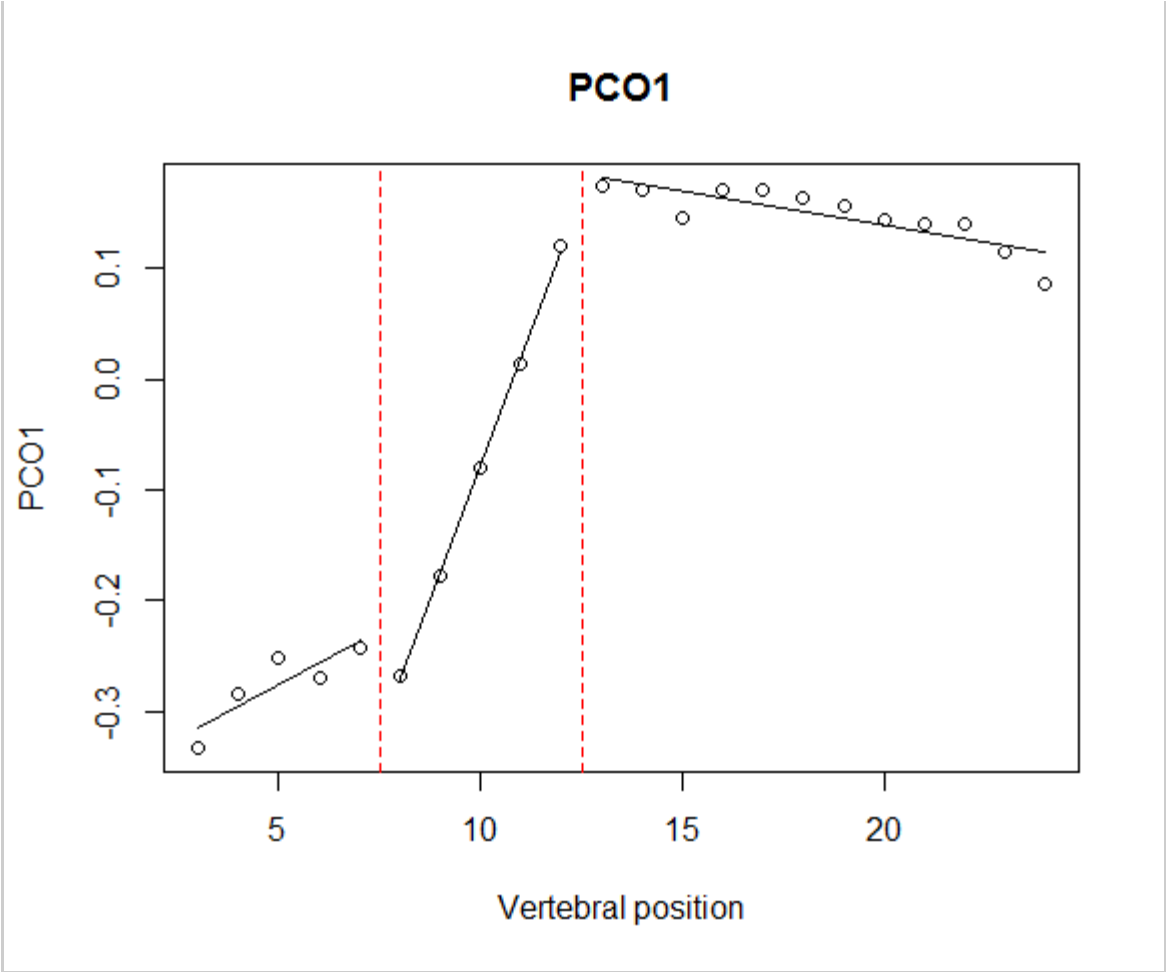
To check the fit of the model, you can calculate the multivariate r-squared using `multvarrsq`

```
rsq<-multvarrsq(Xvar,as.matrix(PCOscores[,1:nopcos]), support$Model_support)
```

Plotting regionalization models

Examine the fit of your model using `plotsegreg`

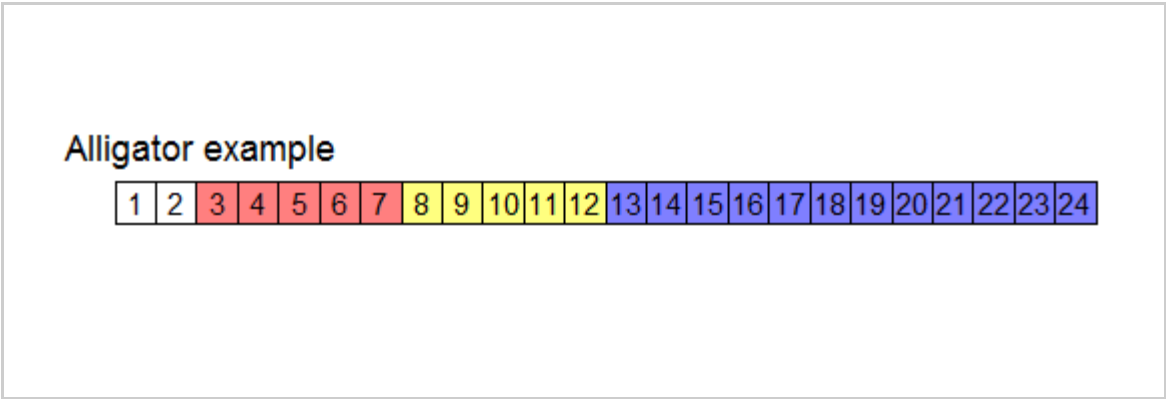
```
plotsegreg(Xvar,pcono=1, data=pco.gower$scores, modelsupport=support$Model_support)
```



Segmented regression model

Plot the region breaks using `regionmodel`

```
plot<-regionmodel(name="Alligator example", Xvar=Xvar, regiondata=support$Model_support)
  print(plot)
```



Best regionalization model