Getting started with `comclim`

Benjamin Blonder

May 30, 2018

1 Introduction

This document provides a narrated example of how to use the `comclim` package. I show how to merge climate niche data and species composition data into the format required by the framework, then, how to run analyses and interpret results. For demonstration purposes, I use simulated data for species' composition and climate niches.

2 Preparing input

2.1 Climate niche data

First, I prepare a dataframe of climate niches for all species in the analysis at time t_1. To do this, I begin by loading in the community climate package.

```
> library(comclim)
```

Next, I generate an object called `climateniches` which describes three-dimensional climate niches of 100 species, approximated by 50 observations each. In a real-world analysis, you would instead fill this dataframe with realized climate niches for each species based on experiments or from observational data. For example, this could be done using georeferenced observations transformed into climate space using the `extract` command in `library(raster)`. The important thing is that the dataframe has one column named `taxon` that can be used for matching species lists. The other columns, with arbitrary names, are assumed to be the climate data.

```
> num_climateaxes = 3
> num_regionalpool = 100
> num_occurrences = 50
> climateniches <- NULL
> for (i in 1:num_regionalpool)
+ {
+    randdata = NULL
+         for (j in 1:num_climateaxes)
+         {
+                 meanpos = runif(num_climateaxes,min=2,max=4)
+                 tcol = rnorm(num_occurrences, mean=meanpos[j]
+             + runif(n=1,min=-2,max=2), sd=runif(1, 0.2,0.4))
+                 randdata <- cbind(randdata, tcol)
+         }
+
+         randdata <- as.data.frame(randdata)
+
```

```
+          names(randdata) <- paste("ClimateAxis", 1:num_climateaxes, sep='')
+          randdata$taxon = paste("Species", i, collapse='')
+
+          climateniches <- rbind(climateniches, randdata)
+ }
> climateniches$taxon <- factor(climateniches$taxon)
```

In a real analysis it is important that the climate axes are on comparable scales, because the community climate statistics are calculated using Euclidean distances. One way to ensure this requirement is met is to rescale all climate variables by z-transformation:

```
> climateniches[,1:num_climateaxes] <-
+     scale(climateniches[,1:num_climateaxes], center=TRUE, scale=TRUE)
```

The final climate niche dataframe is now ready.

```
> print(str(climateniches))

'data.frame':       5000 obs. of  4 variables:
 $ ClimateAxis1: num  -1.228 -1.113 -1.206 -1.645 -0.944 ...
 $ ClimateAxis2: num  0.933 1.317 0.931 1.341 0.903 ...
 $ ClimateAxis3: num  -1.285 -0.767 -1.826 -1.087 -0.657 ...
 $ taxon       : Factor w/ 100 levels "Species 1","Species 10",..: 1 1 1 1 1 1 1 1 1 1 ...
NULL
```

2.2 Local community

The next step is to define the local community to be analyzed at time t_1. Here I take a subset of five species, intentionally choosing those with centroid positions that are closest to the value 1 along each axis. In a real study, the `localcommunity` object would simply be a list of species names that is a subset of those in `levels(climateniches$taxon)`.

```
> num_community = 5
> nichedist <- do.call("rbind",by(
+   climateniches[,1:num_climateaxes],
+   climateniches$taxon, function(x) {
+       cm <- colMeans(x)
+       cm <- cm- rep(1, num_climateaxes);
+       return(data.frame(pos=sqrt(sum(cm^2))))
+     }
+   ))
> # select for species on the lower edge of the climate space
> whichsp <- order(nichedist,decreasing=FALSE)[1:num_community]
> localcommunity <- row.names(nichedist)[whichsp]
> print(localcommunity)

[1] "Species 49" "Species 56" "Species 63" "Species 25" "Species 54"
```

2.3 Regional pool

Like the local community, the regional pool list (`regionalpool`) is simply a vector of species names that exist at time t_1. In a real analysis, you would generate this vector based on some prior knowledge of your system. In this case, we assume that it is equivalent to all the species for which we have already generated climate niches:

```
> regionalpool <- as.character(levels(climateniches$taxon))
> print(regionalpool)

  [1] "Species 1"   "Species 10"  "Species 100" "Species 11"  "Species 12"
  [6] "Species 13"  "Species 14"  "Species 15"  "Species 16"  "Species 17"
 [11] "Species 18"  "Species 19"  "Species 2"   "Species 20"  "Species 21"
 [16] "Species 22"  "Species 23"  "Species 24"  "Species 25"  "Species 26"
 [21] "Species 27"  "Species 28"  "Species 29"  "Species 3"   "Species 30"
 [26] "Species 31"  "Species 32"  "Species 33"  "Species 34"  "Species 35"
 [31] "Species 36"  "Species 37"  "Species 38"  "Species 39"  "Species 4"
 [36] "Species 40"  "Species 41"  "Species 42"  "Species 43"  "Species 44"
 [41] "Species 45"  "Species 46"  "Species 47"  "Species 48"  "Species 49"
 [46] "Species 5"   "Species 50"  "Species 51"  "Species 52"  "Species 53"
 [51] "Species 54"  "Species 55"  "Species 56"  "Species 57"  "Species 58"
 [56] "Species 59"  "Species 6"   "Species 60"  "Species 61"  "Species 62"
 [61] "Species 63"  "Species 64"  "Species 65"  "Species 66"  "Species 67"
 [66] "Species 68"  "Species 69"  "Species 7"   "Species 70"  "Species 71"
 [71] "Species 72"  "Species 73"  "Species 74"  "Species 75"  "Species 76"
 [76] "Species 77"  "Species 78"  "Species 79"  "Species 8"   "Species 80"
 [81] "Species 81"  "Species 82"  "Species 83"  "Species 84"  "Species 85"
 [86] "Species 86"  "Species 87"  "Species 88"  "Species 89"  "Species 9"
 [91] "Species 90"  "Species 91"  "Species 92"  "Species 93"  "Species 94"
 [96] "Species 95"  "Species 96"  "Species 97"  "Species 98"  "Species 99"
```

2.4 Defining the observed climate

The next input is the observed climate at time t_2. Here I choose a vector with all axes set to -1, to simulate an 'extreme' climate. In a real study, you could obtain this vector from observations or gridded climate data.

```
> observedclimate <- rep(-1, num_climateaxes)
> names(observedclimate) <- paste("ClimateAxis", 1:num_climateaxes, sep='')
> print(observedclimate)

ClimateAxis1 ClimateAxis2 ClimateAxis3
          -1           -1           -1
```

2.5 Putting it all together

The final step is to merge all of these data into a `CommunityClimateInput` object. The package provides a helper function.

```
> cci <- inputcommunitydata(
+           localcommunity = localcommunity,
+               regionalpool = regionalpool,
+               climateniches = climateniches,
+               observedclimate = observedclimate)
> summary(cci)

Formal class 'CommunityClimateInput' [package "comclim"] with 5 slots
  ..@ species_list_tinf        : chr [1:5] "Species 49" "Species 56" "Species 63" "Species 25" ...
  ..@ regional_pool_tinf       : chr [1:100] "Species 1" "Species 10" "Species 100" "Species 11" ...
  ..@ regional_pool_weights_tinf: num(0)
  ..@ climate_niches_tinf       :'data.frame':        5000 obs. of  4 variables:
```

```
.. ..$ ClimateAxis1: num [1:5000] -1.228 -1.113 -1.206 -1.645 -0.944 ...
.. ..$ ClimateAxis2: num [1:5000] 0.933 1.317 0.931 1.341 0.903 ...
.. ..$ ClimateAxis3: num [1:5000] -1.285 -0.767 -1.826 -1.087 -0.657 ...
.. ..$ taxon        : Factor w/ 100 levels "Species 1","Species 10",..: 1 1 1 1 1 1 1 1 1 1 ...
..@ observed_climate_tobs    : Named num [1:3] -1 -1 -1
.. ..- attr(*, "names")= chr [1:3] "ClimateAxis1" "ClimateAxis2" "ClimateAxis3"
NULL
```

It is also possible to visualize the regional pool and local community in climate space.

```
> plot(cci,cex.community=0.75)
```
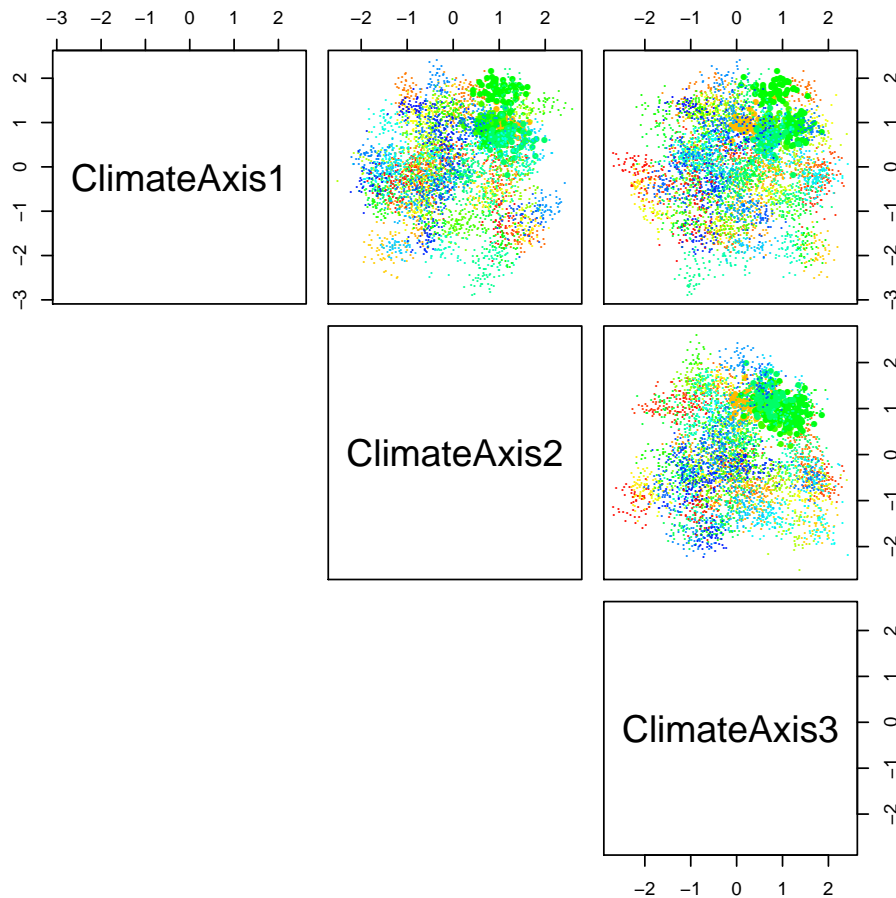

Above you see the species in the community plotted with large symbols, and all the species in the regional pool plotted with small symbols. By construction, the local community is at the 'edge' of the climate space encompassed by the regional pool.

3   Running a community climate analysis

Now that the data are in the correct format, running an analysis is simple. I only have to specify the climate axes to be used (in this case, all of them). In a real analysis you might choose only a subset of axes, or specify larger values for the number of replicates (e.g. `numreplicates = 1000` instead of `100`), etc.

```
> result_community <- communityclimate(cci,
+    climateaxes=c("ClimateAxis1","ClimateAxis2","ClimateAxis3"),
+    numreplicates=100, verbose=F)
```

4   Interpreting and visualizing the results

By construction, the example defined a community whose volume was smaller than the regional pool's, and whose inferred climate was much further away from the observed climate than the regional pool's. As a result, I should expect to find $\delta(t_1) < 0$ and $\lambda(t_1, t_2) > 0$. The actual results can be seen from the output object:

```
> summary(result_community)

*******************************************************************************
*** Community climate statistics - observed
*******************************************************************************
$meanNiches
  ClimateAxis1 ClimateAxis2 ClimateAxis3
1    0.8702600    0.8186069    0.8881399
2    0.8391253    0.8939015    1.3696051
3    0.5750814    1.2008927    0.7463516
4    0.9362184    1.0850574    0.3181616
5    1.7176459    1.0311290    0.8144263


$inferredClimate
ClimateAxis1 ClimateAxis2 ClimateAxis3
   0.8955121    1.0129265    0.8250454


$observedClimate
ClimateAxis1 ClimateAxis2 ClimateAxis3
          -1           -1           -1


$volumeMagnitude
[1] 0.5891773


$mismatchMagnitude
[1] 3.320317


$mismatchDirections
ClimateAxis1 ClimateAxis2 ClimateAxis3
    1.895512     2.012926     1.825045


*******************************************************************************
*** Community climate - deviations (n=100 nulls)
*******************************************************************************
$deviation_volumeMagnitude
      ses     pvalue
-2.532478  0.000000


$deviation_mismatchMagnitude
      ses    pvalue
1.919556 0.000000
```

```
$deviation_mismatchDirections
                  ses pvalue
ClimateAxis1 1.150018   0.04
ClimateAxis2 1.354962   0.14
ClimateAxis3 1.150189   0.10
```

Looking at the deviations, `result_community$deviation_volumeMagnitude` (i.e. $\delta(t_1)$) is significantly smaller than zero, as predicted. Similarly, `result_community$deviation_mismatchMagnitude` (i.e. $\lambda(t_1, t_2) > 0$) is significantly greater than zero, as expected.

To visualize these inferences, it is also possible to plot a community climate diagram.

```
> plot(result_community)
```


$\delta = -2.532, \ \lambda = 1.92$
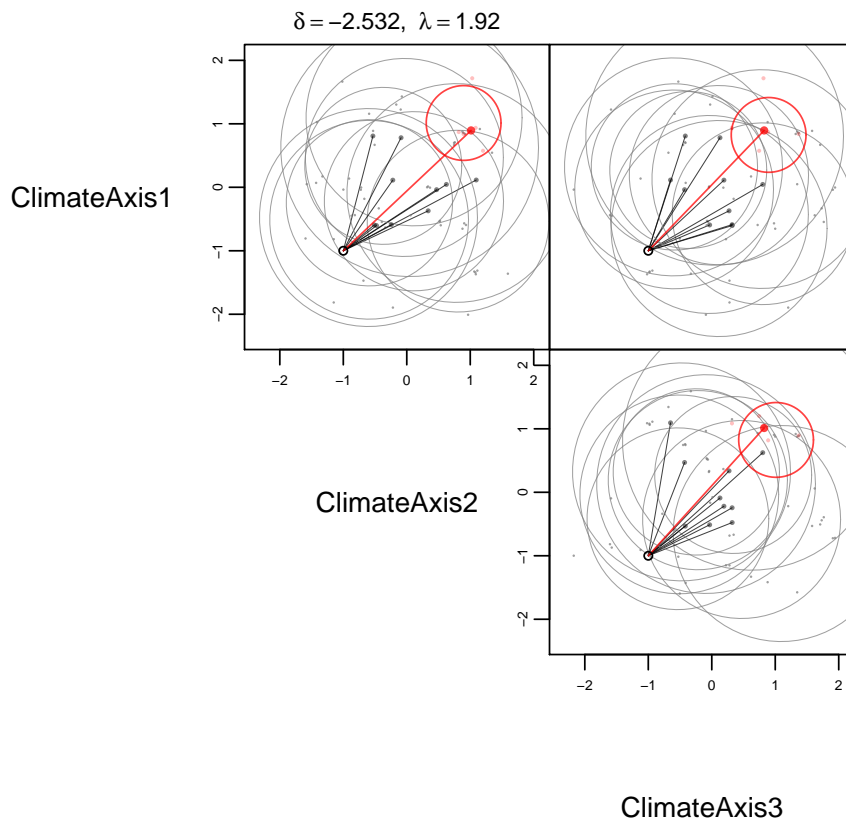
Here I can confirm that the community has smaller climate volume (red circle vs gray circles) and larger climate mismatch (red vector vs gray vectors) than expected. The community climate deviations are summarized as the figure's title.

It is also possible to directly plot the null and observed values for each community climate statisic, along with projections of the mismatch vector along each climate axis. The gray lines show the kernel-smoothed null distribution along with the 25% and 75% quantiles, and the red line shows the observed value.

```
> plot(result_community, deviations=TRUE)
```
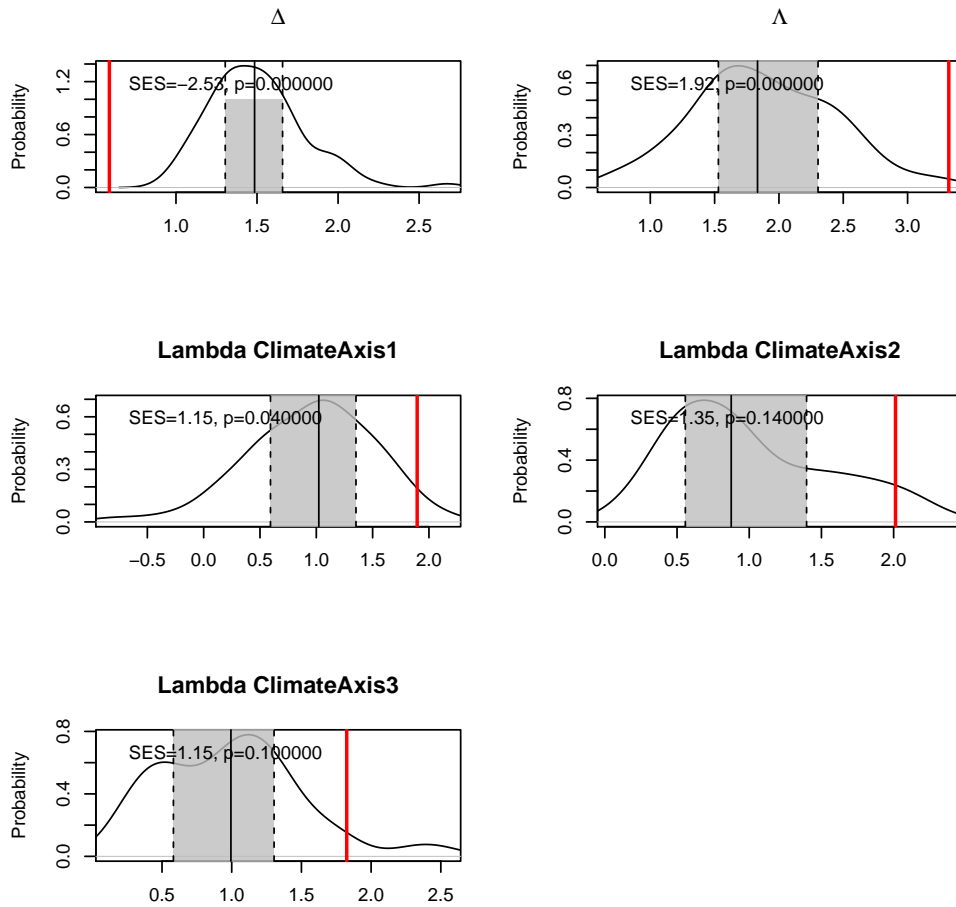

Based on all of these results, I would infer that the community is structured by environmental filtering and also by environmental disequilibrium.

5   Summary

In short, the necessary steps to using the `comclim` package on your own data are:

1. Obtain climate niche data for all species and transform it to standardized axes

2. Use `inputcommunitydata` to merge all input data

3. Run `communityclimate` to calculate community climate statistics

4. Analyze or plot the resulting deviations

That's all!