Tutorial vignette for package ‘specificity’

John L Darcy

28 January 2020

Introduction

In this example analysis, our goal is to analyze the extent to which microbes have specificity to different aspects of their environment. We will use the ‘endophyte’ data set (included with the specificity R package), which contains data from a survey of foliar endophytic fungi living within the leaves of native Hawaiian plants. In this context, a fungus with strong specificity would be one that occupies a narrow range of sampled environment. For example, a fungus that preferentially associates with a narrow clade of host plants (e.g. asteraceae), or prefers a narrow range of elevation (e.g. between 700 and 1200 m.a.s.l). The previous two examples were for unimodal specificity, but specificity as calculated with our software can be multimodal as well (e.g. phylogenetic specificity to both asteraceae and malvaceae). It is worth noting that a strength of this approach is that specificity is agnostic to the ideal habitat of a species: a species with strong specificity to high elevations may be just as specific to elevation as a species specific to low elevations. Indeed, a strength of this approach is that the ideal habitat does NOT need to be modeled in order to calculate specificity. Using default options, we calculate specificity as an index that ranges from -1 to 1, with -1 indicating perfect specificity, 0 indicating no difference from the null model, and 1 indicating perfect cosmopolitanism (`denom_type = "index"`).

Software Requirements

- `R` (run on version 3.6.0, but likely works fine on earlier versions)
- Other dependencies will be automatically installed with the package.

Installation

To install specificity, open up R and run:

```
  library(devtools)
  install_github("darcyj/specificity")
```

Load ‘endophyte’ example data set

The endophyte dataset consists of foliar endophytic fungi sampled from leaves of native Hawaiian plants across the Hawaiian archipelago. These data can be loaded into R with:

```
# load specificity R package
suppressMessages(library(specificity))
# load endophyte data set from specificity
data(endophyte)
# check what objects are within endophyte
names(endophyte)
```

```
## [1] "otutable"  "metadata"  "supertree"
```

```
# add those objects to R namespace so we aren't typing "endophyte$" every time
attach(endophyte)
```

As you can see, there are 3 objects inside ‘endophyte’.

- metadata: table of data where each row corresponds to a sample, and each column is a different metadata category (e.g. PlantGenus, Elevation, Lon=longitude, Lat=latitude)
- otutable: table of OTU (species) observation data, where each row is a sample, and each column is a different fungal OTU (really DADA2 ASVs, but those are a type of OTU).
- supertree: a phylogenetic tree of host plants.

Pre-processing of ‘endophyte’ data

Like most statistics, specificity is useless with a low sample size. In the case of specificity, this ‘sample size’ is the occupancy of a species, or how many samples that species was observed in. Here, we will create a new table containing only species that observed in at least 10 samples.

```
# apply occupancy threshold to remove low-occupancy species
otutable_ovr10 <- occ_threshold(otutable, threshold=10)
# how many species are we left with?
ncol(otutable_ovr10)
```

```
## [1] 416
```

```
# (there are MANY species in this dataset that are extremely rare)
```

Specificity analysis

Calculating specificity for various data types is fairly easy with this package. In this example, we will analyze specificity to elevation, rainfall, host plant phylogeny, and geographic distance. In the function `phy_or_env_spec()`, the `n_sim` argument determines the number of simulations (i.e. permutations) to do. Each analysis below should take 1 or 2 minutes to complete. When you run them, you will see status updates that are ommitted in this vignette for the sake of brevity. To make all of this run faster, we are only using 100 simulations (permutations). `p_method = "gamma_fit"` means we fit a gamma distribution to permuted specificity values in order to calculate P-values for each species. The default is `p_method = "raw"`, which requires a much higher `n_sim` to get reasonable P-values.

```
# specificity for elevation:
elev_spec <- phy_or_env_spec(otutable_ovr10, env=metadata$Elevation,
  n_sim=100, n_cores=3, p_method = "gamma_fit")
# specificity for rainfall
rain_spec <- phy_or_env_spec(otutable_ovr10, env=metadata$Rainfall,
  n_sim=100, n_cores=3, p_method = "gamma_fit")
# specificity for evapotranspiration
evap_spec <- phy_or_env_spec(otutable_ovr10, env=metadata$Evapotranspiration,
  n_sim=100, n_cores=3, p_method = "gamma_fit")
# specificity for host phylogeny
host_spec <- phy_or_env_spec(otutable_ovr10, hosts=metadata$PlantGenus,
    hosts_phylo=supertree, n_sim=100, n_cores=3, p_method = "gamma_fit")
```

For matrix inputs like geographic distance, we need to specify a theoretical maximum distance in the matrix. For many dissimilarity matrices, this theoretical maximum is 1, but for geographic distance we will use the maximum distance observed in the matrix. If you forget to do this, you’ll get an error message that says, “Values grater than matrix_tmax are present in env”. We also need to use the `distcalc` function to calculate geographic distances from our latitude and longitude data.

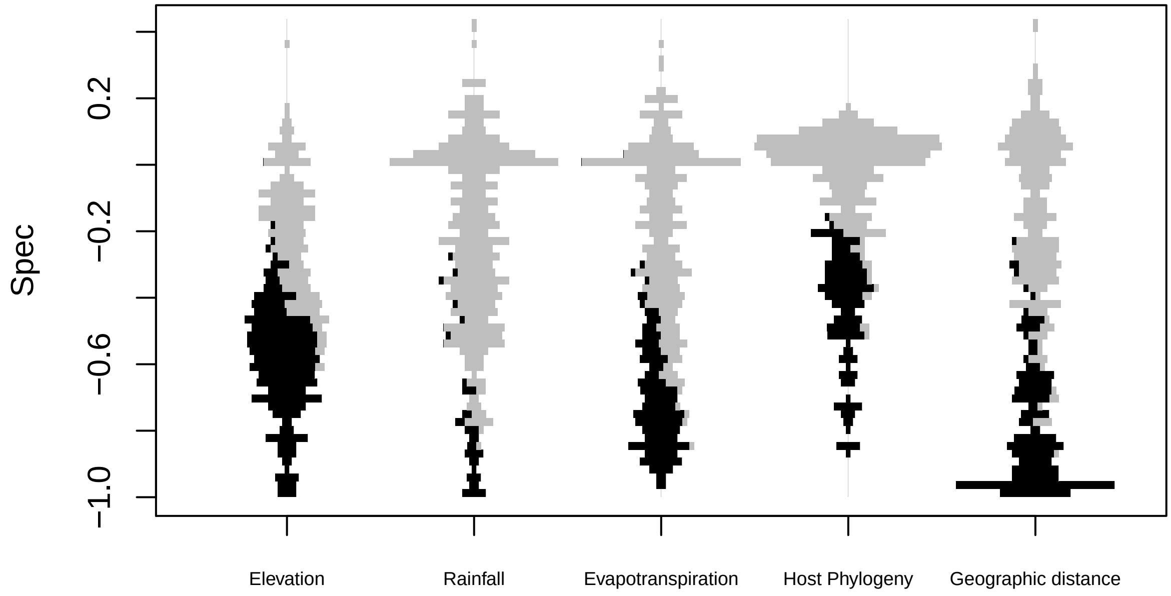
```
# calculate geographic distances
geo_distmat <- distcalc(lat=metadata$Lat, lng=metadata$Lon, sampIDs=metadata$SampleID)
```

```r
# specificity for geographic distance
geo_spec <- phy_or_env_spec(otutable_ovr10, env=geo_distmat, n_sim=100, n_cores=3,
  matrix_tmax=max(geo_distmat), p_method = "gamma_fit")
```

Visualization

Our built-in visualization function takes a list of results from `phy_or_env_spec()` as input, and produces a violin plot composed of bars colored by statistical significance. The names of items within `specs_list` will be the labels used in the plot.

```r
plot_specificities(
    specs_list = list(
        "Elevation" = elev_spec,
        "Rainfall" = rain_spec,
        "Evapotranspiration" = evap_spec,
        "Host Phylogeny" = host_spec,
        "Geographic distance" = geo_spec
    ),
    n_bins = 60
)
```


Spec
Elevation
Rainfall
Evapotranspiration
Host Phylogeny
Geographic distance