Mean Spectrograms

J.P. Meagher

23 October 2017

Spectrograms have been preprocessed in Matlab and the results ported into R. Before continuing, an exploratory analysis of the mean spectrograms to individual and species level will be performed.

Packages

```
library(tidyverse)
library(magrittr)
library(RColorBrewer)
```

Load Data

```
df <- readRDS('preprocessed_calls.RDS')
```

Initial analysis will be performed on the full spectrograms only. Once this has been done a similar analysis of the thresholded spectrograms will be relatively painless and simply a case of plugging in the thresholded spectrograms into the existing code framework.

```
df_full <- df %>% select(bat, species, sex, full)
df_thresh <- df %>% select(bat, species, sex, thresholded)
```

Unnest list-columns

The spectrograms are currently stored in a list-column in the data frame. In order to manipulate these data the must first be vectorised and then unnested so that each point on the spectrogram surface has its own column in the data frame.

```
df_full_un <- df_full %>%
  mutate(vec = map(full, function(x) c(x) %>% t %>% as_tibble)) %>%
  select(bat, species, sex, vec) %>%
  unnest(vec)

df_thresh_un <- df_thresh %>%
  mutate(vec = map(thresholded, function(x) c(x) %>% t %>% as_tibble)) %>%
  select(bat, species, sex, vec) %>%
  unnest(vec)
```

Mean spectrograms

With the unnested spectrogram, the mean for each point on the spectrogram for various subsets of the data can be calculated straightforwardly. These computations do become very time consuming when they involve

a large number of variables and groups.

```r
by_bat_full <- df_full_un %>%
  select(-species, -sex) %>%
  group_by(bat) %>%
  summarise_all(mean)

by_bat_thresh <- df_thresh_un %>%
  select(-species, -sex) %>%
  group_by(bat) %>%
  summarise_all(mean)

by_species_full <- df_full_un %>%
  select(-bat, -sex) %>%
  group_by(species) %>%
  summarise_all(mean)

by_species_thresh <- df_thresh_un %>%
  select(-bat, -sex) %>%
  group_by(species) %>%
  summarise_all(mean)
```

Doing these calculations in this way, using this code is prohibitively expensive computationally and if they need to be done again it would likely prove more time efficient to rewrite the code than to run it again.

These dataframes can be nested again and restructured to to represent spectrograms again

```r
res_bat_full <- by_bat_full  %>%
  nest(starts_with('V')) %>%
  mutate(full = map(data, function(x) unlist(x) %>% matrix(nrow = 104, ncol = 50))) %>%
  select(-data) %>%
  ungroup()

res_bat_thresh <- by_bat_thresh  %>%
  nest(starts_with('V')) %>%
  mutate(thresh = map(data, function(x) unlist(x) %>% matrix(nrow = 104, ncol = 50))) %>%
  select(-data) %>%
  ungroup()

res_species_full <- by_species_full  %>%
  nest(starts_with('V')) %>%
  mutate(full = map(data, function(x) unlist(x) %>% matrix(nrow = 104, ncol = 50))) %>%
  select(-data) %>%
  ungroup()

res_species_thresh <- by_species_thresh  %>%
  nest(starts_with('V')) %>%
  mutate(thresh = map(data, function(x) unlist(x) %>% matrix(nrow = 104, ncol = 50))) %>%
  select(-data) %>%
  ungroup()
```

These dataframes are time consuming to produce and so I am going to save them for future reference.

```r
by_bat <- left_join(res_bat_full, res_bat_thresh, key = 'bat')
by_species <- left_join(res_species_full, res_species_thresh, key = 'species')
```

```r
saveRDS(by_bat, 'by_bat.RDS')
saveRDS(by_species, 'by_species.RDS')
```