

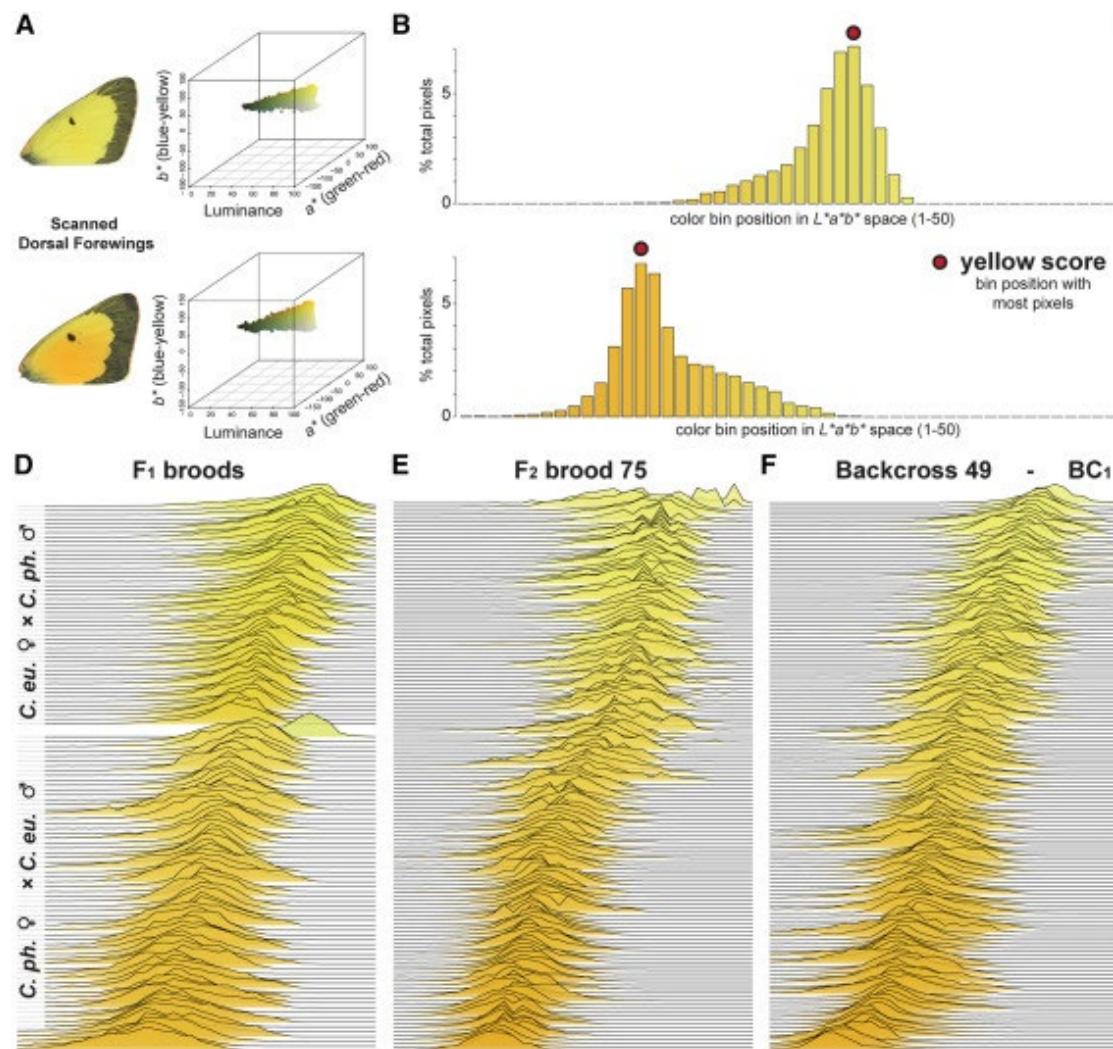
# BIO 724D Final Project Proposal

Leo Tomás Camino

My PhD research will focus on butterfly structural color. Measuring color is inherently complex, especially when it is structural, because it depends on both light and viewing angles. While color is often quantified through calibrated spectrophotometry, I would like to explore a method I have not used before: photographic color metrics. The paper "[Genetics of yellow-orange color variation in a pair of sympatric sulphur butterflies](#)" by Hanly *et al.* (2023) offers an ideal framework to experiment with this approach. In that study, the authors used R to decompose calibrated photographs into different color spaces and to identify variation in UV+ and UV-reflectance through the distribution of yellow to orange tones in butterfly wings.

I propose to recreate [Figure 1](#) from that paper. The figure presents:

1. The composition of color space for representative yellow and orange wings (Fig 1.A).
2. Histograms of the yellow color space from bins 1–50 (Fig 1.B).
3. Joy plots recreating the natural color gradient produced by different individuals (Fig1. D - F).



## Data availability, files and procedure

The data is public on [Dryad](#). I will use the .tiff photographs provided from the following files:

- 00-79.zip (Fig 1. D)
- 00-75.zip (Fig 1. E)
- 00-49.zip (Fig 1. F)

I will begin recreating the Joy Plots on Fig 1. D-F, as this will let me figure out which individuals are the most extreme (yellow and orange). Once I find out these two individuals, I will continue decomposing the specific channels of the most yellow and most orange butterflies to recreate Fig 1. A. & B.

### Part 1: Recreate Figures 1D–F (Joy Plots)

Geoms: `ggridges`, [geom\\_ridgeline](#), [geom\\_density\\_ridges](#)

1. Download .tiff files and use colordistance package to load the RGB values. I anticipate each figure will have a pixelX, pixelY, RGB array.
2. Using the colordistance package I will extract the CIELab color spaces. This will move the figure from the RGB array to the L\*(black or white), a\*(green - red), b\*(blue - yellow) array instead.
3. I will have to wrangle the data to isolate the b channel. This will require a for loop to repeat the process for all the figures.
4. I will generate 50-bin histogram of b values for each image to identify the yellow score (bin with the highest frequency). Instead of counts I will use the percentage of pixels falling on that specific bin.
5. Later, I will use ggplot2 to recreate the joy plots, applying an orange-yellow gradient.
6. Finally, I will reorder plots by yellow score rather than filename.

### Part 2: Recreate Figures 1A-B (Color Space and Histogram)

Geoms: [colordistance::plotPixels \(3D CIELab figures\)](#) & [gghist](#)

1. For these figures I will identify the most extreme individuals (most yellow and most orange) from the previous step. I could do this through the max () and min () function to get the individuals with the biggest and tiniest yellow score.
2. After this I will plot their 3D CIELab\* color space distributions using color distance, producing Figure 1. A.
3. For Figure 1. B, I will use the b values only to plot the percentage of pixels across 50 bins.
4. Mark the yellow score (tallest bin) with a red dot using ggplot2.

### Challenges that might be important to consider

- **Multiple high resolution .tiff images:** The good news is I have enough disk space and processing power to work with this. I hope the figures are calibrated, otherwise I might have to run a calibration on Adobe Lightroom.
- **Wrangling pixel data:** Extracting the b channel and generating histograms for 50 bins across many images is repetitive and could be tricky. I will need loops for this considering the figures are individual files that might be transferred later to a data frame on R.