**Morphometric Analysis of Sexual Dimorphism in Fly Wings**

This repository contains the code and scripts used for a comprehensive project investigating sexual dimorphism in the wing venation patterns of several fly species. The pipeline begins with raw image processing and ends with advanced multivariate statistical analysis and visualization.

**Project Workflow**

The analysis is conducted through a series of sequential steps, with each script performing a specific task. The general workflow is as follows:

1. **Image Segmentation (batch\_segment.py):** The PA2r wing cell is isolated from pre-cleaned wing images.
2. **Feature Extraction (efd\_final.r):** The shape of the segmented cell is converted into raw quantitative data (harmonic coefficients) using Elliptic Fourier Transform (EFT).
3. **Normalization (normalize.py):** The raw harmonic coefficients are processed to remove size as a variable, making the data purely shape-focused.
4. **Data Analysis & Visualization:** A series of scripts are used to explore the data, test hypotheses, and visualize the results.
   * **Exploratory Visualization (3D\_PCA.py, 3D\_LDA\_species.py, LDA\_gender.py):** PCA and LDA are used to visualize the primary patterns of variation and group separation.
   * **Quantifying Variation (manova\_sscp\_pca\_test.py):** MANOVA and SSCP are used to determine the percentage of shape variation explained by species, sex, and their interaction.
   * **Hypothesis Testing (hottelling\_test.r):** Hotelling's T-squared test is used to formally test for significant shape differences between sexes.
   * **Results Visualization (contour\_check.py):** The mean shapes for males and females are reconstructed to visually represent the findings.

**Script Descriptions**

**1. Image Segmentation**

**batch\_segment.py**

* **Purpose:** Automates the segmentation of the PA2r wing cell from a folder of pre-cleaned and cropped images.
* **Method:** Uses Meta's **Segment Anything Model (SAM)** with a pre-defined set of positive and negative point prompts.
* **Inputs:**
  + An INPUT\_FOLDER containing the images.
  + A SAM model checkpoint file (.pth).
  + Relative coordinates for the prompt points.
* **Output:** Saves binary (black and white) masks of the segmented PA2r cell into an OUTPUT\_FOLDER.

**2. Feature Extraction**

**efd\_final.r**

* **Purpose:** Converts the binary masks generated by batch\_segment.py into raw quantitative shape descriptors.
* **Method:** Applies **Elliptic Fourier Transform (EFT)** to the contours of the masks to generate harmonic coefficients.
* **Inputs:**
  + The image\_folder containing subfolders of species, which in turn contain the SAM output masks.
* **Output:** A single CSV file containing the raw, unnormalized harmonic coefficients for every sample, along with its species and gender metadata.

**3. Normalization**

**normalize.py**

* **Purpose:** Normalizes the raw EFD coefficients to make them invariant to size, ensuring the analysis focuses only on shape.
* **Method:** Calculates the semi-major axis (p) from the first harmonic for each sample and divides all coefficients for that sample by its p value.
* **Input:** The raw EFD coefficients CSV file generated by efd\_final.r.
* **Output:** A new CSV file (normalized\_efd\_coefficients\_10h.csv) containing the size-normalized coefficients. This file is the primary input for all subsequent analysis scripts.

**4. Data Analysis and Visualization**

**3D\_PCA.py**

* **Purpose:** Performs an initial exploratory analysis to visualize the main patterns of variation in the dataset.
* **Method:** Conducts a **Principal Component Analysis (PCA)** on the scaled, normalized harmonic coefficients.
* **Output:** An interactive 3D scatter plot (interactive\_pca\_plot.html) where samples are colored by species and marked with different symbols for gender.

**3D\_LDA\_species.py**

* **Purpose:** Assesses how well the shape data can distinguish between the different fly species.
* **Method:** Performs a **Linear Discriminant Analysis (LDA)** using species as the target classes.
* **Output:** An interactive 3D scatter plot (interactive\_lda\_plot\_species\_only.html) that visualizes the separation of species along the discriminant axes.

**LDA\_gender.py**

* **Purpose:** Visualizes the separation between genders along the single axis that best distinguishes them.
* **Method:** Performs an LDA with gender as the target and creates a Kernel Density Estimate (KDE) plot showing the distribution of each species along this axis.
* **Output:** A 2D density plot (species\_density\_on\_lda\_colored.png) that helps visualize the degree of overlap between sexes for each species.

**manova\_sscp\_pca\_test.py**

* **Purpose:** Formally quantifies and partitions the sources of shape variation.
* **Method:** Uses **MANOVA (Multivariate Analysis of Variance)** and decomposes the **Sum of Squares and Cross-Products (SSCP)** to calculate the percentage of variance explained by species, gender, and the species:gender interaction.
* **Output:** A console printout of a table summarizing the variance contributions and a bar plot (sscp\_expanded\_comparison.png) visualizing these results.

**hottelling\_test.r**

* **Purpose:** Performs the primary statistical test to determine if there is a significant difference in wing shape between males and females for each species.
* **Method:** Conducts a multivariate **Hotelling's T-squared test** on the principal component scores of the shape data.
* **Output:** A results table printed to the console, showing the test statistics, p-value, and **Mahalanobis distance** for each species.

**contour\_check.py**

* **Purpose:** Creates a visual representation of the average shape differences between sexes for each species.
* **Method:** Reconstructs the wing cell contours from the mean EFD coefficients for males and females of each species.
* **Output:** A multi-panel plot (male\_vs\_female\_wing\_contours\_by\_species.png) showing the mean female contour (solid line) overlaid on the mean male contour (dotted line).

**How to Run the Pipeline**

1. **Setup:**
   * Install the required R and Python libraries (e.g., Momocs, tidyverse, pandas, scikit-learn, plotly, statsmodels).
   * Organize your cleaned images into species-specific folders.
   * Download the SAM model checkpoint (sam\_vit\_b\_01ec64.pth).
2. **Run Segmentation:**
   * Configure the paths and prompt points in batch\_segment.py.
   * Execute the script for each species folder: python batch\_segment.py.
3. **Extract Features:**
   * Configure the main image folder and output file path in efd\_final.r.
   * Run the script in R to generate the raw efd\_coefficients.csv file.
4. **Normalize Features:**
   * Configure the input and output file paths in normalize.py.
   * Run the script: python normalize.py. This will create the final normalized\_efd\_coefficients\_10h.csv.
5. **Analyze and Visualize:**
   * Ensure the path to the **normalized** CSV file is correct in all subsequent Python and R analysis scripts.
   * Run the analysis scripts (.py and .r files) in any order to generate the plots, tables, and statistical results.