**Data Acquisition Software and Code:**

1. **Object detection model YOLO v4**: Used for locating and cropping specimens from the original image.
   * GitHub Repository: [bubbliiiing/yolov4-pytorch](https://github.com/bubbliiiing/yolov4-pytorch)
2. **opencv-python package** v4.6.0.66: Utilized for finding bounding boxes, perimeters, and areas of wings in images.
3. **WingImageProcessor**: A Matlab tool used for wing image analysis.
   * [Wing Image Processor](https://biomech.web.unc.edu/wing-image-analysis/)  
     We isolated and implemented the Aspect Ratio and Second Moment of Area calculation in python

**Data Analysis Software and Code:**

1. **Deep Learning Environment:**
   * **Python** v3.6.10
   * **GPU Hardware:** Nvidia RTX 3090
   * **CUDA driver**:v10.2
   * **torch** v1.8.1+cu102
   * **torchvision** v0.9.1+cu102
2. **R Packages**:
   * **R** v4.3
   * **ape** v5.7.1: Functions used: **drop.tip**, **ace**
   * **BAMMtools** v2.1.9: Functions used: **plot.bammdata**, **getBestShiftConfiguration**, **addBAMMshifts**, **credibleShiftSet**, **plotRateThroughTime**
   * **phytools** v1.5.1: Functions used: **make.simmap**, **describe.simmap**, **fitDiversityModel**
   * **geiger** v2.0.9: Functions used: **dtt**, **fitContinuous**, **aicw**
   * **PhylogeneticEM** v1.6.0: Function used: **phyloEM**
   * **OUwie** v2.10: Used for statistical modeling.
   * **boot** v1.3-28.1: Utilized for bootstrapping to estimate and compare means and variance in wing color patterning.
   * **hypervolume** v3.0: Used for calculating the trait hypervolumes, their set operations, and other related measures.
   * **TPD** v1.1: Used for calculating the trait probability density (TPD) indices.
   * **vegan** v2.6-4: Employed for the Permutational Multivariate Analysis of Variance (PERMANOVA) using the **adonis** function.
3. **DFC-VSC (a customized variational autoencoder)**: Employed for self-supervised feature extraction and image generation.
4. **ResNet50 model**: Used as a supervisor in DFC-VSC for deep-feature-consistency.
5. **Subfamily Classifier**: A multilayer perceptron with one hidden layer of 1024 neurons for finding key feature dimensions.
6. **BIRCH clustering**: Applied to the RGB values of wings and implemented in the python package scikit-learn v1.0.2.
7. **scikit-bio package** v0.5.6: Utilized to build a similarity tree using the neighbor-joining method.
8. **PCA (Principal Component Analysis)**: Employed to study morphological disparity. Implemented in scikit-learn v1.0.2.
9. **Back-propagation algorithm**: A standard algorithm for training neural networks. Implemented in the pytorch v1.8.1.