# Brief Example Code Outline

*functions are in bold*

Format the song data as described in *Data Structure Setup* section. Make sure to edit **rootdirlist** and **datadirlist** with your directory (“ Y: “,” C: ”,etc.) of choice.

* **rootdirlist** and **datadirlist** -- need to be updated for declaring directories throughout the code

The example*, bird.bmk*, is formatted with the minimal structural requirements, and is ready to be used by the package. The wav files are already prepared and there is another larger song file, birdlarge.bmk, ready as well.

## Hand Labeling

Use **sb3src\_bmklbl** to create a label file and label the song syllables. Use **movelbls** or **newcat** in order to make adjustments to the label file without the viewer. Once you are confident in the self labeling, use **calctemplates** in order to generate templates which will be stored in the label file. Use **listlabels** in order to view category string names and distribution.

* **sb3src\_bmklbl** -- for labeling song files
* **movelbls** or **newcat** --adjust label file; move clips or create new categories
* **calctemplates** – calculate templates for given categories
* **listlabels** – lists the string label names and distribution

## Calculate Features

Perform **calcspecs** on the bird.bmk in order to obtain spectrograms. Use **calcftrslice** to generate bin feature calculations from spectrograms based on the windows of FFT.

* **calcspecs**--generate the spectrograms and spectrogram directory
* **calcftrslice**--generate the feature calculations and feature directory

## Annotation

Perform **calcftrclips** after generating the feature directory from **calcftrslice** in order to obtain averaged feature values representing the spectrograms/clips. Calculate the metric file used in the clustering process with **calcmetric**. Use **noisescreen** in order to filter the song file and screen out artifacts associated with the avian song and recording process. This creates a label file associated with the process. **Noisescreen** should be performed on interested song files, labeled set and unlabeled set. Next, perform **calcdistftr** in order to obtain similarity matrix. We associate this file with a .mtch tag. Afterwards, use **knclust** to categorize the data. Can be used on labeled to labeled set, **knclust**(‘selfclust’,1) or for labeled to unlabeled set. Use **dispconfuse** if self clustering approach used in order to identify performance of algorithm. The function **gciconf** can be used to obtain indices of the cells of the confusion matrix. In order to view multiple spectrograms that had been categorized use **discliparr**(‘clipinds’,X) where X is the vector of indices. For visualization, use **ftrscatter** to view pair wise feature scatter plots and categories.

* **calcftrclips**— geneate feature file
* **calcmetric**—generate metric file for categorization
* **noisescreen**—filter data and generate noise label file
* **calcdistftr**—generate .mtch file holding similarity matrix
* **knclust**—perform k nearest neighbors algorithm to classify or verify
* **dispconfuse**—displays the confusion matrix based on self clustering for evaluation
* **gciconf**—obtain indices within cells of confusion matrix
* **dispcliparr**—display multiple spectrograms for visual review
* **ftrscatter**—display pair wise feature scatter plots

## Alignment

Templates should be prepared or performed with **calctemplates** at this point. Generate the dynamic time warp alignment of exemplars to templates with **calcdtw** function. Calculate alignment of features based on dtw and generate associated feature files/directories with **dtwalignftrs**. In order to see feature alignments and trace while viewing the template of a category use **plotalignftrs**.

* **calctemplates**—calculate temps of desired categories for label file
* **calcdtw**—perform dynamic time warping in order to align exemplars to given templates.
* **dtwalignftrs**—calculate the aligned features per bin per category and generate data file
* **plotalignftrs**—visually see the traced features per bin for exemplars while viewing template