**Instructions on running the algorithm**

The whole approach, from pre-processing to testing, is separated in 3 scripts.

1) The first script to run is the *run\_preprocess* script. This script goes through the BH dataset placed in the “mellisuga” folder, containing 6 good-quality recordings of BH sounds. The script does pre-processing on the samples, as described on the report, and plots the signal and spectrogram of each signal. It also stores the processed signal vectors in a cell array named **“sample”.**

2) Then the *make\_templates script* should be run. From pre-processing, I determined which 1-sec instances of BH sound patterns I will use to create the templates. I use the “sample” cell array to extract these instances and create the SEAV vectors for these sound pattern templates, as described in the report. I store 4 seav vectors, 2 for each sound pattern (chirp, buzz), in a matrix called **“seav”** with 4 columns, each column containing a seav vector. Columns 1 and 4 contain seav vectors for the chirp sound pattern, and columns 2 and 3 for the buzz sound. The script also plots these 4 seav vectors. The Euclidean distance between these pairs of vectors is computed, to give an idea of where the similarity threshold is placed in the detection script.

3) Then we run the *run\_detection* script to detect and test the algorithm for different test samples. Before running, we need to load the ***bh\_timestamps*** mat file containing the vector pairs of the time intervals where BH sound is visually spotted for each BH test recording. These are used to evaluate the algorithm’s accuracy. At the top of the script there are instructions on how to select the type of recording (BH, Other bird, random), and the recording number. The script processes the selected recording and outputs time instances where BH sound was detected. If the recording is a BH type, it also outputs the accuracy of the algorithm in percentage. The script also plots the spectrogram of the test recording.

**NOTE:** It is important not to clear the workspace between running scripts, except before running *run\_preprocess,* since data derived from a previous script are used in the next script.