Bioinformatics Institute
The University of Auckland

# Tutorial standalone Matlab package
# dtwave_cluster

**Louis Ranjard - l.ranjard@auckland.ac.nz**

May 12, 2014

# Introduction

This tutorial illustrates the usage of `dtwave_cluster` package for the automatic classification of bird song syllables. First, ensure that the MATLAB Compiler Runtime (MCR) is installed and ensure you have installed the same version as in the `dtwave_cluster` package (see readme.txt in the package directory). Please note that `dtwave_cluster` was developed to classify large amount of recordings [1] therefore the small data used in this tutorial does not constitute a representative dataset. However, this document shows how to use the package to perform a typical short and simple analysis.

# 1 Data set: Little Barrier Island Song Recordings

A set of 11 song recordings were collected on Little Barrier Island using a SongMeter equipment (Figure 1). Songs were first manually subdivided into their constitutive syllables and the sound signal of each syllable was saved as an independent WAV file (directory *data*). The song segmentation step can also be performed automatically, especially when analysing large dataset, but this is out of the scope of this tutorial.
The species name and syllable files for each song are as below:

- 4 saddleback songs
  - files 1.wav to 4.wav
  - files 5.wav to 9.wav
  - files 25.wav to 31.wav
  - files 37.wav to 42.wav
- 5 hihi songs
  - file 32.wav
  - file 33.wav
  - file 34.wav
  - file 35.wav
  - file 36.wav
- 1 long-tailed cuckoo song
  - files 10.wav to 19.wav
- 1 bellbird song
  - files 20.wav to 24.wav

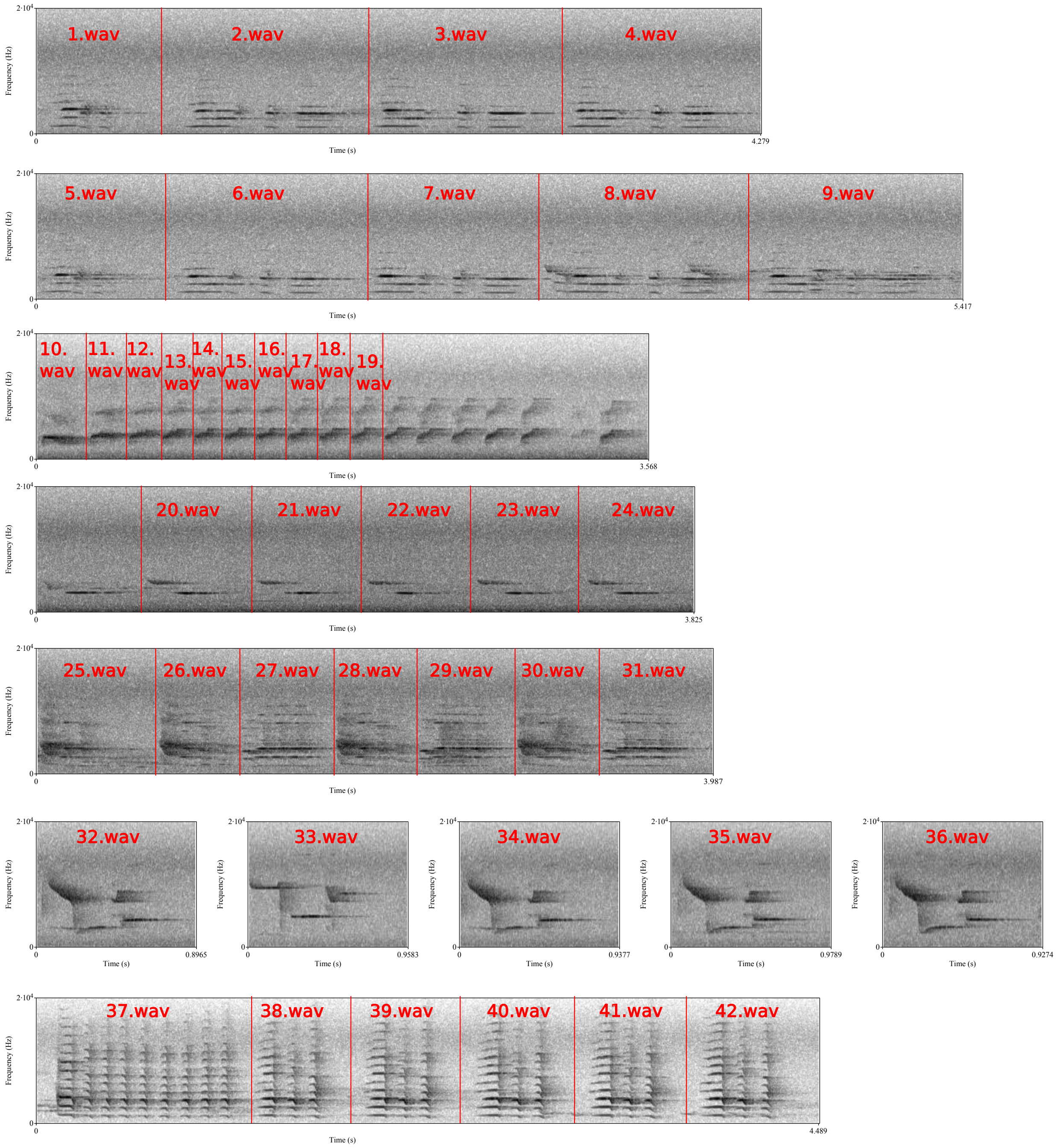


Figure 1: Spectrograms of recorded songs with syllable boundaries and file names.

## 2 Song Classification

Uncompress the package file `DTWave_cluster_pkg.zip` and, in a terminal, `cd` into it.

```
cd /path/to/DTWave_cluster_pkg/
```

The program `dtwave_cluster` can now be used to perform the automatic classification of the syllable recordings, using:

```
./run_dtwave_cluster.sh <mcr_directory> <wav files directory>
                   <argument_list formatted as: 'argument' value>
```

Several options can be passed to this function (see appendix for a complete list of options). In particular, we encode the songs with a short time window, 0.02 seconds, and partial overlap, 0.005 seconds, because we are interested in short variation in the signal. A larger window would be more efficient if for example one is aiming at classifying whole song recordings. The number of evolving tree classifications is set to 10 and the default values for the other parameters are used. Type in terminal:

```
./run_dtwave_cluster.sh /path/to/MATLAB_Compiler_Runtime/v81/ ./data 'wintime' 0.05
                   'hoptime' 0.02 'nrepeat' 2
```

```
****** dtwave_cluster: unsupervised sound classification ******

42 WAV files loaded
Parameters: 4, 0.5, 0.05, 2, 1, 4, 2, 2, 0.95

--------------------- classification tree: 1/10 ---------------------
epoch time -- Neighborhood Size, Mapping precision, Tree size, Leaves number
epoch 1/4 01-04-2014 18:32 -- 1, 10.0715, 41, 31
epoch 2/4 01-04-2014 18:32 -- 2, 10.2637, 41, 31
epoch 3/4 01-04-2014 18:32 -- 2, 10.5012, 44, 33
epoch 4/4 01-04-2014 18:32 -- 1, 9.43704, 71, 51
15 clusters
Davies-Bouldin indice: 5.15243
Classified syllable 1

--------------------- classification tree: 2/10 ---------------------
epoch time -- Neighborhood Size, Mapping precision, Tree size, Leaves number
epoch 1/4 01-04-2014 18:32 -- 1, 9.91937, 41, 31
epoch 2/4 01-04-2014 18:32 -- 2, 9.97057, 44, 33
epoch 3/4 01-04-2014 18:32 -- 2, 9.5719, 68, 49
epoch 4/4 01-04-2014 18:32 -- 1, 9.82024, 95, 67
17 clusters
Davies-Bouldin indice: 60.3539
Classified syllable 1

--------------------- classification tree: 3/10 ---------------------
...
```

A total of 10 classifications is performed. Note that because of the stochasticity in the order with which the syllables are used, the mapping precision, tree size and Davies-bouldin indices will vary between runs. For each classification replicate, the cluster number attributed to each syllable is recorded in the file `./data/dtwave_cluster.csv`.

```
1.wav  13 11 3 1 3 3 3 1 2 5
10.wav 5 6 1 7 2 10 11 3 11 8
11.wav 5 7 4 7 14 9 10 3 11 9
12.wav 5 10 4 7 13 9 10 3 11 9
13.wav 5 10 4 7 13 10 9 3 11 9
14.wav 5 10 4 7 13 9 10 3 11 9
15.wav 5 10 4 7 13 9 10 3 11 9
```

```
16.wav 5 10 4 7 13 9 9 3 11 9
17.wav 5 10 4 7 13 9 9 3 11 9
18.wav 5 10 4 7 13 9 9 3 11 9
19.wav 5 10 4 7 13 8 8 3 11 9
2.wav   4 3 5 4 8 4 13 7 5 4
20.wav 3 9 2 11 12 5 4 2 1 8
21.wav 3 9 2 5 4 5 4 2 1 8
22.wav 3 9 2 5 17 5 4 2 1 8
23.wav 3 9 2 5 12 5 4 2 1 8
24.wav 1 9 2 10 12 5 4 2 1 8
25.wav 6 8 6 2 13 9 4 10 1 8
26.wav 12 8 2 2 2 5 4 11 1 8
27.wav 11 2 9 2 14 1 3 1 13 9
28.wav 2 8 2 2 2 7 4 11 1 8
29.wav 13 4 3 2 15 1 3 10 13 9
3.wav   4 3 5 4 8 4 6 7 5 4
30.wav 1 8 2 2 2 7 4 11 1 8
31.wav 12 4 10 2 14 1 3 1 10 9
32.wav 7 5 7 6 1 6 7 9 9 10
33.wav 14 12 5 4 1 1 12 8 3 3
34.wav 7 5 7 3 1 6 7 9 9 7
35.wav 7 5 7 3 1 6 7 9 9 7
36.wav 7 5 7 3 1 6 7 9 9 7
37.wav 15 3 8 4 16 4 2 5 5 4
38.wav 8 1 11 9 10 2 5 1 6 1
39.wav 10 1 11 8 10 2 5 1 8 1
4.wav   4 3 5 4 8 4 13 4 5 4
40.wav 10 1 11 9 11 2 5 1 8 6
41.wav 10 1 11 8 11 2 5 1 7 1
42.wav 9 1 11 9 9 2 5 1 4 6
5.wav   2 2 10 2 5 5 3 1 2 2
6.wav   16 3 5 4 8 4 13 4 5 4
7.wav   14 3 5 4 6 4 13 6 5 4
8.wav   14 3 5 4 7 4 1 7 12 4
9.wav   14 3 5 4 6 4 6 5 5 4
Davies-bouldin 216.99 9.00 16.92 11.32 1943.98 479.09 184.51 50769.22 12037.95 5.16
```

First column contains the file names and the following columns contain the cluster number of each of the 10 classification replicates. It appears that the recordings 4 to 9 are consistently grouped together while the recordings 2, 3 and 10 are frequently grouped in the same cluster too. On the other hand, the recording 1 is never clustered with the recordings 2 to 10. This result shows that the syllables 2 to 10 share high acoustic similarity and the syllable 1 is different, indicating that it probably belongs to a different type. The syllables 2 to 10 (10.wav, 11.wav,..., 18.wav) belongs to the same song recordings showing high consistency between the syllables of this song.

## 3 Classification replicates summary

A way to summarize the classification replicates is to estimate the distances between recordings using the Jaccard distance, i.e. one minus the percentage of clusters that differ. These distances are calculated between recordings and represented as a dendrogram automatically if the number of syllables is less than 100.
Figure 3 represents the syllable pairwise distances as defined by the classification replicates. From the dendrogram it clearly appears that the syllables group into clusters. Each group constitutes a syllable cluster and is refereed to with a letter from $A$ to $G$.
It is now possible to encode the songs as a sequence of syllables using the summary dendrogram clusters. Figure 3 shows the song spectrograms with the cluster indicated for each syllable ($A$ to $G$).

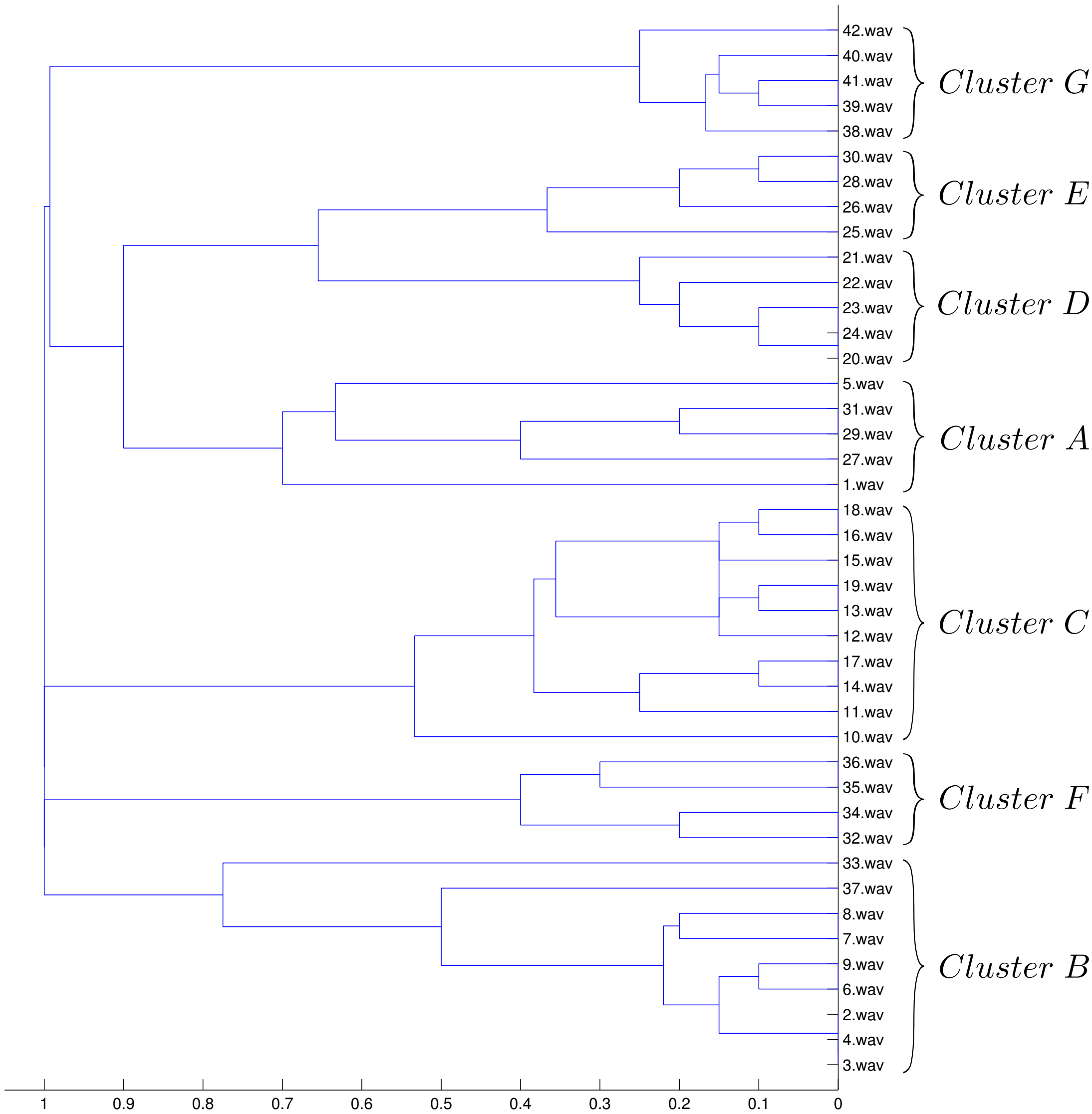


Figure 2: Syllable pairwise distances.

## 4 Classification analysis

The file `dtwave_cluster.csv` can be used to investigate biological questions related to the song recordings. These questions can be treated by integrating results over replicates. For example, one can ask how many syllables are shared between two sets of saddleback songs. We can answer that question by counting the number of times the syllables of the two different sets of songs are classified together. For example, using Matlab, first we load the classification results.

```
fid = fopen('data/dtwave_cluster.csv');
out = textscan(fid,'%s %d %d %d %d %d %d %d %d %d %d','delimiter',',');
fclose(fid);
```

Let's consider set 1 to contain the syllables from the first two saddleback songs (syllable 1.wav to 9.wav) and, set 2, the syllables from the two last saddleback songs (syllables 25.wav to 31.wav and syllables 37.wav to 42.wav). Sets 1 and 2 correspond to a variable (e.g. different individuals, locations, behaviours...) and the question is to investigate the amount of shared syllables across these two states. First, we need to retrieve the indexes of the syllable of each

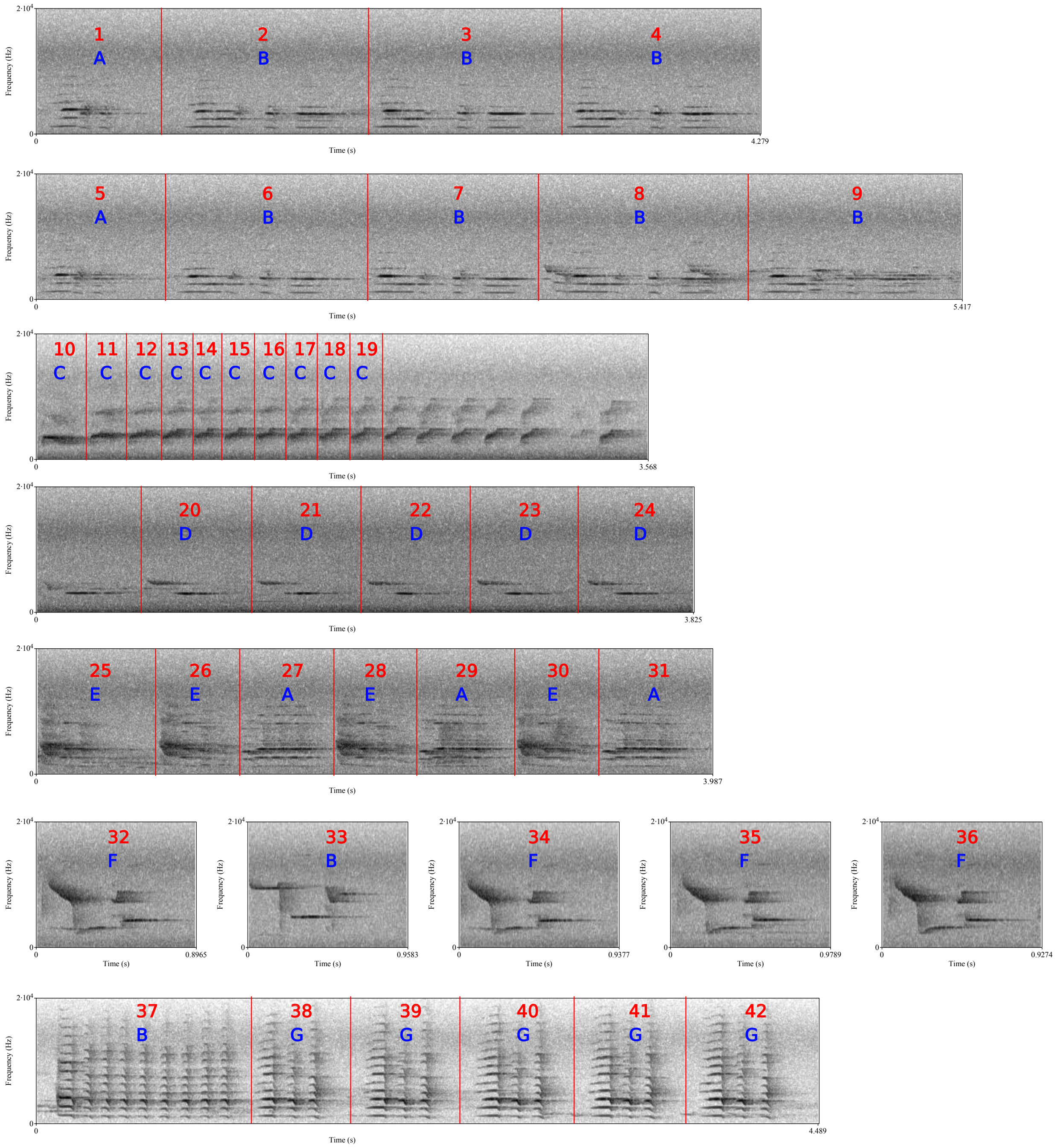


Figure 3: Song with encoded syllables following classification replicates summary.

set.

```
set1 = find( ismember(out{1},{'1.wav';'2.wav';'3.wav';'4.wav';'9.wav';...
      '37.wav';'38.wav';'39.wav';'40.wav';'41.wav';'42.wav'})==1 ) ;
set2 = find( ismember(out{1},{'5.wav';'6.wav';'7.wav';'8.wav';'25.wav';...
      '26.wav';'27.wav';'28.wav';'29.wav';'30.wav';'31.wav'})==1 ) ;
```

Then, we compare the cluster indices of the syllable of the two sets in each classification.

```
common = zeros(1,10) ;
for n=1:10
  common_syllable = numel( intersect( out{n+1}(set1), out{n+1}(set2)) ) ;
```

```
  unique_syllable = numel( unique( [out{n+1}(set1); out{n+1}(set2) ]) ) ;
  common(n) = common_syllable/unique_syllable ;
end
mean(common)
```

```
ans =

    0.2124
```

Therefore, on average the two set of songs share about 21% of their syllables. This result constitutes an estimate of the amount of similarity between the two sets in terms of amount of shared syllables.

# 5 Appendix - dtwave_cluster options

| Parameters | Description | Default value |
|---|---|---|
| htkconfigdir | directory for HTK configuration file(s) | |
| verbose | print information | 0 |
| outputfile | name and location for output file containing class. result(s) | ”wavdir”/dtwave_cluster.csv |
| wintime | window length (sec) | 0.3 |
| hoptime | step between successive windows (sec) | 0.1 |
| numcep | number of cepstra to return | 13 |
| lifterexp | exponent for liftering; 0 = none; $<0$ = HTK sin lifter | -22 |
| sumpower | 1 = sum abs(fft)$^2$; 0 = sum abs(fft) | 1 |
| preemph | apply pre-emphasis filter [1 -preemph] (0 = none) | 0.97 |
| dither | 1 = add offset to spectrum as if dither noise | 0 |
| minfreq | lowest band edge of mel filters (Hz) | 300 |
| maxfreq | highest band edge of mel filters (Hz) | 20000 |
| nbands | number of warped spectral bands to use | 26 |
| bwidth | width of aud spec filters relative to default | 1.0 |
| dcttype | type of DCT used; 1 or 2 (or 3 for HTK or 4 for feac) | 3 |
| fbtype | frequency warp: ’mel’,’bark’,’htkmel’,’fcmel’ | htkmel |
| usecmp | apply equal-loudness weighting and cube-root compr. | 0 |
| modelorder | if $>0$, fit a PLP model of this order | 0 |
| epoch | number of epoch for evolving tree | 4 |
| LR | learning rate for evolving tree | [0.5 0.05] |
| NS | neighborhood size for evolving tree | [2 1] |
| NC | number of children for evolving tree | [4 2] |
| thexpan | threshold above which a node is splitted | -1 |
| gama | weight decay on the hit counter | 0.95 |
| ce | use compression/expansion; 1 or 0 binary | 0 |
| nrepeat | number of classification replicates | 5 |
| depth | use a specific depth for classification; if 0 automatic | 0 |

# References

[1] Ranjard, L. and Ross H.A. *Unsupervised bird song syllable classification using evolving neural networks.* J Acoust Soc Am. 2008 Jun;123(6):4358-68