Documentation for Kaleidoscope Utils

using Kaleidoscope for bat call identification and analysis, free version 5.1.9g

Claudius Kerth
Leipzig
Germany
email: `claudiuskerth@gmail.com`

Contents

1 Pre-processing `meta.csv`

1.1 How to make MANUAL ID column in `meta.csv` blank

After batch processing, the MANUAL ID column in `meta.csv` contains the same as the AUTO ID column. This is unfortunate, since it is not clear whether a manual review of the auto id has taken place. An empty MANUAL ID field should mean that for this recording no manual inspection has been done yet. One way to make this column blank before manual inspection would be to open `meta.csv` in a spreadsheet and clearing the column manually. Another possibility is to use the following command line (one line):

```
perl -F"," -i -lane ’if($.==1){print; for($i=0;$i<@F;$i++)
{if($F[$i] =~ /^MANUAL ID$/){$Spalte=$i}}}
else{$F[$Spalte] = ""; print join(",", @F)}’ meta.csv
```

This command could be part of a pre-processing script for `meta.csv`.

1.2 How to sort `meta.csv` by date and time

Under Kaleidoscope Help → Metadata Panel → Results Window on can find:

> Rows can be sorted by clicking on column headers. Clicking on a column header again will reverse the sort order. Sort by multiple columns. For example, click on one column header to sort the data in that column, then click on a second column header to sort the second column. In this case, all the matching second column rows will be together and sorted in the order of the first column.

This works sort of after pressing 10 times and checking if the result is really sorted according to the two columns DATE and TIME. I generally would like to work chronologically through a recording session from earliest to last. The following command (one line) can sort `meta.csv` first by date then by time:

```
cat <(head -1 meta.csv) <(tail +2 meta.csv | sort -t, -k5 -k6)
> meta_sorted.csv
```

1.3 How to add a line number column to meta.csv

Occasionally I want to sort the table in the Results window (`meta.csv`) of K Viewer according to AUTO ID or maybe some other column. In order to quickly restore the previous chronological sorting order it would be nice to have an extra column with line numbers (or number of record) assigned after chronological sorting. Again, one could open `meta.csv` in a spreadsheet to add this column or one could use the following command line:

```
perl -F"," -ne ’chomp; if($.==1){print $_, ",NR", "\n";}
else{print $_, ",", $.-1, "\n";}’ meta_sorted.csv
> meta_sorted_NRcolumn.csv
```

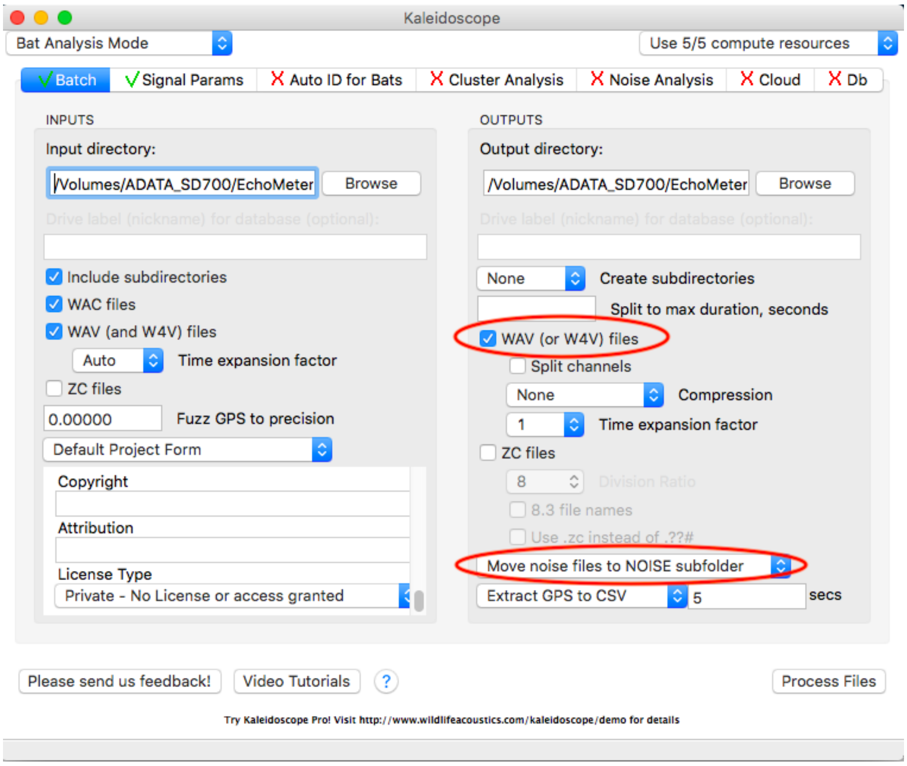
Note, that the NR column cannot be added as the first column or K viewer will not be able to open the file.

1.4 Noise detection

With the free version of Kaleidoscope, noise detection only works by copying files to the output directory, unfortunately. The `meta.csv` file does not contain an indication of whether a file got detected as noise. However, in the output of the Pro version of K, one can find in the `id.csv` file (only produced by the pro version) files that got designated as noise.

1.4.1 How to add the noise label to `meta.csv`

When batch processing with the Kaleidoscope free version one has to copy input files to the output directory. The required settings are shown in figure 1. Recordings that are detected as noise are then moved to a subdirectory called NOISE in the output directory.



Kaleidoscope

Bat Analysis Mode
Use 5/5 compute resources

✓ Batch | ✓ Signal Params | ✗ Auto ID for Bats | ✗ Cluster Analysis | ✗ Noise Analysis | ✗ Cloud | ✗ Db

INPUTS

Input directory: /Volumes/ADATA_SD700/EchoMeter Browse

Drive label (nickname) for database (optional):

☑ Include subdirectories
☑ WAC files
☑ WAV (and W4V) files
Auto Time expansion factor
☐ ZC files
0.00000 Fuzz GPS to precision
Default Project Form
Copyright
Attribution
License Type
Private - No License or access granted

OUTPUTS

Output directory: /Volumes/ADATA_SD700/EchoMeter Browse

Drive label (nickname) for database (optional):

None Create subdirectories
Split to max duration, seconds
☑ WAV (or W4V) files
☐ Split channels
None Compression
1 Time expansion factor
☐ ZC files
8 Division Ratio
☐ 8.3 file names
☐ Use .zc instead of .??#
Move noise files to NOISE subfolder
Extract GPS to CSV 5 secs

Please send us feedback!
Video Tutorials
?
Process Files
Try Kaleidoscope Pro! Visit http://www.wildlifeacoustics.com/kaleidoscope/demo for details

Figure 1: Necessary setting to allow noise detection with the free version of Kaleidoscope.

The following command (one line) inserts the word ”Noise” into the AUTO ID column of `meta.csv` if the recording has been detected as noise:

```
perl -F, -lane ’if($.==1){print}else{
($file = $F[2]) =~ s/^(.*)\.(.*)/$1_000.$2/;
$file = "NOISE/" . $file;
if(-e $file){$F[17] = "Noise"};
print join(",", @F)
}’ meta.csv > meta_with_noise_in_Auto_ID.csv
```

Note that on line 2 in the command, the file name taken from `meta.csv` has to be modified: the string ”_000” has to be inserted. The copied files in the output directory of batch processing have this additional string, but `meta.csv` contains the names of the source files. The command checks whether a recording file has been put into the subfolder NOISE.

1.5 How to add more than one species label per recording file

Under Kaleidoscope Help → Reference Guide → Metadata Panel one can find the following info:

> To assign multiple manual IDs, hold the Control key down while pressing one of the user-defined buttons (or the corresponding shortcut key). This will add the label to the Manual ID field, separating multiple entries with commas while also disabling the Auto next file button.

On a Mac press the cmd key instead of the ctrl key. The choice of a comma to separate species labels is unfortunate since `meta.csv` is comma separated. I would therefore recommend to edit multi-species labels so that they are separated by a semicolon.

Usually, when you save `meta.csv` with K it will add double quotes around each field entry, which disrupts further command line manipulation of this important file. If you have multi-species labels that are separated by a comma then use the following command line to separate them by a semicolon:

```
perl -F/\",\"/  -i -lane
'map {tr/,/;/} @F; print join(",", @F)' meta.csv
```

Afterwards, it is save to remove double quotes:

```
tr -d '"' < meta.csv > meta_noquotmark.csv
```

1.6 How to add notes

In the Metadata panel there is a field called notes. Under Kaleidoscope Help → Reference Guide → Metadata Panel it says:

> Under Notes there is an *editable* field which contains field notes and GUANO format metadata if present.

I have tried to edit this field with the free version of K, but it is not saved. It may be that only during batch processing, information that pertains to the whole session of recordings can be saved here. Notes for individual recordings cannot be saved here, at least with the free version of K.

A workaround, although more error prone than editing directly in the results window of K Viewer, is to create an `id_notes.csv` file that contains a selection of columns from `meta.csv` and then open that file in a spreadsheet application for editing together with K Viewer. Since `meta.csv` already contains a NOTES column, I suggest naming the new notes column "ID NOTES".

The following command will display a numbered list of column headers from `meta.csv`.

```
head -1 meta.csv  | tr ',' '\n' | nl

     1  INDIR
     2  FOLDER
     3  IN FILE
     4  DURATION
     5  DATE
     6  TIME
     7  HOUR
     8  DATE-12
     9  TIME-12
    10  HOUR-12
    11  LATITUDE
    12  LONGITUDE
    13  MODEL
    14  SERIAL NO
    15  FIRMWARE
    16  PREFIX
    17  NOTES
    18  AUTO ID
    19  PULSES
    20  MATCHING
    21  MATCH RATIO
    22  MARGIN
    23  FILES
    24  MANUAL ID
    25  ORGID
    26  USERID
    27  REVIEW ORGID
    28  REVIEW USERID
    29  INPATHMD5
    30  NR
```

I can then use the following command to create a copy of `meta.csv` with a selection of columns

```
cut -d, -f3,5-6,18,24,30 meta.csv > id_notes.csv
```

The new ”ID NOTES” column can be added to `id_notes.csv` in the spreadsheet application. With it `id_notes.csv` will have 7 columns. The file format needs to stay the same.

1.7 Preprocessing script

The command lines documented above have been wrapped up in a script called `preprocessing_meta.sh`. It should be run from within the output directory of K batch processing and before manual review of auto id. The scripts takes no command line arguments. It produces a new `meta.csv` while preserving the original in `meta_orig.csv`.

Specifically the script does the following:

1. check for signs that preprocessing has already been run on meta.csv
2. make a backup of the original meta.csv
3. make the MANUAL ID column blank
4. sort `meta.csv` by date and time
5. add a line number column
6. check if a directory called NOISE exists, if yes, then add noise label to AUTO ID column
7. create `id_notes.csv` for note taking in a spreadsheet app

2 Post-processing `meta.csv`

2.1 How to remove lines for noise recordings from `meta.csv`

The following command line removes lines from `meta.csv` that contain the word ”noise” (upper or lower case) in the column MANUAL ID:

```
perl -F, -i’.withNoise’ -lane’if($.==1)
{print; for($i=0;$i<@F;$i++){if($F[$i] =~ /^MANUAL ID$/){$Spalte=$i}}}
else{print if not $F[$Spalte] =~ /noise/i}’ meta.csv
```

The command also makes a backup of the `meta.csv` with lines for noise recordings.

2.2 How to join ID NOTES to `meta.csv`

After manual review of recordings in K Viewer, filling the MANUAL ID column and adding corresponding notes to `id_notes.csv`, the ”ID NOTES” column (here the seventh column) can be joined back to `meta.csv` with the following command line (one line):

```
join -t, -1 3 -2 1  meta.csv <(cut -d, -f1,7 id_notes.csv)
> meta_with_id_notes.csv
```

The upper command requires that `id_notes.csv` is comma-separated and that the id notes made in the spreadsheet contain no commas. The following command line makes sure that the ID NOTES column, which must be the last column of the file, has semicolons instead of commas if they exist:

```
perl -i -pe’next if not /"$/; ($pre_note, $note) = $_ =~ /(.*,)(".*"$)/;
$note =~ tr/,/;/; $_ = $pre_note . $note . "\n";’ id_notes.csv
```

This command should be run before the join command.

2.3 How to create a `meta.csv` for database upload

The resulting **`meta.csv`** file can be opened by K, if the input directory in the Control Panel of K is still set to the input directory of K batch processing (e.g. the **`Session_20200902_194538`** directory). If **`meta.csv`** also contains a column INDIR with the path to the audio files, then K can find the audio files (if they exist) no matter how the paths are set in the Control Panel.

A version of **`meta.csv`** that points to the audio files in the *output* directory of K batch processing can be created with the following set of commands (assuming the current working directory is still the output directory):

```
mv NOISE/* .
rmdir NOISE
rename 's/_000//' *wav
META_NR=$( echo $(( $(wc -l meta.csv | awk '{print $1}') -1 )) )
paste -d, <(echo "INDIR"; for i in $(seq $META_NR); do pwd; done) meta.csv > meta_withINDIR.csv
```

`meta_withINDIR.csv` would be suitable for upload to a database. Keeping two copies of the audio recording files is a waste of storage. The audio files in the input directory could now be deleted without loss of information.

2.4 How to create a KML file from the `meta.csv` file after manual id

The EMT app creates a KML file for each recording session. It would be nice to create an updated KML file after manual review of the recordings. This is what the programme **`style2kml.pl`** does. It takes a **`meta.csv`** (usually after reviewing the recordings and filling the MANUAL ID column) as well as a(ny) Session*kml file created by the EMT app and creates a KML that contains Placemarks that show the manual id (not the auto id from the EMT app) and the remaining info in **`meta.csv`** in a pop-up table in Google Earth. Note, that the programme relies on the **`ogr2ogr`** utility from GDAL.

`style2kml.pl`

```
my $usage = "usage:

$0 --meta <your meta.csv> --EMT_kml <kml from EMT app> [--species <species abbreviation>]

--meta your meta.csv file name (and path)
--EMT_kml any Session*kml file name from EMT app
[--species species abbreviation as used by Kaleidoscope, e.g. HYPSAV]

prints result to STDOUT
\n";

system("which ogr2ogr > /dev/null") == 0 or die "You need to have the utility ogr2ogr installed and in your PATH.
https://gdal.org/download.html\n";

my ($meta, $EMT_kml, $species);

sub parse_command_line {
    while(@ARGV){
        $_ = shift @ARGV;
        if(/^-{1,2}meta$/){$meta = shift @ARGV;}
        elsif(/^-{1,2}species$/){$species = shift @ARGV;}
        elsif(/^--EMT-kml$/){$EMT_kml = shift @ARGV;}
        elsif(/^-{1,2}h(elp)?$/){die $usage;}
    }
}

```

```
parse_command_line();
die $usage unless (defined($meta) && ($meta ne "") && defined($EMT_kml) && (
    $EMT_kml ne ""));

# extracts the style definition from a session kml produced by EMT app
# we want to reuse the style in our new KML file
sub get_style{
    my $style_fh;
    open($style_fh, "<", $EMT_kml) or die $!;
    while(<$style_fh>){
        print if /<Style/ .. /<\/Style>/;
    }
}


# generate KML from the meta.csv file after manual id
my $cmd;
if(defined $species and $species ne ""){
    $cmd = sprintf("ogr2ogr -f KML meta_raw.kml %s -a_srs \'EPSG:4326\' -oo
        X_POSSIBLE_NAMES=LON\* -oo Y_POSSIBLE_NAMES=LAT\* -oo
        KEEP_GEOM_COLUMNS=NO -where  \"\\\"MANUAL ID\\\" LIKE \'%%%s%%\'\" ",
         $meta, $species);
}else{
    $cmd = sprintf("ogr2ogr -f KML meta_raw.kml %s -a_srs \'EPSG:4326\' -oo
        X_POSSIBLE_NAMES=LON\* -oo Y_POSSIBLE_NAMES=LAT\* -oo
        KEEP_GEOM_COLUMNS=NO", $meta);
}
system($cmd) == 0 or die $?;


# open the KML to which the style should be added
open (my $metaKML, "meta_raw.kml") or die $!;
my ($manual_id, $date, $time, @manual_IDs);

# add required style info
# use MANUAL ID for marker style (the Session KML uses AUTO ID)
# add date and time in name tag (for chronological sorting in Google Earth)
while(<$metaKML>){
    print;
    # add the style definition after the Document XML tag
    if(/<Document.*>/){
        print get_style();
        #       print "\n";
    }
    # add marker style XML tag to each Placemark
    elsif(/"MANUAL ID"/ and /SimpleData/){
        ($manual_id) = $_ =~ />(.*)</;
    }elsif(/<\/ExtendedData>/){
        @manual_IDs = split(/;/, $manual_id);
        # MarkerStyle for first species
        print "\t<styleUrl>#MarkerStyle", $manual_IDs[0], "</styleUrl>\n";
        # put manual ids in name tag
        print "\t<name>", join(', ', @manual_IDs), "</name>\n";
    }
}

# clean up
system("rm -f meta_raw.kml") == 0 or die $?;
```

2.5 Postprocessing script

The above commands and programmes are wrapped up in a script called `postprocessing_meta.sh`. It should be run from within the output directory of K batch processing after manual review of auto id's, filling the MANUAL ID column in `meta.csv` and optionally the ID NOTES column in `id_notes.csv`. The script takes the following arguments:

- a KML file from the EMT app (you can find one in the input directory of K batch processing)
- a species code to create a KML for a certain species only (optional)

Specifically, the script does the following:

1. make a backup of `meta.csv` after manual review before post-processing (your work won't get lost!)
2. check that files `meta.csv` and `id_notes.csv` exist
3. remove double quotes and separate multi-species entries in MANUAL ID column by a semicolon instead of comma
4. check if all fields in MANUAL ID column of `meta.csv` are filled
5. remove carriage returns in `meta.csv` (created by K)
6. add system UID into the column REVIEW USERID of `meta.csv`
7. add the current date in a new column MANUAL ID DATE of `meta.csv`
8. replace commas with semicolons in ID NOTES column of `id_notes.csv`
9. join ID NOTES column of `id_notes.csv` to `meta.csv`
10. discard some useless columns from `meta.csv`
11. discards line with manual id ”noise”
12. creates a version of `meta.csv` that contains a column with the path to the output directory. This file would be suitable for upload to a database.
13. create a `meta.kml` file from `meta.csv` allowing for specification of a K species code to create a KML file for one species only

2.6 How to create a zip archive of a recording session for colleagues

It would also be nice to create a zip file with `meta.csv`, `meta.kml` and the corresponding audio files that can be sent to and opened by colleagues with K. We do not want to send junk files containing only noise. Noise files are not listed in the final `meta.csv` after post-processing. First we need to create a list of audio files to include in the zip archive from `meta.csv`:

```
cut -d, -f1 meta.csv | tail -n +2 | tr -d ’"’ | sed -E ’s/(.*)/*\1/’ >
    include.lst
```

Note, that the postprocessed `meta.csv` contains no quotation marks (”), but if you have opened that file with K after postprocessing and then saved it, K will generally have inserted quotation marks again. Hence, the upper command makes sure that they are removed again, if they exist.

Then we can create a zip archive called, for example, *MyRecordings* containing the chosen files:

```
zip -r MyRecordings . -i@include.lst -i meta.csv meta.kml
```

After sending this zip archive to your colleague, she could then unzip this archive with the following command to the output directory *MyRecordings*:

```
unzip MyRecordings.zip -d MyRecordings
```

. . . and open `meta.csv` with K after setting the input directory in K's Control Panel to *MyRecordings*. Note, that without setting the correct input directory, K will not be able to find the audio files listed in `meta.csv`.

3 Requirements

The commands and utility scripts described here heavily depend on UNIX command line programmes. They will therefore only work on MacOS or a Linux-flavour.

- `ogr2ogr`
- Perl `rename`