

# ffmpeg issues

*Christof Neumann*

Load package but don't provide `ffmpeg` location (to be done manually later).

```
library(avutils)
set_binaries(pathtosox = "~/Documents/utilities/sox", printmessages = FALSE)

test_binaries()
```

```
## ffmpeg seems not to work...
```

```
## sox seems to work; version: v14.4.2
```

```
## git seems to work; version: git version 2.21.0 (Apple Git-122.2)
```

```
## vagrant seems to work; version: Vagrant 2.2.4
```

Provide paths to different `ffmpeg` versions (note that the binaries are not included in the repo because the files are pretty large).

```
# set paths to ffmpeg binaries (not on github!)
f2 <- normalizePath("ffmpeg2.8.5/ffmpeg")
f3 <- normalizePath("ffmpeg3.3/ffmpeg")
f4 <- normalizePath("ffmpeg4.2/ffmpeg")
# check versions
system2(command = f2, args = "-version", stdout = TRUE)[1]
```

```
## [1] "ffmpeg version 2.8.5 Copyright (c) 2000-2016 the FFmpeg developers"
```

```
system2(command = f3, args = "-version", stdout = TRUE)[1]
```

```
## [1] "ffmpeg version 3.3 Copyright (c) 2000-2017 the FFmpeg developers"
```

```
system2(command = f4, args = "-version", stdout = TRUE)[1]
```

```
## [1] "ffmpeg version 4.2 Copyright (c) 2000-2019 the FFmpeg developers"
```

The videos I use come from [youtube](#) and were converted using an [online converter](#). It is the same video, but downloaded in four different formats (avi, flv, mp4 and wmv). Let's look at their properties:

filein	video_format	duration	bitrate	resol	width	height	fps
AVI.avi	mpeg4 (Simple Profile) (xvid / 0x64697678)	85.34	374	202x360	202	360	29.97
FLV.flv	h264 (Main)	85.34	504	406x720	406	720	29.97
MP4.mp4	h264 (Main) (avc1 / 0x31637661)	85.31	502	406x720	406	720	29.97
WMV.wmv	wmv2 (WMV2 / 0x32564D57)	85.36	399	202x360	202	360	29.97

So far so good, although here it is already clear that something is going on (i.e. the durations differ between the four videos). Now we have four different videos in different formats. The fact that they all stem from the same source video doesn't matter. What we do next is to extract the audio from each video using the three different `ffmpeg` versions I have available. The audio files go into a folder `tempaudio`.

The command to extract the audio is the following:

```
ffmpeg2.8.5/ffmpeg -i 'testvideo/AVI.avi' -y -ar 44100 -ac 1 'tempaudio/ffmpeg2_AVI.wav'
-hide_banner
```

And here are some info (via `sox`) for the resulting audio, separated by the different video formats. Still, all looking good here, i.e. *within* a video, the audio files seem superficially identical (same number of samples, duration etc. ...).

filename	channels	samplerate	resol	samples	duration	format	filesize
ffmpeg2_AVI.wav	1	44100	16	3763584	85.34204	16-bit Signed Integer PCM	7527246
ffmpeg3_AVI.wav	1	44100	16	3763584	85.34204	16-bit Signed Integer PCM	7527246
ffmpeg4_AVI.wav	1	44100	16	3763584	85.34204	16-bit Signed Integer PCM	7527246

filename	channels	samplerate	resol	samples	duration	format	filesize
ffmpeg2_FLV.wav	1	44100	16	3762176	85.31011	16-bit Signed Integer PCM	7524430
ffmpeg3_FLV.wav	1	44100	16	3762176	85.31011	16-bit Signed Integer PCM	7524430
ffmpeg4_FLV.wav	1	44100	16	3762176	85.31011	16-bit Signed Integer PCM	7524430

filename	channels	samplerate	resol	samples	duration	format	filesize
ffmpeg2_MP4.wav	1	44100	16	3762176	85.31011	16-bit Signed Integer PCM	7524430
ffmpeg3_MP4.wav	1	44100	16	3762176	85.31011	16-bit Signed Integer PCM	7524430
ffmpeg4_MP4.wav	1	44100	16	3762176	85.31011	16-bit Signed Integer PCM	7524430

filename	channels	samplerate	resol	samples	duration	format	filesize
ffmpeg2_WMV.wav	1	44100	16	3762176	85.31011	16-bit Signed Integer PCM	7524430
ffmpeg3_WMV.wav	1	44100	16	3762176	85.31011	16-bit Signed Integer PCM	7524430
ffmpeg4_WMV.wav	1	44100	16	3762176	85.31011	16-bit Signed Integer PCM	7524430

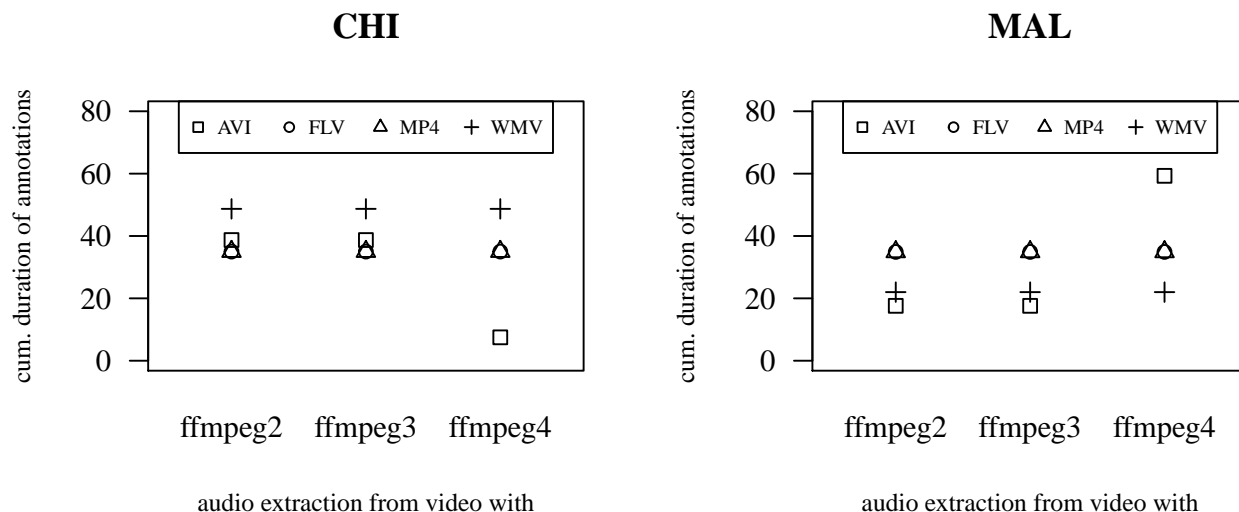
Now I run Marvin's yunitator on these 12 audio files. This is run in a virtual machine following the instructions in the [DiViMe documentation](#). The command I used is this:

```
vagrant ssh -c 'yunitate.sh data/ english'
```

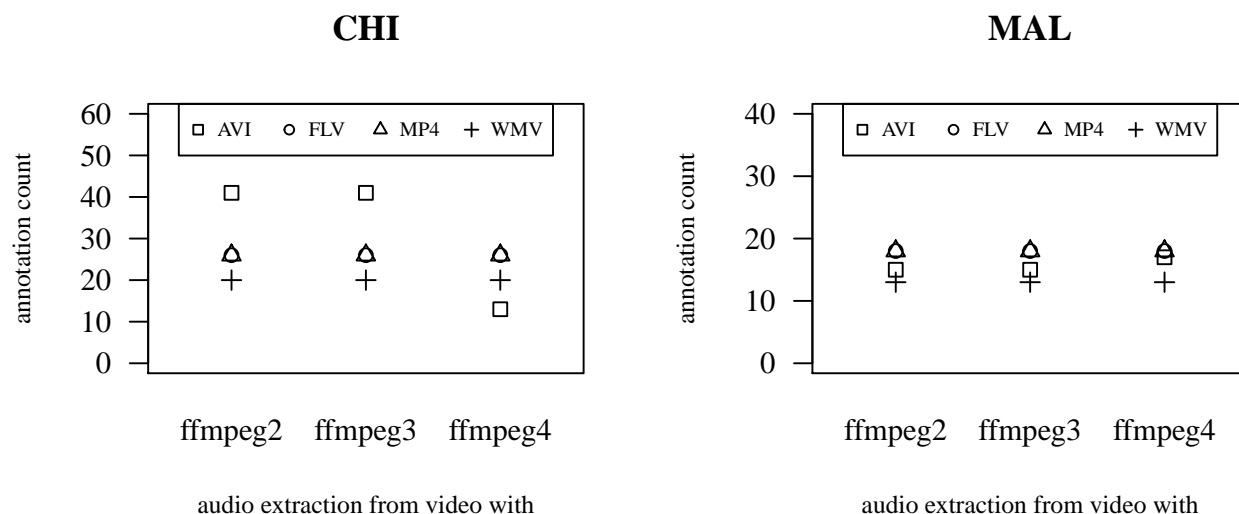
This produces 12 rttm files, corresponding to 12 audio files.

Let's look at the results. Since the recording is MAL-CHI, I drop any FEM that might have been recognized.

The first plot shows the cumulative duration of all annotations separated for CHI and MAL.



The second plot shows the number of annotations detected separated for CHI and MAL.



## summary

There are three take-home message from this exercise.

- 1) When the source for your audio is a video, it matters which version of **ffmpeg** you use to extract the audio. Somewhere between 3.3 and 4.2 there was a change.
- 2) The consequences of **ffmpeg** version seemingly work differently on different video formats, i.e. in the example here, only audio extracted from the **.avi** produced varying results.
- 3) And although consistent across different versions of **ffmpeg**, the different video formats produced different results (only **.flv** and **.mp4** produced stable results). In other words, even if you use a single

version (the most recent?) of **ffmpeg**, it appears to matter in which file format your video is saved.

To be clear, this is far from a comprehensive study (I only used one DiViMe tool and only one source video), but it suggests that this is likely an issue nevertheless.

Also, I don't know whether **ffprobe** and **soxi** might be better suited to extract information about media files, but this is probbaly a secondary issue if it is one at all.