Ease your automation, improve your audio, with FFmpeg

A talk by John Warburton,
freelance newscaster for the More Radio network,
lecturer in the Department of Music and Media
in the University of Surrey, Guildford, England

Ease your automation, improve your audio, with FFmpeg

A talk by John Warburton,
who doesn't have a camera on this machine.

My use of Liquidsoap

I used it to develop a highly convenient, automated system suitable for in-store radio, sustaining services without time constraints, and targeted music and news services.

It’s separate from my professional newscasting job, except I leave it playing when editing my professional on-air bulletins, to keep across world, UK and USA news, and for calming music.

In this way, it’s incredibly useful, professionally, right now!

What is FFmpeg?

It's not:

- "a command line program"
- "a tool for pirating"
- "a hacker's plaything" (that's what they said about GNU/Linux once!)
- "out of date"
- "full of patent problems"

It asks for:

- Some technical knowledge of audio-visual containers and codec
- Some understanding of what makes up picture and sound files and multiplexes

What is FFmpeg?

It is:

- a world-leading multimedia manipulation framework
- a gateway to codecs, filters, multiplexors, demultiplexors, and measuring tools
- exceedingly generous at accepting many flavours of audio-visual input
- aims to achieve standards-compliant output, and often succeeds
- gives the user both library-accessed and command-line-accessed toolkits
- is generally among the fastest of its type
- incorporates many industry-leading tools
- is programmer-friendly
- is cross-platform
- is open source, by most measures of the term
- is at the heart of many broadcast conversion, and signal manipulation systems
- is a viable Internet transmission platform

Integration with Liquidsoap? (1.4.3)

1. As an output processor and encoder
   - Can use Liquidsoap's own internal functions:
     - `output.external.ffmpeg`
     - `output.file.hls.ffmpeg`
     - `output.youtube.live.ffmpeg`
2. As the external coder for `output.icecast`:
     - `%external(...)`

Integration with Liquidsoap? (1.4.3)

3. As a very flexible input decoder:
   - `set("audio.converter.samplerate.converters",["libsamplerate"])`
   - `set("audio.converter.samplerate.libsamplerate.quality","best")`
   - `set("decoder.file_decoders", ["META","WAV","AIFF","MIDI","IMAGE","FFMPEG","FLAC","AAC","MP4","OGG","MAD"])`
   - `set("decoder.file_extensions.gstreamer", [])`
   - `set("decoder.mime_types.gstreamer", [])`
   - `set("decoder.file_extensions.ffmpeg",["mp3","mp4","m4a","wav","flac","ogg","webm","opus","mka"])`
4. As a framework for preparing audio for queue injection:
   - `/usr/local/bin/ffmpeg -y -i `/home/john/src/radio/getCBS.py` -af dynaudnorm=b=1:g=7 -t 02:55.200 -ac 1 -ar 32000 -acodec libfdk_aac -vbr 5 ~/src/radio/cbsnews-temp.mka && /usr/local/bin/ffmpeg -y -i ~/src/radio/cbsnews-temp.mka -filter_complex "[0:0]asplit=2[st][fi];[fi]atrim=start=163,silenceremove=stop_periods=1:stop_threshold=-30dB:stop_duration=0.4[cl];[st]atrim=end=163[stc];[stc][cl]concat=n=2:a=1:v=0,asetpts=N/SR/TB,dynaudnorm=b=1:g=7,volume=-11dB" -acodec libfdk_aac -ar 32000 -ac 2 -vbr 5 ~/src/radio/cbsnews.mka`

Integration with Liquidsoap? (1.4.3)

5. As a preparation tool:
    - for automatic track volume pre-determination;
    - for automatic track start and end detection;
    - for automatic positioning of fade-out point;
    - for semi-automatic library item de-duplication

Playlist preparation and weeding

- I have a directory containing all files available to Liquidsoap
- Incoming files need preparing, for Liquidsoap to use, and for playlist inclusion and annotation
- How to ensure an incoming file isn't already encoded?
- Use the hash function!

```
ffmpeg –v quiet –hide_banner \
–i <FILENAME> \
-vn \
–map 0:a \
–f hash –hash MD5 -
```

Playlist preparation and weeding

Don’t process files twice!

ffmpeg -v quiet \	Call ffmpeg, tell it to print no diags
-hide_banner \	Don’t print the “copyright”, etc., banner
-i <FILENAME> \	Here’s the input filename
-vn \	Don’t even *think* about any images in it
-map 0:a \	Use only the principal audio track
-f hash -hash MD5 -	Tell it to output a hash, of form MD5, and send the text to stdout

Playlist preparation and weeding

Don’t process files twice!

```
john@HP-S01:~/src/radio$
john@HP-S01:~/src/radio$ ffmpeg -v quiet -hide_banner -i /mnt/6TB-OCT2018/3TB-BACKUP/MUSIC/CamelPhat\ -\ Dark\ Matter\ 9/Wildfire_CamelPhat.mp3 -vn -map 0:a -f hash -hash MD5 -
MD5=cf351717a4648177ff38392d4db2ed63
john@HP-S01:~/src/radio$
```

```
'It Runs Through Me (feat. De La Soul)-407174856.94dc938862f75d6f545a38f621b43252.mka'
 Its_Getting_Better_Mama_Cass.41671593f924f77390bfb160d3efe981.mka
 Its_Rough_Out_Here_Montana_Orchestra.8b8f5994789c2b8d0a2c4e33ad77afe3.mka
 Its_Your_Thing_Shirley_Scott_The_Soul_Saxes.ac4718b483b63035660a59f5c3a2381f.mka
 I_Was_Kaiser_Bills_Batman_Whistling_Jack_Smith.924823208d4dcb70c1e90c9ca30e933b.mka
'I Wish-271131013.a549ac07d4b61639e458717032490f7f.mka'
 Ixtapa_Rodrigo_Y_Gabriela.e447003afb0e2e0817961471425 80f6f.mka
'Jack Lee - All of those things.7a5357638dea64ed266148f94f8f440c.mka'
'Jack Lee - An Episode of Journey.9845d4ef0f61c2032a4328ee44c5738a.mka'
'Jack Lee - Scenes From the Past.5c058dad60152e6cd87c992758bf34ef.mka'
'JACKSON SISTERS -  I Believe in Miracles.fc85d5eed58fc9654ab72d26979f5ef3.mka'
'Jack Wilkins - Pinocchio-2015960440.8604b05f6b5d6e284afcd9a6cce26496.mka'
'Jack Wilkins - Red Clay-2642769195.6cd91b4e0d129c5c03b355cb6425895b.mka'
'Jacob Mann Big Band - Bounce House-3998689220.f7e410cdfefdda90e116c4a6c5a8c89f.mka'
'Jacob Mann Big Band - Hold Music-961812703.c22ae63a1eda3eba1cc7fbce801791be.mka'
'Jacob Mann Big Band - Kogi-2933825373.480c8cb54aee9e49496dbde0261c1b51.mka'
'Jacob Mann Big Band - Pete Wheeler-3002525312.1727af467258d20be082d959692cb13d.mka'
'JAMES BROWN - HELL.e351e681d0b99c67438626186fc9f918.mka'
'JAMES BROWN -  My Thang.86a98964e2fa8ca46421a9ff28ffd493.mka'
'Jamiroquai - Too Young to Die (Extended).0adbcb050dcb40f7feeb5807de6a8c4e.mka'
```

You can then incorporate the hash into the filename (whether you choose to re-encode or not), so as to quickly spot if a particular piece of audio (bit-exact) has been processed in a previous session.

Playlist preparation and weeding

How loud is it? (Therefore, what playback gain is required? And when can we start and end/overlap the track on air?)

`ffmpeg \`	Call ffmpeg, but we *will* need diags.
`–hide_banner \`	Don’t print the “copyright”, etc., banner
`–i <FILENAME> \`	Here’s the input filename
`-vn \`	Don’t even *think* about any images in it
`-map 0:a`	Only examine the principal audio track
`-af ebur128 \`	Produce EBU R.128 loudness statistics
`-f null null`	But don’t produce anything else at this stage

Playlist preparation and weeding

```
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 209.4      TARGET:-23 LUFS    M: -63.7 S: -49.8     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 209.5      TARGET:-23 LUFS    M: -65.3 S: -50.6     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 209.6      TARGET:-23 LUFS    M: -66.5 S: -50.9     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 209.7      TARGET:-23 LUFS    M: -69.4 S: -51.3     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 209.8      TARGET:-23 LUFS    M: -70.4 S: -51.7     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 209.9      TARGET:-23 LUFS    M: -71.3 S: -52.3     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 210        TARGET:-23 LUFS    M: -72.2 S: -52.6     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 210.1      TARGET:-23 LUFS    M: -73.1 S: -53.1     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 210.2      TARGET:-23 LUFS    M: -73.9 S: -53.4     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 210.3      TARGET:-23 LUFS    M: -75.8 S: -53.9     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 210.4      TARGET:-23 LUFS    M: -77.3 S: -54.5     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 210.5      TARGET:-23 LUFS    M: -78.1 S: -54.8     I:  -8.5 LUFS       LRA:   8.7 LU
[Parsed_ebur128_0 @ 0x55857ad5c100] t: 210.6      TARGET:-23 LUFS    M: -83.8 S: -55.1     I:  -8.5 LUFS       LRA:   8.7 LU
size=N/A time=00:03:30.66 bitrate=N/A speed=83.2x
video:0kB audio:39500kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
[Parsed_ebur128_0 @ 0x55857ad5c100] Summary:

  Integrated loudness:
    I:         -8.5 LUFS
    Threshold: -19.3 LUFS

  Loudness range:
    LRA:        8.7 LU
    Threshold: -29.3 LUFS
    LRA low:   -15.3 LUFS
    LRA high:   -6.6 LUFS
john@HP-S01:~/src/radio$
```

By parsing this output, you can determine the track's loudness, and the shape of the loudness at the beginning and the end.

I generally ask the system to wait for a track's loudness's final dip to fall by 8dB.

Some internal logic also copes with very long tails e.g. "Bohemian Rhapsody".

Playlist preparation and weeding

- The data is encoded with the 'annotate' protocol for further enjoyment by the Liquidsoap engine.
- NOTE: with gain for replay, *always* set a *low* level as your standard replay gain, so there is never any clipping caused by amplification.
- Remember: Liquidsoap's internal audio processing is 64-bit (Romain, that's what you said, isn't it?)
- Therefore, there is no risk of losing significant audio data, either by clipping or by introducing quantising errors.

```
annotate:liq_cue_in="0.200",liq_cross_duration="4.380",duration="368.780",liq_amplify="-7.000dB":/home/
john/src/radio/mez3/18. Your Majesty Is Like A Cream Donut incorporating Oh What A Lonely Lifetime
[Bonus Track].bf1117eb12305fcd43e7bcaa27a6b6b7.mka
annotate:liq_cue_in="0.000",liq_cross_duration="8.160",duration="242.260",liq_amplify="-2.900dB":/home/
john/src/radio/mez3/04 - Elton John - Your Song.796a7bdb35fbb74a58bd3abdde2546b5.mka
annotate:liq_cue_in="0.000",liq_cross_duration="7.820",duration="467.620",liq_amplify="-13.900dB":/home/
john/src/radio/mez3/03. Your World.36edcfa36ffa09e35665067697321d68.mka
annotate:liq_cue_in="0.000",liq_cross_duration="12.130",duration="330.530",liq_amplify="-6.100dB":/home/
john/src/radio/mez3/05. you're as right as rain.adf1f237e3f724f9e09a75bdd3508a17.mka
annotate:liq_cue_in="0.000",liq_cross_duration="1.260",duration="168.460",liq_amplify="-6.100dB":/home/
john/src/radio/mez3/17 You're Gonna Need Me.9cae91a41925ead9da9edcbebd689e22.mka
annotate:liq_cue_in="0.000",liq_cross_duration="8.860",duration="256.960",liq_amplify="-7.000dB":/home/
john/src/radio/mez3/Carly Simon - You're So Vain.cbcae05e137342ab5080dcbbde993974.mka
annotate:liq_cue_in="0.000",liq_cross_duration="4.320",duration="196.620",liq_amplify="-7.800dB":/home/
john/src/radio/mez3/timmy thomas - 01 - you're the song i've always wanted to sing.
da91b16342433be8ae6e261cd5f54109.mka
annotate:liq_cue_in="0.000",liq_cross_duration="8.920",duration="310.620",liq_amplify="-8.100dB":/home/
john/src/radio/mez3/11. You've Got A Friend.b1c2c90672fa7b5a75559ed33e828509.mka
annotate:liq_cue_in="0.000",liq_cross_duration="8.060",duration="309.660",liq_amplify="-9.100dB":/home/
john/src/radio/mez3/07. You've Got A Friend.dc0ae46a6c416af537d54936e050d5cf.mka
annotate:liq_cue_in="0.000",liq_cross_duration="4.900",duration="269.800",liq_amplify="-3.300dB":/home/
john/src/radio/mez3/Youve_Got_A_Friend_James_Taylor.53204c21879c691826903712021c0c01.mka
annotate:liq_cue_in="0.100",liq_cross_duration="11.860",duration="349.060",liq_amplify="-4.900dB":/home/
john/src/radio/mez3/05 - You've Got It Bad Girl.32abba1893df601ec632b98c05ff47f7.mka
annotate:liq_cue_in="0.200",liq_cross_duration="2.090",duration="349.590",liq_amplify="-13.100dB":/home/
john/src/radio/mez3/08.  Uptown Funk Empire feat. Janice & Ange - You'Vee Got to Have Freedom.
c856f712dd8d21315683b91a99ade74b.mka
annotate:liq_cue_in="0.100",liq_cross_duration="2.680",duration="268.180",liq_amplify="-12.700dB":/home/
john/src/radio/mez3/14 - Yuri Buenaventura - Salsa.267534a7b8ff22b03d9fb863096b368d.mka
annotate:liq_cue_in="0.000",liq_cross_duration="12.270",duration="298.370",liq_amplify="-6.000dB":/home/
john/src/radio/mez3/Yussef Kamaal - Calligraphy _ Brownswood Basement Session-1g826StJhLk.
49884fd580506a4b8bc5bed56efd565b.mka
annotate:liq_cue_in="0.000",liq_cross_duration="8.200",duration="90.600",liq_amplify="-3.000dB":/home/
john/src/radio/mez3/Zahrafat_Al_SaId_Musicians_Of_The_Nile.d9ecfbb1511fc29faea3ae5b964aaac8.mka
@
```

Smooth on-air playback of this data

```
set("playlists.cue_in_metadata", "liq_cue_in")

myplaylist = amplify(override="liq_amplify", 1.0,
      cue_cut(playlist(length=60.0, reload_mode="watch",
      mime_type="audio/x-mpegurl", “MYPLAYLIST.m3u8")))

myplaylist = crossfade(fade_out=0.01, fade_in=0.01,
      default=(fun(a,b)->add(normalize=false,([b, a]))),
      conservative=true,  myplaylist)
```

Playlist preparation and weeding

What if the same piece of music exists in more than one form, but not bit-exact duplicates?

Use the Chromaprint library to process the first 20 seconds of each file:

```
fpcalc –algorithm 4 –overlap –length 20 –raw <FILENAME>
```


```
02 – Astrud Gilberto – Crickets Sing For Anamaria.ac30dcfc9da6acf2cf81340b41ec34c2.mka,
"2021322133211013,2021122132211013,2021122133211311,2021122133211211,2021122133211210,
2021122331211210,2031122131211110,2031122133211010,2021122133211110,2021122133211210,
2021122133211211,2021122133211312,2021122131211312,2021122131211312,2021122131211312,
2021122133211312,2021322133211312,2021122123211112,2021122323213112,2021122323213112,
2021122121313011,2021122122113010,2021322122112010,2021322121112010,2021322121112200,
2020122321112200,2020122321112200,2020022320112201,2020022233312301,3020032202313103,
3020032202311003,3020013203310003,1020013203310001,1020013203310001,1020013202310001,
0021033202331001,0032031100131001,0032021010131001,0032121130131001,0032321130131100,
0032211131330300,1032201032230200,1032301012230200,1032101012230201,3032001212130113,
3012101211030013,3012201210031002,3002203010031202,3000202010031202,3000202031031202,
3010202313033202,3030202212313201,3020112212213100,3020032211213000,3020022231203000,
3020023230213000,1022021223213000,1032021223113000,1032021320013000,1032220321033000,
1032230321031103,1032210123131202,3032200032331202,3032100012330212,3032001002130212,
3033001001031312,3031001203011032,3011201203010032,1010203300010032,1000203110011232,
1000202130113232,1010202123212231,1010102223212131,1010012223212020,1011032221212020,
1012033221212020,0012021323212020,0112020321102020,0312020221003021,0312020321000123,
0312020321000223,0312030332000223,0313010032010222,1311010012030222,1310000002120232,
1311100301020232,1113101200020132,1113303000020132,1110202000020212,1010202100010212,
1010202202001202,1010202302103102",93
```


Then process its output to give a sequence of digits from the set {0, 1, 2, 3} corresponding to strengths of audio across sixteen frequency bands with respect to time.

(Last item is duration in seconds.)

Playlist preparation and weeding

What if the same piece of music exists in more than one form, but not bit-exact duplicates?

Then use fuzzy matching algorithms (e.g. Levenshtein matching) to determine closeness of these fingerprints.

Use multiprocessing via Python front-end for speed.

Typically processes 4,000 tracks in around three hours, giving pairs of tracks with a 'match' estimate.

Filter on match estimate to taste.

Below, example shows invalid match on first line (low score) then three 'real' matches. These are spotted despite non-matching metadata.

```
70,05 - Guilty (2019 Remaster) - pitch fixed.2f4cfb7eea552e6bd861d4e8dd30f5d6.mka,02 Partido Alto Azymuth Light as a Feather.318b943f7efdd8a54736aa2ac23447ac.mka
83,13 Willie & Laura Mae Jones (Bonus Track).d3000e621fbc46cd478c560e3e76316b.mka,Willie_And_Lauramae_Jones_Dusty_Springfield.55eb4059acff46176a5ab1bb3e12439d.mka
100,Walk_On_The_Wild_Side_Lou_Reed.587dd736623e6647e8f8d4b92b75dc0c.mka,00001561_Walk On The Wild Side_Lou Reed.5714d62dff15d344be689ca28149c0f7.mka
92,Os Grilos.e552219d16beaf584d5f9f7212f18935.mka,Marcos Valle - Os Grilos.e57ffc94952b68122733802f81456ebd.mka
96,17 Express Yourself.c0975da4a881f2b4040cdba6571acbc4.mka,Various - Charles Wright && The Watts 103rd Street Rhythm Band   Express Yourself.d69dba2476212deb61c49e025a77b1f7.mka
```

On-air Sound

- FFmpeg's extraordinary audio filtering can give your station flexibility.
- Do you have different audiences requiring different sounds?
- Serve them all!
- My examples:
  - One for in-car or in-kitchen listening:
    with very low bandwidth (32kbit/s) and deep multi-band audio processing
    (like 'Skyrock' but taken rather further)
  - One for high-fidelity listening:
    maximum variable AAC bit-rate, very little limiting,
    EBU R.128 loudness adjustment in real time
  - Same as above, but low bit-rate to save bandwidth on dodgy links

On-air Sound

```
output.icecast(description="Experimental stream using Liquidsoap", genre="Freeform",
        name="Music Too", host="127.0.0.1", port=8000, mount="audio.aac",
        public=true, url="http://warblefly.sytes.net:8000/audio.aac",
        timeout=240.0, format="audio/aac", password=“<redacted>",
        %external(samplerate=48000, channels=2,
        process="ffmpeg -f s16le -ar 48000 -ac 2 -i pipe:0 -acodec libfdk_aac -vbr 1
        -profile:a aac_he_v2 -vn –af
        dynaudnorm=g=15:m=70:r=1.0:c=1:b=1,asetnsamples=2048,
        volume=-18dB,mcompand='0.005\,0.1 6 -47/-40\,-34/-34\,-17/-33 100 |
        0.003\,0.05 6 -47/-40\,-34/-34\,-17/-33 400 |
        0.000625\,0.0125 6 -47/-40\,-34/-34\,-17/-33 1600 |
        0.0001\,0.025 6 -47/-40\,-34/-34\,-17/-33 6400 |
        0\,0.025 6 -47/-40\,-34/-34\,-17/-33 15999’,
        volume=+20dB,aresample=192000,
        alimiter=limit=-4dB:attack=0.1:asc=1:asc_level=1,
        aresample=32000:resampler=swr:cutoff=0.99:filter_type=kaiser:kaiser_beta=16
         -f adts pipe:1"), radio)
```

On-air Sound

- When properly driven oversampled, the FFmpeg delay-line limiter is bomb-proof.
- The multi-band compressor, though lacking band-linking tools such as those in, say, Orban products or “Stereotools”, is still of very high quality, and provides arbitrarily many bands to tune your station sound.

On-air Sound

- http://warblefly.sytes.net:8000/audio.aac
- http://warblefly.sytes.net:8000/audio-hifi.aac
- http://warblefly.sytes.net:8000/audio-hifi-low.aac

- http://warblefly.sytes.net:8000/audio-am.aac

Pitfalls

- FFmpeg with the fdkaac encoding library is best for low-bandwidth audio — but you may need to compile it yourself
- When doing *any* processing, don't forget to oversample before limiting (think Nyquist/Shannon and overshoots)
- Don't imagine that basic limiting is sufficient to avoid over-deviation for an FM multiplex transmission that involves pre-emphasis
- Use raw encoding into FFmpeg for output, rather than assuming .WAV format — sometimes, size limits on .WAV are enforced

Hints

- Resample just before transmission to the lowest sample-rate your application requires. For FM sound-alike radio, 32,000 samples per second. (You can run internally, within Liquidsoap, any reasonable sample rate you like.)
- Try my FFmpeg binaries or compile scripts if you want a version with the FDK AAC library included. Note that Liquidsoap also has FDK AAC encoding capability (including HE-AACv2) but I have (sadly) not been able to get the lowest bandwidths to work in the past. Romain, is this ok now?
- https://github.com/Warblefly