Resampling

upsamp

upsamp(*path*, *outfile*, *up=4*, *order=128*) [source]

Increases the sampling rate of an audio file.

Args: path: string

Full path (including extension) of the audio file to convert.

outfile: string

Full path (including extension) of the new file.

up: int, optional

Upsampling factor. Defaults to 4.

order: int, optional

Length, in samples, of the anti-aliasing lowpass filter. Defaults to 128.

```
>>> import os
>>> home = os.path.expanduser('~')
>>> f = SNDS_PATH+'/transparent.aif'
>>> # upsample a signal 3 times
>>> upfile = os.path.join(home, 'trans_upsamp_2.aif')
>>> upsamp(f, upfile, 2, 256)
>>> # downsample the upsampled signal 3 times
>>> downfile = os.path.join(home, 'trans_downsamp_3.aif')
>>> downsamp(upfile, downfile, 3, 256)
```

downsamp

downsamp(*path*, *outfile*, *down=4*, *order=128*) [source]

Decreases the sampling rate of an audio file.

Args: path: string

Full path (including extension) of the audio file to convert.

outfile: string

Full path (including extension) of the new file.

down: int, optional

Downsampling factor. Defaults to 4.

order: int, optional

Length, in samples, of the anti-aliasing lowpass filter. Defaults to 128.

```
>>> import os
>>> home = os.path.expanduser('~')
>>> f = SNDS_PATH+'/transparent.aif'
>>> # upsample a signal 3 times
>>> upfile = os.path.join(home, 'trans_upsamp_2.aif')
>>> upsamp(f, upfile, 2, 256)
>>> # downsample the upsampled signal 3 times
>>> downfile = os.path.join(home, 'trans_downsamp_3.aif')
>>> downsamp(upfile, downfile, 3, 256)
```