Python Scipy FFT wav files

Asked 6 years, 2 months ago Active 2 years, 3 months ago Viewed 52k times

30

20

I have a handful of wav files. I'd like to use SciPy FFT to plot the frequency spectrum of these wav files. How would I go about doing this?

python scipy fft Edit tags

asked Apr 30 '14 at 0:31

user1802143

10.7k 13 39 47

10 Try googling each step (reading in a wav file, using FFT on the data). It should not be hard at all, come back here if you get stuck. – MattG Apr 30 '14 at 0:48

5 Answers

Active | Oldest | Votes

7

You could use the following code to do the transform:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from __future__ import print_function
import scipy.io.wavfile as wavfile
import scipy
import scipy.fftpack
import numpy as np
from matplotlib import pyplot as plt

fs_rate, signal = wavfile.read("output.wav")
print ("Frequency sampling", fs_rate)
l_audio = len(signal.shape)
print ("Channels", l_audio)
if l_audio == 2:
    signal = signal.sum(axis=1) / 2
N = signal.shape[0]
print ("Complete Samplings N", N)
secs = N / float(fs_rate)
print ("secs", secs)
Ts = 1.0/fs_rate # sampling interval in time
print ("Timestep between samples Ts", Ts)
t = scipy.arange(0, secs, Ts) # time vector as scipy arange field / numpy.ndarray
FFT = abs(scipy.fft(signal))
FFT_side = FFT[range(N/2)] # one side FFT range
freqs = scipy.fftpack.fftfreq(signal.size, t[1]-t[0])
fft_freqs = np.array(freqs)
freqs_side = freqs[range(N/2)] # one side frequency range
fft_freqs_side = np.array(freqs_side)
plt.subplot(311)
p1 = plt.plot(t, signal, "g") # plotting the signal
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.subplot(312)
p2 = plt.plot(freqs, FFT, "r") # plotting the complete fft spectrum
```

```
plt.xlabel('Frequency (Hz)')
plt.ylabel('Count dbl-sided')
plt.subplot(313)
p3 = plt.plot(freqs_side, abs(FFT_side), "b") # plotting the positive fft spectrum
plt.xlabel('Frequency (Hz)')
plt.ylabel('Count single-sided')
plt.show()
```

answered Mar 23 '18 at 11:08

bunkus
547 5 10

5 It works nicely. However, you need to fix the division operators; change N/2 to N//2 because that operation creates a float – pbgnz May 17 '18 at 2:36

2 @pbgnz Only for Python 3, or if you used `from __future__ import division` in Python 2 – supergra Jun 5 '19 at 17:58

0

This post is hidden. It was deleted 3 years ago by Bhargav Rao ♦.

When executing following step: plt.plot(abs(c[:(d-1)]),'r'), I am getting << TypeError: slice indices must be integers or None or have an **index** method>>. All previous steps work fine. Any idea why? Thank you!

answered Mar 23 '17 at 16:52

tiofabby
29 3

1 This does not provide an answer to the question. You can search for similar questions, or refer to the related and linked questions on the right-hand side of the page to find an answer. If you have a related but different question, ask a new question, and include a link to this one to help provide context. See: Ask questions, get answers, no distractions – Natty Mar 23 '17 at 16:53

comments disabled on deleted / locked posts / reviews

-1

This post is hidden. It was deleted 4 years ago by ChrisF ♦.

Ive used the following code from above:

```
import matplotlib.pyplot as plt
from scipy.fftpack import fft
from scipy.io import wavfile # get the api
fs, data = wavfile.read('test.wav') # load the data
a = data.T[0] # this is a two channel soundtrack, I get the first track
b=[(ele/2**8.)*2-1 for ele in a]
# this is 8-bit track, b is now normalized on [-1,1)
c = fft(b) # calculate fourier transform (complex numbers list)
d = len(c)/2  # you only need half of the fft list (real signal symmetry)
plt.plot(abs(c[:(d-1)]),'r')
plt.show()
```

I have used a different wav file.

I am getting the error: 'numpy.uint8' object is not iterable.

It points to line 15: b = [(ele/2**8.)*2-1 for ele in a]

I am new to Python so any help given will be greatly appreciated.

Thanks, Jackie

answered Feb 4 '16 at 10:02

J Hubbard

95 8

1 If you have a NEW question, please ask it by clicking the Ask Question button. Include a link to this question if it helps provide context. – Tunaki Feb 4 '16 at 10:17

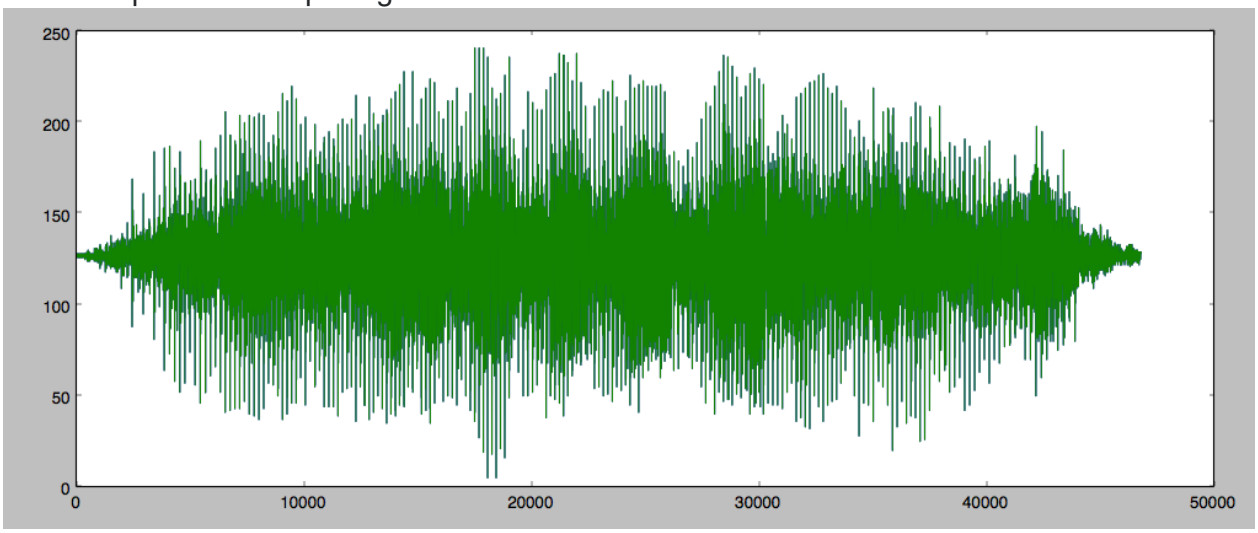
comments disabled on deleted / locked posts / reviews

64

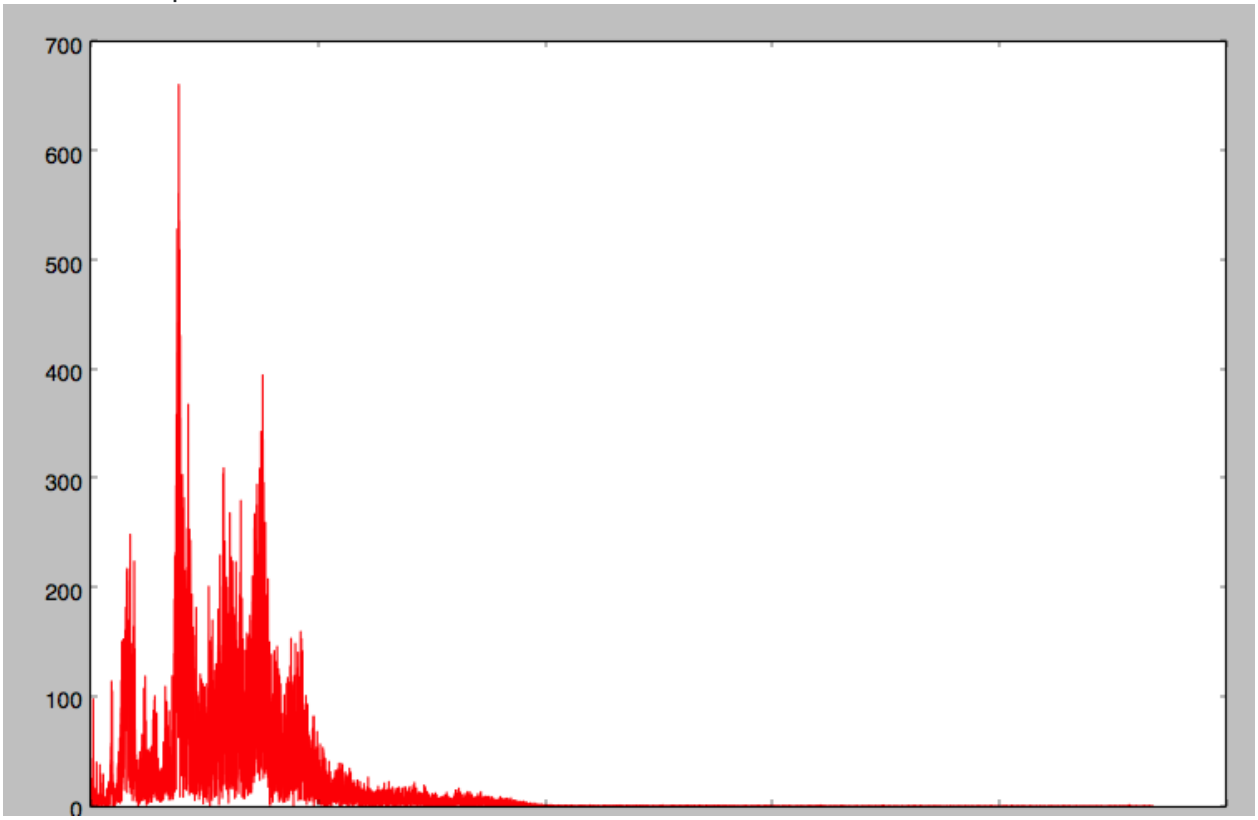
`Python` provides several api to do this fairly quickly. I download the sheep-bleats wav file from this link. You can save it on the desktop and `cd` there within terminal. These lines in the `python` prompt should be enough: (omit `>>>`)

```
import matplotlib.pyplot as plt
from scipy.fftpack import fft
from scipy.io import wavfile # get the api
fs, data = wavfile.read('test.wav') # load the data
a = data.T[0] # this is a two channel soundtrack, I get the first track
b=[(ele/2**8.)*2-1 for ele in a] # this is 8-bit track, b is now normalized on [-1,1)
c = fft(b) # calculate fourier transform (complex numbers list)
d = len(c)/2  # you only need half of the fft list (real signal symmetry)
plt.plot(abs(c[:(d-1)]),'r')
plt.show()
```

Here is a plot for the input signal:



Here is the spectrum



For the correct output, you will have to convert the `xlabel` to the frequency for the spectrum plot.

```
k = arange(len(data))
T = len(data)/fs  # where fs is the sampling frequency
frqLabel = k/T
```

If you are have to deal with a bunch of files, you can implement this as a function: put these lines in the `test2.py` :

```
import matplotlib.pyplot as plt
from scipy.io import wavfile # get the api
from scipy.fftpack import fft
from pylab import *

def f(filename):
    fs, data = wavfile.read(filename) # load the data
    a = data.T[0] # this is a two channel soundtrack, I get the first track
    b=[(ele/2**8.)*2-1 for ele in a] # this is 8-bit track, b is now normalized on
[-1,1)
    c = fft(b) # create a list of complex number
    d = len(c)/2  # you only need half of the fft list
    plt.plot(abs(c[:(d-1)]),'r')
    savefig(filename+'.png',bbox_inches='tight')
```

Say, I have `test.wav` and `test2.wav` in the current working dir, the following command in `python` prompt interface is sufficient: import test2 map(test2.f, ['test.wav','test2.wav'])

Assuming you have 100 such files and you do not want to type their names individually, you need the `glob` package:

```
import glob
import test2
files = glob.glob('./*.wav')
```

```
for ele in files:
    f(ele)
quit()
```

You will need to add `getparams` in the test2.f if your .wav files are not of the same bit.

edited Dec 1 '15 at 20:06

eusoubrasileiro
2,030 21 32

answered Apr 30 '14 at 1:45

yshk
741 5 7

5 Good answer! You may want to remove the `>>>` so the OP and others can copy and paste. Also I've found it helps the answer if you include a picture if your code makes a plot. – Hooked Apr 30 '14 at 2:54

Thanks. I have update the thread with prompt removed and new pictures. – yshk Apr 30 '14 at 3:53

1 How would you concatenate multiple wav files? I have a lot of small wav files. – user1802143 Apr 30 '14 at 6:38

I have on the order of a few hundred small wave files. So I need an efficient way to do it. – user1802143 Apr 30 '14 at 6:44

2 It `a = data.T[0]` should be it `a = data.T[0:data.size]` ? – Darleison Rodrigues Jun 8 '16 at 18:07

|

-1

This post is hidden. It was deleted 5 years ago by the post author.

I have been trying to get your code example to work. There seem to be an issue with the way you call fft. I have tried to fix it in the following way but it throws an error:

```
import matplotlib.pyplot as plt
from scipy.io import wavfile  # get the api
from scipy.fftpack import fft as sfft
from pylab import *

def f(filename):
    fs, data = wavfile.read(filename)  # load the data
    a = data.T[0]  # this is a two channel soundtrack, I get the first track
    b = [(ele/2**8.)*2-1 for ele in a]  # this is 8-bit track, b is now normalized on
[-1,1)
    c = sfft(b)  # create a list of complex number
    d = len(c)/2  # you only need half of the fft list
    plt.plot(abs(c[:(d-1)]), 'r')
    savefig(filename+'.png', bbox_inches='tight')

f("test.wav")
```

ERROR:

line 10, in f b = [(ele/2**8.)*2-1 for ele in a] # this is 8-bit track, b is now normalized on [-1,1) TypeError: 'numpy.uint8' object is not iterable

Any ideas?

answered Feb 9 '15 at 14:57

Søren Seeberg
1 1

This does not provide an answer to the question. To critique or request clarification from an author, leave a comment below their post - you can always comment on your own posts, and once you have sufficient reputation you will be able to comment on any post. – user3374348 Feb 9 '15 at 15:37

This does not really answer the question. If you have a different question, you can ask it by clicking Ask Question. You can also add a bounty to draw more attention to this question once you have enough reputation. – Bhargav Rao ♦ Feb 9 '15 at 16:02

comments disabled on deleted / locked posts / reviews