C10 Application of Maximum likelihood estimate in signal separation

SUPPLEMENTARY INFORMATION

Experimenter: Ziwei Huang 20980066
Participant: Ruijie Huang 20980062
Temperature: 21°C
Humidity: 35%
Date: 2022.03.15 Tue.

Contents

1 Exp.1 Linux basic operation

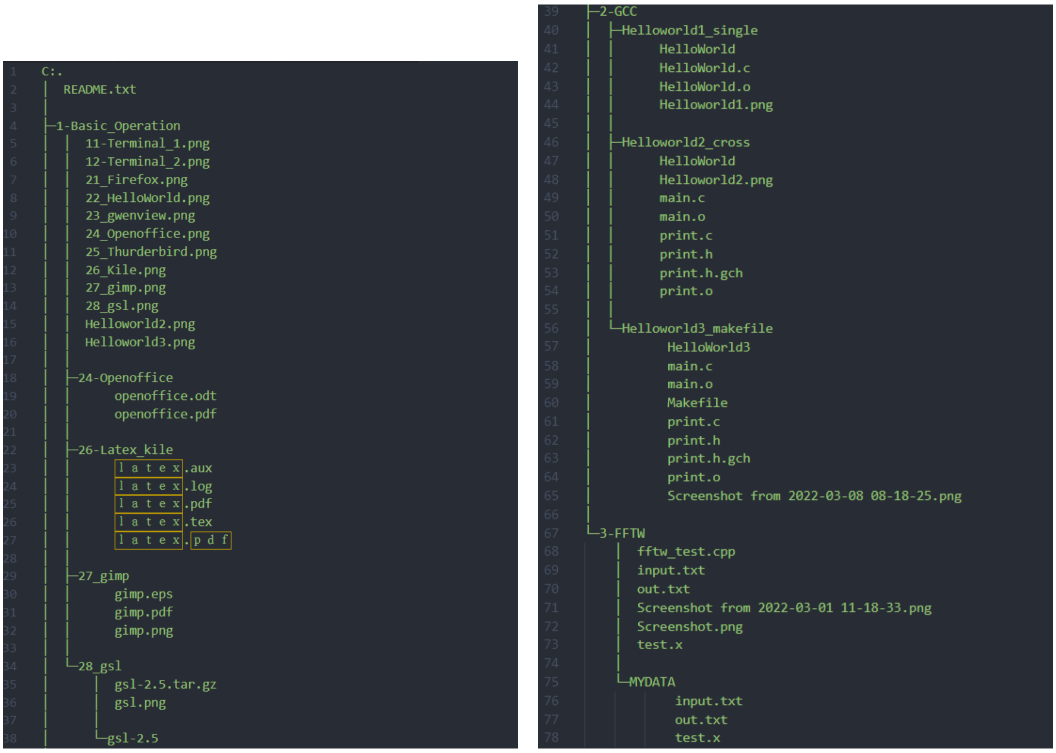
1.1 Main parameters

Item	parameters
Linux	Ubuntu 16.9

Table S1: **Parameters adopted in Exp. 1**

1.2 Supplementary data and figure

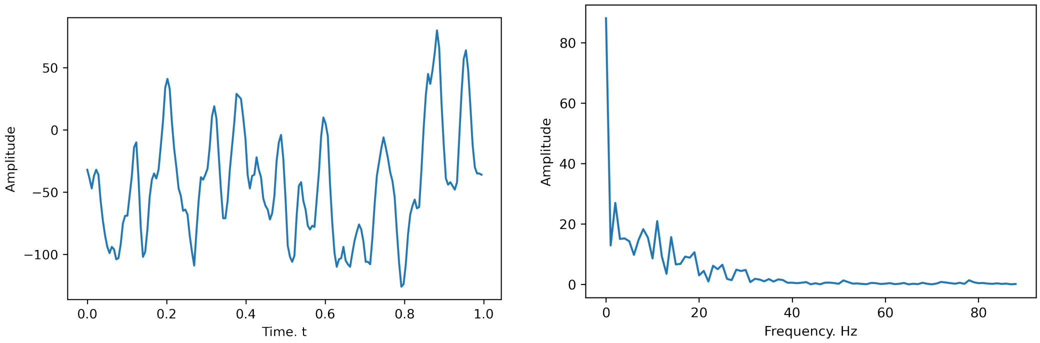
All Screenshots and output files have been uploaded onto the server. The directories and files are shown in Fig. S1a and Fig. S1b (More details are described in README.txt file)



(a) (b)

Figure S1: **Screenshots and output files in Exp.1**

Note: the data used in "3-FFTW/MYDATA" for the fftw computation are real-world data truncated from the electroencephalogram (EEG) signal of an epilepsy patient. Using *scipy.fftpack* we can perform fast fourier transform in Python. The results is shown in Fig. S2a and Fig. S2b.



(a) Original signal (b) FFT output

Figure S2: **Fast fourier transform of the real-world EEG data with** *scipy.fftpack*

1.3 Question: Usage of the fftw library

The Fastest Fourier Transform in the West (fftw) library is a comprehensive collection of fast C routines for computing the discrete Fourier transform (DFT) in one or more dimensions, of both real and complex data, and of arbitrary input size. It provides functions for four modes of transformation: complex one-dimensional transforms, complex multi-dimensional transforms, real one-dimensional transforms, and real multi-dimensional transforms.

The detailed usage can be found in `http://www.fftw.org/fftw2_doc/fftw_3.html`

2 Exp.2 Python basic operation

2.1 Main parameters

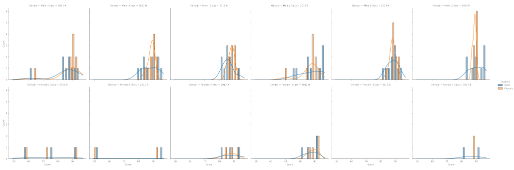
Item	parameters
Python	3.9.7

Table S2: **Parameters adopted in Exp. 2**

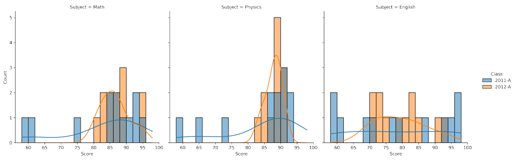
2.2 Supplementary data and figure

All codes and results are illustrated in the Notebook "C10.2.ipynb", which together with all output files, has been uploaded onto the server.

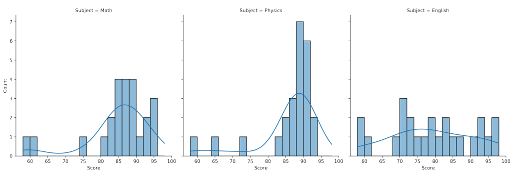
Note: Beside using the basic graphics engine in python, I also used an advanced drawing package "*Seaborn*"(`https://seaborn.pydata.org/`) to perform the statistics much elegentlier. Some results are illustrated in Fig. S3a Fig. S3b Fig. S3c and Fig. S3d.



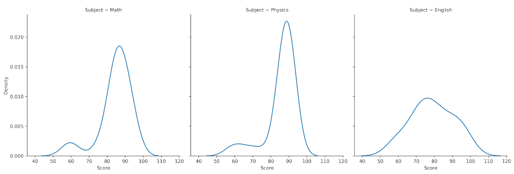
(a)



(b)



(c)

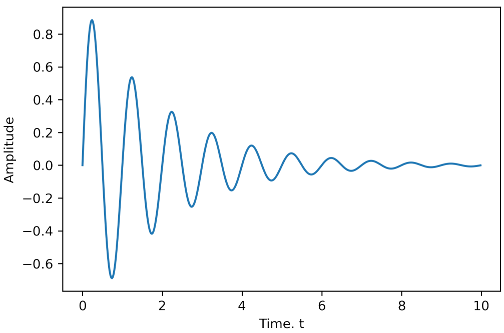


(d)

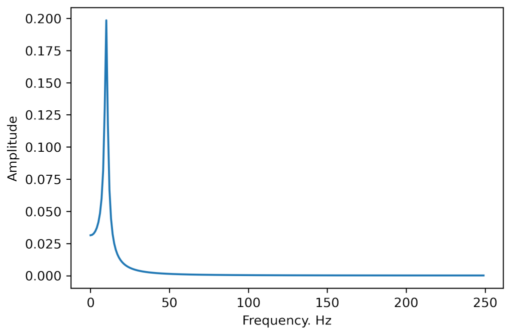
Figure S3: **Statistics performed with** ***Seaborn***

2.3 Question: Fast fourier transform of the Oscillation attenuation signal

The method of performing fast fourier transform on a signal has been illustrated in the Notebook. To be brief, I use $fftfreq$ to calculate the frequency, and fft to calculate the amplitude of every frequency. These functions are implanted in $scipy.fftpack$. The results are shown in Fig. S4a and Fig. S4b



(a) The original signal



(b) FFT result

Figure S4: **Fast fourier transform of the Oscillation attenuation signal**

3 Exp.3 Maximum likelihood estimate

3.1 Main parameters

3.2 Supplementary data and figure

3.2.1 Confidence intervals of different data

Note: All these data and corresponding outputs have been uploaded onto the server. The workflow was designed to allow reproduce the outputs easily. See the Notebook ”C10.3.ipynb”

Item	parameters
Python	3.9.7
numpy	1.20.3
pandas	1.3.4
scipy	1.7.1
matplotlib	3.4.3
seaborn	.11.2

Table S3: **Parameters adopted in Exp. 3**

Item	Setting	Estimate	+	-
Data_1-A events	100	92.84	+12.63	-11.89
Data_1-B events	200	207.16	+16.65	-15.88
Data_2_1-A events	150	139.90	+14.98	-14.26
Data_2_1-B events	200	210.10	+17.25	-16.45
Data_2_2-A events	200	205.62	+16.79	-16.11
Data_2_2-B events	200	194.38	+16.51	-15.70
Data_2_3-A events	250	249.20	+18.62	-17.92
Data_2_3-B events	200	200.80	+17.31	-16.48
Data_2_4-A events	300	310.35	+20.18	-19.49
Data_2_4-B events	200	189.65	+16.93	-16.09
Data_3_1-A events	50	43.88	+8.90	-8.17
Data_3_1-B events	100	106.12	+12.01	-11.23
Data_3_2-A events	75	78.33	+11.44	-10.71
Data_3_2-B events	150	146.67	+14.21	-13.43
Data_3_3-A events	125	124.16	+14.62	-13.88
Data_3_3-B events	250	250.84	+18.55	-17.77
Data_3_4-A events	150	161.35	+16.35	-15.62
Data_3_4-B events	300	288.65	+19.96	-19.17

Table S4: **Confidence intervals of different data** for details.

3.2.2 The regression parameters of the confidence intervals

Change the proportion. Class A: $y = 0.069x + 18.853; r = 0.994$; Class B does not show a linear correlation.

Change the total number. Class A: $y = 0.127x + 12.075; r = 0.994$; Class B: $y = 0.086x + 14.535; r = 0.998$

4 Data and code availability

Data and code are available at https://github.com/Jeg-Vet/SYSU-PHY-EXP/tree/main/