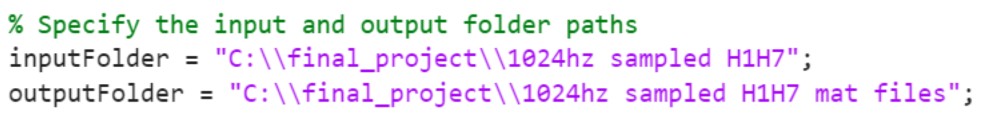
How to use Frequency Power Abstraction for Karma Lego

Step 1: Create a folder that contains all EEG. set and .fdt files.

Step 2: Create an output folder for EEG .mat files.

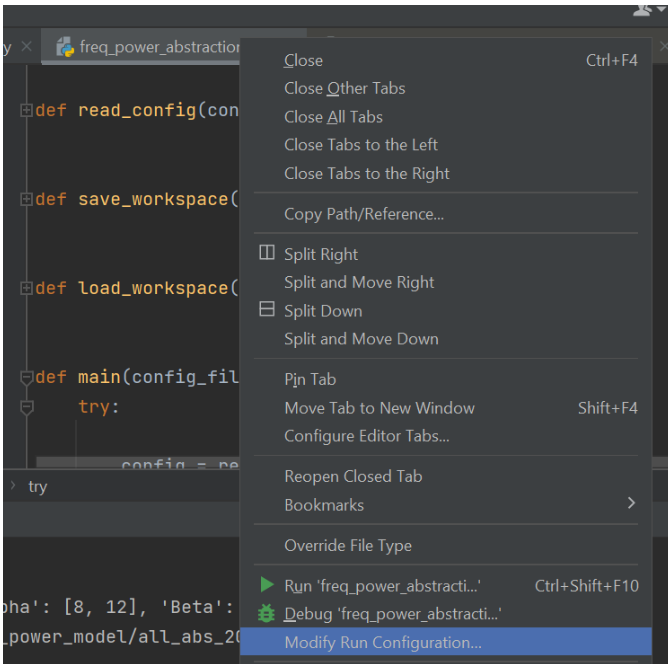
Step 3: Open **turning_set_files_to_mat_files.m** on MATLAB and change the following lines according to your files location folders:



```
% Specify the input and output folder paths
inputFolder = "C:\\final_project\\1024hz sampled H1H7";
outputFolder = "C:\\final_project\\1024hz sampled H1H7 mat files";
```

Step 4: Open pycharm and load **freq_power_abstarction.py** and **freq_power_abstraction_example_config_file.json**

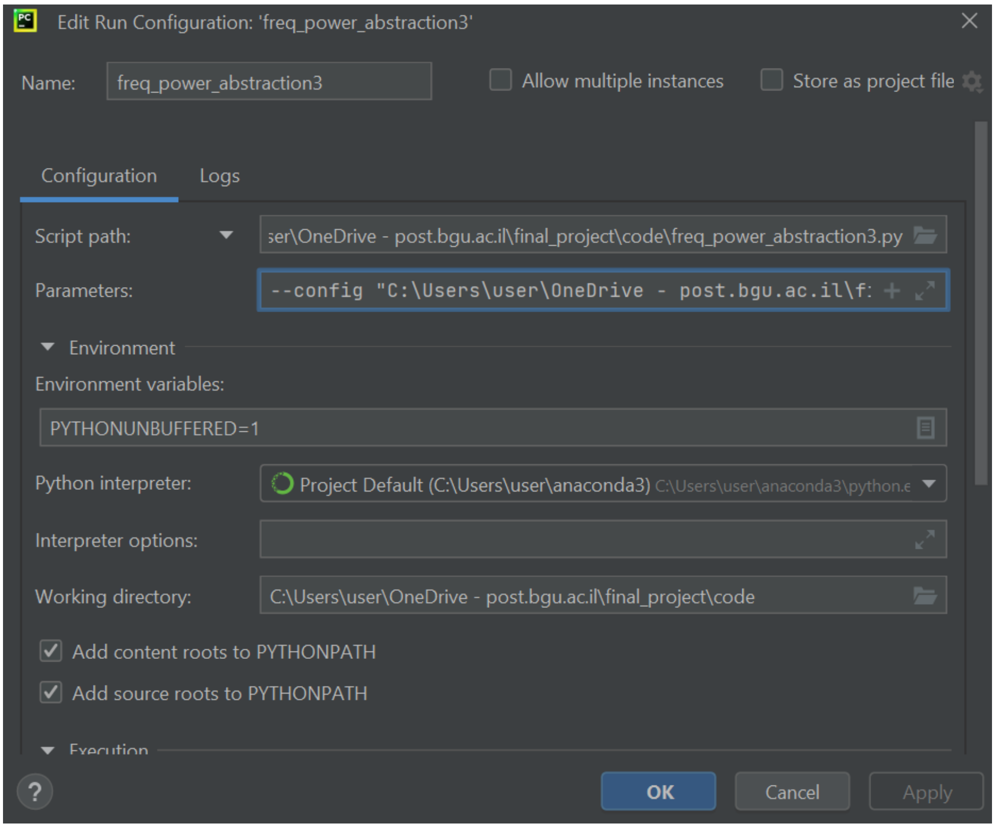
Step 5: Click right click on freq_power_abstarction.py file in pycharm and select "Modify Run Configurations…"



```
def read_config(con

def save_workspace(

def load_workspace(

def main(config_fil
    try:
        config = re
```

```
pha': [8, 12], 'Beta':
_power_model/all_abs_20
```

Under parameters add --config and file path to config file as follows:


Edit Run Configuration: 'freq_power_abstraction3'

Name: freq_power_abstraction3 □ Allow multiple instances □ Store as project file

Script path: ser\OneDrive - post.bgu.ac.il\final_project\code\freq_power_abstraction3.py

Parameters: `--config "C:\Users\user\OneDrive - post.bgu.ac.il\f:`

Environment variables: PYTHONUNBUFFERED=1

Python interpreter: Project Default (C:\Users\user\anaconda3) C:\Users\user\anaconda3\python.e

Interpreter options:

Working directory: C:\Users\user\OneDrive - post.bgu.ac.il\final_project\code

☑ Add content roots to PYTHONPATH

☑ Add source roots to PYTHONPATH

Step 6: Modify config file:

```
"sample_rate": 1024,  #add here your eeg files sample rate
"mats_folder_path": "C:/final_project/1024hz sampled H1H7 mat files",#add here the
.mat folder path in this format
"interval_length_ms": 200, #the length of the time intervals you want to study in
milliseconds
"number_of_categories": 4, #number of ordinal categorical levels of mean power for
each frequency band in frequency bands
"frequency_bands" : { #add here the frequency bands you want to study and their
following frequencies in Hz
  "Theta": [4, 8],
  "Alpha": [8, 12],
  "Beta": [13, 30]
},
"output_folder":#add here the output folder for the Json files abstractions
"C:/final_project/freqs_power_model/all_abs_200ms_L3_ThAB",
"direct_to_step": "False" #this is set to 'False' by default and can be used to
loading parameters from last point where there has been an error (inside main there
is if-else that can be modified for what steps you want to skip) it will only be
used if value = 'True'. At any point of error, the code is set to save the
workspace parameters. ,
"levels_to_save": [0,3] # add here in an array all the levels you want to
save the data of (here I am saving only the lowest and highest levels.
```

Step 7: Run **freq_power_abstraction.py** and hope for the best.

Important Notes:

- The code purpose is to create an ordinal categorical levels abstraction for each electrode (channel) and frequency band in constant time intervals. The calculation is made by using interpolation and FFT, calculation of mean power for each channel, band and time intervals. Afterwards calculate the mean of mean powers and std of mean powers for each channel and band and create the levels based on percentage of normal-distributed data with these mean and std, along with number of levels (categories). After the calculation of levels, the code assign levels for each interval, frequency band and channel (combinations) and saves it under Json file ready for single KarmaLego usage (one file for each entity – EEG record).
- To use multi KarmaLego there is need in merging the Json files and I will make another manual for it soon.
- Make sure that .mat files contain ‘session’, ‘subject’ and ‘chanlocs[‘labels’]’ in them and if not, you will need to modify load_entity_data() function. ‘session’ and ‘subject’ are relevant only for entity names and ‘chanlocs’ is important only for channel names. Each relevant to the final names of abstraction features and entity names in the final config file.
- The code uses roughly (size of .mat folder + 3G) RAM power
- Step 4 in code of calculating power is much longer than the rest so don’t panic…

Good Luck!