TremorTrackPlot

How to create meaningful graphs? Matplotlib!

- Version 2.1.2

```
import matplotlib
```

- Library for making 2D (or 3D) plots of arrays in Python.
- Three conceptual parts:
  - Pyplot: Set of functions to allow a user to create plots, similar to MATLAB

```
import matplotlib.pyplot as plt
```

```
plt.clf()
plt.savefig(pngfile, format='png')
```

  - Frontend/API: Set of classes that create and manage figures, text, lines

```
plt.plot(xlist, ylist, label=namelist[i], color=colorlist[i], linewidth=lwidth, alpha = a)
plt.xlabel("Frequency (Hz)", fontsize='x-large')
```

  - Backend – device specific: Agg creates PNG output

```
matplotlib.use('AGG')
```

What do you do when ‘stdout’ doesn’t work? io

- Matplotlib backend could not interface with ‘stdout’

```
import io
```

- The io module provides Python’s main facilities for dealing with various types of I/O, including:
  - Text
  - Raw
  - **Binary**

```
pngfile = io.BytesIO()
plt.savefig(pngfile, format='png')
pngfile.seek(0)
pngput = s3resource.Object(bucketout, '{}.png'.format(outname))
pngput.put(Body=pngfile.read())
```

How to interface Python with AWS? boto3

- boto3 is the AWS Software Development Kit for Python for interfacing with Amazon S3.

```
import boto3
```

```
s3client = boto3.client('s3', aws_access_key_id='XXXXXXXXXXXXXXXXXXXX', aws_secret_access_key='
s3resource = boto3.resource('s3', aws_access_key_id='XXXXXXXXXXXXXXXXXXXX', aws_secret_access_k
```

```
contents = s3resource.Bucket(bucket).Object(key=key).get()
```

```
s3client.delete_object(Bucket=bucket, Key=key)
```

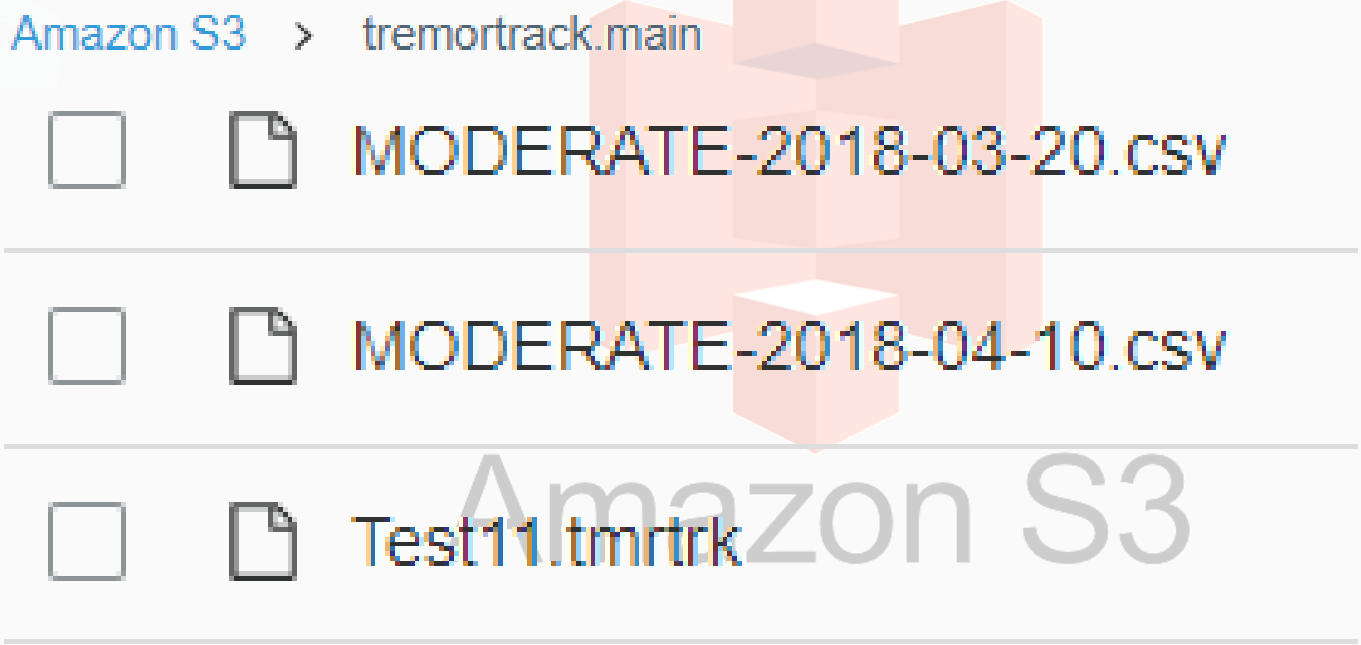
```
pngfile = io.BytesIO()
plt.savefig(pngfile, format='png')
pngfile.seek(0)
pngput = s3resource.Object(bucketout, '{}.png'.format(outname))
pngput.put(Body=pngfile.read())
```

Analyze (Reprise)

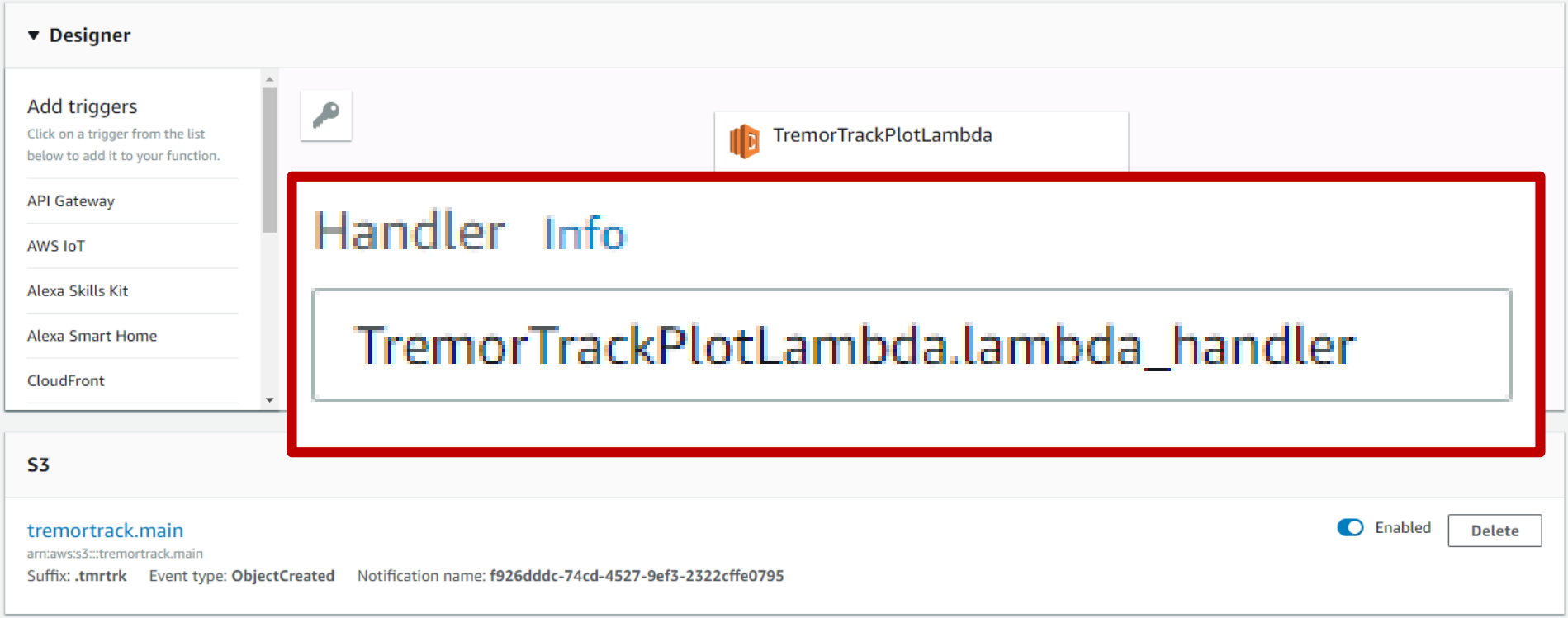
- User can transmit a single recording or multiple recordings
- Multiple recordings can be compared (up to 4 recordings), or averaged (up to 7 recordings)
- Recordings are sent one at a time
- Followed by a metadata file
    - This provides details on which recordings were sent, and how they will be analyzed

AWS S3 (Reprise)

- Repository for uploaded data from TremorTrack



AWS Lambda


Designer
Add triggers
Click on a trigger from the list below to add it to your function.
API Gateway
AWS IoT
Alexa Skills Kit
Alexa Smart Home
CloudFront
TremorTrackPlotLambda
Handler Info
TremorTrackPlotLambda.lambda_handler
S3
tremortrack.main
arn:aws:s3:::tremortrack.main
Suffix: .tmrtrk Event type: ObjectCreated Notification name: f926dddc-74cd-4527-9ef3-2322cffe0795
Enabled
Delete

TremorTrackPlotLambda: handler function

```
def lambda_handler(event, context):

    plots = []
    source = []

    # determine location, name, and contents of metadata file
    bucket=event['Records'][0]['s3']['bucket']['name']
    key=event['Records'][0]['s3']['object']['key']
    print("Bucket: {}\nKey: {}".format(bucket, key))
    action, source = metaFileOpen(bucket, key)
```

Contents of the Metadata File:

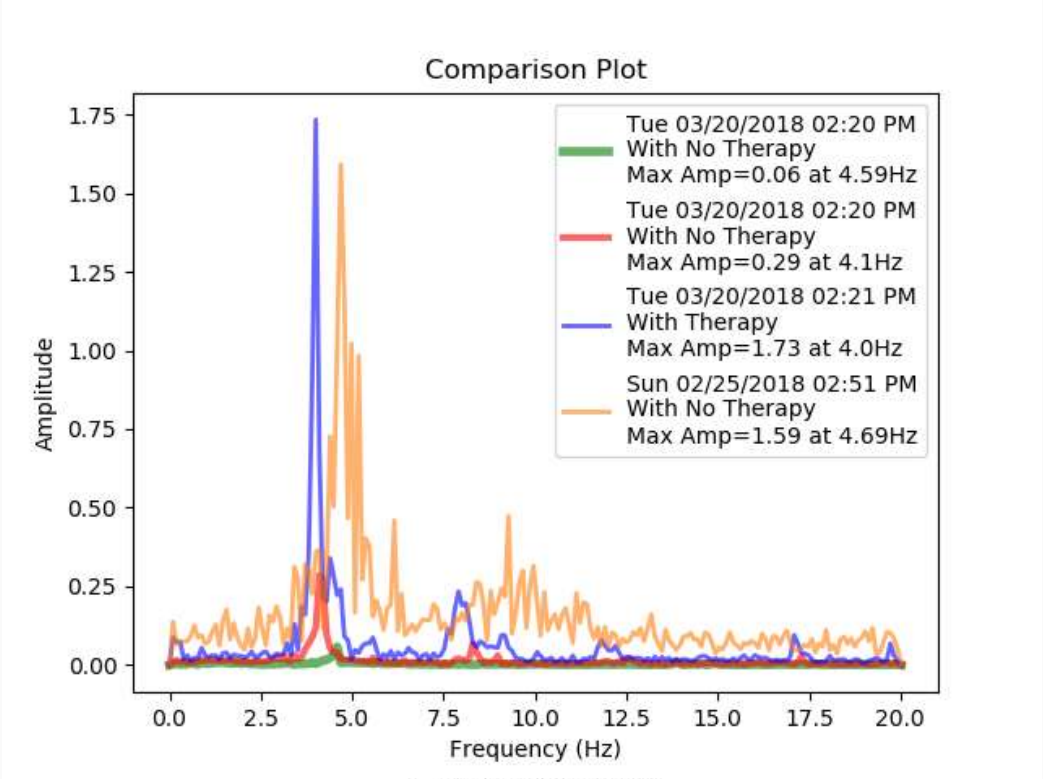
```
compareplot, MODERATE-2018-03-20.csv, MODERATE-2018-04-10.csv
```

TremorTrackPlotLambda Output

```
pngfile = io.BytesIO()
plt.savefig(pngfile, format='png')
pngfile.seek(0)
pngput = s3resource.Object(bucketout, '{}.png'.format(outname))
pngput.put(Body=pngfile.read())
```

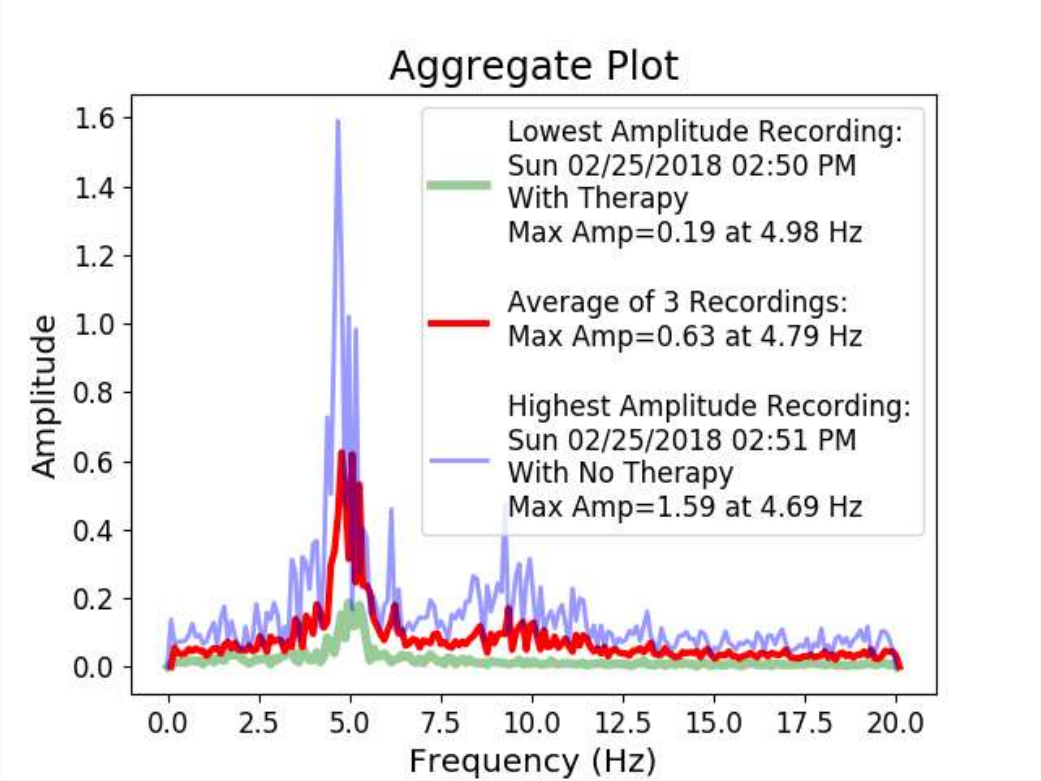
- Output to bucket = tremortrack.dl and key = metaDataFilename.png

TremorTrackPlotLambda Output: compareplot



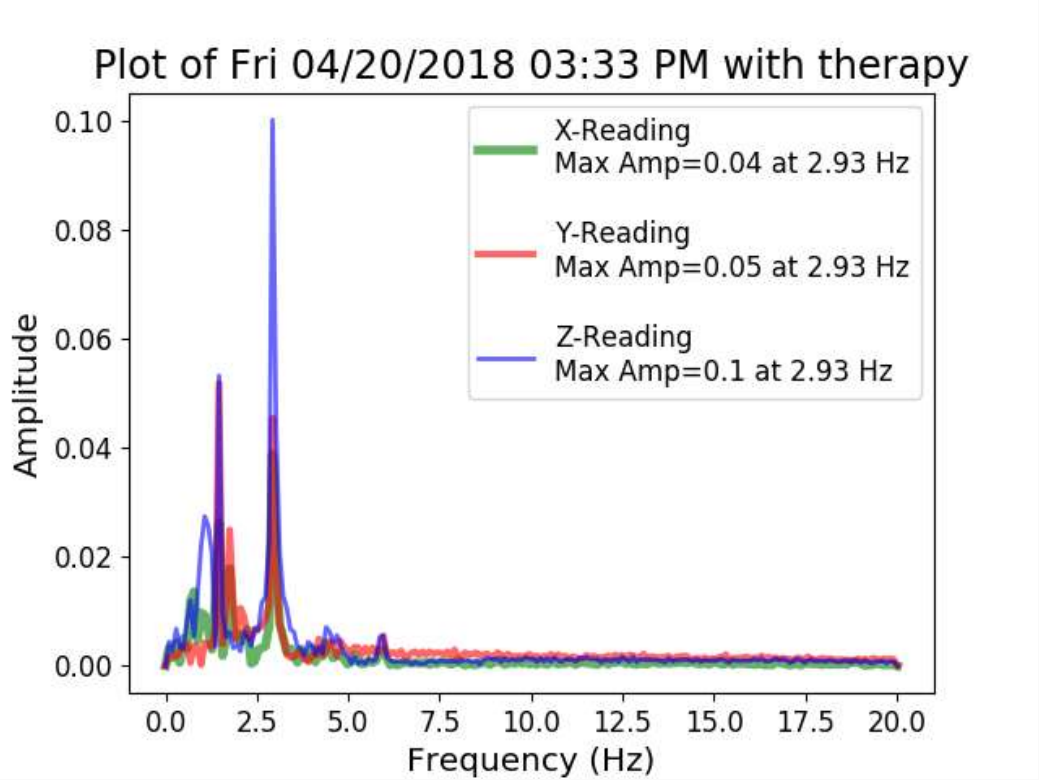
- Lower amplitude plot is always green with a thicker line.
- Higher amplitude plot(s) (by average) are red, blue, then orange with decreasing line thickness.
- Legend: Date, therapy, max amplitude and frequency

TremorTrackPlotLambda Output: aggregate



- Lower amplitude plot is always green with a thicker line. Highest amplitude plot is blue.
- Note, transparency of lines (alpha)
- Average is red and more translucent.
- Legend: Date, therapy, max amplitude and frequency

TremorTrackPlotLambda Output: singleplot



- X is green, Y is red, Z is blue: every time.
- Note, overlap of plots
- Title has date and indication of therapy

- Legend: accelerometer, max amplitude and frequency

The Agile Process

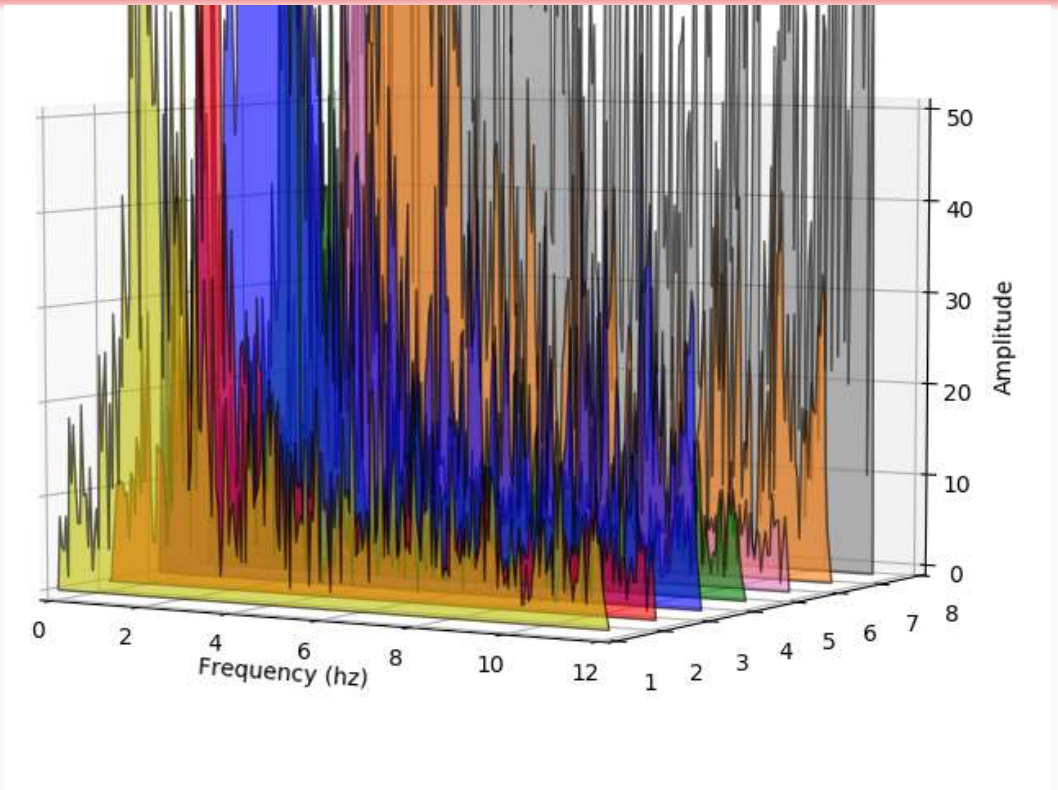
- Iterations of TremorTrackPlot in BitBucket:
  - TremorTrackPlot
  - TremorTrackPlot-boto3
  - TremorTrackPlotLambda

b7c42a8 M	Merge branch 'master' of https://bitbucket.org/jamiles/tremortrackplot	4/20/2018
c46b646	Added "clear figure" to the top of the method for plotting the graphs because graphs on AWS were overlapping and printing erroneous plots on the output.	4/20/2018
dc349ca	TremorTrackPlotLambda is a version of TremorTrackPlot that works within the "serverless" environment of Lambda in Amazon Web Services. Lambda is triggered by events in AWS and runs, in our case,	4/17/2018
760e9bb	TremorTrackPlot-boto3v4.py is a functioning version of TremorTrackPlot utilizing the "boto3" library to read and write files from and to Amazon Web Services s3 data storage buckets from an	4/17/2018
3e1ea35	Again, many updates: cleaned out many unused functions, variables, and libraries (program started with 3D libraries that aren't used), program now assesses peak amplitude and the corresponding	4/10/2018
b66ab6b	Many changes made: program now reads from a MetaData file; added info to output concerning therapy; output file relfects the name of the MetaData file; decreased the number of plots to be a	4/9/2018
c71e8a3	Converted frequencyConvert and amplitudeComvert variables to a decimal value for consistent handling in AWS EC2. Commented the equation that determines the decimal value for future	4/4/2018
659a9b6	Cleaned up the code. Fixed the calculation for frequency. Added a "hertzleng" variable for setting the maximum calculated frequency (hertzleng is actually the last indexed variable calculated in the Fast	3/18/2018
96b0764	TremorTrackPlot now includes methods and functionality for comparing and aggregating data.	3/5/2018
c803b10	TremorTrackPlot created online with Bitbucket	2/27/2018
19c1b42	Initial commit	2/27/2018

The Original TremorTrackPlot

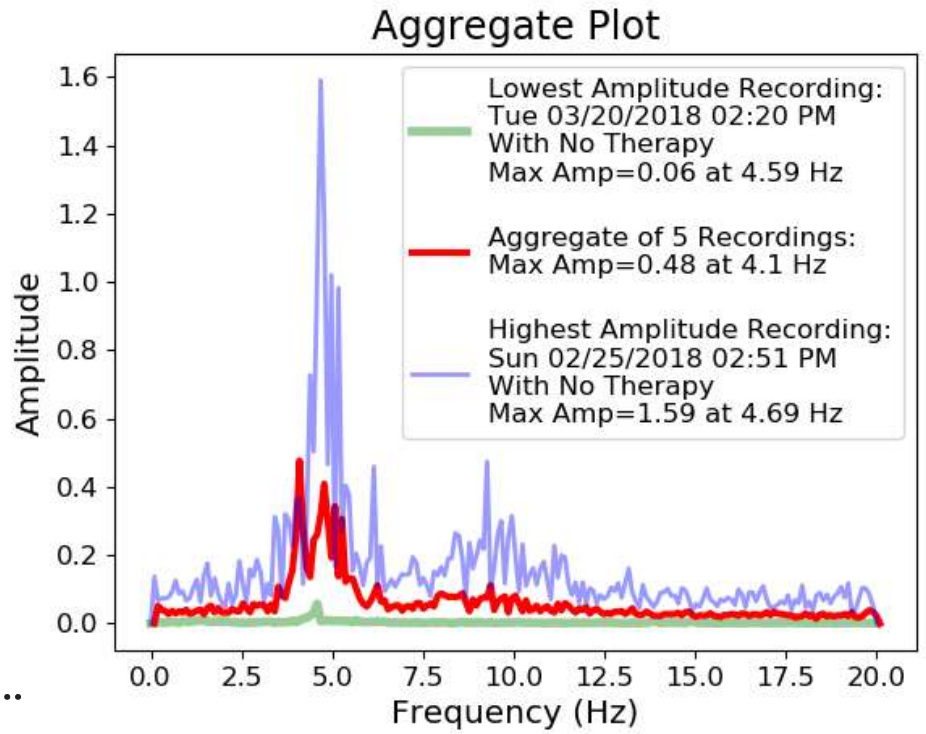
- Create functions to interpret data in the csv files.
- Plot data in the csv files.
- Multiple plots in parallel = 3D graph?
  - Our first deliverable:

    “Mic Drop!”
  - The next step:
    - Time to re-design…



Moving TremorTrackPlot to AWS: EC2 and boto3

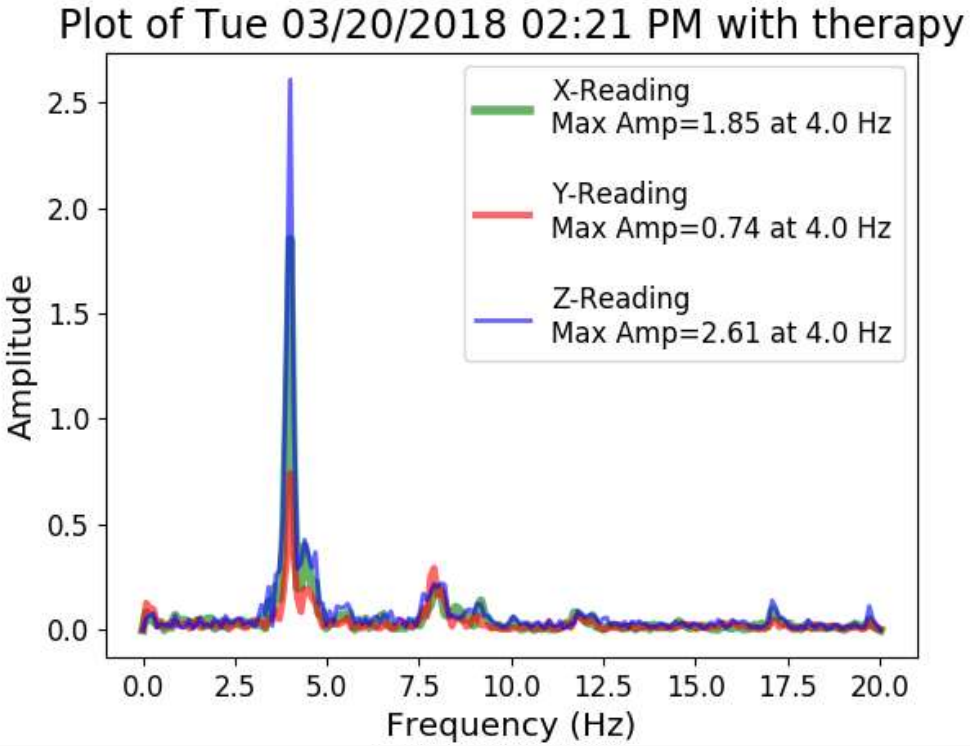
- After improving the graph, TremorTrackPlot was moved to AWS Elastic Compute Cloud (EC2) and used boto3
- At this point, still a command prompt interface.
  - Data read from AWS S3 and output copied back to AWS S3
  - Time to move to Lambda…



Moving TremorTrackPlot-boto3 to Lambda:

- Zip TremorTrackPlot and needed libraries from EC2 into a “Deployment Package”
- Copy to S3
- Upload into Lambda
  - and it works… eventually

- And – Tested over 200 times!



TremorTrack

- Agile Tasks
- Constant Communication
- Teamwork
- Success!