19.08.27 PEM meeting

Development of fujimopy

Fujikawa (Niigata), Mori (Toyama)

About fujimopy

- -Fujikawa と Mori さんで作成
- -バージョンO1.5
- -スペクトルのピークを検出して表示してくれる おそらく、時間など他の領域でも使える
- -様々な条件でピークを検出できる(できたい)
- -すでにKozapyなどにある機能??

How to use

xlogをTrueにするとlogスケールになる

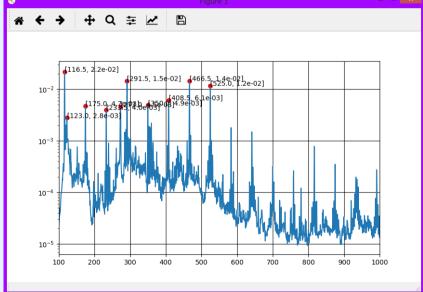
```
①importする
②パラメータ決める
x1,x2:見たい範囲
odr :ピーク検出のオーダー
pp :ピークを何個プロットするか
③データファイルを読み込む
④メソッド「peak」を使う
```

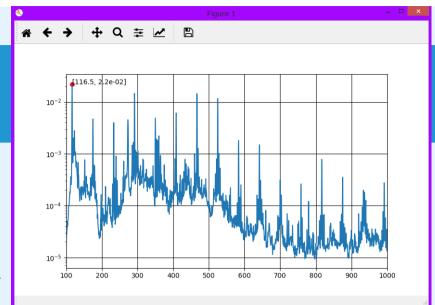
Peak detection

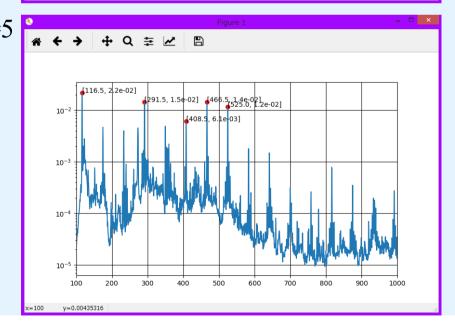
ピークの値が高い上位pp個がプロット 周波数とピークの値も出力される

pp=1









Peak cut

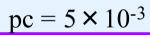
```
7
8 x1 = 100
9 x2 = 1000 #周波数範囲
10 odr = 10 #order (とりあえずデフォルト10)
11 pp = 5 #peak point (どーク何個とるか)
12 pc = 1e-03 #peak cut (これより小さいビークをカット)
13 path = 'data/acc-BS-1.dat'
14
15
16 import numpy
17 d = numpy.loadtxt(path)
18
19
20 import fujimopy01_4 as fm
21 #fm.peak(x1, x2, d[:,0],d[:,1], odr, pp, xlog=False) # <-- True or False
22 fm.pcut(x1, x2, d[:,0],d[:,1], odr, pc, xlog=True)
23
```

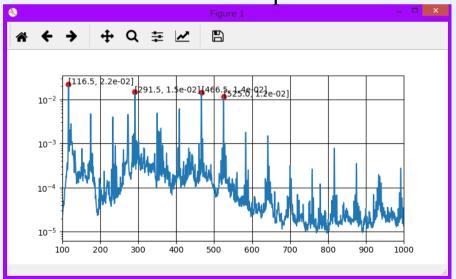
メソッド「pcut」を使う パラメータはpeakとほぼ同じ ppの代わりにpcを使う pcより大きい値のピークだけプロットされる

この機能をpeakの中に入れるかどうか

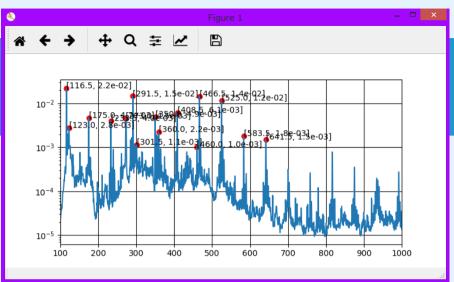
Peak cut

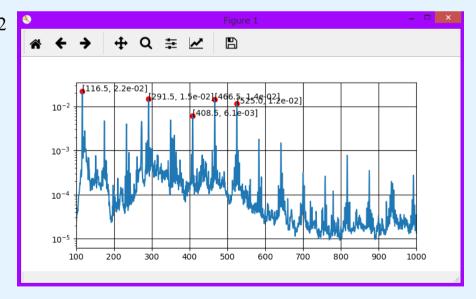
$$pc = 10^{-3}$$



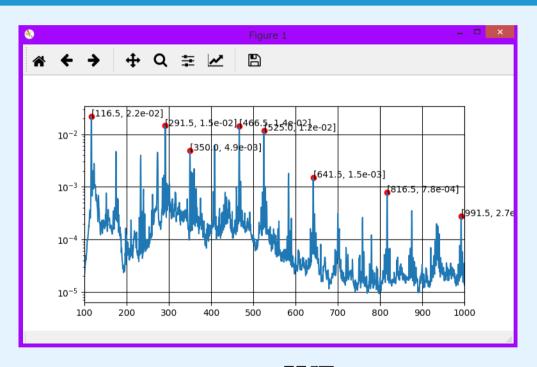


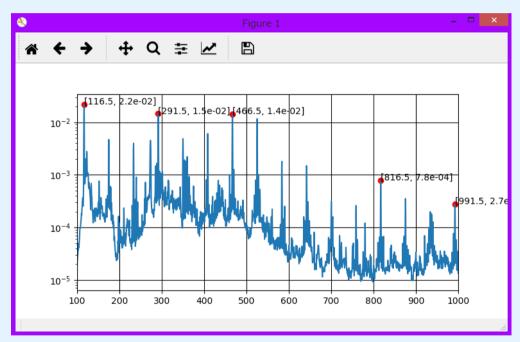
$$pc = 10^{-2}$$





Frequency cut





100 Hz 間隔

300 Hz 間隔

About peak detection

scipyのargrelmaxを使用→ ピークを探しだし、 何番目のデータかを返す

```
7
8 def peak(x1, x2, x, y, odr, pp, xlog):
9
10
11
19
20 #ビークのインデックスを取得
21 from scipy import signal
22 maxid = signal.argrelmax(y, order=odr) #order:最大ビーク値のorder倍以下のビークを排除
23
```

```
if((int(order) != order) or (order < 1)):
    raise ValueError('Order must be an int >= 1')

datalen = data.shape[axis]
locs = np.arange(0, datalen)

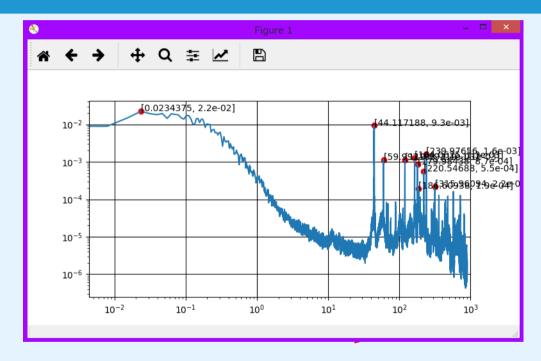
results = np.ones(data.shape, dtype=bool)
main = data.take(locs, axis=axis, mode=mode)
for shift in xrange(1, order + 1):
    plus = data.take(locs + shift, axis=axis, mode=mode)
    minus = data.take(locs - shift, axis=axis, mode=mode)
    results &= comparator(main, plus)
    results &= comparator(main, minus)
    if(~results.any()):
        return results

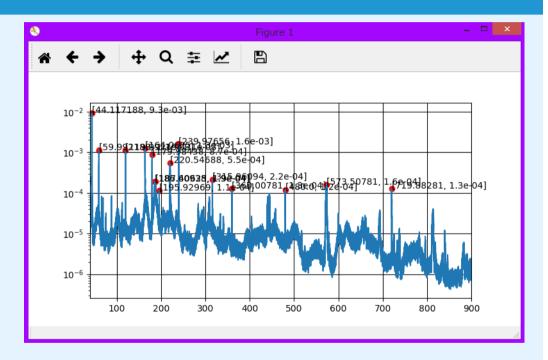
return results
```

←scipyのライブラリの中の argrelmaxに関する部分

まだ解読できてないので、 実際にどのようにピークを 検出しているかは不明

ASD of portable acc





$$odr = 100$$
$$pp = 10$$

横澤さんからもらった txtファイルでも使えた

$$90 \sim 900 \text{ Hz}$$

odr = 100
pp = 15

Summary

- -スペクトルのピークを検出するプログラムfujimopyを作成した
- -ピーク検出原理は不明だが、欲しい機能は作れた
- データファイルがdatファイルにしか対応していないので、他のデータファイルにも対応させる
- 周波数ごとにピークを出したい
- -汎用性が高い&みんなが読めるコードにする
- -追加してほしい機能や、改善点があれば、ぜひお願いします

Code of fujimopy 1 4

```
7
8 x1 = 100
9 x2 = 1000 # 湯波敷範囲
10 odr = 10 #order (どあえずデフオルト10)
11 pp = 5 #peak point (ど一ク何個とるか)
12 pc = 5e-03 #peak cut (これより小さいビークをカット)
13 path = 'data/acc-BS-1.dat'
14
15
16 import numpy
17 d = numpy.loadtxt(path)
18
19
20 import fujimopy01_4 as fm
21 #fm.peak(x1, x2, d[:,0], d[:,1], odr, pp, xlog=False) # <-- True or False
22 fm.pcut(x1, x2, d[:,0], d[:,1], odr, pc, xlog=False)
23
24
25
26
27
```

```
8 def peak(x1, x2, x, y, odr, pp, xlog):
     for j in range(len(x)):
         if x[j]>x1:
     for k in range(len(x)):
         if x[k]>x2:
     x=x[j-1:k]
     y=y[j-1:k]
     from scipy import signal
     maxid = signal.argrelmax(y, order=odr) #order:最大ビーク値のorder倍以下のビークを排除
     if len(maxid[0]) < pp:</pre>
         ppp = len(maxid[0]) #指定するどークの数が、検出されるどークの数より多くならないように
         ppp=pp
     for j in range(ppp):
         tmp = maxid[0][j]
         t=j
         for i in range(len(maxid[0])-j-1):
             if y[tmp] < y[maxid[0][j+1+i]]:</pre>
                 tmp = maxid[0][j+1+i]
                 t=j+1+i
         maxid[0][t] = maxid[0][j]
         maxid[0][j] = tmp
     import matplotlib.pyplot as plt
     plt.plot(x, y)
     plt.yscale("log")
     if xlog==True:
         plt.xscale("log")
     plt.xlim(x1,x2)
     plt.grid(which='major',color='black',linestyle='-')
     cnt = 0
     for i in range(pp):
         plt.scatter(x[maxid[0][i]], y[maxid[0][i]], c='red')
         plt.text(x[maxid[0][i]],y[maxid[0][i]], '['+str(x[maxid[0][i]])+', '+'{:.1e}'.format(y[maxid[0][i]])+']')
         cnt += 1
     plt.show()
```

```
60 def pcut(x1, x2, x, y, odr, pc, xlog):
      for j in range(len(x)):
          if x[j]>x1:
      for k in range(len(x)):
          if x[k]>x2:
      x=x[j-1:k]
      y=y[j-1:k]
      from scipy import signal
      maxid = signal.argrelmax(y, order=odr) #order:最大ビーク値のorder倍以下のビークを排除
      a=[]
      for i in range(len(maxid[0])):
          if y[maxid[0][i]] > pc:
              a.append(maxid[0][i])
      import matplotlib.pyplot as plt
      plt.plot(x, y)
      plt.yscale("log")
      if xlog==True:
          plt.xscale("log")
      plt.xlim(x1,x2)
      plt.grid(which='major',color='black',linestyle='-')
      cnt = 0
      for i in range(len(a)):
          plt.scatter(x[a[i]], y[a[i]], c='red')
          plt.text(x[a[i]],y[a[i]], '['+str(x[a[i]])+', '+'{:.1e}'.format(y[a[i]])+']')
          cnt += 1
      plt.show()
      print(x1, x2, odr, i+1, cnt)
```

```
test py 🔣
                                                   fujimopyO1_4py 🔣
                                                                                                               fujimopyO1_2py 🗵
                                                                                                                                     fujimopyO1_5py 🔀 🚺
                   gazo_shoripy 🔣
                                       fрору 🗵
                                                                         fujimopyO1_4jpy 🔀
                                                                                               triming py 🔣
9 x2 = 1000
10 odr = 80
13 path = 'data/acc-BS-1.dat'
16 import numpy as np
17 import matplotlib.pyplot as plt
18 from scipy import signal
20 data = np.loadtxt(path)
21 x = data[:,0]
22 y = data[:,1]
25 for j in range(len(x)):
26    if x[j]>x1:
      if x[k]>x2:
32 x=x[j-1:k]
33 y=y[j-1:k]
35 plt.plot(x, y)
36 plt.yscale("log")
38 plt.xlim(x1,x2)
40 for i in range(1000):
43 ff=i #1秒に何個データあるか
45 for i in range((x2-x1)//hh):
       yy=y[i*hh*ff:(i*hh+hh)*ff-1]
        xx=x[i*hh*ff:(i*hh+hh)*ff-1]
       maxid = signal.argrelmax(yy, order=odr) #order:最大どーク値のorder倍以下のビークを排除
       for j in range(len(maxid[0])):
    plt.scatter(xx[maxid[0][j]], yy[maxid[0][j]], c='red')
    plt.text(xx[maxid[0][j]],yy[maxid[0][j]], '['+str(xx[maxid[0][j]])+', '+'{:.1e}'.format(yy[maxid[0][j]])+']')
55 plt.show()
56 print(x1, x2, odr)
```