2. Setting up (3 points):
    a. Take a screenshot listing the information on your cluster from your EC2 instance (2 points):

```
[ec2-user@ip-172-31-47-12 ~]$ pcluster describe-cluster-instances --region
ap-southeast-1 --cluster-name MyCluster01
{
  "instances": [
    {
      "launchTime": "2021-11-06T06:32:36.000Z",
      "instanceId": "i-025a990e6e33d214f",
      "publicIpAddress": "13.250.111.118",
      "instanceType": "t3a.large",
      "state": "running",
      "nodeType": "HeadNode",
      "privateIpAddress": "10.0.4.71"
    },
    {
      "launchTime": "2021-11-06T07:42:59.000Z",
      "instanceId": "i-080052078a6c97b6d",
      "queueName": "queue1",
      "publicIpAddress": "18.141.8.62",
      "instanceType": "r5a.2xlarge",
      "state": "running",
      "nodeType": "ComputeNode",
      "privateIpAddress": "10.0.5.50"
    }
  ]
}
```

    [Subtract 1 point if there is more than 1 compute node, if there are no compute nodes, or if the compute node is not at least 2xlarge]

    b. Show the queue (1 point):

```
ec2-user@ip-10-0-4-206 session01]$ date
Mon Nov  8 16:34:29 UTC 2021
[ec2-user@ip-10-0-4-206 session01]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES
NODELIST(REASON)
[ec2-user@ip-10-0-4-206 session01]$ date
Mon Nov  8 16:34:35 UTC 2021
```

4. For 20181016 (15 points)
    a. rs1b-slurm.sh (5 points):

```
(env1) [ec2-user@ip-10-0-4-71 PyHipp]$ cat rs1b-slurm.sh
#!/bin/bash

# Submit this script with: sbatch <this-filename>

#SBATCH --time=24:00:00   # walltime
```

```
#SBATCH --ntasks=1   # number of processor cores (i.e. tasks)
#SBATCH --nodes=1   # number of nodes
#SBATCH --cpus-per-task=5  # number of processors per task
#SBATCH -J "rs1b"   # job name

## /SBATCH -p general # partition (queue)
#SBATCH -o rs1b-slurm.%N.%j.out # STDOUT
#SBATCH -e rs1b-slurm.%N.%j.err # STDERR

# LOAD MODULES, INSERT CODE, AND RUN YOUR PROGRAMS HERE
python -u -c "import PyHipp as pyh; \
import DataProcessingTools as DPT; \
DPT.objects.processDirs(dirs=None, objtype=pyh.RPLSplit,
channel=[*range(1,33)], SkipHPC=False, HPCScriptsDir='/data/src/PyHipp/',
SkipLFP=False, SkipHighPass=False);"

aws sns publish --topic-arn
arn:aws:sns:ap-southeast-1:018084650241:awsnotify --message
"RPLS1bJobDone"
```

[SNS Notification message can be something else that makes sense.]

b. rplhighpass-slurm.sh (4 points):

```
(env1) [ec2-user@ip-10-0-4-71 PyHipp]$ cat rplhighpass-slurm.sh
#!/bin/bash

# Submit this script with: sbatch <this-filename>

#SBATCH --time=24:00:00   # walltime
#SBATCH --ntasks=1   # number of processor cores (i.e. tasks)
#SBATCH --nodes=1   # number of nodes
#SBATCH --cpus-per-task=1  # number of processors per task
#SBATCH -J "rplhp"   # job name

## /SBATCH -p general # partition (queue)
#SBATCH -o rplhp-slurm.%N.%j.out # STDOUT
#SBATCH -e rplhp-slurm.%N.%j.err # STDERR

# LOAD MODULES, INSERT CODE, AND RUN YOUR PROGRAMS HERE
python -u -c "import PyHipp as pyh; \
import time; \
pyh.RPLHighPass(saveLevel=1); \
print(time.localtime());"
```

c. Job submission (2 points):

```
(env1) [ec2-user@ip-10-0-4-71 20181016]$ sbatch rs1b-slurm.sh
```

[Subtract 1 point if submitted from any other directory other than /quiz/picasso/20181016 or /quiz/picasso/20181016/session01.]

d. After job completes (4 points):

i. Check files (2 points):

```
(env1) [ec2-user@ip-10-0-4-71 20181016]$ find . -name
"rplhighpass*hkl" -or -name "rpllfp*hkl" -or -name "rplraw*hkl" |
wc -l
78
```

[26 channels x 3 = 78 files. Do not subtract points if counts were performed separately. Do not subtract points if checkfiles.sh or checkfiles2.sh is used.]

ii. Email received when job completed (2 points):

[Check time received and message to make sure it matches with the slurm script.]

5. For 20181017 (15 points)

a. Generate list of channels in array01 (3 points):

```
(env1) [ec2-user@ip-10-0-4-71 20181017]$ find session01/array01 -name
"channel*" | sort > chs.txt
```

b. Find channels with rpllfp (2 points):

```
(env1) [ec2-user@ip-10-0-4-71 20181017]$ find session01/array01 -name
"rpllfp*hkl" | sort > lfp.txt
```

c. Find channels with rplhighpass (2 points):

```
(env1) [ec2-user@ip-10-0-4-71 20181017]$ find session01/array01 -name
"rplhigh*hkl" | sort > hp.txt
```

d. Submit rpllfp-slurm.sh for channels missing rpllfp (2 points):

```
(env1) [ec2-user@ip-10-0-4-71 20181017]$ cwd=`pwd`; for i in `comm -23
chs.txt lfp.txt`; do echo $i; cd $i; sbatch
/data/src/PyHipp/rpllfp-slurm.sh; cd $cwd; done
```

e. Submit rplhighpass-slurm.sh for channels missing rplhighpass (2 points):

```
(env1) [ec2-user@ip-10-0-4-71 20181017]$ cwd=`pwd`; for i in `comm -23
chs.txt hp.txt`; do echo $i; cd $i; sbatch
/data/src/PyHipp/rplhighpass-slurm.sh; cd $cwd; done
```

f. Count channels with rpllfp (2 points):

```
(env1) [ec2-user@ip-10-0-4-71 20181017]$ find session01/array01 -name
"rpllfp*hkl" | wc -l
26
```

g. Count channels with rplhighpass (2 points):

```
(env1) [ec2-user@ip-10-0-4-71 20181017]$ find session01/array01 -name
"rplhigh*hkl" | wc -l
26
```

6. Generate cumulative FreqSpectrum objects (15 points):
   a. Create fsa1-slurm.sh (7 points):

```
#!/bin/bash

# Submit this script with: sbatch <this-filename>

#SBATCH --time=24:00:00   # walltime
#SBATCH --ntasks=1   # number of processor cores (i.e. tasks)
#SBATCH --nodes=1   # number of nodes
#SBATCH --cpus-per-task=1  # number of processors per task
#SBATCH -J "fsa1"   # job name

## /SBATCH -p general # partition (queue)
#SBATCH -o fsa1-slurm.%N.%j.out # STDOUT
#SBATCH -e fsa1-slurm.%N.%j.err # STDERR

# LOAD MODULES, INSERT CODE, AND RUN YOUR PROGRAMS HERE
python -u -c "import PyHipp as pyh; \
import DataProcessingTools as DPT; \
import time; \
lfall = DPT.objects.processDirs(dirs=None, exclude=['*array02*',
'*array03*', '*array04*'], objtype=pyh.FreqSpectrum); \
lfall.save(); \
hfall = DPT.objects.processDirs(dirs=None, exclude=['*array02*',
'*array03*', '*array04*'], objtype=pyh.FreqSpectrum, loadHighPass=True,
pointsPerWindow=3000); \
hfall.save(); \
print(time.localtime());"

aws sns publish --topic-arn
arn:aws:sns:ap-southeast-1:018084650241:awsnotify --message "FSA1JobDone"
```

   b. Create consol_fsa1jobs.sh (2 points):

```
#!/bin/sh

temp1=($(squeue))
```

```
cmd1="sbatch --dependency=afterok:"

counter1=0
for i in "${temp1[@]}"; do
       if [[ "$i" == "queue1" ]]; then
              id1=${temp1[$counter1-1]}
              cmd1="${cmd1}${id1}:"
       fi
       counter1=$((counter1+1))
done

cmd1=${cmd1::-1}
cmd1="${cmd1} /data/src/PyHipp/fsa1-slurm.sh"

echo $cmd1
eval $cmd1
```

c. Generate freqspectrum for array01 for both days (2 points):

```
(env1) [ec2-user@ip-10-0-5-43 picasso]$ cwd=`pwd`; for i in `find .
-name "channel*" | grep array01 | sort`; do echo $i; cd $i; sbatch
/data/src/PyHipp/freq-slurm.sh; cd $cwd; done
```

d. We can now submit consol_fsa1jobs.sh by doing (2 points):

```
(env1) [ec2-user@ip-10-0-5-43 picasso]$ bash
/data/src/PyHipp/consol_fsa1jobs.sh
```

e. Check size of files (2 points):

```
(env1) [ec2-user@ip-10-0-5-43 picasso]$ ls -l freqspectrum*hkl
total 4024
-rw-rw-r-- 1 ec2-user ec2-user 2690724 Nov  7 11:03
freqspectrum_660e.hkl
-rw-rw-r-- 1 ec2-user ec2-user 1412772 Nov  7 11:03
freqspectrum_9c80.hkl
```

7. Create interactive plots (12 points):
   a. Copy the saved objects (freqspectrum_9c80.hkl and freqspectrum_660e.hkl in the /data/picasso directory) to your computer (2 points):

```
(aws) $ scp -i ~/MyKeyPair.pem
"ec2-user@54.251.74.44:/data/picasso/freq*hkl" .
```

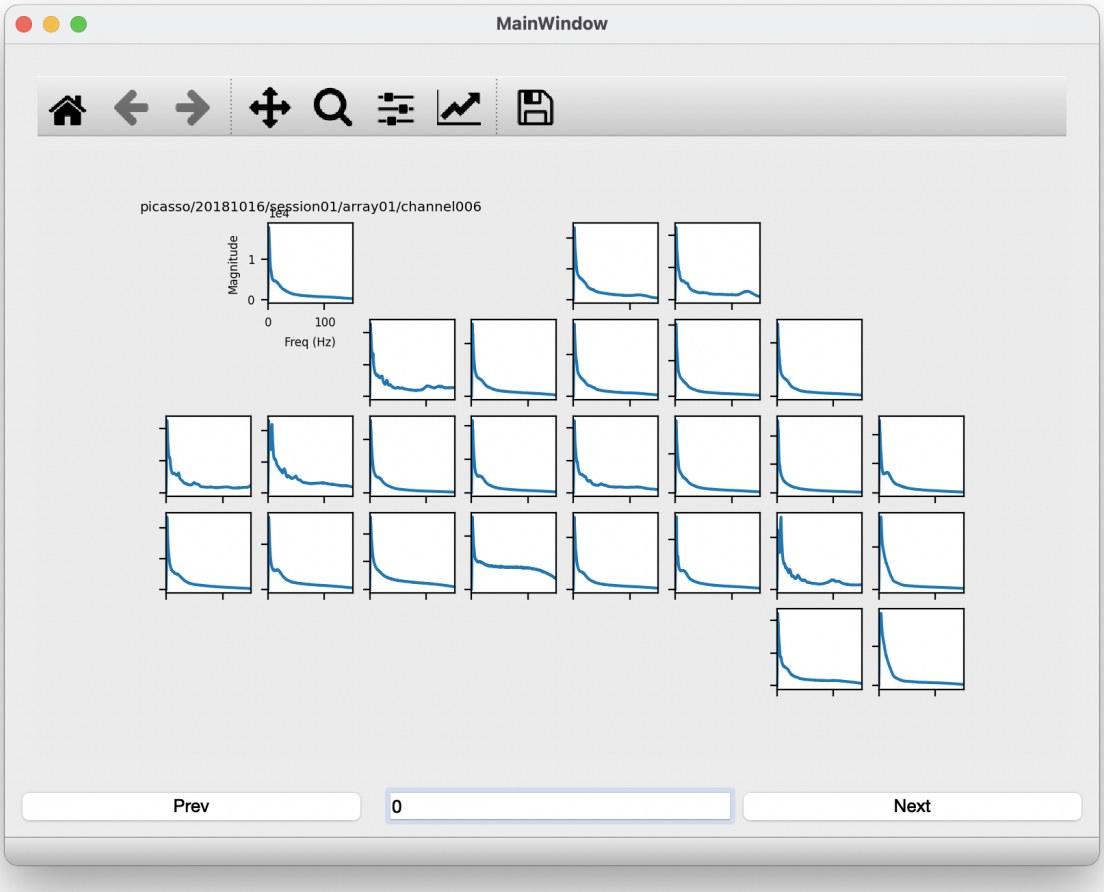
   b. Check the file sizes (2 points):

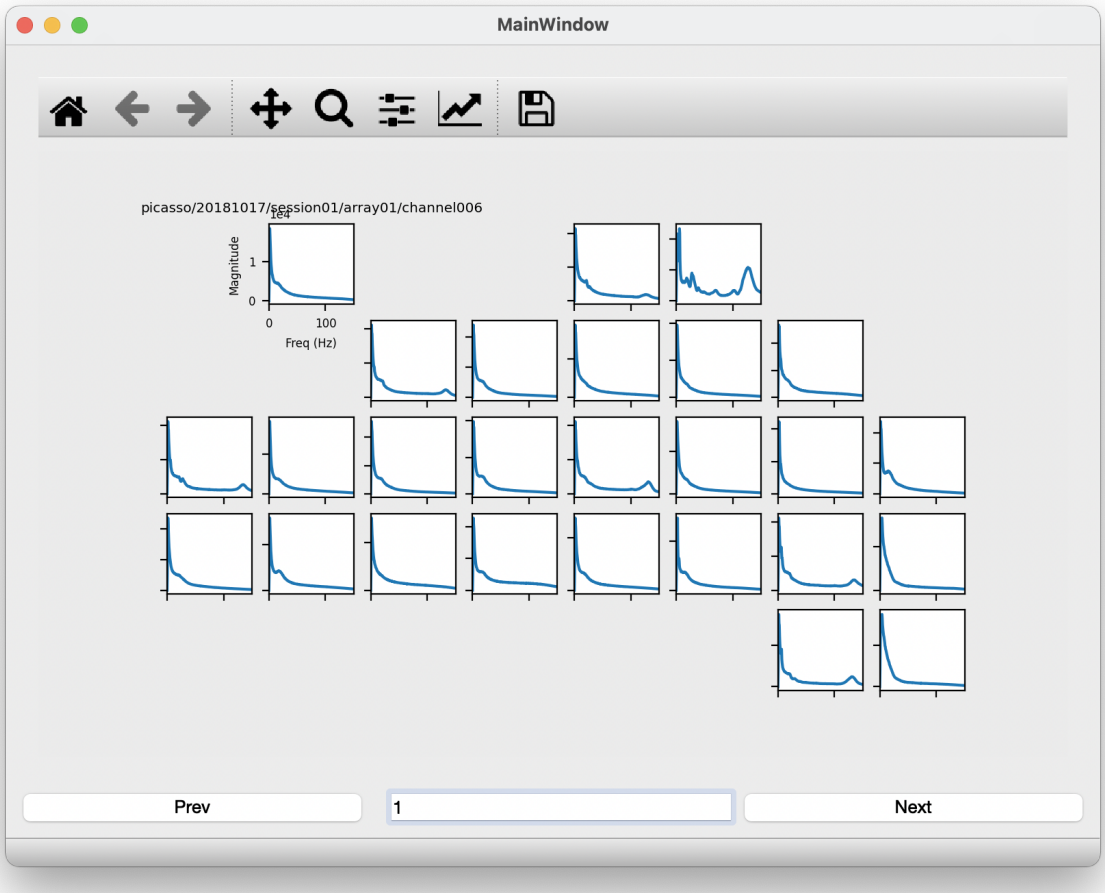
```
(aws) $ ls -l freq*hkl
total 8016
```

```
-rw-r--r--@ 1 shihcheng  staff  2690724 Nov  7 19:12
freqspectrum_660e.hkl
-rw-r--r--@ 1 shihcheng  staff  1412772 Nov  7 19:12
freqspectrum_9c80.hkl
```

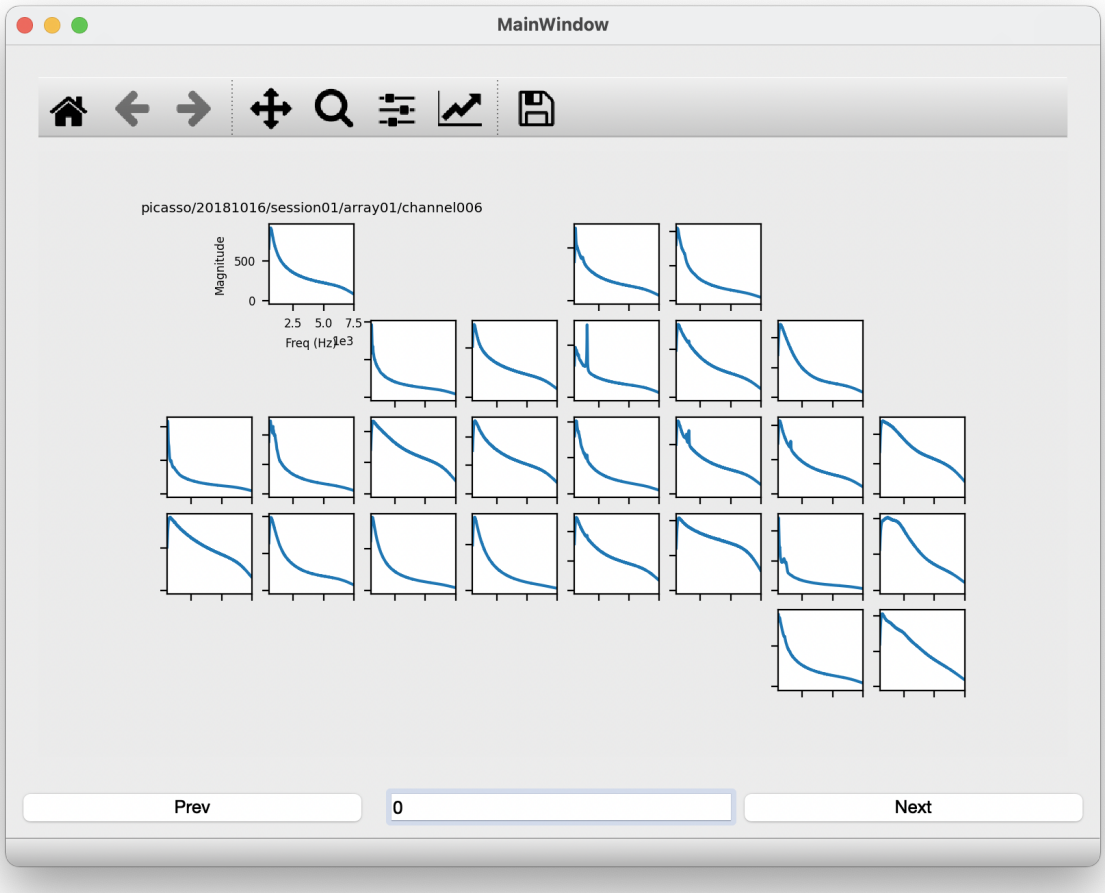
c. Plot freqspectrum for both days (8 points):

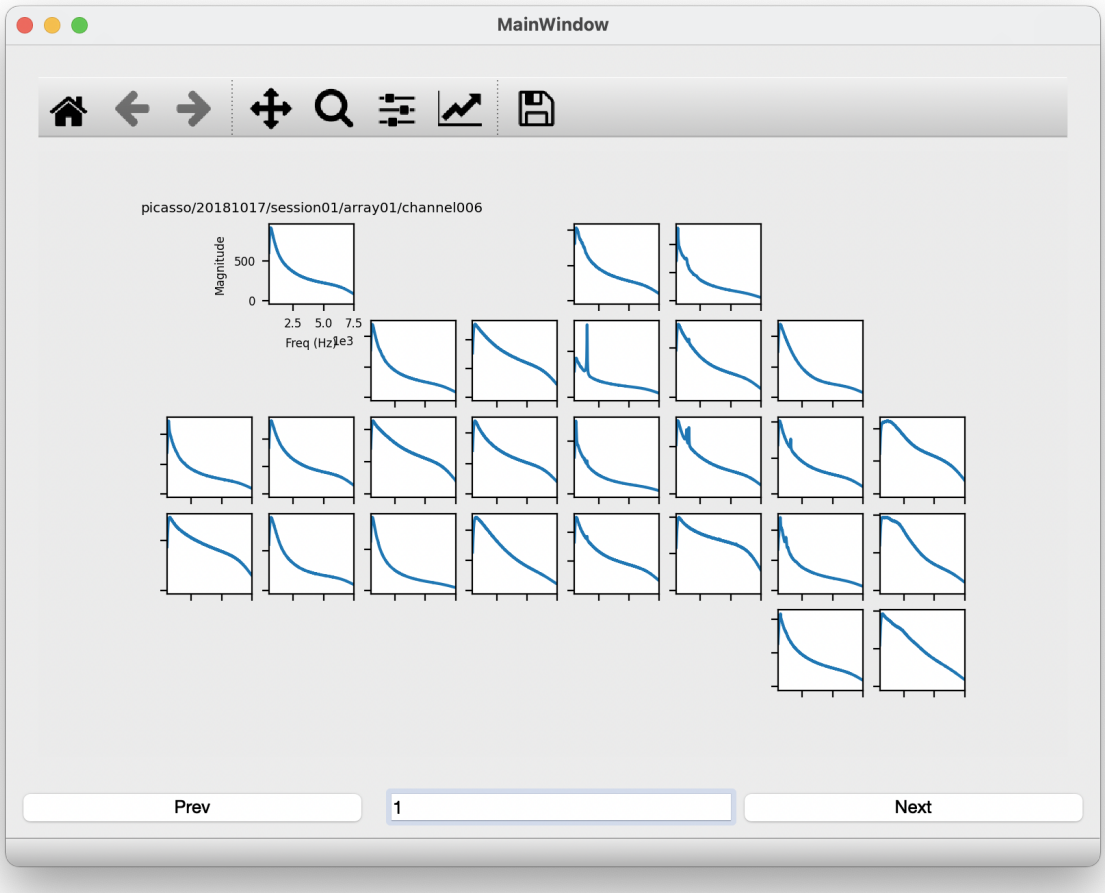
i. Low-frequency plots (2 points each):





ii. High-frequency plots (2 points each):


MainWindow
picasso/20181016/session01/array01/channel006
Magnitude
500
0
2.5
5.0
7.5
Freq (Hz)
1e3
Prev
0
Next


MainWindow
picasso/20181017/session01/array01/channel006
Magnitude
500
0
2.5
5.0
7.5
Freq (Hz)
1e3
Prev
1
Next