Earthquake Analysis with SeisMon

Preface

This document is designed to get the user started on generating data from earthquakes using SeisMon. This particular example is grabbing and organizing data from 6 channels during O1. These channels are two seismometers that measure the motion of three translation degrees of freedom X, Y, and Z under the LIGO optic support structure(two Horizontal and one Vertical). **Important Note:** SeisMon is designed to run from the CIT Cluster on the LIGO servers so the user must log onto there in order to have SeisMon work properly. On the use of brackets within this document, they represent extra explanations and not necessarily exact commands to write out.

Remote access to LIGO clusters

LIGO clusters use the method of SSH (Secure SHell) to establish remote connection to the clusters.

SSH on Windows

Since windows does not come with a UNIX-based terminal, we'll have to grab one from the web. This guide has two options for SSH on Windows, cygwin and PuTTY.

Cygwin

Personally, cgywin is probably the best choice for this kind of environment. Its also pretty much the only one I know, so there's that.

1. Read the following webpage cygwin install.
2. Depending on your version of Windows, you will either download the 32bit or 64bit version. To figure out which version of windows you're running please check out this guide provided by Microsoft.
3. install cygwin by running the file you just downloaded.
   1. It'll ask how you want to install, pick download files from the internet.
   2. The next step will ask you where you want to save cygwin, stick it in your documents, just don't leave it on default(there could be problems just leaving it directly in your root c drive).
   3. Next it'll ask you where you want to save these files, just put them in a directory that you know isn't temporary.
   4. Then you'll get a screen asking you to pick a download mirror site, just pick the first one, it should work just fine.
   5. Then it'll download the listing of the site and you'll be taken to a screen with lists of a whole bunch of packages. Don't worry about these for now, as you'll most of these things alone. Just type ssh into the search bar at the top and then when its done searching, click default on the three categories to tell cygwin to install those packages along with the default ones,
   6. then click next towards the bottom right corner of the screen. Cygwin may ask you to install dependencies just say yes or rather click next. Cygwin will then download and install the packages that you chose and then when its finished, it'll ask you if you want to have a desktop shortcut and put it on the start menu, just leave both of those options ticked. Congratulations! You should now have a working cygwin.
4. Open up cygwin using the shortcut on your desktop. It should run without a hitch at this point. now run on the terminal

```
ssh
[Hopefully output from ssh]
```

```
ssh albert.einstein@ssh.ligo.org
[there will be a query that pops up asking you to trust the key of the server,
type y and you won't have to worry about it.
This applies only to the LIGO clusters.]
[It will then ask you to type in your LIGO Password]
[once that's done, you should see options to log in to various ligo clusters,
please type 2 to go to CIT(which is CalTech) and then pcdev 1, 2, or 3.
Don't worry, it tells which letter to type to get to that particular server.]
[Now you should be on the cluster proper.]
```

PuTTY

If you want a simpler to use program, you can download PuTTY which is an SSH and Telnet client.

1. download PuTTY
2. run PuTTY.exe
3. Inside hostname textbox type: albert.einstein@ssh.ligo.org
4. Then click open in the bottom right corner

SSH on Mac OS and Linux

Open up a terminal and then run

```
ssh
[Hopefully output from ssh]
ssh albert.einstein@ssh.ligo.org
[there will be a query that pops up asking you to trust the key of the server,
type y and you won't have to worry about it.
This applies only to the LIGO clusters.]
[It will then ask you to type in your LIGO Password]
[once that's done, you should see options to log in to various ligo clusters,
please type 2 to go to CIT(which is CalTech) and then pcdev 1, 2, or 3.
Don't worry, it tells which letter to type to get to that particular server.]
[Now you should be on a server within the cluster proper.]
```

This is done so that we can have access to all of the data required for SeisMon to run properly.

Getting SeisMon from github

The following code will provide access to seismon which is needed to run the rest of the guide.

```
cd ~
mkdir gitrepo
cd gitrepo
git clone https://github.com/ligovirgo/seismon.git
[git will fetch the latest version of seismon from github]
```

Getting Started

In order to get started, we want to look and see where the files we need are located within SeisMon's directory. SeisMon's directory should be located here.

```
cd ~/gitrepo/seismon
```

Many of the files that we need to run are located within the folders of the seismon directory itself, usually located within the seismon/bin directory. Run the ls command inside of the seismon directory like this

```
ls
[Output of files and directories in seismon]
cd bin
ls
[Output of files and directories in bin]
cd ..
cd input
ls
[Output of files and directories in input]
```

to make note of the folders and files. For this example, the user will want to take note of both the bin directory and the input directory. The next step is to make sure gwpy is sourced before running any of the scripts mentioned in this file. The user can do this by running

```
pip install --user gwpy
```

Once gwpy is ready to go, then we can move onto the next step of generating the list of xml files needed to do the analysis.

First Stage: seismon_traveltimes

seismon_traveltimes is designed to read data from usgs seismic monitering channels. This data is stored as a series of xml files inside of /home/albert.einstein/eventfiles/iris.

seismon_traveltimes has two overall purposes:

1. It reads the data from usgs seismic monitoring channels.
2. It writes the output to a series of xml files located in the eventfiles directory. This contains the actual earthquake data measured by usgs.

In order to get seismon_traveltimes running we have to go to our home directory and make a directory called eventfiles and then inside eventfiles create a directory called iris.

```
cd ~
mkdir eventfiles
cd eventfiles/
mkdir iris
cd ~
```

The next step after this is to cd into the input directory of seismon. Open up the file seismon_params_traveltimes.txt and inside you should find.

```
cd gitrepo/seismon/input
vi seismon_params_traveltimes.txt
```

```
dataLocation /home/mcoughlin/Seismon/ProductClient/data/receiver_storage/origin
publicdataLocation /home/mcoughlin/Seismon/publicdata
databasedataLocation /home/mcoughlin/Seismon/databasedata
**eventfilesLocation /home/eric.coughlin/eventfiles** ->
**eventfileslocation /home/albert.einstein/eventfiles**
```

If you look at the fourth line, which I bolded for clarity, you'll want to change eric.coughlin to your own albert.einstein directory as long as you followed the above steps correctly. The next step is to cd back to bin then.

```
cd ..
cd bin
screen
python seismon_traveltimes -p /home/albert.einstein/gitrepo/seismon/input/
seismon_params_traveltimes.txt -s 1126569617 -e 1136649617
--minMagnitude 4.0 --doIRIS [still on the same line]
```

`-p`	this is the location of the parameters file
`-s`	this is the gps start time of the program
`-e`	this is the gps end time of the program, this also completes the range of time between start and end
`--minMagnitude`	This defines the minimum magnitude of the earthquakes grabbed by seismon_traveltimes
`--doIRIS`	This tells seismon_traveltimes to grab data from the Incorporated Research Institutions for Seismology(IRIS)'s seismic moniter database

Screen is a program designed to use multiple windows within one terminal session. These screens will continue to operate even if you disconnect from the session. In order to get back to your regular session, just detach from the process by clicking ctrl + a and then d on your keyboard. If you want to reatach just use the following commands.

```
screen -ls
[insert output of screen -ls here]
screen -r [Whatever process you want to reattach]
```

Just copy and paste whichever screen you want to go to from the output of screen -ls after the screen -r command.

This process will take quite a bit of time to complete, think days instead of hours. This is why using screen is a strong recommendation.

Second Stage: seismon_run_run_H1O1 and seismon_run_run_L1O1

These scripts grab the earthquake data from the eventfiles database specifically inside the iris folder and then looks at specific channels in order to get user friendly data output.

After completing the first stage, the next step is to run both H1O1 and L1O1.

The first thing to do in order to run both of these scripts is to

```
cd ~/gitrepo/seismon/input
vi seismon_params_H1O1.txt
```

Inside you'll find a file that looks like this.

```
ifo H1
frameType H1_R
```

```
runName H1O1
user eric.coughlin
dirPath /home/eric.coughlin/gitrepo
publicPath /home/eric.coughlin/public_html
codePath /home/eric.coughlin/gitrepo
executableDir /home/eric.coughlin/gitrepo/seismon/bin
eventfilesLocation /home/eric.coughlin/eventfiles
#eventfilesLocation /home/mcoughlin/Seismon/eventfiles/database
velocitymapsLocation /home/mcoughlin/Seismon/velocity_maps
```

You'll want to change the eric.coughlin directories to your own albert.einstein, don't touch the mcoughlin directories.

A nice way to do that is to use within vim

```
:%s/eric.coughlin/albert.einstein/gc
:wq
```

It'll ask you to confirm each change made.

The next step is to

```
cd ~/gitrepo/seismon/bin
vi seismon_run_run_H1O1
```

Inside you'll find this line

Change the eric.coughlin to albert.einstein

Now do the same steps with L1O1.

If you'd like to change the parameters for this script to look at different channels, you'll want to go to seismon_run_run_H1O1. Inside you'll find these two lines

```
os.system("python seismon_run -p %s -s %d -e %d -c H1:ISI-GND_STS_HAM2_Z_DQ,
H1:ISI-GND_STS_HAM2_Y_DQ,
H1:ISI-GND_STS_HAM2_X_DQ,H1:ISI-GND_STS_HAM5_Z_BLRMS_30M_100M,
H1:ISI-GND_STS_HAM5_Y_BLRMS_30M_100M,
H1:ISI-GND_STS_HAM5_X_BLRMS_30M_100M --doEarthquakes --doEarthquakesAnalysis
 --doPSD --eventfilesType iris --minMagnitude 4.0"%(paramsFile,gpsStart,gpsEnd))

print "python seismon_run -p %s -s %d -e %d -c H1:ISI-GND_STS_HAM2_Z_DQ,
H1:ISI-GND_STS_HAM2_Y_DQ
,H1:ISI-GND_STS_HAM2_X_DQ,H1:ISI-GND_STS_HAM5_Z_BLRMS_30M_100M
,H1:ISI-GND_STS_HAM5_Y_BLRMS_30M_100M
,H1:ISI-GND_STS_HAM5_X_BLRMS_30M_100M --doEarthquakes --doEarthquakesAnalysis
--doPSD --eventfilesType iris --minMagnitude 4.0"%(paramsFile,gpsStart,gpsEnd)
```

`-p`	This is the location of the parameters file
`-s`	This is the gps start time
`-e`	This is the gps end time
`-c`	These are the LIGO channels that you would like to look at for LHO
`--doEarthquakes`	This looks for the earthquake events and gets their information
`--doEarthquakesAnalysis`	This analysizes the earthquakes

`--doPSD` This looks at the Particle Size Distribution?

`--eventfilesType` This determines the database that is used, only option in this guide is iris

`--minMagnitude` This determines the minimum magnitude of the earthquakes looked at, only goes as low as the database generated from the previous script

Don't worry about the %s and %d's

%s

String formater for Python, replaces %s with variable defined by user

%d

decimal replacer for Python, %d with a variable defined by user

Once you are done, you should use screen again to run both seismon_run_run_H1O1 and seismon_run_run_L1O1.

```
cd ~/gitrepo/seismon/bin
screen
python seismon_run_run_H1O1
[on keyboard press ctrl-a then d]
screen
python seismon_run_run_H1O1
```

This will also take some time.

The output will be found in /home/albert.einstein/gitrepo/, within these are a series of directories and files that encompass the output from the two scripts.

Third Stage: seismon_run_prediction_vs_actual_ec

seismon_run_prediction_vs_actual_ec is designed to compare the predicted measurements and the actual measurements to create a nice succinct text file for each channel. For more information check out this document.

```
vi seismon_run_prediction_vs_actual_ec
:%s/eric.coughlin/albert.einstein/gc
:wq
screen
python seismon_run_prediction_vs_actual_ec
[ctrl-a then d]
```

If you've adjusted the channels then you'll need to make the proper changes to seismon_run_prediction_vs_actual_ec

```
inputFileDirectory="/home/eric.coughlin/gitrepo/Text_Files
/Timeseries/H1_ISI-GND_STS_HAM2_Z_DQ/64/"
#accelerationFileDirectory="/home/eric.coughlin/gitrepo/Text_Files
/Acceleration/H1_ISI-GND_STS_HAM2_Z_DQ/64/"
#predictionFile="/home/eric.coughlin/gitrepo/H1/H1O1
/1126569617-1136678417/earthquakes/earthquakes.txt"
predictionFile="/home/eric.coughlin/gitrepo/H1/H1O1
/1126073342-1137283217/earthquakes/earthquakes.txt"
outputDirectory="/home/eric.coughlin/gitrepo/Predictions/H1O1/"
```

1. You will want to change the inputFileDirectory to the channel names that you looked at.
2. change predictionFile to the time range that you looked at.
3. change outputDirectory from H1O1 to the channel names that you looked at
4. repeat for all of the channels.

```
filenames = ["/home/eric.coughlin/gitrepo/Predictions/H1O1/earthquakes.txt",
"/home/eric.coughlin/gitrepo/Predictions/L1O1/earthquakes.txt"]
```

1. add channel directories to these filenames keeping the same format but just changing H1O1 to the channel name

The output directory will be in /home/albert.einstein/gitrepo/Predictions/*