Getting Earthquake Analysis

Preface

This is a document designed to get the user started on generating data from earthquakes using SeisMon. This particular example is designed to grab and organize 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 servers on LIGO so the user should log onto there in order to make sure that things will even begin to 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. 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. It'll ask how you want to install, pick download files from the internet. The next step will ask you where you want to save cygwin, stick in your documents, don't leave it on default. Next it'll ask you where you want to save these files, just put them in a directory that you know isn't temporary. Then you'll get a screen asking you to pick a download mirror site, just pick the first one, it should work just fine. 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, 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 ar
[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 typ
[Now you should be on the cluster proper.]
```

SSH on Mac OS and Linux

Open up a terminal and then run

```
ssh albert.einstein@ssh.ligo.org
[type in password]
[CIT]
[One of the servers: pcdev-1, 2, or 3]
```

Getting SeisMon from github

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

```
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
```

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 take data from usgs seismic monitering channels and write this data to xml files located within the eventfiles database.

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.

```
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/albe
```

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

```
screen
python seismon_traveltimes -p /home/$USER/gitrepo/seismon/input/seismon_params_traveltim
```

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 reatach]
```

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 or albert.einstein directories to your own, 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.

Now do the same steps with L1O1.

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.

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.

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