Instructions for LIGO Line Coherence Tools

Geoffrey Mo, Carleton College
mog@carleton.edu

February 24, 2018

These are instructions for setting up and running the tools necessary for the analysis of spectral artifact (lines) coherence. These tools were largely written by Duo Tao at Carleton College under the supervision of Professor Nelson Christensen.

Setup

LIGO Data Grid Setup

1. Register for a LIGO Data Grid (LDG) account at `https://grouper.ligo.org/ldg/`. These requests are manually approved, so it may take a few days.

2. Install the LDG tools available at `https://www.lsc-group.phys.uwm.edu/lscdatagrid/doc/installclient.html`.

3. In the command line, type `ligo-proxy-init albert.einstein` to get a certificate for the grid. Remember to replace `albert.einstein` with your LSC username. This "logs you in." The certificate will last about 11 days before you need to request another one.

4. You are now ready to use the LIGO Data Grid!

Line Coherence Tools Setup

1. First, you'll need to login to an LDG server. For this example, I'll use the Livingston server (LA) located at `ldas-pcdev1.ligo-la.caltech.edu`.[1] To ssh to the server, type `gsissh ldas-pcdev1.ligo-la.caltech.edu`.

2. To get Duo's tools onto your account on the server, use `git clone http://github.com/taoduo/researchscripts`. This will copy Duo's programs from his GitHub account to the home directory of your account on the LA server.

3. While we are in the home directory, we will make one more directory which will become useful. Use `mkdir public_html` to make these two directories.

4. Navigate to the `ResearchScripts/LineSearch/single_line` directory. You will find a variety of Matlab programs, but we will need a `main.m` program to run the line-finding script. Use `touch main.m` to create an empty main.m file.

[1] The Hanford (HA) server is `ldas-pcdev1.ligo-wa.caltech.edu` and the Caltech server is `ldas-grid.ligo.caltech.edu`.

5. We will use the vim text editor to edit `main.m` using `vim main.m`. In it, write the following code:

```
data01 = '/home/eric.coughlin/public_html/O1/LineSearch';
data02 = '/home/mcoughlin/public_html/O2/LineSearch';
output_path = '/home/albert.einstein/output_buffer';

% defaults
zoom = 1;
filter = -30;
resolution = 0.001;
run = 'O2';
observatory = 'L1';

% for the search
search1 = Search(zoom, filter);
% for the lines
lines = [35.7632];
l1 = line_array(lines, run, observatory, resolution);
% do the search
multiple_line_search(data02, search1, l1, output_path);
```

To quit vim, press `esc`, then type `:wq` (for "write" and "quit").

Remember to change the `albert.einstein` in `output_path` to your username. The `filter` parameter changes how "sensitive" the tool is; since the tool sorts line frequency coherences in a Gaussian distribution, the filter ensures that only lines in the tail of the distribution past $10^{\texttt{filter}}$ show up. Experiment with this if your results do not seem complete, or if the script returns too many results. Put the frequencies of the lines you are searching for in the `lines` list.

Operation of the the tool

Running the Matlab program

1. Once your `main.m` file is configured as you would like, type `matlab` to start Matlab. Once Matlab is open, type `main` to run `main.m`. If you would like this process to run in the background so you can close your command line window and step away from the computer, do step 1 in the window opened by the command `screen`.

2. Depending on how many lines the program is searching for, it will run for up to 12 or so hours. The output will be placed in the `output_buffer` directory (which is created by the script) in the home directory. This output comes in the form of plots in the jpg format.

Displaying the results

1. To view these plots, we will create a webpage viewable in a web browser. First, navigate to `/public_html`. Then, use `touch index.html` to create an html file.

2. Use `vim index.html` to open the file in vim. Then, paste in the following code:

```
<!DOCTYPE html>
```

```
<html>
<head>
<link
       href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
       rel="stylesheet"
       integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
       crossorigin="anonymous">
<script
       src="https://code.jquery.com/jquery-3.1.1.min.js"
       integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8="
       crossorigin="anonymous">
</script>
<script
       src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
       integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Txa"
       crossorigin="anonymous">
</script>
<title>Line Search Results with Coherence Tool</title>
</head>
<body>
<header>
<h1>Coherence Search Results</h1>
<hr>
</header>
<div class='container'>
<div class='row'>
<div class='col-md-4'>
<ul>
</ul>
</div>
</div>
</div>
</body>
</html>
```

Remember to type `:wq` to save and exit vim. This code will create an html page at `https://ldas-jobs.ligo-la.caltech.edu/~albert.einstein/`[2] which can be viewed in your web browser.

3. Now, run the Python script which will generate the webpage for the specific lines and data that you have just run in Matlab. Navigate to `ResearchScripts/Tools`, where you will find the `LineExp.py` script.

4. To run this script, type `python LineExp.py /home/albert.einstein/output_buffer`. After a second or two, the output will be a line of HTML code.

5. Copy this line of code and paste it into `index.html` in the `public_html` directory. It should go near the bottom, between `<ul>` and `</ul>`.

6. Now, we need to copy the results of the Matlab script to `public_html` so the website can reference the necessary images. To do this, type `cp -r /home/albert.einstein/output_buffer/* /home/albert.einstein/public_html/`.

[2] ...ldas-jobs.ligo-ha.caltech.edu... for Hanford

7. Now, on your webpage `https://ldas-jobs.ligo-la.caltech.edu/~albert.einstein/`, you should find a link leading to a graphical display of the data, organized by day. Each plot in the day corresponds to a line with a specific coherence to a dataset. To understand which dataset it is, reference the title of the plot with the LIGO abbreviations and acronyms dictionary found here: `https://dcc.ligo.org/public/0002/M080375/012/LIGO-M080375-V12%20(Abbreviations%20And%20Acronyms).pdf`.

Congratulations! You've just mined the data for line coherences!