Ligo Primary Analysis

- Tested on Intel based Linux.
- Notebook issue with it being too large to save with renders. Hence PDF screenshot will be provided. Request to run the notebook for visuals in the jupyter enviornment.
- Notebook may take a while to render. This is partly due to graphic processing, and also due to the fact this code was not performance optimized. Please be patient (we've seen 15-20 minutes) for the full render.
- Installation instructions to notebook on the README
- python -m pip install -r requirements_primary.txt for installing the relevant libs

0. Imports

In [1]:

```
import logging
import h5py
import json
import inspect
import readligo as rl
import matplotlib.mlab as mlab
from scipy.interpolate import interp1d
import matplotlib.pyplot as plt
import os
import numpy as np
from gwosc.locate import get_urls
import requests
import os
from gwosc.locate import get_urls
from scipy import signal
from gwpy.timeseries import TimeSeries

logger = logging.getLogger()
logger.setLevel(logging.INFO)

from rk_visualizer import *
app = RKModelApplication(master=root)
visualizer = RKGraphVisualizer(app)
```

executed in 1.75s, finished 02:13:37 2022-09-27

1. Read LIGO Strain Data & Load for Analysis

In [2]:

```
detector = 'H1'
event_data  = {'GW170729': {'t': 1185389807.3},
               'GW170817': {'t': 1187008882.4},
               'GW190521': {'t': 1242442967.4},
               'GW190814': {'t': 1249852257.0}}

for name, event in event_data.items():
    url = get_urls(detector, event['t'], event['t'])[-1]
    fn = os.path.basename(url)
    n = url.split("/")[-3]
    event_data[n]["path"] = fn
    if not os.path.exists(fn):
        print('Downloading: ' , url)
        with open(fn,'wb') as strainfile:
            straindata = requests.get(url)
            strainfile.write(straindata.content)
    else:
        print("already downloaded {}. skipping".format(fn))

# using the tool
visualizer.load_data()
print(inspect.getsource(visualizer.load_data))
```

executed in 23.9s, finished 02:14:01 2022-09-27

```
already downloaded H-H1_GWOSC_4KHZ_R1-1185387760-4096.hdf5. skipping
already downloaded H-H1_GWOSC_4KHZ_R1-1187006835-4096.hdf5. skipping
already downloaded H-H1_GWOSC_4KHZ_R1-1242440920-4096.hdf5. skipping
already downloaded H-H1_GWOSC_4KHZ_R1-1249850209-4096.hdf5. skipping
Trying to find event for 1185389807.3: https://www.gw-openscience.org/eventapi/json/GWTC-1-confident/GW170729/v1/H-H1_GWOSC_4KHZ_R1-1185387760-4096.hdf5 (https://www.gw-openscience.org/eventapi/json/GWTC-1-confident/GW170729/v1/H-H1_GWOSC_4KHZ_R1-1185387760-4096.hdf5)
GW170729 is already downloaded
Trying to find event for 1187008882.43: https://www.gw-openscience.org/eventapi/json/GWTC-1-confident/GW170817/v3/H-H1_GWOSC_4KHZ_R1-1187006835-4096.hdf5 (https://www.gw-openscience.org/eventapi/json/GWTC-1-confident/GW170817/v3/H-H1_GWOSC_4KHZ_R1-1187006835-4096.hdf5)
GW170817 is already downloaded
Trying to find event for 1242442967.4: https://www.gw-openscience.org/eventapi/json/GWTC-2.1-confident/GW190521/v4/H-H1_GWOSC_4KHZ_R1-1242440920-4096.hdf5 (https://www.gw-openscience.org/eventapi/json/GWTC-2.1-confident/GW190521/v4/H-H1_GWOSC_4KHZ_R1-1242440920-4096.hdf5)
GW190521 is already downloaded
Trying to find event for 1249852257.0: https://www.gw-openscience.org/eventapi/json/GWTC-2.1-confident/GW190814/v3/H-H1_GWOSC_4KHZ_R1-1249850209-4096.hdf5 (https://www.gw-openscience.org/eventapi/json/GWTC-2.1-confident/GW190814/v3/H-H1_GWOSC_4KHZ_R1-1249850209-4096.hdf5)
GW190814 is already downloaded
    def load_data(self):
        for event, data in event_names.items():
            t0 = data['ts']
            detector='H1'
            url = get_urls(detector, t0, t0)[-1]
            print("Trying to find event for {}: {}".format(t0, url))
            fn = os.path.basename(url)
            data['file'] = fn
```

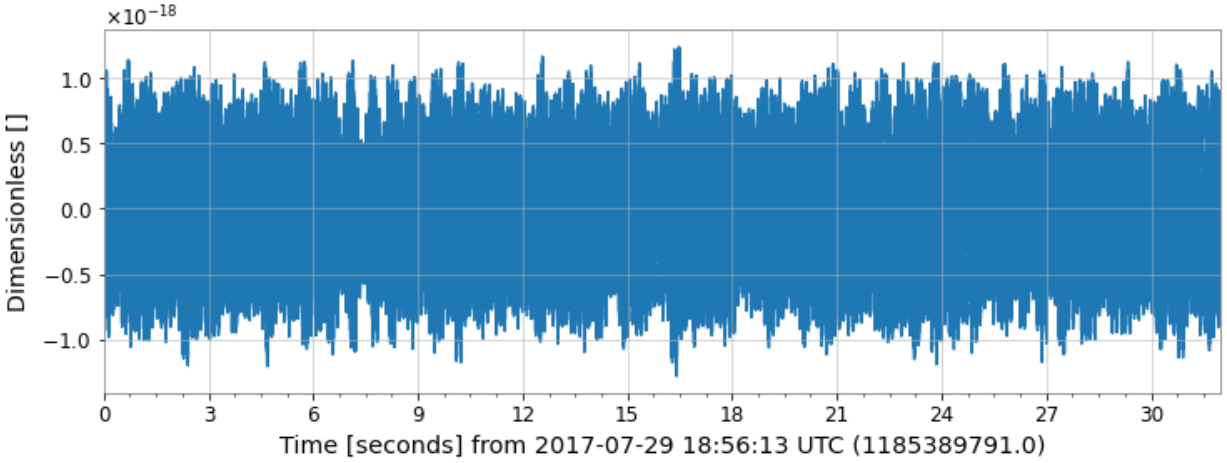
```
if not os.path.exists(fn):
    print('Downloading: ' , url)
    with open(fn,'wb') as strainfile:
        straindata = requests.get(url)
        strainfile.write(straindata.content)
else:
    print("{} is already downloaded".format(event))
```

2. View Strain Data in Time Series

In [3]:

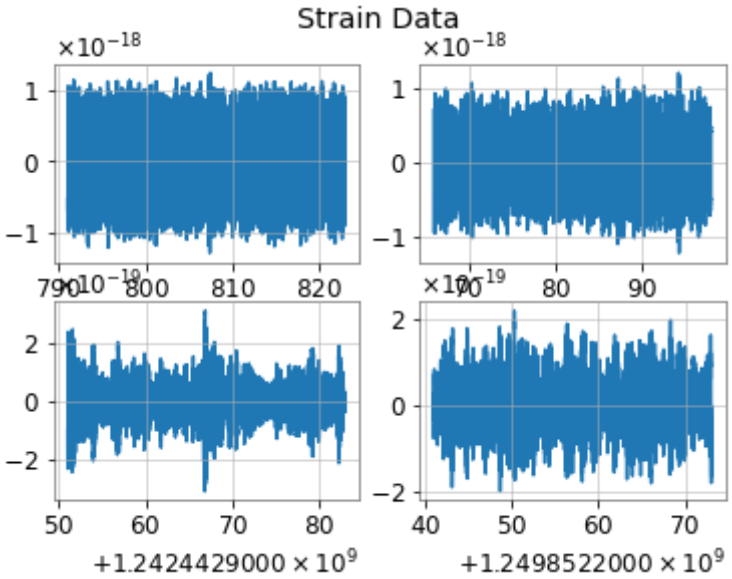
```
1 focus_event = 'GW170729'
2 strain = TimeSeries.read(event_data[focus_event]['path'],format='hdf5
3 center = int(event_data['GW170729']['t'])
4 strain = strain.crop(center-16, center+16)
5 fig1 = strain.plot()
```

executed in 8.64s, finished 02:14:09 2022-09-27



In [4]:
```
1 visualizer.plot_strain()
2 print(inspect.getsource(visualizer.plot_strain))
```

executed in 29.5s, finished 02:14:39 2022-09-27

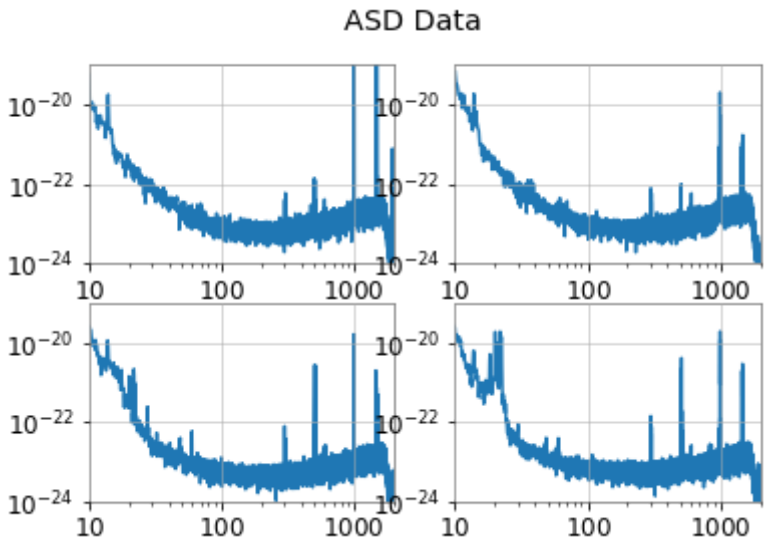



```
def plot_strain(self):
    figure = plt.gcf()
    figure.clf()
    figure.add_subplot(221)
    figure.add_subplot(222)
    figure.add_subplot(223)
    figure.add_subplot(224)
    axes = figure.get_axes()
    count=0
    figure.suptitle("Strain Data")
    for k,f in event_names.items():
        row = math.floor(count/2)
        col = count%2
        fn = f['file']
        if fn == "":
            return
        strain = self.get_strain(fn, f['ts'])
        axes[count].plot(strain)
        count+=1
    plt.show()
```

3. Plot ASD of Whitened & Band-Passed Strain

In [5]:
```
1 visualizer.plot_asd()
2 print(inspect.getsource(visualizer.plot_asd))
```

executed in 29.6s, finished 02:15:08 2022-09-27



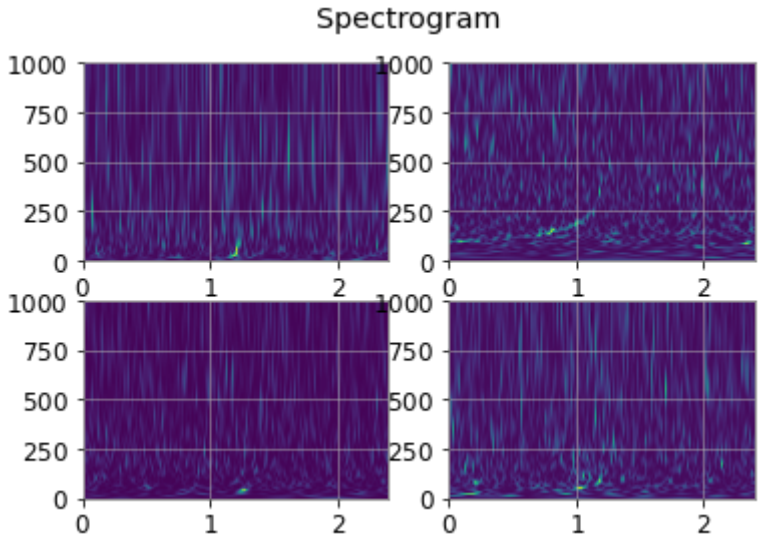

```
def plot_asd(self):
    figure = plt.gcf()
    figure.clf()
    figure.add_subplot(221)
    figure.add_subplot(222)
    figure.add_subplot(223)
    figure.add_subplot(224)
    axes = figure.get_axes()
    count=0
    figure.suptitle("ASD Data")
    for k,f in event_names.items():
        ax = axes[count]

        fn = f['file']
        if fn == "":
            return

        strain = self.get_strain(fn, f['ts'])
        f2 = strain.asd(fftlength=8)
        ax.plot(f2)
        ax.set_xlim(10,2000)
        ax.set_ylim(1e-24, 1e-19)
        ax.set_yscale('log')
        ax.set_xscale('log')
        count+=1
    plt.show()
```

4. Plot Multi-event Spectograms

In [6]:
```
1 visualizer.plot_spectrograms()
2 print(inspect.getsource(visualizer.plot_spectrograms))
```

executed in 35.1s, finished 02:15:44 2022-09-27

```
/opt/conda/lib/python3.8/site-packages/gwpy/plot/axes.py:308: MatplotlibD
eprecationWarning: shading='flat' when X and Y have the same dimensions a
s C is deprecated since 3.3.  Either specify the corners of the quadrilat
erals with X and Y, or pass shading='auto', 'nearest' or 'gouraud', or se
t rcParams['pcolor.shading'].  This will become an error two minor releas
es later.
  return super().pcolormesh(*args, **kwargs)
```




```
    def plot_spectrograms(self):
        figure = plt.gcf()
        figure.clf()
        figure.add_subplot(221)
        figure.add_subplot(222)
        figure.add_subplot(223)
        figure.add_subplot(224)
        axes = figure.get_axes()
        count=0
        figure.suptitle("Spectrogram")
        for k,f in event_names.items():
            ax = axes[count]
            fn = f['file']
            if fn == "":
                return
            t0 = f['ts']
            strain = self.get_strain(fn, f['ts'])
            dt = 1  #-- Set width of q-transform plot, in seconds
            hq = strain.q_transform(outseg=(t0-dt, t0+dt))
            ax.pcolormesh(np.linspace(0,2.4,hq.shape[0]), np.linspace(0,1
000,hq.shape[1]), hq.T)
            count+=1
        plt.show()
```

5. Merge Events in Frequency & Time Space and Unfiltered Topological Eventscape in 3D

In [7]:
```
visualizer.plot_eventscape()
print(inspect.getsource(visualizer.plot_eventscape))
```

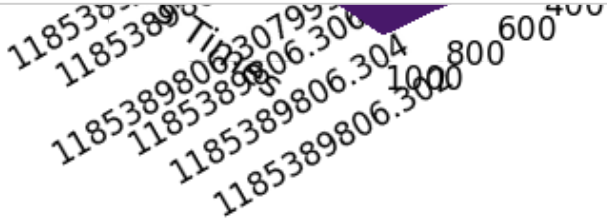
executed in 51.8s, finished 02:16:35 2022-09-27

```
dscape')
        plt.grid(b=None)

        # make the panes transparent
        ax.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
        ax.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
        ax.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
        # make the grid lines transparent
        ax.xaxis._axinfo["grid"]['color'] =  (1,1,1,0)
        ax.yaxis._axinfo["grid"]['color'] =  (1,1,1,0)
        ax.zaxis._axinfo["grid"]['color'] =  (1,1,1,0)

        ax.set_xticklabels(xticksl)

        plt.xticks(rotation=-45)
        ax.view_init(30, 80)
        plt.tight_layout()
        plt.show()
```

6 Filtered Topological Eventscape in 3D with Projected Events in 2D

In [8]:
```
1 visualizer.plot_filteredscape()
2 print(inspect.getsource(visualizer.plot_filteredscape))
```

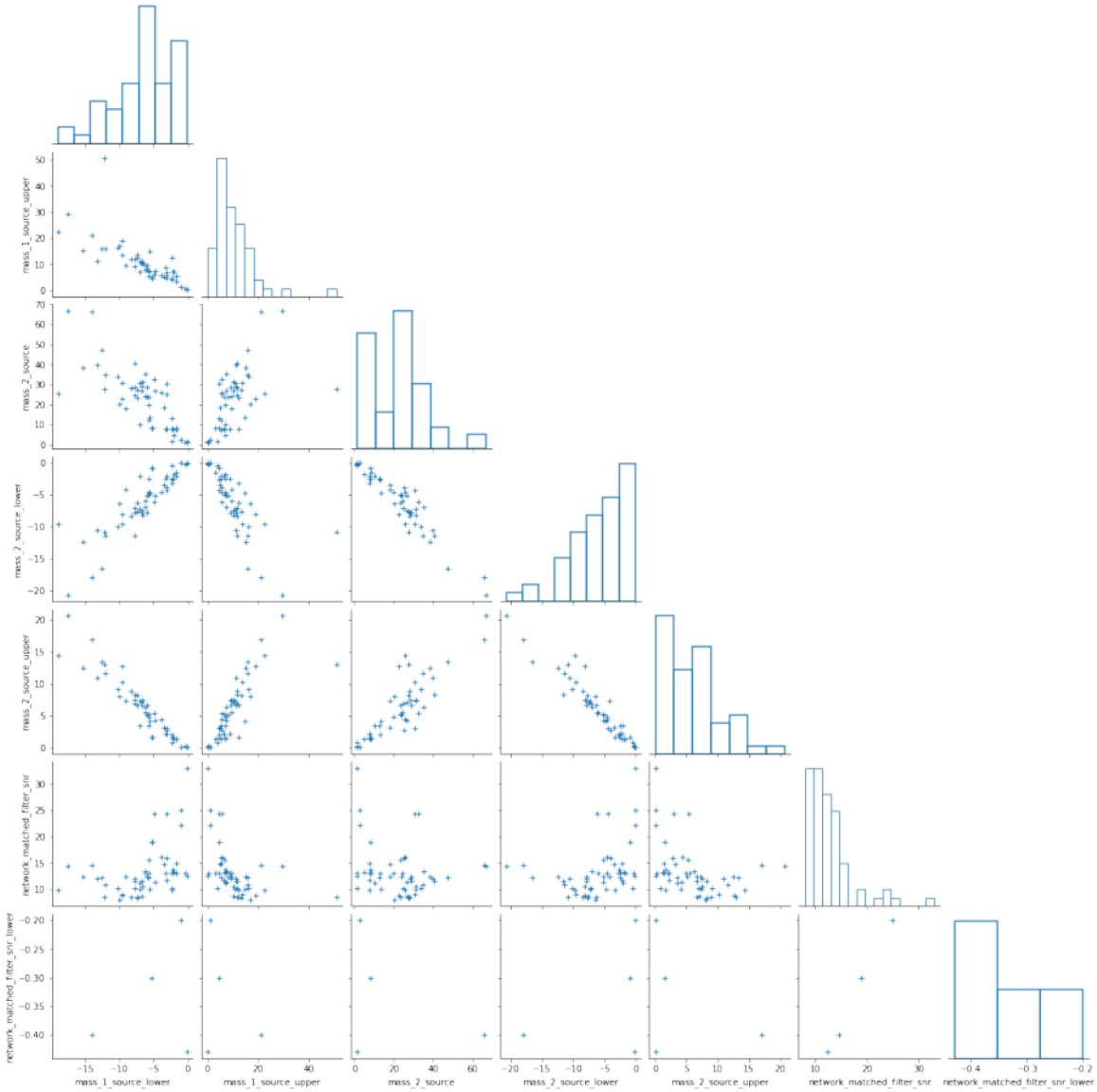
executed in 45.3s, finished 02:17:21 2022-09-27



7. Omniview of All Paramenter Estimate from Best Fit Model in Pair Plots

In [9]:
```
1 visualizer.plot_pair_plots()
2 print(inspect.getsource(visualizer.plot_pair_plots))
```

executed in 446ms, finished 02:17:21 2022-09-27



```
    def plot_pair_plots(self):
        '''
        plot secondary measures
        '''
        #import seaborn as sns
        #j
        #df = pd.read_csv("ligo_events.csv")
        #sns.pairplot(df.iloc[:,8:15], dropna=True, corner=True,     plot
_kws=dict(marker="+", linewidth=1),  diag_kws=dict(fill=False))
        fig, ax = plt.subplots(dpi=300)
        img = plt.imread("pairplot.png")
        ax.imshow(img)
        ax.axis('off')
        plt.show()
```

8. Plot Graph hierarchy for independent PE variable Clusters

In [10]:
```
1 visualizer.plot_heirarchy()
2 print(inspect.getsource(visualizer.plot_heirarchy))
```

executed in 2.09s, finished 02:17:23 2022-09-27

9. Building RK Models from Hierarchy

In [11]:
```
1 visualizer.plot_network()
2 print(inspect.getsource(visualizer.plot_network))
```

executed in 37.0s, finished 02:18:00 2022-09-27

```
4bda
Adding node to cluster Total Mass (TM2)_0:f130420f-a651-4510-ada0-2ff90bb
f2d85
Adding node to cluster Total Mass (TM3)_0:e65bcf38-7733-4e84-ab02-6d6bcd6
2b24a
Adding node to cluster Chirp Mass (CM1)_0:4139b199-1353-44ab-9c43-5d8946a
47701
Adding node to cluster Chirp Mass (CM2)_0:dd75583b-2938-41b2-9323-8f71d17
e9304
Adding node to cluster Chirp Mass (CM3)_0:cc769f99-c855-4f2a-bf87-9deb745
ca755
Adding node to cluster Final Mass (FM1)_0:34f81e18-e1f1-423a-a0a4-c1e247b
647a6
Adding node to cluster Final Mass (FM2)_0:f6ab8a9a-6601-4b56-8e4f-68b0346
41921
Adding node to cluster Final Mass (FM3)_0:8347a4b9-0ae3-4c99-accd-d4e3b37
1894d
RK Cluster has 105 edges for 15 nodes. Edge count 105. NPairs
Adding node to cluster Q-Value (Q1)_0:d1c3f8c7-9ba9-44fc-a3d2-446675889e4
b
```

10 Apply Linkers and Filters to the R-K Model and Plot Final R-K Diagrams

In [12]:
```
1 visualizer.plot_rk_diagrams()
2 print(inspect.getsource(visualizer.plot_rk_diagrams))
```

executed in 18.1s, finished 02:18:19 2022-09-27

```
            ax = plt.gca()
            angle = 0
            ax.view_init(30, angle)
            plt.grid(b=None)
            plt.axis('off')

            # make the panes transparent
            ax.xaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
            ax.yaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
            ax.zaxis.set_pane_color((1.0, 1.0, 1.0, 0.0))
            # make the grid lines transparent
            ax.xaxis._axinfo["grid"]['color'] =  (1,1,1,0)
            ax.yaxis._axinfo["grid"]['color'] =  (1,1,1,0)
            ax.zaxis._axinfo["grid"]['color'] =  (1,1,1,0)
            event.visualize(ax, distance_from_center=self.master.toolbar.
xyz.spread(), center_color=event_colors[event.name], show_legend=False)
            fig.suptitle(event.name)

        plt.show()
```