Introduction to the Compact Binary Coalescence Search Pipeline

Duncan Brown
Syracuse University
Lecture II: ihope and interpreting results
Last time...

- We looked at lalapps_inspiral_hipe
- The pipeline used to generate CBC workflows for the low-mass search
Completing a Search
Completing a Search

• To take a search to a publication requires several steps:
Completing a Search

• To take a search to a publication requires several steps:

• **Zero-lag analysis** filter the data for signals
Completing a Search

• To take a search to a publication requires several steps:
  
  • **Zero-lag analysis** filter the data for signals
  
  • **Time-slide analysis** perform coincidence on time-shifted data to determine the rate of noise triggers
Completing a Search
Completing a Search

- **False Alarm Rate computation** to measure the significance of candidates
Completing a Search

- **False Alarm Rate computation** to measure the significance of candidates
- **Initial follow-up** of candidate triggers
Completing a Search

• **False Alarm Rate computation** to measure the significant of candidates

• **Initial follow-up** of candidate triggers

• **Injection analysis** to test the efficiency of the pipeline
ihope
What is ihope?
What is ihope?

- ihope...
What is ihope?

- ihope...
- it works
What is ihope?

• ihope...
• it works
• we find a gravitational wave!
• ihope is the “uber-pipeline” that runs the entire inspiral analysis
• ihope is the “uber-pipeline” that runs the entire inspiral analysis

• find the data and generate the banks
• ihope is the “uber-pipeline” that runs the entire inspiral analysis

• find the data and generate the banks

• zero-lag analysis
• ihope is the “uber-pipeline” that runs the entire inspiral analysis
  
• find the data and generate the banks

• zero-lag analysis

• time-slides to measure background
• ihope is the “uber-pipeline” that runs the entire inspiral analysis

• find the data and generate the banks

• zero-lag analysis

• time-slides to measure background

• software injections to measure efficiency
• ihope works in a similar way to hipe
• ihope works in a similar way to hipe
• ihope works in a similar way to hipe
• ihope works in a similar way to hipe
• hipe can be told (via the command line) to generate all or part of the pipeline when writing a workflow
• hipe can be told (via the command line) to generate all or part of the pipeline when writing a workflow

• For example, it could skip the data find and template bank steps and just perform filtering and coincidence
• ihope uses this functionality to call hipe multiple times to generate pieces of the workflow
• ihope uses this functionality to call hipe multiple times to generate pieces of the workflow

• ihope chains these pieces together to form an “uber-workflow”
Get science times for detectors
ligolw_segment_query

Get science times for detectors

ligolw_segments_from_cats

Get veto times for detectors
Get science times for detectors

Get veto times for detectors

Determine times to be filtered
Get science times for detectors

Get veto times for detectors

Determine times to be filtered

Generate the “data find” workflow
Get science times for detectors

Get veto times for detectors

Determine times to be filtered

Generate the “data find” workflow

H1 ligo_data_find

L1 ligo_data_find

V1 ligo_data_find

lalapps_tmpltbank

lalapps_tmpltbank

lalapps_tmpltbank
- ligolw_segment_query
- ligolw_segments_from_cats
- segment arithmetic
- lalapps_inspiral_hipe

Get science times for detectors
Get veto times for detectors
Determine times to be filtered
Generate the “data find” workflow
Generate the “data find” workflow
Generate the “data find” workflow
Generate the “data find” workflow

Generate the “full data” workflow
Generate the “data find” workflow

Generate the “full data” workflow
Generate the “data find” workflow

Generate the “full data” workflow
Generate the “data find” workflow
Generate the “full data” workflow
Generate the “CAT2” workflow

CAT 2 vetos
lalapps_thinca
Second thinca
Generate the “data find” workflow

Generate the “full data” workflow

Generate the “CAT2” workflow

Generate the “CAT3” workflow

lalapps_inspiral_hipe

lalapps_inspiral_hipe

lalapps_inspiral_hipe

lalapps_inspiral_hipe

lalapps_inspiral_hipe

CAT 3 vetos

lalapps_thinca

Second thinca
Generate the “data find” workflow

Generate the “full data” workflow

Generate the “CAT2” workflow

Generate the “CAT3” workflow

Generate the “CAT4” workflow
lalapps_inspiral_hipe

Generate the “CAT4” workflow
Generate the “CAT4” workflow
lalapps_inspiral_hipe

Generate the “CAT4” workflow
Generate the “CAT4” workflow

Generate an “injection” workflow
Generate the “CAT4” workflow

Generate an “injection” workflow
Generate the “CAT4” workflow

Generate an “injection” workflow

lalapps_inspiral_hipe

INJ lalapps_inspiral

INJ lalapps_inspiral

INJ lalapps_inspiral

lalapps_thinca

trig_to_tmpltbank

INJ lalapps_inspiral

INJ lalapps_inspiral

INJ lalapps_inspiral

lalapps_thinca
Generate the “CAT4” workflow

Generate an “injection” workflow
lalapps_inspiral_hipe

Generate the “CAT4” workflow

lalapps_inspiral_hipe

Generate an “injection” workflow

lalapps_inspiral_hipe

Generate an “injection” workflow

lalapps_inspiral_hipe

Generate an “injection” workflow

lalapps_inspiral_hipe

Generate an “injection” workflow

lalapps_inspiral_pipedown

Compute the false alarm rates for candidate triggers, plot the results, and follow up loudest candidates
A quick look at pipedown
A quick look at pipedown

- Pipedown is another CBC pipeline
A quick look at pipedown

• Pipedown is another CBC pipeline
• It is responsible for computing the false alarm rate of triggers and plotting the output of the analysis
A quick look at pipedown

• Pipedown is another CBC pipeline

• It is responsible for computing the false alarm rate of triggers and plotting the output of the analysis

• See C. Capano’s PhD thesis for more information
Looking at the results
The full output of the ihope/hipe/pipedown pipeline is presented as an interactive web page.
• The most important part of this page is section 3, which summarizes the key results
• The most important part of this page is section 3, which summarizes the key results.

• Range plots show the sensitivity of the detector and a gross view of detector sensitivity.
• The most important part of this page is section 3, which summarizes the key results

• Range plots show the sensitivity of the detector and a gross view of detector sensitivity

• Inverse False Alarm Rate (IFAR) plots show the significance of the candidate events
• The most important part of this page is section 3, which summarizes the key results

• Range plots show the sensitivity of the detector and a gross view of detector sensitivity

• Inverse False Alarm Rate (IFAR) plots show the significance of the candidate events

• Found and missed injections show how efficient the pipeline is
False Alarm Rate

• Given a zero-lag trigger, how many background triggers in our analysis have a larger signal-to-noise ratio

• By counting the number of time-slide triggers above the SNR of a zero-lag trigger we obtain its FAR
H1,L1 Time: PLAYGROUND Cum. Num. vs Combined IFAR

- **Expected Background**
- **All Coincs**
- **N^{1/2} errors**
- **2N^{1/2} errors**

**Cumulative #**

**Inverse False Alarm Rate (yr)**

Log-log scale graph showing the cumulative number of events vs the inverse false alarm rate for different error scenarios.
Hardware Injections
<table>
<thead>
<tr>
<th>rank (using combined far)</th>
<th>combined far</th>
<th>fap</th>
<th>fap 1yr</th>
<th>snr</th>
<th>end time</th>
<th>end time utc (click for daily hope)</th>
<th>ifos (click for elog)</th>
<th>mass</th>
<th>mchirp</th>
<th>mini followup</th>
<th>omega scan</th>
<th>playground duration (days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>25.29</td>
<td>967824212.13</td>
<td>Mon 06 Sep 2010 16:03:17</td>
<td>H1,L1</td>
<td>25.0</td>
<td>10.88</td>
<td>link</td>
<td>H1 L1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>36.88</td>
<td>968428988.57</td>
<td>Mon 13 Sep 2010 16:02:53</td>
<td>H1,L1</td>
<td>23.95</td>
<td>9.01</td>
<td>link</td>
<td>H1 L1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>210.24</td>
<td>0.27</td>
<td>1.0</td>
<td>8.98</td>
<td>968658410.49</td>
<td>Thu 16 Sep 2010 07:46:35</td>
<td>H1,L1</td>
<td>17.69</td>
<td>4.3</td>
<td>link</td>
<td>H1 L1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>215.92</td>
<td>0.28</td>
<td>1.0</td>
<td>7.95</td>
<td>968690339.48</td>
<td>Thu 16 Sep 2010 16:38:44</td>
<td>H1,L1</td>
<td>18.72</td>
<td>8.12</td>
<td>link</td>
<td>H1 L1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>300.52</td>
<td>0.37</td>
<td>1.0</td>
<td>7.85</td>
<td>968773002.57</td>
<td>Fri 17 Sep 2010 15:36:27</td>
<td>H1,L1</td>
<td>24.08</td>
<td>7.45</td>
<td>link</td>
<td>H1 L1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1257.0</td>
<td>0.85</td>
<td>1.0</td>
<td>8.56</td>
<td>968422708.58</td>
<td>Mon 13 Sep 2010</td>
<td>H1,L1</td>
<td>17.11</td>
<td>3.51</td>
<td>link</td>
<td>H1 L1</td>
<td>0.56</td>
</tr>
</tbody>
</table>
No significant signal
H1,L1,V1 Time: ALL-DATA Cum. Num. vs Combined IFAR

- Expected Background
- Combined Coincs
- $N^{1/2}$ errors
- $2N^{1/2}$ errors

Cumulative # vs Inverse False Alarm Rate (yr)
What do we do if we see something significant?
Channel 1 at 940552123.735 with Q of 45.3
• This particular trigger was due to a “glitch” in L1 related to the detector’s control system
• This particular trigger was due to a “glitch” in L1 related to the detector’s control system

• Moral: looking at the detector’s data is very important!
• This particular trigger was due to a “glitch” in L1 related to the detector’s control system

• Moral: looking at the detector’s data is very important!

• Detector characterization is vital to the search for gravitational waves
How do we know things are working?
• Test the analysis pipeline using software injections
Next time: running ihope