Multi-band template analysis for CB search

Frédérique MARION for the VIRGO Collaboration
GWDGAW 2003
The Multi-Band Template Analysis

- Alternate matched filtering technique
  - designed to release stress on computing resources for CB search

- Split analysis in a few (2 - 3) frequency bands
  - coherent band combination provides result for full (virtual) template
  - for each band, number of templates and FFT size both reduced
  - CPU and storage requirements reduced
    » up to factors 100 for CPU and 500 for storage
      - for 3 bands, low minimal mass, low minimal frequency

- Built-in hierarchical search
  - each band can be analyzed independently
  - coherent combination grants unchanged SNR
MBTA today

- Prototype algorithm implementation ~ complete
  - filtering machinery
  - search algorithm
  - event clustering

- Interface to template computation and placement library
  - inspiral library provides several template generators
  - grid generation currently based on smallest elliptical isomatch contour
    » plan to try true isomatch contours

- VIRGO CITF E4 data
  - simple test analysis in realistic environment (Moriond 2003)

- Mock Data Challenges
  - validation process in well defined conditions
CITF E4 data test analysis (I)

- ~ 10 hours of quiet data
  - ITF & OMC locked

- Monitor horizon distance for a few masses
  - evidence bad periods
CITF E4 data test analysis (II)

- Single template search
  - (3 M\(_\odot\), 3 M\(_\odot\))
  - [50 Hz - 2 kHz]
- Probe ITF noise level
  - quiet enough after simple vetoes
- Compare 1 & 2 bands analyses
  - consistency checked
  - SNR correlation fairly good
Validation through MDCs

3 mock data challenges held in VIRGO in 2003

- data generated with SIESTA
  - based on CITF E4 spectrum (sensitivity mostly above 80 Hz)
    - non-stationarities & unlocked segments introduced in MDC III
  - simulated events from inspiral
    - various models, various SNR

- probe integration of software pieces needed for CB analysis
- probe algorithm performances
  - detection efficiency, SNR recovery
  - robustness to data flaws
Detection efficiency

- **Event selection**
  - event clustering allows to rely on SNR cut
  - regular noise fallout allows detection of events with SNR > ~7

- **Selection efficiency**
  - typically at 95% level for SNR \( \geq 7 \)
  - many studies to understand SNR loss budget
    - grid
    - template generator
    - lower and upper analysis frequency
2 bands vs 1 band

- **Systematic comparisons**
  - same efficiency
  - same purity
  - good SNR correlation
    
    \[
    \frac{SNR}{SNR} = 0.99 \pm 0.07
    \]

- **Increased computing efficiency**
  - limited due to narrow-band spectrum used in MDCs so far
CPU gain estimation

- Measure gain brought by multi-band analysis in realistic conditions
  - wide-band spectrum
    - VIRGO like
    - [40 Hz - 2 kHz] analysis
  - significant mass range
    - [1.35 \( M_\odot \), 5 \( M_\odot \)]
    - \(~ 10000\) templates
  - linux PC
    - P4, 2.4 GHz, 1GB memory

- Measure time needed to process 1800 s of data & memory
  - 1 band analysis
  - 2 bands analysis
    - no search, flat search, hierarchical search
Search cost evolution

- **Restricted mass range**
  - \([1.35 \, M_\odot, 1.45 \, M_\odot]\)

- **2 bands analysis**
  - no search
    - FFT cost only
  - flat search
    - bands always combined
  - hierarchical search
    - bands combined only if SNR \(\geq 5\) in one band

- **Best ratio to 1 band analysis (CPU)**
  - for optimal splitting frequency
    - 1/18 no search
    - 1/9 hierarchical search
Optimal search cost

- 2 bands analysis with 130 Hz splitting frequency
  - full mass range [1.35 M☉, 5 M☉]
  - 9707 templates
  - 1800 s of data

<table>
<thead>
<tr>
<th></th>
<th>Memory (MB)</th>
<th>Mem/T (MB)</th>
<th>Processing (s)</th>
<th>Proc/T (s)</th>
<th>Proc/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>No search</td>
<td>3819</td>
<td>0.39</td>
<td>6640</td>
<td>0.68</td>
<td>3.7</td>
</tr>
<tr>
<td>Hierarchical</td>
<td>11351</td>
<td>1.17</td>
<td>11972</td>
<td>1.23</td>
<td>6.7</td>
</tr>
<tr>
<td>Flat</td>
<td>10685</td>
<td>1.10</td>
<td>38863</td>
<td>4.00</td>
<td>21.6</td>
</tr>
</tbody>
</table>

- 7 similar CPUs would be needed for real time analysis
Plans for improvement

- **Not specific to MBTA**
  - go to FFTW3
  - optimize template placement
  - use increased number of models for templates

- **Specific to MBTA**
  - use single precision?
  - optimize recombination
    - on part of vectors
    - introduce consistency checks beforehand
      - restrain sensitivity to excess noise
  - go to 3 bands
  - technical tuning
    - initialization speed-up (association of virtual and real templates)
Conclusion

- Prototype implementation of MBTA available
- Tested both on real and simulated data
- Gain on analysis cost measured
  - factor ~ 10 now, room for improvement
- Online integration soon
  - MDC IV
  - real-time analysis of engineering run data