

Scheda esperienza

Analisi di onde gravitazionali dalla coalescenza di oggetti compatti

In questa esperienza analizzeremo l'emissione di onde gravitazionali prodotte durante la fase di coalescenza di due oggetti compatti, come ad esempio due buchi neri (*Binary Black Hole*, BBH) o due stelle di neutroni (*Binary Neutron Star*, BNS). Verranno utilizzati i dati raccolti dagli esperimenti LIGO e Virgo nel corso delle campagne osservative Observing Run 1 (O1) e 2 (O2) effettuate fra il 2015 e il 2017. I dati relativi ai run sono disponibili nel portale *Gravitational Wave Open Science Center* (GWOSC) accessibile all'indirizzo www.gw-openscience.org.

Per l'esperienza sarà necessario utilizzare i seguenti strumenti

- Python (Pandas, Astropy)
- Pacchetti per analisi dati gravitazionali *gwpy*, *pycbc*, già disponibili tramite Jupyter notebook sulla macchina virtuale

Come per le esperienze precedenti, anche in questo caso nel pacchetto Github è presente una cartella tutorials che contiene un esempio di analisi Python con gli strumenti necessari per l'esperienza.

1 - Scaricare i dati relativi agli eventi di onde gravitazionali

In questa parte dell'esperienza scaricheremo i dati gravitazionali dal portale GWOSC.

- Utilizzando la funzione `fetch_open_data` di *gwpy*, scaricare i dati intorno al tempo t_0 GPS degli eventi indicati in Tabella 1. Utilizzare un intervallo temporale che copre un periodo di un'ora centrato in t_0 . E' possibile anche interrogare tramite Python l'archivio GWOSC per vedere gli eventi disponibili.
- Per ciascun evento, scaricare i dati relativi agli interferometri indicati in tabella, ricordando che H1 = LIGO Hanford, L1 = LIGO Livingston, V1 = Virgo. In Tabella 1 sono riportati gli interferometri (ifo) che hanno osservato ciascun evento.
- Scaricare i dati relativi sia allo strain gravitazionale che ai segmenti di data quality.

2 - Analisi nel rumore nell'interferometro

Utilizzando il pacchetto *gwpy* (già disponibile su Jupyter Notebook nella macchina virtuale), andremo a studiare lo spettro del rumore intorno all'evento.

Studiare l'evoluzione della ASD del rumore in funzione del tempo. Nell'intervallo di un'ora dei dati scaricati, costruire la ASD selezionando dei sottointervalli di alcuni secondi (ad esempio 4 oppure 8), e per ciascuno di essi costruire la ASD. E' possibile valutare se la ASD varia in funzione del tempo? E' possibile ad esempio valutare se ci sono delle linee spettrali che cambiano intensità nel corso dell'ora coperta dai dati?

E' possibile sviluppare in modo quantitativo un modo per determinare se in un intervallo di frequenza il valore della ASD è costante oppure è variato nel corso della presa dati di un'ora? Un modo per farlo può essere i valori della ASD mediando all'interno di piccoli intervalli di frequenza (ad es. 10 Hz) e vedere se questo valore medio varia in modo significativo nel tempo oppure no (è possibile ad esempio valutare la deviazione rispetto a un valore medio).

Ripetere queste operazioni per gli interferometri che hanno rivelato l'evento, confrontando le ASD. Qual è l'interferometro più sensibile nelle diverse regioni di frequenza?

3 - Analisi temporale e *quicklook* intorno all'evento

Prima di tutto, verificare usando la funzione Plot di gwpy, che l'evento è avvenuto in un periodo di *data quality* valido.

Utilizzando poi gwpy, fare un grafico della serie temporale di strain $h(t)$. E' possibile vedere il segnale gravitazionale? Per quelli con SNR più alto sarà possibile vedere un picco, naturalmente selezionando un intervallo di pochi secondi intorno a t_0 .

Filtrare i dati con un passabanda (ad esempio da 50 a 300 Hz), e successivamente, per i rivelatori LIGO, fare un notch a 60 Hz. Rappresentare la serie temporale, mostrando solo la parte centrale, ad esempio 2 secondi intorno all'evento. E' possibile adesso vedere meglio il segnale?

Ripetere queste operazioni per gli interferometri che hanno rivelato l'evento.

4 – Ricerca del segnale con template matching

Utilizziamo adesso il pacchetto *pyCBC* (pycbc.org) per costruire un template del segnale ed effettuare il matched filter, in modo da riprodurre in modo semplice la procedura di *detection* dei segnali di coalescenze binarie.

Effettuare un downsampling delle frequenze a 2048 Hz, utilizzando la funzione *resample_to_delta_t*. Rimuovere 2 secondi di dati per eliminare gli effetti di ringing ai bordi della serie temporale, come spiegato nel tutorial su Jupyter Notebook.

Utilizzando la funzione *get_td_waveform*, costruire un banco di modelli di funzioni d'onda per un segnale gravitazionale assumendo una coalescenza fra due oggetti di massa m_1 e m_2 . Costruire ad esempio una griglia di coppie (m_1, m_2) a partire da un valore iniziale m_{guess} suggerito in Tabella 1. Se per esempio abbiamo $m_{\text{guess}}=20$, è possibile ad esempio utilizzare una spaziatura in massa pari a 0.5, coprendo ad esempio un intervallo di 5 intorno a m_{guess} . Avremo quindi, ad esempio per m_1 , un intervallo di valori che vanno da 15 a 25, intervallati da 0.5, (quindi 15, 15.5, 16, ecc), costruendo così una griglia di 20 masse di test in m_1 e 20 in m_2 , quindi in teoria $20 \times 20 = 400$ punti. Tuttavia, è possibile sfruttare la simmetria fra m_1 e m_2 ? Costruire una forma d'onda scambiando le masse e vedere se cambia, ed eventualmente usare questo risultato per esplorare anche una regione più ampia di masse. Utilizzare come approssimante la funzione *SEOBNRv4_opt*.

Costruire la *Power Spectral Density* (PSD) del segnale, utilizzando il metodo di Welch come spiegato nel tutorial. Utilizzando il template, i dati e la PSD, costruire il matched filter per ciascuna coppia m_1, m_2 , identificando i valori che danno un SNR più alto. Per implementare il matched filter, è possibile scrivere una piccola funzione che fa il matched filter, oppure utilizzare la funzione *matched_filter* di pycbc. Quanto vale il massimo rapporto segnale-rumore, e per quali valori delle masse m_1 e m_2 ?

Ripetere l'operazione per gli interferometri interessati. Qual è quello che ha rivelato per primo il segnale? Quali sono i rapporti segnale-rumore nei singoli rivelatori? C'è una differenza fra i tempi di arrivo nei rivelatori, e in caso affermativo, è consistente con il tempo di propagazione fra i

rivelatori? Spostare il template in modo da sovrapporlo ai dati gravitazionali, avendo cura di scalare opportunamente e di “sbiancare” i segnali con la PSD, come mostrato nel tutorial.

Tabella 1: Eventi gravitazionali

Nome	tipo	ifo	m_{guess}	t_0
GW150914	BBH	H1, L1	30	1126259462.4
GW151226	BBH	H1, L1	8	1135136350.6
GW170814	BBH	H1, L1,V1	20	1186741861.5
GW170809	BBH	H1, L1,V1	30	1186302519.8
GW170608	BBH	H1, L1	12	1180922494.5
GW170104	BBH	H1, L1	22	1167559936.6