Multi-IFO PyCBC search outline

Blue: Full data
Square: single IFO

Data from external source

Results

Hexagon: Multiple IFOs
Blue: original data

Square: single IFO
Red: Injection

Full Data

Template Bank

H1 Data

L1 data

V1 data

Inspiral Jobs H1

Inspiral Jobs L1

Inspiral Jobs V1

Trigger Merge

Trigger Merge

Trigger Merge

Fit by Template
Fit over multiparam

Fit by Template
Fit over multiparam

Fit by Template
Fit over multiparam

Loudest Singles plots/followup → closed box results

Veto Definer

P/T/A histograms

H1L1 coincs

H1V1 coincs

H1L1V1 coincs

L1V1 coincs

H1L1 statmap

H1V1 statmap

H1L1V1 statmap

L1V1 statmap

H1L1 exclude zerolag

H1V1 exclude zerolag

H1L1V1 exclude zerolag

L1V1 exclude zerolag

H1L1V1 add statmap

Foreground Censor

Foreground coincidences significance and followup → open box results

Injection

Injection Creation

Injection Information

Inspiral Jobs H1

Inspiral Jobs L1

Inspiral Jobs V1

Trigger Merge

Trigger Merge

Trigger Merge

H1L1 coincs

H1V1 coincs

H1L1V1 coincs

L1V1 coincs

(IFO data)

H

L

V

Calculate/Merge PSDs

H1L1 statmap_inj

H1V1 statmap_inj

H1L1V1 statmap_inj

L1V1 statmap_inj

Optimal SNR

H1L1V1 add statmap_inj

Same injection process but for other injection types

Hdf injfind Type I

Hdf injfind Type II

Hdf injfind all_inj

Found/Missed Injections Plotting and Followup → closed box results

Each stage short explanation

Data	P/T/A Histogram	Veto Definer	Template Bank	Coincs
h(t) data from the interferometers. Contains gravitational wave strain	Contains information on distribution of time differences, phase differences and amplitude ratios for coincidences of real signals	Contains information on what times in the data are vetoed (not considered for analysis)	The set of waveform templates used to compare to the h(t) data in the matched filtering stage	• Cross-check triggers between interferometers to see if they are within a certain time window of one another (usually the time-of-flight between detectors) • Calculate the ranking statistic (where it uses p/t/a histogram) this is the transform of the SNR combined with other information to see which coincidence is most like a signal • Requires the triggers to be from same template • Perform timeslides; sequential shifts to collect noise triggers • Split for multi-threaded analysis • One for each coincidence type; HL, HV, LV, HLV • The coincidences will clustered, meaning that only the loudest coincs within a certain window will be kept
Inspiral Jobs	**Trigger Merge**	**Fit by Template**	**Fit over multiparam**	
• Matched filtering for each detector • Whitened data is cross-correlated with whitened templates to calculate SNR • Usually split up for multi-threaded processing	Collation stage where the inspiral jobs from each detector are re-gathered for passing on to next stages	SNRs of triggers are considered to find distributions. This is done for each template	• Each template may have small numbers of triggers • Fit coefficients and trigger counts are smoothed among nearby templates to get better statistics • This is done in multiple dimensions, hence multiparam	

Statmap	Foreground Censor	Exclude Zerolag	Add Statmap
Compare ranking statistics and see how rare they are to calculate FAR	Removes foreground events from loudest singles list for closed box results	Take zerolag coincidences from all coincidence types and remove any background triggers using any of the single-detector triggers which form these coincs This means that e.g. the H trigger from a HL coinc will not show up in HV backgrounds – this would happen if V did not trigger	Combine the coinc statistics from the different coinc types: • cluster by time over different types, i.e. highest ranking statistic in HL vs HV vs LV vs HLV • recalculate overall FAR

Injection Creation	Inspiral	Coincs	Statmap_inj	Add Statmap Inj	Calculate/Merge PSDs	Optimal SNR
Make waveforms which are to be injected into the data of each IFO, store their information	The same as the inspiral for full data, except that the injections have been added in	Same as coincidence stage in full data Except uses trigger fitting from full data	Calculates FARs for injection zerolag Uses background from full_data (from exclude_zerolag)	Combine FAR calculation from statmap_inj Cluster over coincidence types	Calculate and collate power spectral density information for each detector	Calculate the SNR with which a signal would be found in an optimal search