
MBTA O4 uber banks construction and formatting documentation

Aubin Florian, Faedi Francesca, Guidi M. Gianluca,
the MBTA team

VIR-0244A-23

16th March 2023

Contents

1	Construction of the banks	3
1.1	Logic and algorithms	3
1.1.1	Virtual and real template banks	3
1.1.2	Geometric placement	3
1.1.3	Stochastic algorithm	3
1.1.4	Hybrid algorithm	4
1.2	The VT bank	5
1.3	The RT banks	6
2	Match	7
3	Formatting	7
4	Region 4	8
5	Bank verification	10
5.1	Measurement of the effective fitting factor on the targeted space	10
5.2	Measurement of the fitting factor on a larger parameter space	12
5.2.1	Spinning BNS	12
5.2.2	Higher total mass and mass ratio	14

In this document we discuss how the MBTA O4 uber bank is built. All configurations and scripts used are available in the MBTA git: git.ligo.org/mbta/o4-uber-banks

The first section deals with the logic behind the bank and how it is constructed. The second section is about the calculation of the match between the uber bank and the shorter frequency band banks. The third section presents the format used for the bank. The fourth section explains what the historically called "Region 4" is, and how it is constructed. Finally, the fifth section presents a verification of the effectiveness of the bank.

1 Construction of the banks

1.1 Logic and algorithms

1.1.1 Virtual and real template banks

MBTA relies on pre-generated template banks to cover a given parameter space. In O4, MBTA will consider a 4-dimensional space consisting of the masses of the compact objects (m_1, m_2) and their dimensionless spins assumed to be parallel to the orbital angular momentum (χ_1^z, χ_2^z).

The template banks are generated in such way they cover the space with a limited signal-to-noise ratio (SNR) loss (typically less than 3%). This definition assumes given frequency range and detector sensitivity curve (ASD or PSD).

MBTA splits the match filtering into two frequency bands and then recombines the outputs to construct time series over the full detector's bandwidth [1]. We therefore need to generate three banks:

- a bank over the full frequency band, called the virtual template (VT) bank
- a bank over the lower frequency band, called the low frequency real template (LF RT) bank
- a bank over the upper frequency band, called the high frequency real template (HF RT) bank

1.1.2 Geometric placement

There are several ways to build template banks. The challenge is to cover the parameter space with a minimum number of templates.

The space can be mapped such that the distance between two templates represents the mismatch (i.e. the SNR loss).

It can be shown that for low masses, the metric associated with this distance is fairly constant across the space. This can be exploited to make a geometrical placement of templates.

The `pycbc_geom_aligned_bank` algorithm uses this feature to quickly construct template banks for BNS with aligned spins [2].

1.1.3 Stochastic algorithm

For higher masses, the metric changes too quickly to use a purely geometric approach.

During O3, MBTA used a stochastic algorithm, called `lalapps_cbc_sbank`. It basically consists of picking a random point in the space, generating the waveform, calculating the exact match between this template and those already selected, and adding it to the list if necessary to achieve the required

minimum match [3].

This method takes time to converge and tends to over-cover the space. It is not optimal and can lead to a large number of templates for the extended parameter space we want for O4.

1.1.4 Hybrid algorithm

The PyCBC team has proposed an efficient hybrid geometric-random placement algorithm to generate high mass template banks with aligned spins and no higher order modes [4, 5, 6].

This algorithm works on a 3-dimensional space, the two masses (m_1, m_2) and a reduce spin parameter:

$$\chi_r = \left(1 - \frac{76\eta}{113}\right) \chi_s + \delta\chi_a \quad (1)$$

With $\chi_s = (\chi_1^z + \chi_2^z)/2$ the symmetric spin, $\chi_a = (\chi_1^z - \chi_2^z)/2$ the asymmetric spin, $\eta = m_1 m_2 / (m_1 + m_2)^2$ the symmetric mass ratio, and $\delta = (m_1 - m_2) / (m_1 + m_2)$ the asymmetric mass ratio.

Because MBTA expects 4-dimensional templates, the spin degeneracy is resolved by imposing zero spins for neutron stars ($m \leq 2$) and evenly distributed spins for binary black holes (m_1 and $m_2 > 2$). For non-precessing waveforms, this conversion gives exactly the same results up to PN order 1.5, and less than one percent mismatch for higher orders [7].

After generating an uniform grid of a few million points in the (m_1, m_2, χ_r) space, a transformation is applied to move into the dimensionless chirp time coordinate system:

$$\begin{aligned} \theta_0 &= \frac{5}{2^{1/3}} \left(\frac{1}{16\pi f_0 (m_1 + m_2) \eta^{3/5}} \right)^{5/3} \\ \theta_3 &= \left(\frac{16\pi^5 \theta_0^2}{25\eta^3} \right)^{1/5} \\ \theta_{3s} &= \frac{113\chi_r \theta_3}{48\pi} \end{aligned} \quad (2)$$

With f_0 the waveform's lower frequency cut-off.

It has been shown that in this coordinate system, the metric changes more slowly. This feature is used to apply a local geometric placement.

A point of the grid is then randomly selected, added to the final bank, and the local metric is calculated. Using a truncated 3-dimensional octahedral lattice, 14 new points are generated where new templates are expected according to the estimated metric distance. After checking that these points are in the required space, and calculating the exact match between them and the already selected templates, the surviving ones are added to the bank.

This local geometric placement is repeated on the new templates until no more templates are added to the bank.

Then all redundant points in the initial grid are removed. After this step, if the grid is not empty, a new random point is picked and used as the seed of the geometric placement.

The size of the final bank depends strongly on the density of the initial grid, up to a critical value, where it reaches a plateau. However, an initial grid too dense slows down significantly the algorithm

and tends to over-cover the space. With well chosen parameters, this algorithm produces banks with on average 5 – 10% fewer templates than the purely stochastic algorithm for similar coverage.

1.2 The VT bank

It is possible to initialise stochastic algorithms, such as sbank or the hybrid code presented in sections 1.1.3 and 1.1.4, with pre-generated banks. We have used this feature to hierarchically build the MBTA O4 uber bank with variable coverages (i.e. minimal matches) across the phase space.

We first construct a BNS seed bank using the geometric placement presented in section 1.1.2. This bank contains binaries with individual masses between 1 and 3 solar masses (M_{\odot}). Individual spins are limited to ± 0.05 for masses below $2 M_{\odot}$.

Because we want to ensure a good recovery of those sources for which EM counterparts are expected, this bank has a minimal match of 0.98.

We then used the hybrid algorithm discussed in section 1.1.4 to construct a BBH seed bank. This bank is made to improve the coverage of symmetrical BBHs with individual masses between $5 M_{\odot}$ and $500 M_{\odot}$ and a mass ratio (m_1/m_2 , $m_1 \geq m_2$) below 3. Spins are free to vary up to ± 0.997 .

It is in this space that most O3 events have been detected, so we decide to increase the minimal match there up to 0.98.

Finally, we used these two seeds to initialise the uber bank. It has then been completed using the hybrid algorithm to reach a minimal match of 0.965 in the rest of the space defined in Table 1.

We have chosen to apply lower duration cuts to avoid too short templates. These templates tend to produce a lot of noise and cover little or no expected astrophysical population. If they exist, these signals can still be recovered (with higher SNR losses) by other MBTA templates, or by unmodelled burst searches. In the rest of this document, we will discuss the case of a duration cut-off of 200 ms for the BBH seed and the uber bank. We also consider the possibility of reducing this limit to 100 ms for the BBH seed, with the aim of better recovering massive symmetric BBHs.

	BNS seed	BBH seed	Uber bank (including the seeds)
Individual masses ($m_{1,2}$) [M_\odot]	[1, 3]	[5, 500]	[1, 500]
Total mass ($m_1 + m_2$) [M_\odot]	[2, 4]	[10, 500]	[1, 500]
Mass ratio ($m_1/m_2 \geq 1$)	[1, 3]	[1, 3]	[1, 50]
Individual aligned spins ($\chi_{1,2}^z$)		$[-0.05, 0.05]$ if $m_{1,2} \leq 2 M_\odot$ $[-0.997, 0.997]$ else	
Minimal duration [ms]	-	200 (or 100)	200
Minimal match	0.98		0.965
Waveform approximants used for bank generation (freq. domain)	TaylorF2	SEOBNRv4_ROM	
Waveform approximants used for analysis (time domain)	SpinTaylorT4	SEOBNRv4_opt	
PSD used for bank generation	L1 O4 high-sensitivity (BNS range ≈ 190 Mpc)		
Low frequency cut-off (f_0) [Hz] used for bank generation	25	18	
Low frequency cut-off (f_0) [Hz] used for analysis	24	20	
High frequency cut-off (f_{max}) [Hz] used for bank generation	2048	1024	
High frequency cut-off (f_{max}) [Hz] used for analysis		2048	
LF and HF bands separation frequency (f_{cut}) [Hz]	80		

Table 1: MBTA O4 uber bank parameters

1.3 The RT banks

As we saw in section 1.1.1, we also need to generate two real template (RT) banks on two distinct frequency bands. For O4, we choose to set the separation frequency between these two bands to $f_{cut} = 80$ Hz.

The RT banks are generated following the same logic than the VT bank discussed in section 1.2.

They use the same parameters as those defined in Table 1, except that no duration cut-off is applied. Indeed, this cut has been tuned on VT waveforms, for which the duration is defined over the entire detector’s bandwidth.

As some templates of the targeted space can merged before the separation frequency (80 Hz), the hybrid code will return errors when trying to generate these waveforms for the HF RT bank creation. To solve this problem, we set the lower frequency cut-off used by the generator to 25 Hz, and we used an ASD with very high values (set to 1) before 80 Hz. As a result, the SNR calculation does not effectively take into account data below 80 Hz.

2 Match

In practice, MBTA applies matched filtering on the RTs, and then uses the results to reconstruct the SNRs of the VTs. In order to know which RT time series MBTA should use to reconstruct the time series of each VT, we have to search for the best RT-VT association on each frequency band.

To achieve this, we use a modified version of the PyCBC "banksim" algorithm. This algorithm is initially designed to estimate the effective fitting factor of a given bank. To do this, it simulates a large number of signals (called injections) and calculates their best match across the bank [8].

We use this algorithm to match a given RT bank with injections that are actually the VT bank. As output, it produces a ".dat" file that contains one line per VT, with the VT parameters, the best matching RT parameters and the value of the match.

The typical number of VTs is one million and the typical number of RTs is between 10 and 100 thousands, which means that we have to calculate at least 10 billion matches. This takes a lot of time, and banksim is not designed for that purpose.

One solution already implemented in the original algorithm is to limit the relative difference in mass of the chirps when it tries to combine RTs and VTs. However, the density of the bank changes too quickly in the chirp mass space. A unique chirp mass window may not be effective across the full space.

To speed up the process, we decided not to use the chirp mass window option. Instead, we modified the banksim code to stop processing a given VT when the algorithm finds a RT with a match that exceeds a given threshold (typically 0.98).

In addition, we are investigating the possibility of optimising the way the algorithm explores possible associations. At the moment, it randomly tries each VT and RT pair, and stops when it reaches the threshold we discussed earlier. Our idea is to start by trying to match a given VT by the closest RT in chirp mass. If the resulting match exceeds the threshold, we can stop the processing for this VT and move on to the next one. However, if the match is not large enough, we can try with the second closest RT, and so on. This has not yet been implemented.

After running banksim two times (one to match VT and LF RT banks, and one to match VT and HF RT banks), we need to translate the two resulting ".dat" files into four ASCII files (ids start from 0):

- one VT bank, with the format: $m_1 m_2 \chi_1^z \chi_2^z$
- two RT banks, with the format: $m_1 m_2 \chi_1^z \chi_2^z$
- an association file, with the format: $\text{id}_{\text{VT}} \text{id}_{\text{LF}} \text{id}_{\text{HF}}$

To do this, we wrote the shell script named "extractMatch.csh".

3 Formatting

The MBTA pipeline splits the VT bank into several jobs that are run in parallel. In order to minimise overall CPU and memory usage, it is necessary to reorder the bank. As the most CPU demanding step is the HF filtering, we have chosen to minimise the number of HF RTs per job.

In addition, some VTs may merge before the separation frequency, and thus be associated with a more or less random HF RT. We have also seen that VTs associated with a too short HF RT tend to

produce excess noise. We therefore decided to run some VTs directly in one band, ignoring their RT association.

This step is also an opportunity to add comments, format definitions and other useful informative parameters to the files.

The python script `bankReformat.py` conducts this formatting step. It also removes the unused templates from the RT banks.

It takes as input the four ASCII files output from the match step and some parameters that define the 1-band bank. It produces the following five ASCII files (ids always start from 0, parameters in brackets () are optional for MBTA):

- a 2-bands VT bank, sorted by associated HF RT duration, and then LF RT duration for them that share the same HF RT. This file uses the format: $\text{id}_{\text{VT}} m_1 m_2 \chi_1^z \chi_2^z$ (VT duration) (VT peak frequency) (VT ringdwon dominant frequency)
- a 1-band VT bank, sorted by VT duration, containing all VTs with a peak frequency below f_{cut} OR associated with a LF RT whose peak frequency is less than f_{cut} OR associated with a HF RT whose duration is smaller than 20 ms. This file uses the same format as the 2-bands VT bank
- two RT banks, sorted by RT duration. This file uses the format: $\text{id}_{\text{RT}} m_1 m_2 \chi_1^z \chi_2^z$ (RT duration) (RT peak frequency) (RT ringdwon dominant frequency)
- an association file, which keeps the same order as the 2-band VT bank. This file uses the format: $\text{id}_{\text{VT}} \text{id}_{\text{LF}} \text{id}_{\text{HF}}$

It has been measured that this way of ordering banks reduces the CPU usage by about 10%. Another 10% is saved by distributed the VTs linearly across the jobs, and not uniformly as during O3. In order to distrute the HF RTs more evenly, the first job handles 3 times fewer VTs than the last (this number is set in the pipeline configuration).

4 Region 4

In the past, it has been shown that the MBTA gating procedure can be unsafe for some intense and short CBC signals. To solve this issue, it has been proposed to run a small part of the VT bank twice: once with the regular search (with gating), and once without gating but with higher SNR thresholds. This part of the parameter space is historically called "region 4" [1, 9].

For simplicity, and because it only concerns a few templates that are relatively short, region 4 is run in a single band.

Using the macros in the "region 4" directory, we defined the O4 region 4 as all VTs with a duration of less than 6 sec (non-constraining limitation that is used to speed up the process) and triggering the gating with an expected SNR of 25 or less (assuming the O4 L1 high-sensitivity curve).

Finally, we end up with three VT banks and two RT banks containing the number of templates given in Table 2.

The (m_1, m_2) and $(\mathcal{M}_c, \chi_{\text{eff}})$ scatter plots of these VT banks are shown in Figure 1, together with the identified events of O3. In Figure 1a, the two seeds appear as isolated parts of space (m_1, m_2) . But we can see in Figure 1b that there is no discontinuity in the space $(\mathcal{M}_c, \chi_{\text{eff}})$.

	2-bands bank	1-band bank	Region 4 (redundant ungated search on 1 band)
Number of VTs	815555	9381	904
Number of LF RTs	54071		—
Number of HF RTs	19168		—

Table 2: Number of templates used by the MBTA online analysis during O4

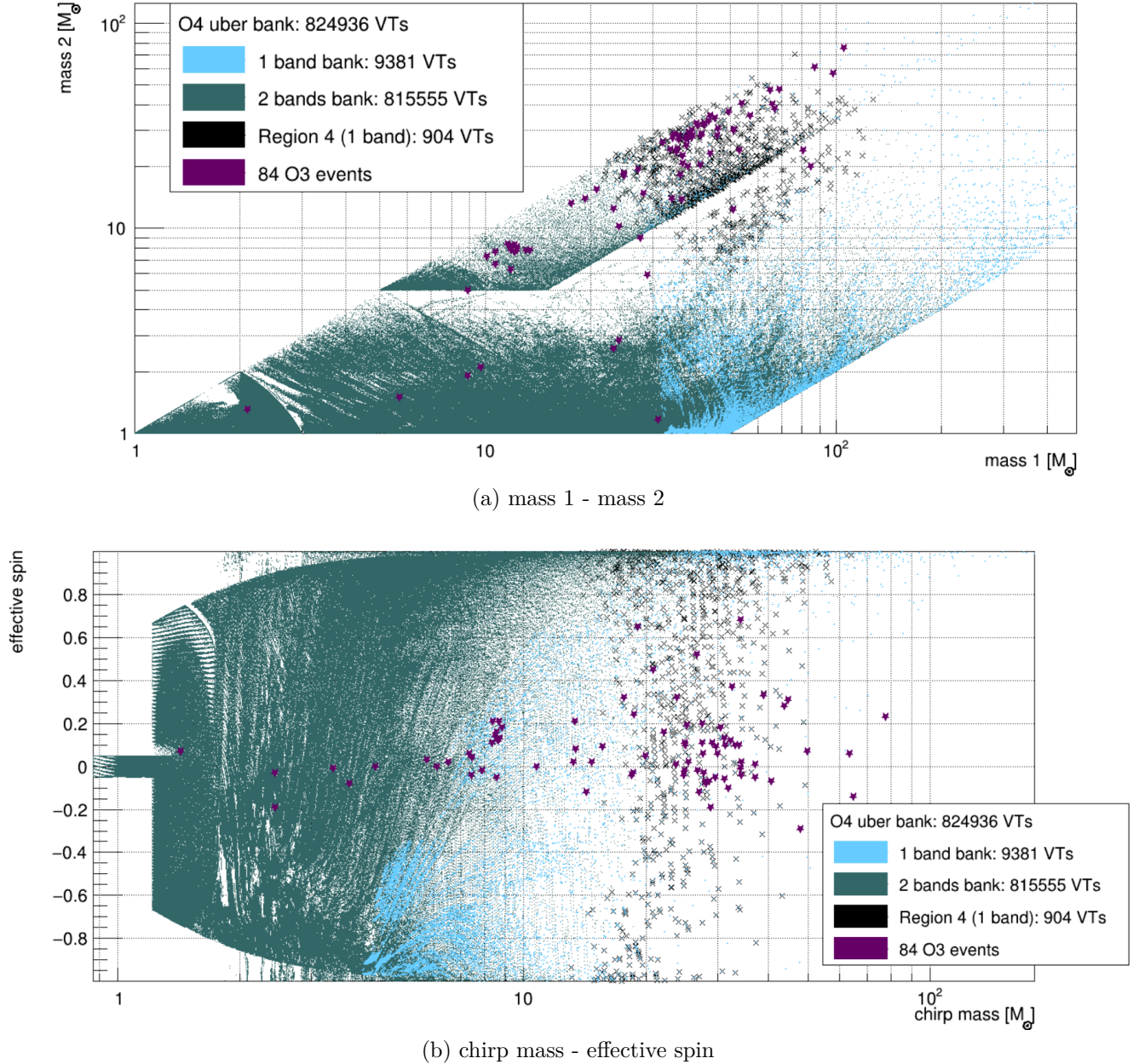


Figure 1: Scatter plots of the template placement of the O4 MBTA uber VT banks. The chirp mass is defined as: $\mathcal{M}_c = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}$, and the effective spin as: $\chi_{eff} = \frac{m_1 \chi_1^z + m_2 \chi_2^z}{m_1 + m_2}$

5 Bank verification

There are several ways to check that the bank is correct.

One is to measure the effective fitting factor. To do this, we can use the PyCBC's "banksim" algorithm discussed earlier in the section 2. This algorithm will create a set of random injections, and for each of them will calculate the match on each template of the bank [8].

In order to speed up the process, the PyCBC team has created another algorithm (which actually uses banksim), called "bank-vericator". Using it, we measured the fitting factor of our VT bank across different parameter spaces.

5.1 Measurement of the effective fitting factor on the targeted space

We applied this bank-vericator algorithm to our VT banks. We used three sets of injections in order to cover the entire targeted space. The parameters of these sets are defined in Table 3.

	BNS	NSBH	BBH
Number of injections	2000	5000	10000
Individual mass m_1 [M_\odot]	Uniform [1, 2]	Logarithmic [2, 499]	Logarithmic [2, 499]
Individual mass m_2 [M_\odot]	Uniform [1, 2]	Uniform [1, 2]	Logarithmic [2, 250]
Total mass ($m_1 + m_2$) [M_\odot]	[2, 4]	[3, 500]	[4, 500]
Mass ratio ($m_1/m_2 \geq 1$)	[1, 2]	[1, 50]	[1, 50]
Individual aligned spin χ_1^z (uniform)	[-0.05, 0.05]	[-0.997, 0.997]	[-0.997, 0.997]
Individual aligned spin χ_2^z (uniform)	[-0.05, 0.05]	[-0.05, 0.05]	[-0.997, 0.997]
Waveform approximant	TaylorF2	SEOBNRv4_ROM	SEOBNRv4_ROM
Generation low frequency cut-off [Hz]	24	17	17
Filtering low frequency cut-off [Hz]	25	18	18

Table 3: Parameters of the injections used to verify the VT bank fitting factor.

Figure 2 shown the best matches measured for these injections, as a function of their individual masses (Figure 2a) and their chirp mass and effective spin (Figure 2b).

We can see from these plots that both seeds are correctly covered with a high minimum match. However, the high mass and low effective spin signals do not seem to be recovered as expected. This is due to the minimum duration limit of 200 ms that we imposed when creating the VT bank. This cut-off is necessary to limit the noise produced by such short templates. We are currently studying new banks with finer cuts.

Figure 3 shown the cumulative density of injections that are recovered with a given minimum match threshold. Again, the seeds appear to be well covered.

Overall, about 97% of the injections with a duration of more than 200 ms are recovered with a match above the usual limit of 0.97.

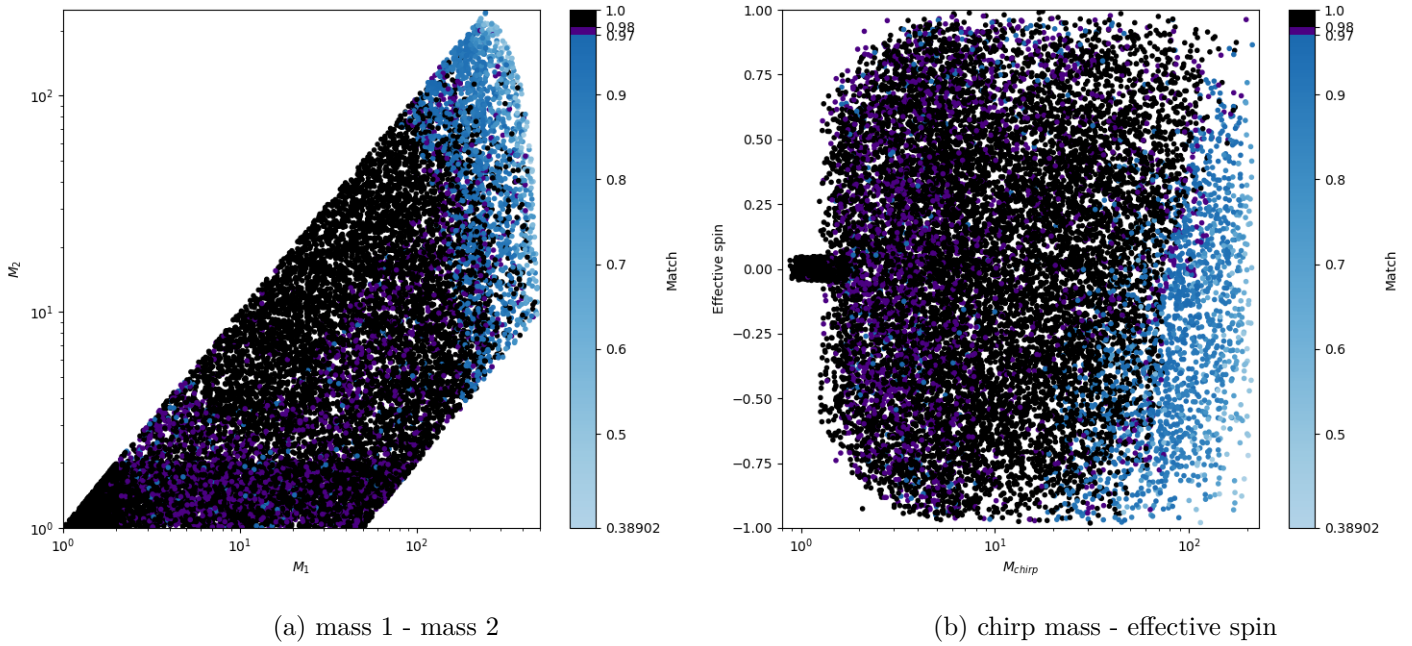


Figure 2: Scatter plots of the measured match of the O4 MBTA uber VT banks. These plots show the best match (in colour) measured for given simulated signals (injections) from the considered banks.

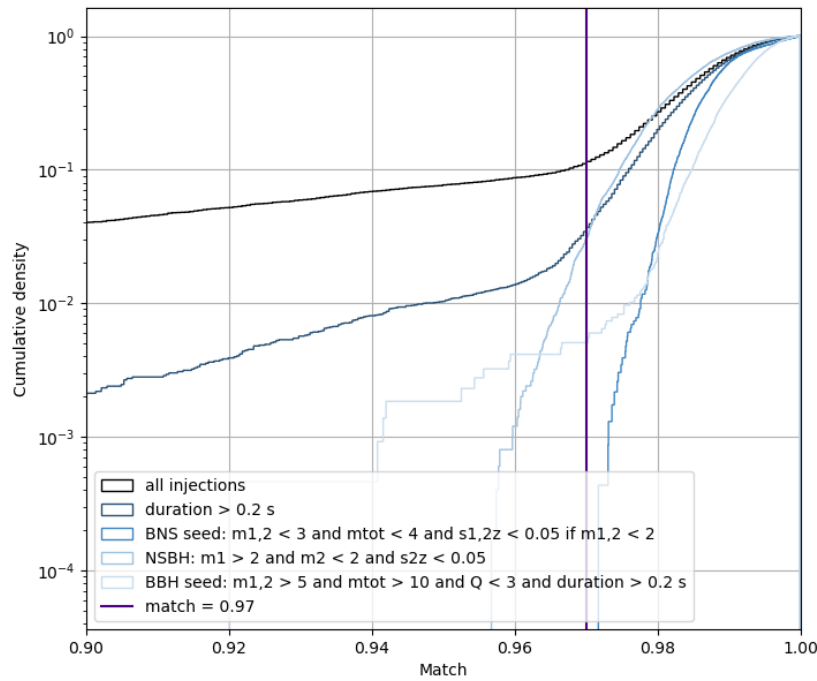


Figure 3: Cumulative density of injections that are recovered with a given match threshold.

5.2 Measurement of the fitting factor on a larger parameter space

In order to estimate the capacity of the banks to recover signals outside of the targeted parameter space, we run other bank-verificators.

5.2.1 Spinning BNS

Our banks are made to cover absolute spin values greater than 0.05 only if the corresponding component mass is greater than $2 M_{\odot}$.

We therefore measure the effective fitting factor of spinning BNS-like signals, with component masses between $1 M_{\odot}$ and $2 M_{\odot}$, and spins between -0.97 and 0.97 . These parameters are summarised in Table 4.

	BNS
Number of injections	5000
Individual mass m_1 [M_{\odot}]	Uniform [1, 2]
Individual mass m_2 [M_{\odot}]	Uniform [1, 2]
Individual aligned spin χ_1^z (uniform)	$[-0.997, 0.997]$
Individual aligned spin χ_2^z (uniform)	$[-0.997, 0.997]$
Waveform approximant	TaylorF2
Generation low frequency cut-off [Hz]	24
Filtering low frequency cut-off [Hz]	25

Table 4: Parameters of the spinning BNS injections used to measure the VT bank fitting factor.

Figure 4 shows the two (m_1, m_2) and $(\mathcal{M}_c, \chi_{eff})$ scatter plots obtained with this set of spinning BNS injections. Figure 5 shows the corresponding cumulative density of injections recovered as a function of a threshold on the match.

On these plots, we can see that the VT banks presented here recover successfully all spinning BNS with a chirp mass above $1.218... M_{\odot}$ (which corresponds to two typical individual masses of $1.4 M_{\odot}$). This is because the rest of the banks, and especially the part of the BNS seed that covers the highly spinning objects with masses between $2 M_{\odot}$ and $3 M_{\odot}$, also covers all spinning CBC with a chirp mass above $1.2 M_{\odot}$.

For smaller chirp masses, we can see that the fitting factor drops rapidly. However, the search is not completely blind to low chirp mass spinning BNS. We could expect to detect them if their SNR is high enough.

Moreover, these BNS could also be recovered by the online SSM search that we want to run during O4. We are currently investigating a way to measure the overlap between the two searches.

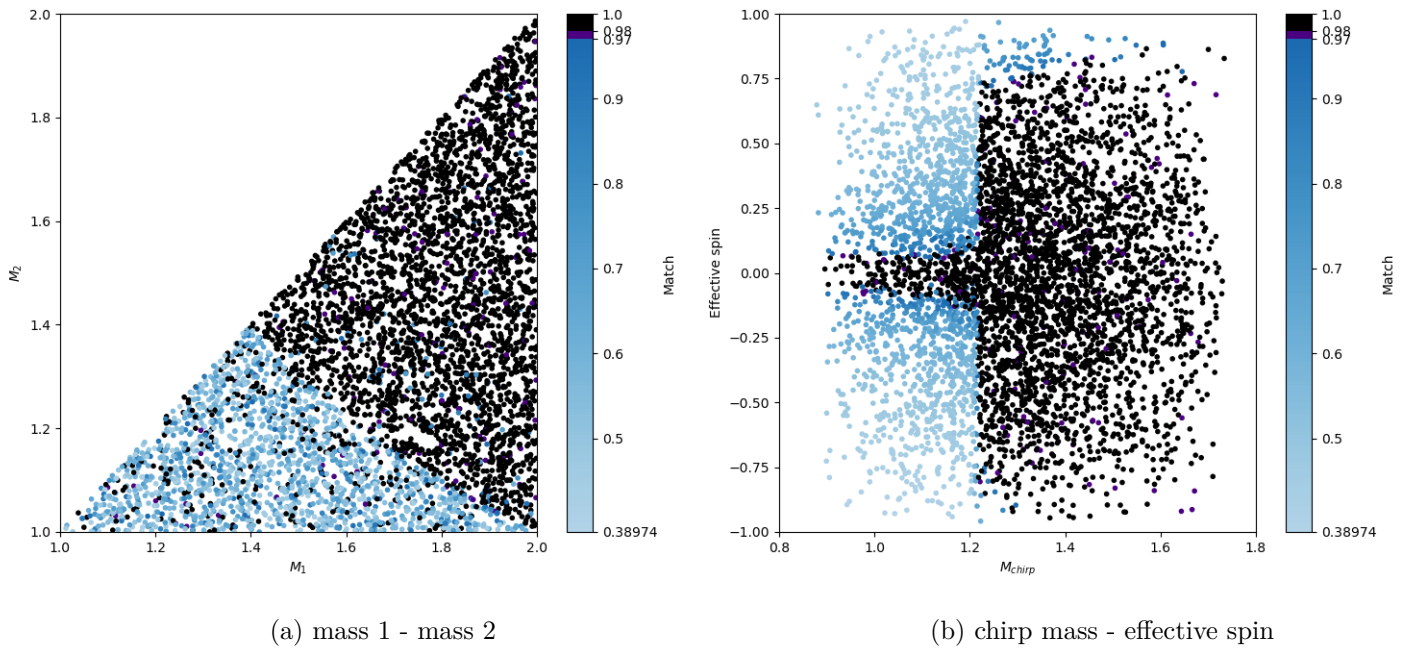


Figure 4: Scatter plots of the measured match of the O4 MBTA uber VT banks, on spinning BNS injections.

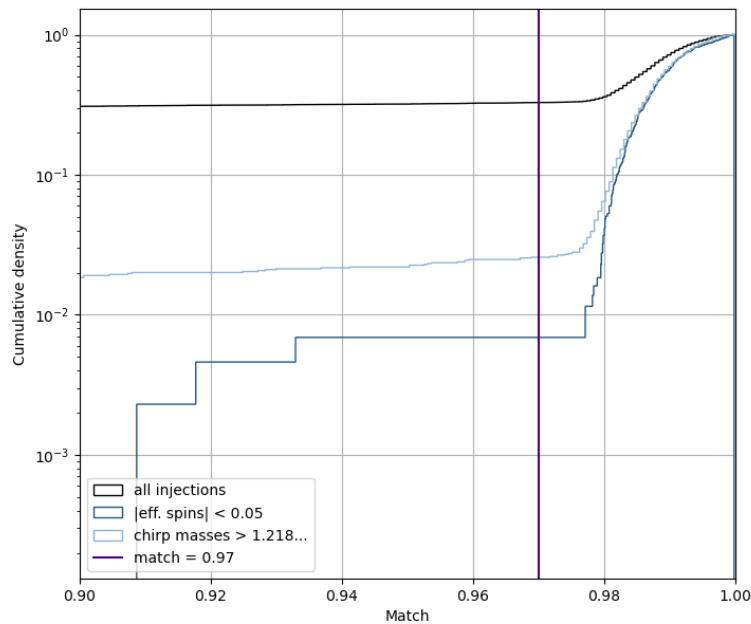


Figure 5: Cumulative density of spinning BNS injections that are recovered with a given match threshold.

5.2.2 Higher total mass and mass ratio

The upper limits on the total mass (500 M_\odot) and the mass ratio (50) have been set arbitrarily, knowing that little or no astrophysical signals are expected at higher values.

We therefore measured the ability of our banks to recover CBC involving higher total masses and mass ratio by running a bank-vericator on two sets of injections similar to those named "NSBH" and "BBH" in Table 3, but with a total mass upper limit of 600 M_\odot , a mass ratio upper limit of 97, and an individual mass m_1 upper limit of 599 M_\odot . These parameters are summarised in Table 5.

	NSBH	BBH
Number of injections	5000	10000
Individual mass m_1 [M_\odot]	Logarithmic [2, 599]	Logarithmic [2, 599]
Individual mass m_2 [M_\odot]	Uniform [1, 2]	Logarithmic [2, 250]
Total mass ($m_1 + m_2$) [M_\odot]	[3, 600]	[4, 600]
Mass ratio ($m_1/m_2 \geq 1$)	[1, 97]	[1, 97]
Individual aligned spin χ_1^z (uniform)	[-0.997, 0.997]	[-0.997, 0.997]
Individual aligned spin χ_2^z (uniform)	[-0.05, 0.05]	[-0.997, 0.997]
Waveform approximant	SEOBNRv4_ROM	SEOBNRv4_ROM
Generation low frequency cut-off [Hz]	17	17
Filtering low frequency cut-off [Hz]	18	18

Table 5: High mass and high mass ratio injection parameters used to measure the VT bank fitting factor.

Results are shown on Figure 6 and Figure 7. We can see that the higher the total mass or the mass ratio, the greater the SNR loss.

However, the minimum match is about 40%, which means that highly asymmetric heavy CBCs can still be detected by the pipeline, provided their SNR is high enough.

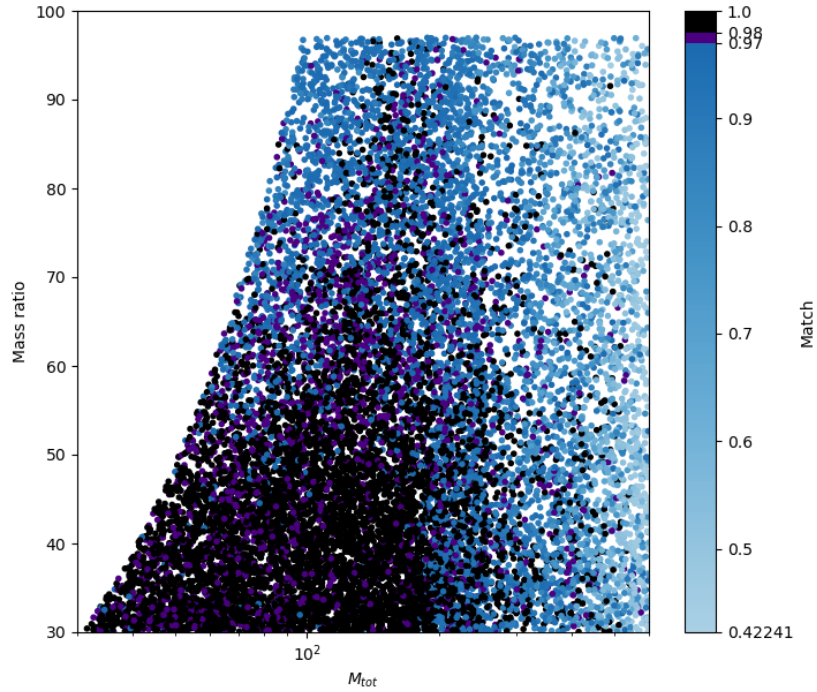


Figure 6: Total mass (M_{tot}) - mass ratio scatter plot of the measured match of the O4 MBTA uber VT banks, on high mass and high mass ratio injections.

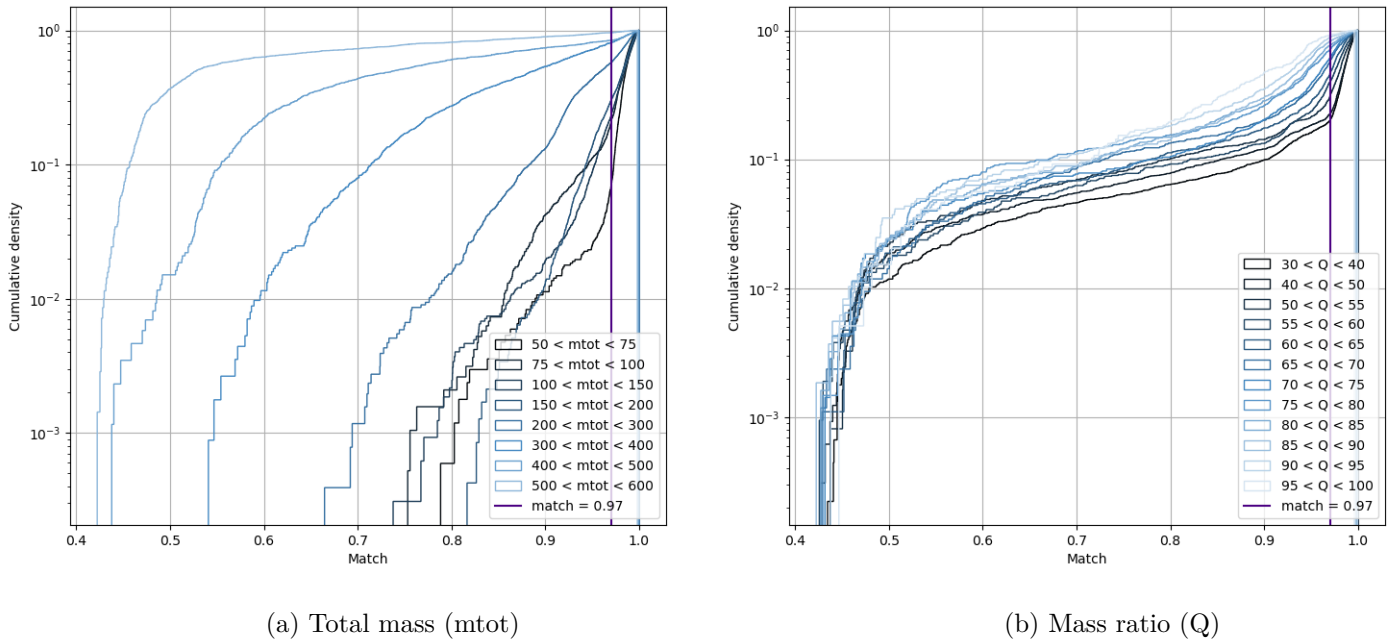


Figure 7: Cumulative densities of high mass and high mass ratio injections that are recovered with a given match threshold, depending on the total mass or the mass ratio.

References

- [1] F Aubin, F Brighenti, R Chierici, D Estevez, G Greco, G M Guidi, V Juste, F Marion, B Mours, E Nitoglia, O Sauter, and V Sordini. The MBTA pipeline for detecting compact binary coalescences in the third LIGO–virgo observing run. *Classical and Quantum Gravity*, 38(9):095004, apr 2021.
- [2] Duncan A. Brown, Ian Harry, Andrew Lundgren, and Alexander H. Nitz. Detecting binary neutron star systems with spin in advanced gravitational-wave detectors. *Physical Review D*, 86(8), oct 2012.
- [3] Collin Capano, Ian Harry, Stephen Privitera, and Alessandra Buonanno. Implementing a search for gravitational waves from binary black holes with nonprecessing spin. *Physical Review D*, 93(12), jun 2016.
- [4] Soumen Roy, Anand S. Sengupta, and Nilay Thakor. Hybrid geometric-random template-placement algorithm for gravitational wave searches from compact binary coalescences. *Physical Review D*, 95(10), may 2017.
- [5] Soumen Roy, Anand S. Sengupta, and Parameswaran Ajith. Effectual template banks for upcoming compact binary searches in advanced-LIGO and virgo data. *Physical Review D*, 99(2), jan 2019.
- [6] Roy et al. Hybrid template bank Git. <https://git.ligo.org/soumen.roy/hdbank/-/tree/master>.
- [7] P. Ajith. Addressing the spin question in gravitational-wave searches: Waveform templates for inspiralling compact binaries with nonprecessing spins. *Physical Review D*, 84(8), oct 2011.
- [8] The PyCBC team. PyCBC banksim documentation. <https://pycbc.org/pycbc/latest/html/banksim.html>.
- [9] The MBTA team. Safety tests on new $h(t)$ gating technique for MBTA. <https://tds.virgo-gw.eu/q1/?c=13832>. VIR-0009B-19.