An Issue with PE Results Using Bilby and TD Approximants

(And Resolution)

Mark Hannam, Charlie Hoy, Jonathan Thompson

CBC call: 15/02/2022
DCC: G2200203

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

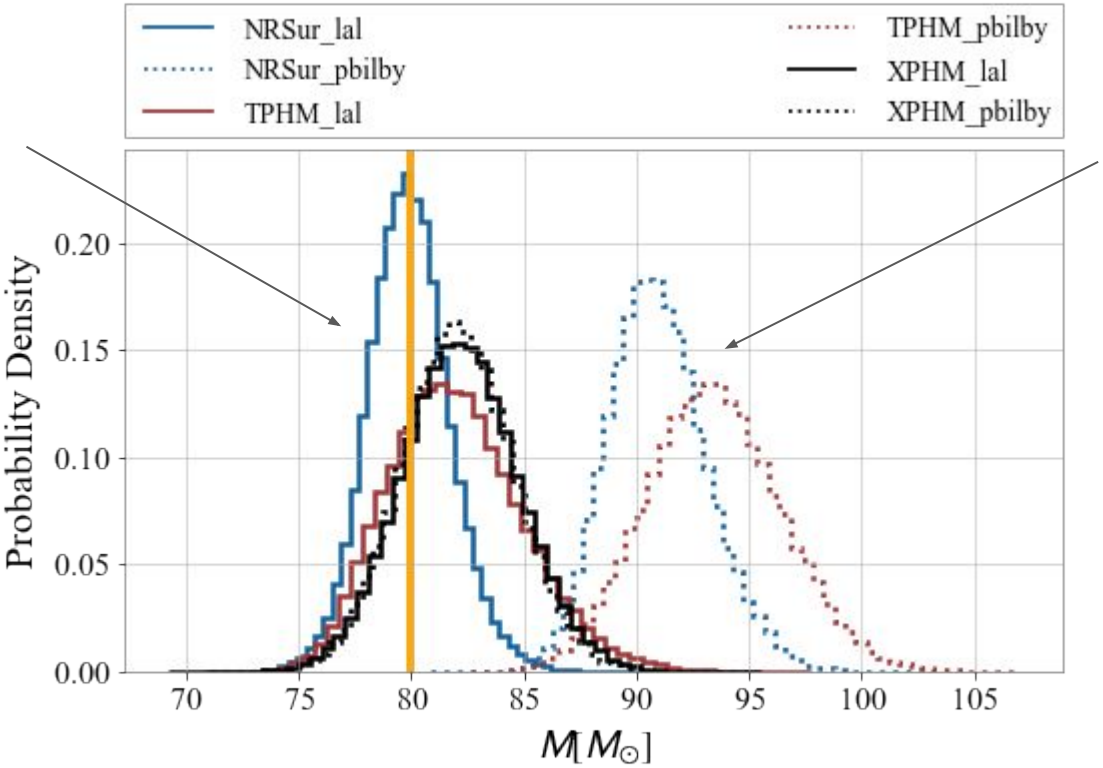
Setup

Our study involved performing parameter estimation on a set of NR injection files. We,

- Used PyCBC to generate NR injection gwfs for H1, L1, V1 in zero-noise
- Used LALInference MCMC and pBilby for the inference. Use the same settings where possible
- Used f_low = 20Hz and f_final = 896Hz to remain consistent with GWTC-3
- Recovered with IMRPhenomXPHM, IMRPhenomTPHM and NRSur7dq4 using only l<4 multipoles

Zero-noise NR injection

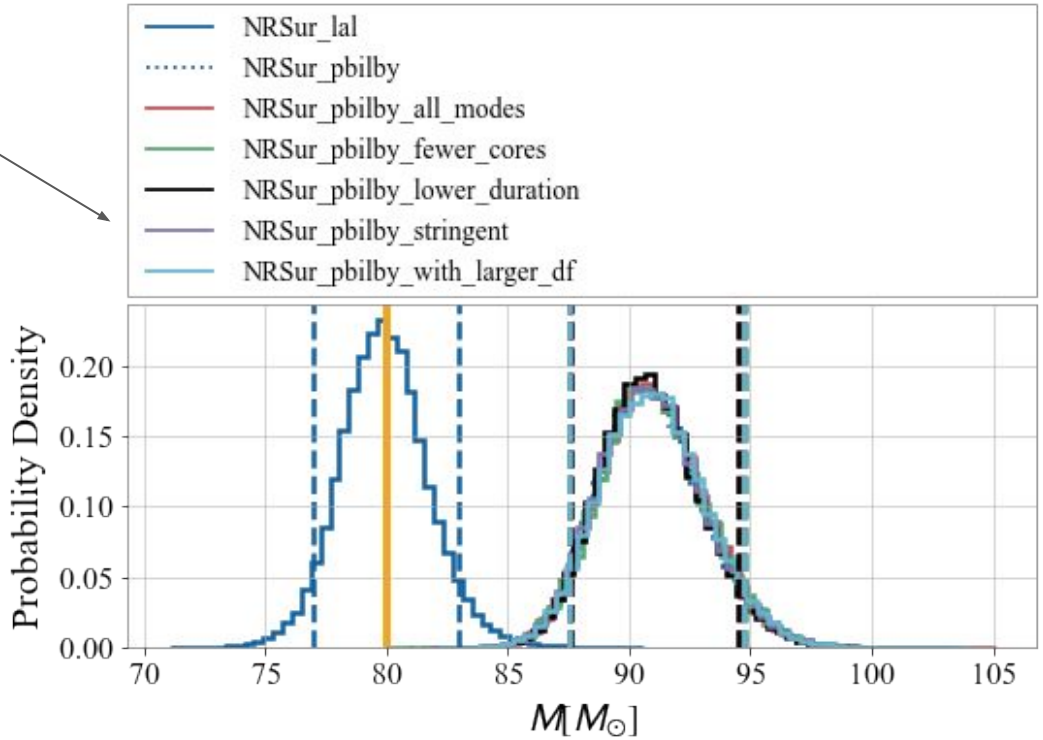
Excellent agreement between pBilby and LALInference for FD models (IMRPhenomXPHM). This is expected from the pBilby review: see e.g. [this page]



See bias in the total mass from pBilby with TD models. Only LALInference recovers the injected parameters

Tested various settings, all gave similar posteriors

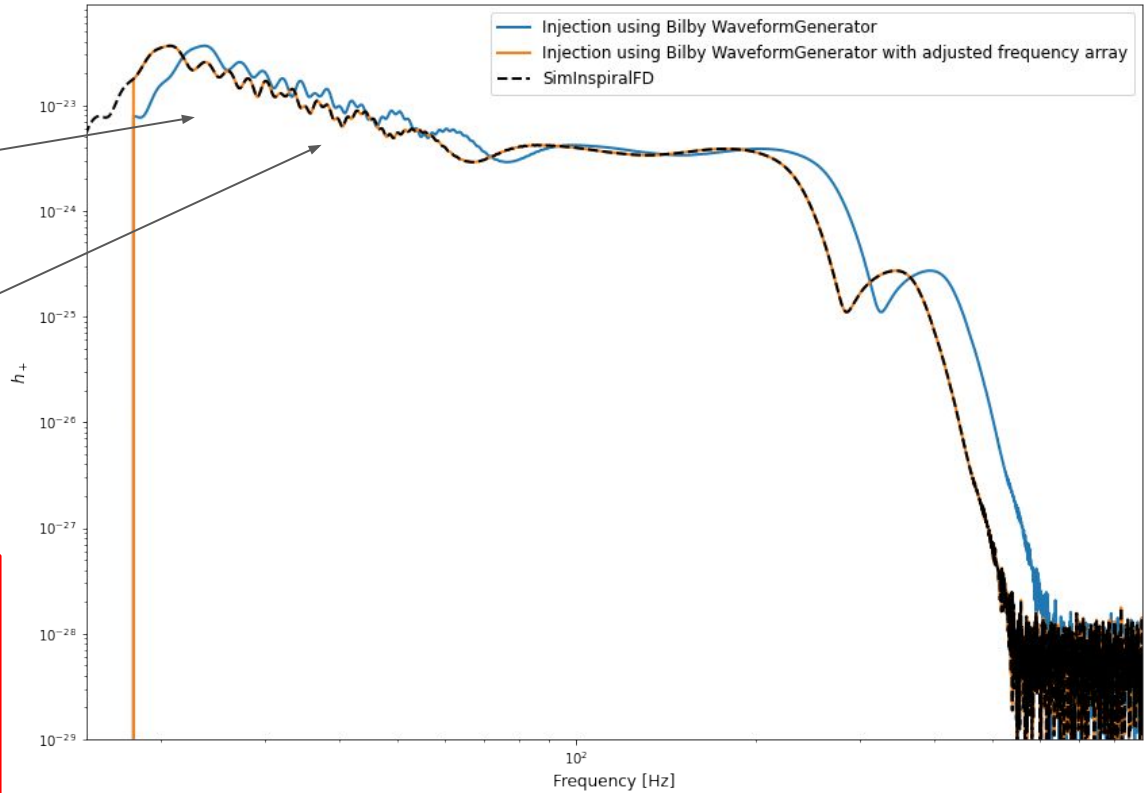
Increased number of live points from 1500 to 2000, increased nact from 20 to 50, increased maxmcmc from 5000 to 20000. Not a convergence issue.



Compare waveforms generated with Bilby and LALSimInspiralFD

When plotting NR injection, we see a difference between Bilby and LALSimInspiralFD

By adjusting the frequency array to account for change in delta_f, Bilby and LALSimInspiralFD agree



Issue resolved if f_max= sampling_frequency / 2 *or* if frequency array adjusted for change in delta_f from SimInspiralFD

Issue explained

Issue resolved if f_max= sampling_frequency / 2 *or* if frequency array adjusted for change in delta_f from SimInspiralFD

```
108 def create_frequency_series(sampling_frequency, duration):
109     """ Create a frequency series with the correct length and spacing.
110
111     Parameters
112     ==========
113     sampling_frequency: float
114     duration: float
115
116     Returns
117     =======
118     array_like: frequency series
119
120     """
121     _check_legal_sampling_frequency_and_duration(sampling_frequency, duration)
122     number_of_samples = int(np.round(duration * sampling_frequency))
123     number_of_frequencies = int(np.round(number_of_samples / 2) + 1)
124
125     return np.linspace(start=0,
126                        stop=sampling_frequency / 2,
127                        num=number_of_frequencies)
```

Bilby assigns a frequency array according to sampling frequency and duration. This means that delta_f = 0.125Hz for our setup

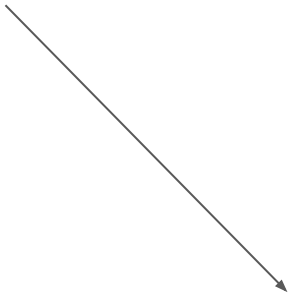
```
362 if lalsim.SimInspiralImplementedFDApproximants(approximant):
363     wf_func = lalsim_SimInspiralChooseFDWaveform
364 else:
365     wf_func = lalsim_SimInspiralFD
366 try:
367     hplus, hcross = wf_func(
368         mass_1, mass_2, spin_1x, spin_1y, spin_1z, spin_2x, spin_2y,
369         spin_2z, luminosity_distance, iota, phase,
370         longitude_ascending_nodes, eccentricity, mean_per_ano, delta_frequency,
371         start_frequency, maximum_frequency, reference_frequency,
372         waveform_dictionary, approximant)
```

Bilby passes delta_f = 0.125Hz, start_frequency = 20Hz, maximum_frequency = 896Hz, and LALSimInspiralFD outputs strain sampled *at a different* delta_f = 0.109375Hz. Bilby does not correct for this change in delta_f

Issue explained

In this example:

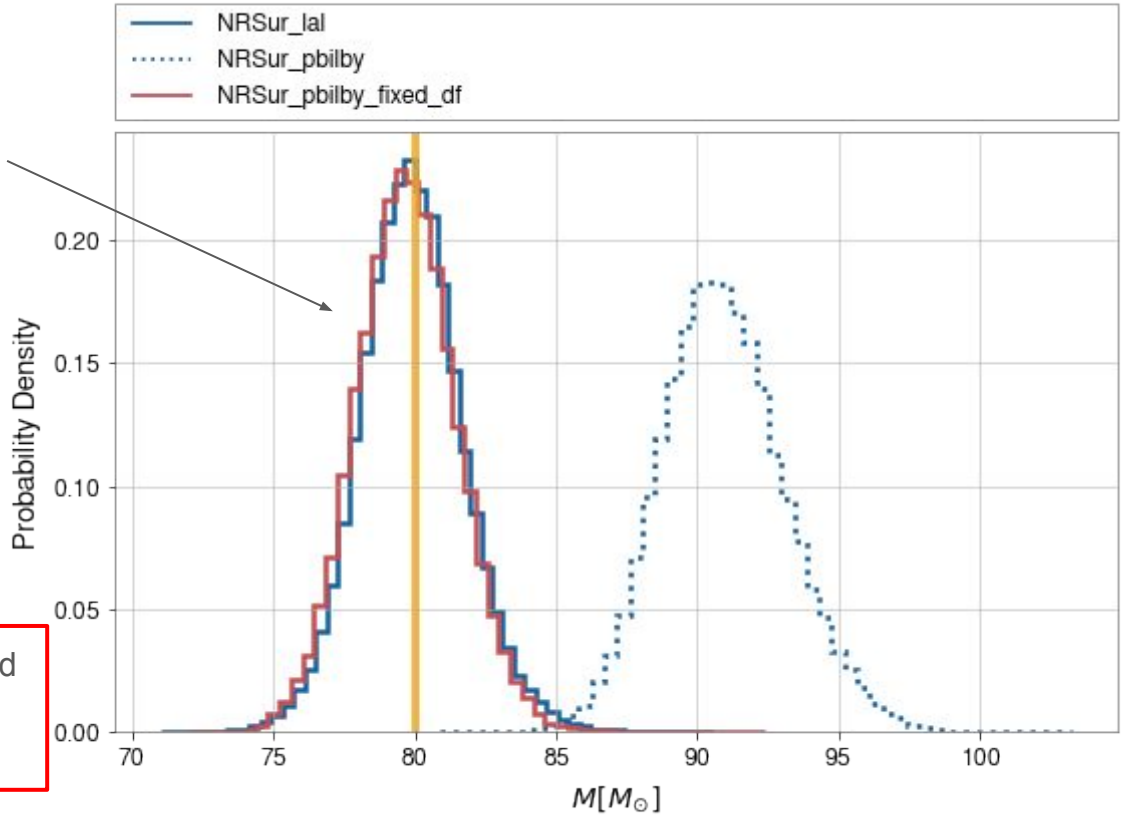
hp_eob_fmax896.deltaF=0.109375Hz
hp_eob_fmax1024.deltaF=0.125Hz



```
laldict_eob = lal.CreateDict()

q=1./0.28
Mtotal=38.4

m1 = q * Mtotal /  (1.+q)
m2 = Mtotal / (1. + q)

distance=740*1e6*lal.PC_SI

siminspiralfd_eob_params = {
    'm1':m1*lal.MSUN_SI,
    'm2':m2*lal.MSUN_SI,
    'S1x':0.1,'S1y':0.1,'S1z':0.1,
    'S2x':0.,'S2y':0.,'S2z':0.,
    'distance':distance,'inclination':0.,'longAscNodes':0.,
    'eccentricity':0.,'meanPerAno':0.,'deltaF':0.125,
    'f_min':10.,'f_ref':20.,'phiRef':0.,
    'LALparams':laldict_eob,'approximant':ls.SEOBNRv4P
}

hp_eob_fmax896, _ = ls.SimInspiralFD(f_max=896.,**siminspiralfd_eob_params)
hp_eob_fmax1024, _ = ls.SimInspiralFD(f_max=1024.,**siminspiralfd_eob_params)

print(hp_eob_fmax896.deltaF)
print(hp_eob_fmax1024.deltaF)
```

Rerun with consistent Bilby and LALSimInspiralFD

We now see excellent agreement between LALInference and pBilby for the NRSur7dq4 case. We would expect to see the same level of improvement for all TD models



Our “quick” fix is not optimised and should not be used as a solution to the problem.

Summary

- We see a difference in the inferred posteriors between LALInference and pBilby for TD models but we see excellent agreement for FD models.

- When f_final does not equal sampling_frequency / 2 there is a potential discrepancy between the delta_f used in LALSimInspiralFD and the delta_f used in the Bilby frequency array.

- Correcting for the change in delta_f produces excellent agreement between LALInference and pBilby for TD models.

- Does not affect FD models because ChooseFDWaveform outputs the data at the given input delta_f.

- Could be an issue for some O3 results if Bilby ran TD approximants. Could also be an issue in O4 for TD models if frequency spacing isn't made consistent.