Dependencies

This notebook requires Python packages listed in the `requirements.txt` file in the repository.
We recommend using a virtual environment to install them:

```
pip install -r requirements.txt
```

In [17]:
```
import bilby
import matplotlib.pyplot as plt
from bilby.core.prior import ConditionalLogUniform, LogUniform, Tru
from bilby.core.prior import PriorDict, Uniform, Constraint,Conditi
import bilby.gw.prior
import numpy as np
from BNSPriorDict_ChirpMassLambda import BNSPriorDict_chirpmass_lam
```

In [24]:
```
# Setup injection for now
# Specify the output directory and the name of the simulation.
outdir = "outdir"
# Now we try to sample and see what the error is in generating the
label = "bns_example"
bilby.core.utils.setup_logger(outdir=outdir, label=label)

# Set up a random seed for result reproducibility.  This is optiona
np.random.seed(88170235)
```

In [25]:
```
# We are going to inject a binary neutron star waveform.  We first
# dictionary of parameters that includes all of the different wavef
# parameters, including masses of the two neutron stars (mass_1, ma
# aligned spins of both NSs (chi_1, chi_2), etc.
mass_1_source = 1.5
mass_2_source = 1.3
lambda_1 = 545
lambda_2 = 1346
injection_parameters = dict(
    mass_1_source=1.5,
    mass_2_source=1.3,
    chi_1=0.02,
    chi_2=0.02,
    luminosity_distance=250.0,
    theta_jn=0.4,
    psi=2.659,
    phase=1.3,
    geocent_time=1126259642.413,
    ra=1.375,
    dec=-1.2108,
    lambda_1=545,
    lambda_2=1346,
)
```

In [26]:
```
# Calculate lambda_tilde
```

```
lambda_tilde = bilby.gw.conversion.lambda_1_lambda_2_to_lambda_tild
print(lambda_tilde)
chirp_mass_source = bilby.gw.conversion.component_masses_to_chirp_m
print(chirp_mass_source)
```

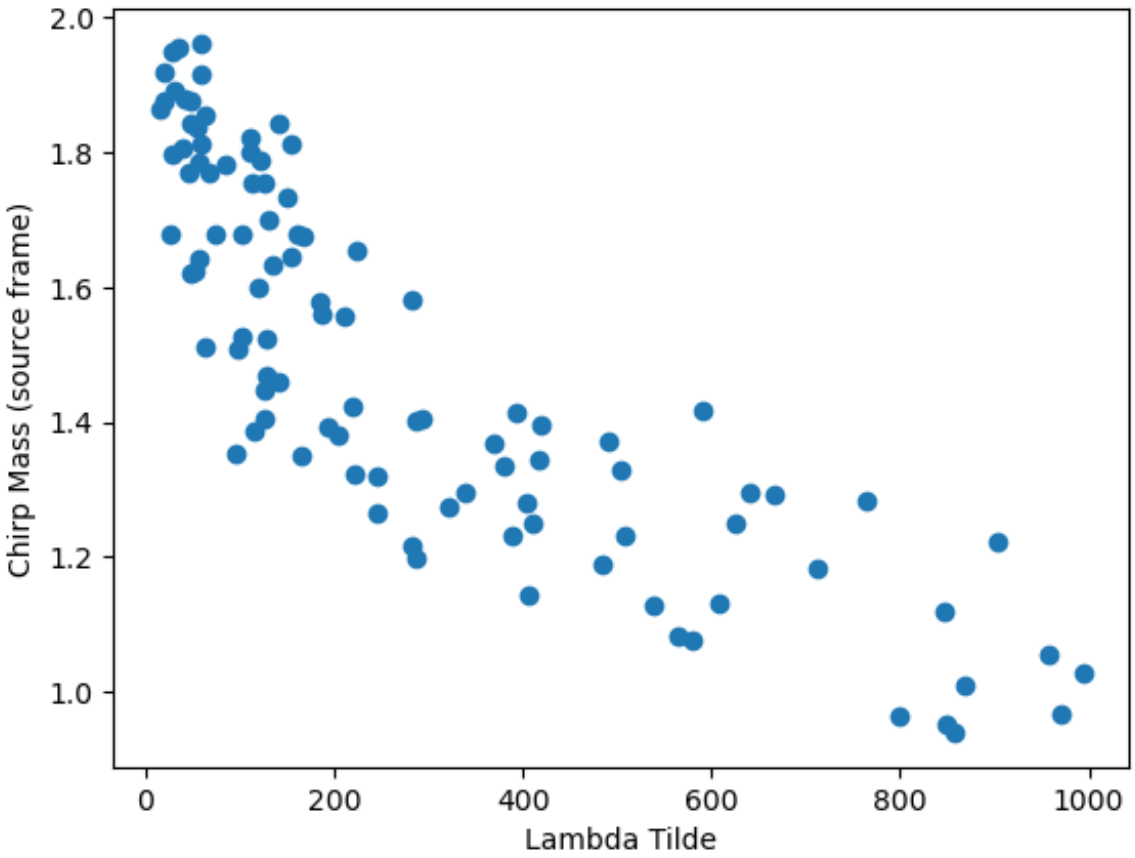
```
867.9931562541493
1.2150360414642816
```

In [27]:
```
# We setup the prior dict using the interpolated prior from the fil
priors_gw = BNSPriorDict_chirpmass_lambda_tilde(MCL_filename='./MCL_
# Delta lambda tilde is defined with a uniform prior
priors_gw['delta_lambda_tilde'] = Uniform(name='delta_lambda_tilde'
# Define the other priors for inference
priors_gw['luminosity_distance'] = 250.0 #bilby.gw.prior.UniformSou
# Fix everything to injected values other than Mchirp and lambda_ti
priors_gw['dec'] = - 1.2108 #Cosine(name='dec')
priors_gw['ra'] = 1.375 #Uniform(name='ra', minimum=0, maximum=2 *
priors_gw['theta_jn'] = 0.4 #Sine(name='theta_jn')
priors_gw['psi'] =  2.659 #Uniform(name='psi', minimum=0, maximum=n
priors_gw['phase'] = 1.3 #Uniform(name='phase', minimum=0, maximum=
priors_gw['chi_1'] = 0.02 #bilby.gw.prior.AlignedSpin(name='chi_1',
priors_gw['chi_2'] = 0#bilby.gw.prior.AlignedSpin(name='chi_2', a_p
priors_gw['mass_ratio'] =1.3/1.5 #bilby.gw.prior.UniformInComponent
priors_gw['mass_1'] = Constraint(name='mass_1', minimum=0.5, maximu
priors_gw['mass_2'] = Constraint(name='mass_2', minimum=0.5, maximu
```

```
22:00 bilby INFO    : No prior given, using default BNS priors in /U
sers/smag0001/opt/anaconda3/lib/python3.10/site-packages/bilby/gw/pr
ior_files/aligned_spins_bns_tides_on.prior.
22:00 bilby INFO    : Interpolating chirp_mass_source and lambda_til
de prior from file.
```

In [28]:
```
# Check that the prior is loaded by sampling from it
samples = priors_gw.sample(100)
# Plot the prior samples
plt.scatter(samples['lambda_tilde'], samples['chirp_mass_source'])
plt.xlabel('Lambda Tilde')
plt.ylabel('Chirp Mass (source frame)')
```

Out[28]:
```
Text(0, 0.5, 'Chirp Mass (source frame)')
```



In [29]:
```
# Fix most of the priors to their injected values
for key in [
    "psi",
    "geocent_time",
    "ra",
    "dec",
    "chi_1",
    "chi_2",
    "theta_jn",
    #"luminosity_distance",
    "phase",
]:
    priors_gw[key] = injection_parameters[key]

print(priors_gw)
```

```
{'mass_1': Constraint(minimum=0.5, maximum=5, name='mass_1', latex_l
abel='$m_1$', unit=None), 'mass_2': Constraint(minimum=0.5, maximum=
5, name='mass_2', latex_label='$m_2$', unit=None), 'mass_ratio': Del
taFunction(peak=0.8666666666666667, name=None, latex_label=None, uni
t=None), 'luminosity_distance': DeltaFunction(peak=250.0, name=None,
latex_label=None, unit=None), 'dec': -1.2108, 'ra': 1.375, 'theta_j
n': 0.4, 'psi': 2.659, 'phase': 1.3, 'chirp_mass_source': Interped(x
x=array([0.803015  , 0.80904515, 0.8150753 , 0.82110545, 0.82713561,
       0.83316576, 0.83919591, 0.84522606, 0.85125621, 0.85728636,
       0.86331652, 0.86934667, 0.87537682, 0.88140697, 0.88743712,
       0.89346727, 0.89949742, 0.90552758, 0.91155773, 0.91758788,
       0.92361803, 0.92964818, 0.93567833, 0.94170848, 0.94773864,
       0.95376879, 0.95979894, 0.96582909, 0.97185924, 0.97788939,
       0.98391955, 0.9899497 , 0.99597985, 1.00201   , 1.00804015,
       1.0140703 , 1.02010045, 1.02613061, 1.03216076, 1.03819091,
```

```
       1.04422106, 1.05025121, 1.05628136, 1.06231152, 1.06834167,
       1.07437182, 1.08040197, 1.08643212, 1.09246227, 1.09849242,
       1.10452258, 1.11055273, 1.11658288, 1.12261303, 1.12864318,
       1.13467333, 1.14070348, 1.14673364, 1.15276379, 1.15879394,
       1.16482409, 1.17085424, 1.17688439, 1.18291455, 1.1889447 ,
       1.19497485, 1.201005  , 1.20703515, 1.2130653 , 1.21909545,
       1.22512561, 1.23115576, 1.23718591, 1.24321606, 1.24924621,
       1.25527636, 1.26130652, 1.26733667, 1.27336682, 1.27939697,
       1.28542712, 1.29145727, 1.29748742, 1.30351758, 1.30954773,
       1.31557788, 1.32160803, 1.32763818, 1.33366833, 1.33969848,
       1.34572864, 1.35175879, 1.35778894, 1.36381909, 1.36984924,
       1.37587939, 1.38190955, 1.3879397 , 1.39396985, 1.4       ,
       1.40603015, 1.4120603 , 1.41809045, 1.42412061, 1.43015076,
       1.43618091, 1.44221106, 1.44824121, 1.45427136, 1.46030152,
       1.46633167, 1.47236182, 1.47839197, 1.48442212, 1.49045227,
       1.49648242, 1.50251258, 1.50854273, 1.51457288, 1.52060303,
       1.52663318, 1.53266333, 1.53869348, 1.54472364, 1.55075379,
       1.55678394, 1.56281409, 1.56884424, 1.57487439, 1.58090455,
       1.5869347 , 1.59296485, 1.598995  , 1.60502515, 1.6110553 ,
       1.61708545, 1.62311561, 1.62914576, 1.63517591, 1.64120606,
       1.64723621, 1.65326636, 1.65929652, 1.66532667, 1.67135682,
       1.67738697, 1.68341712, 1.68944727, 1.69547742, 1.70150758,
       1.70753773, 1.71356788, 1.71959803, 1.72562818, 1.73165833,
       1.73768848, 1.74371864, 1.74974879, 1.75577894, 1.76180909,
       1.76783924, 1.77386939, 1.77989955, 1.7859297 , 1.79195985,
       1.79799   , 1.80402015, 1.8100503 , 1.81608045, 1.82211061,
       1.82814076, 1.83417091, 1.84020106, 1.84623121, 1.85226136,
       1.85829152, 1.86432167, 1.87035182, 1.87638197, 1.88241212,
       1.88844227, 1.89447242, 1.90050258, 1.90653273, 1.91256288,
       1.91859303, 1.92462318, 1.93065333, 1.93668348, 1.94271364,
       1.94874379, 1.95477394, 1.96080409, 1.96683424, 1.97286439,
       1.97889455, 1.9849247 , 1.99095485, 1.996985  ]), yy=array([
9.64756405e-05, 1.48836432e-04, 3.51808936e-04, 8.65292980e-04,
       1.81238603e-03, 3.54188071e-03, 6.11792898e-03, 9.64760848e-0
3,
       1.40515240e-02, 1.97273200e-02, 2.64384924e-02, 3.50501843e-0
2,
       4.46473439e-02, 5.66487899e-02, 7.04940102e-02, 8.63700541e-0
2,
       1.04469190e-01, 1.23792773e-01, 1.44980163e-01, 1.67432436e-0
1,
       1.91198363e-01, 2.14030753e-01, 2.37699664e-01, 2.61777538e-0
1,
       2.85560583e-01, 3.10697155e-01, 3.35732633e-01, 3.60544570e-0
1,
       3.86031857e-01, 4.11840954e-01, 4.38338615e-01, 4.63928323e-0
1,
       4.89044989e-01, 5.14458131e-01, 5.39998913e-01, 5.64388951e-0
1,
       5.87314583e-01, 6.11874565e-01, 6.34785282e-01, 6.56410531e-0
1,
       6.78502821e-01, 6.99946210e-01, 7.20157821e-01, 7.41180074e-0
1,
       7.61589947e-01, 7.80633858e-01, 7.99912413e-01, 8.18106760e-0
1,
       8.36255161e-01, 8.53038258e-01, 8.70246735e-01, 8.86916902e-0
```

```
1,
        9.02519918e-01, 9.18524662e-01, 9.33808171e-01, 9.47818099e-0
1,
        9.62008016e-01, 9.75828531e-01, 9.88716050e-01, 1.00135821e+0
0,
        1.01409742e+00, 1.02507651e+00, 1.03628148e+00, 1.04732403e+0
0,
        1.05699078e+00, 1.06706918e+00, 1.07621992e+00, 1.08485528e+0
0,
        1.09300940e+00, 1.10041052e+00, 1.10728109e+00, 1.11386927e+0
0,
        1.11971092e+00, 1.12489957e+00, 1.12949873e+00, 1.13353546e+0
0,
        1.13675518e+00, 1.13959132e+00, 1.14179916e+00, 1.14345637e+0
0,
        1.14456292e+00, 1.14502294e+00, 1.14515235e+00, 1.14515588e+0
0,
        1.14514411e+00, 1.14515706e+00, 1.14515176e+00, 1.14512822e+0
0,
        1.14513587e+00, 1.14514294e+00, 1.14514235e+00, 1.14514706e+0
0,
        1.14514352e+00, 1.14514118e+00, 1.14512883e+00, 1.14514527e+0
0,
        1.14512881e+00, 1.14513055e+00, 1.14512883e+00, 1.14512355e+0
0,
        1.14513641e+00, 1.14512409e+00, 1.14512586e+00, 1.14512231e+0
0,
        1.14511589e+00, 1.14511469e+00, 1.14511996e+00, 1.14511293e+0
0,
        1.14511769e+00, 1.14511417e+00, 1.14510586e+00, 1.14510465e+0
0,
        1.14508997e+00, 1.14509057e+00, 1.14508002e+00, 1.14509118e+0
0,
        1.14509172e+00, 1.14509471e+00, 1.14507175e+00, 1.14506118e+0
0,
        1.14503998e+00, 1.14504999e+00, 1.14504002e+00, 1.14496766e+0
0,
        1.14492592e+00, 1.14478055e+00, 1.14472469e+00, 1.14453874e+0
0,
        1.14436987e+00, 1.14407588e+00, 1.14369817e+00, 1.14300814e+0
0,
        1.14219692e+00, 1.14134505e+00, 1.14011671e+00, 1.13860720e+0
0,
        1.13655672e+00, 1.13404775e+00, 1.13158582e+00, 1.12843378e+0
0,
        1.12442346e+00, 1.11994362e+00, 1.11546623e+00, 1.11046492e+0
0,
        1.10423037e+00, 1.09799754e+00, 1.09114877e+00, 1.08361811e+0
0,
        1.07503456e+00, 1.06648173e+00, 1.05764933e+00, 1.04825746e+0
0,
        1.03764666e+00, 1.02719991e+00, 1.01529721e+00, 1.00342157e+0
0,
        9.91457941e-01, 9.78701727e-01, 9.66827831e-01, 9.54000351e-0
1,
        9.41455297e-01, 9.29020804e-01, 9.16415102e-01, 9.03886781e-0
```

```
1,
       8.91736464e-01, 8.79129573e-01, 8.66800972e-01, 8.54380588e-0
1,
       8.42035546e-01, 8.29588961e-01, 8.17446865e-01, 8.05740042e-0
1,
       7.93639126e-01, 7.81705230e-01, 7.70040200e-01, 7.58055190e-0
1,
       7.46477988e-01, 7.35593551e-01, 7.23850317e-01, 7.12838906e-0
1,
       7.01280372e-01, 6.90181824e-01, 6.78889936e-01, 6.68114340e-0
1,
       6.57662911e-01, 6.46956197e-01, 6.36044211e-01, 6.25695670e-0
1,
       6.14188460e-01, 6.04007566e-01, 5.93807348e-01, 5.83640660e-0
1,
       5.73598711e-01, 5.63655616e-01, 5.53710125e-01, 5.43579985e-0
1,
       5.34454908e-01, 5.24590669e-01, 5.15486990e-01]), minimum=0.8
03015, maximum=1.996985, name=None, latex_label=None, unit=None, bou
ndary=None), 'lambda_tilde': ConditionalInterped(condition_func='BNS
PriorDict_ChirpMassLambda.conditional_func_y', name=None, latex_labe
l=None, unit=None, boundary=None, xx=array([  3.51005 ,   8.530151,
13.550251,  18.570352,  23.590452,
        28.610553,  33.630653,  38.650754,  43.670854,  48.690955,
        53.711055,  58.731156,  63.751256,  68.771357,  73.791457,
        78.811558,  83.831658,  88.851759,  93.871859,  98.89196 ,
       103.91206 , 108.932161, 113.952261, 118.972362, 123.992462,
       129.012563, 134.032663, 139.052764, 144.072864, 149.092965,
       154.113065, 159.133166, 164.153266, 169.173367, 174.193467,
       179.213568, 184.233668, 189.253769, 194.273869, 199.29397 ,
       204.31407 , 209.334171, 214.354271, 219.374372, 224.394472,
       229.414573, 234.434673, 239.454774, 244.474874, 249.494975,
       254.515075, 259.535176, 264.555276, 269.575377, 274.595477,
       279.615578, 284.635678, 289.655779, 294.675879, 299.69598 ,
       304.71608 , 309.736181, 314.756281, 319.776382, 324.796482,
       329.816583, 334.836683, 339.856784, 344.876884, 349.896985,
       354.917085, 359.937186, 364.957286, 369.977387, 374.997487,
       380.017588, 385.037688, 390.057789, 395.077889, 400.09799 ,
       405.11809 , 410.138191, 415.158291, 420.178392, 425.198492,
       430.218593, 435.238693, 440.258794, 445.278894, 450.298995,
       455.319095, 460.339196, 465.359296, 470.379397, 475.399497,
       480.419598, 485.439698, 490.459799, 495.479899, 500.5     ,
       505.520101, 510.540201, 515.560302, 520.580402, 525.600503,
       530.620603, 535.640704, 540.660804, 545.680905, 550.701005,
       555.721106, 560.741206, 565.761307, 570.781407, 575.801508,
       580.821608, 585.841709, 590.861809, 595.88191 , 600.90201 ,
       605.922111, 610.942211, 615.962312, 620.982412, 626.002513,
       631.022613, 636.042714, 641.062814, 646.082915, 651.103015,
       656.123116, 661.143216, 666.163317, 671.183417, 676.203518,
       681.223618, 686.243719, 691.263819, 696.28392 , 701.30402 ,
       706.324121, 711.344221, 716.364322, 721.384422, 726.404523,
       731.424623, 736.444724, 741.464824, 746.484925, 751.505025,
       756.525126, 761.545226, 766.565327, 771.585427, 776.605528,
       781.625628, 786.645729, 791.665829, 796.68593 , 801.70603 ,
       806.726131, 811.746231, 816.766332, 821.786432, 826.806533,
       831.826633, 836.846734, 841.866834, 846.886935, 851.907035,
```

```
       856.927136, 861.947236, 866.967337, 871.987437, 877.007538,
       882.027638, 887.047739, 892.067839, 897.08794 , 902.10804 ,
       907.128141, 912.148241, 917.168342, 922.188442, 927.208543,
       932.228643, 937.248744, 942.268844, 947.288945, 952.309045,
       957.329146, 962.349246, 967.369347, 972.389447, 977.409548,
       982.429648, 987.449749, 992.469849, 997.48995 ]), yy=array([
5.80859711e-07, 8.96082414e-07, 2.11801287e-06, 5.21002822e-06,
       1.09123709e-05, 2.13253438e-05, 3.68350083e-05, 5.80859721e-0
5,
       8.46000937e-05, 1.18771643e-04, 1.59183898e-04, 2.11032716e-0
4,
       2.68814089e-04, 3.41070913e-04, 4.24427813e-04, 5.20011155e-0
4,
       6.28979020e-04, 7.45338652e-04, 8.72901097e-04, 1.00807850e-0
3,
       1.15116478e-03, 1.28863023e-03, 1.43113209e-03, 1.57612035e-0
3,
       1.71930935e-03, 1.87064810e-03, 2.02137768e-03, 2.17076135e-0
3,
       2.32421106e-03, 2.47959807e-03, 2.63915741e-03, 2.79322342e-0
3,
       2.94444169e-03, 3.09744512e-03, 3.25121715e-03, 3.39806123e-0
3,
       3.53611110e-03, 3.68397881e-03, 3.82191534e-03, 3.95211288e-0
3,
       4.08512269e-03, 4.21422588e-03, 4.33591242e-03, 4.46250083e-0
3,
       4.58538103e-03, 4.70003711e-03, 4.81610640e-03, 4.92564799e-0
3,
       5.03491336e-03, 5.13595816e-03, 5.23958139e-03, 5.33994624e-0
3,
       5.43388609e-03, 5.53024497e-03, 5.62226171e-03, 5.70661038e-0
3,
       5.79205693e-03, 5.87526509e-03, 5.95285587e-03, 6.02896971e-0
3,
       6.10566809e-03, 6.17176916e-03, 6.23923034e-03, 6.30572460e-0
3,
       6.36392396e-03, 6.42460250e-03, 6.47969567e-03, 6.53168614e-0
3,
       6.58077942e-03, 6.62533916e-03, 6.66671121e-03, 6.70637619e-0
3,
       6.74154645e-03, 6.77278542e-03, 6.80047545e-03, 6.82477946e-0
3,
       6.84416744e-03, 6.86124262e-03, 6.87453505e-03, 6.88451238e-0
3,
       6.89117457e-03, 6.89394426e-03, 6.89472349e-03, 6.89474473e-0
3,
       6.89467385e-03, 6.89475183e-03, 6.89471990e-03, 6.89457822e-0
3,
       6.89462426e-03, 6.89466685e-03, 6.89466327e-03, 6.89469166e-0
3,
       6.89467034e-03, 6.89465623e-03, 6.89458188e-03, 6.89468084e-0
3,
       6.89458171e-03, 6.89459223e-03, 6.89458187e-03, 6.89455009e-0
3,
       6.89462750e-03, 6.89455332e-03, 6.89456398e-03, 6.89454259e-0
```

```
3,
       6.89450396e-03, 6.89449673e-03, 6.89452846e-03, 6.89448616e-0
3,
       6.89451481e-03, 6.89449358e-03, 6.89444357e-03, 6.89443630e-0
3,
       6.89434791e-03, 6.89435153e-03, 6.89428801e-03, 6.89435519e-0
3,
       6.89435845e-03, 6.89437643e-03, 6.89423820e-03, 6.89417455e-0
3,
       6.89404695e-03, 6.89410718e-03, 6.89404717e-03, 6.89361145e-0
3,
       6.89336017e-03, 6.89248492e-03, 6.89214865e-03, 6.89102912e-0
3,
       6.89001244e-03, 6.88824214e-03, 6.88596802e-03, 6.88181351e-0
3,
       6.87692948e-03, 6.87180073e-03, 6.86440540e-03, 6.85531740e-0
3,
       6.84297017e-03, 6.82786442e-03, 6.81304198e-03, 6.79406471e-0
3,
       6.76992012e-03, 6.74294863e-03, 6.71599213e-03, 6.68587604e-0
3,
       6.64833963e-03, 6.61081403e-03, 6.56958011e-03, 6.52424099e-0
3,
       6.47256252e-03, 6.42106043e-03, 6.36788363e-03, 6.31133840e-0
3,
       6.24745444e-03, 6.18455861e-03, 6.11289671e-03, 6.04139779e-0
3,
       5.96935715e-03, 5.89255629e-03, 5.82106802e-03, 5.74383830e-0
3,
       5.66830901e-03, 5.59344547e-03, 5.51755104e-03, 5.44211001e-0
3,
       5.36895744e-03, 5.29305581e-03, 5.21882976e-03, 5.14405107e-0
3,
       5.06972607e-03, 4.99477723e-03, 4.92167416e-03, 4.85119169e-0
3,
       4.77833639e-03, 4.70648663e-03, 4.63625580e-03, 4.56409816e-0
3,
       4.49438444e-03, 4.42885339e-03, 4.35815145e-03, 4.29185579e-0
3,
       4.22226589e-03, 4.15544561e-03, 4.08744994e-03, 4.02257417e-0
3,
       3.95964997e-03, 3.89518867e-03, 3.82949143e-03, 3.76718697e-0
3,
       3.69790556e-03, 3.63659989e-03, 3.57518804e-03, 3.51397808e-0
3,
       3.45351912e-03, 3.39365532e-03, 3.33377717e-03, 3.27278670e-0
3,
       3.21783878e-03, 3.15844952e-03, 3.10363966e-03])), 'delta_lam
bda_tilde': Uniform(minimum=-1000, maximum=1000, name='delta_lambda_
tilde', latex_label='$\\delta\\tilde{\\Lambda}$', unit=None, boundar
y=None), 'chi_1': 0.02, 'chi_2': 0.02, 'geocent_time': 1126259642.41
3}
```

In [31]:
```
# Set the duration and sampling frequency of the data segment that
# to inject the signal into. For the
# TaylorF2 waveform, we cut the signal close to the isco frequency
```

```
duration = 32
minimum_frequency=40
sampling_frequency = 2048
start_time = injection_parameters["geocent_time"] + 2 - duration

# Fixed arguments passed into the source model. The analysis starts
waveform_arguments = dict(
    waveform_approximant="IMRPhenomPv2_NRTidal",
    reference_frequency=50.0,
    minimum_frequency=minimum_frequency,
)
```

In [34]:
```
# Create the waveform_generator using a LAL Binary Neutron Star sou
waveform_generator = bilby.gw.WaveformGenerator(
    duration=duration,
    sampling_frequency=sampling_frequency,
    frequency_domain_source_model=bilby.gw.source.lal_binary_neutro
    parameter_conversion=convert_to_lal_binary_neutron_star_paramet
    waveform_arguments=waveform_arguments,
)
```

```
22:01 bilby INFO    : Waveform generator initiated with
  frequency_domain_source_model: bilby.gw.source.lal_binary_neutron_
star
  time_domain_source_model: None
  parameter_conversion: BNSPriorDict_ChirpMassLambda.convert_to_lal_
binary_neutron_star_parameters_mchirp
```

In [35]:
```
# Set up interferometers.  In this case we'll use three interferome
# (LIGO-Hanford (H1), LIGO-Livingston (L1), and Virgo (V1)).
# These default to their design sensitivity and start at 40 Hz.
interferometers = bilby.gw.detector.InterferometerList(["H1", "L1",
for interferometer in interferometers:
    interferometer.minimum_frequency = 40
interferometers.set_strain_data_from_power_spectral_densities(
    sampling_frequency=sampling_frequency, duration=duration, start
)
interferometers.inject_signal(
    parameters=injection_parameters, waveform_generator=waveform_ge
)
```

```
22:01 bilby INFO    : Injected signal in H1:
22:01 bilby INFO    :   optimal SNR = 7.19
22:01 bilby INFO    :   matched filter SNR = 8.42-0.86j
22:01 bilby INFO    :   mass_1_source = 1.5
22:01 bilby INFO    :   mass_2_source = 1.3
22:01 bilby INFO    :   chi_1 = 0.02
22:01 bilby INFO    :   chi_2 = 0.02
22:01 bilby INFO    :   luminosity_distance = 250.0
22:01 bilby INFO    :   theta_jn = 0.4
22:01 bilby INFO    :   psi = 2.659
22:01 bilby INFO    :   phase = 1.3
22:01 bilby INFO    :   geocent_time = 1126259642.413
22:01 bilby INFO    :   ra = 1.375
22:01 bilby INFO    :   dec = -1.2108
22:01 bilby INFO    :   lambda_1 = 545
22:01 bilby INFO    :   lambda_2 = 1346
22:01 bilby INFO    : Injected signal in L1:
22:01 bilby INFO    :   optimal SNR = 5.82
22:01 bilby INFO    :   matched filter SNR = 5.35+0.19j
22:01 bilby INFO    :   mass_1_source = 1.5
22:01 bilby INFO    :   mass_2_source = 1.3
22:01 bilby INFO    :   chi_1 = 0.02
22:01 bilby INFO    :   chi_2 = 0.02
22:01 bilby INFO    :   luminosity_distance = 250.0
22:01 bilby INFO    :   theta_jn = 0.4
22:01 bilby INFO    :   psi = 2.659
22:01 bilby INFO    :   phase = 1.3
22:01 bilby INFO    :   geocent_time = 1126259642.413
22:01 bilby INFO    :   ra = 1.375
22:01 bilby INFO    :   dec = -1.2108
22:01 bilby INFO    :   lambda_1 = 545
22:01 bilby INFO    :   lambda_2 = 1346
22:01 bilby INFO    : Injected signal in V1:
22:01 bilby INFO    :   optimal SNR = 6.17
22:01 bilby INFO    :   matched filter SNR = 7.63-0.39j
22:01 bilby INFO    :   mass_1_source = 1.5
22:01 bilby INFO    :   mass_2_source = 1.3
22:01 bilby INFO    :   chi_1 = 0.02
22:01 bilby INFO    :   chi_2 = 0.02
22:01 bilby INFO    :   luminosity_distance = 250.0
22:01 bilby INFO    :   theta_jn = 0.4
22:01 bilby INFO    :   psi = 2.659
22:01 bilby INFO    :   phase = 1.3
22:01 bilby INFO    :   geocent_time = 1126259642.413
22:01 bilby INFO    :   ra = 1.375
22:01 bilby INFO    :   dec = -1.2108
22:01 bilby INFO    :   lambda_1 = 545
22:01 bilby INFO    :   lambda_2 = 1346
```

Out[35]:
```
[{'plus': array([ 0.00000000e+00-0.00000000e+00j,  0.00000000e+00-
0.00000000e+00j,
          0.00000000e+00-0.00000000e+00j, ...,
         -8.66842781e-26-2.07358035e-26j, -8.66759549e-26-2.075487
81e-26j,
          0.00000000e+00-0.00000000e+00j]),
  'cross': array([ 0.00000000e+00+0.00000000e+00j,  0.00000000e+00
+0.00000000e+00j,
          0.00000000e+00+0.00000000e+00j, ...,
         -2.06658968e-26+8.63920388e-26j, -2.06849070e-26+8.638374
36e-26j,
          0.00000000e+00+0.00000000e+00j])},
 {'plus': array([ 0.00000000e+00-0.00000000e+00j,  0.00000000e+00-
0.00000000e+00j,
          0.00000000e+00-0.00000000e+00j, ...,
         -8.66842781e-26-2.07358035e-26j, -8.66759549e-26-2.075487
81e-26j,
          0.00000000e+00-0.00000000e+00j]),
  'cross': array([ 0.00000000e+00+0.00000000e+00j,  0.00000000e+00
+0.00000000e+00j,
          0.00000000e+00+0.00000000e+00j, ...,
         -2.06658968e-26+8.63920388e-26j, -2.06849070e-26+8.638374
36e-26j,
          0.00000000e+00+0.00000000e+00j])},
 {'plus': array([ 0.00000000e+00-0.00000000e+00j,  0.00000000e+00-
0.00000000e+00j,
          0.00000000e+00-0.00000000e+00j, ...,
         -8.66842781e-26-2.07358035e-26j, -8.66759549e-26-2.075487
81e-26j,
          0.00000000e+00-0.00000000e+00j]),
  'cross': array([ 0.00000000e+00+0.00000000e+00j,  0.00000000e+00
+0.00000000e+00j,
          0.00000000e+00+0.00000000e+00j, ...,
         -2.06658968e-26+8.63920388e-26j, -2.06849070e-26+8.638374
36e-26j,
          0.00000000e+00+0.00000000e+00j])}]
```

In [37]:
```
# Initialise the likelihood by passing in the interferometer data (
# and the waveform generator
likelihood = bilby.gw.GravitationalWaveTransient(
    interferometers=interferometers,
    waveform_generator=waveform_generator,
)
```

In [ ]:
```
# WHY DOES THIS WORK!!!!!!
priors_gw = dict(priors_gw)


#nsteps = 2000


# Run sampler.  In this case we're going to use the `nestle` sample
# For production runs use dynesty
# This will be very slow depending on your chosen signal
result = bilby.run_sampler(
```

```
    likelihood=likelihood,
    priors=priors_gw,
    sampler="nestle",
    nlive=200,
    injection_parameters=injection_parameters,
    outdir=outdir,
    label=label,
    npool=1,
)
```

```
22:03 bilby INFO    : Running for label 'bns_example', output will b
e saved to 'outdir'
22:03 bilby INFO    : Using lal version 7.3.1
22:03 bilby INFO    : Using lal git version Branch: None;Tag: lalsui
te-v7.19;Id: 84d780c102cf51ea1fdf7a1cbf0a626a5eca0d0a;;Builder: Dunc
an Macleod <duncan.macleod@ligo.org>;Repository status: CLEAN: All m
odifications committed
22:03 bilby INFO    : Using lalsimulation version 5.2.1
22:03 bilby INFO    : Using lalsimulation git version Branch: None;T
ag: lalsuite-v7.19;Id: 84d780c102cf51ea1fdf7a1cbf0a626a5eca0d0a;;Bui
lder: Duncan Macleod <duncan.macleod@ligo.org>;Repository status: CL
EAN: All modifications committed
22:03 bilby INFO    : Analysis priors:
22:03 bilby INFO    : chirp_mass_source=Interped(xx=array([0.803015
, 0.80904515, 0.8150753 , 0.82110545, 0.82713561,
       0.83316576, 0.83919591, 0.84522606, 0.85125621, 0.85728636,
       0.86331652, 0.86934667, 0.87537682, 0.88140697, 0.88743712,
       0.89346727, 0.89949742, 0.90552758, 0.91155773, 0.91758788,
       0.92361803, 0.92964818, 0.93567833, 0.94170848, 0.94773864,
       0.95376879, 0.95979894, 0.96582909, 0.97185924, 0.97788939,
       0.98391955, 0.9899497 , 0.99597985, 1.00201   , 1.00804015,
       1.0140703 , 1.02010045, 1.02613061, 1.03216076, 1.03819091,
       1.04422106, 1.05025121, 1.05628136, 1.06231152, 1.06834167,
       1.07437182, 1.08040197, 1.08643212, 1.09246227, 1.09849242,
       1.10452258, 1.11055273, 1.11658288, 1.12261303, 1.12864318,
       1.13467333, 1.14070348, 1.14673364, 1.15276379, 1.15879394,
       1.16482409, 1.17085424, 1.17688439, 1.18291455, 1.1889447 ,
       1.19497485, 1.201005  , 1.20703515, 1.2130653 , 1.21909545,
       1.22512561, 1.23115576, 1.23718591, 1.24321606, 1.24924621,
       1.25527636, 1.26130652, 1.26733667, 1.27336682, 1.27939697,
       1.28542712, 1.29145727, 1.29748742, 1.30351758, 1.30954773,
       1.31557788, 1.32160803, 1.32763818, 1.33366833, 1.33969848,
       1.34572864, 1.35175879, 1.35778894, 1.36381909, 1.36984924,
       1.37587939, 1.38190955, 1.3879397 , 1.39396985, 1.4       ,
       1.40603015, 1.4120603 , 1.41809045, 1.42412061, 1.43015076,
       1.43618091, 1.44221106, 1.44824121, 1.45427136, 1.46030152,
       1.46633167, 1.47236182, 1.47839197, 1.48442212, 1.49045227,
       1.49648242, 1.50251258, 1.50854273, 1.51457288, 1.52060303,
       1.52663318, 1.53266333, 1.53869348, 1.54472364, 1.55075379,
       1.55678394, 1.56281409, 1.56884424, 1.57487439, 1.58090455,
       1.5869347 , 1.59296485, 1.598995  , 1.60502515, 1.6110553 ,
       1.61708545, 1.62311561, 1.62914576, 1.63517591, 1.64120606,
       1.64723621, 1.65326636, 1.65929652, 1.66532667, 1.67135682,
       1.67738697, 1.68341712, 1.68944727, 1.69547742, 1.70150758,
       1.70753773, 1.71356788, 1.71959803, 1.72562818, 1.73165833,
       1.73768848, 1.74371864, 1.74974879, 1.75577894, 1.76180909,
```

```
       1.76783924, 1.77386939, 1.77989955, 1.7859297 , 1.79195985,
       1.79799   , 1.80402015, 1.8100503 , 1.81608045, 1.82211061,
       1.82814076, 1.83417091, 1.84020106, 1.84623121, 1.85226136,
       1.85829152, 1.86432167, 1.87035182, 1.87638197, 1.88241212,
       1.88844227, 1.89447242, 1.90050258, 1.90653273, 1.91256288,
       1.91859303, 1.92462318, 1.93065333, 1.93668348, 1.94271364,
       1.94874379, 1.95477394, 1.96080409, 1.96683424, 1.97286439,
       1.97889455, 1.9849247 , 1.99095485, 1.996985  ]), yy=array([
9.64756405e-05, 1.48836432e-04, 3.51808936e-04, 8.65292980e-04,
       1.81238603e-03, 3.54188071e-03, 6.11792898e-03, 9.64760848e-0
3,
       1.40515240e-02, 1.97273200e-02, 2.64384924e-02, 3.50501843e-0
2,
       4.46473439e-02, 5.66487899e-02, 7.04940102e-02, 8.63700541e-0
2,
       1.04469190e-01, 1.23792773e-01, 1.44980163e-01, 1.67432436e-0
1,
       1.91198363e-01, 2.14030753e-01, 2.37699664e-01, 2.61777538e-0
1,
       2.85560583e-01, 3.10697155e-01, 3.35732633e-01, 3.60544570e-0
1,
       3.86031857e-01, 4.11840954e-01, 4.38338615e-01, 4.63928323e-0
1,
       4.89044989e-01, 5.14458131e-01, 5.39998913e-01, 5.64388951e-0
1,
       5.87314583e-01, 6.11874565e-01, 6.34785282e-01, 6.56410531e-0
1,
       6.78502821e-01, 6.99946210e-01, 7.20157821e-01, 7.41180074e-0
1,
       7.61589947e-01, 7.80633858e-01, 7.99912413e-01, 8.18106760e-0
1,
       8.36255161e-01, 8.53038258e-01, 8.70246735e-01, 8.86916902e-0
1,
       9.02519918e-01, 9.18524662e-01, 9.33808171e-01, 9.47818099e-0
1,
       9.62008016e-01, 9.75828531e-01, 9.88716050e-01, 1.00135821e+0
0,
       1.01409742e+00, 1.02507651e+00, 1.03628148e+00, 1.04732403e+0
0,
       1.05699078e+00, 1.06706918e+00, 1.07621992e+00, 1.08485528e+0
0,
       1.09300940e+00, 1.10041052e+00, 1.10728109e+00, 1.11386927e+0
0,
       1.11971092e+00, 1.12489957e+00, 1.12949873e+00, 1.13353546e+0
0,
       1.13675518e+00, 1.13959132e+00, 1.14179916e+00, 1.14345637e+0
0,
       1.14456292e+00, 1.14502294e+00, 1.14515235e+00, 1.14515588e+0
0,
       1.14514411e+00, 1.14515706e+00, 1.14515176e+00, 1.14512822e+0
0,
       1.14513587e+00, 1.14514294e+00, 1.14514235e+00, 1.14514706e+0
0,
       1.14514352e+00, 1.14514118e+00, 1.14512883e+00, 1.14514527e+0
0,
       1.14512881e+00, 1.14513055e+00, 1.14512883e+00, 1.14512355e+0
```

```
0,
       1.14513641e+00, 1.14512409e+00, 1.14512586e+00, 1.14512231e+0
0,
       1.14511589e+00, 1.14511469e+00, 1.14511996e+00, 1.14511293e+0
0,
       1.14511769e+00, 1.14511417e+00, 1.14510586e+00, 1.14510465e+0
0,
       1.14508997e+00, 1.14509057e+00, 1.14508002e+00, 1.14509118e+0
0,
       1.14509172e+00, 1.14509471e+00, 1.14507175e+00, 1.14506118e+0
0,
       1.14503998e+00, 1.14504999e+00, 1.14504002e+00, 1.14496766e+0
0,
       1.14492592e+00, 1.14478055e+00, 1.14472469e+00, 1.14453874e+0
0,
       1.14436987e+00, 1.14407588e+00, 1.14369817e+00, 1.14300814e+0
0,
       1.14219692e+00, 1.14134505e+00, 1.14011671e+00, 1.13860720e+0
0,
       1.13655672e+00, 1.13404775e+00, 1.13158582e+00, 1.12843378e+0
0,
       1.12442346e+00, 1.11994362e+00, 1.11546623e+00, 1.11046492e+0
0,
       1.10423037e+00, 1.09799754e+00, 1.09114877e+00, 1.08361811e+0
0,
       1.07503456e+00, 1.06648173e+00, 1.05764933e+00, 1.04825746e+0
0,
       1.03764666e+00, 1.02719991e+00, 1.01529721e+00, 1.00342157e+0
0,
       9.91457941e-01, 9.78701727e-01, 9.66827831e-01, 9.54000351e-0
1,
       9.41455297e-01, 9.29020804e-01, 9.16415102e-01, 9.03886781e-0
1,
       8.91736464e-01, 8.79129573e-01, 8.66800972e-01, 8.54380588e-0
1,
       8.42035546e-01, 8.29588961e-01, 8.17446865e-01, 8.05740042e-0
1,
       7.93639126e-01, 7.81705230e-01, 7.70040200e-01, 7.58055190e-0
1,
       7.46477988e-01, 7.35593551e-01, 7.23850317e-01, 7.12838906e-0
1,
       7.01280372e-01, 6.90181824e-01, 6.78889936e-01, 6.68114340e-0
1,
       6.57662911e-01, 6.46956197e-01, 6.36044211e-01, 6.25695670e-0
1,
       6.14188460e-01, 6.04007566e-01, 5.93807348e-01, 5.83640660e-0
1,
       5.73598711e-01, 5.63655616e-01, 5.53710125e-01, 5.43579985e-0
1,
       5.34454908e-01, 5.24590669e-01, 5.15486990e-01]), minimum=0.8
03015, maximum=1.996985, name=None, latex_label=None, unit=None, bou
ndary=None)
22:03 bilby INFO    : lambda_tilde=ConditionalInterped(condition_fun
c='BNSPriorDict_ChirpMassLambda.conditional_func_y', name=None, late
x_label=None, unit=None, boundary=None, xx=array([  3.51005 ,   8.53
0151,  13.550251,  18.570352,  23.590452,
```

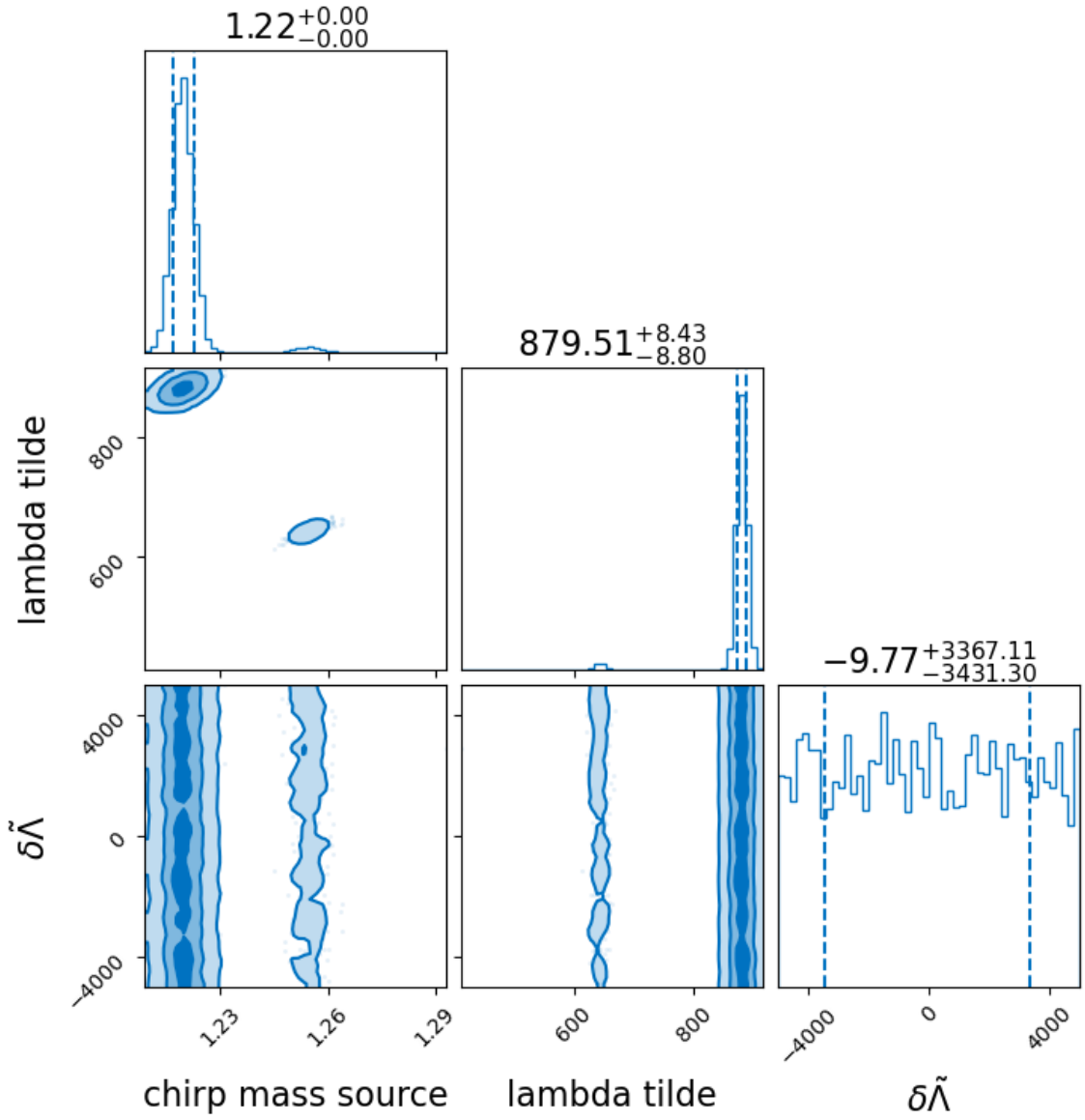
```
        28.610553,  33.630653,  38.650754,  43.670854,  48.690955,
        53.711055,  58.731156,  63.751256,  68.771357,  73.791457,
        78.811558,  83.831658,  88.851759,  93.871859,  98.89196 ,
       103.91206 , 108.932161, 113.952261, 118.972362, 123.992462,
       129.012563, 134.032663, 139.052764, 144.072864, 149.092965,
       154.113065, 159.133166, 164.153266, 169.173367, 174.193467,
       179.213568, 184.233668, 189.253769, 194.273869, 199.29397 ,
       204.31407 , 209.334171, 214.354271, 219.374372, 224.394472,
       229.414573, 234.434673, 239.454774, 244.474874, 249.494975,
       254.515075, 259.535176, 264.555276, 269.575377, 274.595477,
       279.615578, 284.635678, 289.655779, 294.675879, 299.69598 ,
       304.71608 , 309.736181, 314.756281, 319.776382, 324.796482,
       329.816583, 334.836683, 339.856784, 344.876884, 349.896985,
       354.917085, 359.937186, 364.957286, 369.977387, 374.997487,
       380.017588, 385.037688, 390.057789, 395.077889, 400.09799 ,
       405.11809 , 410.138191, 415.158291, 420.178392, 425.198492,
       430.218593, 435.238693, 440.258794, 445.278894, 450.298995,
       455.319095, 460.339196, 465.359296, 470.379397, 475.399497,
       480.419598, 485.439698, 490.459799, 495.479899, 500.5     ,
       505.520101, 510.540201, 515.560302, 520.580402, 525.600503,
       530.620603, 535.640704, 540.660804, 545.680905, 550.701005,
       555.721106, 560.741206, 565.761307, 570.781407, 575.801508,
       580.821608, 585.841709, 590.861809, 595.88191 , 600.90201 ,
       605.922111, 610.942211, 615.962312, 620.982412, 626.002513,
       631.022613, 636.042714, 641.062814, 646.082915, 651.103015,
       656.123116, 661.143216, 666.163317, 671.183417, 676.203518,
       681.223618, 686.243719, 691.263819, 696.28392 , 701.30402 ,
       706.324121, 711.344221, 716.364322, 721.384422, 726.404523,
       731.424623, 736.444724, 741.464824, 746.484925, 751.505025,
       756.525126, 761.545226, 766.565327, 771.585427, 776.605528,
       781.625628, 786.645729, 791.665829, 796.68593 , 801.70603 ,
       806.726131, 811.746231, 816.766332, 821.786432, 826.806533,
       831.826633, 836.846734, 841.866834, 846.886935, 851.907035,
       856.927136, 861.947236, 866.967337, 871.987437, 877.007538,
       882.027638, 887.047739, 892.067839, 897.08794 , 902.10804 ,
       907.128141, 912.148241, 917.168342, 922.188442, 927.208543,
       932.228643, 937.248744, 942.268844, 947.288945, 952.309045,
       957.329146, 962.349246, 967.369347, 972.389447, 977.409548,
       982.429648, 987.449749, 992.469849, 997.48995 ]), yy=array([
5.80859711e-07, 8.96082414e-07, 2.11801287e-06, 5.21002822e-06,
       1.09123709e-05, 2.13253438e-05, 3.68350083e-05, 5.80859721e-0
5,
       8.46000937e-05, 1.18771643e-04, 1.59183898e-04, 2.11032716e-0
4,
       2.68814089e-04, 3.41070913e-04, 4.24427813e-04, 5.20011155e-0
4,
       6.28979020e-04, 7.45338652e-04, 8.72901097e-04, 1.00807850e-0
3,
       1.15116478e-03, 1.28863023e-03, 1.43113209e-03, 1.57612035e-0
3,
       1.71930935e-03, 1.87064810e-03, 2.02137768e-03, 2.17076135e-0
3,
       2.32421106e-03, 2.47959807e-03, 2.63915741e-03, 2.79322342e-0
3,
       2.94444169e-03, 3.09744512e-03, 3.25121715e-03, 3.39806123e-0
3,
```

```
        3.53611110e-03, 3.68397881e-03, 3.82191534e-03, 3.95211288e-0
3,
        4.08512269e-03, 4.21422588e-03, 4.33591242e-03, 4.46250083e-0
3,
        4.58538103e-03, 4.70003711e-03, 4.81610640e-03, 4.92564799e-0
3,
        5.03491336e-03, 5.13595816e-03, 5.23958139e-03, 5.33994624e-0
3,
        5.43388609e-03, 5.53024497e-03, 5.62226171e-03, 5.70661038e-0
3,
        5.79205693e-03, 5.87526509e-03, 5.95285587e-03, 6.02896971e-0
3,
        6.10566809e-03, 6.17176916e-03, 6.23923034e-03, 6.30572460e-0
3,
        6.36392396e-03, 6.42460250e-03, 6.47969567e-03, 6.53168614e-0
3,
        6.58077942e-03, 6.62533916e-03, 6.66671121e-03, 6.70637619e-0
3,
        6.74154645e-03, 6.77278542e-03, 6.80047545e-03, 6.82477946e-0
3,
        6.84416744e-03, 6.86124262e-03, 6.87453505e-03, 6.88451238e-0
3,
        6.89117457e-03, 6.89394426e-03, 6.89472349e-03, 6.89474473e-0
3,
        6.89467385e-03, 6.89475183e-03, 6.89471990e-03, 6.89457822e-0
3,
        6.89462426e-03, 6.89466685e-03, 6.89466327e-03, 6.89469166e-0
3,
        6.89467034e-03, 6.89465623e-03, 6.89458188e-03, 6.89468084e-0
3,
        6.89458171e-03, 6.89459223e-03, 6.89458187e-03, 6.89455009e-0
3,
        6.89462750e-03, 6.89455332e-03, 6.89456398e-03, 6.89454259e-0
3,
        6.89450396e-03, 6.89449673e-03, 6.89452846e-03, 6.89448616e-0
3,
        6.89451481e-03, 6.89449358e-03, 6.89444357e-03, 6.89443630e-0
3,
        6.89434791e-03, 6.89435153e-03, 6.89428801e-03, 6.89435519e-0
3,
        6.89435845e-03, 6.89437643e-03, 6.89423820e-03, 6.89417455e-0
3,
        6.89404695e-03, 6.89410718e-03, 6.89404717e-03, 6.89361145e-0
3,
        6.89336017e-03, 6.89248492e-03, 6.89214865e-03, 6.89102912e-0
3,
        6.89001244e-03, 6.88824214e-03, 6.88596802e-03, 6.88181351e-0
3,
        6.87692948e-03, 6.87180073e-03, 6.86440540e-03, 6.85531740e-0
3,
        6.84297017e-03, 6.82786442e-03, 6.81304198e-03, 6.79406471e-0
3,
        6.76992012e-03, 6.74294863e-03, 6.71599213e-03, 6.68587604e-0
3,
        6.64833963e-03, 6.61081403e-03, 6.56958011e-03, 6.52424099e-0
3,
```

```
       6.47256252e-03, 6.42106043e-03, 6.36788363e-03, 6.31133840e-0
3,
       6.24745444e-03, 6.18455861e-03, 6.11289671e-03, 6.04139779e-0
3,
       5.96935715e-03, 5.89255629e-03, 5.82106802e-03, 5.74383830e-0
3,
       5.66830901e-03, 5.59344547e-03, 5.51755104e-03, 5.44211001e-0
3,
       5.36895744e-03, 5.29305581e-03, 5.21882976e-03, 5.14405107e-0
3,
       5.06972607e-03, 4.99477723e-03, 4.92167416e-03, 4.85119169e-0
3,
       4.77833639e-03, 4.70648663e-03, 4.63625580e-03, 4.56409816e-0
3,
       4.49438444e-03, 4.42885339e-03, 4.35815145e-03, 4.29185579e-0
3,
       4.22226589e-03, 4.15544561e-03, 4.08744994e-03, 4.02257417e-0
3,
       3.95964997e-03, 3.89518867e-03, 3.82949143e-03, 3.76718697e-0
3,
       3.69790556e-03, 3.63659989e-03, 3.57518804e-03, 3.51397808e-0
3,
       3.45351912e-03, 3.39365532e-03, 3.33377717e-03, 3.27278670e-0
3,
       3.21783878e-03, 3.15844952e-03, 3.10363966e-03]))
22:03 bilby INFO    : delta_lambda_tilde=Uniform(minimum=-1000, maxi
mum=1000, name='delta_lambda_tilde', latex_label='$\\delta\\tilde{\\
Lambda}$', unit=None, boundary=None)
22:03 bilby INFO    : mass_1=Constraint(minimum=0.5, maximum=5, name
='mass_1', latex_label='$m_1$', unit=None)
22:03 bilby INFO    : mass_2=Constraint(minimum=0.5, maximum=5, name
='mass_2', latex_label='$m_2$', unit=None)
22:03 bilby INFO    : mass_ratio=0.8666666666666667
22:03 bilby INFO    : luminosity_distance=250.0
22:03 bilby INFO    : dec=-1.2108
22:03 bilby INFO    : ra=1.375
22:03 bilby INFO    : theta_jn=0.4
22:03 bilby INFO    : psi=2.659
22:03 bilby INFO    : phase=1.3
22:03 bilby INFO    : chi_1=0.02
22:03 bilby INFO    : chi_2=0.02
22:03 bilby INFO    : geocent_time=1126259642.413
22:03 bilby INFO    : Analysis likelihood class: <class 'bilby.gw.li
kelihood.base.GravitationalWaveTransient'>
22:03 bilby INFO    : Analysis likelihood noise evidence: -94806.536
77287433
22:03 bilby INFO    : Single likelihood evaluation took nan s
22:03 bilby INFO    : Using sampler Nestle with kwargs {'method': 'm
ulti', 'npoints': 200, 'update_interval': None, 'npdim': None, 'maxi
ter': None, 'maxcall': None, 'dlogz': None, 'decline_factor': None,
'rstate': None, 'callback': <function print_progress at 0x7fdc905bc7
90>, 'steps': 20, 'enlarge': 1.2}
it=  1121 logz=-33.0272594
```

In [16]:
```
result.plot_corner()
```

Out[16]:



In [ ]:
```
# Convert the posterior using bilby's standard conversion function
result.posterior = bilby.gw.conversion.generate_all_bns_parameters(
#priors = bilby.gw.prior.BNSPriorDict()

# Some hacky stuff bellow, for some reason bilby doesn't like to se
# so I convert the priors back to uniform
# This works fine in parallel bilby so I'm not sure what the issue

# Overwrite custom priors with uniform to save the bilby object
priors_gw['lambda_tilde'] = Uniform(name='lambda_tilde',minimum=0,m
priors_gw['chirp_mass_source'] = Uniform(name='chirp_mass_source',m
```

In [ ]:
```
# Again, something that should be fixed at some point!

# Create a new serializable result object
serializable_result = bilby.core.result.Result(
    label=label,
    outdir=outdir,
    sampler='nestle',
    search_parameter_keys=['chirp_mass_source', 'mass_ratio', 'lamb
    fixed_parameter_keys=[],
    priors=priors_gw,
```

```
    posterior=result.posterior,
    log_evidence=result.log_evidence,
    log_evidence_err=result.log_evidence_err,
    log_noise_evidence=result.log_noise_evidence,
    log_bayes_factor=result.log_bayes_factor,
    log_likelihood_evaluations=result.log_likelihood_evaluations,
    log_prior_evaluations=result.log_prior_evaluations,
    sampling_time=result.sampling_time,
    meta_data=result.meta_data
)

# Save using standard bilby format
serializable_result.save_to_file()
```

In [ ]:
```
# Just in case save the posteriors to a dat too!
result.save_posterior_samples()
```