In [14]:
```
1 lc_split=lc.split_by_gti(min_points=1000)
```

```
WARNING:root:Segment with a single time bin! Ignoring this segment!
```

In [15]:
```
1 lc_split
```

Out[15]:
```
[<stingray.lightcurve.Lightcurve at 0x7f1ecbdaa290>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecbda9ab0>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecbda9e10>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecbdaa470>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecbda93c0>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecbda92a0>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecbdaa320>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecbda92d0>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecb98dd80>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecb98f190>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecb98ee90>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecb98f670>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecb98e2c0>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecb98fe50>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecb98ee30>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecb98ed10>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecad78ac0>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecad79720>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecad79a80>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecb1611b0>,
 <stingray.lightcurve.Lightcurve at 0x7f1ecb161300>]
```

In [16]:
```
1 lc0=lc_split[0]
```

In [17]:
```
1 lc0.gti
```

Out[17]:
```
array([[1.41446052e+08, 1.41447961e+08]])
```

In [18]:
```
1 lc0.gti[0][1]-lc0.gti[0][0]
```

Out[18]:
```
1908.9999999403954
```

In [19]:
```
1 len(lc0.time)
```

Out[19]:
```
38179
```

In [20]:
```
1 lc0.time[38178]-lc0.time[0]
```

Out[20]:
```
1908.9000000059605
```

In [21]:
```
1 ps0=Powerspectrum(lc0,norm='leahy')
```

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
Cell In [21], line 1
----> 1 ps0=Powerspectrum(lc0,norm='leahy')

File ~/.local/lib/python3.10/site-packages/stingray/powerspectrum.py:137, in Powerspectrum.__init__(self, data, nor
m, gti, dt, lc, skip_checks, legacy)
    134     return self._initialize_empty()
    136 if not legacy and data is not None:
--> 137     return self._initialize_from_any_input(data, dt=dt, norm=norm)
    139 Crossspectrum.__init__(self, data1=data, data2=data, norm=norm,
    140                         gti=gti, dt=dt, skip_checks=True, legacy=legacy)
    141 self.nphots = self.nphots1

File ~/.local/lib/python3.10/site-packages/stingray/powerspectrum.py:611, in Powerspectrum._initialize_from_any_inpu
t(self, data, dt, segment_size, gti, norm, silent, use_common_mean)
    601     spec = powerspectrum_from_events(
    602         data,
    603         dt,
   (...)
    608         gti=gti,
    609     )
    610 elif isinstance(data, Lightcurve):
--> 611     spec = powerspectrum_from_lightcurve(
    612         data,
    613         segment_size,
    614         norm=norm,
    615         silent=silent,
    616         use_common_mean=use_common_mean,
    617         gti=gti,
    618     )
    619     spec.lc1 = data
    620 elif isinstance(data, (tuple, list)):

File ~/.local/lib/python3.10/site-packages/stingray/powerspectrum.py:1310, in powerspectrum_from_lightcurve(lc, segm
ent_size, gti, norm, silent, use_common_mean)
   1307 if gti is None:
   1308     gti = lc.gti
-> 1310 table = avg_pds_from_events(lc.time, gti, segment_size, lc.dt, norm=norm,
   1311                             use_common_mean=use_common_mean, silent=silent,
   1312                             fluxes=lc.counts, errors=err)
   1314 return _create_powerspectrum_from_result_table(table,
   1315                                                 force_averaged=force_averaged)
```

```
File ~/.local/lib/python3.10/site-packages/stingray/fourier.py:1582, in avg_pds_from_events(times, gti, segment_size, dt, norm, use_common_mean, silent, fluxes, errors)
   1577 dt = segment_size / n_bin
   1579 flux_iterable = get_flux_iterable_from_segments(times, gti, segment_size,
   1580                                                 n_bin, fluxes=fluxes,
   1581                                                 errors=errors)
-> 1582 cross = avg_pds_from_iterable(flux_iterable, dt, norm=norm,
   1583                               use_common_mean=use_common_mean,
   1584                               silent=silent)
   1585 if cross is not None:
   1586     cross.meta["gti"] = gti

File ~/.local/lib/python3.10/site-packages/stingray/fourier.py:982, in avg_pds_from_iterable(flux_iterable, dt, norm, use_common_mean, silent)
    980 sum_of_photons = 0
    981 common_variance = None
--> 982 for flux in local_show_progress(flux_iterable):
    983     if flux is None or np.all(flux == 0):
    984         continue

File ~/.local/lib/python3.10/site-packages/stingray/fourier.py:901, in get_flux_iterable_from_segments(times, gti, segment_size, n_bin, fluxes, errors)
    897     dt = np.median(np.diff(times[:100]))
    899 fun = _which_segment_idx_fun(binned, dt)
--> 901 for s, e, idx0, idx1 in fun(times, gti, segment_size):
    902     if idx1 - idx0 < 2:
    903         yield None

File ~/.local/lib/python3.10/site-packages/stingray/gti.py:1597, in generate_indices_of_segment_boundaries_binned(times, gti, segment_size, dt)
   1595 gti = np.asarray(gti)
   1596 times = np.asarray(times)
-> 1597 startidx, stopidx = bin_intervals_from_gtis(gti, segment_size, times,
   1598                                              dt=dt)
   1600 if dt is None:
   1601     dt = 0
```

```
File ~/.local/lib/python3.10/site-packages/stingray/gti.py:1282, in bin_intervals_from_gtis(gtis, segment_size, time, dt, fraction_step, epsilon)
   1277     newbins = calculate_segment_bin_start(
   1278         startbin, stopbin, nbin, fraction_step=fraction_step)
   1279     spectrum_start_bins = \
   1280         np.append(spectrum_start_bins,
   1281                   newbins)
-> 1282 assert len(spectrum_start_bins) > 0, \
   1283     ("No GTIs are equal to or longer than segment_size.")
   1284 return spectrum_start_bins, spectrum_start_bins + nbin

AssertionError: No GTIs are equal to or longer than segment_size.
```