fluorescenceCalibration

August 20, 2018

1 Gain Calibration of Spray's fluorescence data

The objective here is to find an offset for each mission.

Concept:

Steps applied here:

```
In [16]: %matplotlib inline

         from datetime import timedelta

         import numpy as np
         import pandas as pd
         from pandas import Timedelta
         import xarray as xr
         import pymc3 as pm

         import matplotlib.pyplot as plt
         from matplotlib.ticker import NullFormatter
```

2 Loading data

The file flMatched.hdf was created by matchup.py and contains all spray missions with fluorescence and the profile matchups with MODIS-Aqua, i.e. whenever there is a coincident or near by MODIS-Aqua measurement.

```
In [8]: inputFilename = '../data/flMatched_4km.hdf'
        profile = pd.read_hdf(inputFilename, key='profile')
        data = pd.read_hdf(inputFilename, key='data')

In [9]: spray = xr.merge([profile.to_xarray(), data.to_xarray()],
                         join='inner')

        aux_coords = ['ndive', 'datetime', 'lat', 'lon', 'mission',
                      'mission_id', 'experiment', 'experiment_id']
        spray.set_coords(aux_coords, inplace=True)
        print(spray)
```

```
<xarray.Dataset>
Dimensions:        (depth: 100, profile_id: 99752)
Coordinates:
  * profile_id     (profile_id) int64 20215 20216 20217 20218 20219 20220 ...
    ndive          (profile_id) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
    datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
    lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
    lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
    mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
    mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
    experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
    experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
  * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
Data variables:
    temp           (depth, profile_id) float64 16.57 16.38 16.24 16.16 16.39 ...
    sal            (depth, profile_id) float64 33.42 33.41 33.43 33.43 33.43 ...
    fl             (depth, profile_id) float64 1.382 1.23 1.429 2.101 2.857 ...
```

2.1 Defining matchups

Here I'm getting all data in a range (dL_agg) for each profile, and recovering the number of satellite pixels (count), mean, standard deviation, standard error (sem), median, and pseudo-standard-deviation.

```
In [10]: dL_agg = 10e3
         satellites = ['seawifs', 'aqua', 'terra', 'viirs']


         matchup = []
         for satname in satellites:
             try:
                 inrange = pd.read_hdf(inputFilename, key=satname)

                 grp = inrange[inrange.dL<=dL_agg].groupby('profile_id')
                 tmp = pd.DataFrame({
                     'sat': satname,
                     'chl_count': grp.chlor_a.count(),
                     'chl_mean': grp.chlor_a.mean(),
                     'chl_median': grp.chlor_a.median(),
                     'chl_psdstd': (grp.chlor_a.quantile(.75) - grp.chlor_a.quantile(.25))/1.34
                     'chl_std': grp.chlor_a.std(),
                     'chl_sem': grp.chlor_a.sem()})
                 matchup.append(tmp.set_index('sat', append=True).to_xarray())
```

```
            except:
                print('Failed to extract data for %s' % satname)

        matchup = xr.merge(matchup, join='outer')
        print(matchup)

        del(inrange)
        del(grp)
        del(tmp)
```

```
Failed to extract data for seawifs
<xarray.Dataset>
Dimensions:     (profile_id: 67004, sat: 3)
Coordinates:
  * profile_id  (profile_id) int64 793 797 798 800 803 805 809 816 820 823 ...
  * sat         (sat) object 'aqua' 'terra' 'viirs'
Data variables:
    chl_count   (profile_id, sat) float64 12.0 11.0 13.0 12.0 11.0 13.0 13.0 ...
    chl_mean    (profile_id, sat) float32 0.7363032 0.80052286 0.42922863 ...
    chl_median  (profile_id, sat) float32 0.6272783 0.53259164 0.4206165 ...
    chl_psdstd  (profile_id, sat) float64 0.1824 0.4808 0.05015 0.1824 ...
    chl_std     (profile_id, sat) float32 0.30510858 0.40528625 0.056684013 ...
    chl_sem     (profile_id, sat) float64 0.08808 0.1222 0.01572 0.08808 ...
```

```
In [11]: print('==== Only Spray data ====')
         print(spray)
         spray = spray.merge(matchup, join='left')
         print('')
         print('==== Added satellite data ====')
         print(spray)
```

```
==== Only Spray data ====
<xarray.Dataset>
Dimensions:        (depth: 100, profile_id: 99752)
Coordinates:
  * profile_id     (profile_id) int64 20215 20216 20217 20218 20219 20220 ...
    ndive          (profile_id) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
    datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
    lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
    lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
    mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
    mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
    experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
    experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
  * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
Data variables:
    temp           (depth, profile_id) float64 16.57 16.38 16.24 16.16 16.39 ...
    sal            (depth, profile_id) float64 33.42 33.41 33.43 33.43 33.43 ...
    fl             (depth, profile_id) float64 1.382 1.23 1.429 2.101 2.857 ...
==== Added satellite data ====
<xarray.Dataset>
Dimensions:        (depth: 100, profile_id: 99752, sat: 3)
Coordinates:
  * profile_id     (profile_id) int64 20215 20216 20217 20218 20219 20220 ...
    ndive          (profile_id) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
    datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
    lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
    lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
    mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
    mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
    experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
    experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
  * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
  * sat            (sat) object 'aqua' 'terra' 'viirs'
Data variables:
    temp           (depth, profile_id) float64 16.57 16.38 16.24 16.16 16.39 ...
    sal            (depth, profile_id) float64 33.42 33.41 33.43 33.43 33.43 ...
    fl             (depth, profile_id) float64 1.382 1.23 1.429 2.101 2.857 ...
    chl_count      (profile_id, sat) float64 14.0 25.0 nan 15.0 27.0 nan ...
    chl_mean       (profile_id, sat) float32 2.1242242 0.9714916 nan ...
    chl_median     (profile_id, sat) float32 1.6580045 0.7507338 nan ...
    chl_psdstd     (profile_id, sat) float64 0.5084 0.3523 nan 0.529 0.354 ...
    chl_std        (profile_id, sat) float32 1.0751781 0.3345816 nan ...
    chl_sem        (profile_id, sat) float64 0.2874 0.06692 nan 0.2675 ...
```

3 Preparing data

3.0.1 Local night time

From longitude estimate the local time as a dt from Grenweeich, so there are no jumps between the timezones but a a continuous time offset. From that I assume daylight between 6 to 18 hrs.

One potential improvement is to consider latitude and period of the year to the estimate the day extension. But for now this should be a good approximation.

```
In [17]: local_dt = spray.lon.to_series().apply(lambda x: Timedelta(x/360., 'D')).to_xarray()
         local_time = (spray.datetime - local_dt)
         spray['night_time'] = (local_time.dt.hour < 6) | (local_time.dt.hour > 18)
         del(local_dt)
         del(local_time)
         spray.set_coords('night_time', inplace=True)
         spray
```

```
Out[17]: <xarray.Dataset>
         Dimensions:        (depth: 100, profile_id: 99752, sat: 3)
```

```
Coordinates:
  * profile_id     (profile_id) int64 20215 20216 20217 20218 20219 20220 ...
    ndive          (profile_id) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
    datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
    lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
    lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
    mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
    mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
    experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
    experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
  * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
  * sat            (sat) object 'aqua' 'terra' 'viirs'
    night_time     (profile_id) bool True True True True False False False ...
Data variables:
    temp           (depth, profile_id) float64 16.57 16.38 16.24 16.16 16.39 ...
    sal            (depth, profile_id) float64 33.42 33.41 33.43 33.43 33.43 ...
    fl             (depth, profile_id) float64 1.382 1.23 1.429 2.101 2.857 ...
    chl_count      (profile_id, sat) float64 14.0 25.0 nan 15.0 27.0 nan ...
    chl_mean       (profile_id, sat) float32 2.1242242 0.9714916 nan ...
    chl_median     (profile_id, sat) float32 1.6580045 0.7507338 nan ...
    chl_psdstd     (profile_id, sat) float64 0.5084 0.3523 nan 0.529 0.354 ...
    chl_std        (profile_id, sat) float32 1.0751781 0.3345816 nan ...
    chl_sem        (profile_id, sat) float64 0.2874 0.06692 nan 0.2675 ...
```

3.0.2 Fluorescence sum in the top layer

```
In [165]: # spray['fl_sum'] = 5 * spray.fl_unbias_exp.sel(depth=10).reset_coords(drop=True) \
          #                     + 10 * spray.fl_unbias_exp.isel(depth=spray.depth<=30).sum(dim='

          # plt.figure(figsize=(15,4))
          # plt.plot(spray.fl_sum)
          #spray['fl_sum'] = 5 * spray.fl_unbias.sel(depth=10).reset_coords(drop=True) \
          #                    + 10 * spray.fl_unbias.isel(depth=spray.depth<=20).sum(dim='depth
          spray['fl_sum'] = 5 * spray.fl.sel(depth=10).reset_coords(drop=True) \
                             + 10 * spray.fl.isel(depth=spray.depth<=20).sum(dim='depth')
          spray['fl_sum05'] = 5 * spray.fl.sel(depth=10).reset_coords(drop=True)
          spray['fl_sum30'] = 5 * spray.fl.sel(depth=10).reset_coords(drop=True) \
                             + 10 * spray.fl.isel(depth=spray.depth<=30).sum(dim='depth')

          # plt.plot(spray.fl_sum)

          spray
```

```
Out[165]: <xarray.Dataset>
          Dimensions:        (depth: 100, profile_id: 99752, sat: 3)
          Coordinates:
            * profile_id     (profile_id) int64 20215 20216 20217 20218 20219 20220 ...
              ndive          (profile_id) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
```

```
    datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
    lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
    lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
    mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
    mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
    experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
    experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
  * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
  * sat            (sat) object 'aqua' 'terra' 'viirs'
    night_time     (profile_id) bool True True True True False False False ...
Data variables:
    temp           (depth, profile_id) float64 16.57 16.38 16.24 16.16 16.39 ...
    sal            (depth, profile_id) float64 33.42 33.41 33.43 33.43 33.43 ...
    fl             (depth, profile_id) float64 1.382 1.23 1.429 2.101 2.857 ...
    chl_count      (profile_id, sat) float64 14.0 25.0 nan 15.0 27.0 nan ...
    chl_mean       (profile_id, sat) float32 2.1242242 0.9714916 nan ...
    chl_median     (profile_id, sat) float32 1.6580045 0.7507338 nan ...
    chl_psdstd     (profile_id, sat) float64 0.5084 0.3523 nan 0.529 0.354 ...
    chl_std        (profile_id, sat) float32 1.0751781 0.3345816 nan ...
    chl_sem        (profile_id, sat) float64 0.2874 0.06692 nan 0.2675 ...
    fl_sum         (profile_id) float64 39.29 41.99 45.07 53.51 63.31 56.52 ...
    fl_sum_alt1    (profile_id) float64 6.912 6.15 7.147 10.51 14.29 9.165 ...
    fl_sum_alt2    (profile_id) float64 50.63 52.99 56.84 64.87 74.65 69.2 ...
    fl_sum05       (profile_id) float64 6.912 6.15 7.147 10.51 14.29 9.165 ...
    fl_sum30       (profile_id) float64 50.63 52.99 56.84 64.87 74.65 69.2 ...
```

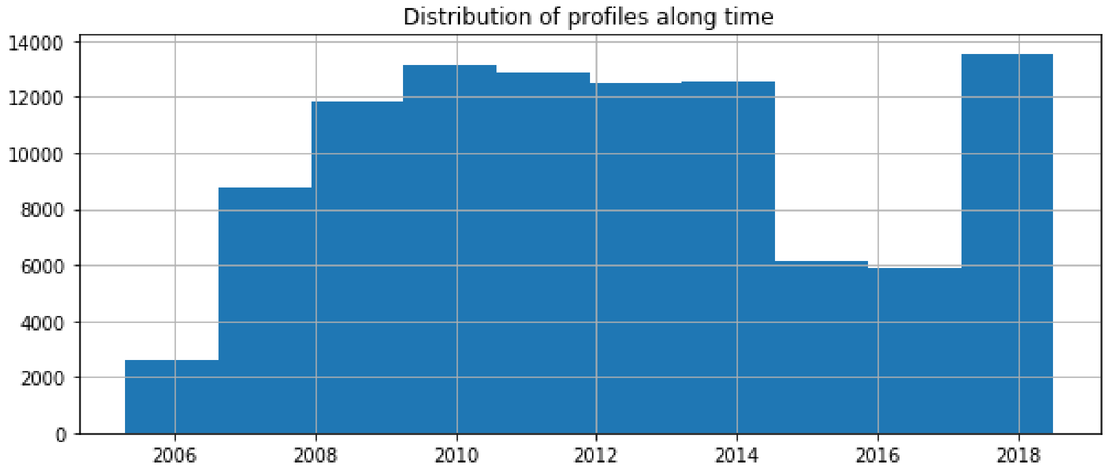
3.1 Sanity check

Let's check if the loaded data makes sense. An earlier version of the matchup script had a bug that restricted to only one line, so I want to be sure that everything looks good before starting the analysis.

```
In [51]: print("The dataset considered here covers from {0} to {1}".format(
             spray.datetime.min().values,
             spray.datetime.max().values))
```

```
The dataset considered here covers from 2005-04-21T20:29:09.000000000 to 2018-07-02T17:43:21.00
```

```
In [52]: fig = plt.figure(figsize=(10,4))
         spray.datetime.to_series().hist()
         noprint = plt.title('Distribution of profiles along time')
```

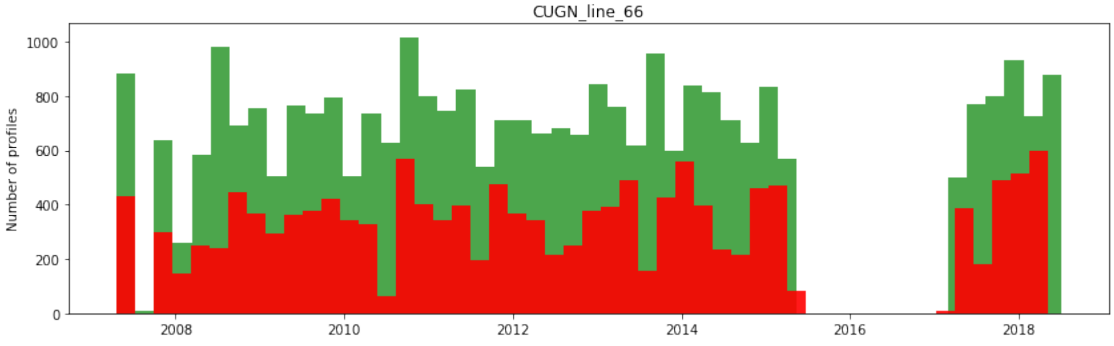


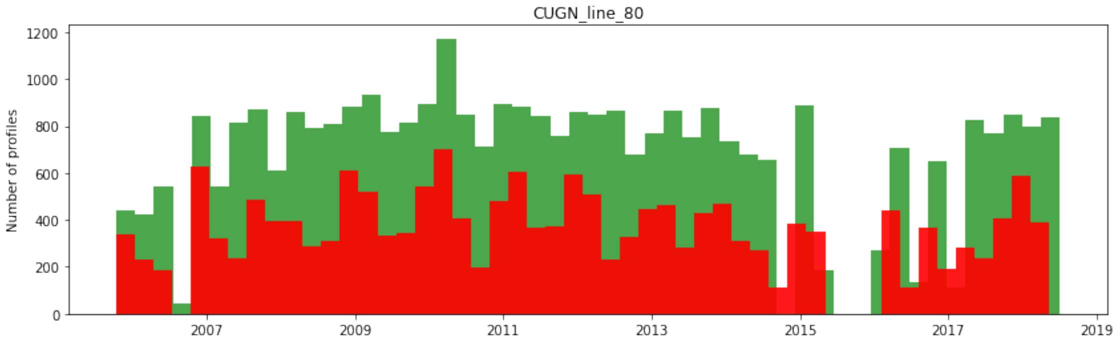
How representative are the matched up satellite measurements compared with the number of profiles? Ideally it should be just below the Spray profiles.

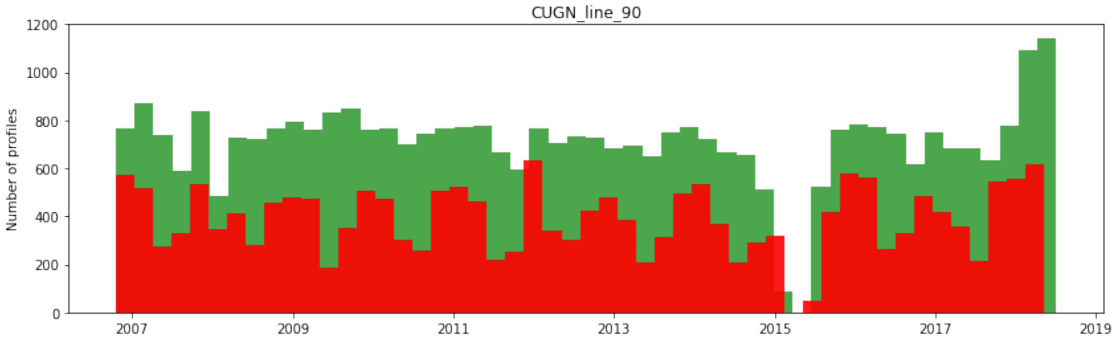
3.1.1 Profiles available and profiles with a matchup along time

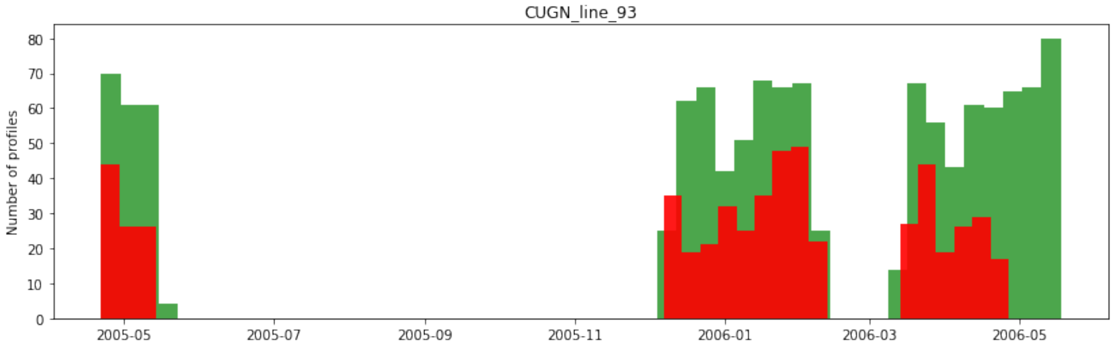
IMPROVE THIS. FIRST DOUBLE CHECK, THEN SPLIT NIGHT TIME ONLY.

```
In [67]: for experiment_name, grp in spray.sel(sat='aqua').groupby('experiment'):
             plt.figure(figsize=(14,4))
             #grp.datetime.to_series().hist()
             plt.hist(grp.datetime.to_series(), color='green', bins=50, alpha=0.7)
             tmp = grp.dropna(dim='profile_id', how='all', subset=['chl_mean'])
             if tmp.dims['profile_id'] > 0:
                 plt.hist(tmp.datetime.to_series(), bins=50, color='red', alpha=0.9)
             plt.title(experiment_name)
             plt.ylabel('Number of profiles')
```









4 Proposed linear model: Spray x Satellite Chlorophyll

```
In [81]: def fit_fl(fl, chl, chl_std=0):

             basic_model = pm.Model()

             with basic_model:
```

```
        # Priors for unknown model parameters
        f0 = pm.Normal('f0', mu=0, sd=10)
        gain = pm.Normal('gain', mu=1, sd=1)
        sigma = pm.HalfNormal('sigma', sd=5)

        # Expected value of outcome
        mu = f0 + gain * fl

        sd = sigma + chl_std

        # Likelihood (sampling distribution) of observations
        Y_obs = pm.Normal('Y_obs', mu=mu, sd=sd, observed=chl)

    with basic_model:
        trace = pm.sample(draws=1000, chains=10, tune=1000, progressbar=False)

    return trace
```

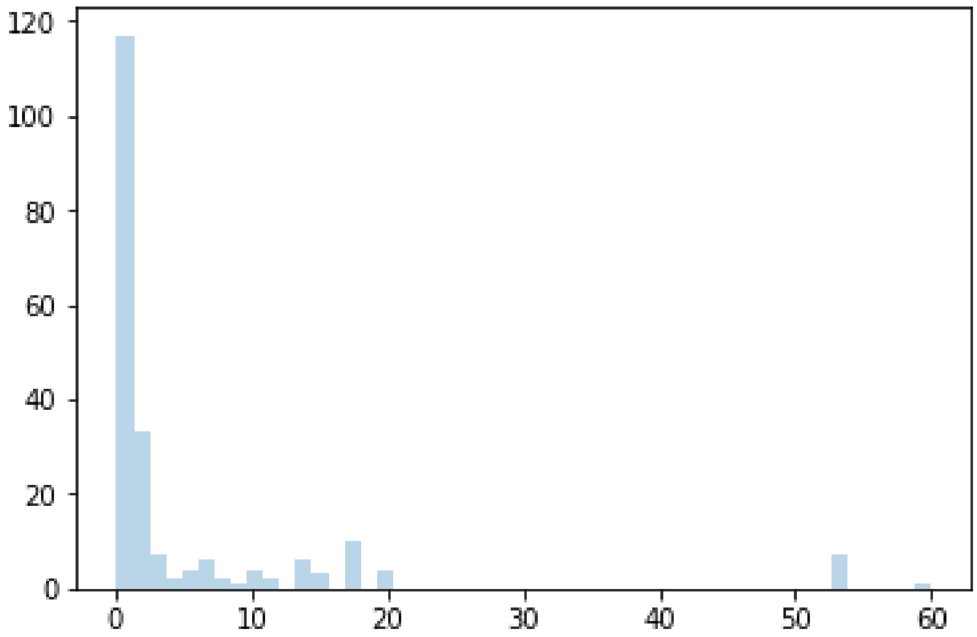
4.1 A test case on a single mission

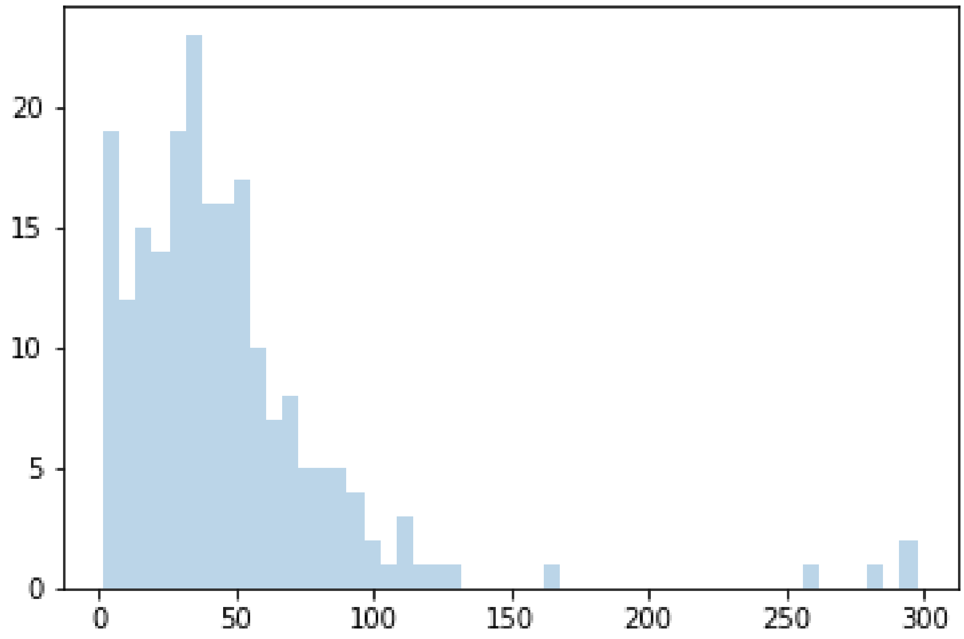
Only night time Spray measurements.

```
In [145]: mission_id = '07401401'

          single_mission = spray.sel(sat='aqua').isel(profile_id=(spray.night_time & (spray.mis

          single_mission

          X = np.array(single_mission.fl_sum)
          Y = np.array(single_mission.chl_mean)
          S = np.array(single_mission.chl_std.where(((single_mission.chl_count > 8) | (single_m
```

```
In [146]: plt.hist(single_mission.chl_mean.dropna('profile_id'), bins=50, alpha=.3)

          plt.figure()
          plt.hist(single_mission.fl_sum.dropna('profile_id'), bins=50, alpha=.3)

          #plt.figure()
          #plt.hist(np.log(spray.sel(sat='aqua').chl_mean.dropna('profile_id')), bins=50, alph
          #plt.hist(np.log(spray.sel(sat='aqua').fl_sum.dropna('profile_id')+1e-4), bins=50, a
          #plt.hist(np.log(spray.sel(sat='aqua').fl_sum_alt1.dropna('profile_id')+1e-4), bins=
          #plt.hist(np.log(spray.sel(sat='aqua').fl_sum_alt2.dropna('profile_id')+1e-4), bins=
```

```
Out[146]: (array([19., 12., 15., 14., 19., 23., 16., 16., 17., 10.,  7.,  8.,  5.,
                   5.,  5.,  4.,  2.,  1.,  3.,  1.,  1.,  1.,  0.,  0.,  0.,  0.,
                   0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
                   0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,  1.,  0.,  2.]),
           array([  2.07709091,   7.98477409,  13.89245727,  19.80014045,
                   25.70782364,  31.61550682,  37.52319   ,  43.43087318,
                   49.33855636,  55.24623955,  61.15392273,  67.06160591,
                   72.96928909,  78.87697227,  84.78465545,  90.69233864,
                   96.60002182, 102.507705  , 108.41538818, 114.32307136,
                  120.23075455, 126.13843773, 132.04612091, 137.95380409,
                  143.86148727, 149.76917045, 155.67685364, 161.58453682,
                  167.49222   , 173.39990318, 179.30758636, 185.21526955,
                  191.12295273, 197.03063591, 202.93831909, 208.84600227,
                  214.75368545, 220.66136864, 226.56905182, 232.476735  ,
                  238.38441818, 244.29210136, 250.19978455, 256.10746773,
                  262.01515091, 267.92283409, 273.83051727, 279.73820045,
                  285.64588364, 291.55356682, 297.46125   ]),
           <a list of 50 Patch objects>)
```

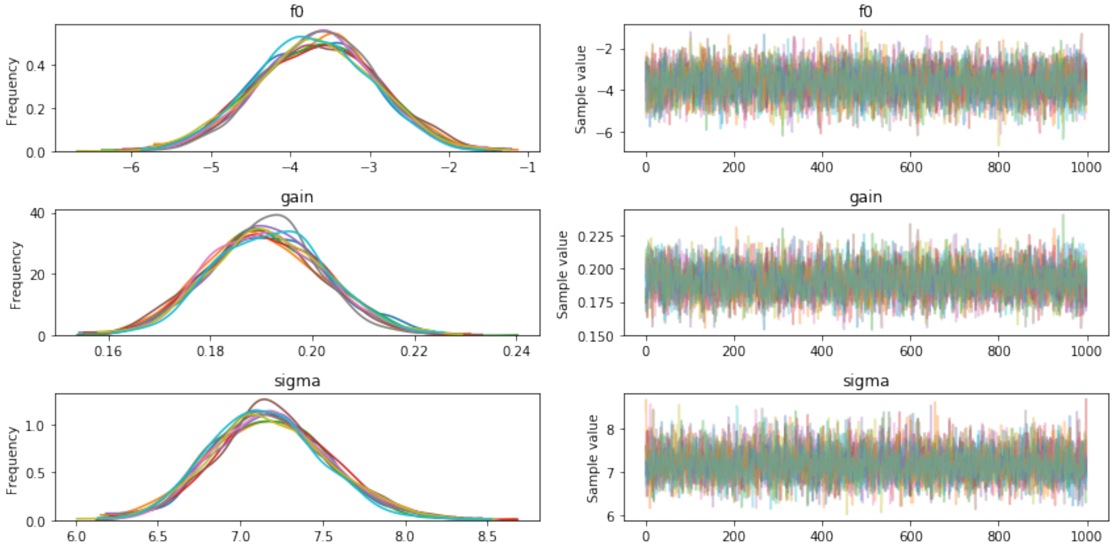




```
In [147]: trace = fit_fl(fl=X, chl=Y)
          pm.traceplot(trace)
          summary = pm.summary(trace)
          print(summary)
```

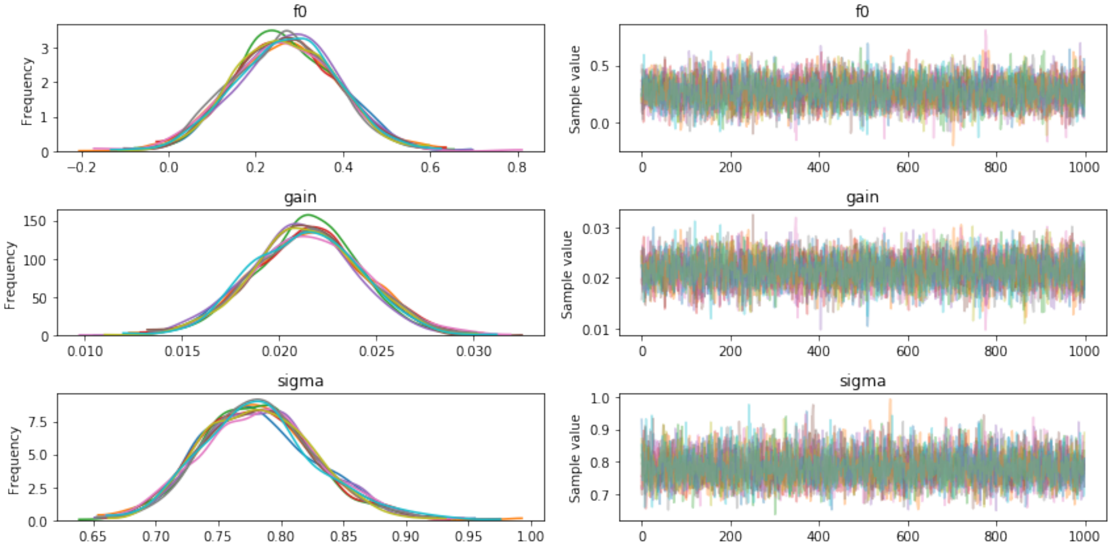
```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
```

```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0    -3.681364  0.739142  0.010383 -5.083049 -2.219673  5173.660864  1.001073
gain   0.190782  0.011353  0.000165  0.169660  0.214033  5062.734931  1.001391
sigma  7.180883  0.354964  0.004573  6.506057  7.899931  6607.790103  1.001195
```



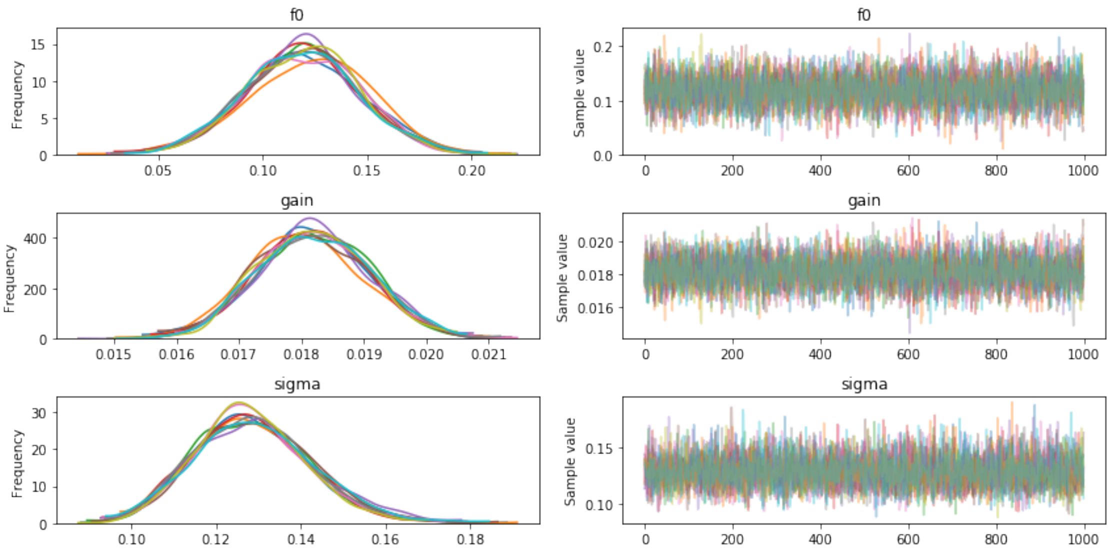
```
In [148]: idx = single_mission['chl_std'] <= 5
          trace = fit_fl(fl=X[idx], chl=Y[idx])
          pm.traceplot(trace)
          summary = pm.summary(trace)
          print(summary)
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.266936  0.117085  0.001756  0.038794  0.489081  4894.318295  1.000839
gain   0.021380  0.002822  0.000043  0.016077  0.027112  4979.272274  1.000986
sigma  0.781995  0.045582  0.000510  0.694526  0.873245  6259.671269  0.999758
```



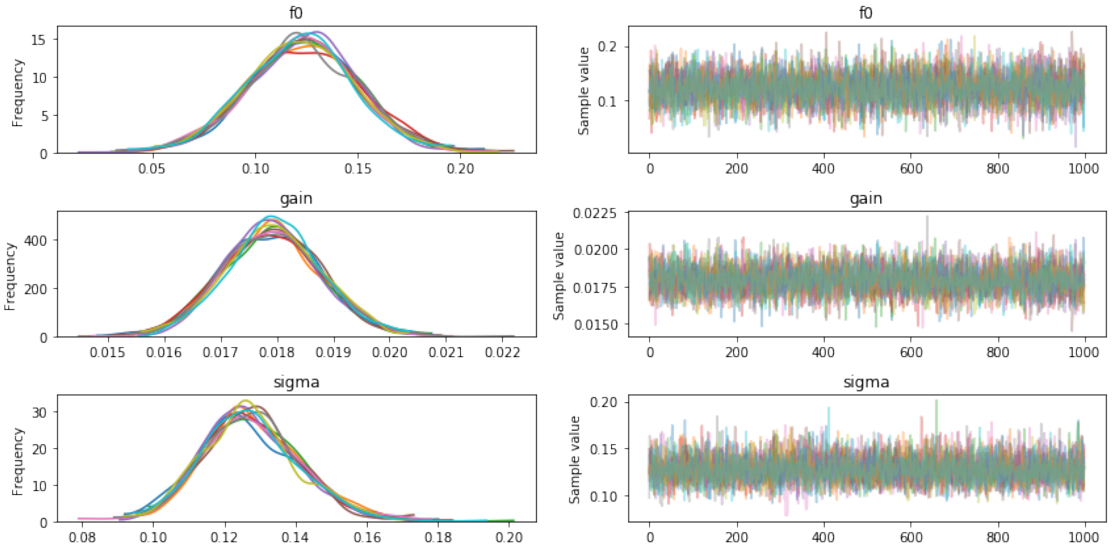
```
In [149]: trace = fit_fl(fl=X, chl=Y, chl_std=S)
          pm.traceplot(trace)
          summary = pm.summary(trace)
          print(summary)
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.119027  0.027492  0.000398  0.064437  0.172313  4973.668285  1.000812
gain   0.018170  0.000899  0.000013  0.016444  0.019936  5233.959544  1.001006
sigma  0.128478  0.013695  0.000181  0.102839  0.155651  5505.772292  1.000727
```



```
In [150]: idx = single_mission['chl_std'] <= 2
          trace = fit_fl(fl=X[idx], chl=Y[idx], chl_std=S[idx])
          pm.traceplot(trace)
          summary = pm.summary(trace)
          print(summary)
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.122799  0.026719  0.000370  0.067396  0.173142  5712.754595  1.000328
gain   0.017916  0.000877  0.000012  0.016288  0.019708  5553.173737  1.000276
sigma  0.127519  0.013597  0.000180  0.101651  0.154398  6024.571663  1.000815
```



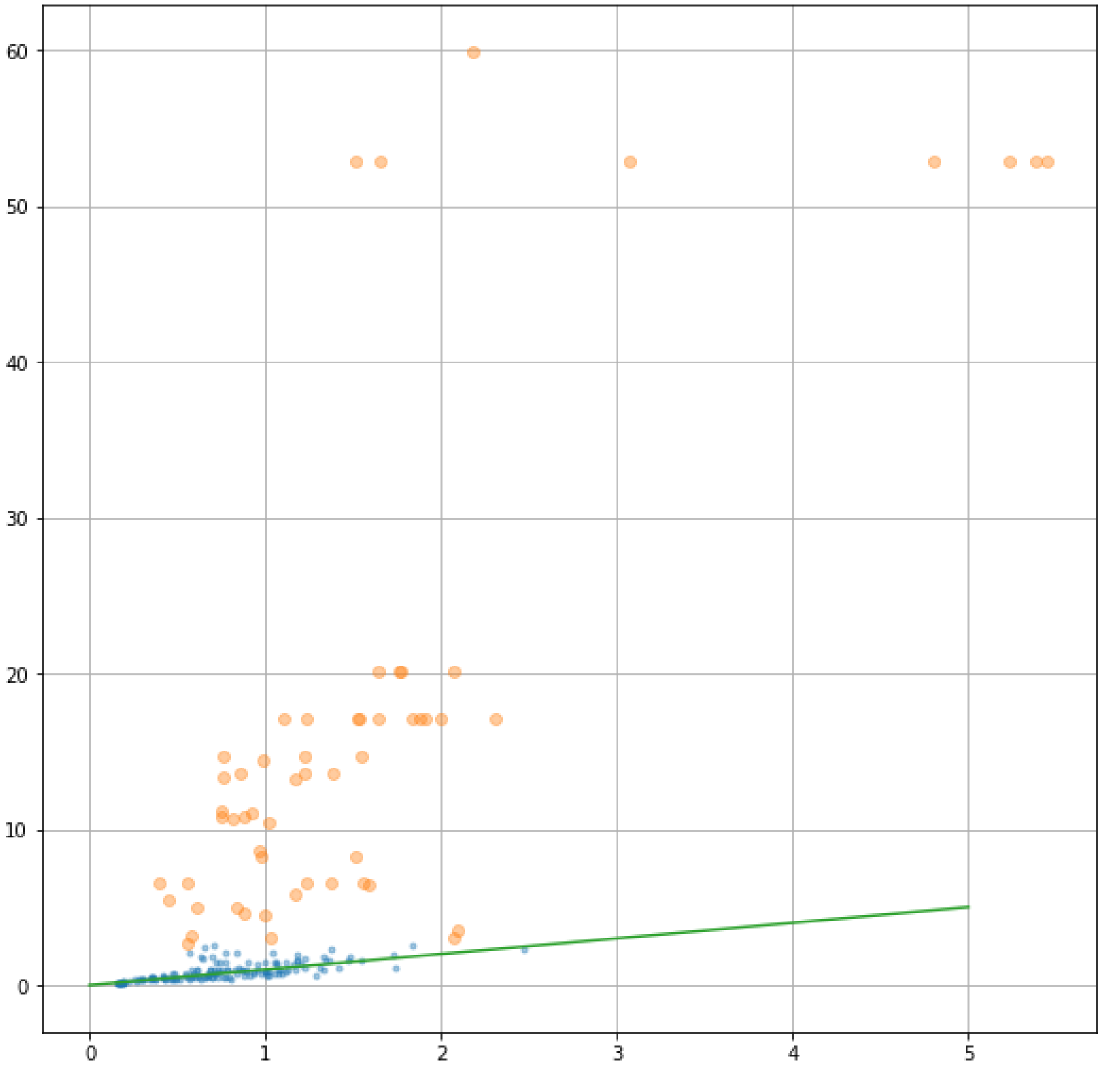
```
In [157]: print(X.shape)
          idx = single_mission['chl_std'] > 2
          idx.shape
```

```
(209,)
```

```
Out[157]: (209,)
```

```
In [160]: fig = plt.figure(figsize=(10, 10))

          idx = single_mission['chl_std'] > 2

          plt.plot(summary['mean']['f0'] + summary['mean']['gain'] * X[~idx], Y[~idx], '.', al
          # plt.plot(X*summary['mean']['beta'], np.array(subset.chl_std), '.')


          plt.plot(summary['mean']['f0'] + summary['mean']['gain'] * X[idx], Y[idx], 'o', alpha
          #plt.plot(X[idx]*summary['mean']['beta'], Y[idx], 'rx')
          # plt.plot(X[idx]*summary['mean']['beta'], np.array(subset.chl_std)[idx], 'rx')

          # plt.ylim(0, 30)
          plt.plot([0,5], [0,5])
          plt.grid()
          plt.title(mission_id)
          plt.xlabel('Corrected Spray fluorescence')
          plt.ylabel('MODIS-Aqua Chlorophyll')
```

```
#plt.xlim(0, 5)
#plt.ylim(0, 30)
```



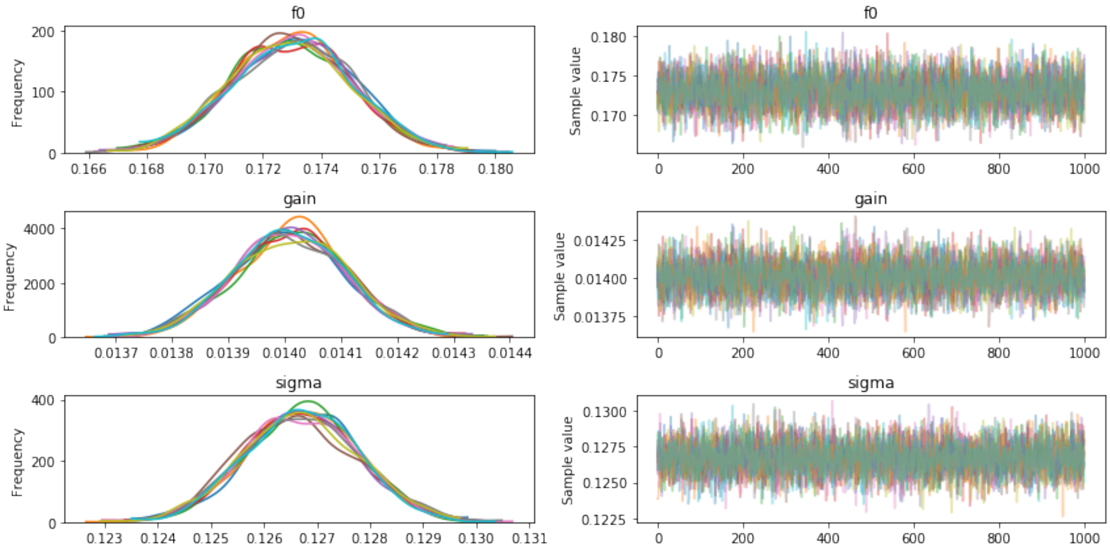
```
In [162]: alldata = spray.sel(sat='aqua').isel(profile_id=(spray.night_time)).dropna(dim='profi

          print(alldata)

          X = np.array(alldata.fl_sum)
          Y = np.array(alldata.chl_mean)
          S = np.array(alldata.chl_std.where(((alldata.chl_count > 8) | (alldata.chl_std >= 5.


          trace = fit_fl(fl=X, chl=Y, chl_std=S)
          pm.traceplot(trace)
```

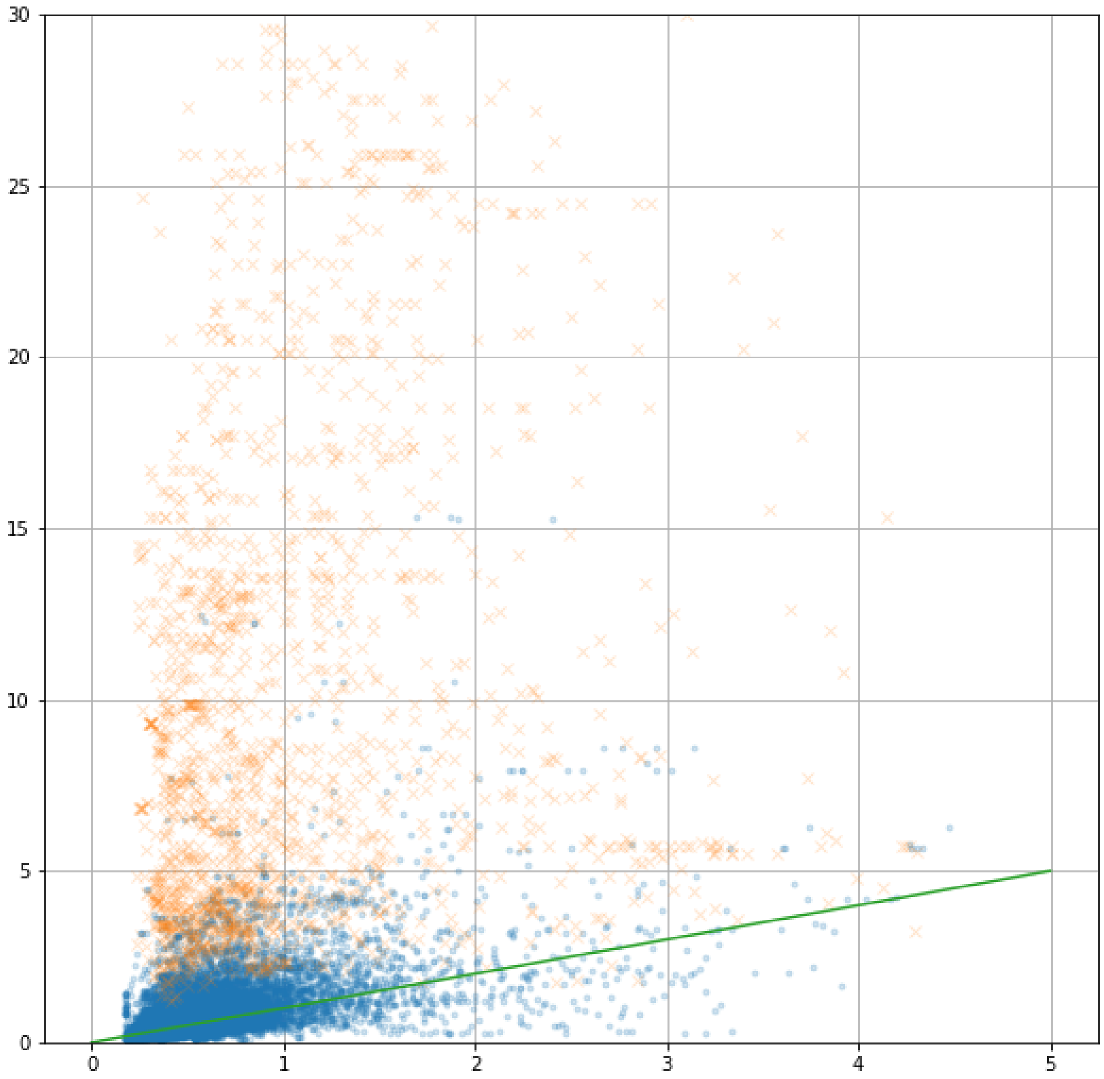
```
summary = pm.summary(trace)
print(summary)


idx = alldata['chl_std'] > 2
fig = plt.figure(figsize=(10, 10))
plt.plot(summary['mean']['f0'] + summary['mean']['gain'] * X[~idx], Y[~idx], '.', al
# plt.plot(X*summary['mean']['beta'], np.array(subset.chl_std), '.')


plt.plot(summary['mean']['f0'] + summary['mean']['gain'] * X[idx], Y[idx], 'x', alpha
#plt.plot(X[idx]*summary['mean']['beta'], Y[idx], 'rx')
# plt.plot(X[idx]*summary['mean']['beta'], np.array(subset.chl_std)[idx], 'rx')

plt.ylim(0, 30)
plt.plot([0,5], [0,5])
plt.grid()
```

```
<xarray.Dataset>
Dimensions:        (depth: 100, profile_id: 24430)
Coordinates:
  * profile_id     (profile_id) int64 20215 20216 20217 20218 20231 20233 ...
    ndive          (profile_id) int64 1 2 3 4 15 16 17 18 19 25 26 27 28 788 ...
    datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
    lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.31 34.31 ...
    lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -120.0 ...
    mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
    mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
    experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 1 3 3 3 3 3 ...
    experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
  * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
    sat            <U4 'aqua'
    night_time     (profile_id) bool True True True True True True True True ...
Data variables:
    temp           (depth, profile_id) float64 16.57 16.38 16.24 16.16 16.41 ...
    sal            (depth, profile_id) float64 33.42 33.41 33.43 33.43 33.43 ...
    fl             (depth, profile_id) float64 1.382 1.23 1.429 2.101 1.457 ...
    chl_count      (profile_id) float64 14.0 15.0 16.0 17.0 29.0 44.0 34.0 ...
    chl_mean       (profile_id) float32 2.1242242 2.1278057 1.9993627 ...
    chl_median     (profile_id) float32 1.6580045 1.7377731 1.5632914 ...
    chl_psdstd     (profile_id) float64 0.5084 0.529 0.5902 0.6447 1.011 ...
    chl_std        (profile_id) float32 1.0751781 1.0361605 1.0999595 ...
    chl_sem        (profile_id) float64 0.2874 0.2675 0.275 0.2615 0.1298 ...
    fl_sum         (profile_id) float64 39.29 41.99 45.07 53.51 37.5 36.63 ...
    fl_sum_alt1    (profile_id) float64 6.912 6.15 7.147 10.51 7.285 7.251 ...
    fl_sum_alt2    (profile_id) float64 50.63 52.99 56.84 64.87 46.44 47.68 ...


Auto-assigning NUTS sampler...
```

```
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
```

```
           mean        sd  mc_error  hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.172962  0.002066  0.000025  0.168974  0.176990  5827.970570  1.000230
gain   0.014009  0.000100  0.000001  0.013809  0.014201  5869.735387  1.000340
sigma  0.126695  0.001068  0.000012  0.124640  0.128793  6860.047323  1.000012
```





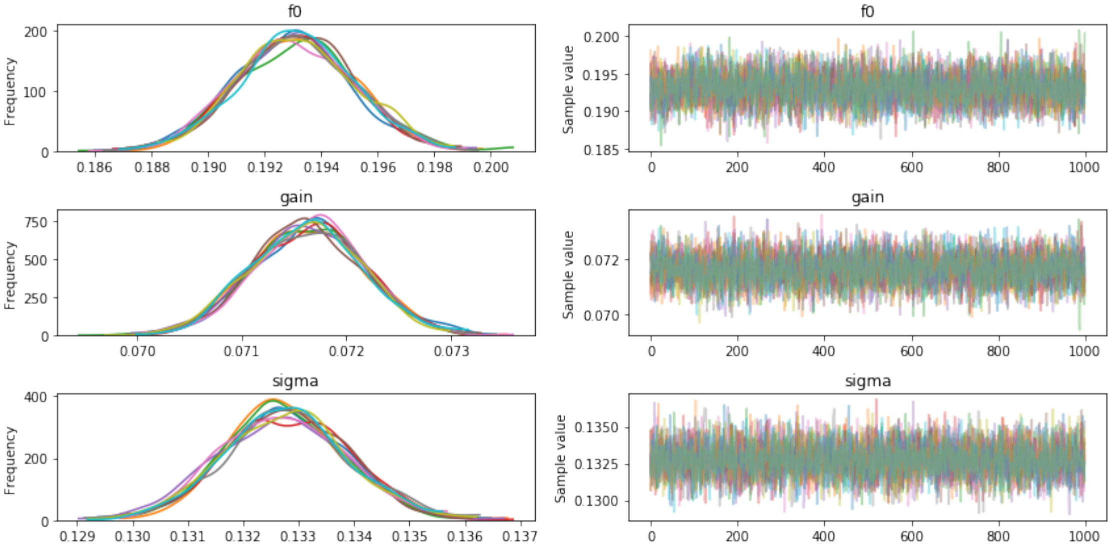
```
In [163]: alldata = spray.sel(sat='aqua').isel(profile_id=(spray.night_time)).dropna(dim='prof

          print(alldata)

          X = np.array(alldata.fl_sum05)
          Y = np.array(alldata.chl_mean)
          S = np.array(alldata.chl_std.where(((alldata.chl_count > 8) | (alldata.chl_std >= 5.


          trace = fit_fl(fl=X, chl=Y, chl_std=S)
          pm.traceplot(trace)
          summary = pm.summary(trace)
          print(summary)
```

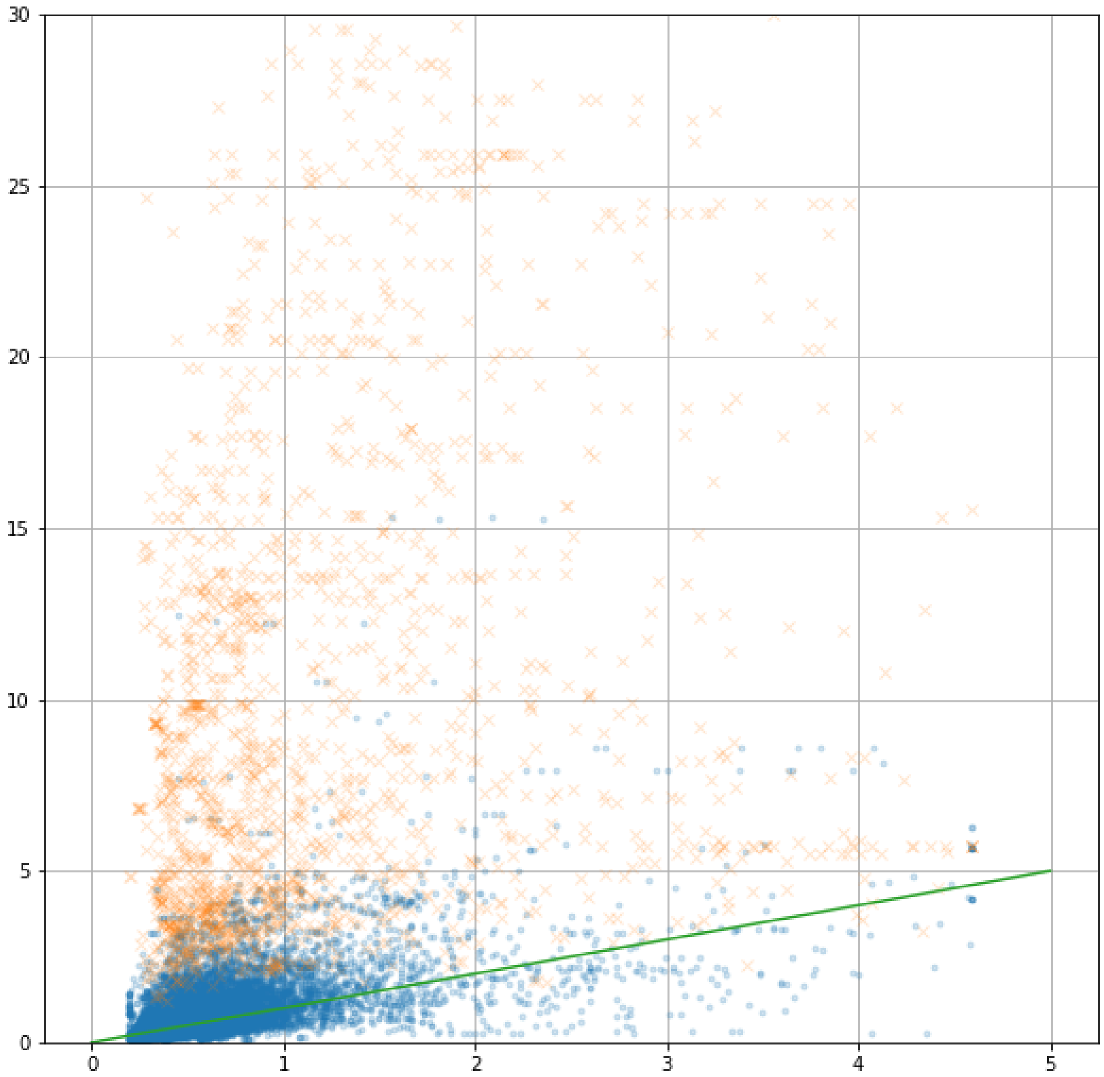
```
idx = alldata['chl_std'] > 2
fig = plt.figure(figsize=(10, 10))
plt.plot(summary['mean']['f0'] + summary['mean']['gain'] * X[~idx], Y[~idx], '.', al
# plt.plot(X*summary['mean']['beta'], np.array(subset.chl_std), '.')


plt.plot(summary['mean']['f0'] + summary['mean']['gain'] * X[idx], Y[idx], 'x', alpha
#plt.plot(X[idx]*summary['mean']['beta'], Y[idx], 'rx')
# plt.plot(X[idx]*summary['mean']['beta'], np.array(subset.chl_std)[idx], 'rx')

plt.ylim(0, 30)
plt.plot([0,5], [0,5])
plt.grid()
```

```
<xarray.Dataset>
Dimensions:        (depth: 100, profile_id: 24430)
Coordinates:
  * profile_id     (profile_id) int64 20215 20216 20217 20218 20231 20233 ...
    ndive          (profile_id) int64 1 2 3 4 15 16 17 18 19 25 26 27 28 788 ...
    datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
    lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.31 34.31 ...
    lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -120.0 ...
    mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
    mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
    experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 1 3 3 3 3 3 ...
    experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
  * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
    sat            <U4 'aqua'
    night_time     (profile_id) bool True True True True True True True True ...
Data variables:
    temp           (depth, profile_id) float64 16.57 16.38 16.24 16.16 16.41 ...
    sal            (depth, profile_id) float64 33.42 33.41 33.43 33.43 33.43 ...
    fl             (depth, profile_id) float64 1.382 1.23 1.429 2.101 1.457 ...
    chl_count      (profile_id) float64 14.0 15.0 16.0 17.0 29.0 44.0 34.0 ...
    chl_mean       (profile_id) float32 2.1242242 2.1278057 1.9993627 ...
    chl_median     (profile_id) float32 1.6580045 1.7377731 1.5632914 ...
    chl_psdstd     (profile_id) float64 0.5084 0.529 0.5902 0.6447 1.011 ...
    chl_std        (profile_id) float32 1.0751781 1.0361605 1.0999595 ...
    chl_sem        (profile_id) float64 0.2874 0.2675 0.275 0.2615 0.1298 ...
    fl_sum         (profile_id) float64 39.29 41.99 45.07 53.51 37.5 36.63 ...
    fl_sum_alt1    (profile_id) float64 6.912 6.15 7.147 10.51 7.285 7.251 ...
    fl_sum_alt2    (profile_id) float64 50.63 52.99 56.84 64.87 46.44 47.68 ...


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
```

```
NUTS: [sigma, gain, f0]
```

```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.193065  0.002047  0.000025  0.189123  0.197127  6325.655170  1.000641
gain   0.071627  0.000533  0.000007  0.070575  0.072646  6441.914366  1.000997
sigma  0.132748  0.001107  0.000013  0.130624  0.134917  7148.279276  1.000421
```





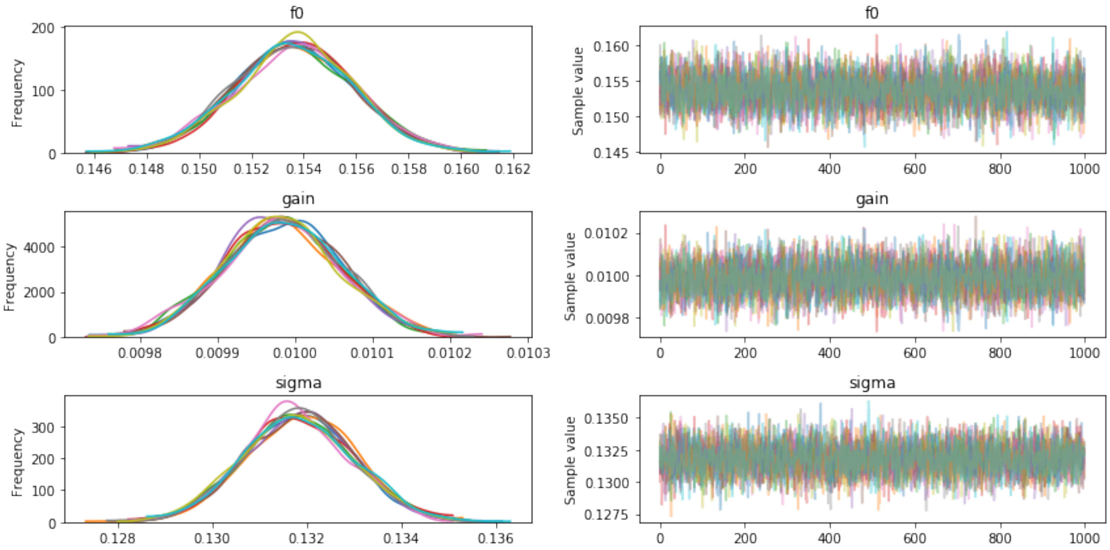
```
In [164]: alldata = spray.sel(sat='aqua').isel(profile_id=(spray.night_time)).dropna(dim='profi

          print(alldata)

          X = np.array(alldata.fl_sum30)
          Y = np.array(alldata.chl_mean)
          S = np.array(alldata.chl_std.where(((alldata.chl_count > 8) | (alldata.chl_std >= 5.(


          trace = fit_fl(fl=X, chl=Y, chl_std=S)
          pm.traceplot(trace)
          summary = pm.summary(trace)
          print(summary)
```

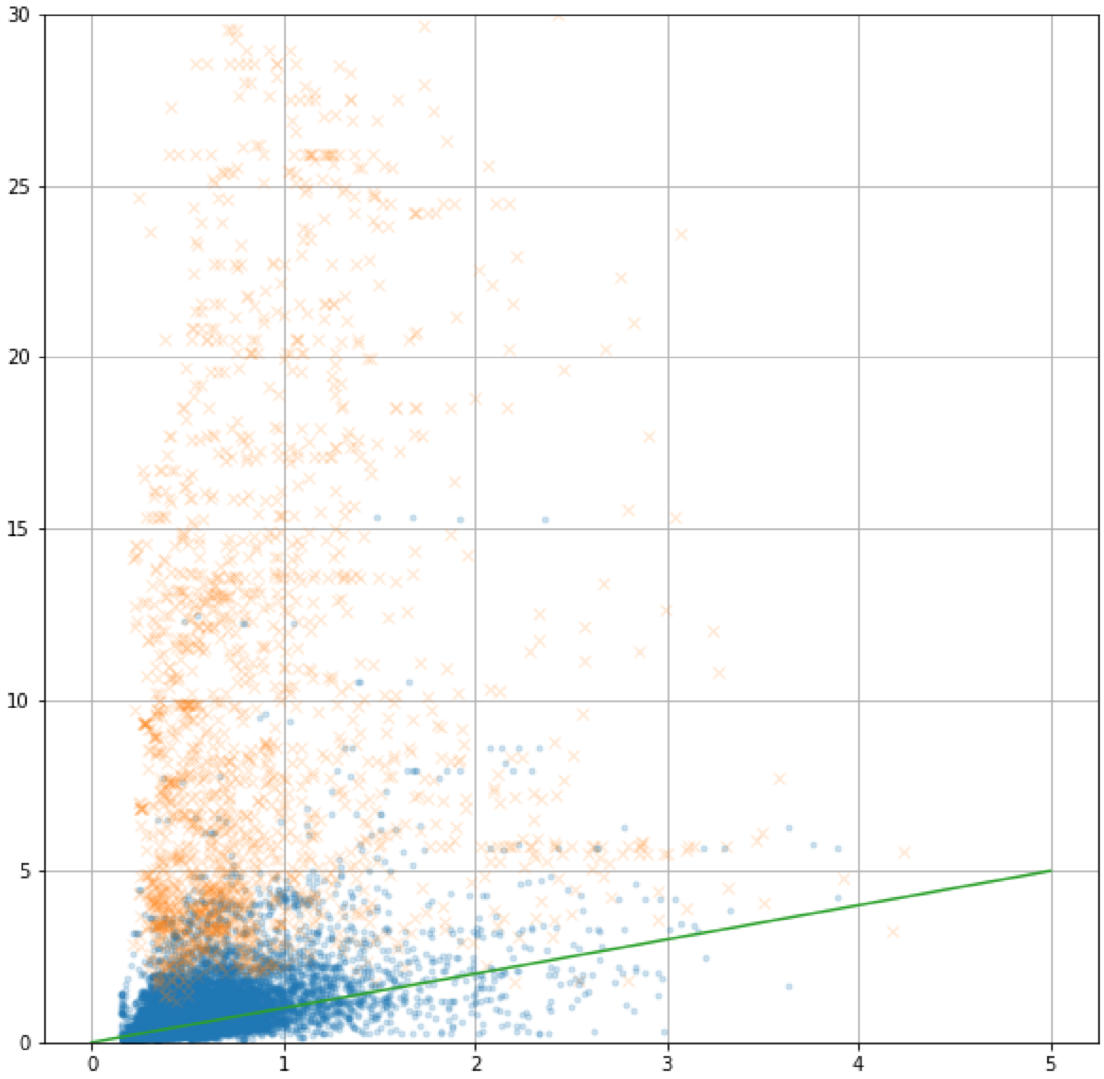
```
idx = alldata['chl_std'] > 2
fig = plt.figure(figsize=(10, 10))
plt.plot(summary['mean']['f0'] + summary['mean']['gain'] * X[~idx], Y[~idx], '.', alp
# plt.plot(X*summary['mean']['beta'], np.array(subset.chl_std), '.')


plt.plot(summary['mean']['f0'] + summary['mean']['gain'] * X[idx], Y[idx], 'x', alpha
#plt.plot(X[idx]*summary['mean']['beta'], Y[idx], 'rx')
# plt.plot(X[idx]*summary['mean']['beta'], np.array(subset.chl_std)[idx], 'rx')

plt.ylim(0, 30)
plt.plot([0,5], [0,5])
plt.grid()
```

```
<xarray.Dataset>
Dimensions:         (depth: 100, profile_id: 24430)
Coordinates:
  * profile_id      (profile_id) int64 20215 20216 20217 20218 20231 20233 ...
    ndive           (profile_id) int64 1 2 3 4 15 16 17 18 19 25 26 27 28 788 ...
    datetime        (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
    lat             (profile_id) float64 34.35 34.35 34.35 34.34 34.31 34.31 ...
    lon             (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -120.0 ...
    mission_id      (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
    mission         (profile_id) object '06A00501' '06A00501' '06A00501' ...
    experiment_id   (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 1 3 3 3 3 3 ...
    experiment      (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
  * depth           (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
    sat             <U4 'aqua'
    night_time      (profile_id) bool True True True True True True True True ...
Data variables:
    temp            (depth, profile_id) float64 16.57 16.38 16.24 16.16 16.41 ...
    sal             (depth, profile_id) float64 33.42 33.41 33.43 33.43 33.43 ...
    fl              (depth, profile_id) float64 1.382 1.23 1.429 2.101 1.457 ...
    chl_count       (profile_id) float64 14.0 15.0 16.0 17.0 29.0 44.0 34.0 ...
    chl_mean        (profile_id) float32 2.1242242 2.1278057 1.9993627 ...
    chl_median      (profile_id) float32 1.6580045 1.7377731 1.5632914 ...
    chl_psdstd      (profile_id) float64 0.5084 0.529 0.5902 0.6447 1.011 ...
    chl_std         (profile_id) float32 1.0751781 1.0361605 1.0999595 ...
    chl_sem         (profile_id) float64 0.2874 0.2675 0.275 0.2615 0.1298 ...
    fl_sum          (profile_id) float64 39.29 41.99 45.07 53.51 37.5 36.63 ...
    fl_sum_alt1     (profile_id) float64 6.912 6.15 7.147 10.51 7.285 7.251 ...
    fl_sum_alt2     (profile_id) float64 50.63 52.99 56.84 64.87 46.44 47.68 ...


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
```

```
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.9252329565387309, but should be
The acceptance probability does not match the target. It is 0.9118242543306284, but should be
```

```
            mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0      0.153634  0.002260  0.000029  0.149123  0.157995  5789.816613  0.999935
gain    0.009983  0.000076  0.000001  0.009831  0.010126  5812.509970  1.000208
sigma   0.131814  0.001153  0.000014  0.129545  0.134055  6798.137394  1.000367
```





```
In [177]: mission.isel(profile_id=mission.night_time)

Out[177]: <xarray.Dataset>
          Dimensions:         (depth: 100, profile_id: 91)
          Coordinates:
            * profile_id      (profile_id) int64 2417 2418 2446 2449 2453 2455 2461 ...
              ndive           (profile_id) int64 1 2 11 12 13 14 15 20 21 22 23 28 29 ...
              datetime        (profile_id) datetime64[ns] 2005-04-21T20:29:09 ...
              lat             (profile_id) float64 32.7 32.71 32.66 32.66 32.66 32.66 ...
              lon             (profile_id) float64 -117.4 -117.4 -117.5 -117.5 -117.5 ...
              mission_id      (profile_id) int64 206 206 206 206 206 206 206 206 206 ...
              mission         (profile_id) object '05400601' '05400601' '05400601' ...
              experiment_id   (profile_id) int64 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ...
              experiment      (profile_id) object 'CUGN_line_93' 'CUGN_line_93' ...
```

```
      * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
        sat            <U4 'aqua'
        night_time     (profile_id) bool True True True True True True True True ...
    Data variables:
        temp           (depth, profile_id) float64 14.28 13.38 15.98 16.11 16.06 ...
        sal            (depth, profile_id) float64 33.22 33.23 33.15 33.15 33.15 ...
        fl             (depth, profile_id) float64 0.5574 0.5857 0.3023 0.35 ...
        chl_count      (profile_id) float64 18.0 18.0 17.0 17.0 nan nan nan nan ...
        chl_mean       (profile_id) float32 0.7469786 0.7469786 0.41865477 ...
        chl_median     (profile_id) float32 0.59614235 0.59614235 0.3535363 ...
        chl_psdstd     (profile_id) float64 0.4494 0.4494 0.1204 0.1169 nan nan ...
        chl_std        (profile_id) float32 0.37071112 0.37071112 0.20041542 ...
        chl_sem        (profile_id) float64 0.08738 0.08738 0.04861 0.03323 nan ...
        fl_sum         (profile_id) float64 24.41 28.78 24.87 27.94 27.25 34.12 ...
        fl_sum_alt1    (profile_id) float64 2.787 2.929 1.511 1.75 1.506 1.164 ...
        fl_sum_alt2    (profile_id) float64 48.51 38.63 45.02 52.83 60.08 57.74 ...
        fl_sum05       (profile_id) float64 2.787 2.929 1.511 1.75 1.506 1.164 ...
        fl_sum30       (profile_id) float64 48.51 38.63 45.02 52.83 60.08 57.74 ...
```

```
In [199]: nColumns = 4
          fl_to_fit = 'fl_min'

          # for experiment_name, experiment in spray.sel(sat='aqua').isel(profile_id=spray.exp
          for experiment_name, experiment in spray.sel(sat='aqua').groupby('experiment'):
              print(experiment_name)
              nFigs = len(np.unique(experiment.mission))
              nRows = int(np.ceil(float(nFigs)/nColumns))
              fig, axes = plt.subplots(nrows=nRows, ncols=nColumns, figsize=(18, int(nRows * 3
              i = -1
              for mission_name, mission in experiment.groupby('mission'):
                  print(mission_name)
                  mission = mission.isel(profile_id=mission.night_time).dropna(dim='profile_id

                  i += 1
                  if mission.profile_id.size > 10:

                      X = np.array(mission.fl_sum)
                      Y = np.array(mission.chl_mean)
                      S = np.array(mission.chl_std.where(((mission.chl_count > 8) | (mission.cl

                      trace = fit_fl(fl=X, chl=Y, chl_std=S)
                      #pm.traceplot(trace)
                      summary = pm.summary(trace)
                      print(summary)


                      if nRows > 1:
                          nr = int(i/nColumns)
```

```
            nc = i%nColumns
            ax = axes[nr, nc]
        else:
            ax = axes[i]

        idx = mission['chl_std'] > 2


        ax.plot(summary['mean']['f0'] + summary['mean']['gain'] * X[~idx], Y[~id
        # plt.plot(X*summary['mean']['beta'], np.array(subset.chl_std), '.')

        ax.plot(summary['mean']['f0'] + summary['mean']['gain'] * X[idx], Y[idx]
        #plt.plot(X[idx]*summary['mean']['beta'], Y[idx], 'rx')
        # plt.plot(X[idx]*summary['mean']['beta'], np.array(subset.chl_std)[idx]

        #if mission_name in fl_exp_fit:
        #    ax.plot(x, y)
        #    fl_exp_fit[mission_name] = summary
        #else:
        #    ax.plot(x, y, 'r')
        #ax.set_xlim(0, 120)
        ax.set_title("%s: %.2f + %.2fx +/- %.2f " % (mission_name, summary['mean
        ax.xaxis.set_visible(False)

        ax.plot([0,5], [0,5])
        ax.grid()
        #ax.xlabel('Corrected Spray fluorescence')
        #ax.ylabel('MODIS-Aqua Chlorophyll')
```

```
CUGN_line_66
0025


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
There were 25 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.6693155348457409, but should be c
There were 11 divergences after tuning. Increase `target_accept` or reparameterize.
There were 18 divergences after tuning. Increase `target_accept` or reparameterize.
There were 29 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.6510380121989193, but should be c
There were 15 divergences after tuning. Increase `target_accept` or reparameterize.
There were 19 divergences after tuning. Increase `target_accept` or reparameterize.
There were 6 divergences after tuning. Increase `target_accept` or reparameterize.
There were 35 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.7115169421834273, but should be
There were 7 divergences after tuning. Increase `target_accept` or reparameterize.
There were 1 divergences after tuning. Increase `target_accept` or reparameterize.
The number of effective samples is smaller than 25% for some parameters.


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.215576  0.133563  0.002519 -0.054498  0.500492  2592.948924  1.001057
gain   0.029385  0.007811  0.000148  0.013695  0.045608  2657.792466  1.001058
sigma  0.070949  0.051513  0.001124  0.002129  0.170086  2296.447302  1.000914
07401401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.119165  0.027575  0.000393  0.063359  0.172087  5109.302777  1.000505
gain   0.018175  0.000902  0.000012  0.016447  0.019979  5090.947360  1.000441
sigma  0.128605  0.013544  0.000164  0.103356  0.155296  6142.605737  0.999974
07A01401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8939421560745208, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.212042  0.034993  0.000575  0.143874  0.281548  3964.025765  1.000159
gain   0.011697  0.001363  0.000022  0.009102  0.014434  3959.940066  1.000135
sigma  0.105829  0.010009  0.000132  0.086873  0.125926  5574.671966  1.000128
08401401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.124483  0.027044  0.000346  0.071304  0.176644  5626.697853  1.000333
```

```
gain   0.014232  0.001336  0.000017  0.011650  0.016905  5539.430553  1.000221
sigma  0.099434  0.012410  0.000170  0.076929  0.124827  6061.139568  1.000239
08703001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.138386  0.013626  0.000176  0.111884  0.165587  5723.148560  1.000271
gain   0.013677  0.000802  0.000010  0.012129  0.015244  5730.627220  0.999887
sigma  0.077584  0.007932  0.000092  0.062041  0.092729  7372.031711  0.999930
08B01401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8887917867230932, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.266423  0.030746  0.000485  0.207572  0.329237  4382.307221  1.000470
gain   0.007419  0.001091  0.000017  0.005298  0.009580  4278.289959  1.000442
sigma  0.089581  0.007163  0.000087  0.076142  0.104401  4976.890799  0.999807
09301401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.146946  0.029293  0.000430  0.086759  0.201759  4541.053502  1.000835
gain   0.014959  0.001067  0.000015  0.012883  0.017076  4600.293996  1.000551
sigma  0.129102  0.012517  0.000157  0.104318  0.153065  6210.548299  1.000719
09701101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
```

```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.123899  0.009183  0.000122  0.105775  0.141579  5729.903477  0.999780
gain   0.015158  0.000965  0.000013  0.013195  0.016969  5520.247734  1.000063
sigma  0.047482  0.004528  0.000049  0.039149  0.056711  7147.232856  0.999885
09B01401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.188642  0.032914  0.000425  0.124945  0.254640  5006.838149  1.000160
gain   0.015794  0.001243  0.000015  0.013460  0.018343  5052.825765  1.000373
sigma  0.148286  0.013804  0.000180  0.122233  0.175872  6466.029129  1.000683
10301401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.256015  0.038304  0.000532  0.177604  0.329837  5116.066892  1.000123
gain   0.016052  0.001656  0.000023  0.012767  0.019298  4942.937549  0.999820
sigma  0.145734  0.018190  0.000219  0.111458  0.180964  6149.029509  0.999988
10704101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8852233257034143, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.258449  0.045633  0.000572  0.173280  0.350657  5673.059910  1.000197
gain   0.023282  0.001668  0.000023  0.020012  0.026522  5659.061659  1.000810
sigma  0.288895  0.024507  0.000304  0.245415  0.340432  6547.891758  0.999853
10A01401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
```

```
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8788199581647412, but should be c

           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.065496  0.043189  0.000689 -0.019173  0.151918  4734.382895  1.001040
gain   0.026631  0.001750  0.000027  0.023327  0.030202  4837.442012  1.001199
sigma  0.155771  0.018025  0.000221  0.122654  0.193082  6507.217995  0.999892
11104101

Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.891919404192607, but should be cl

           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.227847  0.028820  0.000445  0.170604  0.282498  3719.557590  1.000620
gain   0.013186  0.001296  0.000019  0.010729  0.015772  3816.056179  1.000296
sigma  0.093611  0.009975  0.000132  0.074303  0.112897  5887.749744  1.000137
11501401

Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]

           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.288775  0.030965  0.000393  0.224881  0.346708  5542.098751  1.000707
gain   0.006556  0.000832  0.000010  0.004988  0.008242  5980.798575  1.000340
sigma  0.148578  0.019008  0.000222  0.114866  0.188388  7294.438296  1.000133
11904101

Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]

           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.130659  0.028805  0.000348  0.072273  0.186186  5677.691709  1.001144
gain   0.016839  0.001046  0.000012  0.014770  0.018888  5670.901433  1.001127
sigma  0.160807  0.012528  0.000171  0.136595  0.184930  5859.352502  1.000380
12101401
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.230682  0.015048  0.000244  0.202220  0.261397  4365.984774  1.001334
gain   0.009355  0.001206  0.000019  0.007034  0.011692  4390.929564  1.001008
sigma  0.057633  0.005658  0.000069  0.046997  0.068765  6322.673819  0.999822
12404101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.101833  0.013226  0.000169  0.076362  0.127833  5560.495531  1.000123
gain   0.012863  0.000629  0.000007  0.011623  0.014087  5924.807466  1.000173
sigma  0.066923  0.008616  0.000100  0.051258  0.084657  6713.533119  1.000446
12801401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.137407  0.015477  0.000183  0.108515  0.168822  5368.530429  1.000004
gain   0.015170  0.000892  0.000011  0.013327  0.016826  5524.043578  0.999811
sigma  0.081240  0.007907  0.000093  0.066219  0.096790  6686.737513  0.999997
12B04101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.049919  0.042670  0.000587 -0.034691  0.130588  4354.397153  0.999897
gain   0.019165  0.001610  0.000022  0.016054  0.022336  4287.700884  0.999865
sigma  0.131878  0.013345  0.000168  0.106557  0.158055  6062.152917  1.000751
13301401
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.126187  0.075942  0.001335 -0.024646  0.271400  2724.196651  1.001063
gain   0.019924  0.004700  0.000085  0.011198  0.029430  2611.680855  1.000921
sigma  0.120513  0.020807  0.000395  0.081620  0.161633  2936.363176  1.000746
13704101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.108931  0.009638  0.000152  0.090662  0.128358  5083.908915  1.000235
gain   0.011931  0.000626  0.000009  0.010761  0.013234  5294.910025  1.000123
sigma  0.047041  0.006487  0.000082  0.034619  0.059566  7013.805102  1.000378
13A06001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
There were 3 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.7145157242841822, but should be c
There were 3 divergences after tuning. Increase `target_accept` or reparameterize.
There were 1 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.8887399334146885, but should be c
There were 1 divergences after tuning. Increase `target_accept` or reparameterize.
There were 2 divergences after tuning. Increase `target_accept` or reparameterize.


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.289852  0.239430  0.004197 -0.171358  0.783470  2749.639621  1.001785
gain   0.017106  0.013041  0.000219 -0.007800  0.044101  2943.356972  1.001628
sigma  0.270410  0.076789  0.001260  0.140298  0.427276  3096.779941  0.999956
13B04101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
```

```
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.449632  0.050657  0.000712  0.352636  0.550180  3475.754619  1.000668
gain   0.010736  0.001690  0.000024  0.007317  0.013949  3462.978343  1.000762
sigma  0.186995  0.013848  0.000188  0.160154  0.214235  5930.667148  1.001023
14305701


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.196005  0.019674  0.000270  0.158176  0.236083  5271.447693  1.000263
gain   0.017078  0.001097  0.000016  0.014871  0.019189  5145.579596  1.000548
sigma  0.071923  0.009724  0.000115  0.053780  0.091236  5806.273846  0.999751
14602801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.191267  0.012595  0.000169  0.166719  0.215954  5073.017333  1.000094
gain   0.006909  0.000955  0.000013  0.005024  0.008724  5145.074377  1.000193
sigma  0.040739  0.005309  0.000077  0.030787  0.051310  6109.072267  0.999796
14901101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8839236883527753, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.195425  0.020089  0.000253  0.156110  0.234091  5453.871174  0.999696
gain   0.016695  0.001480  0.000019  0.013938  0.019711  5380.403094  0.999772
sigma  0.126603  0.008604  0.000102  0.110558  0.143785  6740.879857  1.000540
15103001
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8856532948783133, but should be c
The acceptance probability does not match the target. It is 0.8826207550386135, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0    -0.142541  0.037020  0.000606 -0.214491 -0.069625  3811.995526  1.000241
gain   0.026391  0.001985  0.000032  0.022652  0.030411  3818.160370  1.000092
sigma  0.067581  0.006620  0.000084  0.055470  0.080938  5244.320022  0.999869
17302501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.100186  0.030119  0.000434  0.041753  0.159520  5171.295195  1.000372
gain   0.028018  0.001988  0.000029  0.024103  0.031847  5133.575046  1.000468
sigma  0.133331  0.016680  0.000191  0.101220  0.165893  6288.655753  1.000060
17702801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
There were 1 divergences after tuning. Increase `target_accept` or reparameterize.
There were 7 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.8958617585884779, but should be c
There were 9 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.7208085583928288, but should be c
There were 4 divergences after tuning. Increase `target_accept` or reparameterize.
There were 1 divergences after tuning. Increase `target_accept` or reparameterize.
There were 10 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.708413463412721, but should be cl
There were 1 divergences after tuning. Increase `target_accept` or reparameterize.
The number of effective samples is smaller than 25% for some parameters.


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.094780  0.059567  0.001248 -0.026156  0.200202  1972.338043  1.000619
gain   0.014295  0.004157  0.000094  0.006794  0.022502  1758.059976  1.000521
sigma  0.104902  0.032426  0.000787  0.050201  0.171960  1792.318382  1.000845
```

```
17A02501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0    -0.011461  0.018313  0.000288 -0.049768  0.022064  3828.426058  1.000832
gain   0.038374  0.002323  0.000037  0.033782  0.042870  3800.428338  1.001145
sigma  0.067567  0.007252  0.000106  0.053960  0.081816  4940.789638  1.000237
18104001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8863883149119776, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0    -0.054469  0.038623  0.000559 -0.129766  0.019905  3579.820181  1.000240
gain   0.023299  0.001626  0.000024  0.020278  0.026573  3520.496278  1.000391
sigma  0.152043  0.011323  0.000175  0.130977  0.175454  6010.786299  1.000632
CUGN_line_80
0030


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     2.482304  2.146146  0.038571 -1.610225  6.841118  3185.366607  1.002516
gain   0.041884  0.041814  0.000766 -0.038613  0.124895  3137.159767  1.002501
sigma  0.579948  0.648152  0.008328  0.000017  1.875376  5294.749027  1.000446
05A00501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
There were 3 divergences after tuning. Increase `target_accept` or reparameterize.
There were 5 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.7126582294900876, but should be
The number of effective samples is smaller than 25% for some parameters.


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0    -0.010862  0.066776  0.001123 -0.151635  0.107446  2864.295529  1.000952
gain   0.024967  0.003192  0.000062  0.018878  0.031046  2406.499670  1.001760
sigma  0.226562  0.075404  0.001384  0.092648  0.378180  2731.850980  1.000981
06200501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8845265131161003, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.015329  0.029486  0.000463 -0.044627  0.070527  4286.801954  0.999847
gain   0.014486  0.001196  0.000019  0.012131  0.016763  4081.046977  0.999753
sigma  0.095782  0.017528  0.000265  0.063562  0.129729  5157.488752  1.000441
06500901


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
There were 1 divergences after tuning. Increase `target_accept` or reparameterize.


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.197500  0.043780  0.000735  0.111118  0.282957  4348.757162  1.001117
gain   0.012852  0.001229  0.000019  0.010441  0.015244  4705.565661  1.000371
sigma  0.065349  0.023642  0.000332  0.023429  0.110995  4972.295247  0.999953
06A00501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.111483  0.017263  0.000206  0.075054  0.143705  5699.082044  1.000654
gain   0.019947  0.001071  0.000014  0.017858  0.022036  5568.737813  1.000537
```

```
sigma  0.108673  0.010592  0.000113  0.088977  0.129961  6847.008952  1.000109
07101101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
There were 1 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.8813613462949464, but should be c
There were 2 divergences after tuning. Increase `target_accept` or reparameterize.


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.524789  0.079301  0.001287  0.376846  0.684486  3489.072012  1.000798
gain   0.008999  0.002608  0.000044  0.003988  0.013988  3328.718829  1.001096
sigma  0.069161  0.022251  0.000411  0.029402  0.114592  4258.184998  1.001087
07301101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8806719981897614, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.173197  0.020334  0.000263  0.134690  0.215264  5317.563935  1.000247
gain   0.010245  0.000958  0.000013  0.008418  0.012186  5105.385938  1.000036
sigma  0.086422  0.009320  0.000108  0.068129  0.104116  6265.129089  1.000018
07501201


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
There were 6 divergences after tuning. Increase `target_accept` or reparameterize.
There were 3 divergences after tuning. Increase `target_accept` or reparameterize.
The number of effective samples is smaller than 25% for some parameters.


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0    -1.410150  0.227655  0.004205 -1.879916 -0.982253  2458.547204  1.001074
gain   0.062799  0.008359  0.000155  0.046892  0.079909  2446.309543  1.001122
sigma  0.123573  0.018827  0.000323  0.089135  0.161124  3489.627031  1.001640
07801301
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.083950  0.008696  0.000123  0.065957  0.100273  5934.463686  1.001636
gain   0.016243  0.000620  0.000008  0.014998  0.017441  6044.640305  1.001557
sigma  0.053059  0.005824  0.000073  0.042069  0.064480  7224.995390  1.000479
07B01201


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8800162722781977, but should be c
The acceptance probability does not match the target. It is 0.8797597950172035, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.147656  0.033148  0.000476  0.079631  0.208343  4898.522048  1.001217
gain   0.014457  0.001532  0.000021  0.011510  0.017435  4973.522155  1.001155
sigma  0.069818  0.014462  0.000203  0.045412  0.100431  5427.825168  1.000608
08101101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.103700  0.038389  0.000575  0.030621  0.180197  4395.084750  1.002014
gain   0.016051  0.001277  0.000019  0.013479  0.018426  4224.708714  1.001532
sigma  0.165531  0.015319  0.000220  0.135335  0.194794  5725.623024  1.000350
08402801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8836202204384991, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.095309  0.032906  0.000460  0.032980  0.160350  4572.249015  1.000218
```

```
gain   0.022404  0.001520  0.000021  0.019439  0.025363  4777.825496  1.000438
sigma  0.139918  0.020647  0.000275  0.102847  0.181862  5369.988901  1.000490
08701301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.161686  0.026772  0.000315  0.108976  0.214486  5901.333113  1.000603
gain   0.012845  0.000840  0.000010  0.011148  0.014436  5818.926403  1.000175
sigma  0.151912  0.012528  0.000165  0.128992  0.178112  6422.031301  1.000259
08B01101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8837184264398505, but should be c
The acceptance probability does not match the target. It is 0.8931986591825957, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.630594  0.051919  0.000860  0.531260  0.735251  3558.978714  1.001226
gain   0.000154  0.001672  0.000028 -0.003087  0.003470  3597.617541  1.001170
sigma  0.139491  0.010278  0.000146  0.119663  0.159414  5410.557229  1.000439
09201301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.185962  0.040582  0.000524  0.100836  0.259457  5579.425749  0.999803
gain   0.013506  0.001286  0.000018  0.011047  0.016085  5766.353475  0.999855
sigma  0.190266  0.018681  0.000214  0.153441  0.225400  6124.107144  1.000414
09502801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
```

```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0    -0.204142  0.082350  0.001068 -0.360780 -0.036637  5774.672318  0.999847
gain   0.030983  0.002452  0.000037  0.026307  0.036045  5565.682746  1.000570
sigma  0.359509  0.046807  0.000650  0.269644  0.452451  6024.092021  0.999900
09801301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.088210  0.019303  0.000259  0.051995  0.127318  5796.946982  1.000167
gain   0.020170  0.000967  0.000012  0.018267  0.022091  6016.010582  1.000148
sigma  0.125887  0.014442  0.000192  0.098908  0.154890  6323.463614  1.000623
09C02801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.269143  0.029878  0.000461  0.212033  0.327407  3630.203921  1.000491
gain   0.013900  0.000987  0.000015  0.012051  0.015862  3647.301138  1.000440
sigma  0.097969  0.008855  0.000109  0.081721  0.116083  5163.412646  1.000085
10202501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.237488  0.043478  0.000620  0.155213  0.325845  5357.023885  1.000389
gain   0.036542  0.004174  0.000063  0.028335  0.044661  4806.090310  1.000241
sigma  0.154855  0.028869  0.000367  0.101272  0.211631  5663.485024  1.000032
10402501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
```

```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.271303  0.022354  0.000263  0.227413  0.314126  6736.243751  1.000021
gain   0.019130  0.001022  0.000012  0.017171  0.021187  6833.457940  1.000264
sigma  0.088700  0.015438  0.000173  0.060266  0.119646  7639.026437  1.000425
10403001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8845411681821888, but should be c
The acceptance probability does not match the target. It is 0.8936594148575565, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.646789  0.056654  0.000962  0.530391  0.753399  3657.740354  1.000399
gain   0.006991  0.003209  0.000052  0.000885  0.013539  3871.610455  1.000539
sigma  0.036423  0.024993  0.000331  0.000007  0.083763  5331.278672  1.000314
10603001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.261023  0.034475  0.000571  0.193731  0.329077  4978.436436  1.000810
gain   0.018962  0.001793  0.000029  0.015517  0.022556  5031.170560  1.000557
sigma  0.124757  0.013058  0.000162  0.099255  0.150365  6340.660979  1.000160
10B01301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8817201517606095, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.108905  0.052579  0.000811  0.002668  0.209852  4241.294818  1.000886
gain   0.023668  0.001673  0.000026  0.020436  0.027052  4175.335360  1.001237
sigma  0.194907  0.020971  0.000286  0.155628  0.238055  5401.212419  1.000522
11202801
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8831698208703422, but should be c
The acceptance probability does not match the target. It is 0.8831822724651702, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.364697  0.023172  0.000363  0.318836  0.410350  3332.470728  1.000178
gain   0.005884  0.000909  0.000014  0.004194  0.007788  3263.678188  1.000272
sigma  0.045578  0.007183  0.000097  0.031985  0.060259  4517.991842  1.000344
11501301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.129904  0.030549  0.000388  0.070214  0.190714  5621.123153  1.000027
gain   0.017073  0.001304  0.000016  0.014639  0.019745  6012.714828  0.999889
sigma  0.110991  0.023830  0.000275  0.068132  0.158566  7137.755358  0.999985
11602501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.760569  0.105081  0.001344  0.555024  0.969749  4410.283331  1.000940
gain   0.006551  0.002006  0.000026  0.002468  0.010345  4578.515870  1.000618
sigma  0.354537  0.049380  0.000546  0.258295  0.450429  6761.671572  0.999964
11802801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8830765125157288, but should be c
The acceptance probability does not match the target. It is 0.8792439953014065, but should be c
The acceptance probability does not match the target. It is 0.878616998514293, but should be cl
```

```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     2.308025  0.313487  0.005828  1.663271  2.893309  2848.615969  1.002840
gain  -0.009485  0.005418  0.000099 -0.020043  0.001100  2902.865042  1.002376
sigma  0.067483  0.074649  0.000846  0.000003  0.211925  5658.204642  1.000087
11901101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.092179  0.016917  0.000212  0.059064  0.125888  5381.873828  1.000073
gain   0.021751  0.001039  0.000012  0.019755  0.023884  5438.554411  0.999717
sigma  0.079555  0.006963  0.000075  0.066616  0.093671  6908.301313  1.000306
11C02501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.221755  0.032831  0.000519  0.160131  0.288981  3681.937983  1.000750
gain   0.018367  0.001826  0.000029  0.014820  0.022028  3628.548094  1.000837
sigma  0.114885  0.010731  0.000136  0.095057  0.136149  5866.292669  1.001002
12301101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.125082  0.016931  0.000241  0.091927  0.157763  5198.068108  1.000959
gain   0.014572  0.001249  0.000019  0.012130  0.017006  5339.634298  1.000663
sigma  0.069513  0.008658  0.000112  0.053173  0.086698  6490.937705  1.000242
12602501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
```

```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     1.031984  0.050002  0.000576  0.940411  1.135027  8085.541760  1.000374
gain   0.011930  0.015331  0.000165 -0.019529  0.040586  8105.008164  0.999892
sigma  0.130198  0.036281  0.000402  0.063827  0.199546  7908.926491  1.000443
12801101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.267009  0.013198  0.000190  0.241925  0.293629  6260.948846  1.000507
gain   0.008691  0.000550  0.000008  0.007630  0.009786  6235.549987  1.000106
sigma  0.087430  0.008989  0.000099  0.070241  0.104981  7376.422065  1.000154
12C02801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.190084  0.042858  0.000754  0.107240  0.273755  3767.159669  1.000836
gain   0.017649  0.001850  0.000031  0.013989  0.021215  3810.781049  1.000777
sigma  0.141973  0.013116  0.000172  0.116892  0.167272  5820.478377  0.999934
13301301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.245669  0.017223  0.000230  0.212260  0.279474  6405.286149  1.001559
gain   0.008309  0.000725  0.000009  0.006905  0.009727  6534.866067  1.001028
sigma  0.098041  0.010619  0.000126  0.077098  0.118591  6931.533833  1.000014
13605701


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
```

```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.159499  0.039949  0.000580  0.080161  0.237773  5010.169452  1.001122
gain   0.011393  0.000975  0.000014  0.009596  0.013401  5162.040344  1.001022
sigma  0.186197  0.023741  0.000355  0.142317  0.234976  5526.195344  1.000454
13A01101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.175856  0.021679  0.000296  0.134943  0.218609  4780.581807  0.999975
gain   0.019315  0.001097  0.000015  0.017232  0.021477  4813.421209  1.000133
sigma  0.089123  0.007977  0.000094  0.074340  0.105422  5882.981140  1.000114
14103001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.881759214329275, but should be cl


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.284436  0.032034  0.000529  0.222798  0.346540  4724.851200  1.002017
gain   0.014563  0.001379  0.000023  0.011860  0.017268  4620.421233  1.001928
sigma  0.069236  0.012536  0.000152  0.046200  0.094363  6070.488804  1.000156
14501101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.227027  0.017987  0.000251  0.192064  0.262344  5313.896250  1.000929
gain   0.006809  0.001157  0.000017  0.004552  0.009115  5332.137627  1.001041
sigma  0.067672  0.007332  0.000093  0.053588  0.082075  6309.315976  1.000714
14803001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
```

```
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.9007588592982169, but should be
The acceptance probability does not match the target. It is 0.8945535913144191, but should be
The number of effective samples is smaller than 25% for some parameters.


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.287771  0.069325  0.001521  0.155970  0.425459  2061.546777  1.000352
gain   0.004976  0.001700  0.000037  0.001565  0.008144  2092.511194  1.000244
sigma  0.012133  0.011106  0.000216  0.000006  0.033833  2826.871603  1.002839
14C02801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.147968  0.005620  0.000076  0.136887  0.158751  5550.217501  1.000928
gain   0.015000  0.000588  0.000008  0.013862  0.016151  5780.547526  1.000960
sigma  0.034352  0.003023  0.000034  0.028486  0.040190  7211.990889  1.000006
16203001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8812228772472982, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0    -0.415136  0.029543  0.000516 -0.469408 -0.353482  3730.220658  1.001473
gain   0.019766  0.000838  0.000014  0.018098  0.021346  3709.177346  1.001693
sigma  0.058414  0.006519  0.000102  0.045979  0.071156  4672.555910  1.002529
16903001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8804726067790494, but should be
The acceptance probability does not match the target. It is 0.8894050424137693, but should be
```

```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.043118  0.010558  0.000136  0.023063  0.064671  5650.425169  1.001291
gain   0.021834  0.000750  0.000010  0.020413  0.023300  5362.777754  1.000840
sigma  0.062476  0.006365  0.000083  0.050006  0.074839  7061.304827  0.999674
17305501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8853049876006214, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.073561  0.004961  0.000064  0.063855  0.083303  5552.685406  1.000548
gain   0.015775  0.000543  0.000006  0.014725  0.016859  5848.079546  1.001050
sigma  0.025621  0.003888  0.000042  0.018421  0.033353  7293.602789  1.000246
17605801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.049178  0.028749  0.000404 -0.008476  0.104262  5239.151862  0.999692
gain   0.031430  0.002050  0.000029  0.027502  0.035535  5152.059883  0.999775
sigma  0.105887  0.015195  0.000180  0.077113  0.135518  5958.263627  1.000306
17A03001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.099050  0.015575  0.000199  0.069241  0.130363  5161.230442  1.000030
gain   0.025466  0.001015  0.000014  0.023415  0.027372  5407.561217  1.000198
sigma  0.062037  0.007436  0.000084  0.047975  0.076907  6573.389628  0.999997
18105801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
```

```
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0    -0.014582  0.029191  0.000419 -0.072118  0.043057  4995.755039  1.000492
gain   0.035753  0.001856  0.000026  0.032192  0.039402  4801.880447  1.000597
sigma  0.136124  0.015037  0.000181  0.107869  0.165382  5801.382818  1.000342
CUGN_line_90
0014


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.030904  0.009871  0.000118  0.012233  0.050898  4849.927203  1.000615
gain   0.018525  0.000912  0.000012  0.016802  0.020385  4684.906699  1.000802
sigma  0.032964  0.004294  0.000054  0.025003  0.041608  5704.015876  1.000281
0041


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8802663133019983, but should be
The acceptance probability does not match the target. It is 0.8898885987133979, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.004997  0.018451  0.000308 -0.031354  0.041683  3936.682313  1.001943
gain   0.022260  0.001951  0.000033  0.018483  0.026086  3930.235955  1.001694
sigma  0.032786  0.003992  0.000054  0.025592  0.040909  4785.184159  1.000007
0055
0064
06A01301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.144721  0.010037  0.000118  0.125540  0.164485  5997.136364  1.000962
```

```
gain   0.014249  0.000554  0.000007  0.013177  0.015324  5735.362484  1.000617
sigma  0.077742  0.006330  0.000068  0.065290  0.089890  7239.902694  0.999977
07101201


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8789588678659735, but should be


           mean        sd  mc_error  hpd_2.5  hpd_97.5        n_eff      Rhat
f0    -0.184600  0.092879  0.001547 -0.37163 -0.007930  3746.949653  1.000380
gain   0.022617  0.003318  0.000055  0.01629  0.029243  3739.123021  1.000389
sigma  0.111089  0.010313  0.000154  0.09166  0.131551  4784.748846  1.001371
07401301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8949543517068572, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.278294  0.052429  0.000697  0.176793  0.384150  5447.887834  1.000155
gain   0.008067  0.001354  0.000018  0.005479  0.010744  5468.631027  1.000536
sigma  0.180433  0.027273  0.000320  0.128813  0.234431  6823.161912  1.000280
07701101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.184411  0.014416  0.000190  0.156891  0.213536  5589.581686  1.000436
gain   0.009584  0.000567  0.000008  0.008466  0.010690  5691.291545  1.000629
sigma  0.093778  0.007429  0.000087  0.079900  0.108859  7058.846489  1.000860
07A02801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
```

```
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8791125296540917, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.285519  0.027204  0.000428  0.233508  0.339211  3728.657170  1.000700
gain   0.017468  0.001094  0.000018  0.015367  0.019635  3885.591394  1.000860
sigma  0.089686  0.007113  0.000091  0.076126  0.103860  6137.578067  1.000181
08301301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.893324690077895, but should be cl


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.341167  0.028596  0.000392  0.283697  0.395971  4505.924179  1.000281
gain   0.006135  0.000751  0.000011  0.004567  0.007524  4529.848133  1.000073
sigma  0.121196  0.010008  0.000141  0.102622  0.140923  5683.199078  1.000437
08601101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8861569777576834, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.160750  0.009185  0.000127  0.143570  0.179944  4652.982047  1.000045
gain   0.007408  0.000434  0.000006  0.006579  0.008289  4719.444424  0.999899
sigma  0.019387  0.004388  0.000054  0.011165  0.028060  5373.334667  1.000272
08902501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.125297  0.012975  0.000192  0.099692  0.150305  4740.910436  0.999841
gain   0.031946  0.001756  0.000025  0.028542  0.035445  4661.971047  0.999937
sigma  0.057502  0.005261  0.000066  0.047704  0.067823  6315.360949  1.000354
09102801
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.196270  0.016081  0.000246  0.165055  0.228681  4719.309963  1.001464
gain   0.014888  0.000998  0.000015  0.012884  0.016835  4851.330158  1.000860
sigma  0.080844  0.006084  0.000077  0.069356  0.093047  6431.821127  1.000031
09403001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.880977132052143, but should be cl


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.165207  0.017828  0.000224  0.129221  0.199180  5858.227471  1.000056
gain   0.014769  0.001222  0.000016  0.012420  0.017230  5694.677102  1.000402
sigma  0.075353  0.008275  0.000102  0.060034  0.092359  6354.088932  0.999958
09702501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.166884  0.008582  0.000124  0.149947  0.183588  4881.438562  1.000719
gain   0.010031  0.001362  0.000019  0.007344  0.012647  5005.434072  1.000158
sigma  0.043362  0.003707  0.000039  0.036733  0.051062  6804.126764  0.999915
09B03001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.285682  0.015264  0.000211  0.254490  0.315080  5043.643790  1.000630
gain   0.015875  0.001241  0.000016  0.013445  0.018333  4912.904085  1.000984
```

```
sigma  0.064688  0.005021  0.000052  0.055152  0.074782  6572.810258  1.000398
10204101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8883407762113044, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.134857  0.014142  0.000195  0.106142  0.161687  5591.796035  1.000453
gain   0.016679  0.001303  0.000019  0.014100  0.019162  5366.164997  1.000734
sigma  0.065128  0.005803  0.000064  0.054198  0.076776  6781.115730  1.000345
10602801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.124843  0.025229  0.000387  0.072800  0.171925  4879.896436  1.000693
gain   0.016081  0.001142  0.000018  0.013896  0.018333  4606.203861  1.000897
sigma  0.108095  0.010289  0.000134  0.088337  0.128618  6108.020700  0.999842
10902501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.136631  0.015306  0.000204  0.107219  0.167550  4946.105711  1.000498
gain   0.020179  0.000749  0.000010  0.018656  0.021599  5315.576219  1.000983
sigma  0.090019  0.010072  0.000111  0.071152  0.110211  6953.265902  0.999904
11101101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
```

```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.170009  0.014706  0.000199  0.139709  0.197809  5846.553030  1.000091
gain   0.017303  0.000671  0.000008  0.016010  0.018629  5632.422223  1.000034
sigma  0.102701  0.008610  0.000112  0.085926  0.119582  6610.848915  1.000246
11402501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8974878445369283, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.262295  0.047357  0.000755  0.167331  0.352584  4256.511204  1.000525
gain   0.025688  0.002473  0.000039  0.020692  0.030418  4265.390269  1.000892
sigma  0.132634  0.021442  0.000336  0.093634  0.175745  5603.025141  1.000443
11603001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.215742  0.025339  0.000364  0.167598  0.267247  4188.642792  1.000510
gain   0.008852  0.000909  0.000013  0.007023  0.010600  4153.096571  1.000362
sigma  0.061939  0.007486  0.000103  0.047534  0.076705  5516.158881  1.001402
11A01301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.010532  0.009537  0.000136 -0.007580  0.029839  4169.400823  1.000224
gain   0.019868  0.000723  0.000010  0.018444  0.021290  4338.269444  1.000798
sigma  0.047361  0.004267  0.000048  0.039157  0.055917  6457.546955  0.999733
11A02801
12103001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
```

```
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.191655  0.023730  0.000346  0.146856  0.240292  4498.171141  1.001456
gain   0.011650  0.001097  0.000017  0.009436  0.013729  4339.382634  1.001316
sigma  0.117103  0.008263  0.000097  0.101543  0.133441  6597.154607  0.999947
12501301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.122326  0.021395  0.000295  0.081987  0.165194  5511.377289  0.999846
gain   0.011366  0.000644  0.000008  0.010106  0.012645  5675.584332  0.999747
sigma  0.086903  0.011885  0.000133  0.064489  0.110351  8501.061160  1.000153
12803001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8790008244886146, but should be
The acceptance probability does not match the target. It is 0.8906154123658099, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.135046  0.007772  0.000137  0.119647  0.150134  3682.815799  1.000456
gain   0.009103  0.000904  0.000015  0.007356  0.010871  3723.588190  1.000394
sigma  0.026689  0.002216  0.000030  0.022455  0.031071  6210.823829  1.000357
12B02501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.154018  0.017457  0.000239  0.120150  0.187325  4505.427612  1.000633
gain   0.020132  0.001077  0.000014  0.018093  0.022300  4611.026267  1.000127
sigma  0.081796  0.007825  0.000093  0.067149  0.097561  5904.907619  1.000269
13301101
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8880438281440654, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.087679  0.023677  0.000336  0.040111  0.132718  4763.804124  1.000552
gain   0.016887  0.001399  0.000019  0.014250  0.019715  4950.051335  1.000554
sigma  0.113812  0.011453  0.000149  0.093139  0.137489  6286.805936  0.999740
13705801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8835051014362184, but should be
The acceptance probability does not match the target. It is 0.8804176090880285, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.179399  0.011048  0.000156  0.157838  0.201471  4998.929960  1.000474
gain   0.005590  0.000736  0.000010  0.004201  0.007088  5012.855918  1.000421
sigma  0.028065  0.004392  0.000056  0.020193  0.036946  6349.879130  0.999847
13903001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.7110196817393469, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.149215  0.014625  0.000214  0.121214  0.178189  4518.314073  1.000801
gain   0.009311  0.000897  0.000013  0.007578  0.011063  4373.659507  1.000822
sigma  0.039260  0.004896  0.000057  0.029660  0.048636  6361.700086  1.000034
13B05801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
```

```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.211507  0.012496  0.000155  0.185476  0.235433  5479.009332  1.000680
gain   0.009211  0.000822  0.000010  0.007647  0.010847  5394.764704  1.000812
sigma  0.050183  0.004422  0.000056  0.041986  0.059238  6506.841988  1.000438
14105901


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.6910073434160839, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.231482  0.015408  0.000240  0.201157  0.261073  4080.530894  1.000264
gain   0.010508  0.001505  0.000024  0.007579  0.013479  4145.108215  1.000116
sigma  0.033249  0.004688  0.000063  0.024605  0.042728  5506.741946  1.000491
14205801


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.200980  0.007455  0.000086  0.186409  0.215399  7332.062399  0.999655
gain   0.003979  0.000430  0.000005  0.003168  0.004837  7834.438542  0.999852
sigma  0.058768  0.005070  0.000052  0.048807  0.068441  9116.239226  0.999806
14602501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.138219  0.005362  0.000075  0.127900  0.148886  5715.276576  1.000392
gain   0.015171  0.001211  0.000016  0.012683  0.017480  5545.897713  1.000307
sigma  0.026430  0.002650  0.000029  0.021353  0.031716  6556.776092  0.999807
14906301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
```

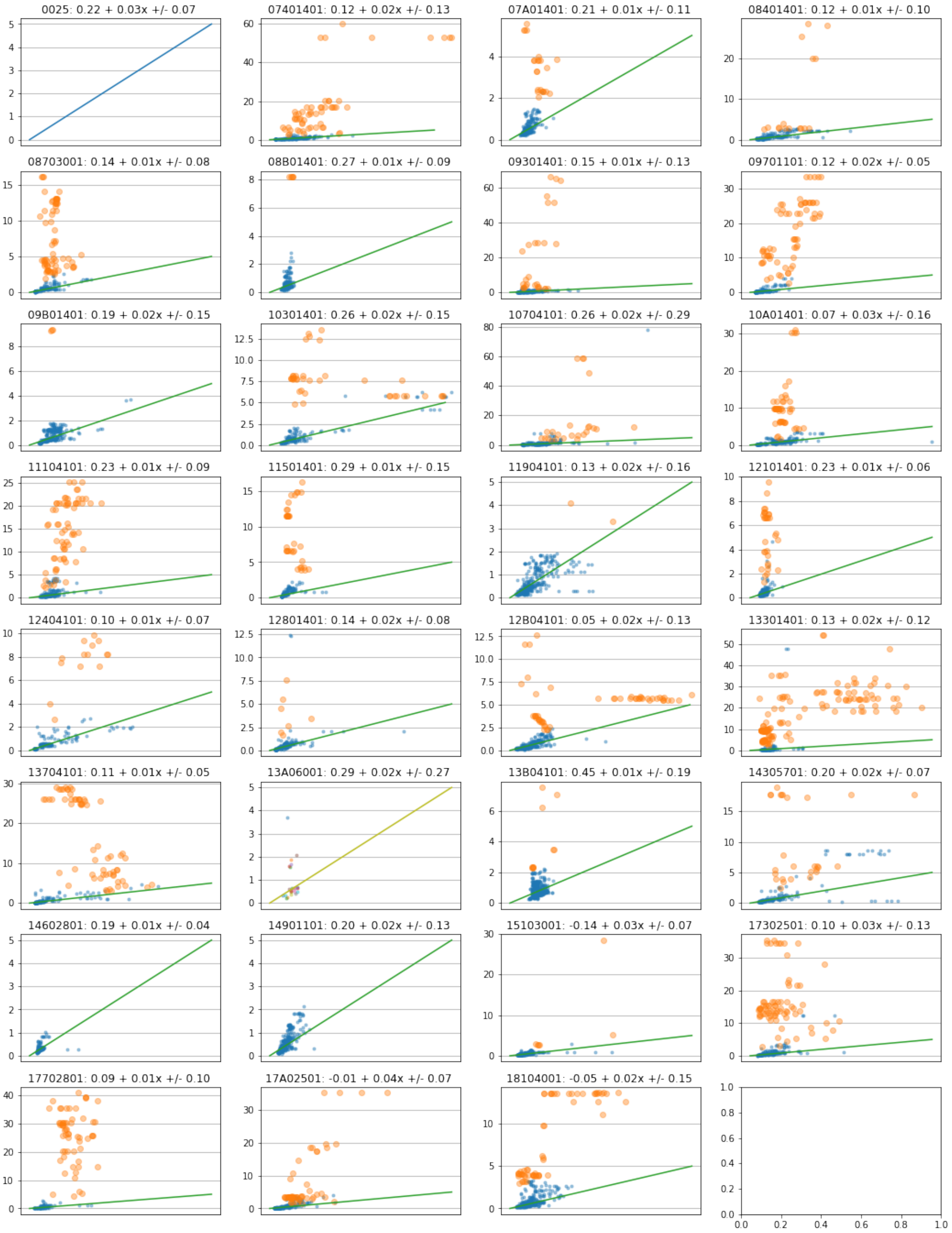
```
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8897658525712009, but should be c
There were 6 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.7197426442139775, but should be c
The acceptance probability does not match the target. It is 0.8812332327494439, but should be c


           mean        sd  mc_error       hpd_2.5  hpd_97.5        n_eff  \
f0     0.180396  0.032110  0.000633  1.182164e-01  0.244318  3105.794369
gain  -0.001061  0.004695  0.000093 -1.037604e-02  0.008041  3062.198228
sigma  0.005815  0.006544  0.000113  2.553304e-07  0.017962  3170.072260

           Rhat
f0     1.002125
gain   1.002434
sigma  1.000455
14B02501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.7118624508194362, but should be c
There were 26 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.46910540051832117, but should be


           mean        sd  mc_error       hpd_2.5  hpd_97.5        n_eff  \
f0     0.171558  0.028224  0.000491  1.172460e-01  0.226792  2820.027722
gain   0.003279  0.006674  0.000115 -1.006750e-02  0.016187  2848.488824
sigma  0.001336  0.001332  0.000020  1.583091e-08  0.004035  4476.296149

           Rhat
f0     1.000407
gain   1.000374
sigma  1.000310
14B05101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8848179646632695, but should be c
The acceptance probability does not match the target. It is 0.8848983539520715, but should be c
```

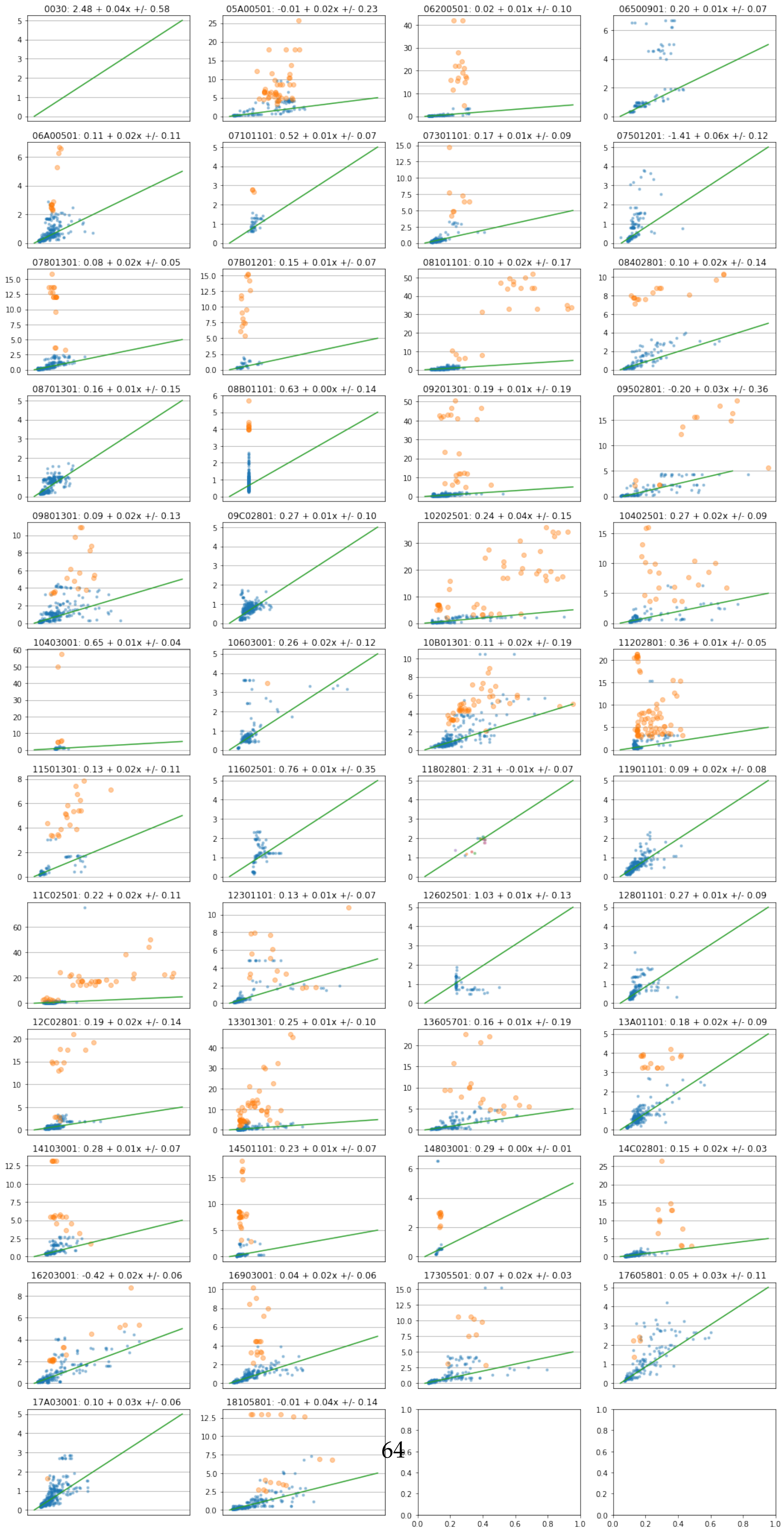
```
           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.083021  0.008838  0.000134  0.065929  0.100540  3572.141022  1.000304
gain   0.019534  0.001905  0.000029  0.016010  0.023494  3649.059127  1.000339
sigma  0.017755  0.001788  0.000025  0.014395  0.021298  5182.360500  1.000576
15703001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.7109955656590813, but should be
The acceptance probability does not match the target. It is 0.6704263601564552, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0    -0.149396  0.029729  0.000489 -0.205494 -0.090814  3324.639961  1.000254
gain   0.021785  0.002134  0.000035  0.017694  0.025904  3324.303793  1.000223
sigma  0.025987  0.001898  0.000031  0.022411  0.029693  4394.717503  1.001035
15A06401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.086523  0.006575  0.000101  0.073183  0.099094  5108.244620  1.001326
gain   0.023360  0.000946  0.000015  0.021505  0.025260  5147.050224  1.001479
sigma  0.038318  0.002818  0.000037  0.032924  0.043879  5933.980444  1.000014
16206301


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.060144  0.004993  0.000064  0.050433  0.070016  6175.001150  1.000127
gain   0.027817  0.000629  0.000008  0.026624  0.029089  6001.160217  1.000301
sigma  0.031524  0.003351  0.000039  0.025294  0.038215  7586.292304  0.999734
16506401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
```

```
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.132078  0.010505  0.000138  0.111066  0.152080  6035.879821  1.000894
gain   0.020806  0.001577  0.000020  0.017670  0.023772  5962.244163  1.000227
sigma  0.049308  0.007157  0.000087  0.036177  0.063671  6635.817012  1.000125
16904101


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.8900775906670055, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.104576  0.008732  0.000129  0.087420  0.121779  4938.579723  1.000787
gain   0.019411  0.000799  0.000012  0.017766  0.020942  4936.064654  1.000524
sigma  0.039422  0.003310  0.000039  0.033169  0.045960  6446.355350  1.000332
16C06401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.156868  0.011962  0.000173  0.132993  0.180150  5387.729202  1.000331
gain   0.022573  0.001300  0.000019  0.020030  0.025155  5408.349554  1.000419
sigma  0.065352  0.005002  0.000064  0.055550  0.075091  6539.068246  1.000179
17403001


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.084062  0.011934  0.000160  0.060373  0.107266  5943.268241  1.001062
gain   0.018830  0.001106  0.000016  0.016597  0.020873  5891.996758  1.000682
sigma  0.052125  0.005590  0.000061  0.042047  0.063755  6729.105646  1.000293
17705701
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.891176165716524, but should be cl
There were 1 divergences after tuning. Increase `target_accept` or reparameterize.
There were 9 divergences after tuning. Increase `target_accept` or reparameterize.
There were 1 divergences after tuning. Increase `target_accept` or reparameterize.
There were 5 divergences after tuning. Increase `target_accept` or reparameterize.
There were 3 divergences after tuning. Increase `target_accept` or reparameterize.


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.247089  0.429303  0.008118 -0.604834  1.088015  3329.734547  1.002050
gain   0.015726  0.009665  0.000184 -0.003892  0.034387  3185.597150  1.002061
sigma  0.382361  0.165376  0.002682  0.119565  0.716508  4075.122835  1.000950
17805501


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.7025566731874543, but should be c


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.020202  0.013073  0.000194 -0.005407  0.046093  3882.835985  0.999996
gain   0.033512  0.002217  0.000032  0.029318  0.038053  3937.262425  1.000159
sigma  0.047243  0.003331  0.000042  0.040902  0.053945  5131.972720  1.000286
17B06401


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.003426  0.009656  0.000170 -0.015625  0.022025  3716.534231  1.002607
gain   0.076255  0.002789  0.000049  0.070758  0.081655  3671.890082  1.002562
sigma  0.038285  0.002704  0.000035  0.033297  0.043843  5668.565259  1.000126
CUGN_line_93
05400601


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
```

```
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]
The acceptance probability does not match the target. It is 0.7146034340635464, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.072427  0.017140  0.000285  0.037927  0.105615  4306.641080  1.001511
gain   0.017482  0.002753  0.000044  0.012035  0.022852  4443.238809  1.001189
sigma  0.039075  0.007796  0.000123  0.025698  0.055118  4783.163936  0.999717
05C00601


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.100499  0.004490  0.000059  0.091756  0.109347  5523.838570  1.000096
gain   0.011646  0.000520  0.000006  0.010672  0.012679  5613.035691  1.000119
sigma  0.018917  0.002708  0.000030  0.014017  0.024320  7928.202047  1.000070
06300601


Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (10 chains in 2 jobs)
NUTS: [sigma, gain, f0]


           mean        sd  mc_error   hpd_2.5  hpd_97.5        n_eff      Rhat
f0     0.089558  0.009696  0.000133  0.070289  0.107870  4914.557533  1.001387
gain   0.012793  0.000736  0.000010  0.011371  0.014285  5038.135720  1.001092
sigma  0.040427  0.006008  0.000071  0.029850  0.052800  7094.426720  1.000245
```


0025: 0.22 + 0.03x +/- 0.07
07401401: 0.12 + 0.02x +/- 0.13
07A01401: 0.21 + 0.01x +/- 0.11
08401401: 0.12 + 0.01x +/- 0.10
08703001: 0.14 + 0.01x +/- 0.08
08B01401: 0.27 + 0.01x +/- 0.09
09301401: 0.15 + 0.01x +/- 0.13
09701101: 0.12 + 0.02x +/- 0.05
09B01401: 0.19 + 0.02x +/- 0.15
10301401: 0.26 + 0.02x +/- 0.15
10704101: 0.26 + 0.02x +/- 0.29
10A01401: 0.07 + 0.03x +/- 0.16
11104101: 0.23 + 0.01x +/- 0.09
11501401: 0.29 + 0.01x +/- 0.15
11904101: 0.13 + 0.02x +/- 0.16
12101401: 0.23 + 0.01x +/- 0.06
12404101: 0.10 + 0.01x +/- 0.07
12801401: 0.14 + 0.02x +/- 0.08
12B04101: 0.05 + 0.02x +/- 0.13
13301401: 0.13 + 0.02x +/- 0.12
13704101: 0.11 + 0.01x +/- 0.05
13A06001: 0.29 + 0.02x +/- 0.27
13B04101: 0.45 + 0.01x +/- 0.19
14305701: 0.20 + 0.02x +/- 0.07
14602801: 0.19 + 0.01x +/- 0.04
14901101: 0.20 + 0.02x +/- 0.13
15103001: -0.14 + 0.03x +/- 0.07
17302501: 0.10 + 0.03x +/- 0.13
17702801: 0.09 + 0.01x +/- 0.10
17A02501: -0.01 + 0.04x +/- 0.07
18104001: -0.05 + 0.02x +/- 0.15
0.0
0.2
0.4
0.6
0.8
1.0



0030: 2.48 + 0.04x +/- 0.58

05A00501: -0.01 + 0.02x +/- 0.23

06200501: 0.02 + 0.01x +/- 0.10

06500901: 0.20 + 0.01x +/- 0.07

06A00501: 0.11 + 0.02x +/- 0.11

07101101: 0.52 + 0.01x +/- 0.07

07301101: 0.17 + 0.01x +/- 0.09

07501201: -1.41 + 0.06x +/- 0.12

07801301: 0.08 + 0.02x +/- 0.05

07B01201: 0.15 + 0.01x +/- 0.07

08101101: 0.10 + 0.02x +/- 0.17

08402801: 0.10 + 0.02x +/- 0.14

08701301: 0.16 + 0.01x +/- 0.15

08B01101: 0.63 + 0.00x +/- 0.14

09201301: 0.19 + 0.01x +/- 0.19

09502801: -0.20 + 0.03x +/- 0.36

09801301: 0.09 + 0.02x +/- 0.13

09C02801: 0.27 + 0.01x +/- 0.10

10202501: 0.24 + 0.04x +/- 0.15

10402501: 0.27 + 0.02x +/- 0.09

10403001: 0.65 + 0.01x +/- 0.04

10603001: 0.26 + 0.02x +/- 0.12

10B01301: 0.11 + 0.02x +/- 0.19

11202801: 0.36 + 0.01x +/- 0.05

11501301: 0.13 + 0.02x +/- 0.11

11602501: 0.76 + 0.01x +/- 0.35

11802801: 2.31 + -0.01x +/- 0.07

11901101: 0.09 + 0.02x +/- 0.08

11C02501: 0.22 + 0.02x +/- 0.11

12301101: 0.13 + 0.01x +/- 0.07

12602501: 1.03 + 0.01x +/- 0.13

12801101: 0.27 + 0.01x +/- 0.09

12C02801: 0.19 + 0.02x +/- 0.14

13301301: 0.25 + 0.01x +/- 0.10

13605701: 0.16 + 0.01x +/- 0.19

13A01101: 0.18 + 0.02x +/- 0.09

14103001: 0.28 + 0.01x +/- 0.07

14501101: 0.23 + 0.01x +/- 0.07

14803001: 0.29 + 0.00x +/- 0.01

14C02801: 0.15 + 0.02x +/- 0.03

16203001: -0.42 + 0.02x +/- 0.06

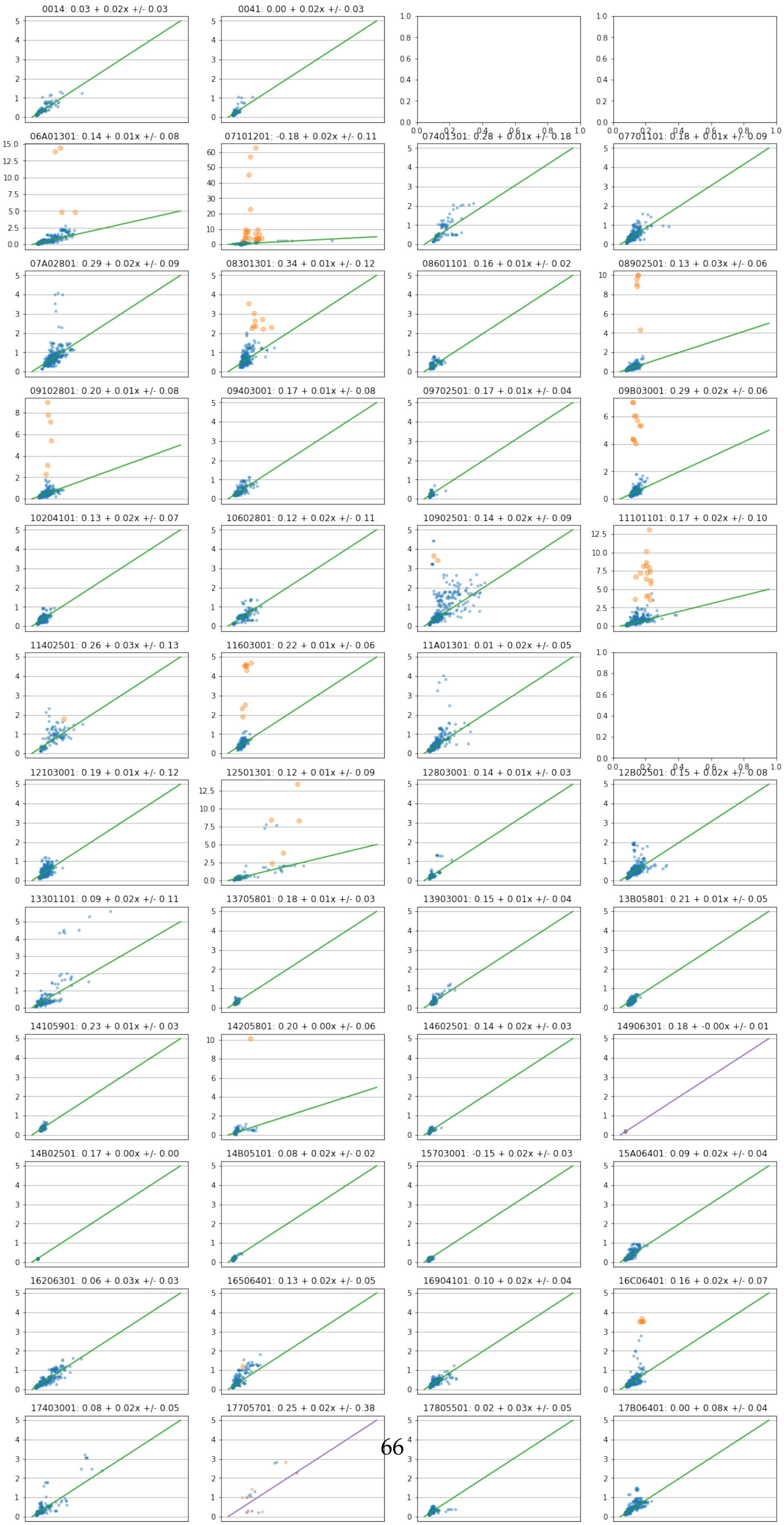
16903001: 0.04 + 0.02x +/- 0.06

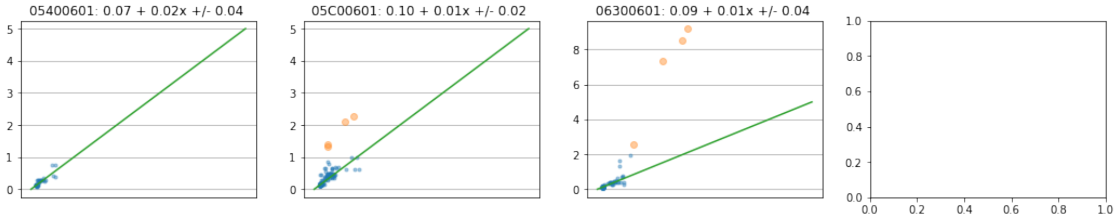
17305501: 0.07 + 0.02x +/- 0.03

17605801: 0.05 + 0.03x +/- 0.11

17A03001: 0.10 + 0.03x +/- 0.06

18105801: -0.01 + 0.04x +/- 0.14


0014: 0.03 + 0.02x +/- 0.03
0041: 0.00 + 0.02x +/- 0.03
06A01301: 0.14 + 0.01x +/- 0.08
07101201: -0.18 + 0.02x +/- 0.11
07401301: 0.28 + 0.01x +/- 0.18
07701101: 0.18 + 0.01x +/- 0.09
07A02801: 0.29 + 0.02x +/- 0.09
08301301: 0.34 + 0.01x +/- 0.12
08601101: 0.16 + 0.01x +/- 0.02
08902501: 0.13 + 0.03x +/- 0.06
09102801: 0.20 + 0.01x +/- 0.08
09403001: 0.17 + 0.01x +/- 0.08
09702501: 0.17 + 0.01x +/- 0.04
09B03001: 0.29 + 0.02x +/- 0.06
10204101: 0.13 + 0.02x +/- 0.07
10602801: 0.12 + 0.02x +/- 0.11
10902501: 0.14 + 0.02x +/- 0.09
11101101: 0.17 + 0.02x +/- 0.10
11402501: 0.26 + 0.03x +/- 0.13
11603001: 0.22 + 0.01x +/- 0.06
11A01301: 0.01 + 0.02x +/- 0.05
12103001: 0.19 + 0.01x +/- 0.12
12501301: 0.12 + 0.01x +/- 0.09
12803001: 0.14 + 0.01x +/- 0.03
12B02501: 0.15 + 0.02x +/- 0.08
13301101: 0.09 + 0.02x +/- 0.11
13705801: 0.18 + 0.01x +/- 0.03
13903001: 0.15 + 0.01x +/- 0.04
13B05801: 0.21 + 0.01x +/- 0.05
14105901: 0.23 + 0.01x +/- 0.03
14205801: 0.20 + 0.00x +/- 0.06
14602501: 0.14 + 0.02x +/- 0.03
14906301: 0.18 + -0.00x +/- 0.01
14B02501: 0.17 + 0.00x +/- 0.00
14B05101: 0.08 + 0.02x +/- 0.02
15703001: -0.15 + 0.02x +/- 0.03
15A06401: 0.09 + 0.02x +/- 0.04
16206301: 0.06 + 0.03x +/- 0.03
16506401: 0.13 + 0.02x +/- 0.05
16904101: 0.10 + 0.02x +/- 0.04
16C06401: 0.16 + 0.02x +/- 0.07
17403001: 0.08 + 0.02x +/- 0.05
17705701: 0.25 + 0.02x +/- 0.38
17805501: 0.02 + 0.03x +/- 0.05
17B06401: 0.00 + 0.08x +/- 0.04



```
In [192]: mission.profile_id.size

Out[192]: 81

In [169]: spray.isel(profile_id=spray.experiment=='CUGN_line_93')

Out[169]: <xarray.Dataset>
          Dimensions:        (depth: 100, profile_id: 1180, sat: 3)
          Coordinates:
            * profile_id     (profile_id) int64 2417 2418 2420 2422 2425 2428 2431 ...
              ndive          (profile_id) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
              datetime       (profile_id) datetime64[ns] 2005-04-21T20:29:09 ...
              lat            (profile_id) float64 32.7 32.71 32.71 32.71 32.7 32.7 ...
              lon            (profile_id) float64 -117.4 -117.4 -117.4 -117.4 -117.4 ...
              mission_id     (profile_id) int64 206 206 206 206 206 206 206 206 206 ...
              mission        (profile_id) object '05400601' '05400601' '05400601' ...
              experiment_id  (profile_id) int64 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ...
              experiment     (profile_id) object 'CUGN_line_93' 'CUGN_line_93' ...
            * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
            * sat            (sat) object 'aqua' 'terra' 'viirs'
              night_time     (profile_id) bool True True False False False False False ...
          Data variables:
              temp           (depth, profile_id) float64 14.28 13.38 13.89 14.03 15.59 ...
              sal            (depth, profile_id) float64 33.22 33.23 33.2 33.21 33.17 ...
              fl             (depth, profile_id) float64 0.5574 0.5857 0.847 1.025 ...
              chl_count      (profile_id, sat) float64 18.0 nan nan 18.0 nan nan 17.0 ...
              chl_mean       (profile_id, sat) float32 0.7469786 nan nan 0.7469786 nan ...
              chl_median     (profile_id, sat) float32 0.59614235 nan nan 0.59614235 ...
              chl_psdstd     (profile_id, sat) float64 0.4494 nan nan 0.4494 nan nan ...
              chl_std        (profile_id, sat) float32 0.37071112 nan nan 0.37071112 ...
              chl_sem        (profile_id, sat) float64 0.08738 nan nan 0.08738 nan nan ...
              fl_sum         (profile_id) float64 24.41 28.78 25.61 26.52 19.94 18.72 ...
              fl_sum_alt1    (profile_id) float64 2.787 2.929 4.235 5.124 1.787 1.628 ...
              fl_sum_alt2    (profile_id) float64 48.51 38.63 47.21 54.9 40.73 32.05 ...
              fl_sum05       (profile_id) float64 2.787 2.929 4.235 5.124 1.787 1.628 ...
              fl_sum30       (profile_id) float64 48.51 38.63 47.21 54.9 40.73 32.05 ...
```

```
In [ ]: print(ds)
```

```
In [ ]: ds = xr.open_dataset('../data/flMatch_SprayxAqua.nc')
        print(ds)
```

4.1.1 Crop the only the usefull data

Removing depths without any data (temp, sal, and fl), therefore depths of NaN. The number of profiles should be preserved.

```
In [57]: ds = ds.dropna(dim='depth', how='all', subset=['fl', 'temp', 'sal'])
         print('After removing depths without data')
         print(ds.dims)
         print('Included depths')
         print(ds.depth)
```

```
After removing depths without data
Frozen(SortedKeysDict(OrderedDict([('profile_id', 99752), ('depth', 100)])))
Included depths
<xarray.DataArray 'depth' (depth: 100)>
array([  10.,   20.,   30.,   40.,   50.,   60.,   70.,   80.,   90.,  100.,
        110.,  120.,  130.,  140.,  150.,  160.,  170.,  180.,  190.,  200.,
        210.,  220.,  230.,  240.,  250.,  260.,  270.,  280.,  290.,  300.,
        310.,  320.,  330.,  340.,  350.,  360.,  370.,  380.,  390.,  400.,
        410.,  420.,  430.,  440.,  450.,  460.,  470.,  480.,  490.,  500.,
        510.,  520.,  530.,  540.,  550.,  560.,  570.,  580.,  590.,  600.,
        610.,  620.,  630.,  640.,  650.,  660.,  670.,  680.,  690.,  700.,
        710.,  720.,  730.,  740.,  750.,  760.,  770.,  780.,  790.,  800.,
        810.,  820.,  830.,  840.,  850.,  860.,  870.,  880.,  890.,  900.,
        910.,  920.,  930.,  940.,  950.,  960.,  970.,  980.,  990., 1000.])
Coordinates:
  * depth    (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 90.0 ...
```

Only few profiles go below 500m. Let's restrict the data to the upper 500m

```
In [58]: ds = ds.isel(depth=ds.depth<=500)
         print(ds.depth)
         print(ds.dims)
```

```
<xarray.DataArray 'depth' (depth: 50)>
array([ 10.,  20.,  30.,  40.,  50.,  60.,  70.,  80.,  90., 100., 110., 120.,
       130., 140., 150., 160., 170., 180., 190., 200., 210., 220., 230., 240.,
       250., 260., 270., 280., 290., 300., 310., 320., 330., 340., 350., 360.,
       370., 380., 390., 400., 410., 420., 430., 440., 450., 460., 470., 480.,
       490., 500.])
Coordinates:
  * depth    (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 90.0 ...
Frozen(SortedKeysDict(OrderedDict([('profile_id', 99752), ('depth', 50)])))
```

4.1.2 Exceptions with bad time or position

Some profiles didn't have a valid date. This should be cutted off in the matchup script. I already updated matchup.py to do like that, so next time I run it the next code line will not be necessary

```
In [15]: #print('Before remove bad time or location')
         #print(ds.dims['profileid'])
         #ds = ds.dropna(dim='profileid', how='all', subset=['datetime', 'longitude', 'latitud
         #print('After')
         #print(ds.dims['profileid'])
         #ds.experiment[:4]
```

5 Preparing data

- Removing depths without any valid data
- Getting some derivate values, like relative time since start of the mission, etc.

5.0.1 Create dt as days since start of the mission

```
In [305]: def local_dt(lon):
              return np.timedelta64(lon, 's')
```

```
In [306]: (spray.lon/360*24).values
```

```
Out[306]: array([-7.9857025 , -7.98619383, -7.98648367, ..., -7.9998855 ,
                 -8.00221133, -8.0043205 ])
```

```
In [95]: spray.lon.to_series().apply(lambda x: np.timedelta64(x/360*24, 'D'))
```

```
        ---------------------------------------------------------------------------

        ValueError                                Traceback (most recent call last)

        <ipython-input-95-a969f70c2110> in <module>()
    ----> 1 spray.lon.to_series().apply(lambda x: np.timedelta64(x/360*24, 'D'))


        ~/.virtualenvs/fluorescence/lib/python3.6/site-packages/pandas/core/series.py in apply
       3192             else:
       3193                 values = self.astype(object).values
    -> 3194                 mapped = lib.map_infer(values, f, convert=convert_dtype)
       3195
       3196         if len(mapped) and isinstance(mapped[0], Series):


        pandas/_libs/src/inference.pyx in pandas._libs.lib.map_infer()
        <ipython-input-95-a969f70c2110> in <lambda>(x)
    ----> 1 spray.lon.to_series().apply(lambda x: np.timedelta64(x/360*24, 'D'))


        ValueError: Could not convert object to NumPy timedelta
```

```
In [308]: from pandas import Timedelta

In [309]: local_dt = spray.lon.to_series().apply(lambda x: Timedelta(x/360., 'D')).to_xarray()
          spray.datetime + local_dt

Out[309]: <xarray.DataArray (profile_id: 102058)>
          array(['2006-10-16T11:47:42.471014400', '2006-10-16T12:24:12.702233600',
                 '2006-10-16T12:59:24.658780800', ..., '2017-10-12T03:15:50.412243200',
                 '2017-10-12T06:12:44.039219200', '2017-10-12T09:06:10.446214400'],
                dtype='datetime64[ns]')
          Coordinates:
            * profile_id     (profile_id) int64 20215 20216 20217 20218 20219 20220 ...
              ndive          (profile_id) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
              datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
              lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
              lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
              mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
              mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
              experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
              experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...

In [310]: pd.Timedelta(spray.lon/360, unit='D')
```

```
    ---------------------------------------------------------------------------

    ValueError                                Traceback (most recent call last)

    <ipython-input-310-8e418e6d36f3> in <module>()
----> 1 pd.Timedelta(spray.lon/360, unit='D')


    pandas/_libs/tslibs/timedeltas.pyx in pandas._libs.tslibs.timedeltas.Timedelta.__new__


    ValueError: Value must be Timedelta, string, integer, float, timedelta or convertible
```

```
In [311]: spray.datetime

Out[311]: <xarray.DataArray 'datetime' (profile_id: 102058)>
          array(['2006-10-16T19:46:51.000000000', '2006-10-16T20:23:23.000000000',
                 '2006-10-16T20:58:36.000000000', ..., '2017-10-12T11:15:50.000000000',
                 '2017-10-12T14:12:52.000000000', '2017-10-12T17:06:26.000000000'],
                dtype='datetime64[ns]')
          Coordinates:
            * profile_id     (profile_id) int64 20215 20216 20217 20218 20219 20220 ...
              ndive          (profile_id) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
              datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
              lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
              lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
              mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
              mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
              experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
              experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
```

```
In [312]: local_time = (spray.datetime - local_dt)
          spray['night_time'] = (local_time.dt.hour < 6) | (local_time.dt.hour > 18)
          spray
```

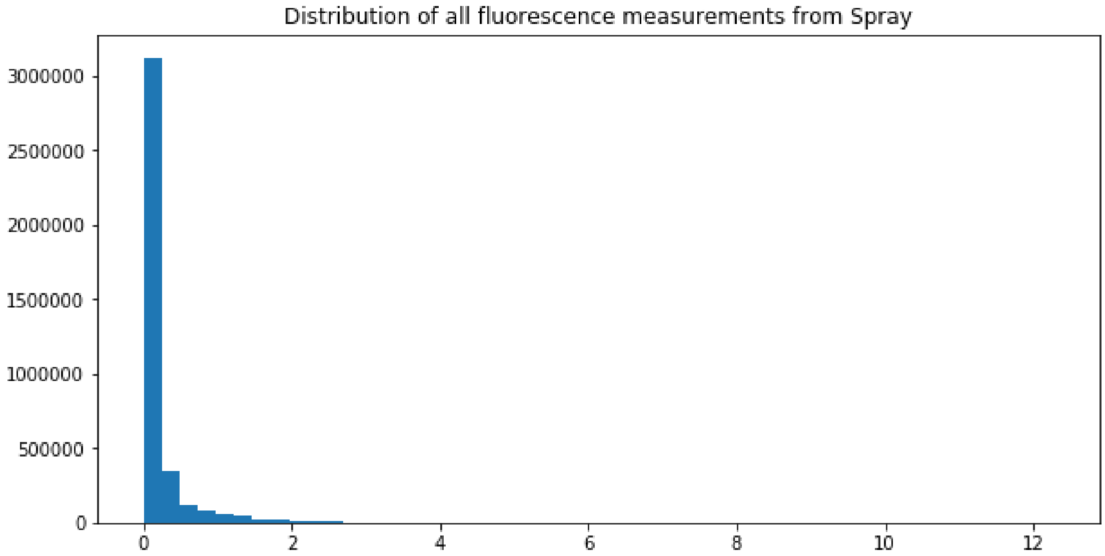
```
Out[312]: <xarray.Dataset>
          Dimensions:        (depth: 100, profile_id: 102058, sat: 4)
          Coordinates:
            * profile_id     (profile_id) int64 20215 20216 20217 20218 20219 20220 ...
              ndive          (profile_id) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
              datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
              lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
              lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
              mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
              mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
              experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
              experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
            * sat            (sat) object 'aqua' 'seawifs' 'terra' 'viirs'
            * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
          Data variables:
              chl_count      (profile_id, sat) float64 5.0 5.0 10.0 nan 4.0 4.0 8.0 ...
              chl_mean       (profile_id, sat) float32 1.8665981 1.3372794 1.0204775 ...
              chl_std        (profile_id, sat) float64 0.6485 0.3335 0.3682 nan 0.5478 ...
              chl_sem        (profile_id, sat) float64 0.29 0.1491 0.1164 nan 0.2739 ...
              temp           (depth, profile_id) float64 16.57 16.38 16.24 16.16 16.39 ...
              sal            (depth, profile_id) float64 33.42 33.41 33.43 33.43 33.43 ...
              fl             (depth, profile_id) float64 1.382 1.23 1.429 2.101 2.857 ...
              night_time     (profile_id) bool True True True True False False False ...
```

```
In [ ]: (spray.datetime - local_dt).dt
```

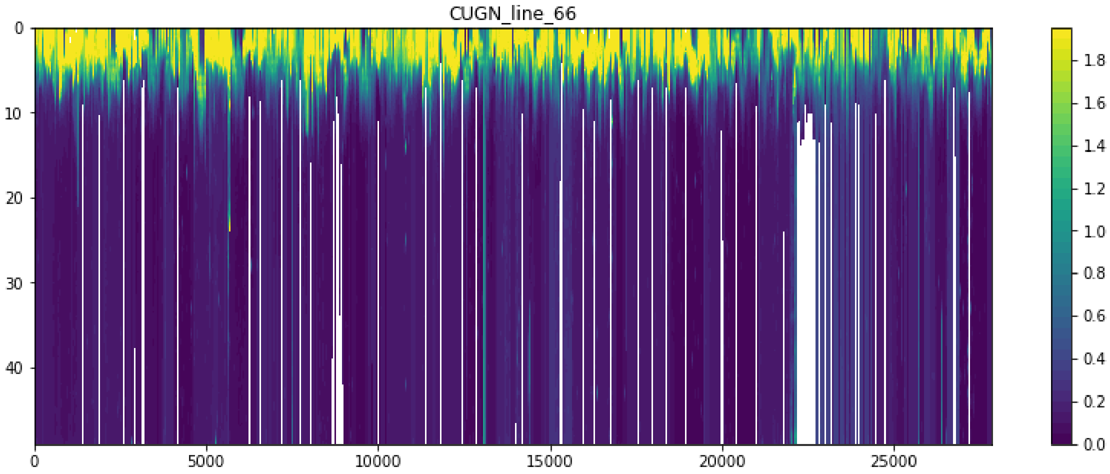
```
In [84]: np.timedelta?
```

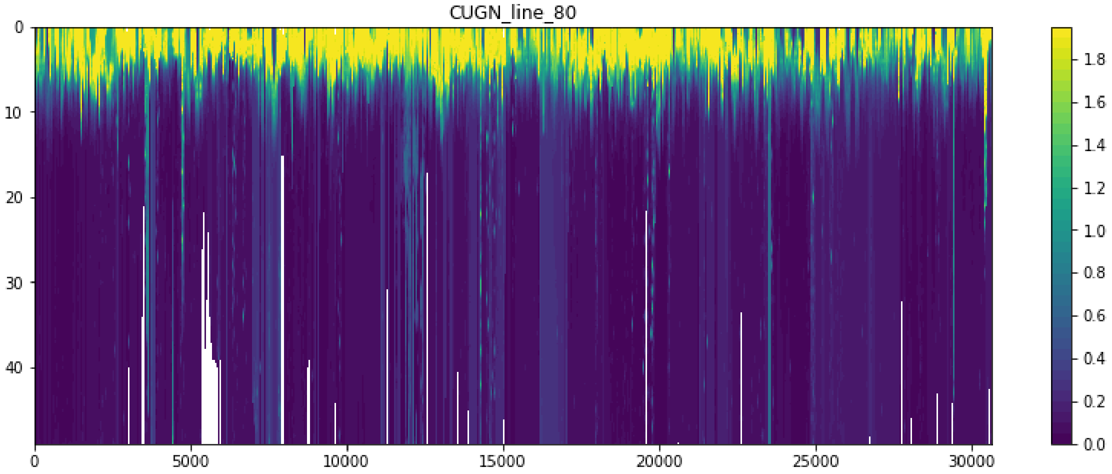
```
Object `np.timedelta` not found.
```

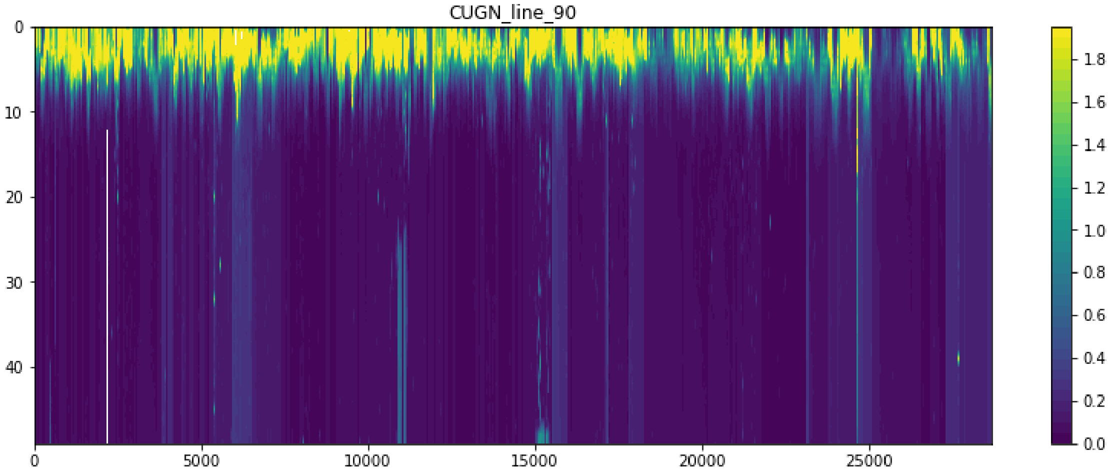
```
In [19]: x=ds.fluorescence.to_masked_array().compressed()
         fig = plt.figure(figsize=(10,5))
         trash = plt.hist(x, 50)
         trash = plt.title('Distribution of all fluorescence measurements from Spray')
```

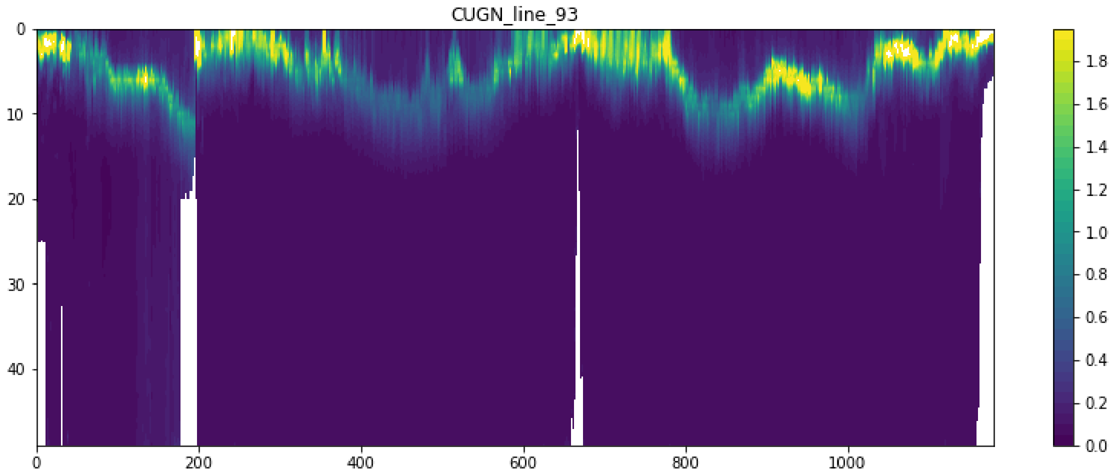


```
In [20]: scale = np.arange(0, 2, 0.05)
         for experiment_name, grp in ds.groupby('experiment'):
             plt.figure(figsize=(14,5))
             #plt.contourf(grp.datetime, grp.depth, grp.fluorescence, scale)
             plt.contourf(grp.fluorescence, scale)
             plt.colorbar()
             plt.gca().invert_yaxis()
             noprint = plt.title(experiment_name)
```




CUGN_line_80
0
10
20
30
40
0
5000
10000
15000
20000
25000
30000
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0


CUGN_line_90
0
10
20
30
40
0
5000
10000
15000
20000
25000
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0



What is the distribution of maximum profile depth? This will be important to guess how many profiles might measure a true zero. The shallow profiles will probably show a minimum fluorescence that is not zero.

ATENTION: find out the maximum depth of observation for each profile.

ATENTION: Some missions are clearly different, when depth show higher values. But it doesn't look like as a simple offset. Build a PDF of fl observed by each mission. Could I cluster the missions quality from the PDF?

6 Working on the offset (bias)

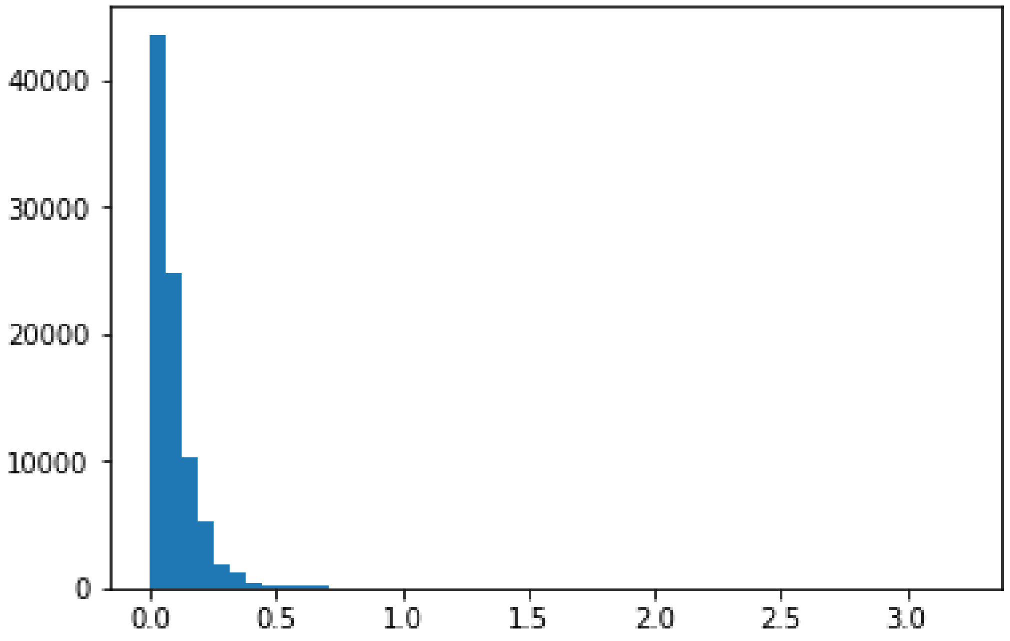
The fluorescence profile from Spray can lack any zero measurement. That could be due to particulates other than Chlorophyll or a bias in the sensor reading. In any case it is necessary to find this offset.

ADD AN EXAMPLE PROFILE SHOWING THE SUBSURFACE MAXIMA AND A MINIMA THAT IS NOT NECESSARILY IN THE BOTTOM

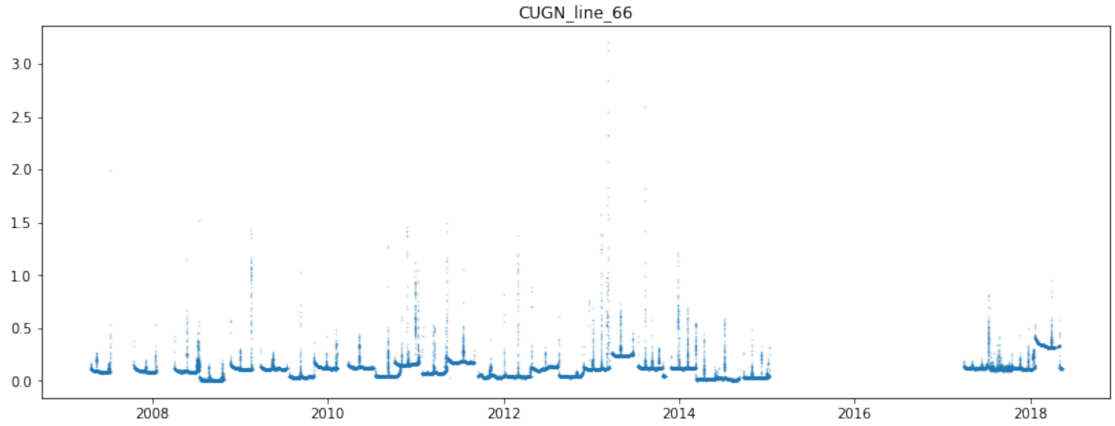
6.1 General statistics on minima fluorescence of each profile

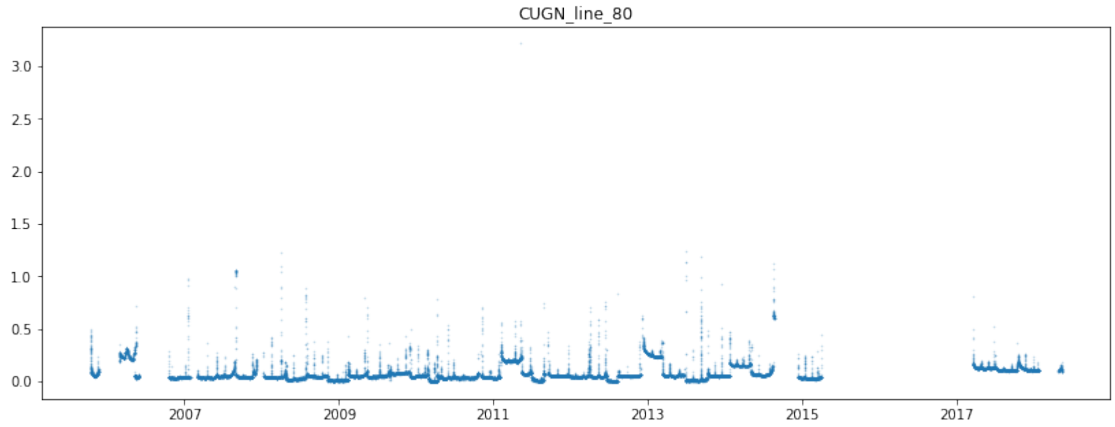
```
In [21]: # Minima fluorescence observed on each profile
         ds['fl_min'] = ds.fluorescence.min(dim='depth')
         h = plt.hist(ds.fl_min, bins=50)
         ds.fl_min.to_series().describe()
```

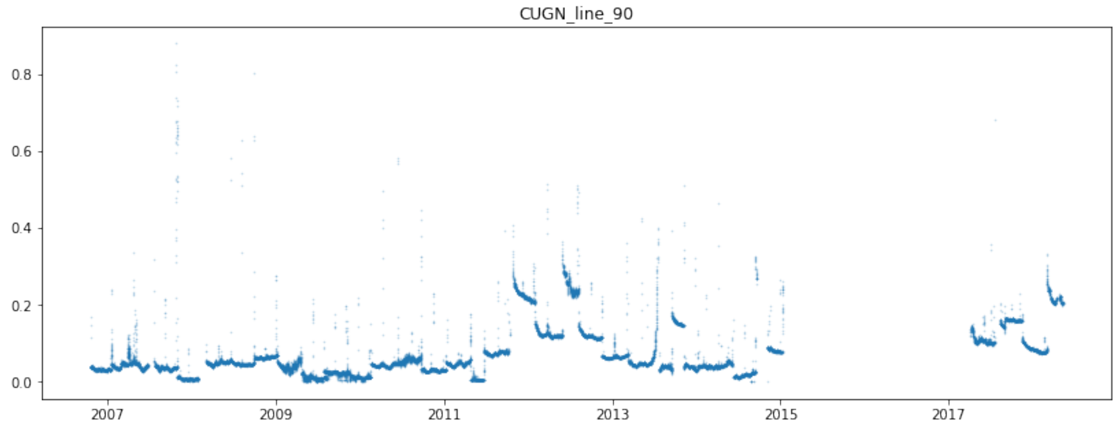
```
Out[21]: count    88349.000000
         mean         0.093620
         std          0.100737
         min         -0.005143
         25%          0.039273
         50%          0.060600
         75%          0.120000
         max          3.217909
         Name: fl_min, dtype: float64
```

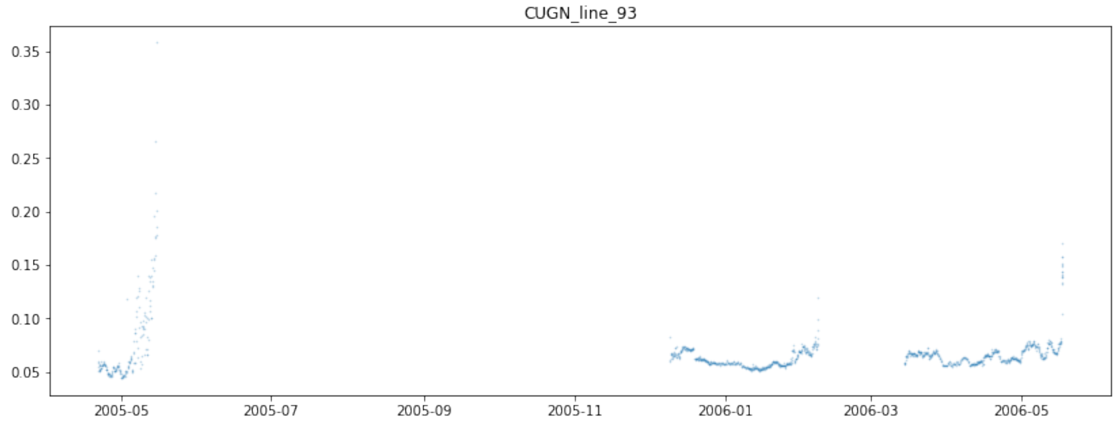


```
In [22]: for experiment_name, grp in ds.groupby('experiment'):
             fig = plt.figure(figsize=(14,5))
             plt.plot(grp.datetime, grp.fl_min, '.', markersize=1, alpha=0.3)
             trash = plt.title(experiment_name)
```





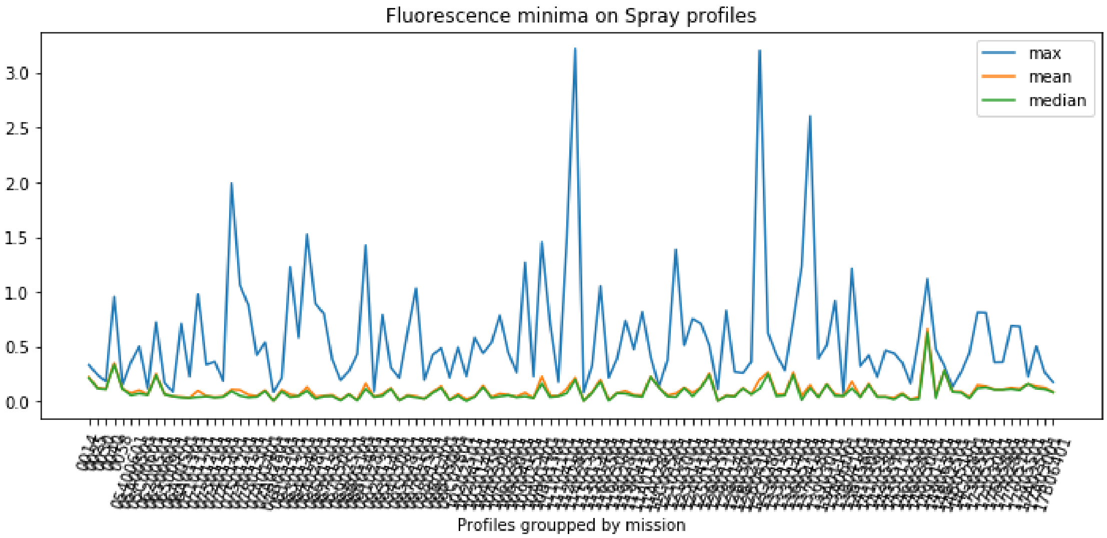




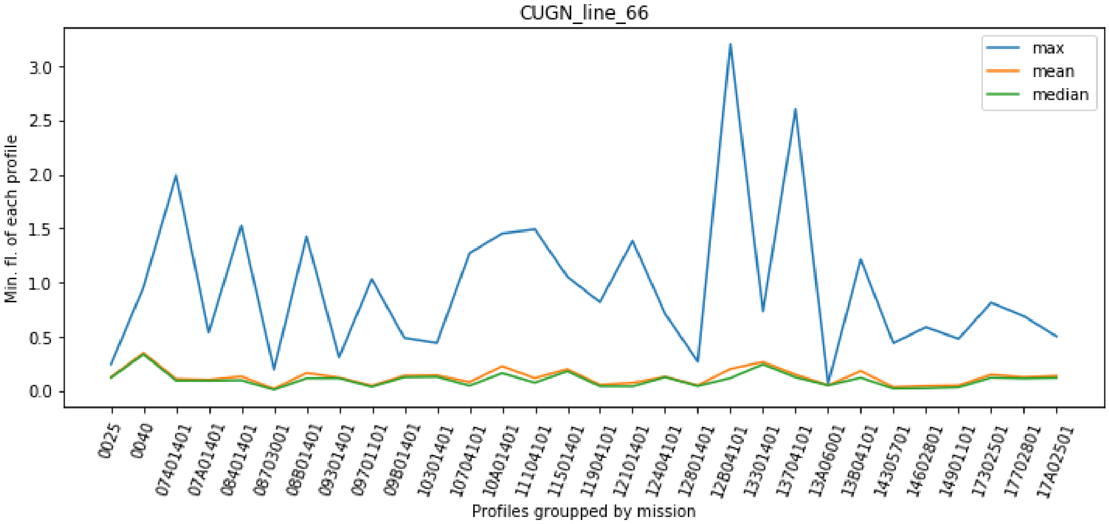
Looks like most of the cases are close to zero, which is a good thing. The few

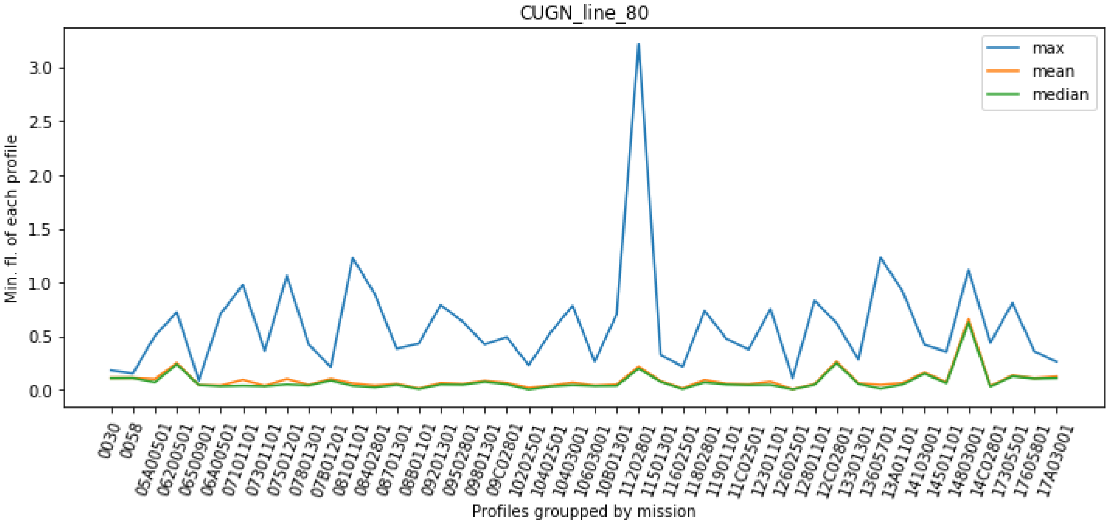
6.1.1 Are there missions with persistent strong bias?

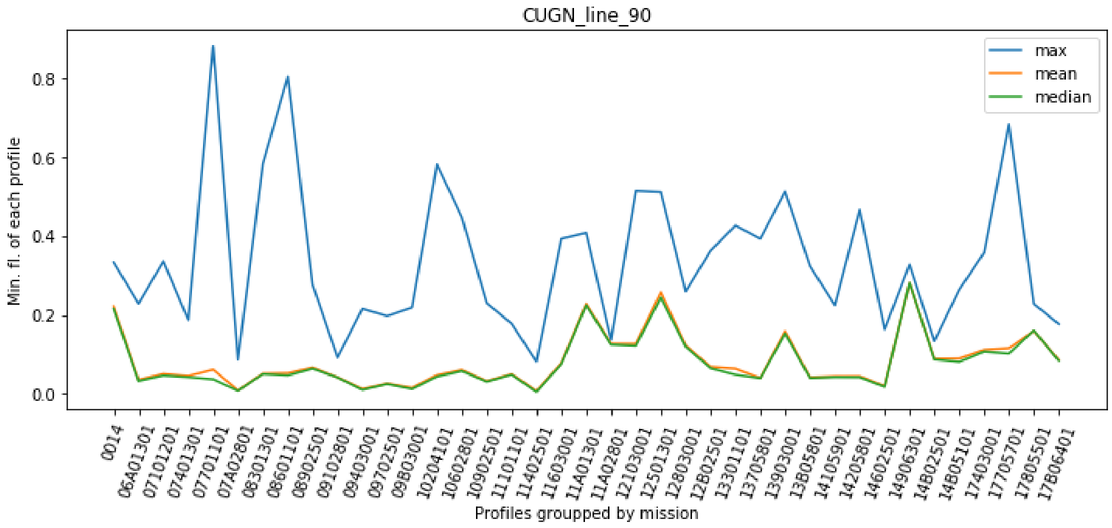
```
In [23]: # Groupping dataset by mission, so that apply procedures for each mission subset.
         grp = ds.groupby('mission')

         grp_mission = grp.first()['mission']

         plt.figure(figsize=(12, 4.5))
         plt.plot(grp_mission, grp.max()['fl_min'], label='max')
         plt.plot(grp_mission, grp.mean()['fl_min'], label='mean')
         plt.plot(grp_mission, grp.median()['fl_min'], label='median')

         plt.title('Fluorescence minima on Spray profiles')
         plt.xlabel('Profiles groupped by mission')
         plt.legend()
         ticks = plt.xticks(rotation=70)
```

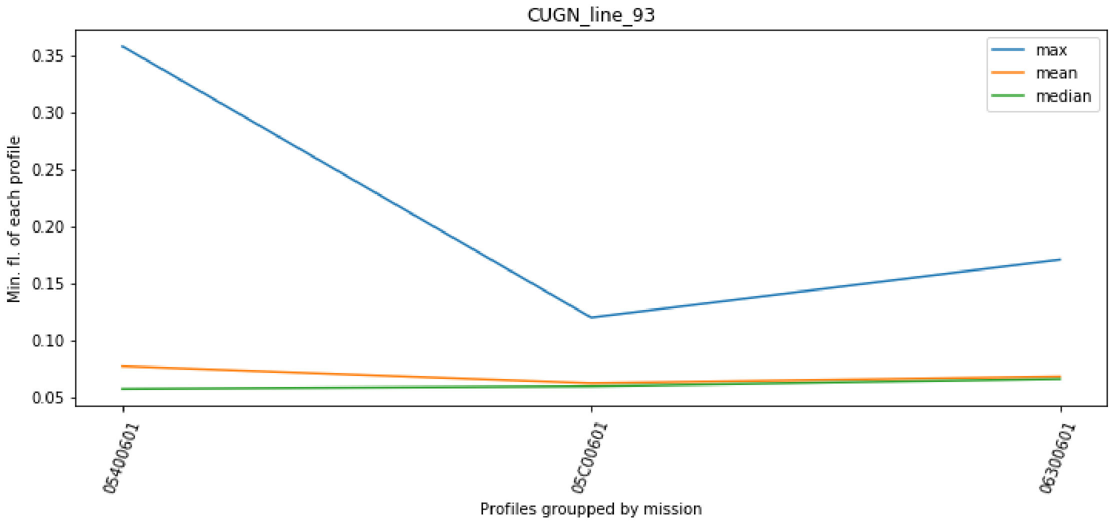


```
In [24]: for experiment_name, experiment in ds.groupby('experiment'):
             grp = experiment.groupby('mission')

             grp_mission = grp.first()['mission']

             plt.figure(figsize=(12, 4.5))
             plt.plot(grp_mission, grp.max()['fl_min'], label='max')
             plt.plot(grp_mission, grp.mean()['fl_min'], label='mean')
             plt.plot(grp_mission, grp.median()['fl_min'], label='median')

             plt.title(experiment_name)
             plt.xlabel('Profiles groupped by mission')
             plt.ylabel('Min. fl. of each profile')
             plt.legend()
             ticks = plt.xticks(rotation=70)
```









6.1.2 Removing missions with strong and consistent bias

Looks like most of the missions are fine. But at least one mission has a minimum persistently too high. I'll remove completely this mission for now.

ATENTION!!! I might want to go back here. Is this mission really bad measurements or an special event?

```
In [25]: grp = ds.groupby('mission')
         grp_mission = grp.first()['mission']
         compromised_missions = (grp_mission[grp.median()['fl_min'] > 1]).to_series()
         print("Removing missions: {0}".format(list(compromised_missions)))
```

```
        print("Original number of profiles: ", ds.dims['profileid'])
        idx = [m in compromised_missions for m in ds.mission.to_series()]
        ds = ds.drop(ds.profileid[idx], dim='profileid')
        print("Profiles after cleanned: ", ds.dims['profileid'])
        ds
```

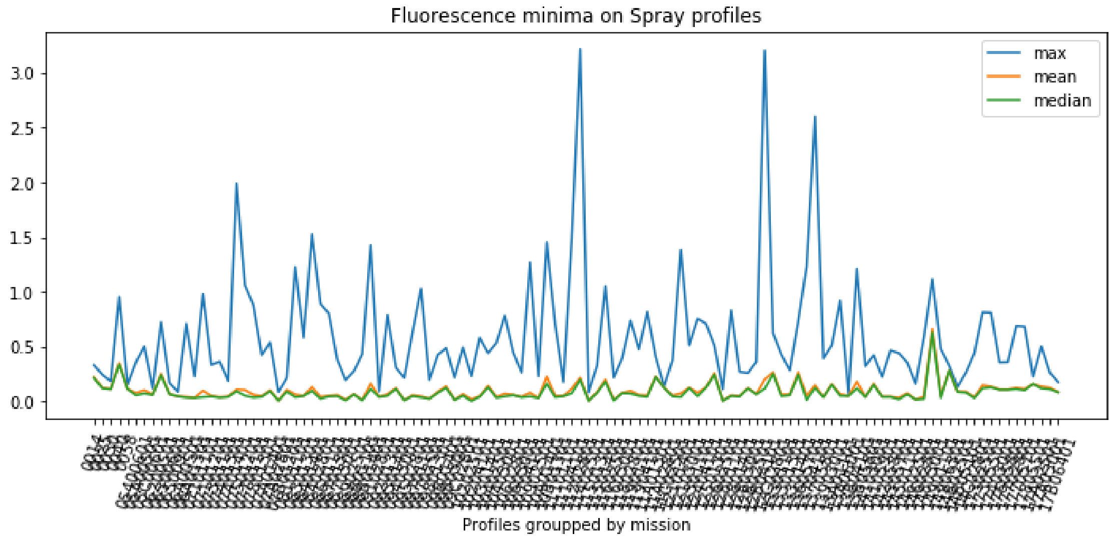
```
Removing missions: []
Original number of profiles:  88349
Profiles after cleanned:  88349
```

```
Out[25]: <xarray.Dataset>
         Dimensions:         (depth: 50, profileid: 88349)
         Coordinates:
           * profileid      (profileid) int64 20215 20216 20217 20218 20219 20220 ...
             ndive          (profileid) int64 ...
             datetime       (profileid) datetime64[ns] 2006-10-16T19:46:51 ...
             latitude       (profileid) float64 ...
             longitude      (profileid) float64 ...
             mission_id     (profileid) int64 ...
             mission        (profileid) object '06A00501' '06A00501' '06A00501' ...
             experiment_id  (profileid) int64 ...
             experiment     (profileid) object 'CUGN_line_80' 'CUGN_line_80' ...
           * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
         Data variables:
             chlor_a        (profileid) float32 ...
             dL             (profileid) float64 ...
             dt             (profileid) timedelta64[ns] ...
             temperature    (depth, profileid) float64 16.57 16.38 16.24 16.16 16.39 ...
             salinity       (depth, profileid) float64 33.42 33.41 33.43 33.43 33.43 ...
             fluorescence   (depth, profileid) float64 1.382 1.23 1.429 2.101 2.857 ...
             fl_min         (profileid) float64 0.2884 0.2508 0.2347 0.1904 0.144 ...
         Attributes:
             title:         Matchup between Spray and Aqua Chl measurements
             date_created:  2018-06-12 17:06:36 UTC
```

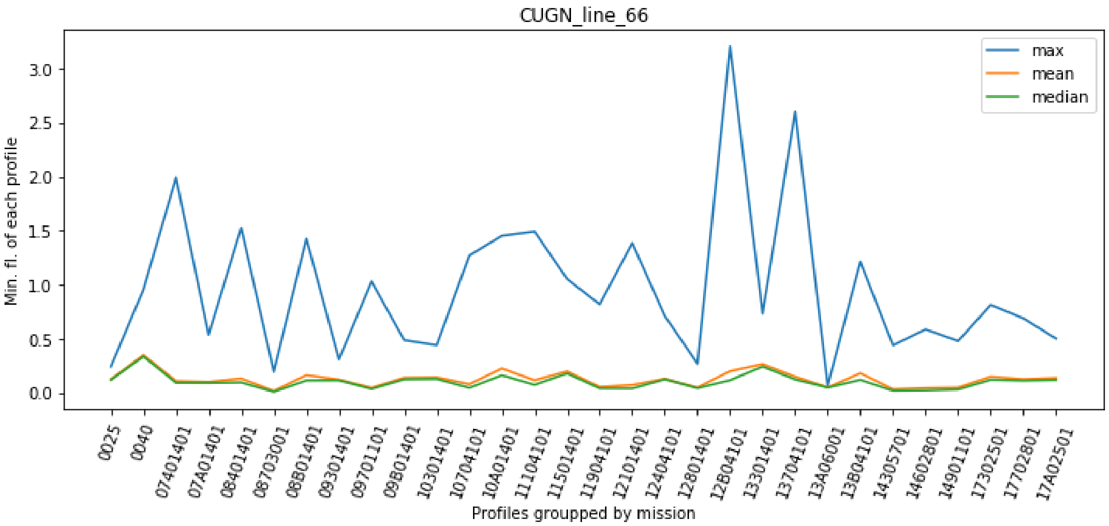
6.1.3 Reviewing plots without bad missions

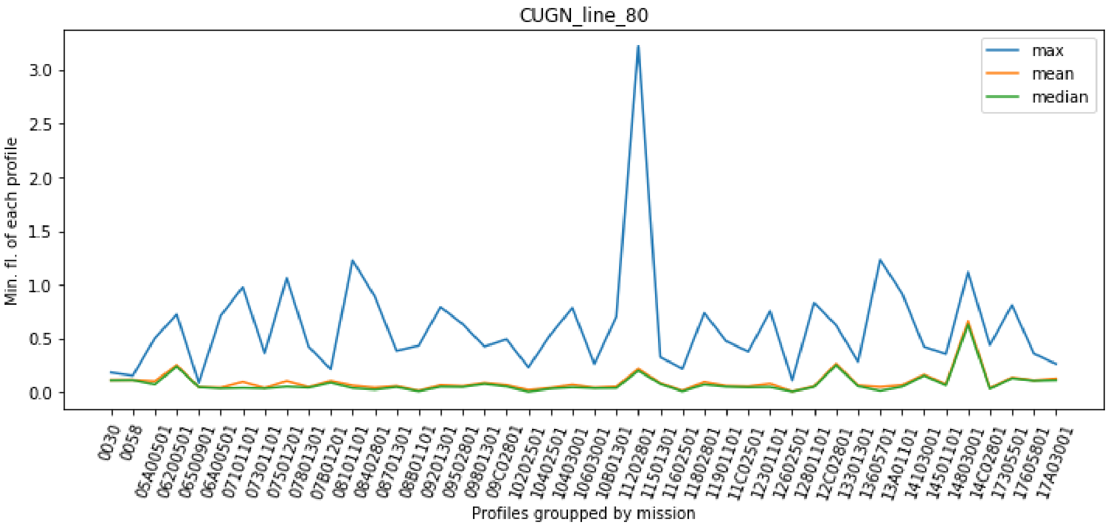
```
In [26]: # Update groupping to reflect removed missions
         grp = ds.groupby('mission')

         grp_mission = grp.first()['mission']

         plt.figure(figsize=(12, 4.5))
         plt.plot(grp_mission, grp.max()['fl_min'], label='max')
         plt.plot(grp_mission, grp.mean()['fl_min'], label='mean')
         plt.plot(grp_mission, grp.median()['fl_min'], label='median')
```

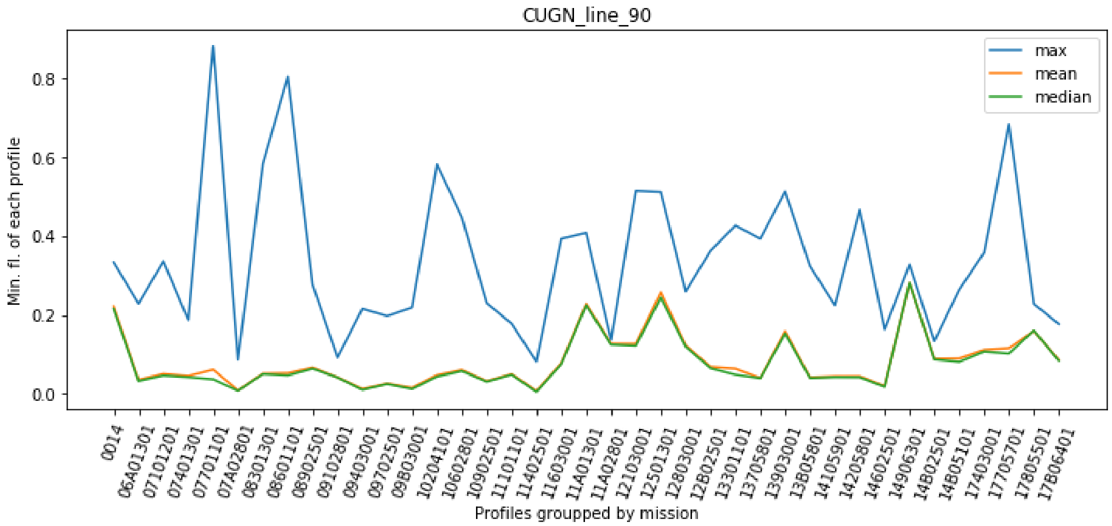
```
    plt.title('Fluorescence minima on Spray profiles')
    plt.xlabel('Profiles groupped by mission')
    plt.legend()
    ticks = plt.xticks(rotation=70)
```

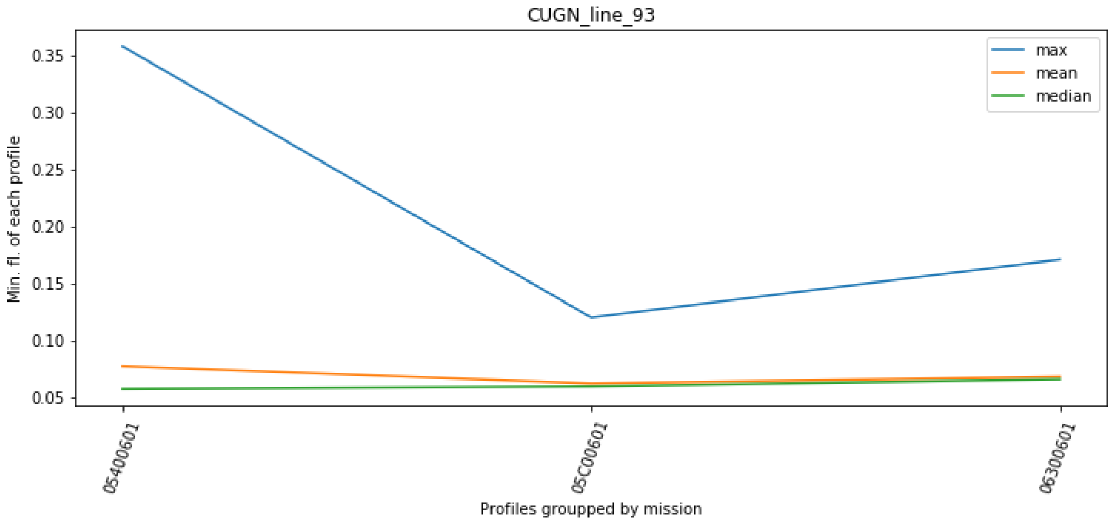


```
In [27]: for experiment_name, experiment in ds.groupby('experiment'):
             grp = experiment.groupby('mission')

             grp_mission = grp.first()['mission']

             plt.figure(figsize=(12, 4.5))
             plt.plot(grp_mission, grp.max()['fl_min'], label='max')
             plt.plot(grp_mission, grp.mean()['fl_min'], label='mean')
             plt.plot(grp_mission, grp.median()['fl_min'], label='median')

             plt.title(experiment_name)
             plt.xlabel('Profiles groupped by mission')
             plt.ylabel('Min. fl. of each profile')
             plt.legend()
             ticks = plt.xticks(rotation=70)
```

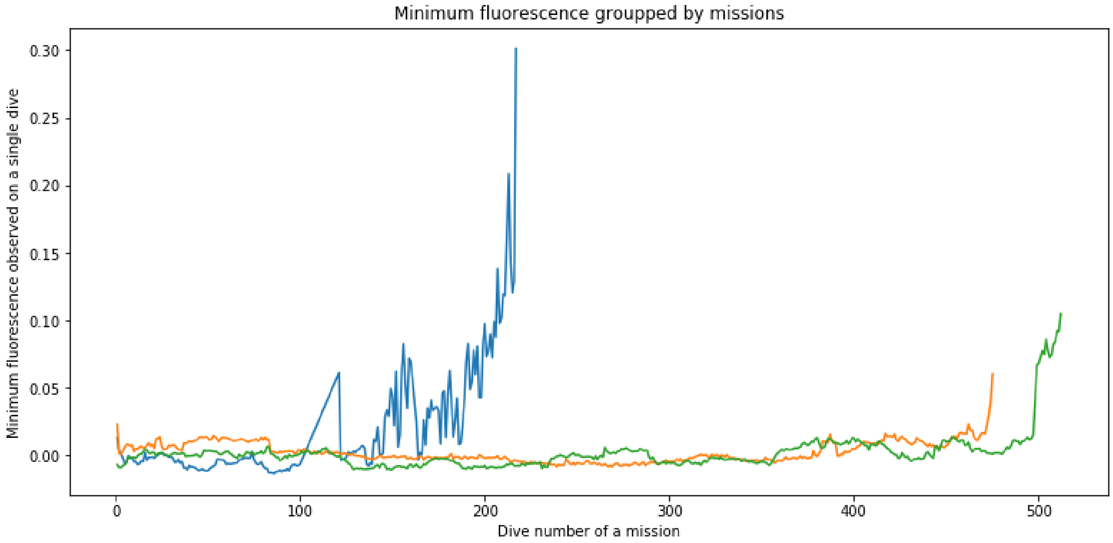

CUGN_line_66
max
mean
median
Min. fl. of each profile
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0025
0040
07401401
07A01401
08401401
08703001
08B01401
09301401
09701101
09B01401
10301401
10704101
10A01401
11104101
11501401
11904101
12101401
12404101
12801401
12B04101
13301401
13704101
13A06001
13B04101
14305701
14602801
14901101
17302501
17702801
17A02501
Profiles groupped by mission


CUGN_line_80
max
mean
median
Min. fl. of each profile
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0030
0058
05A00501
06200501
06500901
06A00501
07101101
07301101
07501201
07801301
07B01201
08101101
08402801
08701301
08B01101
09201301
09502801
09801301
09C02801
10202501
10402501
10403001
10603001
10B01301
11202801
11501301
11602501
11802801
11901101
11C02501
12301101
12602501
12801101
12C02801
13301301
13605701
13A01101
14103001
14501101
14803001
14C02801
17305501
17605801
17A03001
Profiles groupped by mission





```
In [28]: fig = plt.figure(figsize=(13,6))

         for label, m in grp:
             plt.plot(m.ndive, m.fl_min-m.fl_min.median())

         trash = plt.title('Minimum fluorescence groupped by missions')
         trash = plt.xlabel('Dive number of a mission')
         trash = plt.ylabel('Minimum fluorescence observed on a single dive')
```



It is interesting how many events seems to be consistent with neighbor dives. Could that be a real nature event? What are the other particulates that can optically respond like Chl-a?

There is a pattern here. Something happens around dive 350, latter again around dive 700. Note that the biggest event is coherent with that pattern, happenning around 1050.

ATENTION!!! Create a plot of offset versus distance to coast and another versus local depth. Could those events coincide with near coast profiles, thus more particulates including Chl-a?

6.2 Defining bias

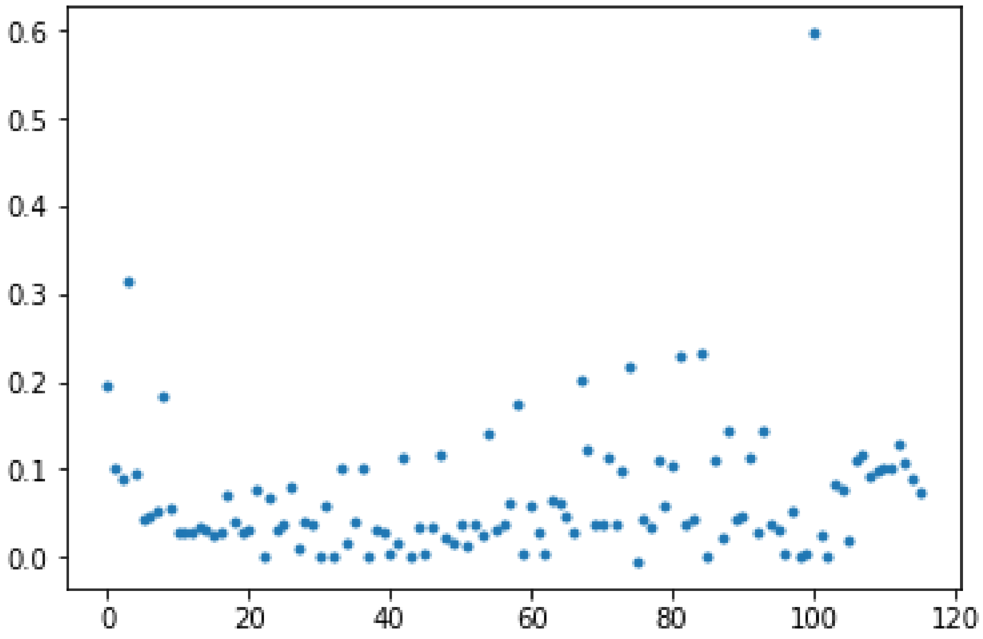
Let's define one consistent bias per mission, and adjust each full mission by that single bias.

6.2.1 Minimum Fluorescence per mission

!!ATENTION!! Improve this. Instead of taking the minimum value for the mission, think about a a more robust estimate. Maybe the bottom 1 percentile (teste between 0.1 - 1). Anything below that, i.e. negative, turns into zero.

```
In [29]: fl_mission_bias = ds.groupby('mission').min()['fluorescence']
         plt.plot(fl_mission_bias, '.')
```

```
Out[29]: [<matplotlib.lines.Line2D at 0x1144c9cf8>]
```

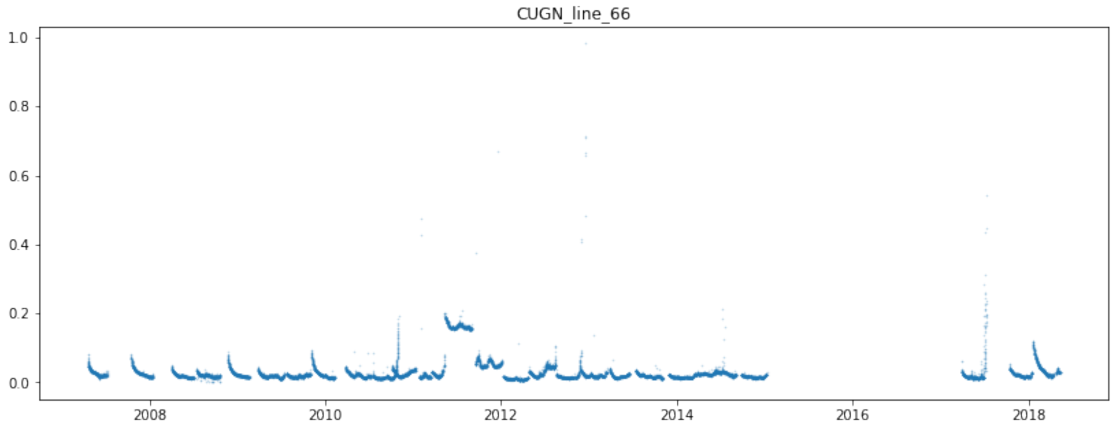


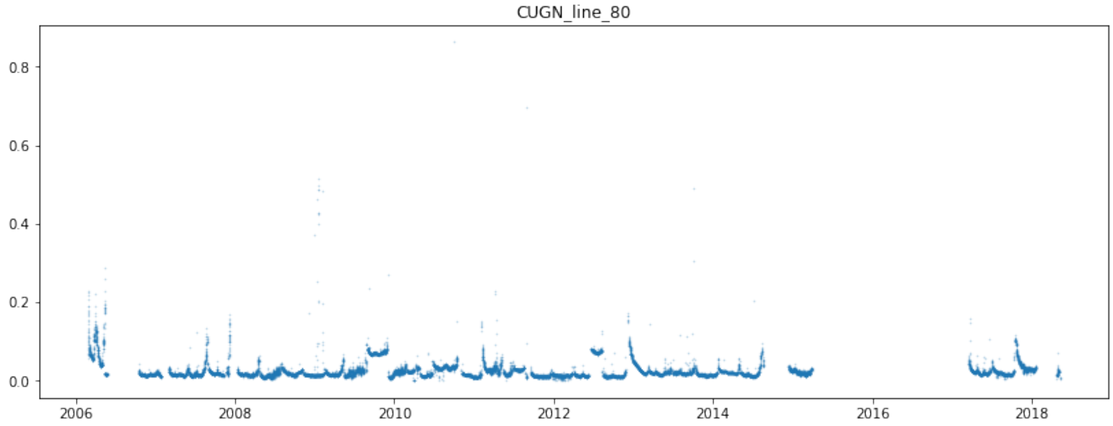
```
In [30]: fl_mission_bias = ds.groupby('mission').min()['fluorescence']
         print('Bias per mission')
         print(fl_mission_bias)

         ds['fl_unbias'] = ds.fluorescence.copy(deep=True)

         for m in fl_mission_bias:
             bias = float(m)
             m_name = str(m.mission.values)
             idx = ds.mission == m_name
             ds['fl_unbias'][:, idx] = ds.fluorescence.isel(profileid=idx) - bias
```

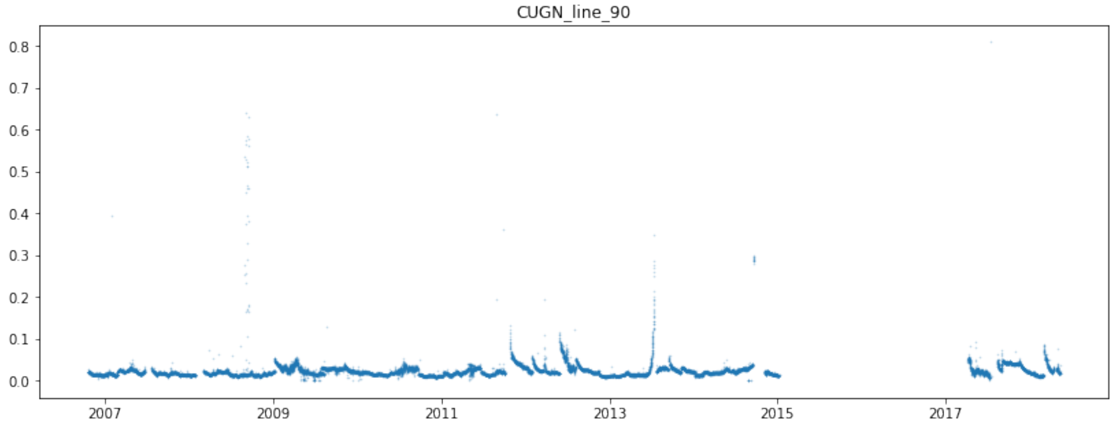
```
Bias per mission
<xarray.DataArray 'fluorescence' (mission: 116)>
array([ 1.970000e-01,  1.012500e-01,  9.000000e-02,  3.135000e-01,
        9.400000e-02,  4.400000e-02,  4.527273e-02,  5.127273e-02,
        1.847000e-01,  5.525000e-02,  2.900000e-02,  2.700000e-02,
        2.725000e-02,  3.300000e-02,  3.090000e-02,  2.618182e-02,
        2.700000e-02,  7.233333e-02,  4.133333e-02,  2.666667e-02,
        3.190909e-02,  7.740000e-02,  1.875000e-03,  6.900000e-02,
        3.042857e-02,  3.720000e-02,  8.100000e-02,  1.125000e-02,
        4.080000e-02,  3.600000e-02,  0.000000e+00,  5.775000e-02,
        1.500000e-03,  1.011429e-01,  1.620000e-02,  4.050000e-02,
        1.027500e-01,  0.000000e+00,  2.975000e-02,  2.910000e-02,
        3.000000e-03,  1.500000e-02,  1.122500e-01,  0.000000e+00,
        3.525000e-02,  3.000000e-03,  3.463636e-02,  1.155000e-01,
        2.280000e-02,  1.466667e-02,  3.600000e-02,  1.410000e-02,
        3.845455e-02,  2.370000e-02,  1.415455e-01,  3.000000e-02,
        3.666667e-02,  6.060000e-02,  1.740000e-01,  3.545455e-03,
        6.000000e-02,  2.754545e-02,  3.000000e-03,  6.450000e-02,
        6.262500e-02,  4.687500e-02,  2.890909e-02,  2.023636e-01,
        1.230000e-01,  3.790909e-02,  3.600000e-02,  1.130000e-01,
        3.825000e-02,  9.840000e-02,  2.173636e-01, -5.142857e-03,
        4.470000e-02,  3.525000e-02,  1.099091e-01,  5.966667e-02,
        1.057500e-01,  2.286667e-01,  3.840000e-02,  4.375000e-02,
        2.325000e-01,  4.615385e-04,  1.106667e-01,  2.100000e-02,
        1.443333e-01,  4.470000e-02,  4.500000e-02,  1.133333e-01,
        2.727273e-02,  1.442308e-01,  3.700000e-02,  3.270000e-02,
        4.500000e-03,  5.350000e-02,  0.000000e+00,  2.400000e-03,
        5.974615e-01,  2.430000e-02,  0.000000e+00,  8.154545e-02,
        7.560000e-02,  1.900000e-02,  1.118182e-01,  1.152000e-01,
        9.360000e-02,  9.736364e-02,  1.020000e-01,  1.020000e-01,
        1.278750e-01,  1.067500e-01,  9.042857e-02,  7.328571e-02])
Coordinates:
  * mission  (mission) object '0014' '0025' '0030' '0040' '0058' '05400601' ...
```

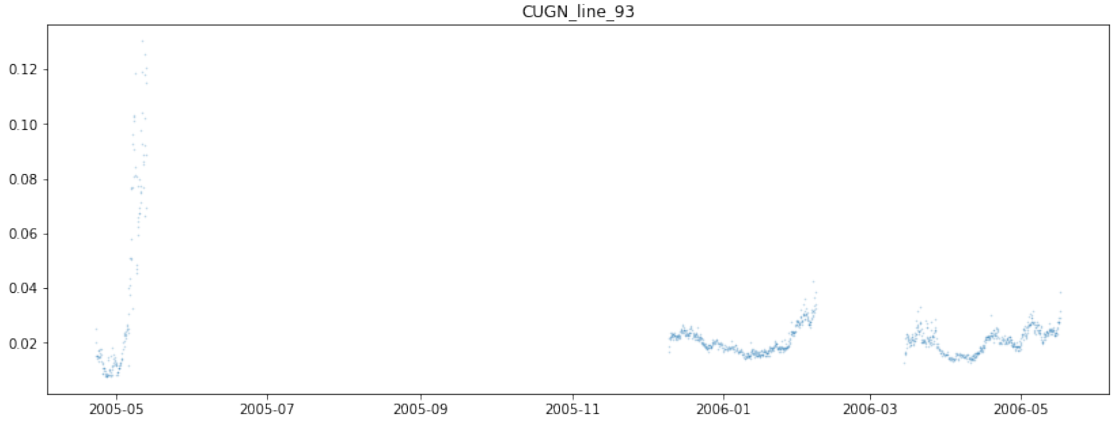
6.2.2 Reviewing plots for unbiased fluorescence

```
In [31]: for experiment_name, grp in ds.groupby('experiment'):
             fig = plt.figure(figsize=(14,5))
             plt.plot(grp.datetime, grp.fl_unbias.sel(depth=450), '.', markersize=1, alpha=0.3)
             trash = plt.title(experiment_name)
```

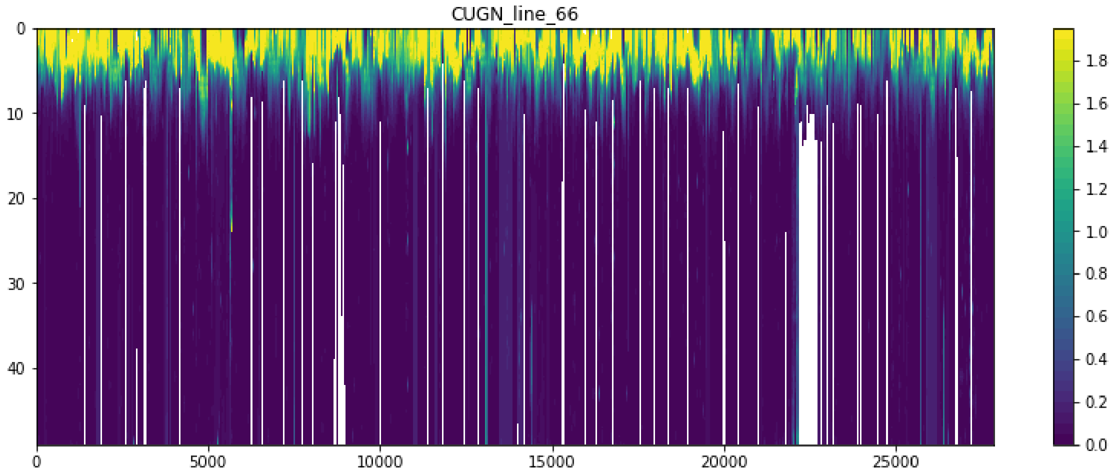


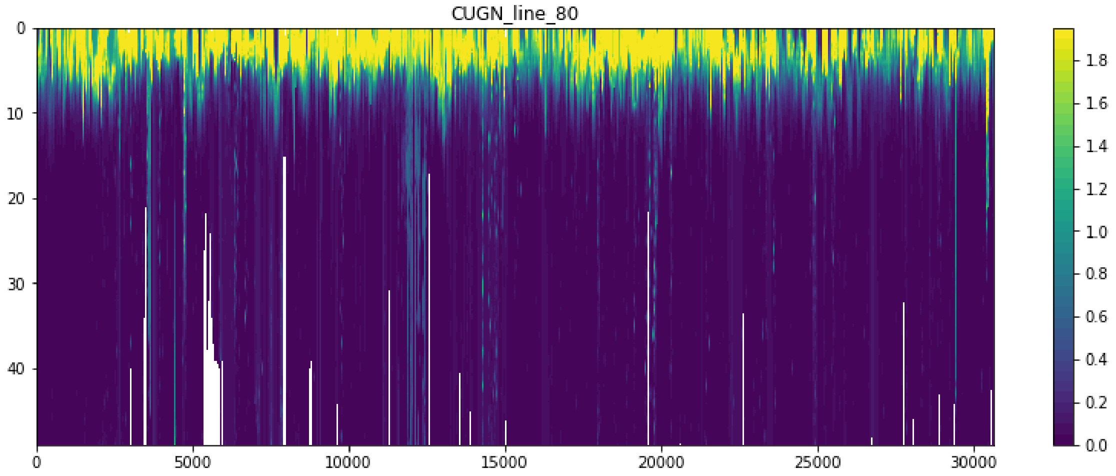

CUGN_line_80
0.8
0.6
0.4
0.2
0.0
2006
2008
2010
2012
2014
2016
2018

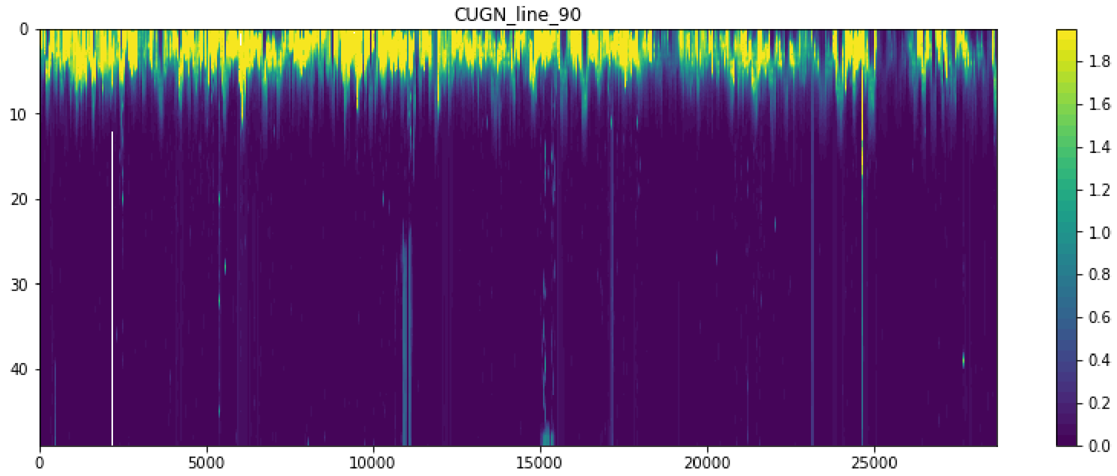

CUGN_line_90
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
2007
2009
2011
2013
2015
2017

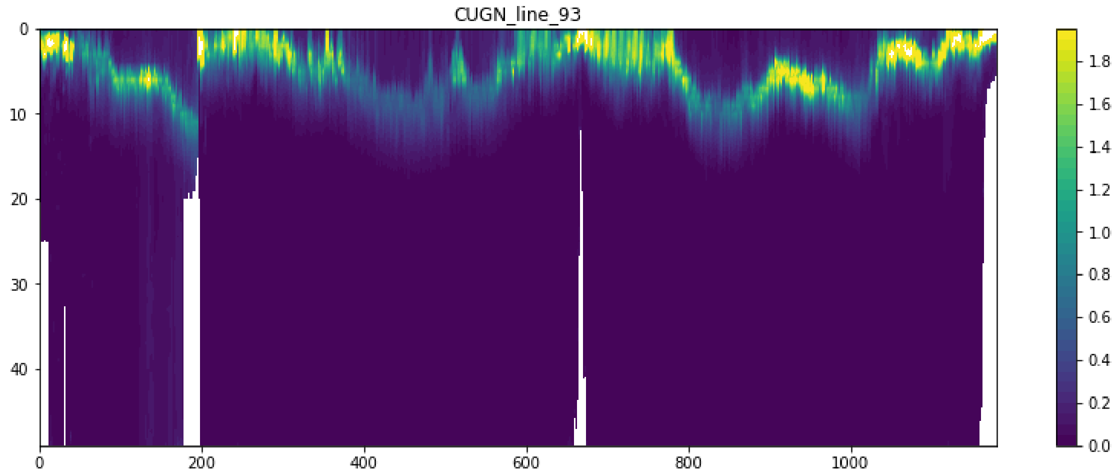

CUGN_line_93
0.12
0.10
0.08
0.06
0.04
0.02
2005-05
2005-07
2005-09
2005-11
2006-01
2006-03
2006-05

```
In [32]: scale = np.arange(0, 2, 0.05)
         for experiment_name, grp in ds.groupby('experiment'):
             plt.figure(figsize=(14,5))
             #plt.contourf(grp.datetime, grp.depth, grp.fluorescence, scale)
             plt.contourf(grp.fl_unbias, scale)
             plt.colorbar()
             plt.gca().invert_yaxis()
             noprint = plt.title(experiment_name)
```





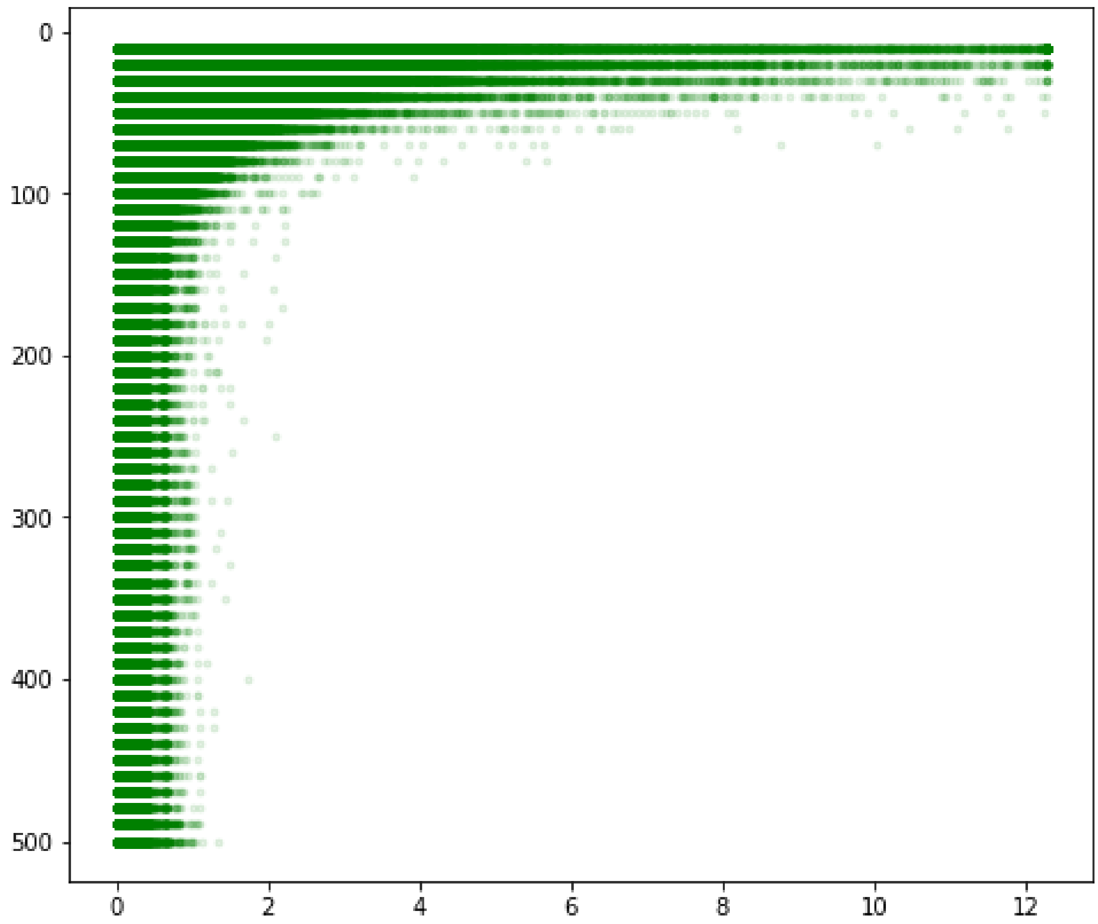




```
In [33]: ds

Out[33]: <xarray.Dataset>
         Dimensions:          (depth: 50, profileid: 88349)
         Coordinates:
           * profileid        (profileid) int64 20215 20216 20217 20218 20219 20220 ...
             ndive            (profileid) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
             datetime         (profileid) datetime64[ns] 2006-10-16T19:46:51 ...
             latitude         (profileid) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
             longitude        (profileid) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
             mission_id       (profileid) int64 106 106 106 106 106 106 106 106 106 106 ...
             mission          (profileid) object '06A00501' '06A00501' '06A00501' ...
             experiment_id    (profileid) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
```

```
    experiment     (profileid) object 'CUGN_line_80' 'CUGN_line_80' ...
  * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
Data variables:
    chlor_a        (profileid) float32 ...
    dL             (profileid) float64 ...
    dt             (profileid) timedelta64[ns] ...
    temperature    (depth, profileid) float64 16.57 16.38 16.24 16.16 16.39 ...
    salinity       (depth, profileid) float64 33.42 33.41 33.43 33.43 33.43 ...
    fluorescence   (depth, profileid) float64 1.382 1.23 1.429 2.101 2.857 ...
    fl_min         (profileid) float64 0.2884 0.2508 0.2347 0.1904 0.144 ...
    fl_unbias      (depth, profileid) float64 1.355 1.203 1.403 2.074 2.83 ...
Attributes:
    title:         Matchup between Spray and Aqua Chl measurements
    date_created:  2018-06-12 17:06:36 UTC
```

In [34]:
```
# Change this to a sequence of PDFs per level, or maybe violin plots

fig = plt.figure(figsize=(8,7))
#for label, p in ds.groupby('profileid'):
#    plt.plot(p.fluorescence, ds.depth, '.', color='g', alpha=0.1)

plt.plot(ds.fluorescence, ds.depth, '.', color='g', alpha=0.1)
plt.gca().invert_yaxis()
```

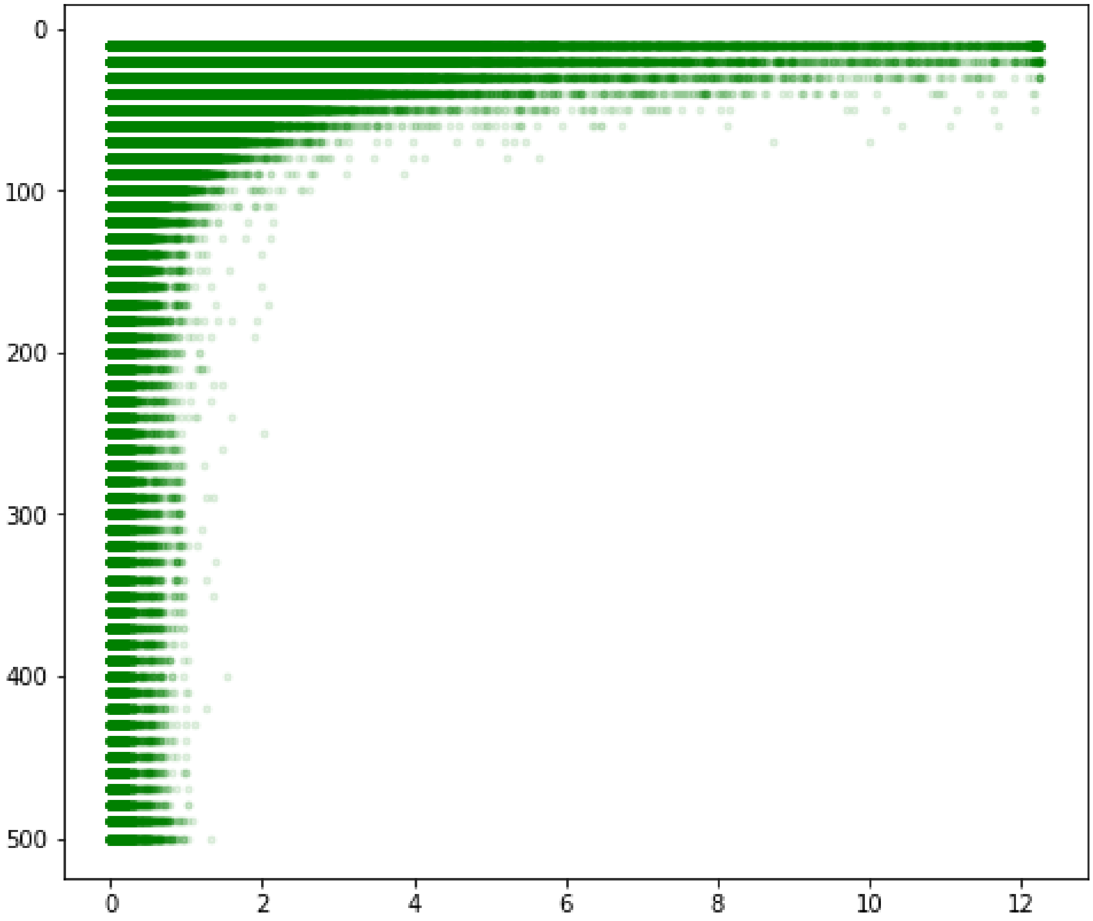


The simplest approach to correct for the offset is to assume that every profile should measure at least once a zero concentration, and a second assumption is that the sensor cannot output less than an equivalent to zero.

Let's subtract each profile by its minimum value observed.

```
In [35]: #ds['fl0'] = ds.fluorescence - ds.fl_min
```

```
In [36]: fig = plt.figure(figsize=(8,7))
         #for label, p in ds.groupby('profileid'):
         #    plt.plot(p.fl_unbias, ds.depth, '.', color='g', alpha=0.1)

         plt.plot(ds.fl_unbias, ds.depth, '.', color='g', alpha=0.1)
         plt.gca().invert_yaxis()
```



7 Gain: How to scale?

The idea here is that the Spray measurements should be coherent with the satellite reading. Let's compare with MODIS-Aqua.

In this dataset are all matchups that I found between Spray profiles and MODIS-Aqua L2. The L2 level means all calibrations and corrections were applied, but the data was not interpolated, therefore, these satellite values are the instantaneous readings from Aqua. The next alternative are daily or 8-day regularly binned, which changes the time scale of the satellite data, truncating all higher frequency. Working with L2 avoid the requirement on assumptions on timescales included in the samples.

A matchup is the nearest (in distance) pixel from the closest (in time) swath from Aqua. The limits allowed were 24hrs and 30 km difference. These are probably wider than desired, but the matchup process is a little slow so it is better to allow more matchups and filter them here before fit the corrections.

```
In [51]: ds['fl_max'] = ds.fl_unbias.max(dim='depth')
         h = plt.hist(ds.fl_max, bins=50)
         ds.fl_max.to_series().describe()
```

```
---------------------------------------------------------------------------

AttributeError                            Traceback (most recent call last)

<ipython-input-51-fe6b6ad194bc> in <module>()
----> 1 ds['fl_max'] = ds.fl_unbias.max(dim='depth')
      2 h = plt.hist(ds.fl_max, bins=50)
      3 ds.fl_max.to_series().describe()


~/.virtualenvs/fluorescence/lib/python3.6/site-packages/xarray/core/common.py in __geta
173                     return source[name]
174         raise AttributeError("%r object has no attribute %r" %
--> 175                              (type(self).__name__, name))
176
177     def __setattr__(self, name, value):


AttributeError: 'Dataset' object has no attribute 'fl_unbias'
```

```
In [50]: valid_only = ds[['fl_max', 'chlor_a']].dropna(dim='profileid', how='any')


         # the random data
         x = valid_only.fl_max
         y = valid_only.chlor_a

         nullfmt = NullFormatter()         # no labels

         # definitions for the axes
         left, width = 0.1, 0.65
         bottom, height = 0.1, 0.65
         bottom_h = left_h = left + width + 0.02

         rect_scatter = [left, bottom, width, height]
         rect_histx = [left, bottom_h, width, 0.2]
         rect_histy = [left_h, bottom, 0.2, height]

         # start with a rectangular Figure
         plt.figure(1, figsize=(8, 8))

         axScatter = plt.axes(rect_scatter)
         axHistx = plt.axes(rect_histx)
         axHisty = plt.axes(rect_histy)

         # no labels
```

```
axHistx.xaxis.set_major_formatter(nullfmt)
axHisty.yaxis.set_major_formatter(nullfmt)

# the scatter plot:
axScatter.scatter(x, y, alpha=.2)

# now determine nice limits by hand:
binwidth = 0.25
xymax = 10
lim = (int(xymax/binwidth) + 1) * binwidth

axScatter.set_xlim((-binwidth, lim))
axScatter.set_ylim((-binwidth, lim))

bins = np.arange(-binwidth, lim + binwidth, binwidth)
axHistx.hist(x, bins=bins)
axHisty.hist(y, bins=bins, orientation='horizontal')

axHistx.set_xlim(axScatter.get_xlim())
axHisty.set_ylim(axScatter.get_ylim())
```

```
    ---------------------------------------------------------------------------

    KeyError                                  Traceback (most recent call last)

    ~/.virtualenvs/fluorescence/lib/python3.6/site-packages/xarray/core/dataset.py in _copy
    761             try:
--> 762                 variables[name] = self._variables[name]
    763             except KeyError:


    KeyError: 'fl_max'


During handling of the above exception, another exception occurred:


    KeyError                                  Traceback (most recent call last)

    <ipython-input-50-cff4abced3c3> in <module>()
      1
----> 2 valid_only = ds[['fl_max', 'chlor_a']].dropna(dim='profileid', how='any')
      3
      4
      5 # the random data
    ~/.virtualenvs/fluorescence/lib/python3.6/site-packages/xarray/core/dataset.py in __ge
    873             return self._construct_dataarray(key)
    874         else:
--> 875             return self._copy_listed(np.asarray(key))
    876 
    877     def __setitem__(self, key, value):


    ~/.virtualenvs/fluorescence/lib/python3.6/site-packages/xarray/core/dataset.py in _copy
    763             except KeyError:
    764                 ref_name, var_name, var = _get_virtual_variable(
--> 765                     self._variables, name, self._level_coords, self.dims)
    766                 variables[var_name] = var
    767                 if ref_name in self._coord_names or ref_name in self.dims:


    ~/.virtualenvs/fluorescence/lib/python3.6/site-packages/xarray/core/dataset.py in _get_
     70         ref_var = dim_var.to_index_variable().get_level_variable(ref_name)
     71     else:
---> 72         ref_var = variables[ref_name]
     73 
     74     if var_name is None:


    KeyError: 'fl_max'
```

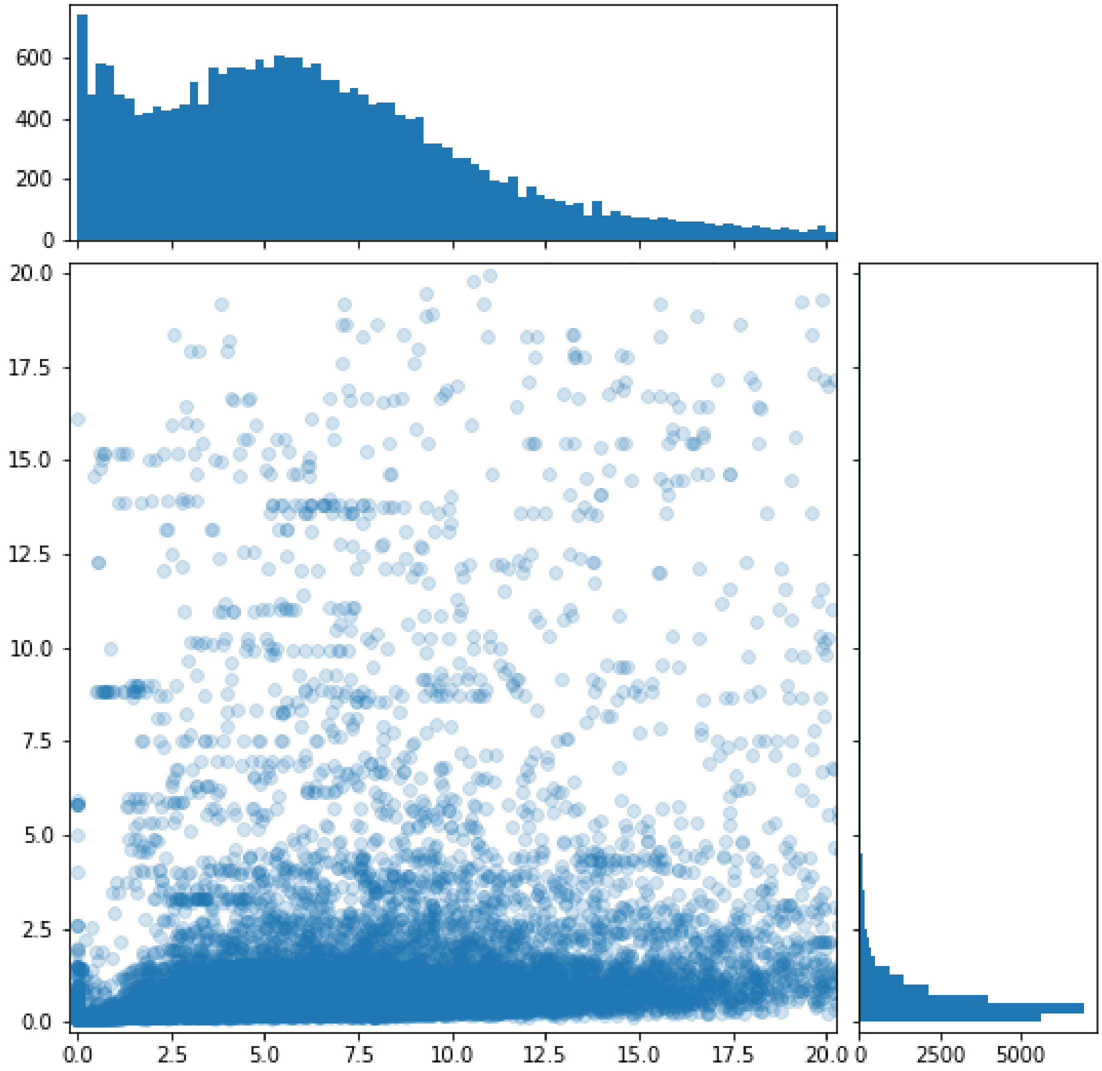
```
In [58]: ds['fl_sum'] = ds.fl_unbias.isel(depth=slice(8)).sum(axis=0)

In [332]: spray['fl_sum'] = spray.fl.isel(depth=slice(8)).sum(axis=0)

In [150]: ds['chlor_a'] = ds.chlor_a_x

In [320]: spray.sel(sat='aqua').isel(profile_id=spray.night_time)

Out[320]: <xarray.Dataset>
          Dimensions:        (depth: 100, profile_id: 47120)
          Coordinates:
            * profile_id     (profile_id) int64 20215 20216 20217 20218 20231 20233 ...
              ndive          (profile_id) int64 1 2 3 4 15 16 17 18 19 25 26 27 28 788 ...
              datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
              lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.31 34.31 ...
              lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -120.0 ...
              mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
              mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
              experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 3 3 3 3 ...
              experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
              sat            <U4 'aqua'
            * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
```

```
        Data variables:
            chl_count      (profile_id) float64 5.0 4.0 6.0 6.0 8.0 14.0 8.0 8.0 8.0 ...
            chl_mean       (profile_id) float32 1.8665981 2.0643373 1.6775862 ...
            chl_std        (profile_id) float64 0.6485 0.5478 0.8044 0.8044 0.7846 ...
            chl_sem        (profile_id) float64 0.29 0.2739 0.3284 0.3284 0.2774 ...
            temp           (depth, profile_id) float64 16.57 16.38 16.24 16.16 16.41 ...
            sal            (depth, profile_id) float64 33.42 33.41 33.43 33.43 33.43 ...
            fl             (depth, profile_id) float64 1.382 1.23 1.429 2.101 1.457 ...
            night_time     (profile_id) bool True True True True True True True True ...
```

```
In [649]: valid_only = spray.sel(sat='aqua').isel(profile_id=spray.night_time)[['fl_sum', 'chl_


          # the random data
          x = valid_only.fl_sum*0.2
          y = valid_only.chl_mean

          nullfmt = NullFormatter()         # no labels

          # definitions for the axes
          left, width = 0.1, 0.65
          bottom, height = 0.1, 0.65
          bottom_h = left_h = left + width + 0.02

          rect_scatter = [left, bottom, width, height]
          rect_histx = [left, bottom_h, width, 0.2]
          rect_histy = [left_h, bottom, 0.2, height]

          # start with a rectangular Figure
          plt.figure(1, figsize=(8, 8))

          axScatter = plt.axes(rect_scatter)
          axHistx = plt.axes(rect_histx)
          axHisty = plt.axes(rect_histy)

          # no labels
          axHistx.xaxis.set_major_formatter(nullfmt)
          axHisty.yaxis.set_major_formatter(nullfmt)

          # the scatter plot:
          axScatter.scatter(x, y, alpha=.2)

          # now determine nice limits by hand:
          binwidth = 0.25
          xymax = 20
          lim = (int(xymax/binwidth) + 1) * binwidth

          axScatter.set_xlim((-binwidth, lim))
```

```
        axScatter.set_ylim((-binwidth, lim))

        bins = np.arange(-binwidth, lim + binwidth, binwidth)
        axHistx.hist(x, bins=bins)
        axHisty.hist(y, bins=bins, orientation='horizontal')

        axHistx.set_xlim(axScatter.get_xlim())
        axHisty.set_ylim(axScatter.get_ylim())
```

```
Out[649]: (-0.25, 20.25)
```



```
In [180]: (ds.night_time & (ds.mission=='07A01401'))
```

```
Out[180]: <xarray.DataArray (profile_id: 99752)>
          array([False, False, False, ..., False, False, False])
          Coordinates:
            * profile_id  (profile_id) int64 20215 20216 20217 20218 20219 20220 20221 ...

In [661]: def fl0_mission_minima(x):
              return x - x.min()

          spray['fl_unbias'] = spray.fl.groupby('mission').apply(fl0_mission_minima)
          spray['fl_sum'] = spray.fl_unbias.isel(depth=spray.depth<=40).sum(axis=0)*10
          spray.fl_sum

Out[661]: <xarray.DataArray 'fl_sum' (profile_id: 102058)>
          array([50.331512, 52.081667, 55.266667, ..., 32.491109, 41.457637, 35.815357])
          Coordinates:
            * profile_id     (profile_id) int64 20215 20216 20217 20218 20219 20220 ...
              ndive          (profile_id) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
              datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
              lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
              lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
              mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
              mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
              experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
              experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...

In [625]: subset

Out[625]: <xarray.Dataset>
          Dimensions:        (profile_id: 210)
          Coordinates:
            * profile_id     (profile_id) int64 110842 110847 110848 110850 110852 ...
              lat            (profile_id) float64 36.87 36.88 36.87 36.87 36.87 36.87 ...
              lon            (profile_id) float64 -122.0 -122.0 -122.0 -122.0 -122.0 ...
              sat            <U4 'aqua'
              mission_id     (profile_id) int64 66 66 66 66 66 66 66 66 66 66 66 66 66 ...
              experiment     (profile_id) object 'CUGN_line_66' 'CUGN_line_66' ...
              experiment_id  (profile_id) int64 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
              mission        (profile_id) object '07A01401' '07A01401' '07A01401' ...
              datetime       (profile_id) datetime64[ns] 2007-12-03T15:28:27 ...
              ndive          (profile_id) int64 467 468 469 470 471 472 473 474 475 ...
          Data variables:
              fl_sum         (profile_id) float64 3.761 4.39 3.112 3.038 3.844 5.628 ...
              chl_mean       (profile_id) float32 3.8753846 3.8753846 3.8753846 ...

In [630]: spray.chl_mean.mean()

Out[630]: <xarray.DataArray 'chl_mean' ()>
          array(1.352428, dtype=float32)

In [631]: spray['z001'] = 34.0 * spray.chl_mean**(-0.39)
          spray['z001'] = 568.2 * spray.chl_mean**(-0.746)
          print(spray.z001.median())
          spray['z001'] = 200.0 * spray.chl_mean**(-0.293)
          print(spray.z001.median())
```

```
<xarray.DataArray 'z001' ()>
array(1009.9049, dtype=float32)
<xarray.DataArray 'z001' ()>
array(250.68805, dtype=float32)
```

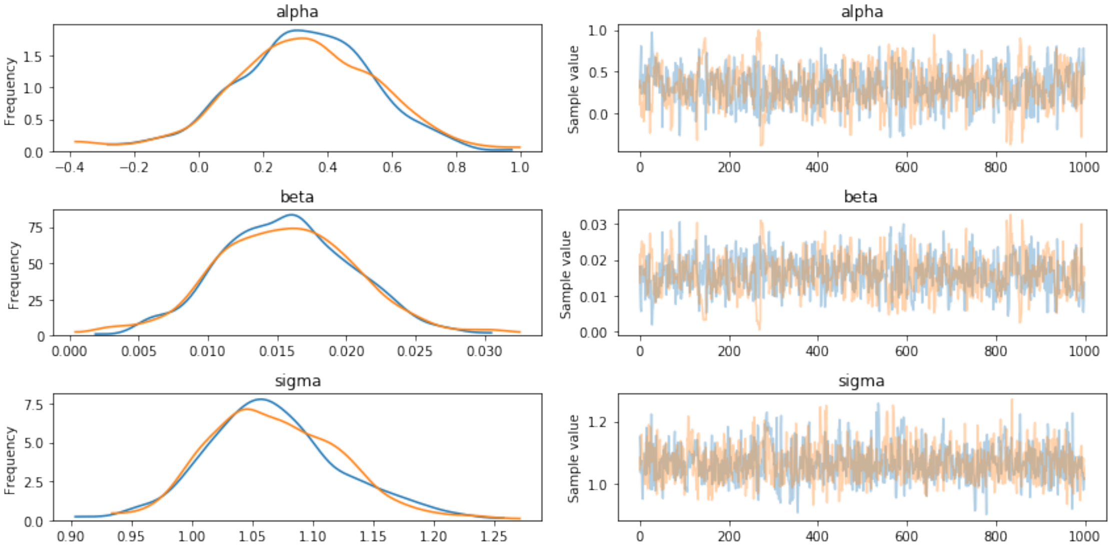
```
In [653]: subset = spray.sel(sat='aqua').isel(profile_id=(spray.night_time & (spray.mission=='
          subset
```

```
Out[653]: <xarray.Dataset>
          Dimensions:        (profile_id: 200)
          Coordinates:
            * profile_id     (profile_id) int64 110842 110847 110848 110850 110852 ...
              lat            (profile_id) float64 36.87 36.88 36.87 36.87 36.87 36.87 ...
              lon            (profile_id) float64 -122.0 -122.0 -122.0 -122.0 -122.0 ...
              sat            <U4 'aqua'
              mission_id     (profile_id) int64 66 66 66 66 66 66 66 66 66 66 66 66 66 ...
              experiment     (profile_id) object 'CUGN_line_66' 'CUGN_line_66' ...
              experiment_id  (profile_id) int64 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
              mission        (profile_id) object '07A01401' '07A01401' '07A01401' ...
              datetime       (profile_id) datetime64[ns] 2007-12-03T15:28:27 ...
              ndive          (profile_id) int64 467 468 469 470 471 472 473 474 475 ...
          Data variables:
              fl_sum         (profile_id) float64 37.61 43.9 31.12 30.38 38.44 56.28 ...
              chl_mean       (profile_id) float32 3.8753846 3.8753846 3.8753846 ...
              chl_std        (profile_id) float64 2.775 2.775 2.775 2.775 2.775 2.553 ...
```

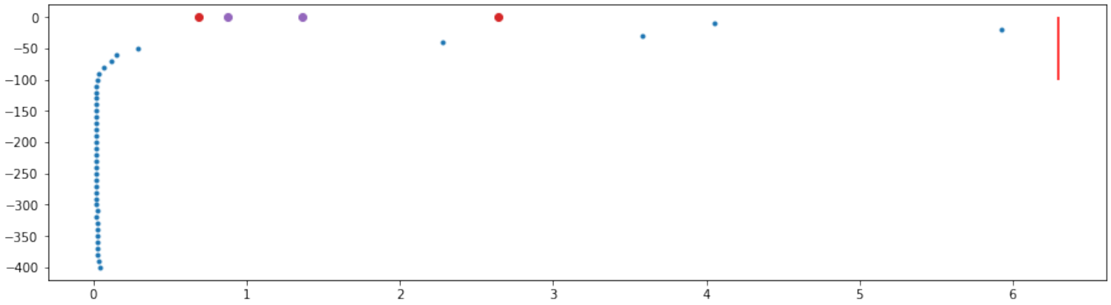
```
In [660]: subset = spray.sel(sat='aqua').isel(profile_id=(spray.night_time & (spray.mission=='
          #subset = spray.sel(sat='aqua').isel(profile_id=spray.night_time)[['fl_sum', 'chl_me


          import pymc3 as pm

          def fit_fl(fl, chl):

              basic_model = pm.Model()

              with basic_model:

                  # Priors for unknown model parameters
                  alpha = pm.Normal('alpha', mu=0, sd=10)
                  beta = pm.HalfNormal('beta', sd=2)
                  sigma = pm.HalfNormal('sigma', sd=1)

                  # Expected value of outcome
```

```
        mu = alpha + beta * fl

        # Likelihood (sampling distribution) of observations
        Y_obs = pm.Normal('Y_obs', mu=mu, sd=sigma, observed=chl)

    with basic_model:
        trace = pm.sample(1000)

    return trace

X = np.array(subset.fl_sum)
Y = np.array(subset.chl_mean)
trace = fit_fl(fl=X, chl=Y)
pm.traceplot(trace)

summary = pm.summary(trace)
print(summary)
x = np.arange(int(max(X)))
#y = summary['mean']['fl_0'] + summary['mean']['fl_scale'] * np.exp(-x / summary['me

#fig = plt.figure(figsize=(15,8))
#plt.plot(X, Y)
#plt.plot(x, y)
#plt.title("Regression model")
#plt.xlabel("Days since start of the mission")
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (2 chains in 2 jobs)
NUTS: [sigma, beta, alpha]
Sampling 2 chains: 100%|| 3000/3000 [00:04<00:00, 649.97draws/s]
The acceptance probability does not match the target. It is 0.9753766073754366, but should be


           mean        sd  mc_error   hpd_2.5  hpd_97.5       n_eff      Rhat
alpha  0.323647  0.218536  0.008551 -0.113087  0.753216  529.101812  0.999507
beta   0.015643  0.004950  0.000194  0.005173  0.024512  528.530867  0.999504
sigma  1.067801  0.055529  0.002167  0.976091  1.196334  671.643050  0.999735
```



```
In [584]: any_sat = spray.isel(profile_id=np.any(spray.chl_std, axis=1))

          pid = any_sat.profile_id[np.random.permutation(any_sat.profile_id.size)[0]]
          subset = spray.sel(profile_id=pid)

          plt.figure(figsize=(15,4))
          plt.plot(subset.fl_unbias, -subset.depth, '.')
          plt.vlines(subset.fl_unbias.isel(depth=subset.depth<=50).sum() * .39, ymin=-100, yma

          for m, s in zip(subset.chl_mean, subset.chl_std):
              plt.plot([m-s, m+s], [0, 0], 'o')
```



```
In [469]: subset.fl_unbias.isel(depth=subset.depth<=50).sum()

Out[469]: <xarray.DataArray 'fl_unbias' ()>
          array(11.3495)
          Coordinates:
              profile_id     int64 140072
              ndive          int64 845
              datetime       datetime64[ns] 2009-08-23T01:41:21
              lat            float64 34.21
              lon            float64 -120.3
              mission_id     int64 119
              mission        <U8 '09502801'
              experiment_id  int64 2
              experiment     <U12 'CUGN_line_80'
```

```
In [662]: summary['mean']['beta']
```

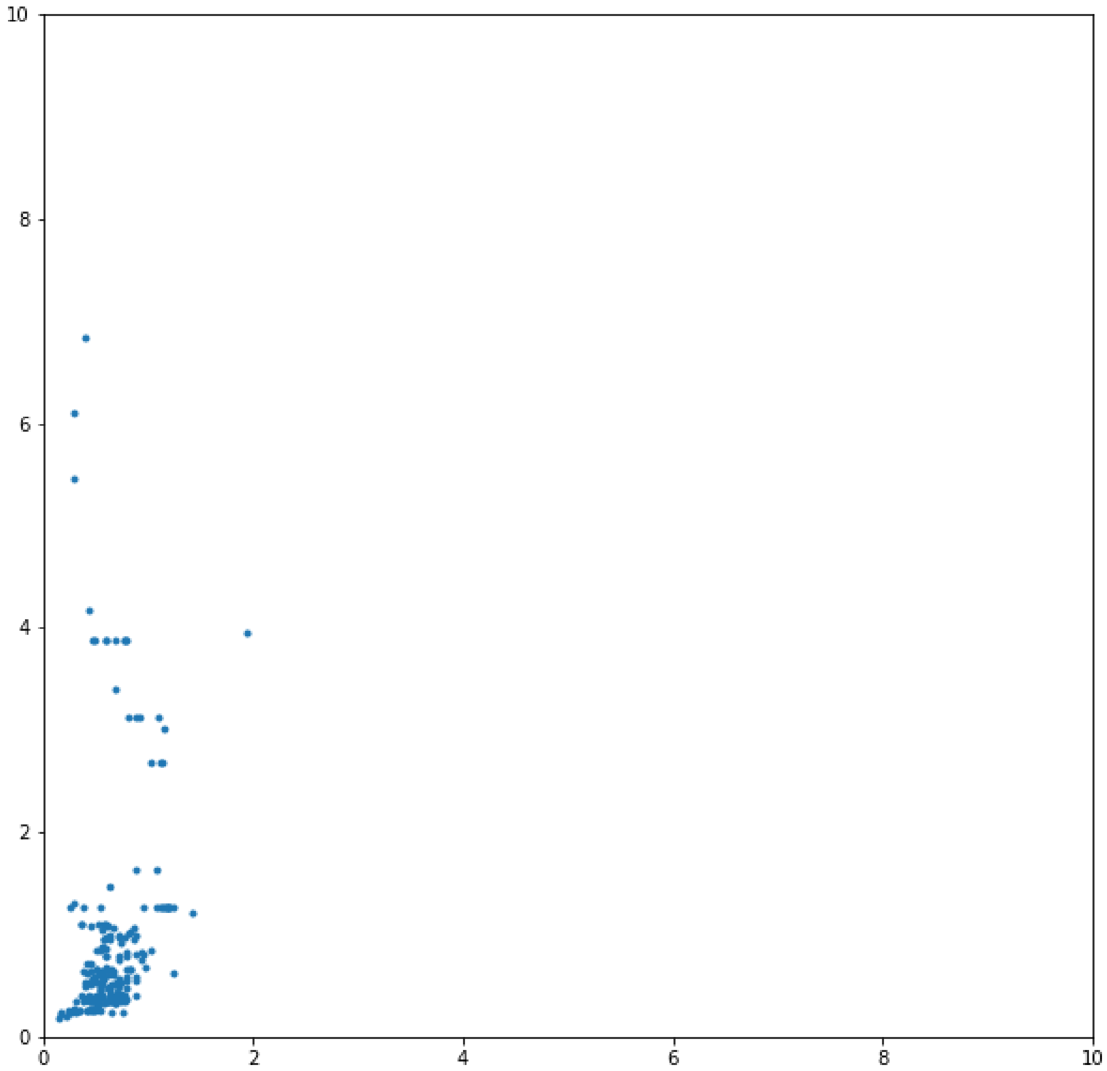
```
Out[662]: 0.0156434525399214
```

```
In [588]:
```

```
Out[588]: array([3.8753846 , 3.8753846 , 3.8753846 , 3.8753846 , 3.8753846 ,
                 3.1254363 , 3.1254363 , 2.681215  , 1.6386046 , 1.6386046 ,
                 0.65539056, 0.3286246 , 0.34801555, 0.34801555, 0.35401934,
                 0.56339556, 0.56339556, 0.61251915, 0.54973674, 0.538702  ,
                 0.5239816 , 0.5069627 , 0.4989309 , 0.5307251 , 0.5626614 ,
                 0.63853115, 0.4860134 , 0.48263657, 0.47079167, 0.45257366,
                 0.38876417, 0.25586984, 0.25586984, 0.25586984, 0.5567838 ,
                 0.5819165 , 0.5805631 , 0.5771697 , 0.5593084 , 0.5338904 ,
                 0.5338904 , 0.51026154, 0.52323174, 0.40635717, 0.39640686,
                 0.38517007, 0.3822085 , 0.40947732, 0.40575597, 0.3758512 ,
                 0.39087048, 0.50118905, 0.50268656, 0.5190279 , 0.5190279 ,
                 0.2980711 , 0.27468273, 0.27263713, 0.27263713, 0.34471518,
                 0.34415793, 0.34396166, 0.34047976, 0.26582056, 0.24629295,
                 0.23929958, 0.22926088, 0.2568648 , 0.25636858, 0.18096419,
                 0.20060277, 0.22024137, 0.23723894, 0.23723894, 0.25423652,
                 0.25423652, 0.25423652, 0.24903813, 0.24903813, 0.37369   ,
                 0.37073708, 0.3664323 , 0.3537287 , 0.32388642, 0.40417644,
                 0.40281767, 0.4008083 , 0.41896027, 0.61243033, 0.6441157 ,
                 0.6791528 , 0.87760967, 0.8563753 , 0.8563753 , 0.7929451 ,
                 0.74226815, 0.63301253, 0.6062964 , 0.63159525, 0.65343606,
                 0.7054991 , 0.7210913 , 1.2133819 , 1.272859  , 1.272859  ,
                 1.4618683 , 3.0050595 , 0.6199786 , 0.65539056, 0.8248458 ,
                 2.681215  , 2.681215  , 3.1254363 , 3.1254363 , 3.8753846 ,
                 3.8753846 , 3.8753846 , 0.3946316 , 0.39145637, 0.38584164,
                 0.3445644 , 0.34573275, 0.34362787, 0.3948835 , 0.38924333,
                 0.37906677, 0.381876  , 0.39107573, 0.972281  , 0.9789298 ,
                 0.9899734 , 0.9820631 , 1.0233064 , 1.0002398 , 0.95044756,
                 0.91439605, 0.6074949 , 0.6207756 , 0.64885277, 0.9934952 ,
                 1.0633445 , 1.0633445 , 1.0793293 , 1.272892  , 1.272892  ,
                 1.272892  , 1.272892  , 1.272892  , 1.272892  , 1.272892  ,
                 1.272892  , 1.272892  , 1.272892  , 1.272892  , 0.96607864,
                 0.957204  , 1.1056606 , 1.1056606 , 0.95470554, 1.0492114 ,
                 1.095215  , 1.095215  , 1.095215  , 1.2570319 , 1.2950699 ,
                 6.0951514 , 5.457715  , 6.844856  , 4.1697946 , 3.9446437 ,
                 3.398448  , 1.0727339 , 0.94420624, 0.8482121 , 0.8482121 ,
                 0.651719  , 0.65590316, 0.68481624, 0.8015748 , 0.8268066 ,
                 0.8400508 , 0.7988761 , 0.79382604, 0.7951439 , 0.7838157 ,
                 0.7472511 , 0.66504186, 0.6494443 , 0.64447874, 0.5406686 ,
                 0.48067975, 0.38206238, 0.38206238, 0.36944753, 0.3475887 ,
                 0.32953334, 0.32953334, 0.306349  , 0.24892847, 0.43985265,
                 0.45279798, 0.45279798, 0.40168113, 0.40168113, 0.4169231 ,
                 0.39760956, 0.36481375, 0.34676495, 0.23306437, 0.23306437],
                dtype=float32)
```

```
In [663]: fig = plt.figure(figsize=(10,10))
          plt.plot(X*summary['mean']['beta'], Y, '.')
          plt.xlim(0, 10)
          plt.ylim(0, 10)
          #plt.plot(x, y)
          #plt.title("Regression model")
          #plt.xlabel("Days since start of the mission")
```

```
Out[663]: (0, 10)
```



```
In [41]: np.max([np.max(np.fabs(x)), np.max(np.fabs(y))])
         np.fabs(x)

Out[41]: <xarray.DataArray 'fl_sum' (profileid: 39064)>
         array([5.69608  , 6.249717, 6.539196, ..., 4.58375  , 7.39725  , 9.53625 ])
         Coordinates:
             longitude      (profileid) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
           * profileid      (profileid) int64 20215 20216 20217 20218 20219 20220 ...
             experiment_id  (profileid) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
             ndive          (profileid) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
             mission_id     (profileid) int64 106 106 106 106 106 106 106 106 106 106 ...
             latitude       (profileid) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
             experiment     (profileid) object 'CUGN_line_80' 'CUGN_line_80' ...
             mission        (profileid) object '06A00501' '06A00501' '06A00501' ...
             datetime       (profileid) datetime64[ns] 2006-10-16T19:46:51 ...
```

Let's find exact localtime to define if measurement was done on daylight or night time. This will be important for the NPQ correction.

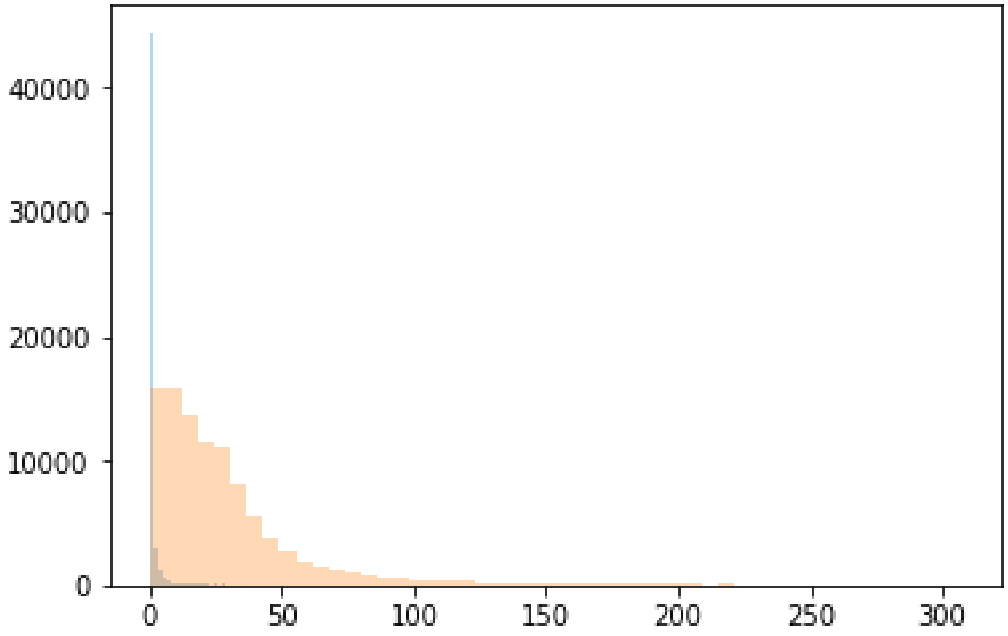
```
In [164]: assert ds.lon.max() <= 180, "I'm assuming longitudes between -180 to 180"

          tmp = ds[['datetime', 'lon']].to_dataframe()[['datetime', 'lon']]
          ds['localtime'] = tmp.apply(lambda row: row['datetime'] + timedelta(hours=row['lon']/
```

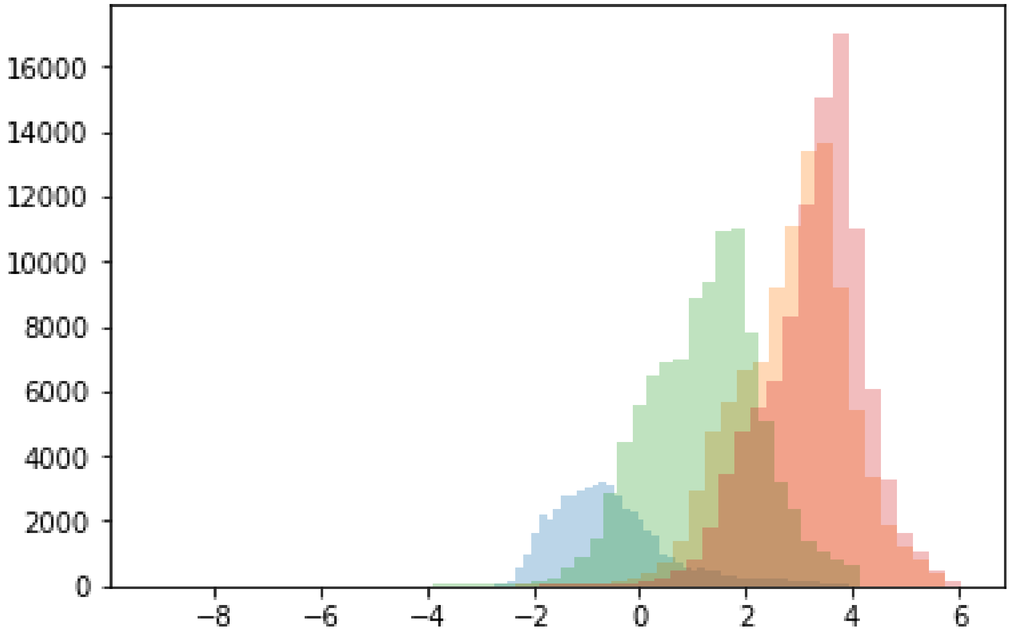
```
In [165]: ds
```

```
Out[165]: <xarray.Dataset>
          Dimensions:        (depth: 100, profile_id: 99752)
          Coordinates:
            * profile_id     (profile_id) int64 20215 20216 20217 20218 20219 20220 ...
            * depth          (depth) float64 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 ...
          Data variables:
              ndive          (profile_id) int64 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
              datetime       (profile_id) datetime64[ns] 2006-10-16T19:46:51 ...
              lat            (profile_id) float64 34.35 34.35 34.35 34.34 34.34 34.34 ...
              lon            (profile_id) float64 -119.8 -119.8 -119.8 -119.8 -119.8 ...
              mission_id     (profile_id) int64 106 106 106 106 106 106 106 106 106 ...
              mission        (profile_id) object '06A00501' '06A00501' '06A00501' ...
              experiment_id  (profile_id) int64 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...
              experiment     (profile_id) object 'CUGN_line_80' 'CUGN_line_80' ...
              dL_x           (profile_id) float64 1.066e+04 1.012e+04 9.594e+03 ...
              dt_x           (profile_id) timedelta64[ns] 00:00:00 00:00:00 00:00:00 ...
              chlor_a_x      (profile_id) float32 0.38446707 0.38446707 0.38446707 ...
              dL_y           (profile_id) float64 1.066e+04 1.012e+04 9.594e+03 ...
              dt_y           (profile_id) timedelta64[ns] 00:00:00 00:00:00 00:00:00 ...
              chlor_a_y      (profile_id) float32 0.38446707 0.38446707 0.38446707 ...
              temp           (depth, profile_id) float64 16.57 16.38 16.24 16.16 16.39 ...
              sal            (depth, profile_id) float64 33.42 33.41 33.43 33.43 33.43 ...
              fl             (depth, profile_id) float64 1.382 1.23 1.429 2.101 2.857 ...
              night_time     (profile_id) bool True True True True False False False ...
              fl_sum         (profile_id) float64 5.858 6.466 6.755 7.407 8.37 8.296 ...
              chlor_a        (profile_id) float32 0.38446707 0.38446707 0.38446707 ...
              localtime      (profile_id) datetime64[ns] 2006-10-16T11:47:42.471000 ...
```

```
In [49]: plt.hist(spray.sel(sat='aqua').chl_mean.dropna('profile_id'), bins=50, alpha=.3)
         plt.hist(spray.sel(sat='aqua').fl_sum.dropna('profile_id'), bins=50, alpha=.3)

         plt.figure()
         plt.hist(np.log(spray.sel(sat='aqua').chl_mean.dropna('profile_id')), bins=50, alpha=
         plt.hist(np.log(spray.sel(sat='aqua').fl_sum.dropna('profile_id')+1e-4), bins=50, alpl
         plt.hist(np.log(spray.sel(sat='aqua').fl_sum_alt1.dropna('profile_id')+1e-4), bins=50
         plt.hist(np.log(spray.sel(sat='aqua').fl_sum_alt2.dropna('profile_id')+1e-4), bins=50
```

```
Out[49]: (array([1.1000e+01, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
                 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
       0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
       0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
       0.0000e+00, 1.0000e+00, 0.0000e+00, 1.1000e+01, 3.4000e+01,
       3.5000e+01, 3.0000e+01, 3.1000e+01, 5.4000e+01, 4.3000e+01,
       1.1900e+02, 2.5000e+02, 4.8600e+02, 8.2000e+02, 1.8060e+03,
       3.4210e+03, 4.7290e+03, 5.4880e+03, 6.3200e+03, 8.3350e+03,
       1.1774e+04, 1.5085e+04, 1.7044e+04, 1.1036e+04, 6.0460e+03,
       3.2620e+03, 1.6520e+03, 1.0830e+03, 4.7200e+02, 1.4300e+02]),
array([-9.21034037, -8.90485902, -8.59937767, -8.29389631, -7.98841496,
       -7.6829336 , -7.37745225, -7.0719709 , -6.76648954, -6.46100819,
       -6.15552684, -5.85004548, -5.54456413, -5.23908278, -4.93360142,
       -4.62812007, -4.32263872, -4.01715736, -3.71167601, -3.40619466,
       -3.1007133 , -2.79523195, -2.4897506 , -2.18426924, -1.87878789,
       -1.57330654, -1.26782518, -0.96234383, -0.65686248, -0.35138112,
       -0.04589977,  0.25958158,  0.56506294,  0.87054429,  1.17602564,
        1.481507  ,  1.78698835,  2.09246971,  2.39795106,  2.70343241,
        3.00891377,  3.31439512,  3.61987647,  3.92535783,  4.23083918,
        4.53632053,  4.84180189,  5.14728324,  5.45276459,  5.75824595,
        6.0637273 ]),
<a list of 50 Patch objects>)
```




16000
14000
12000
10000
8000
6000
4000
2000
0
−8
−6
−4
−2
0
2
4
6