Applying XID+SED to Extreme Starbursts

In this notebook, we read in the data files and prepare them for fitting with XID+SED, the SED prior model extension to XID+. Here we focus on the source `HELP_J104631.170+591036.829`. This source is in Rowan-Robinson et al. 2017 (https://arxiv.org/abs/1704.07783) and claimed to have a star formation rate of $> 10^{10} \mathrm{M}_{\odot} \mathrm{yr}^{-1}$

Import required modules

In [1]:

```
from astropy.io import ascii, fits
from astropy.table import Table
import pylab as plt
%matplotlib inline
from astropy import wcs


import numpy as np
import xidplus
from xidplus import moc_routines
import pickle
```

Set image and catalogue filenames

In [2]:

```
#Folder containing maps

pswfits='/Users/pdh21/Work/Astro/dmu_products/dmu26/data/Lockman-SWIRE/SPIRE/Loc
kman-NEST_image_250_SMAP_v6.0.fits'#SPIRE 250 map
pmwfits='/Users/pdh21/Work/Astro/dmu_products/dmu26/data/Lockman-SWIRE/SPIRE/Loc
kman-NEST_image_350_SMAP_v6.0.fits'#SPIRE 350 map
plwfits='/Users/pdh21/Work/Astro/dmu_products/dmu26/data/Lockman-SWIRE/SPIRE/Loc
kman-NEST_image_500_SMAP_v6.0.fits'#SPIRE 500 map




#output folder
output_folder='./'
```

Load in images, noise maps, header info and WCS information

In [3]:

```
#-----250-------------
hdulist = fits.open(pswfits)
im250phdu=hdulist[0].header
im250hdu=hdulist[1].header

im250=hdulist[1].data*1.0E3 #convert to mJy
nim250=hdulist[2].data*1.0E3 #convert to mJy
w_250 = wcs.WCS(hdulist[1].header)
pixsize250=3600.0*w_250.wcs.cd[1,1] #pixel size (in arcseconds)
hdulist.close()
#-----350-------------
hdulist = fits.open(pmwfits)
im350phdu=hdulist[0].header
im350hdu=hdulist[1].header

im350=hdulist[1].data*1.0E3 #convert to mJy
nim350=hdulist[2].data*1.0E3 #convert to mJy
w_350 = wcs.WCS(hdulist[1].header)
pixsize350=3600.0*w_350.wcs.cd[1,1] #pixel size (in arcseconds)
hdulist.close()
#-----500-------------
hdulist = fits.open(plwfits)
im500phdu=hdulist[0].header
im500hdu=hdulist[1].header
im500=hdulist[1].data*1.0E3 #convert to mJy
nim500=hdulist[2].data*1.0E3 #convert to mJy
w_500 = wcs.WCS(hdulist[1].header)
pixsize500=3600.0*w_500.wcs.cd[1,1] #pixel size (in arcseconds)
hdulist.close()
```

XID+ uses Multi Order Coverage (MOC) maps for cutting down maps and catalogues so they cover the same area. It can also take in MOCs as selection functions to carry out additional cuts. Lets use the python module pymoc (http://pymoc.readthedocs.io/en/latest/) to create a MOC, centered on a specific position we are interested in. We will use a HEALPix order of 15 (the resolution: higher order means higher resolution)

Read in the source we are interested in from Rowan-Robinsons's catalogue.

In [4]:

```
ysb=Table.read('./data/MRRfiles/fromPaper.txt', format='ascii')
```

In [6]:

```
from astropy.coordinates import SkyCoord
from astropy import units as u
import os

c = SkyCoord(ra=[ysb['RA'][32]]*u.degree, dec=[ysb['dec'][32]]*u.degree)

import pymoc
moc=pymoc.util.catalog.catalog_to_moc(c,70,15)
```

Load in catalogue you want to fit (and make any cuts). Here we use HELP's VO database and directly call it using PyVO

In [7]:

```
import pyvo as vo
service = vo.dal.TAPService("http://vohedamtest.lam.fr/__system__/tap/run/tap")
```

In [8]:

```
resultset = service.search("SELECT TOP 10000 * FROM herschelhelp.main WHERE 1=CO
NTAINS(POINT('ICRS', ra, dec),CIRCLE('ICRS',"+str(c.ra.deg[0])+", "+str(c.dec.de
g[0])+", 0.025 ))")
```

```
WARNING: W27: None:2:48070: W27: COOSYS deprecated in VOTable 1.2 [a
stropy.io.votable.tree]
WARNING: W06: None:38:1318: W06: Invalid UCD 'src.redsfhit': Unknown
word 'src.redsfhit' [astropy.io.votable.tree]
WARNING: W06: None:38:132517: W06: Invalid UCD 'phot.flux;em.opt.100
-200nm': Unknown word 'em.opt.100-200nm' [astropy.io.votable.tree]
WARNING: W06: None:38:132803: W06: Invalid UCD 'stat.error;phot.flu
x;em.opt.100-200nm': Unknown word 'em.opt.100-200nm' [astropy.io.vot
able.tree]
WARNING: W06: None:38:132996: W06: Invalid UCD 'phot.flux;em.opt.100
-200nm': Unknown word 'em.opt.100-200nm' [astropy.io.votable.tree]
WARNING: W06: None:38:133351: W06: Invalid UCD 'stat.error;phot.flu
x;em.opt.100-200nm': Unknown word 'em.opt.100-200nm' [astropy.io.vot
able.tree]
WARNING: W06: None:38:133553: W06: Invalid UCD 'phot.mag;em.opt.100-
200nm': Unknown word 'em.opt.100-200nm' [astropy.io.votable.tree]
WARNING: W06: None:38:133838: W06: Invalid UCD 'stat.error;phot.mag;
em.opt.100-200nm': Unknown word 'em.opt.100-200nm' [astropy.io.votab
le.tree]
WARNING: W06: None:38:134029: W06: Invalid UCD 'phot.mag;em.opt.100-
200nm': Unknown word 'em.opt.100-200nm' [astropy.io.votable.tree]
WARNING: W06: None:38:134383: W06: Invalid UCD 'stat.error;phot.mag;
em.opt.100-200nm': Unknown word 'em.opt.100-200nm' [astropy.io.votab
le.tree]
WARNING: W06: None:38:134582: W06: Invalid UCD 'meta.code.qual;em.op
t.100-200nm': Unknown word 'em.opt.100-200nm' (suppressing further w
arnings of this type...) [astropy.io.votable.tree]
```

In [9]:

```
masterlist=resultset.table
```

In [10]:

```
masterlist
```

Out[10]:

<Table masked=True length=612>

field	help_id	ra	dec
		deg	deg
object	object	float64	float64
Lockman SWIRE	HELP_J104638.221+591148.656	161.65925293770201	59.196848862853301
Lockman SWIRE	HELP_J104635.749+591159.797	161.64895354004301	59.199943686663303
Lockman SWIRE	HELP_J104635.673+591151.488	161.64863897318801	59.197635549980497
Lockman SWIRE	HELP_J104636.716+591153.499	161.65298183626501	59.1981942725632
Lockman SWIRE	HELP_J104636.549+591152.862	161.65228846237	59.198017308322697
Lockman SWIRE	HELP_J104636.329+591150.217	161.65136947320499	59.197282393400002
Lockman SWIRE	HELP_J104637.753+591146.805	161.65730338938499	59.196334773275403
Lockman SWIRE	HELP_J104636.653+591148.348	161.65272231304399	59.196763373943398
Lockman SWIRE	HELP_J104636.590+591146.670	161.65245722956101	59.196297147097198
...	...	...	...
Lockman SWIRE	HELP_J104626.750+590919.397	161.61145906657001	59.155388006727698
Lockman SWIRE	HELP_J104629.666+590909.147	161.623606773419	59.152540885587101
Lockman SWIRE	HELP_J104630.125+590908.544	161.625521722077	59.152373370401598
Lockman SWIRE	HELP_J104623.180+590943.550	161.59658256820299	59.162097269620702
Lockman SWIRE	HELP_J104622.434+590944.268	161.59347306076501	59.162296694316701
Lockman SWIRE	HELP_J104623.154+590939.857	161.59647342365699	59.1610713127777
Lockman SWIRE	HELP_J104623.619+590935.614	161.59841323013299	59.159892872450797
Lockman SWIRE	HELP_J104623.136+590934.412	161.59639803626499	59.159558822563199

field	help_id	ra	dec
Lockman SWIRE	HELP_J104623.326+590931.630	161.59719156002399	59.158786015749797
Lockman SWIRE	HELP_J104623.554+590930.911	161.59814096002401	59.158586515749803

In [11]:

```
for i in range(0,len(masterlist)):
    masterlist['help_id'][i]=masterlist['help_id'][i].decode('UTF-8')
masterlist['help_id']=masterlist['help_id'].astype(str)
```

Filter table

We will only use objects that are not stars (i.e. have `flag_gaia` = 3)

In [12]:

```
prior_list=masterlist[(masterlist['flag_gaia']!=3)]
```

Get Redshift and Uncertianty

As there is an issue with `help_id`s lets add a new column and fill

In [13]:

```
from astropy.table import Column, MaskedColumn

t = Table([[1, 2]], names=['a'])
b = MaskedColumn([3, 4], mask=[True, False])
t['b'] = b
```

```
INFO: Upgrading Table to masked Table. Use Table.filled() to convert
to unmasked table. [astropy.table.table]
```

In [15]:

```
from astropy.table import Column
prior_list['redshift']=MaskedColumn(np.full((len(prior_list)),fill_value=3.0),ma
sk=[True]*len(prior_list))
prior_list.add_column(MaskedColumn(np.full((len(prior_list)),fill_value=3.0),mas
k=[True]*len(prior_list),name='redshift_unc'))
```

Ken Duncan defines a median and a hierarchical bayes combination redshift. We need uncertianty so lets match via `help_id`

In [16]:

```
photoz=Table.read("/Users/pdh21/Work/Astro/dmu_products/dmu24/dmu24_Lockman-SWIR
E/data/master_catalogue_Lockman-SWIRE_20170710_photoz_20170802_r_and_irac1_optim
ised_UPDATED_IDs_20180219.fits")
```

In [17]:

```
help_id=np.empty((len(photoz)),dtype=np.dtype('U27'))
for i in range(0,len(photoz)):
    help_id[i]=photoz['help_id'][i].strip()
photoz['help_id']=help_id
```

In [18]:

```
from astropy.table import Column
ii=0
for i in range(0,len(prior_list)):
    ind=photoz['help_id'] == prior_list['help_id'][i]
    try:
        prior_list['redshift'][i]=photoz['z1_median'][ind]
        prior_list['redshift_unc'][i]=np.max(np.array([np.abs(photoz['z1_median'
][ind]-photoz['z1_min'][ind]),np.abs(photoz['z1_max'][ind]-photoz['z1_median'][i
nd])]))
        prior_list['redshift_unc'].mask[i]=False
        prior_list['redshift'].mask[i]=False

    except ValueError:
```

1
2
3
4
5
6
7
8
9
10
11
13
14
15
16
18
19
20
21
23
25
26
27
28
29
31
32
33
35
37
38
39
40
41
42
44
45
46
47
48
49
50
52
53
54
55
56
57
58
59
60
63
64
65
66
67
68
69
70
72
75

79
84
85
86
87
88
91
92
95
98
99
102
103
104
105
107
108
109
110
111
112
113
114
115
116
118
120
121
123
124
125
126
127
128
129
130
131
132
135
136
138
139
141
142
143
144
145
147
148
150
152
153
154
155
156
158
159
160
161
162
163

166
167
170
172
175
176
177
178
180
182
185
187
189
190
191
193
194
196
198
201
202
204
205
207
208
209
210
212
214
215
216
217
219
220
221
224
225
226
227
229
230
231
233
234
235
236
237
239
242
246
247
249
251
252
253
255
256
257
258
259
261

```
262
263
264
265
266
267
269
274
276
278
280
282
283
284
286
287
288
289
290
291
295
296
299
302
303
305
306
309
311
312
314
315
316
317
318
319
320
322
324
325
326
328
331
336
337
339
340
341
342
343
344
345
346
349
351
352
353
354
355
356
357
```

```
359
360
361
362
365
366
368
369
370
371
372
373
374
376
377
378
379
381
382
386
387
389
390
392
393
394
395
396
397
398
399
400
401
402
404
405
407
411
412
414
415
417
418
419
420
421
422
423
424
427
430
431
432
433
435
436
438
440
443
444
448
```

449
450
454
456
457
459
461
462
466
469
470
471
472
473
475
476
477
478
479
480
481
483
485
487
494
498
499
501
506
508
510
511
512
513
514
515
516
517
519
523
524
526
528
530
532
533
534
538
541
542
543
544
548
549
550
551
553
555
556
557
562

```
563
564
565
567
568
571
572
573
574
575
578
580
582
583
586
587
588
590
594
595
596
597
598
599
600
601
```

In [19]:

```
prior_list[prior_list['redshift'].mask == False]
```

Out[19]:

<Table masked=True length=212>

field	help_id	ra	dec
		deg	deg
object	str27	float64	float64
Lockman SWIRE	HELP_J104638.221+591148.656	161.65925293770201	59.196848862853301
Lockman SWIRE	HELP_J104633.388+591159.665	161.63911810007801	59.199906823277999
Lockman SWIRE	HELP_J104630.543+591203.244	161.62726356002401	59.200901215749802
Lockman SWIRE	HELP_J104631.224+591152.487	161.63009821670201	59.1979130228532
Lockman SWIRE	HELP_J104629.196+591154.383	161.621651153702	59.198439621853197
Lockman SWIRE	HELP_J104628.970+591150.623	161.62070706002501	59.1973952157498
Lockman SWIRE	HELP_J104633.756+591145.612	161.64065098326699	59.196003338779803
Lockman SWIRE	HELP_J104634.716+591142.000	161.64465066002401	59.1950001157497
Lockman SWIRE	HELP_J104632.701+591143.101	161.63625211049899	59.195305701881203
...	...	...	...
Lockman SWIRE	HELP_J104631.265+590915.262	161.63027066002499	59.1542394157497
Lockman SWIRE	HELP_J104635.251+590913.191	161.64688034249599	59.153664048738101
Lockman SWIRE	HELP_J104628.984+590922.830	161.62076751470201	59.1563415608532
Lockman SWIRE	HELP_J104628.783+590920.883	161.619931050703	59.155800854853197
Lockman SWIRE	HELP_J104628.294+590913.733	161.61789288070301	59.153814649853302
Lockman SWIRE	HELP_J104626.672+590923.765	161.61113516002499	59.156601515749699
Lockman SWIRE	HELP_J104626.064+590921.433	161.608600060024	59.1559535157498
Lockman SWIRE	HELP_J104626.000+590920.911	161.608333960024	59.155808515749797

field	help_id	ra	dec
Lockman SWIRE	HELP_J104623.326+590931.630	161.59719156002399	59.158786015749797
Lockman SWIRE	HELP_J104623.554+590930.911	161.59814096002401	59.158586515749803

Add additional source

The fit appears bad and requires another source. There appears to be a suitable one in masterlist so lets add it with redshift 3 and uncertianty 3.

In [20]:

```
z_sig_new=3
prior_list['redshift'].mask[prior_list['help_id']=='HELP_J104631.170+591036.829'
]=False
prior_list['redshift_unc'].mask[prior_list['help_id']=='HELP_J104631.170+591036.
829']=False
prior_list=prior_list[prior_list['redshift'].mask == False]
```

Remove HELP_J104631.082+591021.253

As a comparison model, lets remove `HELP_J104631.082+591021.253` such that the flux is only going to HELP_J104631.170+591036.829. This will indicate if this is feasible, given the data.

In [21]:

```
prior_list['redshift'].mask[prior_list['help_id']=='HELP_J104631.082+591021.253'
]=True
prior_list=prior_list[prior_list['redshift'].mask == False]
```

XID+ is built around two python classes. A prior and posterior class. There should be a prior class for each map being fitted. It is initiated with a map, noise map, primary header and map header and can be set with a MOC. It also requires an input prior catalogue and point spread function.

In [22]:

```
#---prior250--------
prior250=xidplus.prior(im250,nim250,im250phdu,im250hdu, moc=moc)#Initialise with
 map, uncertianty map, wcs info and primary header
prior250.prior_cat(prior_list['ra'],prior_list['dec'],'photoz', z_median=prior_l
ist['redshift']._get_data(), z_sig=prior_list['redshift_unc']._get_data(),ID=pri
or_list['help_id'])#Set input catalogue
prior250.prior_bkg(-5.0,5)#Set prior on background (assumes Gaussian pdf with mu
 and sigma)
#---prior350--------
prior350=xidplus.prior(im350,nim350,im350phdu,im350hdu, moc=moc)
prior350.prior_cat(prior_list['ra'],prior_list['dec'],'photoz', z_median=prior_l
ist['redshift']._get_data(), z_sig=prior_list['redshift_unc']._get_data(),ID=pri
or_list['help_id'])
prior350.prior_bkg(-5.0,5)

#---prior500--------
prior500=xidplus.prior(im500,nim500,im500phdu,im500hdu, moc=moc)
prior500.prior_cat(prior_list['ra'],prior_list['dec'],'photoz', z_median=prior_l
ist['redshift']._get_data(), z_sig=prior_list['redshift_unc']._get_data(),ID=pri
or_list['help_id'])
prior500.prior_bkg(-5.0,5)
```

Set PSF. For SPIRE, the PSF can be assumed to be Gaussian with a FWHM of 18.15, 25.15, 36.3 '' for 250, 350 and 500 μm respectively. Lets use the astropy module to construct a Gaussian PSF and assign it to the three XID+ prior classes.

In [23]:

```
#pixsize array (size of pixels in arcseconds)
pixsize=np.array([pixsize250,pixsize350,pixsize500])
#point response function for the three bands
prfsize=np.array([18.15,25.15,36.3])
#use Gaussian2DKernel to create prf (requires stddev rather than fwhm hence pfwh
m/2.355)
from astropy.convolution import Gaussian2DKernel

##---------fit using Gaussian beam-----------------------
prf250=Gaussian2DKernel(prfsize[0]/2.355,x_size=101,y_size=101)
prf250.normalize(mode='peak')
prf350=Gaussian2DKernel(prfsize[1]/2.355,x_size=101,y_size=101)
prf350.normalize(mode='peak')
prf500=Gaussian2DKernel(prfsize[2]/2.355,x_size=101,y_size=101)
prf500.normalize(mode='peak')

pind250=np.arange(0,101,1)*1.0/pixsize[0] #get 250 scale in terms of pixel scale
 of map
pind350=np.arange(0,101,1)*1.0/pixsize[1] #get 350 scale in terms of pixel scale
 of map
pind500=np.arange(0,101,1)*1.0/pixsize[2] #get 500 scale in terms of pixel scale
 of map

prior250.set_prf(prf250.array,pind250,pind250)#requires psf as 2d grid, and x an
d y bins for grid (in pixel scale)
prior350.set_prf(prf350.array,pind350,pind350)
prior500.set_prf(prf500.array,pind500,pind500)
```

In [24]:

```
print('fitting '+ str(prior250.nsrc)+' sources \n')
print('using ' +  str(prior250.snpix)+', '+ str(prior350.snpix)+' and '+ str(pri
or500.snpix)+' pixels')
```

```
fitting 122 sources

using 427, 219 and 108 pixels
```

Before fitting, the prior classes need to take the PSF and calculate how muich each source contributes to each pixel. This process provides what we call a pointing matrix. Lets calculate the pointing matrix for each prior class

In [25]:

```
prior250.get_pointing_matrix()
prior350.get_pointing_matrix()
prior500.get_pointing_matrix()
```

Default prior on flux is a uniform distribution, with a minimum and maximum of 0.00 and 1000.0 mJy respectively for each source. running the function upper_lim _map resets the upper limit to the maximum flux value (plus a 5 sigma Background value) found in the map in which the source makes a contribution to.

In [26]:

```
pacs100='/Users/pdh21/Work/Astro/dmu_products/dmu18/dmu18_HELP-PACS-maps/data/Lo
ckman-SWIRE_PACS100_v0.9.fits'
#PACS 100 map
pacs160='/Users/pdh21/Work/Astro/dmu_products/dmu18/dmu18_HELP-PACS-maps/data/Lo
ckman-SWIRE_PACS160_v0.9.fits'#PACS 160 map
```

In [27]:

```
#-----100-------------
hdulist = fits.open(pacs100)
im100phdu=hdulist[1].header
im100hdu=hdulist[1].header
im100=hdulist[1].data
w_100 = wcs.WCS(hdulist[1].header)
pixsize100=3600.0*np.abs(hdulist[1].header['CDELT1']) #pixel size (in arcsecond
s)
nim100=hdulist[2].data
hdulist.close()

#-----160-------------
hdulist = fits.open(pacs160)
im160phdu=hdulist[1].header
im160hdu=hdulist[1].header

im160=hdulist[1].data #convert to mJy
w_160 = wcs.WCS(hdulist[1].header)
pixsize160=3600.0*np.abs(hdulist[1].header['CDELT1']) #pixel size (in arcsecond
s)
nim160=hdulist[2].data
hdulist.close()
```

In [28]:

```
#---prior100--------
prior100=xidplus.prior(im100,nim100,im100phdu,im100hdu,moc=moc)#Initialise with
 map, uncertianty map, wcs info and primary header
#prior100.prior_cat(mips24['INRA'],mips24['INDEC'],'photoz')#Set input catalogue

prior100.prior_cat(prior_list['ra'],prior_list['dec'],'photoz', z_median=prior_l
ist['redshift']._get_data(), z_sig=prior_list['redshift_unc']._get_data(),ID=pri
or_list['help_id'])#Set input catalogue
prior100.prior_bkg(0,1)#Set prior on background

#---prior160--------
prior160=xidplus.prior(im160,nim160,im160phdu,im160hdu,moc=moc)
prior160.prior_cat(prior_list['ra'],prior_list['dec'],'photoz', z_median=prior_l
ist['redshift']._get_data(), z_sig=prior_list['redshift_unc']._get_data(),ID=pri
or_list['help_id'])
#prior160.prior_cat(mips24['INRA'],mips24['INDEC'],'photoz')

prior160.prior_bkg(0,1)
```

In [29]:

```
pacs100_psf=fits.open('/Users/pdh21/Work/Astro/dmu_products/dmu18/dmu18_Lockman-
SWIRE/dmu18_PACS_100_PSF_Lockman-SWIRE_20171214.fits')
pacs160_psf=fits.open('/Users/pdh21/Work/Astro/dmu_products/dmu18/dmu18_Lockman-
SWIRE/dmu18_PACS_160_PSF_Lockman-SWIRE_20171214.fits')

centre100=np.long((pacs100_psf[1].header['NAXIS1']-1)/2)
radius100=15
centre160=np.long((pacs160_psf[1].header['NAXIS1']-1)/2)
radius160=25

pind100=np.arange(0,radius100+1+radius100,1)*3600*np.abs(pacs100_psf[1].header[
'CDELT1'])/pixsize100 #get 100 scale in terms of pixel scale of map
pind160=np.arange(0,radius160+1+radius160,1)*3600*np.abs(pacs160_psf[1].header[
'CDELT1'])/pixsize160 #get 160 scale in terms of pixel scale of map

# Divide by 1000 so that units are mJy
prior100.set_prf(pacs100_psf[1].data[centre100-radius100:centre100+radius100+1,c
entre100-radius100:centre100+radius100+1]/1000.0,
                 pind100,pind100)
prior160.set_prf(pacs160_psf[1].data[centre160-radius160:centre160+radius160+1,c
entre160-radius160:centre160+radius160+1]/1000.0,
                 pind160,pind160)
```

In [30]:

```
mipsfits='/Users/pdh21/Work/Astro/dmu_products/dmu26/data/Lockman-SWIRE/MIPS/wp4
_lockman-swire_mips24_map_v1.0.fits.gz'
#-----24-------------
hdulist = fits.open(mipsfits)
im24phdu=hdulist[0].header
im24hdu=hdulist[1].header

im24=hdulist[1].data
nim24=hdulist[2].data
w_24 = wcs.WCS(hdulist[1].header)
pixsize24=3600.0*w_24.wcs.cdelt[1] #pixel size (in arcseconds)
hdulist.close()




# Point response information, at the moment its 2D Gaussian,

#pixsize array (size of pixels in arcseconds)
pixsize=np.array([pixsize24])
#point response function for the three bands


#Set prior classes
#---prior24--------
prior24=xidplus.prior(im24,nim24,im24phdu,im24hdu,moc=moc)#Initialise with map,
 uncertianty map, wcs info and primary header
prior24.prior_cat(prior_list['ra'],prior_list['dec'],'photoz', z_median=prior_li
st['redshift']._get_data(), z_sig=prior_list['redshift_unc']._get_data(),ID=prio
r_list['help_id'])#Set input catalogue
prior24.prior_bkg(0,2)#Set prior on background
```

In [31]:

```
##---------fit using seb's empiricall beam-----------------------
#-----24-------------
hdulist = fits.open('/Users/pdh21/Work/Astro/dmu_products/dmu17/dmu17_Lockman-SW
IRE/data/dmu17_MIPS_Lockman-SWIRE_20171122.fits')
prf24=hdulist[1].data
centre=np.long((hdulist[1].header['NAXIS1']-1)/2)
radius=20
hdulist.close()


import scipy.ndimage
prior24.set_prf(prf24[centre-radius:centre+radius+1,centre-radius:centre+radius+
1]/1.0E3,np.arange(0,41/2.0,0.5),np.arange(0,41/2.0,0.5))
```

In [32]:

```
prior100.get_pointing_matrix()
prior160.get_pointing_matrix()
prior24.get_pointing_matrix()
```

In [33]:

```
priors=[prior24,prior100,prior160,prior250,prior350,prior500]
```

In [34]:

```
xidplus.save([prior24,prior100,prior160,prior250,prior350,prior500],'None','./da
ta/priors/prior_XID+IR_SED_ESB_32_added_source_flag_gaia_removed_source')
```