Jacopo Fritz, IRyA - UNAM

DataCubes in Astrophysics

Lesson 5

PACS source of error

1. Flux calibration uncertainty: 5%;
2. instrumental and mapping uncertainty: use the errormap (see the structure of the fits file);
3. uncertainties on the sky (background measurement): includes both correlated and uncorrelated noise.

- All errors must be summed… …how?

PACS source of error: implementation

1. 5% of total flux = …?
2. use the errormap and sum all the pixel's errors (do it PROPERLY!!!);
3. background uncertainty:
    a. correlated error (stdev of the values from ALL the sky regions, multiply by the number of sky regions),
    b. uncorrelated error (stdev of the values from ALL the sky pixels, multiply by the number of sky pixels);

SPIRE flux and errors

- Follow the very same procedure as for PACS, but check the units;
- errors are from: flux calibration (7%), instrumental noise, confusion noise, background (instrumental, confusion and large-scale).

SPIRE flux and errors

- confusion noise depends on the beam;

```
cnoise[wind]*np.sqrt(on_s_pix*pixsz**2/beam[wind])
```

- instrumental background, calculated as error on the background, normalized by the number of pixels;
- confusion background, calculated as the confusion error on the background, normalized by the number of pixels;

```
cnoise[wind]*np.sqrt(pixsz**2/beam[wind])*on_s_pix/np.sqrt(pix_b)
```

- large-scale background, calculated the stddev of the background flux on the background regions.

Errors evaluation (PACS)

- Calibration uncertainties;
- Instrumental and map-making uncertainty;
- Sky (background) evaluation uncertainty;

1. Correlated noise;
2. Uncorrelated noise.

Errors evaluation (PACS)

Calibration uncertainty

- Depends on the instrument calibration;
- directly proportional to the measured flux;
- estimated to be **5**%.

Errors evaluation (PACS)

Instrumental and map-making uncertainty

Same mask used to calculate the source's flux!

```
errsq = 0.0

for i in range(nx):

    for n in range(ny):

        if smask[n,i] == True:

            errsq = errsq + errmap[n,i]**2

err2 = numpy.sqrt(errsq)
```

Error (stdv) map, found in one of the other plane o the data

Errors evaluation (PACS)

Un-correlated sky noise

It is the **squared standard deviation** of the flux values of the **single pixels** within the **regions** chosen to measure the average **background flux level**, multiplied by the **number of the pixels within the source** (and then root-squared).

- While calculating the average sky flux per pixel on all the regions, memorize the values of the pixels in an array (`append`);
- while calculating the source's flux, count the pixels within it;
- the standard deviation is simply calculated with a `numpy` function.

Errors evaluation (PACS)

Correlated sky noise

It is the **squared product** between the **standard deviation** of the average flux values per pixel of the **single regions** chosen to measure the average **background flux level**, multiplied by the **number of the pixels within the source** (and then root-squared)

- While calculating the average sky flux per pixel on all the regions, memorize the values of the total flux in each region;
- while doing so also calculate the number of pixels within each region;
- then, calculate the average sky per pixel value for each region.

Errors evaluation (SPIRE)

- Calibration uncertainties;
- Instrumental and map-making uncertainty;
- Confusion noise;
- Sky (background) evaluation uncertainty;

1. Instrumental background;
2. Confusion background;
3. Large-scale background.

Errors evaluation (SPIRE)

Calibration uncertainty

- Depends on the instrument calibration;
- directly proportional to the measured flux;
- estimated to be 7%.

Errors evaluation (SPIRE)

Instrumental and map-making uncertainty

Same mask used to calculate the source's flux!

```
errsq = 0.0

for i in range(nx):

    for n in range(ny):

        if smask[n,i] == True:

            errsq = errsq + errmap[n,i]**2

err2 = numpy.sqrt(errsq)
```

Error (stdv) map, found in one of the other plane o the data

Errors evaluation (SPIRE)

Confusion noise

It is the **square root** of the product between number of pixels on the source and the area of the pixel, divided by the instrument beam. This is multiplied by the typical confusion noise (as from Nguyen+ 2010)

```
c_noise = [0.0058,0.0063,0.0068]   Jy/beam

beam = [465, 822, 1769]  arcsec²
```

Errors evaluation (SPIRE)

Instrumental background

Is the sum of the pixels’ errormap values in the regions where the background is calculated, multiplied by the number of the pixels on-source, and divided by the number of pixels within ALL of the regions.

Note: the values of the errormap for each pixel must be summed in quadrature, while the error calculation is made using the square root.

Errors evaluation (SPIRE)

Confusion background

Easier written than described:

```
c_bck = c_noise[lamb] * np.sqrt(pixsz[lamb]**2/beam[lamb]) *
pix_on_source/np.sqrt(pix_on_bck_regions)
```

Errors evaluation (SPIRE)

Large-scale background

It is the standard deviation of the sky values calculated in the apertures, divided by the square root of the number of apertures.