

1.1_LegacySurvey

March 8, 2018

1 XMM-13hr master catalogue

1.1 Preparation of Legacy Survey data

The catalogue comes from `dmu0_LegacySurvey`.

In the catalogue, we keep:

- The identifier (it's unique in the catalogue);
- The position;
- The stellarity;
- The aperture fluxes. Are these aperture corrected?
- The kron magnitude to be used as total magnitude (no "auto" magnitude is provided).

We don't know when the maps have been observed. We will use the year of the reference paper.

This notebook was run with `herschelhelp_internal` version:
0246c5d (Thu Jan 25 17:01:47 2018 +0000) [with local modifications]

This notebook was executed on:
2018-02-21 14:19:35.176768

WARNING: UnitsWarning: '1/deg^2' did not parse as fits unit: Numeric factor not supported by FITS

WARNING: UnitsWarning: 'nanomaggy' did not parse as fits unit: At col 0, Unit 'nanomaggy' not supported

WARNING: UnitsWarning: '1/nanomaggy^2' did not parse as fits unit: Numeric factor not supported

WARNING: UnitsWarning: '1/arcsec^2' did not parse as fits unit: Numeric factor not supported by FITS

1.2 I - Aperture correction

To compute aperture correction we need to determine two parameters: the target aperture and the range of magnitudes for the stars that will be used to compute the correction.

Target aperture: To determine the target aperture, we simulate a curve of growth using the provided apertures and draw two figures:

- The evolution of the magnitudes of the objects by plotting on the same plot aperture number vs the mean magnitude.

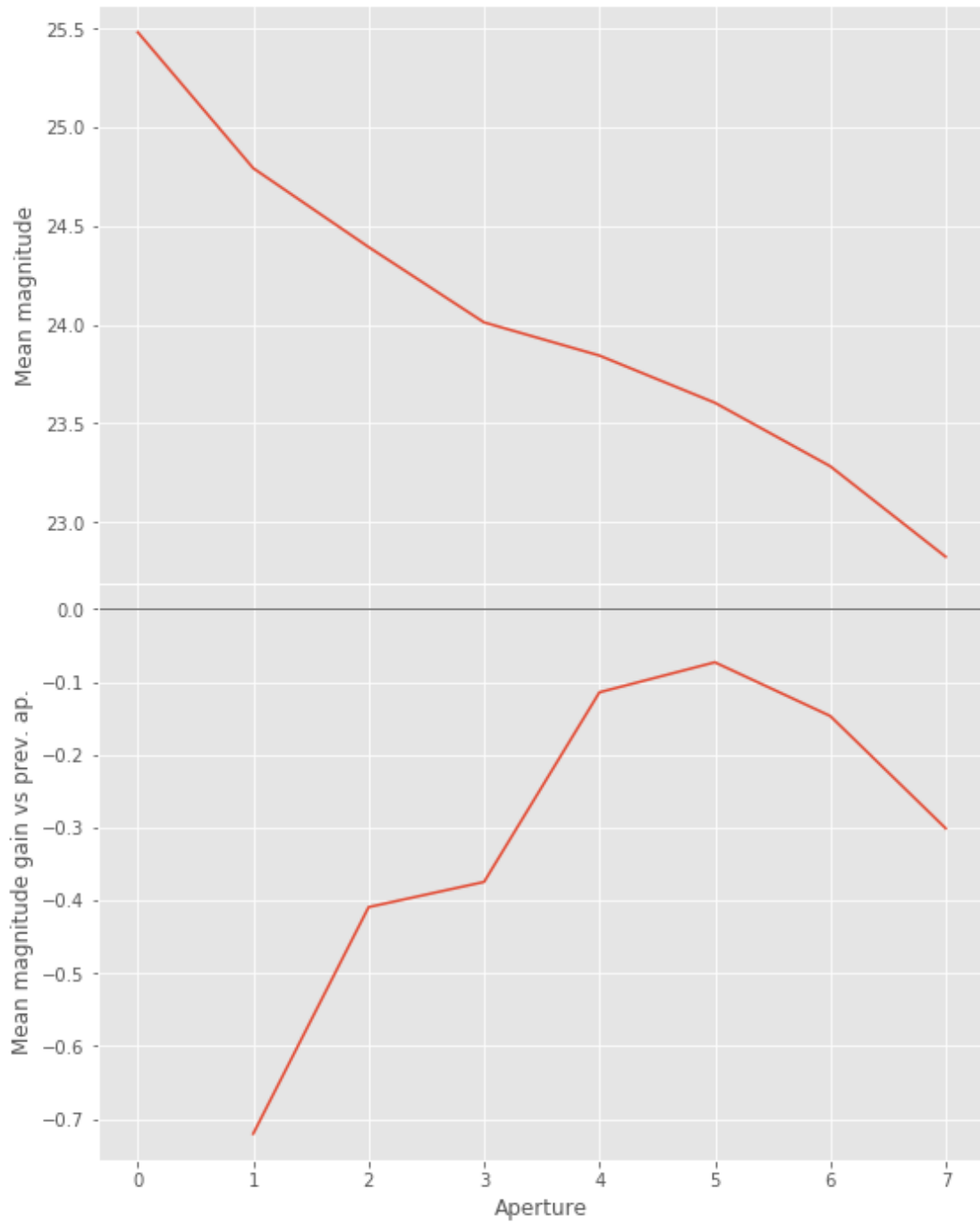
- The mean gain (loss when negative) of magnitude is each aperture compared to the previous (except for the first of course).

As target aperture, we should use the smallest (i.e. less noisy) aperture for which most of the flux is captured.

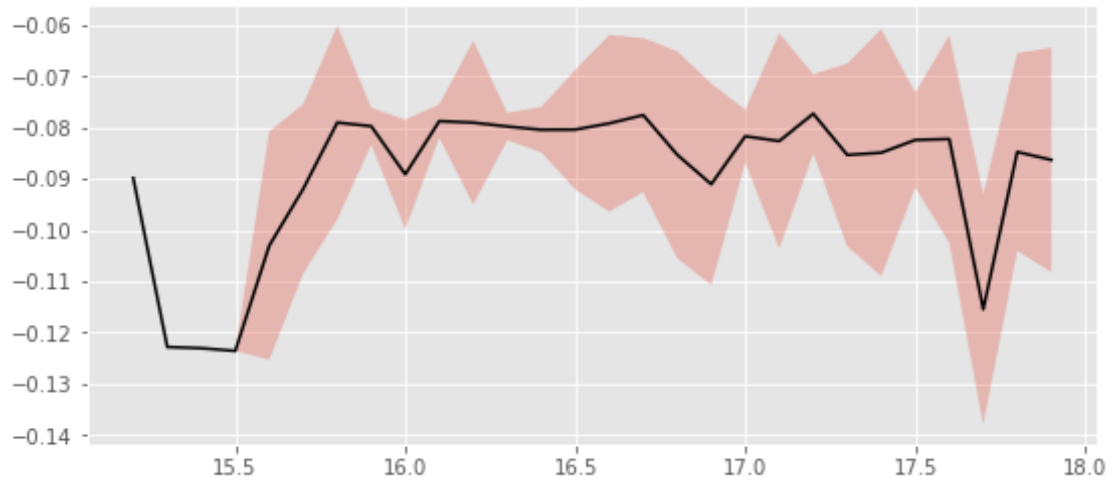
Magnitude range: To know what limits in aperture to use when doing the aperture correction, we plot for each magnitude bin the correction that is computed and its RMS. We should then use the wide limits (to use more stars) where the correction is stable and with few dispersion.

```
/opt/herschelhelp_internal/herschelhelp_internal/utils.py:76: RuntimeWarning: invalid value encountered in divide
magnitudes = 2.5 * (23 - np.log10(fluxes)) - 48.6
/opt/herschelhelp_internal/herschelhelp_internal/utils.py:76: RuntimeWarning: divide by zero encountered in divide
magnitudes = 2.5 * (23 - np.log10(fluxes)) - 48.6
/opt/herschelhelp_internal/herschelhelp_internal/utils.py:80: RuntimeWarning: invalid value encountered in divide
errors = 2.5 / np.log(10) * errors_on_fluxes / fluxes
```

1.2.1 I.a - g band



We will use aperture 5 as target.



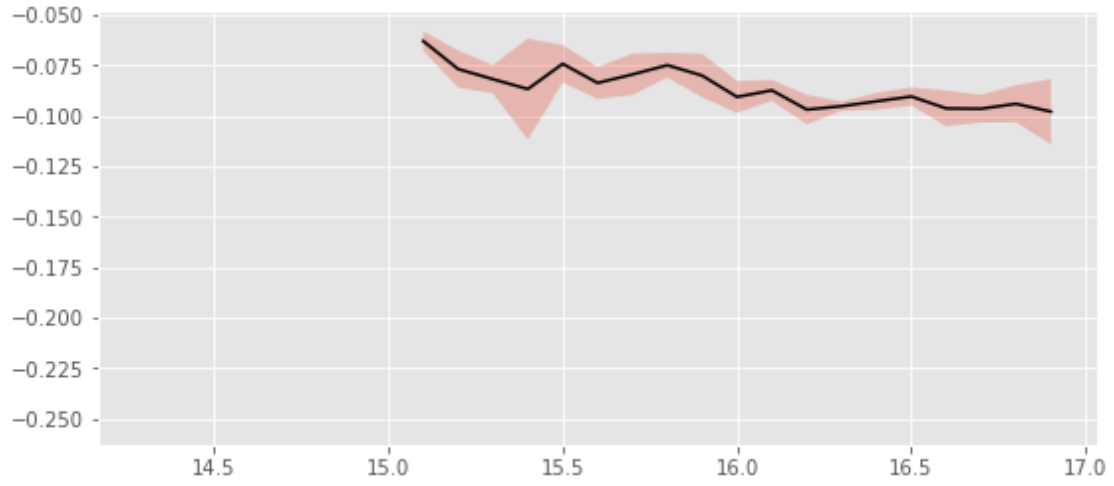
We will use magnitudes between 17.0 and 18.5

Aperture correction for g band:
Correction: -0.08293471348039816
Number of source used: 177
RMS: 0.016816685900966275

1.2.2 I.b - r band



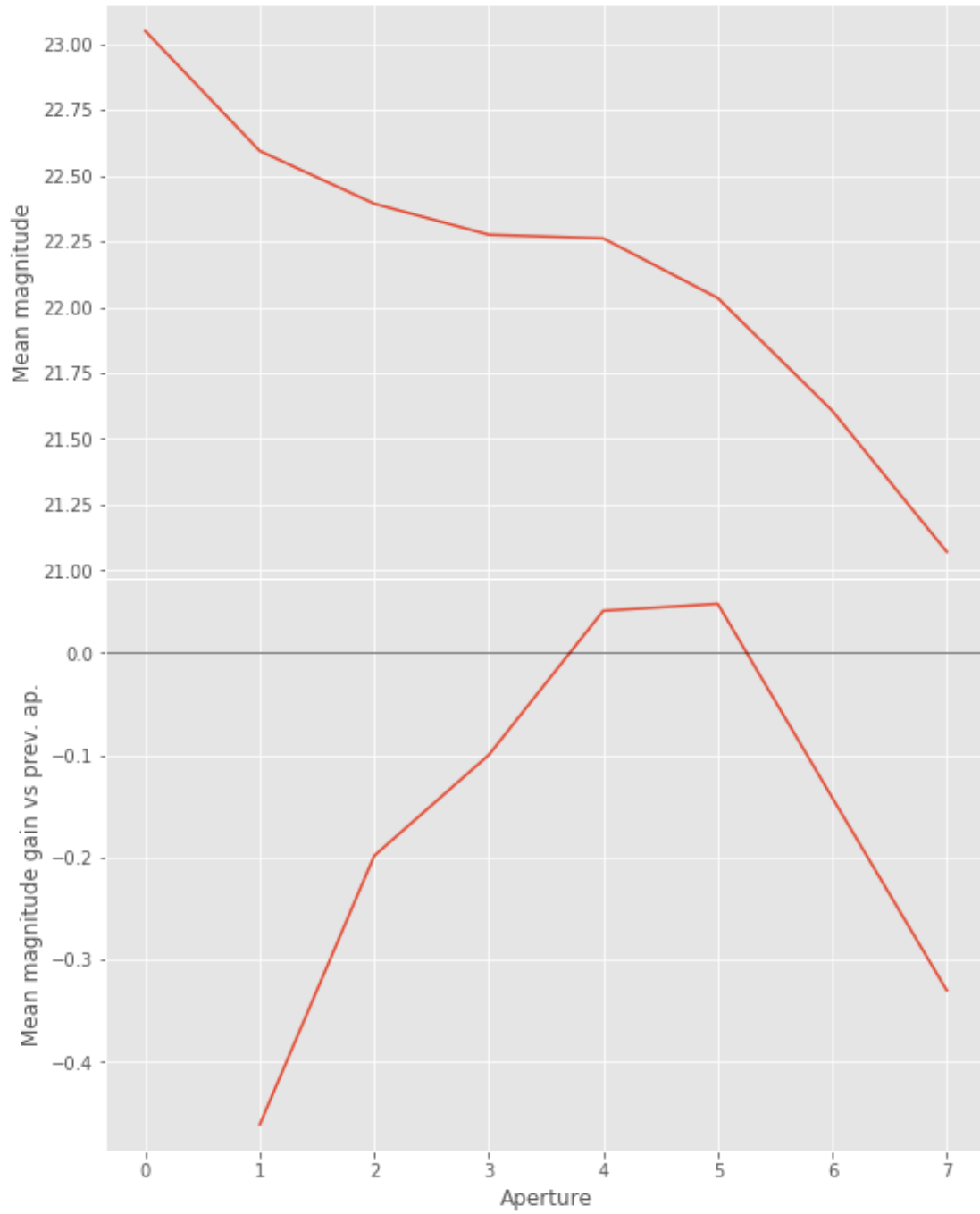
We will use aperture 5 as target.



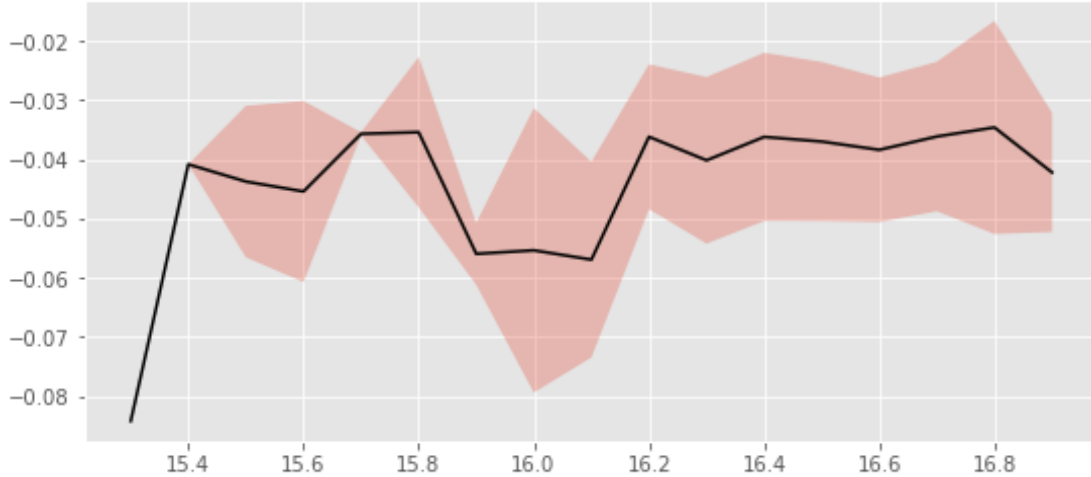
We use magnitudes between 15.5 and 18.5.

Aperture correction for r band:
Correction: -0.09303633544851664
Number of source used: 422
RMS: 0.007916295949312613

1.2.3 I.c - z band



We will use aperture 5 as target.



We use magnitudes between 16.2 and 17.5.

Aperture correction for z band:
 Correction: -0.03720670618650246
 Number of source used: 306
 RMS: 0.013323724091558286

1.3 II - Stellarity

Legacy Survey does not provide a 0 to 1 stellerity so we replace items flagged as PSF according to the following table:

$$P(\text{star}) = \frac{\prod_i P(\text{star})_i}{\prod_i P(\text{star})_i + \prod_i P(\text{galaxy})_i}$$

where i is the band, and with using the same probabilities as UKDISS:

HSC flag	UKIDSS flag	Meaning	P(star)	P(galaxy)	P(noise)	P(saturated)
	-9	Saturated	0.0	0.0	5.0	95.0
	-3	Probable galaxy	25.0	70.0	5.0	0.0
	-2	Probable star	70.0	25.0	5.0	0.0
0	-1	Star	90.0	5.0	5.0	0.0
	0	Noise	5.0	5.0	90.0	0.0
1	+1	Galaxy	5.0	90.0	5.0	0.0

1.4 II - Column selection

```
/opt/anaconda3/envs/herschelhelp_internal/lib/python3.6/site-packages/ipykernel/__main__.py:17:
/opt/herschelhelp_internal/herschelhelp_internal/utils.py:76: RuntimeWarning: divide by zero enc
magnitudes = 2.5 * (23 - np.log10(fluxes)) - 48.6
```



```
/opt/herschelhelp_internal/herschelhelp_internal/utils.py:76: RuntimeWarning: invalid value encountered in multiply
  magnitudes = 2.5 * (23 - np.log10(fluxes)) - 48.6
/opt/herschelhelp_internal/herschelhelp_internal/utils.py:80: RuntimeWarning: divide by zero encountered in divide
  errors = 2.5 / np.log(10) * errors_on_fluxes / fluxes
```

Out[19]: <IPython.core.display.HTML object>

1.5 III - Removal of duplicated sources

We remove duplicated objects from the input catalogues.

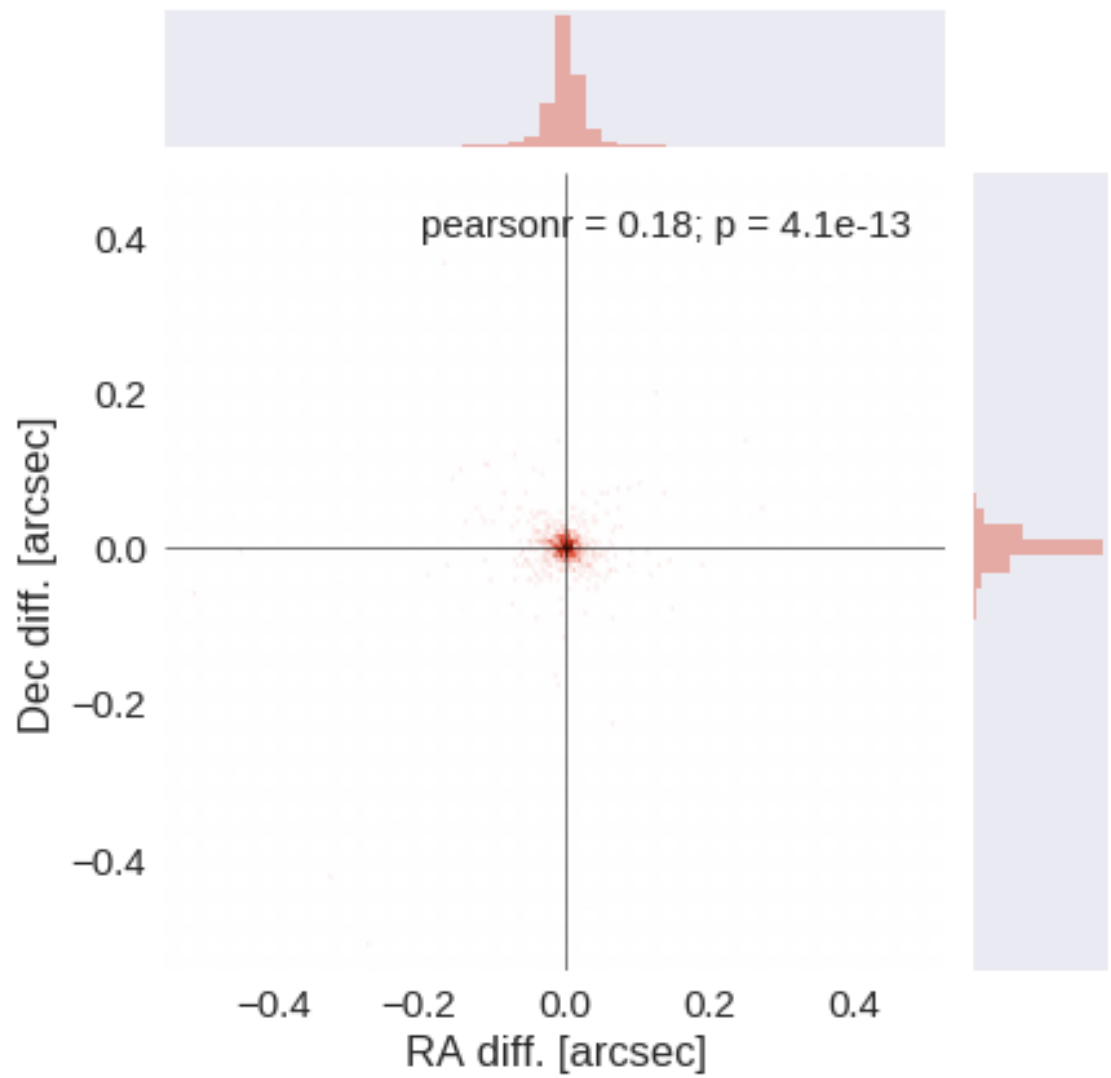
The initial catalogue had 38302 sources.

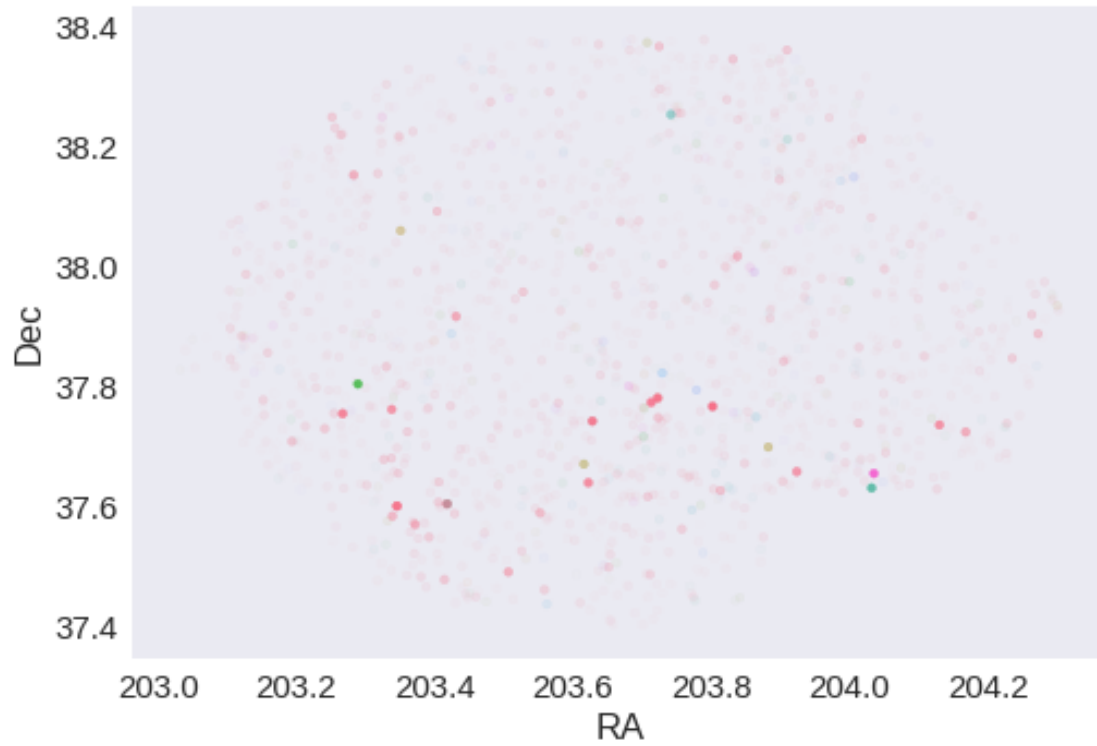
The cleaned catalogue has 37693 sources (609 removed).

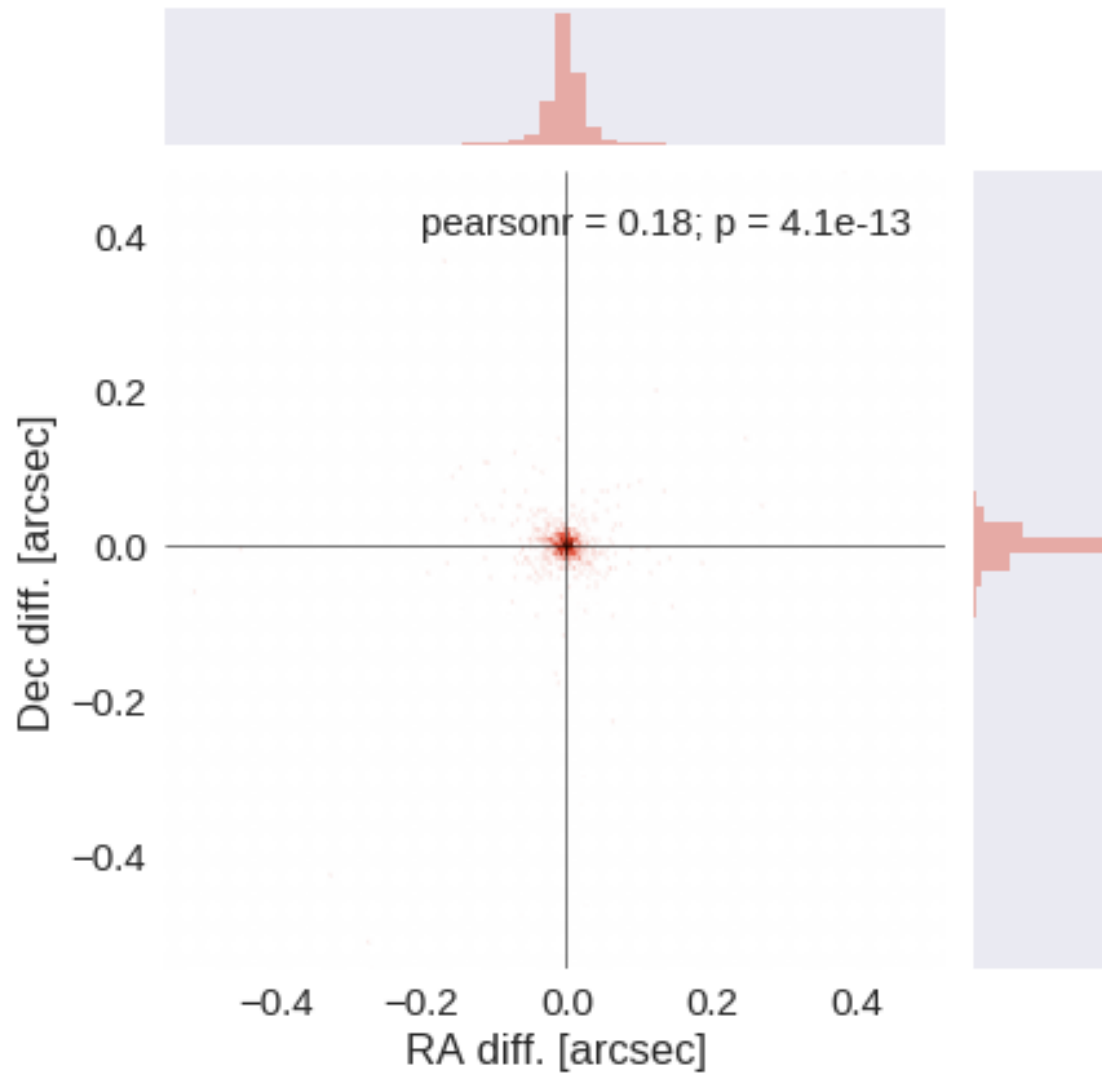
The cleaned catalogue has 600 sources flagged as having been cleaned

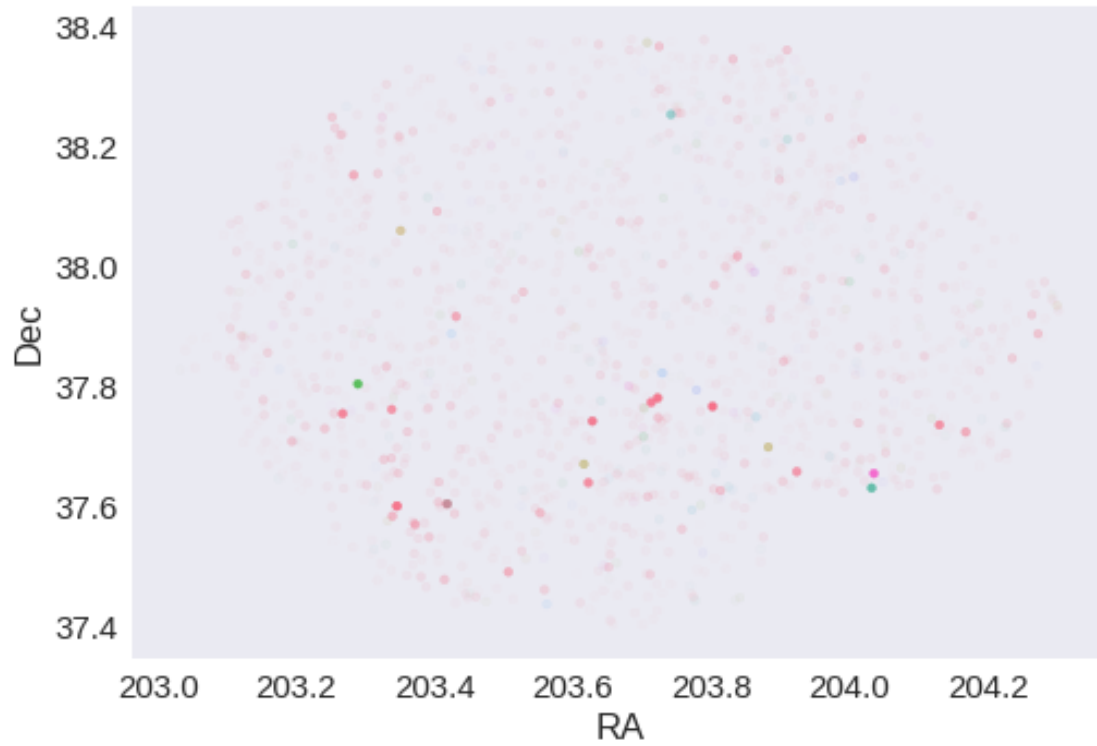
1.6 III - Astrometry correction

We match the astrometry to the Gaia one. We limit the Gaia catalogue to sources with a g band flux between the 30th and the 70th percentile. Some quick tests show that this give the lower dispersion in the results.









1.7 IV - Flagging Gaia objects

1641 sources flagged.

2 V - Saving to disk

1.2_UHS

March 8, 2018

1 XMM-13hr master catalogue

1.1 Preparation of UKIRT Hemisphere Survey (UHS) data

The catalogue comes from `dmu0_UHS`. This is a J band only survey documented in <https://arxiv.org/pdf/1707.09975.pdf>

In the catalogue, we keep:

- The identifier (it's unique in the catalogue);
- The position;
- The stellarity;
- The magnitude for each band in aperture 4 (2 arcsec aperture corrected).
- The kron magnitude to be used as total magnitude (no "auto" magnitude is provided).

We don't know when the maps have been observed. We will use the year of the reference paper.

```
This notebook was run with herchelhelp_internal version:
0246c5d (Thu Jan 25 17:01:47 2018 +0000) [with local modifications]
This notebook was executed on:
2018-02-21 14:07:59.843473
```

1.2 I - Column selection

```
0.925175419285
```

```
/opt/anaconda3/envs/herchelhelp_internal/lib/python3.6/site-packages/astropy/table/column.py:10
Check the NumPy 1.11 release notes for more information.
ma.MaskedArray.__setitem__(self, index, value)
```

```
Out[7]: <IPython.core.display.HTML object>
```

1.3 II - Removal of duplicated sources

We remove duplicated objects from the input catalogues.

```
/opt/anaconda3/envs/herschelhelp_internal/lib/python3.6/site-packages/astropy/table/column.py:10
```

Check the NumPy 1.11 release notes for more information.

```
ma.MaskedArray.__setitem__(self, index, value)
```

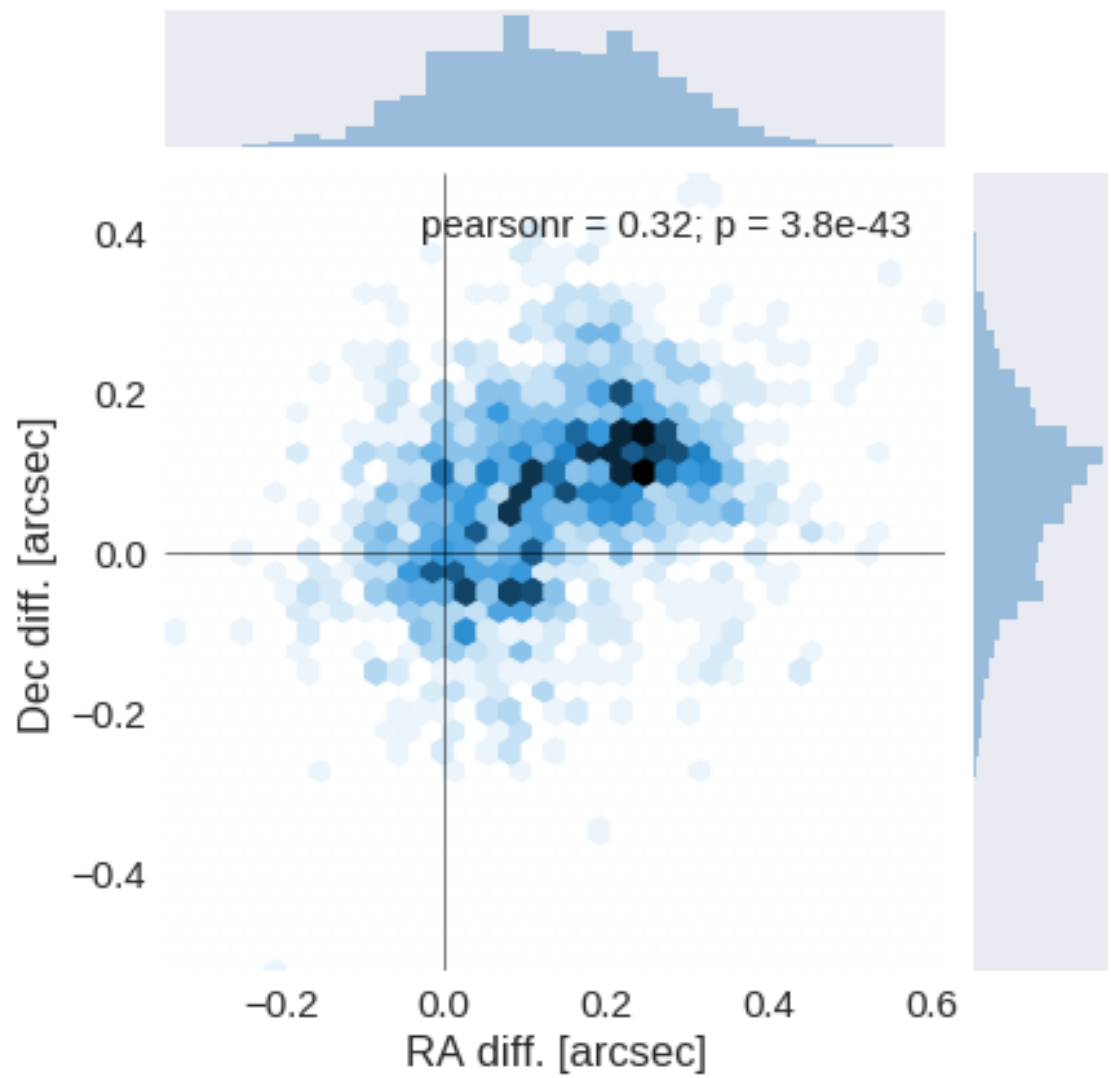
The initial catalogue had 7879 sources.

The cleaned catalogue has 7186 sources (693 removed).

The cleaned catalogue has 666 sources flagged as having been cleaned

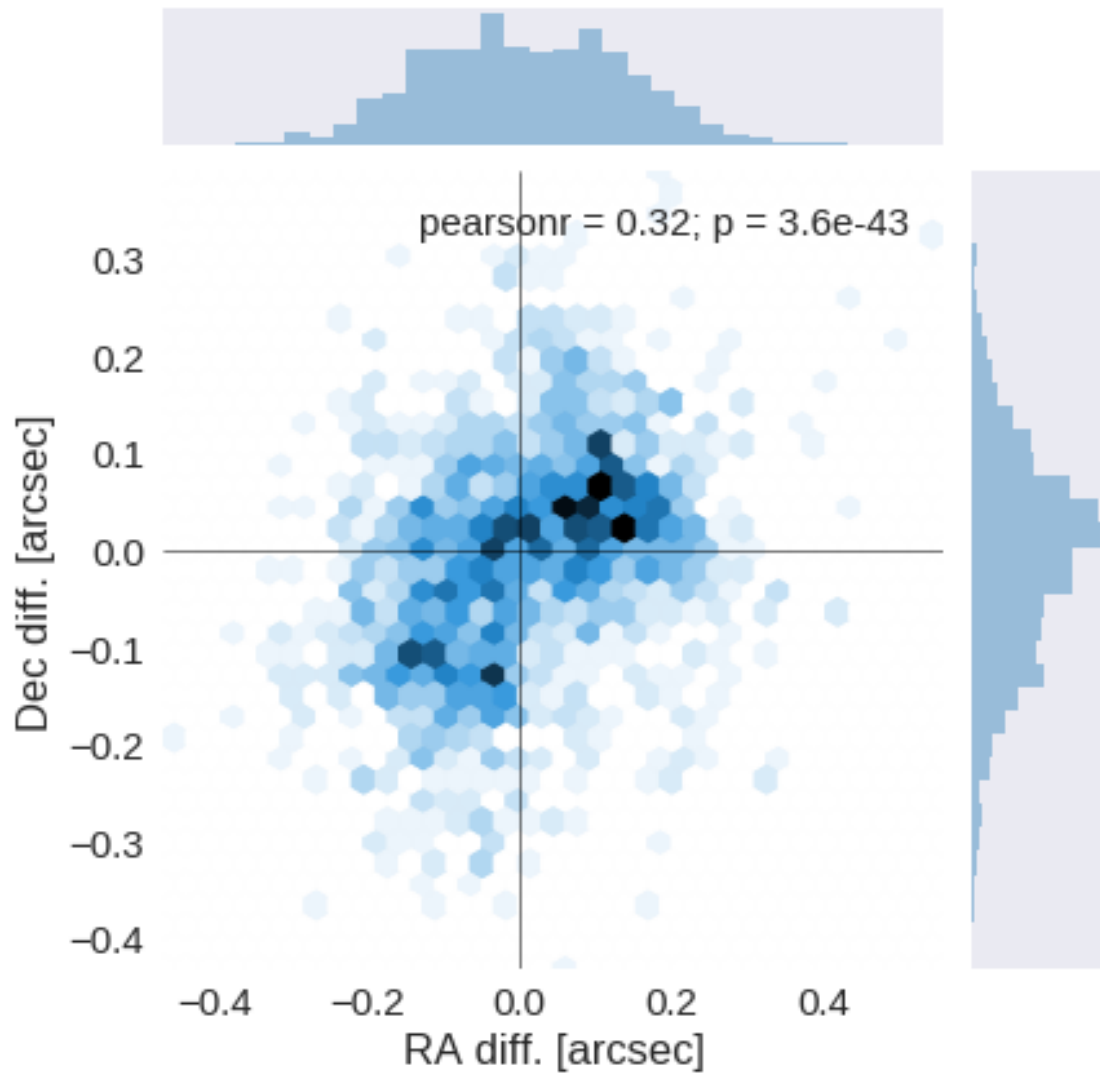
1.4 III - Astrometry correction

We match the astrometry to the Gaia one. We limit the Gaia catalogue to sources with a g band flux between the 30th and the 70th percentile. Some quick tests show that this give the lower dispersion in the results.





RA correction: -0.12516495181102982 arcsec
Dec correction: -0.082956100101228 arcsec





1.5 IV - Flagging Gaia objects

1733 sources flagged.

2 V - Saving to disk

2_Merging

March 8, 2018

1 XMM-13hr master catalogue

This notebook presents the merge of the various pristine catalogues to produce HELP mater catalogue on XMM-13hr.

This notebook was run with `herschelhelp_internal` version:
0246c5d (Thu Jan 25 17:01:47 2018 +0000) [with local modifications]
This notebook was executed on:
2018-02-21 14:21:01.887837

1.1 I - Reading the prepared pristine catalogues

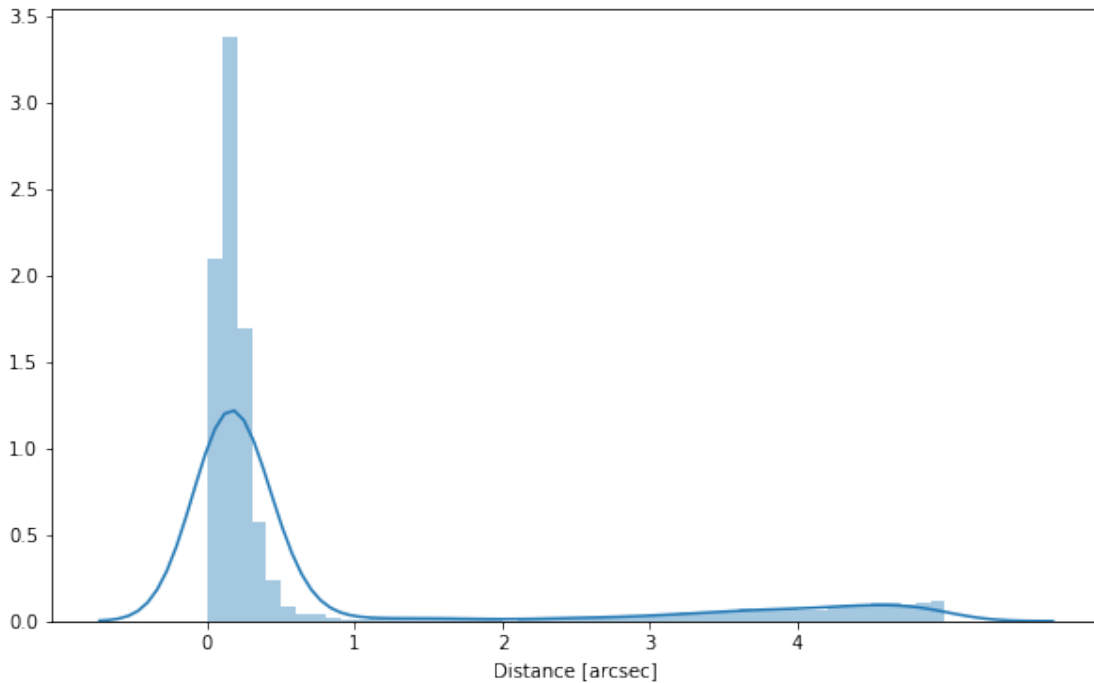
1.2 II - Merging tables

We merge UHS and Legacy Survey.

At every step, we look at the distribution of the distances to the nearest source in the merged catalogue to determine the best crossmatching radius.

1.2.1 UHS

1.2.2 Add Legacy Survey



1.2.3 Cleaning

When we merge the catalogues, astropy masks the non-existent values (e.g. when a row comes only from a catalogue and has no counterparts in the other, the columns from the latest are masked for that row). We indicate to use NaN for masked values for floats columns, False for flag columns and -1 for ID columns.

Out [9]: <IPython.core.display.HTML object>

1.3 III - Merging flags and stellerity

Each pristine catalogue contains a flag indicating if the source was associated to a another nearby source that was removed during the cleaning process. We merge these flags in a single one.

Each pristine catalogue contains a flag indicating the probability of a source being a Gaia object (0: not a Gaia object, 1: possibly, 2: probably, 3: definitely). We merge these flags taking the highest value.

Each prisitine catalogue may contain one or several stellerity columns indicating the probability (0 to 1) of each source being a star. We merge these columns taking the highest value. We keep trace of the origin of the stellerity.

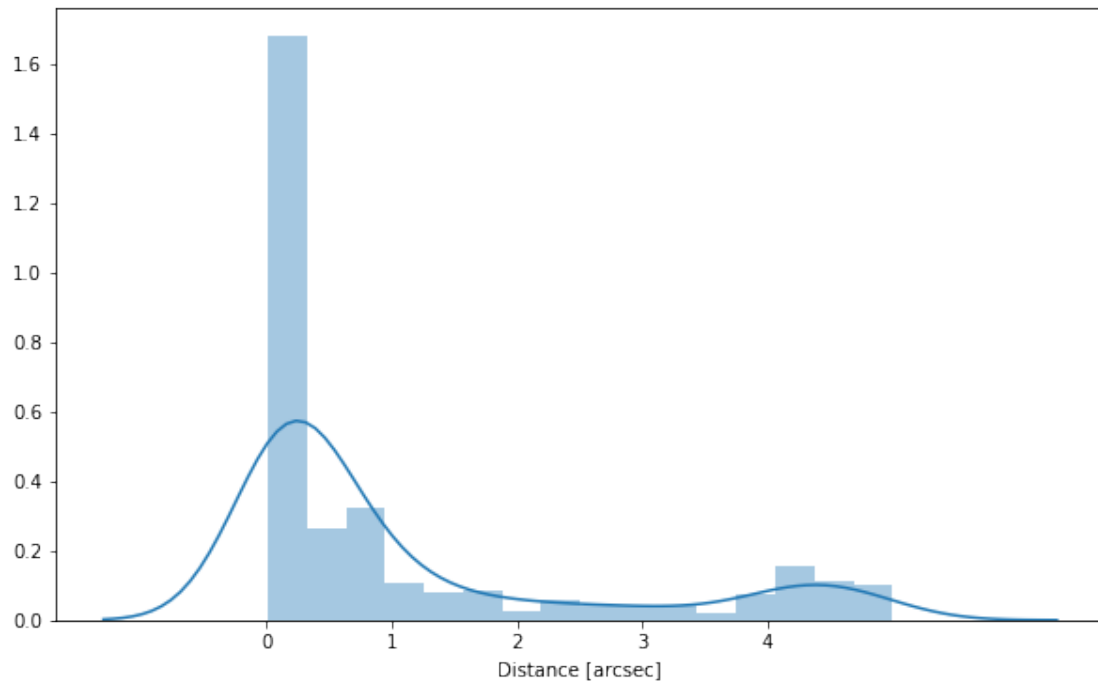
uhs_stellariry, legacy_stellariry

1.4 IV - Adding E(B-V) column

1.5 V - Adding HELP unique identifiers and field columns

OK!

1.6 VI - Cross-matching with spec-z catalogue



1.7 VII - Choosing between multiple values for the same filter

There are no duplicate filters

1.8 VIII.a Wavelength domain coverage

We add a binary `flag_optnir_obs` indicating that a source was observed in a given wavelength domain:

- 1 for observation in optical;
- 2 for observation in near-infrared;
- 4 for observation in mid-infrared (IRAC).

It's an integer binary flag, so a source observed both in optical and near-infrared but not in mid-infrared would have this flag at $1 + 2 = 3$.

Note 1: The observation flag is based on the creation of multi-order coverage maps from the catalogues, this may not be accurate, especially on the edges of the coverage.

Note 2: Being on the observation coverage does not mean having fluxes in that wavelength domain. For sources observed in one domain but having no flux in it, one must take into consideration de different depths in the catalogue we are using.

1.9 VIII.b Wavelength domain detection

We add a binary `flag_optnir_det` indicating that a source was detected in a given wavelength domain:

- 1 for detection in optical;
- 2 for detection in near-infrared;
- 4 for detection in mid-infrared (IRAC).

It's an integer binary flag, so a source detected both in optical and near-infrared by not in mid-infrared would have this flag at $1 + 2 = 3$.

Note 1: We use the total flux columns to know if the source has flux, in some catalogues, we may have aperture flux and no total flux.

To get rid of artefacts (chip edges, star flares, etc.) we consider that a source is detected in one wavelength domain when it has a flux value in **at least two bands**. That means that good sources will be excluded from this flag when they are on the coverage of only one band.

1.10 IX - Cross-identification table

We are producing a table associating to each HELP identifier, the identifiers of the sources in the pristine catalogues. This can be used to easily get additional information from them.

There is no SDSS on XMM-13hr.

```
['uhs_id', 'legacy_id', 'help_id', 'specz_id']
```

1.11 X - Adding HEALPix index

We are adding a column with a HEALPix index at order 13 associated with each source.

1.12 XI - Saving the catalogue

```
Missing columns: set()
```

3_Checks_and_diagnostics

March 8, 2018

1 XMM-13hr master catalogue

1.1 Checks and diagnostics

There is very little data on XMM-13hr so there is very little data to compare to.

This notebook was run with `herschelhelp_internal` version:
0246c5d (Thu Jan 25 17:01:47 2018 +0000) [with local modifications]
This notebook was executed on:
2018-02-21 14:21:47.320364

Diagnostics done using: `master_catalogue_xmm-13hr_20180221.fits`

1.2 0 - Quick checks

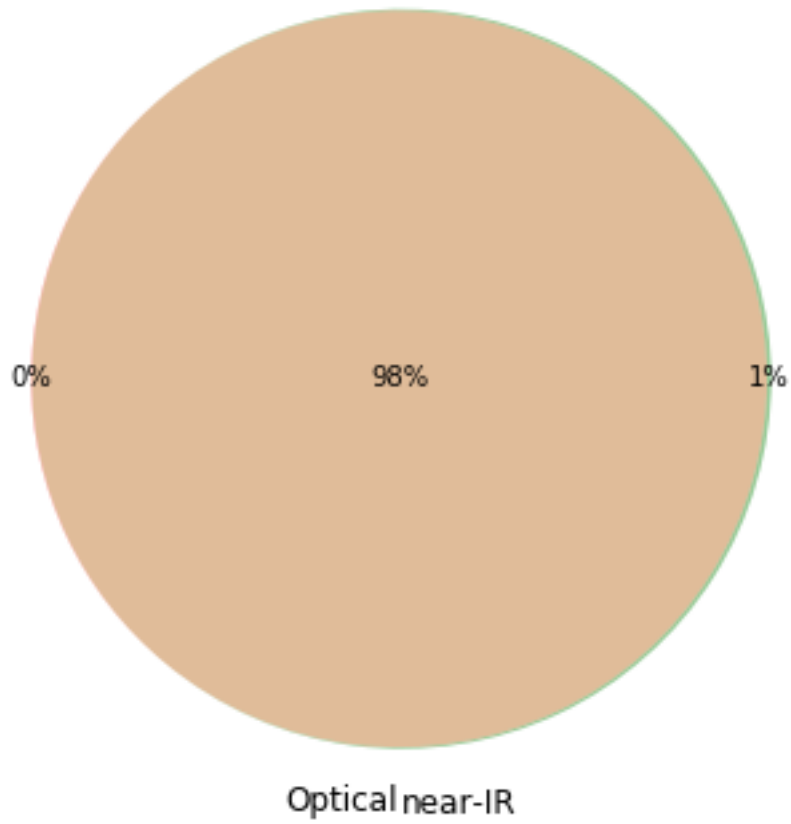
Table shows only problematic columns.

```
/opt/anaconda3/envs/herschelhelp_internal/lib/python3.6/site-packages/numpy/core/numeric.py:301:
  format(shape, fill_value, array(fill_value).dtype), FutureWarning)
/opt/anaconda3/envs/herschelhelp_internal/lib/python3.6/site-packages/numpy/core/numeric.py:301:
  format(shape, fill_value, array(fill_value).dtype), FutureWarning)
```

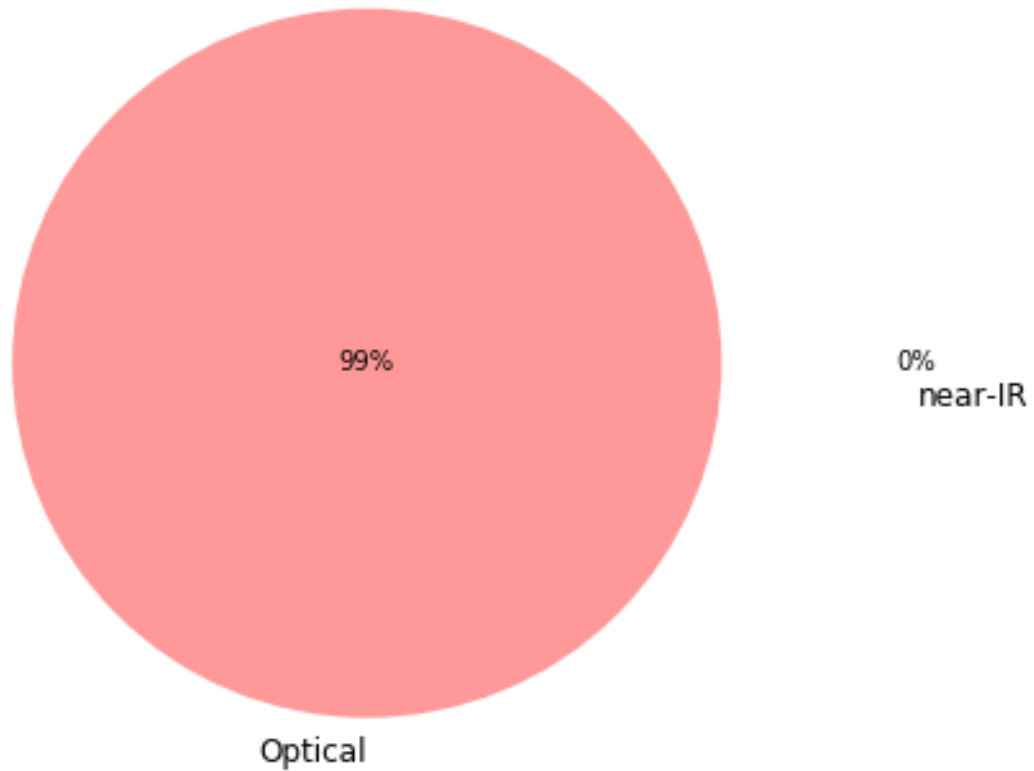
Out[4]: <IPython.core.display.HTML object>

1.3 I - Summary of wavelength domains

Wavelength domain observations



Detection of the 37,693 sources detected
in any wavelength domains (among 38,629 sources)

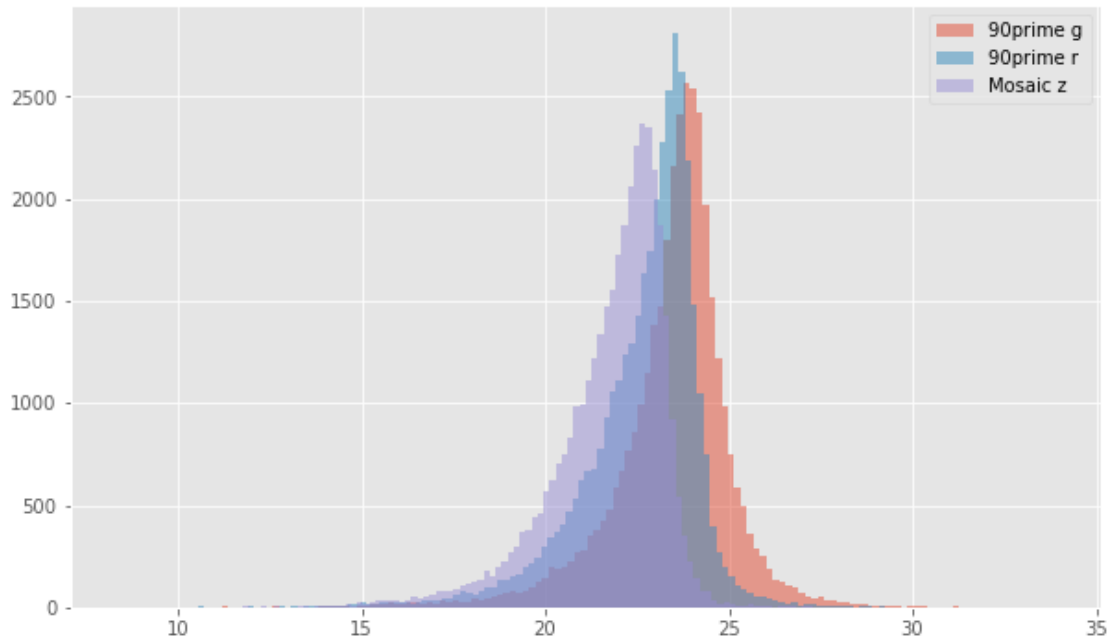


1.4 II - Comparing magnitudes in similar filters

The master list is composed of several catalogues containing magnitudes in similar filters on different instruments. We are comparing the magnitudes in these corresponding filters.

1.4.1 II.a - Comparing depths

We compare the histograms of the total aperture magnitudes of similar bands.



1.4.2 II.b - Comparing magnitudes

There are no columns with similar bands.

1.5 III - Comparing magnitudes to reference bands

Cross-match the master list to 2MASS to compare magnitudes.

1.5.1 III.b - Comparing J band to 2MASS

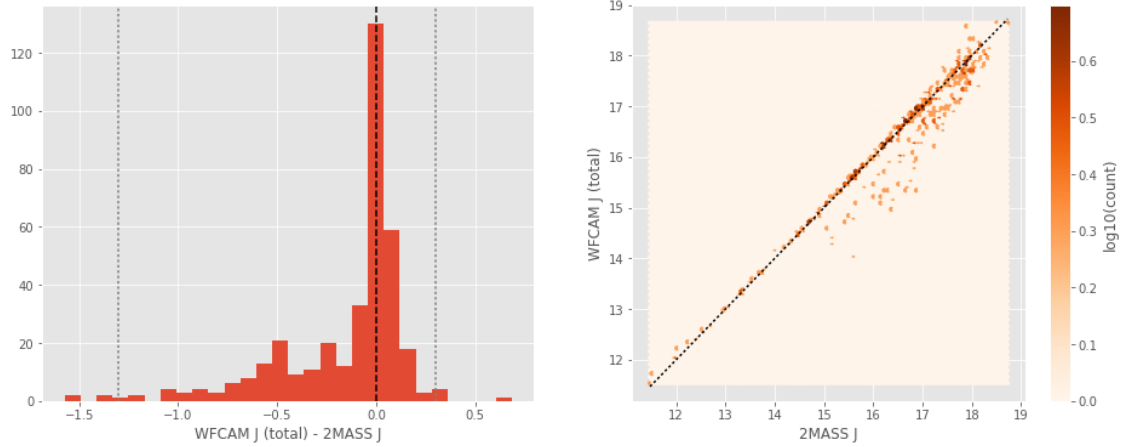
The catalogue is cross-matched to 2MASS-PSC within 0.2 arcsecond. We compare the UKIDSS total J and K magnitudes to those from 2MASS.

The 2MASS magnitudes are “Vega-like” and we have to convert them to AB magnitudes using the zero points provided on [this page](#):

Band	F - 0 mag (Jy)
J	1594
H	1024
Ks	666.7

WFCAM J (total) - 2MASS J:

- Median: -0.02
- Median Absolute Deviation: 0.08
- 1% percentile: -1.3014426617338006
- 99% percentile: 0.2973091131685438



Based on the first run of the above graph it became clear that uhs requires conversion from Vega to AB.

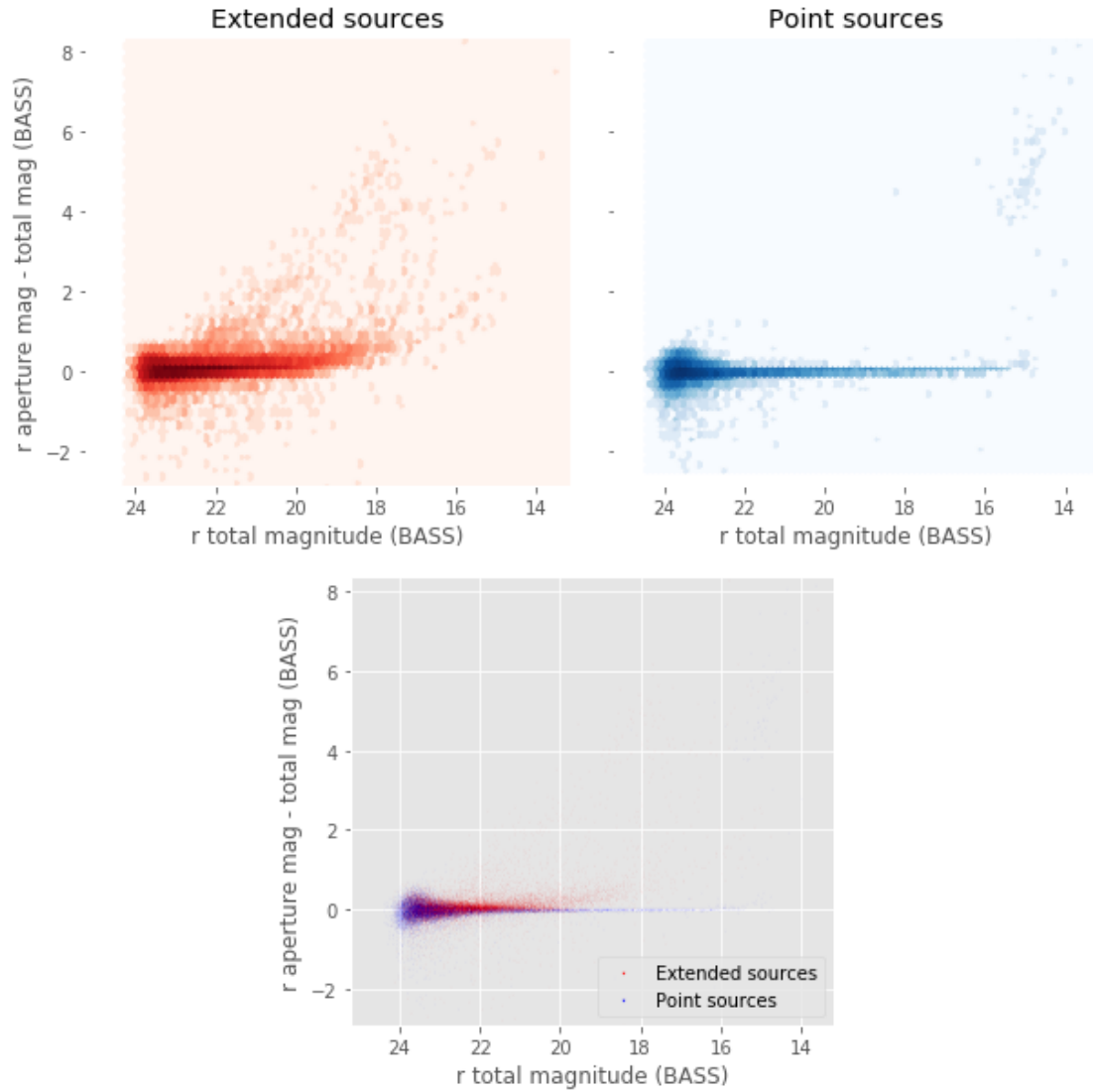
1.6 Keeping only sources with good signal to noise ratio

From here, we are only comparing sources with a signal to noise ratio above 3, i.e. roughly we a magnitude error below 0.3.

To make it easier, we are setting to NaN in the catalogue the magnitudes associated with an error above 0.3 so we can't use these magnitudes after the next cell.

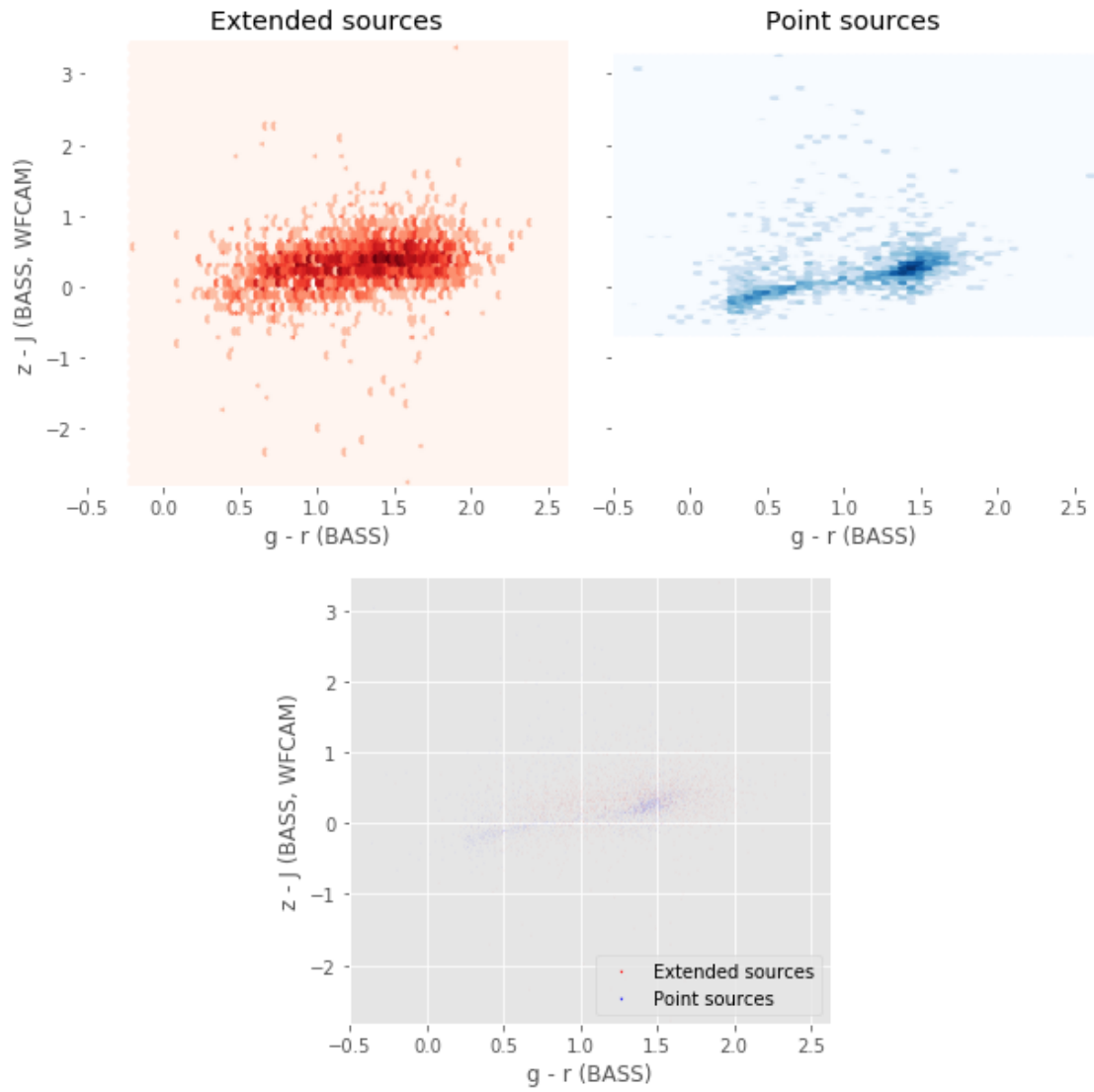
1.7 IV - Comparing aperture magnitudes to total ones.

Number of source used: 31910 / 38629 (82.61%)



1.8 V - Color-color and magnitude-color plots

Number of source used: 4355 / 38629 (11.27%)



4_Selection_function

March 8, 2018

1 XMM-13hr Selection Functions

1.1 Depth maps and selection functions

The simplest selection function available is the field MOC which specifies the area for which there is Herschel data. Each pristine catalogue also has a MOC defining the area for which that data is available.

The next stage is to provide mean flux standard deviations which act as a proxy for the catalogue's 5σ depth

```
This notebook was run with herschelhelp_internal version:  
0246c5d (Thu Jan 25 17:01:47 2018 +0000) [with local modifications]  
This notebook was executed on:  
2018-02-28 12:57:50.327686
```

Depth maps produced using: master_catalogue_xmm-13hr_20180221.fits

1.2 I - Group masterlist objects by healpix cell and calculate depths

We add a column to the masterlist catalogue for the target order healpix cell per object.

1.3 II Create a table of all Order=13 healpix cells in the field and populate it

We create a table with every order=13 healpix cell in the field MOC. We then calculate the healpix cell at lower order that the order=13 cell is in. We then fill in the depth at every order=13 cell as calculated for the lower order cell that that the order=13 cell is inside.

```
Out[9]: <IPython.core.display.HTML object>
```

```
Out[11]: <IPython.core.display.HTML object>
```

```
Out[12]: <IPython.core.display.HTML object>
```

1.4 III - Save the depth map table

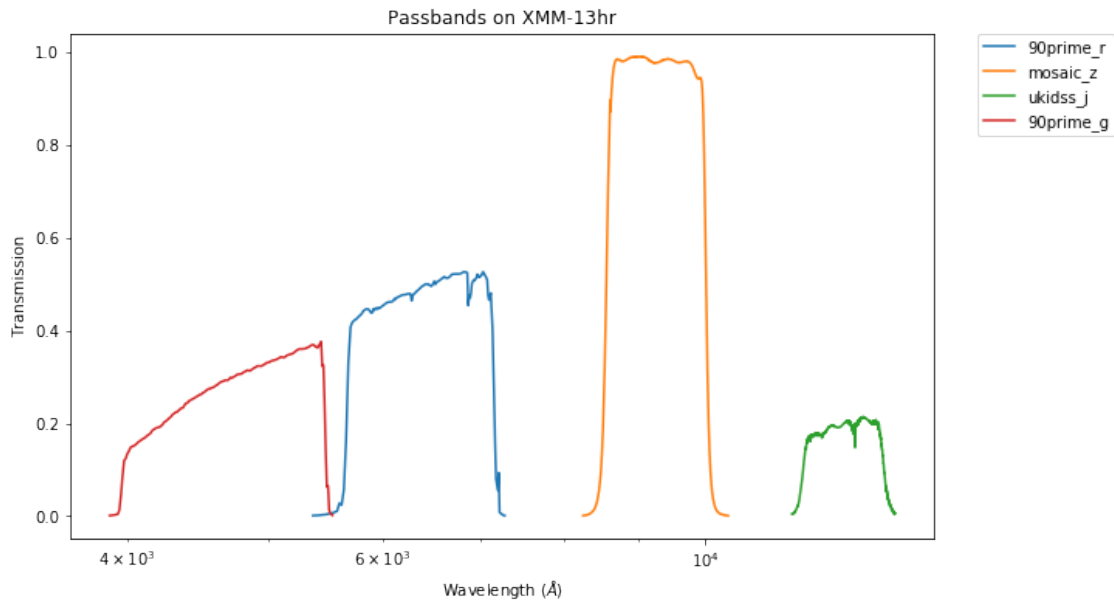
1.5 IV - Overview plots

1.5.1 IV.a - Filters

First we simply plot all the filters available on this field to give an overview of coverage.

```
Out[14]: {'90prime_g', '90prime_r', 'mosaic_z', 'ukidss_j'}
```

```
Out[15]: <matplotlib.text.Text at 0x7fe329be9c18>
```



1.5.2 IV.a - Depth overview

Then we plot the mean depths available across the area a given band is available

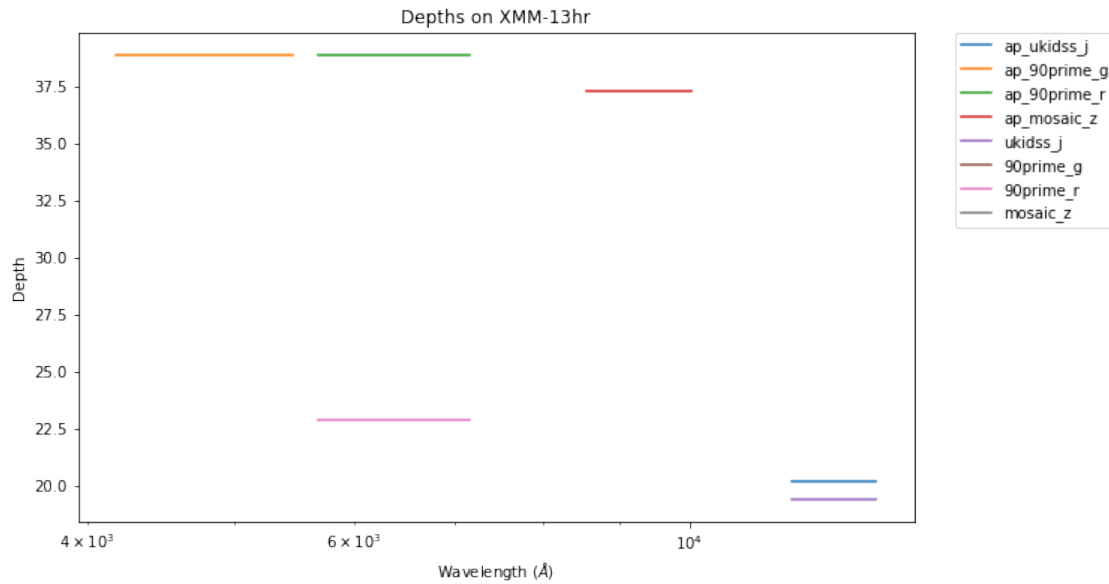
```
ukidss_j: mean flux error: 6.260478496551514, 3sigma in AB mag (Aperture): 20.715678042766548
90prime_g: mean flux error: 2.1222248847152514e-07, 3sigma in AB mag (Aperture): 39.390218356540
90prime_r: mean flux error: 2.0418607959982182e-07, 3sigma in AB mag (Aperture): 39.432131536441
mosaic_z: mean flux error: 9.113107353186933e-07, 3sigma in AB mag (Aperture): 37.80803064737608
ukidss_j: mean flux error: 12.898414611816406, 3sigma in AB mag (Total): 19.930856030796242
90prime_g: mean flux error: inf, 3sigma in AB mag (Total): -inf
90prime_r: mean flux error: 0.5266921520233154, 3sigma in AB mag (Total): 23.4033047451995
mosaic_z: mean flux error: inf, 3sigma in AB mag (Total): -inf
```

```
ap_ukidss_j (11695.0, 13280.0, 1585.0)
ap_90prime_g (4180.0, 5470.0, 1290.0)
ap_90prime_r (5680.0, 7150.0, 1470.0)
```



```
ap_mosaic_z (8552.0, 10018.0, 1466.0)
ukidss_j (11695.0, 13280.0, 1585.0)
90prime_g (4180.0, 5470.0, 1290.0)
90prime_r (5680.0, 7150.0, 1470.0)
mosaic_z (8552.0, 10018.0, 1466.0)
```

Out[20]: <matplotlib.text.Text at 0x7fe32974b400>



1.5.3 IV.c - Depth vs coverage comparison

How best to do this? Colour/intensity plot over area? Percentage coverage vs mean depth?

Out[21]: <matplotlib.text.Text at 0x7fe329baaba8>

