DIALS as LEGO

Build your own data processing toys!

Ben Williams – Diamond Light Source

diamond

diamond

wellcome

BERKELEY LAB
Lawrence Berkeley National Laboratory

BioStruct-X

Science & Technology Facilities Council

CCP4

EU FP7: #283570 NIH: GM095887 & GM102520

diamond

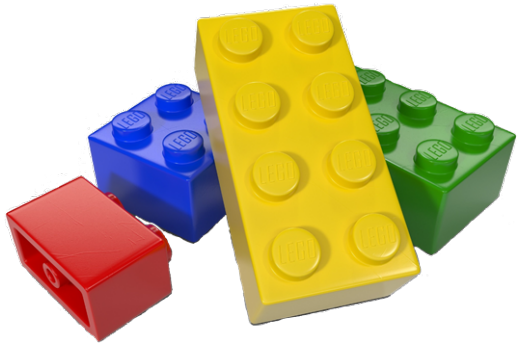
A pipeline, and a toolkit too!

DIALS is a toolkit for analysing diffraction data.

It's built in Python (for the most part).

You already have the toolkit!

```
$ dials.python
```



diamond

A pipeline, and a toolkit too!

Documentation:

https://dials.github.io

diamond

The DIALS data structure

Data files

The DIALS programs read and write three main types of data file for storing the experimental geometry, image data and processed reflection data. These are summarized in the following table and described in more detail in the sections below.

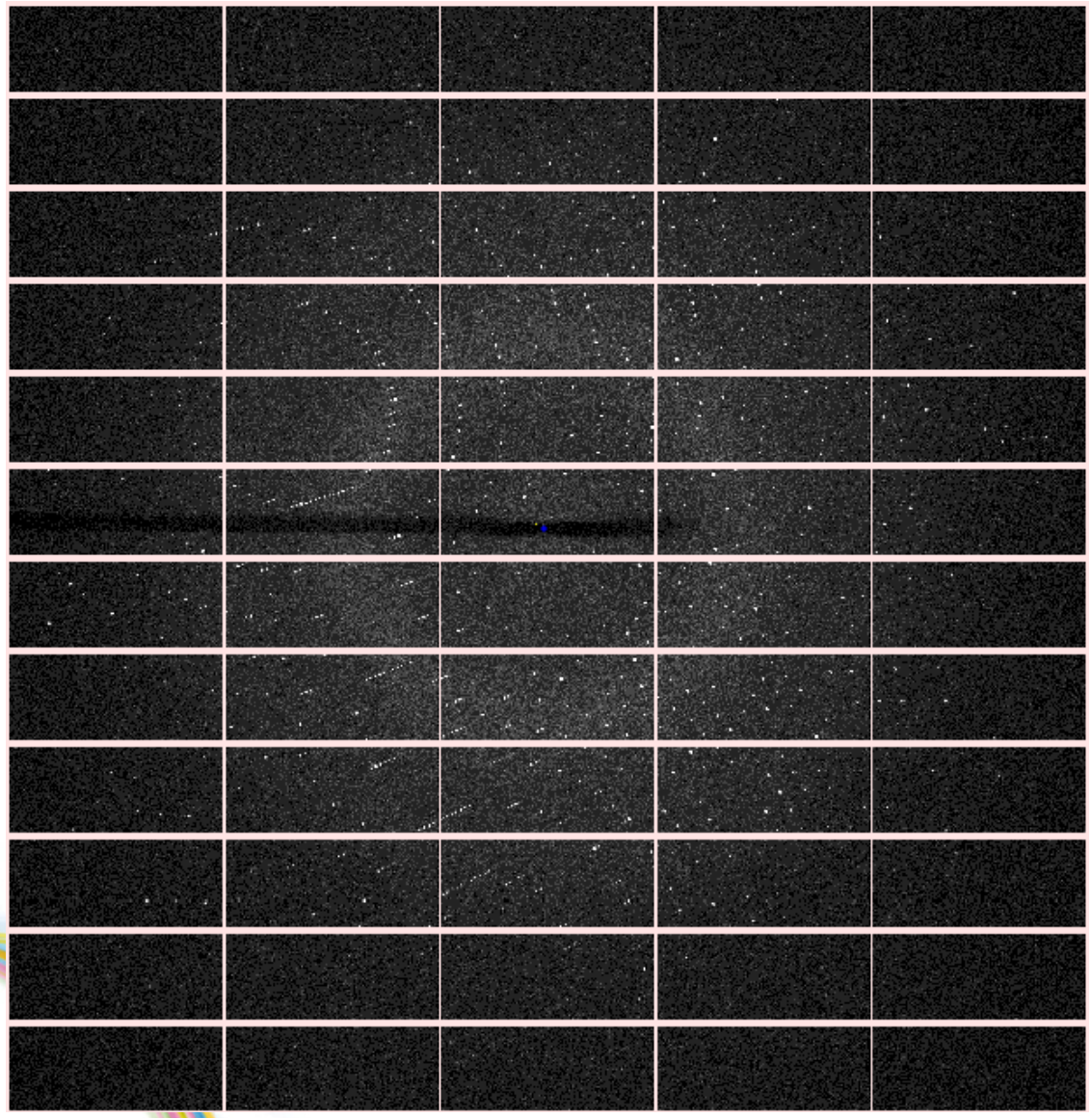
File type	Contains
Datablock	Experimental geometry and image data
Experiment list	Experimental geometry (plus crystal) and image data
Reflection table	Processed reflection data

https://dials.github.io/documentation/data_files.html

diamond

Four little Lego sets

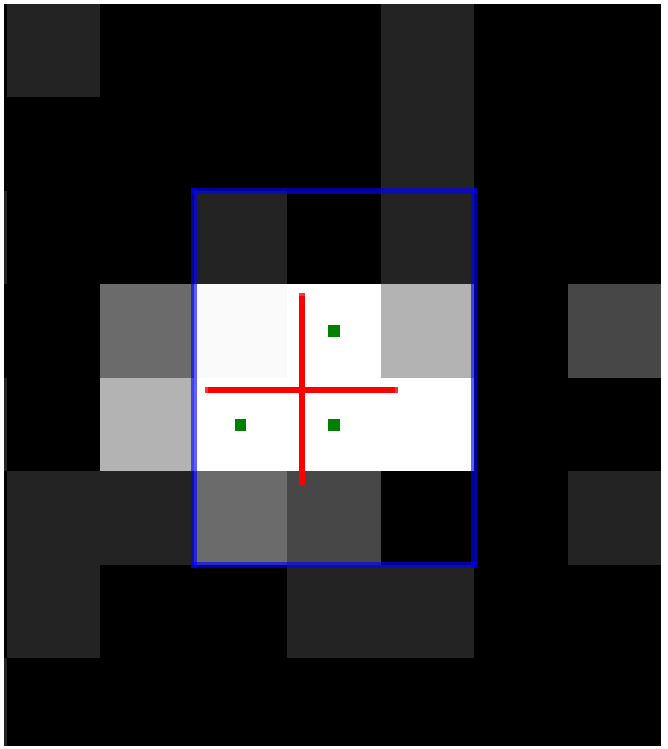
1. How big are the spots?
2. Which spots appear on only one image?
3. Can we extract the background?
4. Ooh, let's look at integrated intensities for systematically absent reflections!



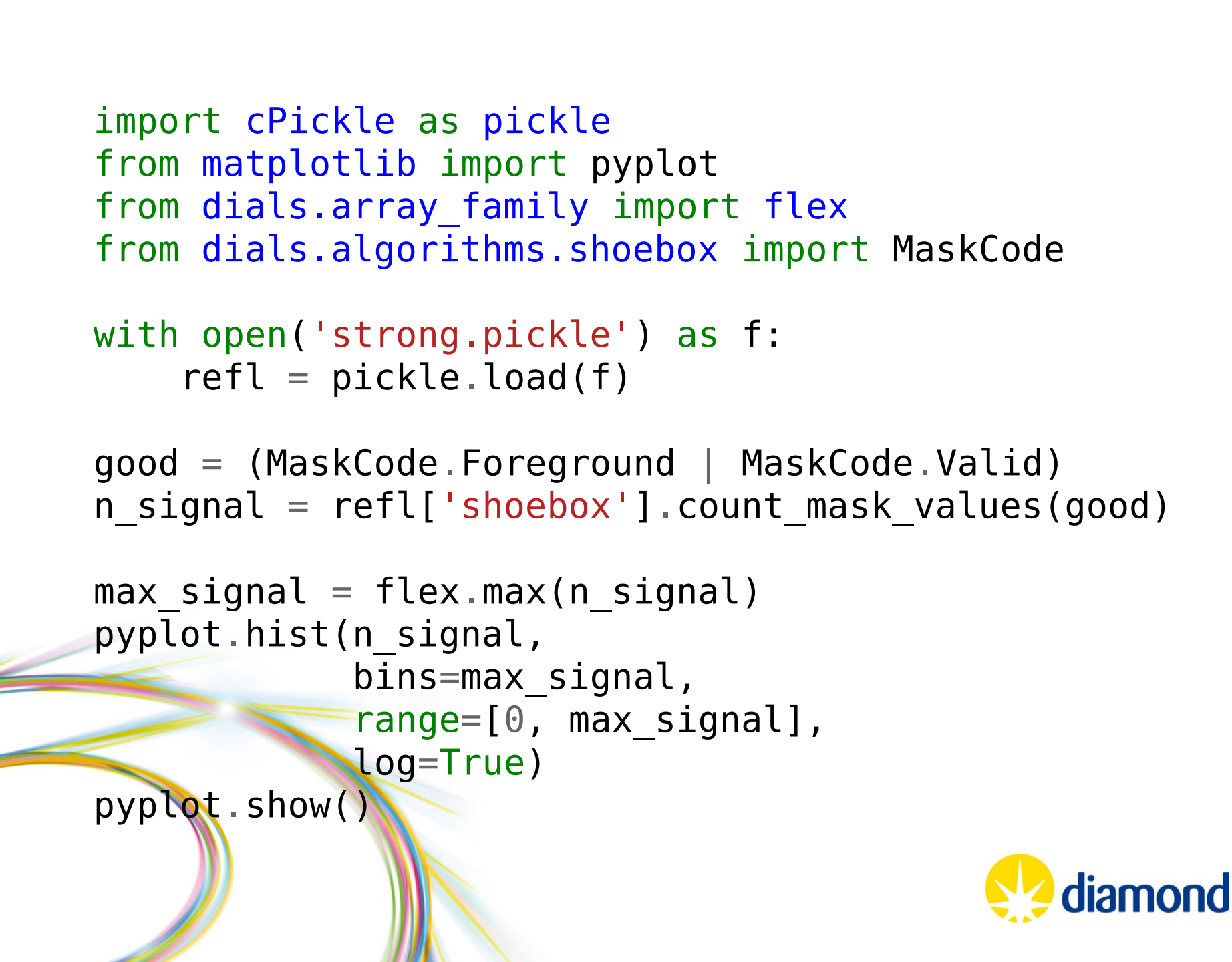
https://dials.github.io/documentation/tutorials/processing_in_detail_tutorial.html

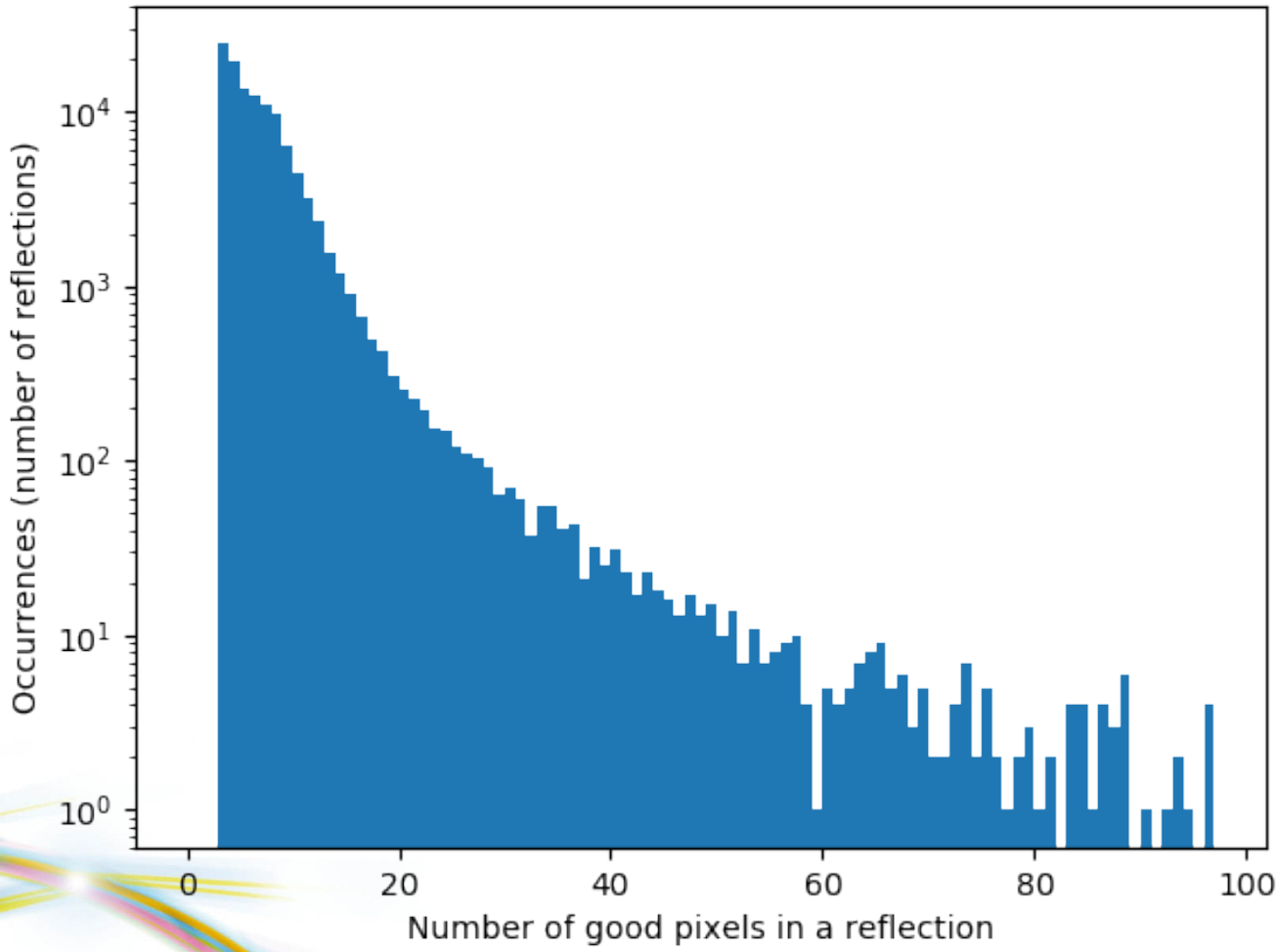
What do the spots consist of?

How big are they?



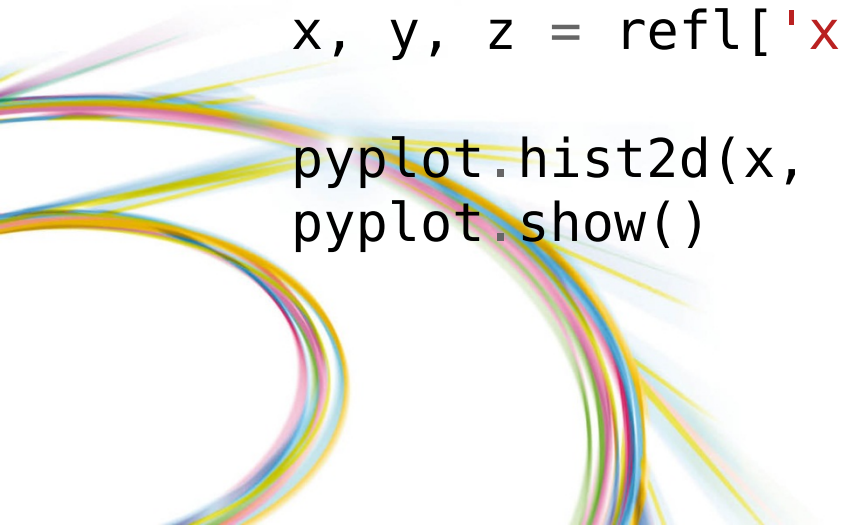
```
import cPickle as pickle
from matplotlib import pyplot
from dials.array_family import flex
from dials.algorithms.shoebox import MaskCode

with open('strong.pickle') as f:
    refl = pickle.load(f)

good = (MaskCode.Foreground | MaskCode.Valid)
n_signal = refl['shoebox'].count_mask_values(good)

max_signal = flex.max(n_signal)
pyplot.hist(n_signal,
            bins=max_signal,
            range=[0, max_signal],
            log=True)
pyplot.show()
```



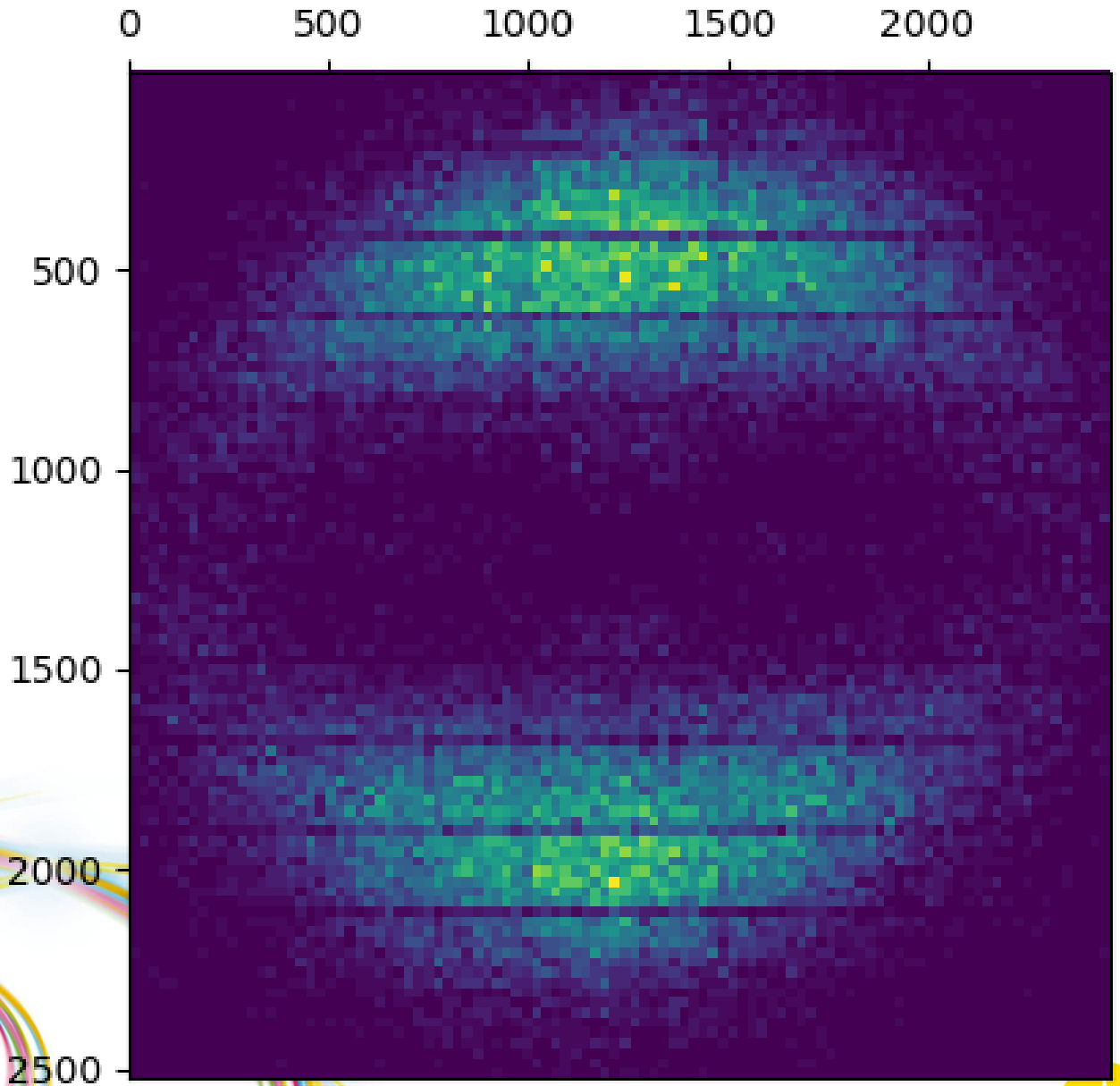

Occurrences (number of reflections)
10^4
10^3
10^2
10^1
10^0
0
20
40
60
80
100
Number of good pixels in a reflection

In which part of the detector do we most often see spots only appearing on a single image?

```
import cPickle as pickle

with open('strong.pickle') as f:
    refl = pickle.load(f)

z = refl['xyzobs.px.value'].parts()[2]
refl = refl.select(z < 1200)

z0, z1 = refl['bbox'].parts()[4:]
refl = refl.select((z1 - z0) == 1)

x, y, z = refl['xyzobs.px.value'].parts()

pyplot.hist2d(x, y, bins=[100,100])
pyplot.show()
```



diamond

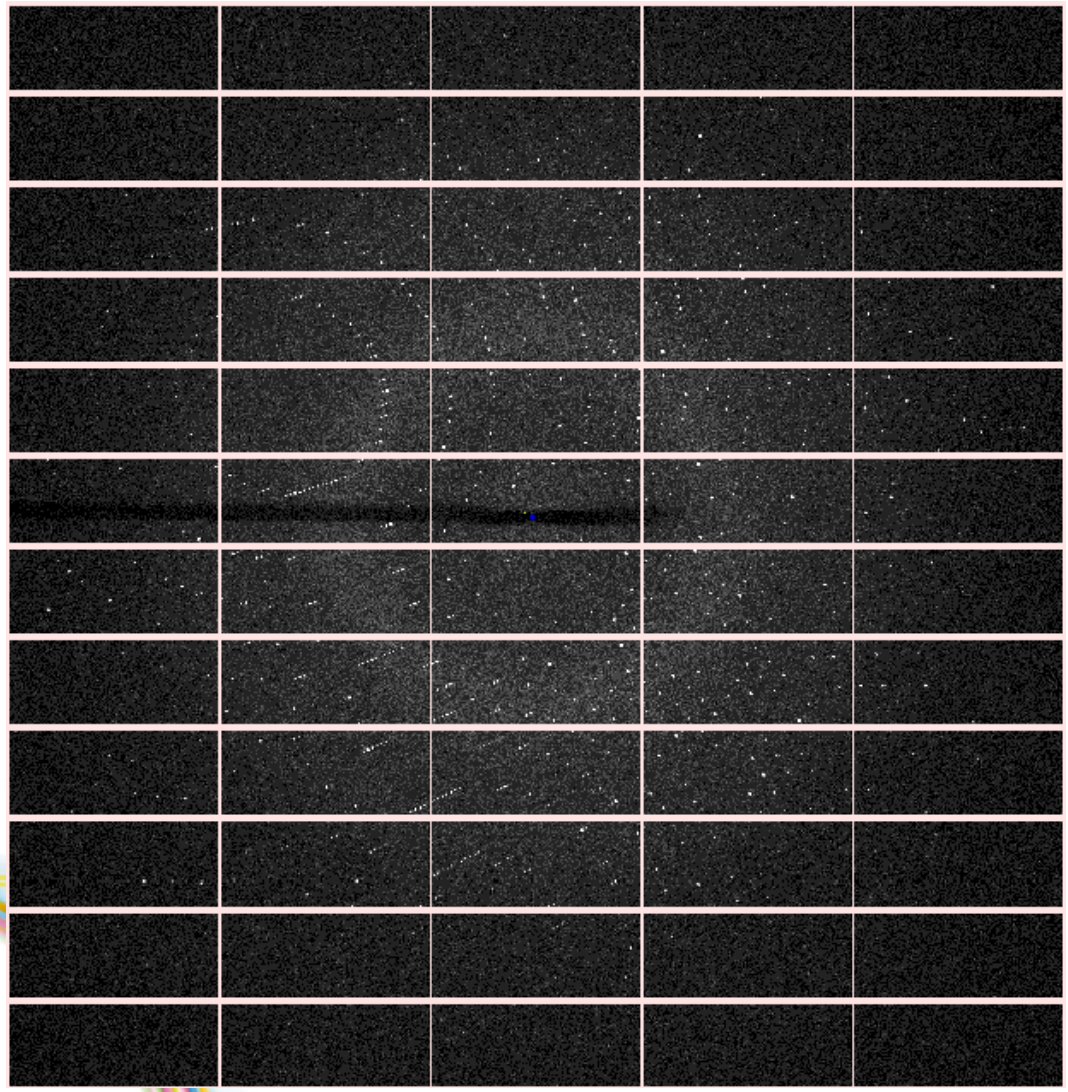

0
500
1000
1500
2000
500
1000
1500
2000
2500

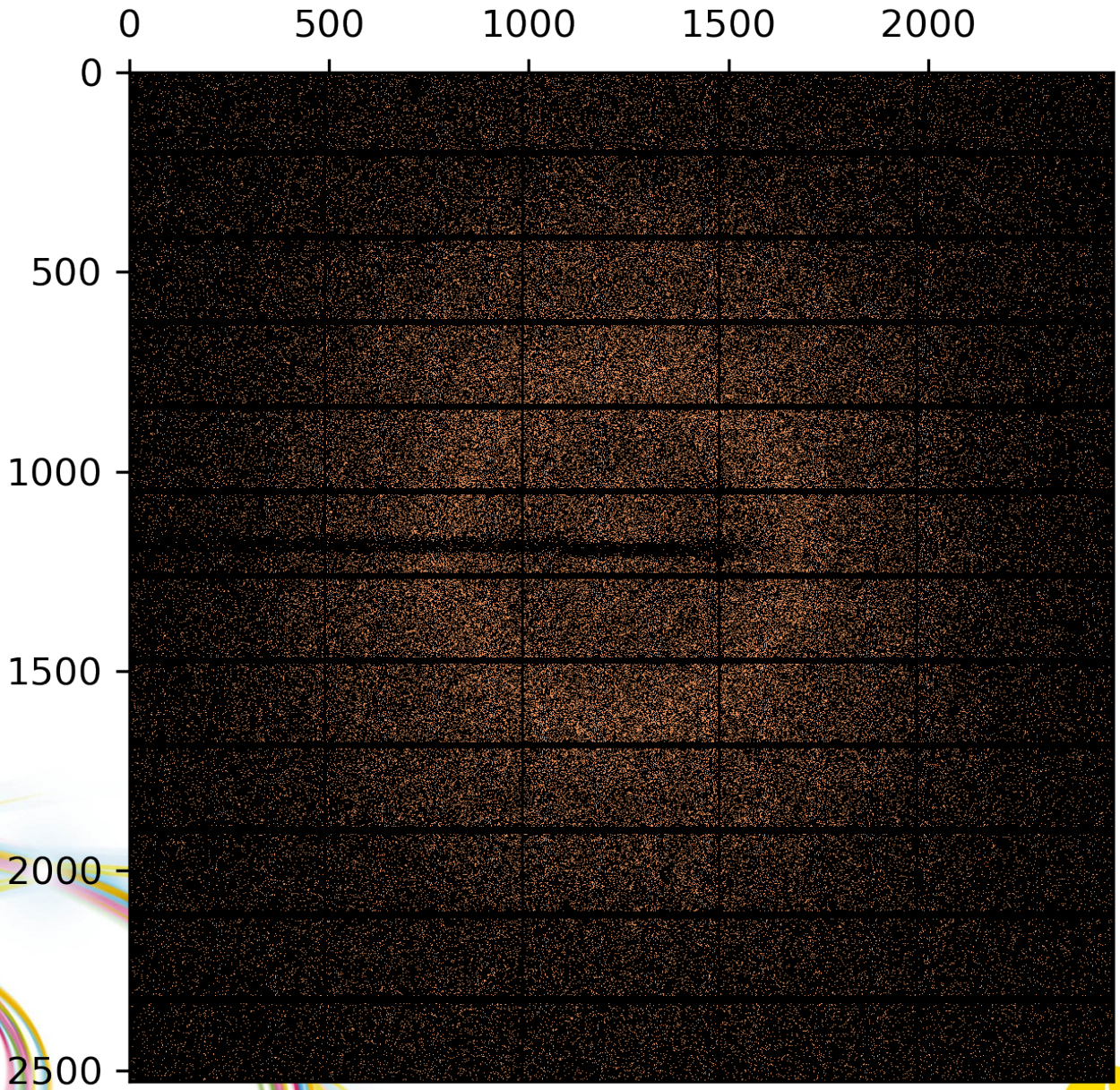
Can we separate the raw data

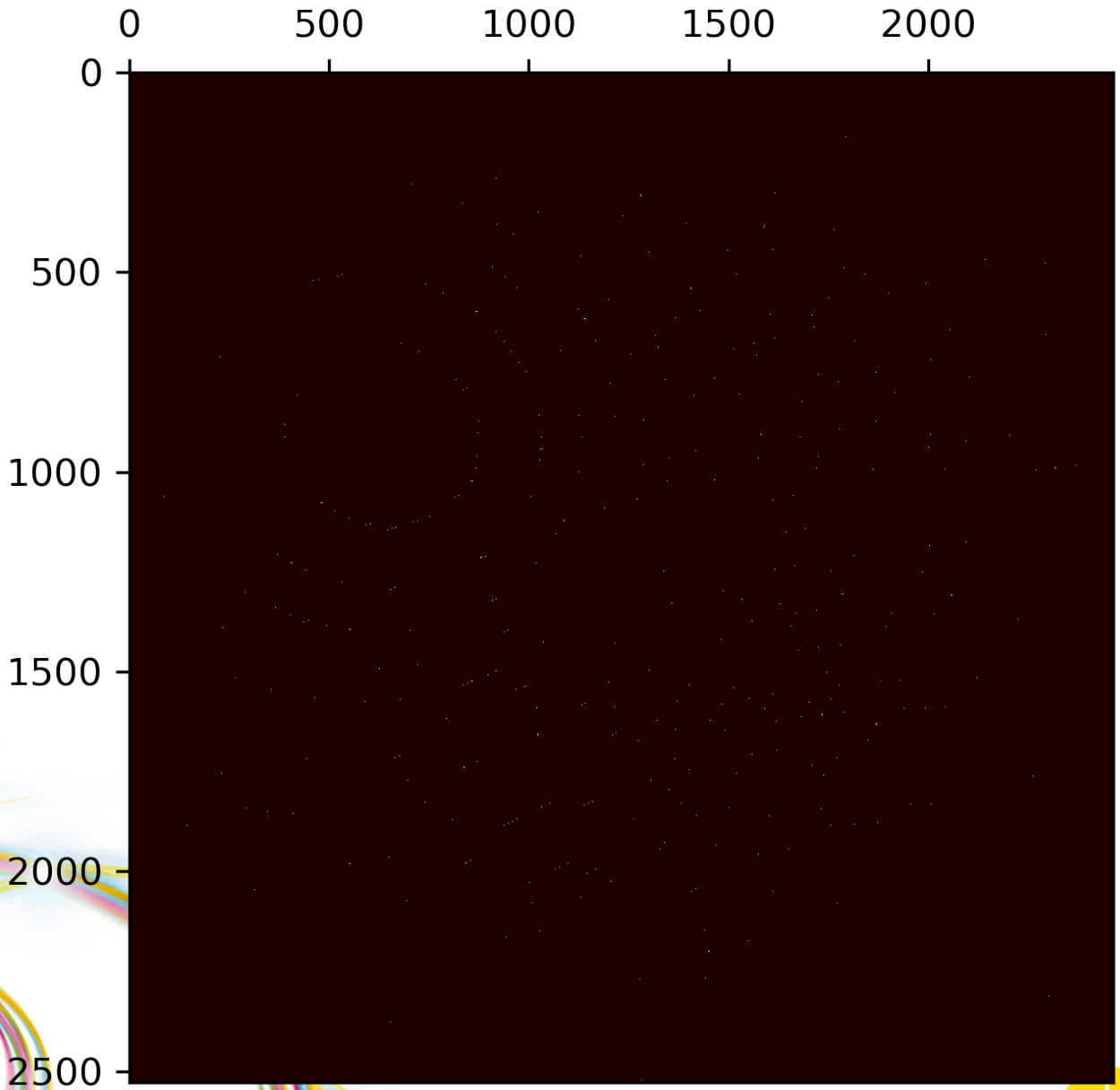
into signal and background?

```
from dxtbx.datablock import DataBlockFactory
from dials.extensions.dispersion_spotfinder_threshold_ext\
    import DispersionSpotFinderThresholdExt as finder
from dials.command_line.find_spots import phil_scope

dblock = DataBlockFactory.from_args(['datablock.json'])
imageset = dblock.extract_imagesets()[0]
detector = imageset.get_detector()[0]
trusted = detector.get_trusted_range()

data = imageset.get_raw_data(0)[0]
negative = (data < 0)
hot = (data > int(round(trusted[1])))
bad = negative | hot

data = data.as_double()

thresholder = finder(phil_scope.extract())
peak = thresholder.compute_threshold(data, ~bad)

signal = data.select(peak.iselection())
background = data.select((~bad & ~peak).iselection())
```

diamond




0
500
1000
1500
2000
0
500
1000
1500
2000
2500

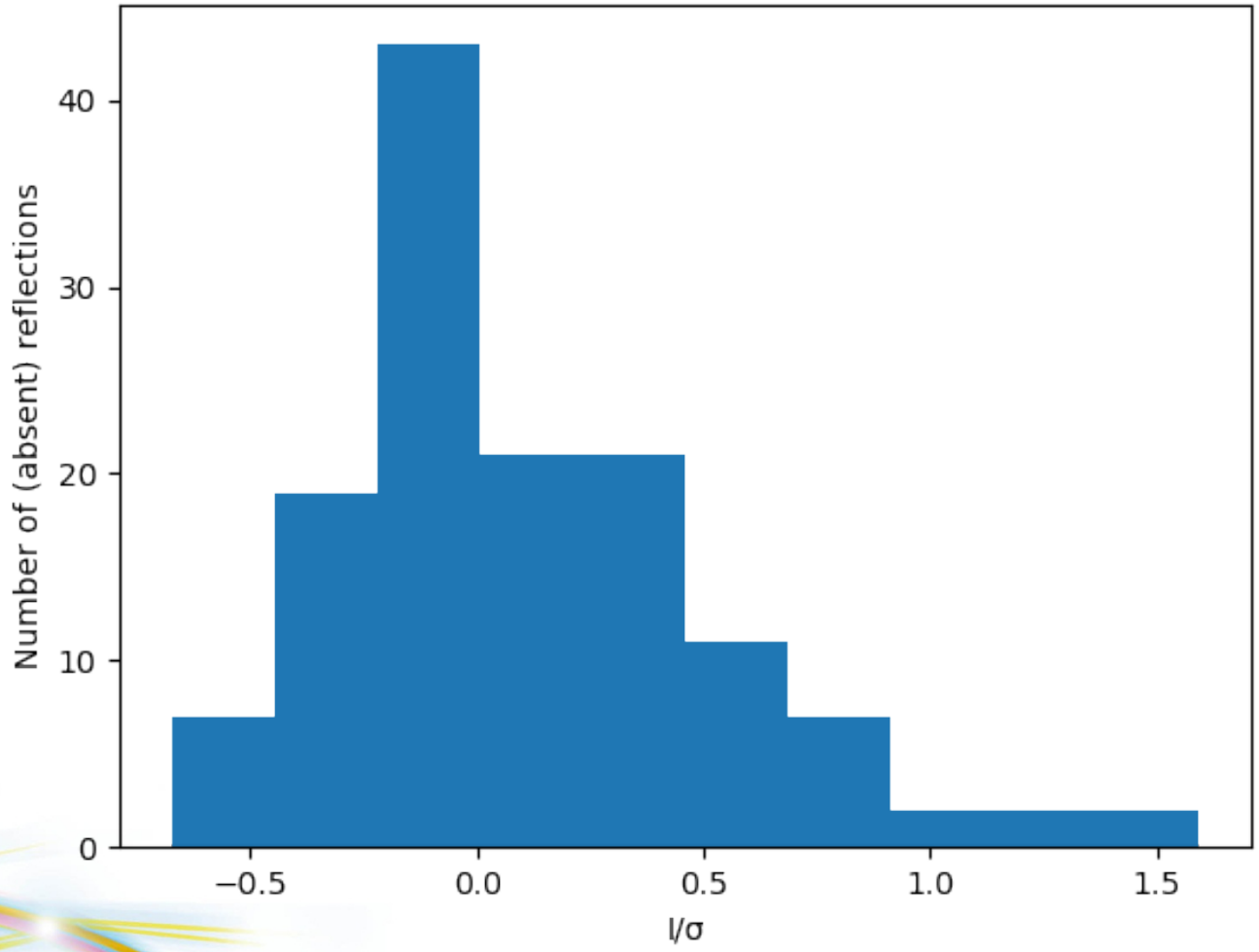

0
500
1000
1500
2000
0
500
1000
1500
2000
2500

Let's look at the integrated intensity

where there is a systematic absence.

diamond

```
import cPickle as pickle
from dials.array_family import flex
from cctbx import sgtbx

sg = sgtbx.space_group_info('P43212').group()

with open('integrated.pickle') as f:
    data = pickle.load(f)

data = data.select(data.get_flags(data.flags.integrated))
absent = data.select(flex.bool(
    [sg.is_sys_absent(m) for m in data['miller_index']]))

i = absent['intensity.prf.value']
s = flex.sqrt(absent['intensity.prf.variance'])

pyplot.hist(i/s)
pyplot.show()
```


Number of (absent) reflections
40
30
20
10
0
−0.5
0.0
0.5
1.0
1.5
I/σ

Go and play!

Documentation:

https://dials.github.io

https://dials.github.io/documentation/library_reference/index.html

The documentation is not perfect!

Please do email us with your questions.