The SExtractor++ package

Emmanuel Bertin and the SE++ development team
SExtractor has been around for too long!
Dealing with large ditherings
Development team

- Alejandro Álvarez Ayllón
- Nikolaos Apostolakos
- Pierre Dubath
- Marc Schefer

- Martin Kümmel
- Emmanuel Bertin
- C++ code
  - Efficiency + abstraction
- Multi-exposure
  - Rely on WCS
- Multi-object
  - Simultaneous model-fitting
- Multichannel
  - Define groups
- Multithreaded
  - One instance per node
- Modular
  - « Property » scheme
  - Plug-ins
Multiframe measurements

Detection image

Weight map

Measurement group #1

Measurement group #2
Photometry comparison with the original SExtractor

Δm vs mag for SE

Δm vs mag for SE++
Flexible model-fitting engine

- Multiple models in one run
- Can combine models
- Can create custom catalog columns
- Uncertainties automatically computed and propagated (from the approximate Hessian of the fit)
- $\chi^2 +$priors (MAP estimator in the Gaussian case)
Proper motions from pixels

```python
from glob import glob
import numpy as np
from sextractorXX.config import *

top = load_fits_images(
    sorted(glob('im_?.??.fits')),
    sorted(glob('im_?.??.psf')),
)

top.split(ByKeyword('FILTER'))

# Split each band by image filename (one image per subgroup)
for n, filter in top:
    filter.split(ByKeyword('IMAGE_FILENAME'))

mesgroup = MeasurementGroup(top)

pmx = FreeParameter(0.0, Range([-100.0, 100.0], RangeType.LINEAR))
py = FreeParameter(0.0, Range([-100.0, 100.0], RangeType.LINEAR))
add_output_column('pmx', pmx)
add_output_column('py', py)

x, y = get_pos_parameters()

# Loop over filters
for filter, group in mesgroup:
    flux = get_flux_parameter()

    # Loop over exposures
    for filename, subgroup in group:
        t = (float(subgroup[0].meta['MJD- OBS']) - 6000.0) / 365.25
        tp = ConstantParameter(t)
        xr = DependentParameter(lambda x, dx, tp: x + tp*dx, x, pmx, tp)
        yr = DependentParameter(lambda y, dy, tp: y + tp*dy, y, py, tp)

        add_model(subgroup, PointSourceModel(xr, yr, flux))
```
Shape priors
Photometric priors
Performance (pure Sersic fits)

PRELIMINARY

<table>
<thead>
<tr>
<th></th>
<th>1 thread</th>
<th>4 threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE 1 image</td>
<td>48.6</td>
<td>0</td>
</tr>
<tr>
<td>SE++ 1 image</td>
<td>23</td>
<td>64.4</td>
</tr>
<tr>
<td>SE++ 30 images</td>
<td>1.6</td>
<td>4.7</td>
</tr>
</tbody>
</table>
The road ahead

• Early alpha release at https://github.com/astrorama/sextactorxx
• Feedback from users
  – Fix the naming issue
  – Tune behavior
• Improve performance
• Compatibility with existing companion packages
• Improve documentation
• High level Python interface
• New detection/deblending module
• More modules
  – Check out Martin’s poster at this conference