Surface photometry tools in Astro-WISE

For the Astro-WISE consortium
Gijs Verdoes Kleijn
OmegaCEN/Kapteyn
Groningen University

GALPHOT in Astro-WISE

- Introduction
- General >
- AW Environment >
- AW Tutorials >
- Calibrations >
- Image Pipeline >
- Analysis Tools >
- Visualization >
- Development >
- FAQ
- Manual (pdf)
- News Mailinglist
- Issues Mailinglist
- Code updates

1.1 *HOW-TO* use Galphot in *Astro-WISE*

1.1.1 Introduction

Galphot is a surface photometry tool, which fits ellipses to isophotes in galaxy profiles. It was written by Marijn Franx and is available for download from his website. The version on his website only works by using IRAF, for *Astro-WISE* a number of changes were done to make it work outside of IRAF.

1.1.2 *Astro-WISE* implementation

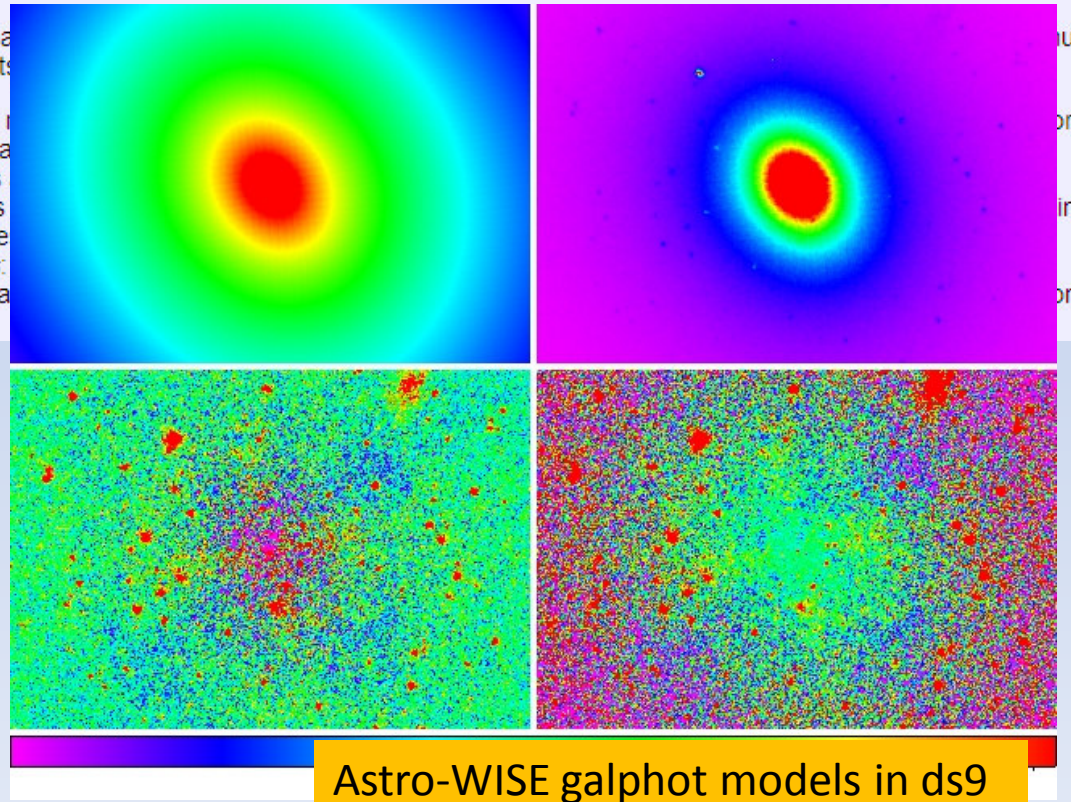
The main classes in the Ga ust
be queried on to get results

- *GalPhotModel*: The r
  on the dataservers a
  model also contains
- *GalPhotEllipse*: This
  list of GalPhotEllipse
- *GalPhotParameters*:
- *GalPhotList*: This cla
  a name.

ored

ins a

onally



Astro-WISE galphot models in ds9

GALFIT

- Introduction
- General >
- AW Environment >
- AW Tutorials >
- Calibrations >
- Image Pipeline >
- Analysis Tools >
- Visualization >
- Development >
- FAQ
- Manual (pdf)
- News Mailinglist
- Issues Mailinglist
- Code updates

1.1 *HOW-TO* use Galfit in *Astro-WISE*

1.1.1 Introduction

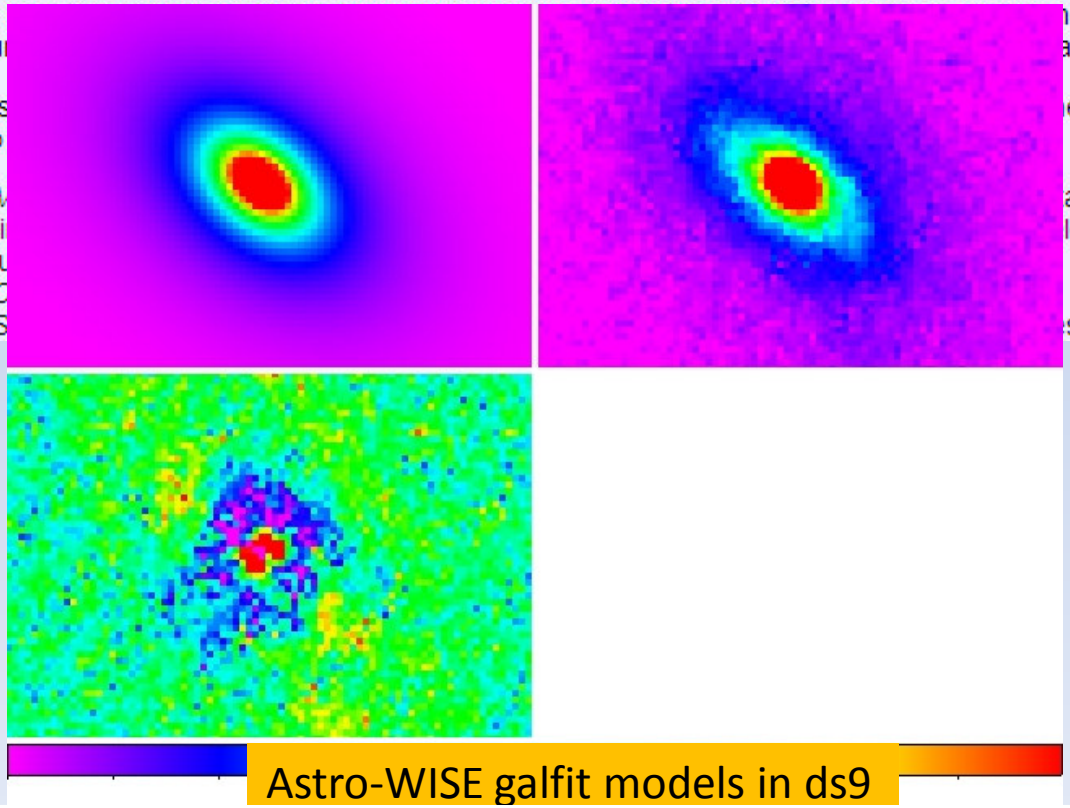
GALFIT is a galaxy/point source fitting algorithm that fits 2-D parameterized, axisymmetric functions directly to images (Peng, Ho, Impey, & Rix 2002, AJ, 124, 266). The program has been developed by Chien Peng who maintains a Galfit homepage.

1.1.2 *Astro-WISE* implementation

The program
wrapper arou
e by providing a *Python*
abase.

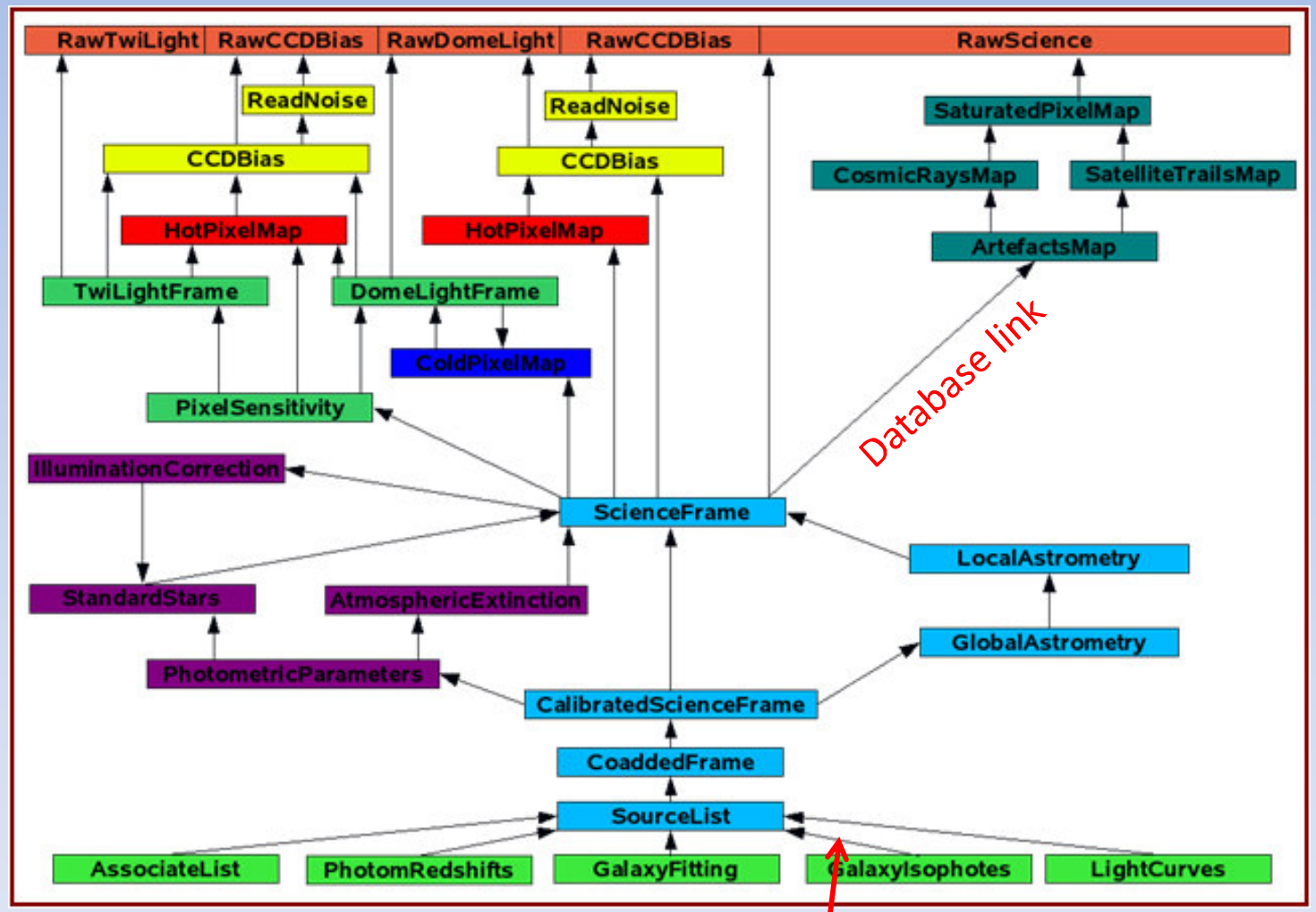
The main clas
queried on to
e database, and must be

- *GalFitM*
  model i
  constitu
  ains methods to create a
  low), which together
- *GalFitC*
- *GalFitS*
  s, and errors of fitted



Astro-WISE galfit models in ds9

Data lineage and source analysis



SourceLink: SourceListID+SourceID (“SLID+SID”)

Data lineage & configuration of GALFIT run

1.1.5 Configuring GalFitModel

In red: info provided by data lineage

While it is possible to configure GalFitModel manually, a configuration is determined automatically when no configuration is specified. The following steps can be distinguished in this process:

- Based on the Sextractor parameters of the source you derive a GalFitModel for, a region is extracted from the larger image the SourceList is made from. The size of the region is related to the semi-major axis of the source (Sextractor "A" parameter).
- Pixelscale and magnitude zeropoint are obtained from the AstrometricParameters resp. SourceList objects.
- By default a Sersic and Sky profile are fit to the modelled source.
- The initial parameters of the specified models are set based on the Sextractor parameters. For a Sersic profile the initial magnitude value (imag) is set to MAG_ISO in the SourceList. Similarly the Xpos and Ypos parameters are used to define the initial position.
- Neighbouring sources that are both close and bright enough to influence the fit are detected and assigned their own Sersic profile.

- The usage of data lineage allows to run with SLID+SID as only input:

1.1.3 Running GalFit

For example to fit a sersic profile plus a sky background to 4 sources from the sourcelist with SLID=57424 which have SID=7,3,9,33 using the cpu on your own machine, enter at the `awe`-prompt:

```
# Run GalFit (see previous example for explanation of parameters):
dpu.run('GalFit', i='WFI', slid=57424, sids=[1,3,9,33], m=[{'name':'sersic'},\
        {'name':'sky'}], C=1)
```

Grouping GalFit runs

- Analogy with SLID+SID: GFLID GFID

The GalFitList class is intended as a simple way to group GalFitModels (and GalFitComponents). The way to use this class is to first create and commit one, and then specify its GalFitList identifier GFLID when running the GalFit task on the DPU or locally:

```
# Create the GalFitList object
l = GalFitList()
l.name = 'test-run-1'
l.make()
l.commit()

# [schmidt] 16:27:12 - Set GalFitList identifier GFLID to 100231

# Refer to the GalFitList object by specifying its GFLID, as reported
# after committing the GalFitList (see above).
dpu.run('GalFit', i='WFI', slid=75637, sids=range(10,20), gflid= l.GFLID    )
```

Data lineage and GALFIT input galaxy selection

```
def select_flaming(self):
    attrlist=self.alrbvmpe.get_attributes_on_associates().
    keys()
    i_b,i_v,i_r=attrlist.index('b'),attrlist.index('v'),at
    trlist.index('r')
    i_zphot=attrlist.index('zphot')
    data = self.alrbvmpe.associates.get_data(attrlist,
    mask=2**0+2**3, mode='INTERSECT')
    self.flaming_sids=([int(data[aid][0][1]) for aid in
    data.keys() if (data[aid][3][i_b]-
    data[aid][3][i_r])>0.8 and (data[aid][3][i_b]-
    data[aid][3][i_r])<1.3 and data[aid][3][i_r]>17.6 and
    data[aid][3][i_r]<19.3 and data[aid][3][i_zphot]>0.28
    and data[aid][3][i_zphot]<0.4])
```

Inspecting GALFIT results

1.1.6 Description of useful methods of GalFitModel

- get_model()

  Creates the model image and returns it as a BaseFrame object.

- get_residual()

  Creates the residual image and returns it as a BaseFrame object.

- get_science()

  Extracts and downloads the region in the science image for which the model was derived, and returns it as a BaseFrame object.

- get_weight()

  Extracts and downloads the region in the weight image for which the model was derived and returns it as a BaseFrame object.

- show_model_parameters()

  Display a list of all ellipse parameters.

- get_model_parameters()

  Returns a list of dictionaries of all components. I.e. each item of the list is a dictionary which contains the

Inspecting GALFIT results

1.1.6 Description of useful methods of GalFitModel

- get_model()

  Creates the model image and retur

- get_residual()

  Creates the residual image and retu

- get_science()

  Extracts and downloads the region i
  BaseFrame object.

- get_weight()

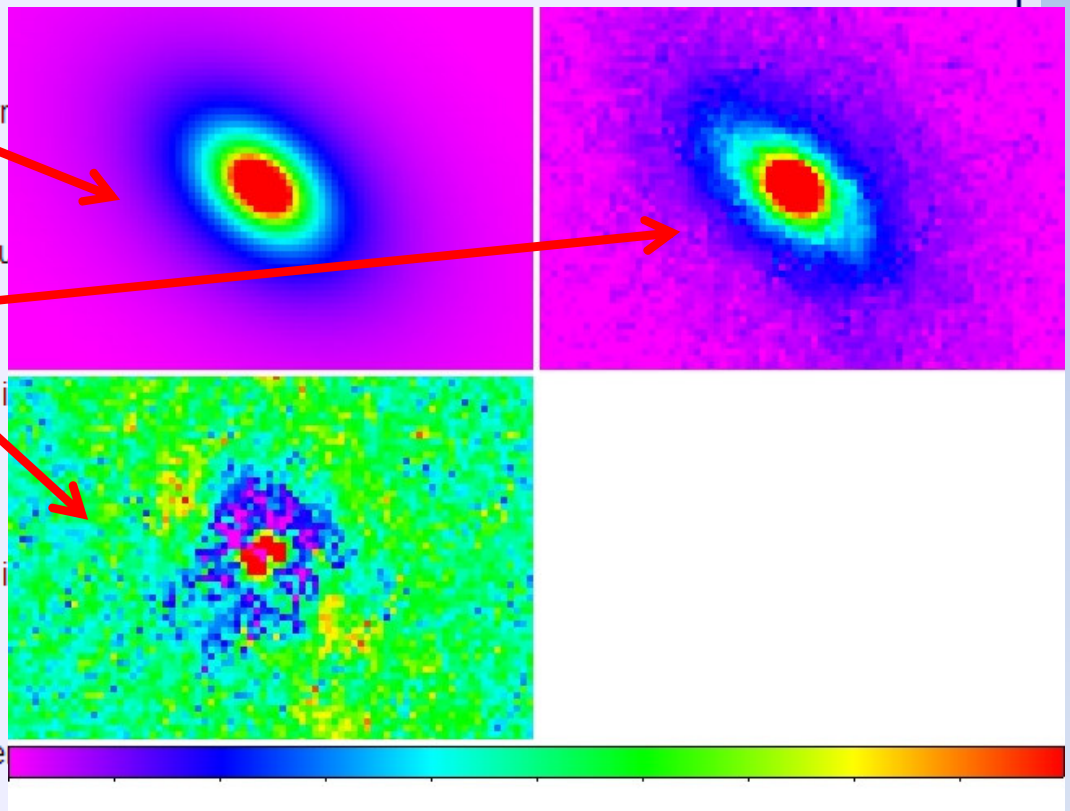
  Extracts and downloads the region i
  BaseFrame object.

- show_model_parameters()

  Display a list of all ellipse paramete

- get_model_parameters()

  Returns a list of dictionaries of all components. I.e. each item of the list is a dictionary which contains the



Data lineage and querying for GALFIT results

1.1.4 Querying the database for GalFitModel results

```
# Query the database for the GalFitComponents which contain the results
# for those sources in SourceList with SLID=57424 which
# were fitted with a sersic profile:
query = (GalFitComponent.name == 'sersic') & (GalFitComponent.SLID==57424)
```

```
awe>
awe> qgfs=(GalFitSersic.SLID!=63656) & (GalFitSersic.iN==2.5) & (GalFitSer
sic.free_N==0)
awe>
```

Next steps

- Source analysis tools
  - Data lineage: splinter on SourceLink method consistently (Ewout)
  - More tools for surveys?: collaborate with survey teams
- Create user-contributed corner in Astro-WISE
  - Re-ingest galphot/galfit subtracted images Verdoes/Bos
  - Exports to aladin/topcat (Buddelmeijer/Belikov)
  - Relative astrometry Koppenhoefer
  - etc….