Point Source Photometry (Using HIPE)

Kevin Xu
NHSC/IPAC
on behalf of the SPIRE ICC

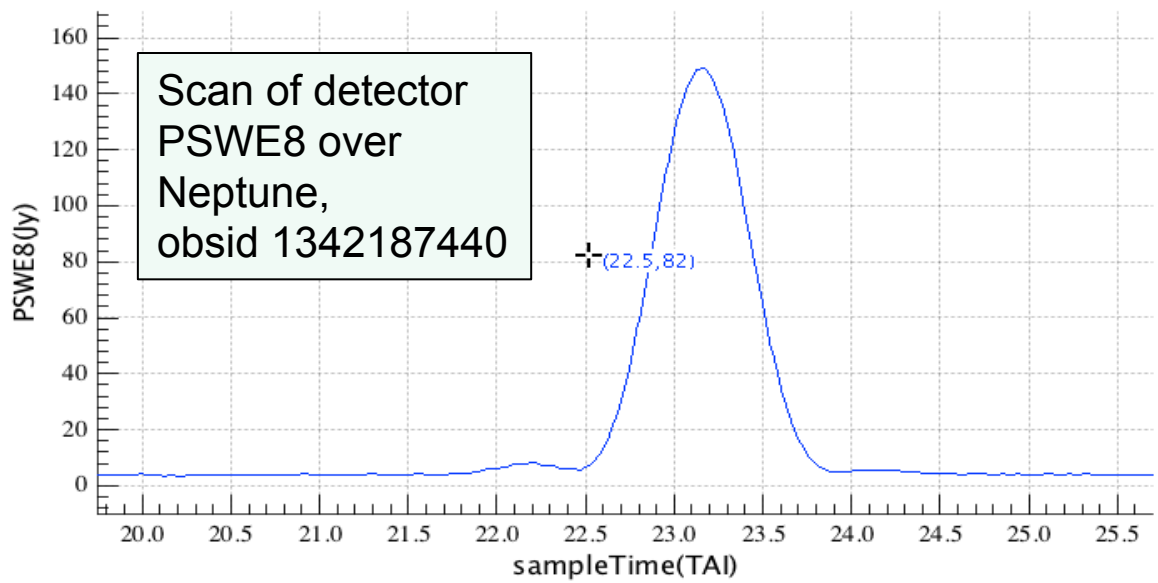
Outline

- Timeline (Level-1) Fitting (Recommended)
- Aperture Photometry on Maps (Level 2, 2.5, 3)
- Source Detection & Photometry in Blank Fields
  - HIPE Task: sourceExtractorSussextactor
  - HIPE Task: sourceExtractorDaophot

Reference: *Section 5.7, "SPIRE Data Reduction Guide"*

SPIRE Point Source Calibration

- **Standard Calibration Source**: Neptune
- **Method**: calibration observations (in fine-scan mode) are so designed as every bolometer can see the source multiple times and measure the deflection of the source with better than adequate sampling. The calibration is then defined by relating the deflection peak to the source flux.
- **Units:** Jy/Beam
- **Error:** 6%
  - 2% random error
  - 4% model error



Timeline Fitting (Recommended for Point Source Photometry)

- **Inputs**:
  - Level 1 timelines (consist of measurements of individual bolometers before map-making)
  - Initial guess of the coordinates of the source
- **Outputs:** source flux & position (best-fit)
- **Algorithm:** fitting individual measurements near the source position with a 2-d Gaussian function (a good match to the inner part of the PSF)
- **Advantage** over photometry using maps: avoid biases introduced by map-making (e.g. pixelization)
- **HIPE Task:** sourceExtractorTimeline

sourceExtractorTimeline

Editor ×

Jython script

*test_AperPhotoDemo.py × | *test_time...itting.py ×

```
# Example:  NGC5313 (a Planetary Nebula)
myObsid     =  1342183475L                    # Observation ID (OBSID)
myDataPool = 's_1342183475'                     # Name of data Pool
#  Load in an observation context from your data pool into HIPE:
obs = getObservation(myObsid, poolName=myDataPool)
################  demo for sourceExtractorTimeline ################
# Get level-1 timeines (already destriped):
level1 = obs.level1
# (2) source position:
ra0=208.4868
dec0=-66.5137
# (3) execute the task (for PLW flux):
result = sourceExtractorTimeline(input = level1, array = 'PLW', \
        inputSourceList =[ra0, dec0], rPeak = 42.0)
# (4) Output is in result["sources"]:
flux = result["sources"]["flux"].data
fluxErr = result["sources"]["fluxPlusErr"].data
print flux, fluxErr
# [631.6781081949591] [6.106946242452472]  in mJy
ra = result["sources"]["ra"].data
raErr = result["sources"]["raPlusErr"].data
print ra, raErr
#[208.4868100990549] [9.782115566425004E-5]
dec = result["sources"]["dec"].data
decErr = result["sources"]["decPlusErr"].data
print dec, decErr
#[-66.51377665089737] [3.874234695941744E-5]
ki2 = result["sources"]["reducedChiSquare"].data
print ki2
#  [0.002625954064697422] nearly perfect fit!!
```

Radius (“) for Gaussian fitting

default values:
PLW: 42
PMW: 30
PSW: 22

sourceExtractorTimeline: GUI

Editor

*test_AperPhotoDemo.py | *test_time...itting.py

```
# Example:  NGC5313 (a Planetary Nebula)
myObsid     =  1342183475L                    # Observation ID (OBSID)
myDataPool = 's_1342183475'                     # Name of data Pool
#  Load in an observation context from your data pool into HIPE:
obs = getObservation(myObsid, poolName=myDataPool)
################  demo for sourceExtractorTimeline ################
# Get level-1 timeines (already destriped):
level1 = obs.level1
# (2) source position:
ra0=208.4868
dec0=-66.5137
# (3) execute the task (for PLW flux):
result = sourceExtractorTimeline(input = level1, array = 'PLW', \
        inputSourceList =[ra0, dec0], rPeak = 42.0)
# (4) Output is in result["sources"]:
flux = result["sources"]["flux"].data
fluxErr = result["sources"]["fluxPlusErr"].data
print flux, fluxErr
# [631.6781081949591] [6.106946242452472]  in mJy
ra = result["sources"]["ra"].data
raErr = result["sources"]["raPlusErr"].data
print ra, raErr
#[208.4868100990549] [9.782115566425004E-5]
dec = result["sources"]["dec"].data
decErr = result["sources"]["decPlusErr"].data
print dec, decErr
#[-66.51377665089737] [3.874234695941744E-5]
ki2 = result["sources"]["reducedChiSquare"].data
print ki2
#  [0.002625954064697422] nearly perfect fit!!
```

Variab...

Observations

Other Variables

annularPLW
basedir
dec
dec0
decErr
flux
flux_aper
flux_tot
fluxErr
ki2
level1
mapPlw1
mapPlwJyPix1
myDataPool
myObsid
plugin
pluginRegistry
ra
ra0
raErr
result
sys
toolRegistry
ver

Outline

Tasks

Applicable

baselineRemovalMedian
baselineRemovalPolynomi
convolutionMapper
createTodBuffer
destriper
madScanMapper
naiveScanMapper
overwriteWithTodBuffer
qualAssessSpireLevel05
qualAssessSpireLevel1
qualAssessSpireLevel2
saveProduct
sourceExtractorTimeline
sourceSubtractor

By Category

All

sourceExtractorTimeline: GUI

Mandatory inputs

Editor

DTV for le...].product | *New-1 | sourceExtr...rTimeline

Inputs

input*:	level1	array*:	PLW
rPeak*:	42.0	inputSourceList*:	inputSourceList
rBackground:	<Default>	useBackInFit:	
allowVaryBackground:		allowTiltBack:	
fitEllipticalGauss2d:		modelGauss2dSigma:	-1.0
fitMaxIterations:	10000	fitTolerance:	1.0E-4
slow:		roi:	<No variable>

Outputs

Variable name for sourceList: result1

Execution Status

Source | Help | Clear | Accept

History | Log | Console

Variab...

Observations
- obs

Other Variables
- array
- basedir
- dec
- decErr0
- decErr1
- flux
- fluxErr0
- fluxErr1
- inputSourceList
- ki2
- level1
- myDataPool
- myObsid
- plugin
- pluginRegistry
- ra
- raErr0

Tasks

Applicable
- baselineRemovalMedian
- baselineRemovalPolynom
- convolutionMapper
- createTodBuffer
- destriper
- madScanMapper
- metaDataSorter
- naiveScanMapper
- overwriteWithTodBuffer
- qualAssessSpireLevel05
- qualAssessSpireLevel1
- qualAssessSpireLevel2
- saveProduct
- sourceExtractorTimeline
- sourceSubtractor

By Category

All

sourceExtractorTimeline: GUI

Result (a product):

Editor

*test_AperPhotoDemo.py | *test_time...itting.py | sourceExtr...rTimeline | result1

SourceListDataset

Meta Data

name	value	unit	description

Data

result1
- sources
- History

result1["sources"]

Index	ra [deg]	dec [deg]	raPlusErr...	decPlusErr...	...	...	flux [mJy]	fluxPlusErr...
0	208.4868...	-66.5137...	9.782115...	3.874234695...	...	...	631.6781081...	6.1069462...

Variab...

Observations

Other Variables
- plugin
- pluginRegistry
- ra
- ra0
- raErr
- result
- result1
- sourceList
- sys
- toolRegistry

Outline

Name	result1["sou
Class	SourceListD
Package	herschel.ia.

Reminders:

- Flux error: should add (quadratically) the calibration error (6%).
- Flux: color correction (for spectra not in the form of $f_\nu \sim \nu^{-1}$).

Color corrections are described in the SPIRE Data Reduction Guide (Section 5.7.1.6)

- The flux (to be corrected) should be multiplied by the correction factor found in the table.

Color correction table

Spectral Index ($F_{nu}=nu^{alpha}$)	PSW Correction	PMW Correction	PLW Correction
-4	0.94071	0.94049	0.87029
-3.5	0.95492	0.95471	0.89817
-3	0.96753	0.96734	0.92392
-2.5	0.97844	0.97827	0.94725
-2	0.98755	0.98741	0.96787
-1.5	0.99476	0.99468	0.98553
-1	1.00000	1.00000	1.00000
-0.5	1.00321	1.00332	1.01110
0	1.00434	1.00459	1.01868
0.5	1.00337	1.00380	1.02266
1	1.00028	1.00093	1.02299
1.5	0.99507	0.99599	1.01969
2	0.98777	0.98900	1.01282
2.5	0.97843	0.98002	1.00249
3	0.96710	0.96910	0.98887
3.5	0.95385	0.95632	0.97215
4	0.93878	0.94176	0.95257
4.5	0.92198	0.92553	0.93040
5	0.90359	0.90773	0.90592

Checklist for timeline-fitting photometry

- ❑ Use the Level 1 timelines
- ❑ Supply a starting position
- ❑ Run the sourceExtractorTimeline Task
- ❑ Apply color correction as needed

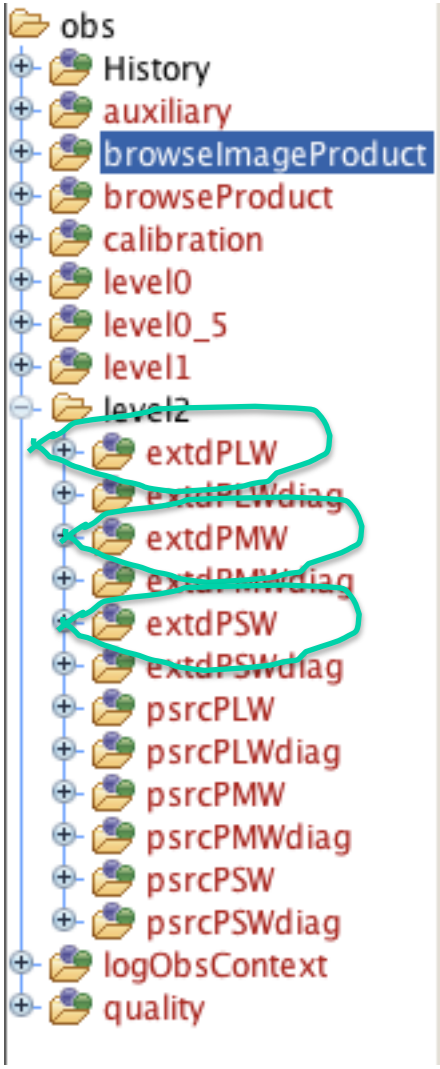
Outline

- Timeline (Level-1) Fitting
  - SPIRE Point Source Calibration
  - HIPE Task: sourceExtractorTimeline
- Aperture Photometry on Maps (Level 2, 2.5, 3)
  - SPIRE Extended Maps (extdPxW)
  - Hipe Task: annularSkyAperturePhotometry
- Source Detection & Photometry in Blank Fields
  - HIPE Task: sourceExtractorSussextactor
  - HIPE Task: sourceExtractorDaophot

Aperture Photometry on Maps

- Applicable to maps in level 2, level 2.5, and level 3
  - Always use extended maps (e.g. extdPLW)

  (this also applies when using your own aperture photometry tools)
- extended (extd) maps vs. point-source (psrc) maps:
  - Units: MJy/sr vs. Jy/Beam
  - Flux calibration: scaling with the total counts of a source in the entire beam vs. scaling with the peak of the deflection of the source (more details will be duscussed in the next talk on extended source photometry)
- HIPE task: annularSkyAperturePhotometry



annularSkyAperturePhotometry

same obs used in the timeline fitting demo

Editor ×

Photometer...Finder.py × | *demo_aperPhot.py ×

```
29  ################  demo for annularSkyAperturePhotomery #######
30  # (1) get the extd map (PLW) of the source:
31  mapPlw = obs.level2.getProduct("extdPLW")
32  # (2) convert units from MJy/sr to Jy/pixel (important)!
33  mapPlwJyPix = convertImageUnit(image=mapPlw, newUnit='Jy/pixel')
34  # source position:
35  ra0='208.4868'
36  dec0='-66.5137'
37  # aperture photometry, aperture =42.0
38  annularPLW = annularSkyAperturePhotometry(image= mapPlwJyPix, \
39                 centerRA=ra0, \
40                 centerDec=dec0, fractional=1, radiusArcsec=42.0, \
41                 innerArcsec=60.0, outerArcsec=90.0)
42  # total flux:
43  flux_tot = annularPLW.getTargetTotal()
44  # new aperture correction: PLW, PMW, PSW: 1.226,
45  flux_aper = flux_tot * 1.226
46  print flux_aper
47  # 0.574700348308 Jy, compared to 631.7 mJy from timelinefitting.
```

The difference is ~ 9% between timeline fitting and aperture photometry. Large errors in aperture correction?

annularSkyAperturePhotometry: GUI

Editor

*test_AperPhotoDemo.py | *test_time...itting.py

```
29  ##################  demo for annularSkyAperturePhotomery #####
30  # (1) get the extd map (PLW) of the source:
31  mapPlw = obs.level2.getProduct("extdPLW")
32  # (2) convert units from MJy/sr to Jy/pixel (important)!
33  mapPlwJyPix = convertImageUnit(image=mapPlw, newUnit='Jy/pixel')
34  # source position:
35  ra0='208.4868'
36  dec0='-66.5137'
37  # aperture photometry, aperture =42.0
38  annularPLW = annularSkyAperturePhotometry(image= mapPlwJyPix, \
39                 centerRA=ra0, \
40                 centerDec=dec0, fractional=1, radiusArcsec=42.0, \
41                 innerArcsec=60.0, outerArcsec=90.0)
42  # total flux:
43  flux_tot = annularPLW.getTargetTotal()
44  # aperture correction: PLW, PMW, PSW: 1.2750, 1.253, 1.295
45  flux_aper = flux_tot * 1.275
46  print flux_aper
47  # 0.597669611821 Jy, compared to 631.7 mJy from timelinefitting.
```

Variab...

Observations

Other Variables

- flux_aper
- flux_tot
- fluxErr
- inputSourceList
- ki2
- level1
- mapPlw
- mapPlw1
- mapPlwJyPix
- mapPlwJyPix1
- myDataPool

Outline

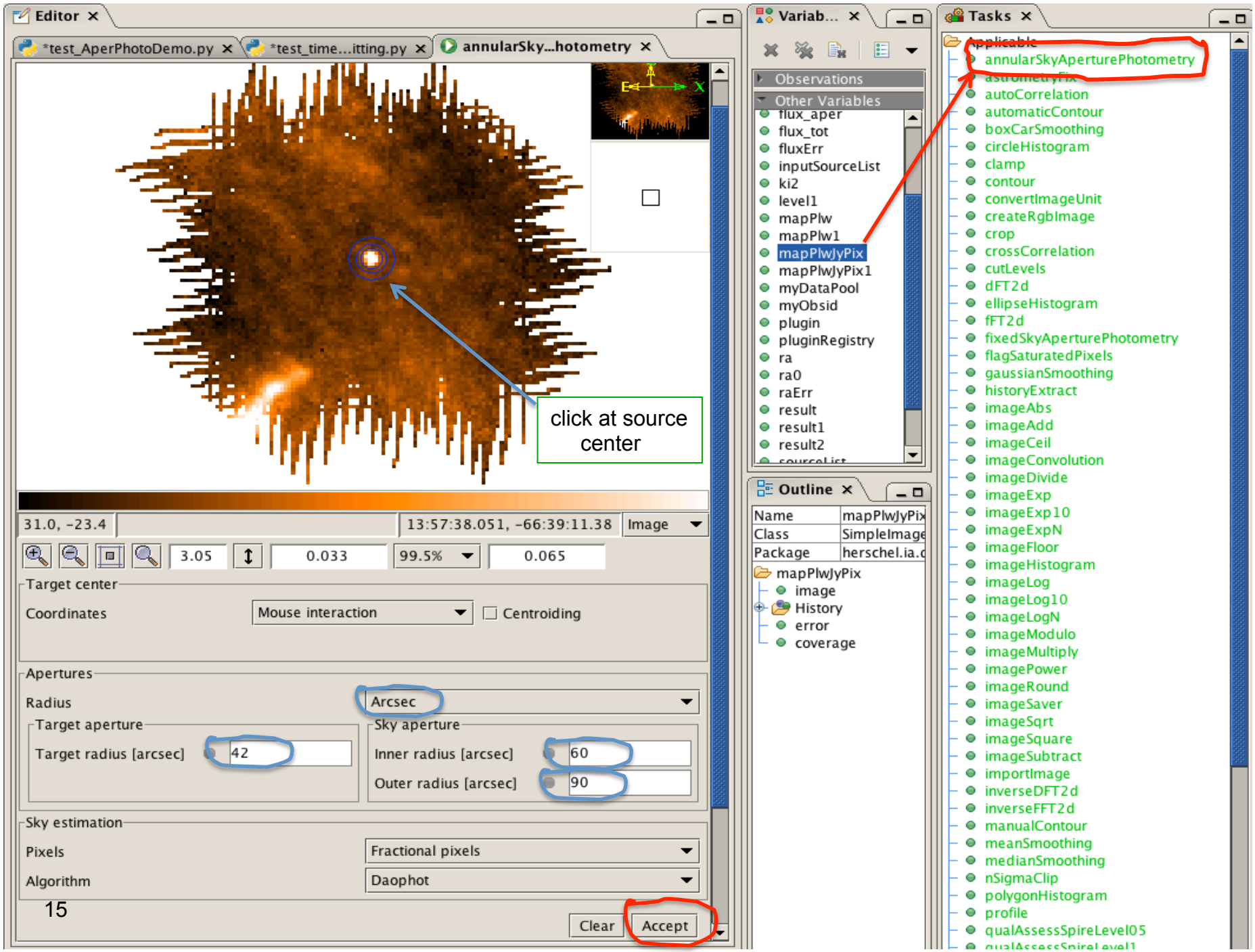
Name	mapPlwJyPix
Class	SimpleImage
Package	herschel.ia.c

mapPlwJyPix

Tasks

Applicable

- annularSkyAperturePhotometry
- astrometryFix
- autoCorrelation
- automaticContour
- boxCarSmoothing
- circleHistogram
- clamp
- contour
- convertImageUnit
- createRgbImage
- crop
- crossCorrelation
- cutLevels
- dFT2d
- ellipseHistogram
- fFT2d
- fixedSkyAperturePhotometry
- flagSaturatedPixels
- gaussianSmoothing
- historyExtract
- imageAbs


Editor
*test_AperPhotoDemo.py
*test_time...itting.py
annularSky...hotometry
click at source center
31.0, -23.4
13:57:38.051, -66:39:11.38
Image
3.05
0.033
99.5%
0.065
Target center
Coordinates
Mouse interaction
Centroiding
Apertures
Radius
Arcsec
Target aperture
Target radius [arcsec]
42
Sky aperture
Inner radius [arcsec]
60
Outer radius [arcsec]
90
Sky estimation
Pixels
Fractional pixels
Algorithm
Daophot
Clear
Accept
Variab...
Observations
Other Variables
flux_aper
flux_tot
fluxErr
inputSourceList
ki2
level1
mapPlw
mapPlw1
mapPlwJyPix
mapPlwJyPix1
myDataPool
myObsid
plugin
pluginRegistry
ra
ra0
raErr
result
result1
result2
Outline
Name
mapPlwJyPix
Class
SimpleImage
Package
herschel.ia.c
mapPlwJyPix
image
History
error
coverage
Tasks
Applicable
annularSkyAperturePhotometry
autoCorrelation
automaticContour
boxCarSmoothing
circleHistogram
clamp
contour
convertImageUnit
createRgbImage
crop
crossCorrelation
cutLevels
dFT2d
ellipseHistogram
fFT2d
fixedSkyAperturePhotometry
flagSaturatedPixels
gaussianSmoothing
historyExtract
imageAbs
imageAdd
imageCeil
imageConvolution
imageDivide
imageExp
imageExp10
imageExpN
imageFloor
imageHistogram
imageLog
imageLog10
imageLogN
imageModulo
imageMultiply
imagePower
imageRound
imageSaver
imageSqrt
imageSquare
imageSubtract
importImage
inverseDFT2d
inverseFFT2d
manualContour
meanSmoothing
medianSmoothing
nSigmaClip
polygonHistogram
profile
qualAssessSpireLevel05

annularSkyAperturePhotometry: Result

Editor

*test_time...itting.py | annularSky...hotometry | result2

Parameters

Key	Value
Target center (x,y)	(70.604, 67.655)
Target center (RA,Dec)	(13:53:56.534, -66:30:47.82)
Target radius [pixels]	2.9999999999999143
Target radius [arcsec]	42.00
Inner radius [pixels]	4.29
Inner radius [arcsec]	60.00
Outer radius [pixels]	6.43
Outer radius [arcsec]	90.00
Sky estimation algorithm	Daophot
Pixel type	Fractional
Intensity unit	Jy/pixel

Results table

	Total [Jy]	# pixels	Per pixel [Jy...	Error [Jy]
Target	1.582732e+00	2.827433e+01	5.597770e-02	1.258067e+00
Background	2.848423e+00	7.212840e+01	3.943329e-02	1.046895e-03
Target (bg subtr)	4.677820e-01	2.827433e+01	1.654440e-02	6.839693e-01

Variab...

Observations

Other Variables

- ki2
- level1
- mapPlw
- mapPlw1
- mapPlwJyPix
- mapPlwJyPix1
- myDataPool
- myObsid
- plugin
- pluginRegistry
- ra
- ra0
- raErr
- result
- result1
- result2
- sourceList
- sys
- toolRegistry
- ver

Outline

flux, need to do aperture correction

Don't trust this error!!!!

Checklist for aperture photometry

- ❑ Use the level 2 (or 2.5/3) extd maps
- ❑ Convert to Jy/pixel using "convertImageUnit" task
- ❑ Run the annularSkyAperturePhotometry task
- ❑ Apply an appropriate aperture correction
- ❑ Apply color correction as needed

Outline

- Timeline (Level-1) Fitting
  - SPIRE Point Source Calibration
  - HIPE Task: sourceExtractorTimeline
- Aperture Photometry on Maps (Level 2, 2.5, 3)
  - SPIRE Extended Maps (extdPxW)
  - Hipe Task: annularSkyAperturePhotometry
- Source Detection & Photometry in Blank Fields
  - HIPE Task: sourceExtractorSussextactor
  - HIPE Task: sourceExtractorDaophot

Two Source Extractors in HIPE

- sourceExtractorSussextractor
  - Sussextractor algorithm (Savage & Oliver 2007)
  - A Bayesian algorithm for detection
  - Flux density is peak of smoothed image
- sourceExtractorDaophot
  - Algorithms from IDL AstroLib
  - FIND for detection
  - APER for photometry (not PSF-fitting!)

The simplest operation of extractors uses the FWHM of each band

- Averages for nominal pixels:
  - 250um: 17.6 arcsec
  - 350um: 23.9 arcsec
  - 500um: 35.2 arcsec
- The fine-scale beam areas are needed (or the images must be converted)
  - (465, 822, 1769) sq arcsec

(can be found in *Section 5.7, SPIRE Data Reduction Guide*)

sourceExtractorSussextractor

Editor

result2 | *demo_daophot.py | *demo_susextractor.py | *New-1

```
cal = spireCal(pool="spire_cal_11_0")
##################  demo for sourceExtractorSussextractor ######
# (1) get the point source map (PLW) of the source:
mapPlw = obs.level2.getProduct("psrcPLW")
# (2) set the parameters:
thr = 5.                          # Detection Threshold
# FWHM for PSW, PMW and PLW arrays (for 1 arcsec pixels)
fwhm = [17.6, 23.9, 35.2]
# Beam area for 1 arcsec pixel maps for alpha =-1 spectral index
beam = [465., 822., 1769.]
# run SUSSEXtractor (Gaussian PRF):
srcListPLWSussex = sourceExtractorSussextractor( \
  image=mapPlw, detThreshold=thr, fwhm=fwhm[2], beamArea=beam[2])
# result (no aperture correction needed):
flux_sussex = srcListPLWSussex["sources"]["flux"].data
print flux_sussex[0]
# 606.189808204 mJy , compared to 631.7 mJy from timelinefitting.
fluxErr_sussex = srcListPLWSussex["sources"]["fluxPlusErr"].data
print fluxErr_sussex[0]
# 8.99215021117
```

~ 4% difference

sourceExtractorSussextractor

Editor

result2 | *demo_daophot.py | *demo_susextractor.py | *New-1

```
cal = spireCal(pool="spire_cal_11_0")
##################  demo for sourceExtractorSussextractor ######
# (1) get the point source map (PLW) of the source:
mapPlw = obs.level2.getProduct("psrcPLW")
# (2) set the parameters:
thr = 5.                          # Detection Threshold
# FWHM for PSW, PMW and PLW arrays (for 1 arcsec pixels)
fwhm = [17.6, 23.9, 35.2]
# Beam area for 1 arcsec pixel maps for alpha =-1 spectral index
beam = [465., 822., 1769.]
# run SUSSEXtractor (Gaussian PRF):
srcListPLWSussex = sourceExtractorSussextractor( \
  image=mapPlw, detThreshold=thr, fwhm=fwhm[2], beamArea=beam[2])
# result (no aperture correction needed):
flux_sussex = srcListPLWSussex["sources"]["flux"].data
print flux_sussex[0]
# 606.189808204 mJy , compared to 631.7 mJy from timelinefitting.
fluxErr_sussex = srcListPLWSussex["sources"]["fluxPlusErr"].data
print fluxErr_sussex[0]
# 8.99215021117
```

Variab...

Observations

Other Variables

level1
mapPlw
mapPlw1
mapPlwJyPix
mapPlwJyPix1

Outline

Name	mapPlw
Class	SimpleImage
Package	herschel.ia.c

mapPlw
- image
- error
- coverage

History

Tasks

- manualContour
- meanSmoothing
- medianSmoothing
- nSigmaClip
- polygonHistogram
- profile
- qualAssessSpireLevel05
- qualAssessSpireLevel1
- qualAssessSpireLevel2
- rectangleHistogram
- rectangularSkyAperturePhotomet
- regrid
- rotate
- saveProduct
- scale
- simpleFitsWriter
- sourceExtractor
- sourceExtractorDaophot
- sourceExtractorSimultaneous
- sourceExtractorSussextractor
- sourceFitting
- tiledImage

sourceExtractorDaophot

Editor

annularSky...hotometry | result2 | *demo_daophot.py | *demo_suse

```
cal = spireCal(pool="spire_cal_11_0")
##################  demo for sourceExtractorDaophot ##############
# (1) get the point source map (PLW) of the source:
mapPlw = obs.level2.getProduct("psrcPLW")
# (2) set the parameters:
thr = 5.                          # Detection Threshold
# FWHM for PSW, PMW and PLW arrays (for 1 arcsec pixels)
fwhm = [17.6, 23.9, 35.2]
# Beam area for 1 arcsec pixel maps for alpha =-1 spectral index
beam = [465., 822., 1769.]
# run DAOphot:
srcListPLWDaoPhot = sourceExtractorDaophot( \
            image=mapPlw, detThreshold=thr, \
             fwhm=fwhm[2], beamArea=beam[2])
#
# result (including the daophot aperture correction as default):
flux_daophot = srcListPLWDaoPhot["sources"]["flux"].data
print flux_daophot[0]
# 582.488863215 mJy , compared to 631.7 mJy from timelinefitting.
fluxErr_daophot = srcListPLWDaoPhot["sources"]["fluxPlusErr"].data
print fluxErr_daophot[0]
#13.3104082296 mJy
```

~ 8 % difference

sussexExtractor:
flux = 603 mJy

sourceExtractorDaophot

Editor

annularSky...hotometry | result2 | *demo_daophot.py | *demo_suse

```
cal = spireCal(pool="spire_cal_11_0")
##################  demo for sourceExtractorDaophot #############
# (1) get the point source map (PLW) of the source:
mapPlw = obs.level2.getProduct("psrcPLW")
# (2) set the parameters:
thr = 5.                          # Detection Threshold
# FWHM for PSW, PMW and PLW arrays (for 1 arcsec pixels)
fwhm = [17.6, 23.9, 35.2]
# Beam area for 1 arcsec pixel maps for alpha =-1 spectral index
beam = [465., 822., 1769.]
# run DAOphot:
srcListPLWDaoPhot = sourceExtractorDaophot( \
            image=mapPlw, detThreshold=thr, \
             fwhm=fwhm[2], beamArea=beam[2])
#
# result (including the daophot aperture correction as default):
flux_daophot = srcListPLWDaoPhot["sources"]["flux"].data
print flux_daophot[0]
# 582.488863215 mJy , compared to 631.7 mJy from timelinefitting.
fluxErr_daophot = srcListPLWDaoPhot["sources"]["fluxPlusErr"].data
print fluxErr_daophot[0]
#13.3104082296 mJy
```

Variab...

Observations

Other Variables

- inputSourceList
- ki2
- level1
- mapPlw
- mapPlw1
- mapPlwJyPix
- mapPlwJyPix1

Outline

Name	mapPlw
Class	SimpleImage
Package	herschel.ia.c

- mapPlw
  - image
  - error
  - coverage
- History

Tasks

- importImage
- inverseDFT2d
- inverseFFT2d
- manualContour
- meanSmoothing
- medianSmoothing
- nSigmaClip
- polygonHistogram
- profile
- qualAssessSpireLevel05
- qualAssessSpireLevel1
- qualAssessSpireLevel2
- rectangleHistogram
- rectangularSkyAperturePhotor
- regrid
- rotate
- saveProduct
- scale
- simpleFitsWriter
- sourceExtractor
- sourceExtractorDaophot
- sourceExtractorSimultaneous
- sourceExtractorSussextractor
- sourceFitting
- tiledImage

Summary

- Timeline-fitting photometry is highly recommended for point source photometry. It makes the best use of the information stored in individual measurements, and the results are free from biases introduced by the map-making.
- The flux measured with aperture photometry using HIPE task annularSkyAperturePhotometry is ~9% different from the result of timeline-fitting photometry, suggesting large uncertainties in the aperture correction.
- For sources in blank fields, the HIPE task sourceExtractorSussextractor provides reasonable measures for the flux and error, while results of HIPE task sourceExtractorDaophot are not as good.