Using CIAO Region Files

CIAO 3.4 Science Threads

Table of Contents

Using CIAO Region Files

CIAO 3.4 Science Threads

Overview

Last Update: 1 Dec 2006 – reviewed for CIAO 3.4: no changes

Synopsis:

Regions, two−dimensional filters used to include and exclude data, are commonly used in CIAO data analysis. This thread describes how regions are defined, how they are used in CIAO, and the caveats associated with their use. *We recommend that the "CIAO" region format be used for all data analysis.*

In CIAO 3.3, basic support was added for version 4 of the ds9 region files, which have a slightly different format than the version 3 files. Users working with more complicated files, such as "composite" regions, may encounter problems. Refer to the Libraries: regionlib section of the CIAO 3.3 release notes for a comparison of the two formats.

Related Links:

- The dmregions ahelp file
- ds9 Reference Manual
- Chandra Coordinate Systems: Imaging document (PS, 25pp)
- Using SAOImage ds9 thread

Proceed to the HTML or hardcopy (PDF: A4 | letter) version of the thread.

Getting Started

Sample ObsID used: 1712 (ACIS−S, 3C 273)

File types needed: evt2

Coordinate Definitions: Image, Physical/Sky, Celestial, and WCS

There are a few terms needed to discuss coordinate systems:

- **Image:** image coordinates specify a location in an image. The lower, left pixel of the array will have the image coordinates (1,1). Coordinates increase to the right in x and up in y. Note that the lower left *corner* of the array will have coordinates (0.5,0.5).
- **Physical/Sky:** "sky" is shorthand for (x,y) coordinates, the system used most commonly in Chandra data analysis. Physical coordinates are the generalized system used by ds9. The sky coordinates of an event are defined as the point where an event crosses the imaginary tangent plane for a given pointing of the telescope's mirrors. This is calculated (by the CIAO tools reproject_events and acis_process_events / hrc_process_events) from knowledge of the location of the event on the detector, the location of the detector in the observatory, and the pointing of the telescope. For Chandra data, the range of sky coordinates is 0.5:8192.5 for ACIS, 0.5:32768.5 for HRC–I, and 0.5:65536.5 for HRC–S.

  When saving regions in ds9, use "Ciao" formatted regions to ensure total compatibility with the CIAO tools. "Ciao" format regions are saved in physical coordinates by default. The CIAO and ds9 region formats section of the Using CIAO Regions thread has a discussion on the ASCII region types available for use. If necessary, it is possible to convert region formats/coordinate systems by loading one type (e.g. ds9 format in celestial coordinates) and saving it in a different format/coordinate system (e.g. CIAO format in physical coordinates).

- **Celestial:** the celestial coordinate system is the standard method of measuring positions in the sky. The transformations from sky (x,y) to celestial (RA, Dec) are described at the end of the event file "columns" listing:

```
unix% dmlist acisf01712N002_evt2.fits cols
...cut...

8:    EQPOS(RA ) = (+187.2761) +TAN[(−0.000136667)* (sky(x)−(+4096.50))]
      (DEC)    (+2.0539  )       (+0.000136667)  (   (y) (+4096.50))
```

  The two systems are related by the aimpoint of the observation. When events are projected onto the celestial sphere, RA increases to the left (hence the minus sign in the definition) and Dec increases upward. In this example, the celestial coordinates of the reference pixel in degrees (J2000) are (187.2761,2.0539).

  Although CIAO regions may be saved in celestial coordinates, only sexagesimal coordinates are currently supported (e.g. circle(17:46:14.065,−28:51:39,1.0')). Since some tools are not yet able to transform from celestial to sky coordinates, we recommend using the physical format.

- **WCS:** the World Coordinate System (WCS) is a generalized name for the celestial coordinate system. The default celestial coordinate system in CIAO and ds9 is the J2000 system. A complete description of the WCS standard is given on the WCSTools home page.

  As described above, the (x,y) coordinate system is related to the celestial coordinate system by specifying the (RA, Dec) of a reference pixel in the file's header. It is sometimes necessary to edit these header keywords to improve upon Chandra's pointing accuracy. For more information, see the Aspect caveats and the Improving the Astrometry of your Data: Correct for a Known Processing Offset thread.

For a detailed discussion of the relationships between Chandra instrument coordinate systems, see the Chandra Coordinate Systems: Imaging document (PS, 25pp).

CIAO and ds9 Formats

Regions

There are several region formats available in ds9 from the "Region –> File Format" menu. The default region format of ds9, called "DS9/Funtools", is not fully compatible with CIAO's Data Model. *ds9 does support a "CIAO" region format, which we recommend for all CIAO analysis.* While some of the non–CIAO region formats may work in CIAO analysis, more complicated regions (e.g. the "annulus" region) will not work properly.

In general, the ds9 region format has many more options for storing display properties (color, coordinate system, comments), while the CIAO region format is a more simplified, stripped down version. To compare the same regions in each format:

- Load the event file into ds9:

  ```
  ds9 acisf01712N002_evt2.fits &
  ```
- Left–click and drag on the image to create a region. The shape of the region is set from the "Region –> Shape" menu and the color can be changed from "Region –> Color". This FAQ explains on how to rotate shapes in ds9. The two regions we defined are shown in Figure 1 📷.
- To save the regions:

  1. Region –> File Format–> DS9/Funtools
  2. Region –> File Coordinate System –> Physical
  3. Region –> Save Regions... –> Save As "ds9.reg"

  Then repeat:

  1. Region –> File Format–> CIAO
  2. Region –> File Coordinate System –> Physical
  3. Region –> Save Regions... –> Save As "ciao.reg"

Now we can compare the two files. First, the ds9 format:

```
unix% more ds9.reg
# Region file format: DS9 version 4.0
# Filename: acisf01712N002_evt2.fits[EVENTS]
global color=green font="helvetica 10 normal" select=1 highlite=1 edit=1 move=1 delete=1 include=1 f
physical
circle(4083,4086,18)
box(4106.5225,4059.8722,25.441981,13.199825,312.70347) # color=red
```

The first non–comment line is a declaration of "global" conditions, such as default color and editing flags. The next line indicates that the coordinate system is "physical" (a.k.a. "sky"). Finally, the color for the red box is defined after the pound symbol.

Note that the region format changed slightly in ds9 version 4.0. The file coordinate system (here "physical") is only printed once in the new file, not at the beginning of every region definition as in v3.0 (e.g. "physical;circle(4083,4086,18)").

The same regions in CIAO format look like:

```
unix% more ciao.reg
# Region file format: CIAO version 1.0
```

```
circle(4083,4086,18)
rotbox(4106.5225,4059.8722,25.441981,13.199825,312.70347)
```

Since the "physical" coordinate system is assumed for all CIAO regions, it is not indicated for each shape. Notice that the global conditions and colors are also omitted. The ds9 `box` is called `rotbox` in CIAO, but they are otherwise identical.

Files

The term "format" may also be used to describe the filetype used to store the region information; the two allowable formats are ASCII and FITS. They are loaded into ds9 in slightly different ways. Assuming that an event file had been opened:

- *ASCII files:*

  Load the region file from the ds9 menu "Regions" –> "Load Regions..." –> `ascii_file.reg`.
- *FITS files*

  Load the region file from the ds9 menu "Regions" –> "Load Regions..." –>
  `fits_file.fits[REGION]`

  Here the block which contains the region information must be specified via the DM syntax. The block is not necessarily named `REGION`; the output of the detect tools have a `SRCLIST` extension instead.

  The filename must end in ".`fits`" for ds9 to recognize it as FITS format.

Several examples of working with region files are available in the Using the Output of Detect Tools thread.

Regions with Multiple Shapes vs. Region Stacks

A region file can define a single region that has multiple shapes. When you use the region file in an analysis task, CIAO decides whether or not points are in the region, but does not return any information about which of the shapes a particular point falls in. A region stack, however, contains multiple regions. CIAO treats each of the regions separately, essentially running the tool for each region in the stack.

This region file:

```
unix% more circles.reg
# Region file format: CIAO version 1.0
circle(4066,4191,12)
circle(4085,4087,20)
circle(4107,4059,13)
```

may either be considered a single region composed of three circles or a stack of regions, each of which is a single circle. How CIAO interprets this file depends on the syntax used in inputting it to a tool.

A common use of region stacks is in dmextract, where binning on a region creates a histogram. First we tell dmextract to treat the file as a stack by using the @ syntax:

```
unix% punlearn dmextract
```

```
unix% dmextract "acisf01712N002_evt2.fits[bin sky=@circles.reg]" circles_stack.fits
```

This command operates on the region stack and make three histograms (rows), one for each circle, in the output file:

```
unix% dmlist circles_stack.fits blocks


--------------------------------------------------------------------------------
Dataset: circles_stack.fits
--------------------------------------------------------------------------------

     Block Name                          Type         Dimensions
--------------------------------------------------------------------------------
Block    1: PRIMARY                      Null
Block    2: HISTOGRAM                    Table         18 cols x 3      rows
Block    3: GTI7                         Table          2 cols x 1      rows
Block    4: GTI4                         Table          2 cols x 2      rows
Block    5: GTI5                         Table          2 cols x 1      rows
Block    6: GTI6                         Table          2 cols x 2      rows
Block    7: GTI8                         Table          2 cols x 1      rows
Block    8: GTI9                         Table          2 cols x 1      rows
```

To compare, replacing the @ with *region()* tells CIAO to treat the shapes as a single region:

```
unix% dmextract "acisf01712N002_evt2.fits[bin sky=region(circles.reg)]" circles_region.fits
```

The output file contains only one histogram, instead of three:

```
unix% dmlist circles_region.fits blocks


--------------------------------------------------------------------------------
Dataset: circles_region.fits
--------------------------------------------------------------------------------

     Block Name                          Type         Dimensions
--------------------------------------------------------------------------------
Block    1: PRIMARY                      Null
Block    2: HISTOGRAM                    Table         18 cols x 1      rows
Block    3: GTI7                         Table          2 cols x 1      rows
Block    4: GTI4                         Table          2 cols x 2      rows
Block    5: GTI5                         Table          2 cols x 1      rows
Block    6: GTI6                         Table          2 cols x 2      rows
Block    7: GTI8                         Table          2 cols x 1      rows
Block    8: GTI9                         Table          2 cols x 1      rows
```

Excluded Regions

The most important thing to remember when working with excluded regions is that *order matters*. When generating regions with excluded shapes, the order of shape generation is relevant to how they are interpreted by the software.

For similar reasons, CIAO does not understand a region file in which the first shape is negative. Since the software processes the shapes in their listed order, CIAO attempts to exclude this shape from an area that has not yet been defined. The field() region, which defines the entire field of view, can be added to the beginning of the region file to avoid syntax errors; see "ahelp dmregions" and this FAQ for more information.

The regions are saved by ds9 in the order in which they were created. For example, Figure 2 📷 shows three regions, one of which is marked as excluded (shown by the red strike through it) via the "Region –> Properties" menu.

If the regions had been defined with the smallest (#1) first, the excluded region (#2) second, and the largest region (#3) third, the file would look like:

```
unix% more exclude_one.reg
circle(4086,4084,20)
-circle(4102,4128,100)
circle(4074,4120,200)
```

This tells CIAO to take the first region (#1), subtract the second (#2), then add in the third (#3). Running dmcopy on this region file:

```
unix% dmcopy \
     "acisf01712N002_evt2.fits[sky=region(exclude_one.reg)][bin sky]" one_image.fits

unix% ds9 one_image.fits &
```

produces Figure 3 📷.

A more sensible order of regions for this example might be:

```
unix% more exclude_two.reg
# Region file format: CIAO version 1.0
circle(4074,4120,200)
-circle(4102,4128,100)
circle(4086,4084,20)
```

which starts with the largest region (#3), removes the second (#2), and then adds back in the third (#1). Running dmcopy again:

```
unix% dmcopy \
     "acisf01712N002_evt2.fits[sky=region(exclude_two.reg)][bin sky]" two_image.fits

unix% ds9 two_image.fits &
```

creates Figure 4 📷.

Region Area Calculation

CIAO calculates the area of a region differently depending on whether it is considered simple or complex. The calculation of simple areas – those of basic, well–separated shapes – is done analytically. For complicated regions – where shapes overlap or are close enough that it's hard to determine if they are separate – CIAO divides up the area into bins (which may be smaller than a single pixel). It then counts the number of bins which are true for the test "I am inside this region." This approach has a finite error, which is usually on the order of 1% percent of the region area.

A useful way to evaluate region areas is the subspace option in dmlist:

```
unix% dmlist "acisf01712N002_evt2.fits[sky=circle(4086,4084,20)]" subspace | less

--------------------------------------------------------------------------------
```

```
Data subspace for block EVENTS: Components: 6 Descriptors: 15
------------------------------------------------------------------------------

 --- Component 1 ---
...
   8 sky                   Real4                  Circle(4086,4084,20)
   8 sky                   Real4                  Field area = 6.71089e+07 Region area = 1256.64
...
```

The "Region area" calculated for a circle of radius 20 pixels is 1256.64 square pixels, which is equal to the analytical evaluation of the circle's area.

Region areas are used in the BACKSCAL keyword added to PHA and PI spectral files generated by <u>dmextract</u>. The error in the area is almost always small compared to other calibration uncertainties, but be particularly careful when making regions comparable in size to the pixel size. Although filtering does occur in exactly the area specified when applying such a region to a file which contains fractional pixel positions, the user should be careful about interpretation.

History

03 Jan 2005 reviewed for CIAO 3.2: no changes

02 Dec 2005 updated for CIAO 3.3: ds9 v4.0 region format is slightly different than v3.0.

01 Dec 2006 reviewed for CIAO 3.4: no changes

Image 1: Image with regions defined
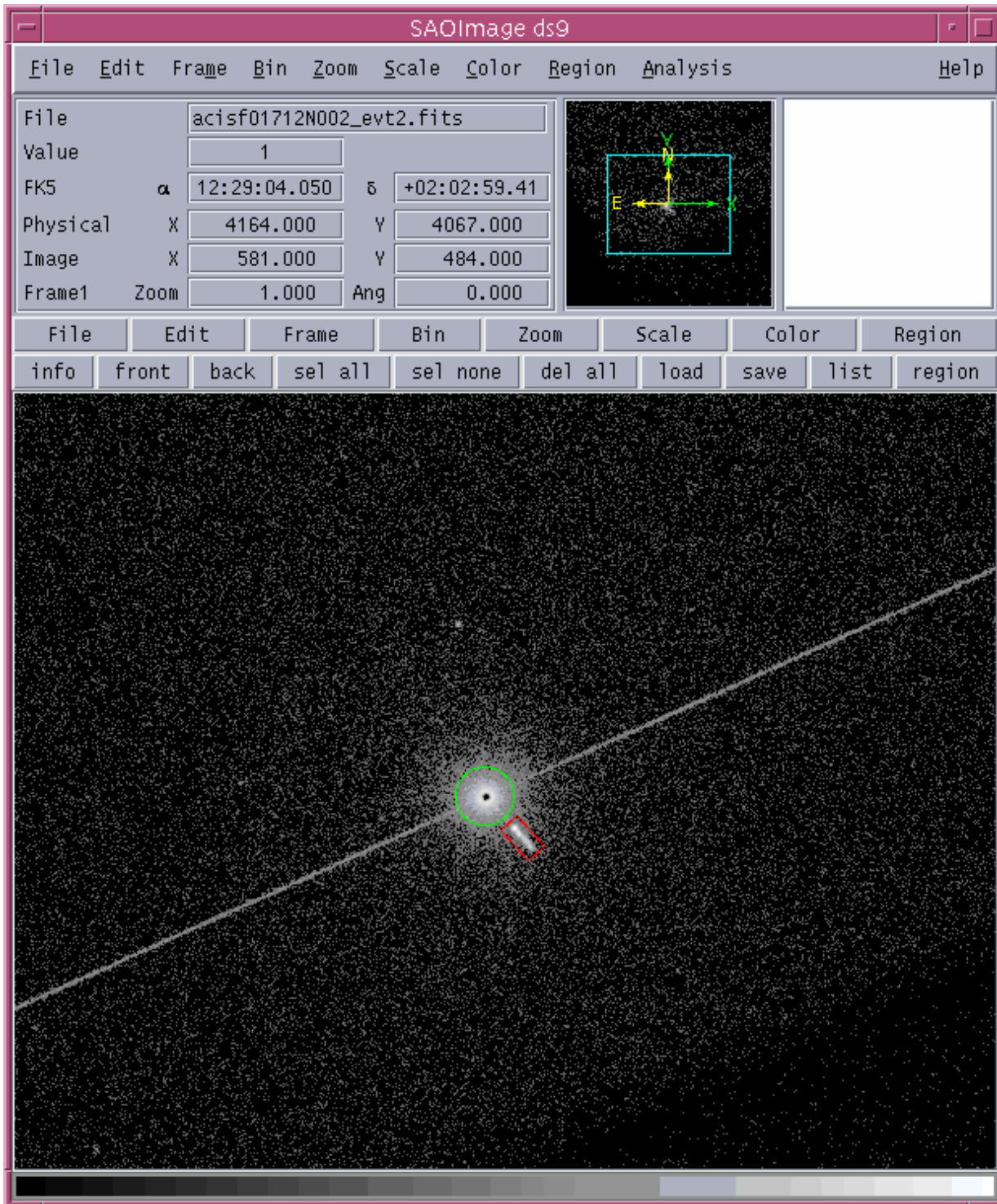
Image 1: Image with regions defined

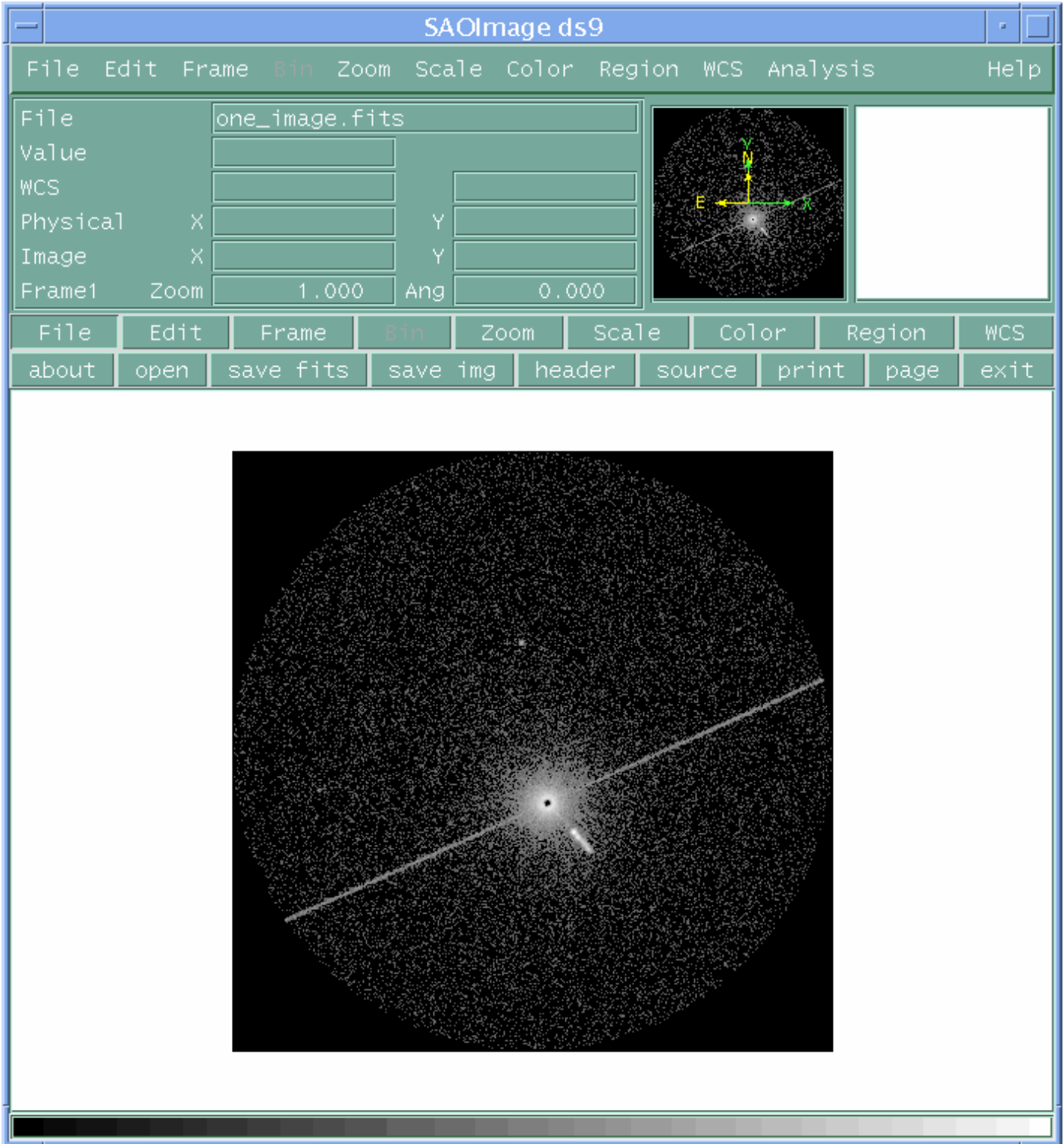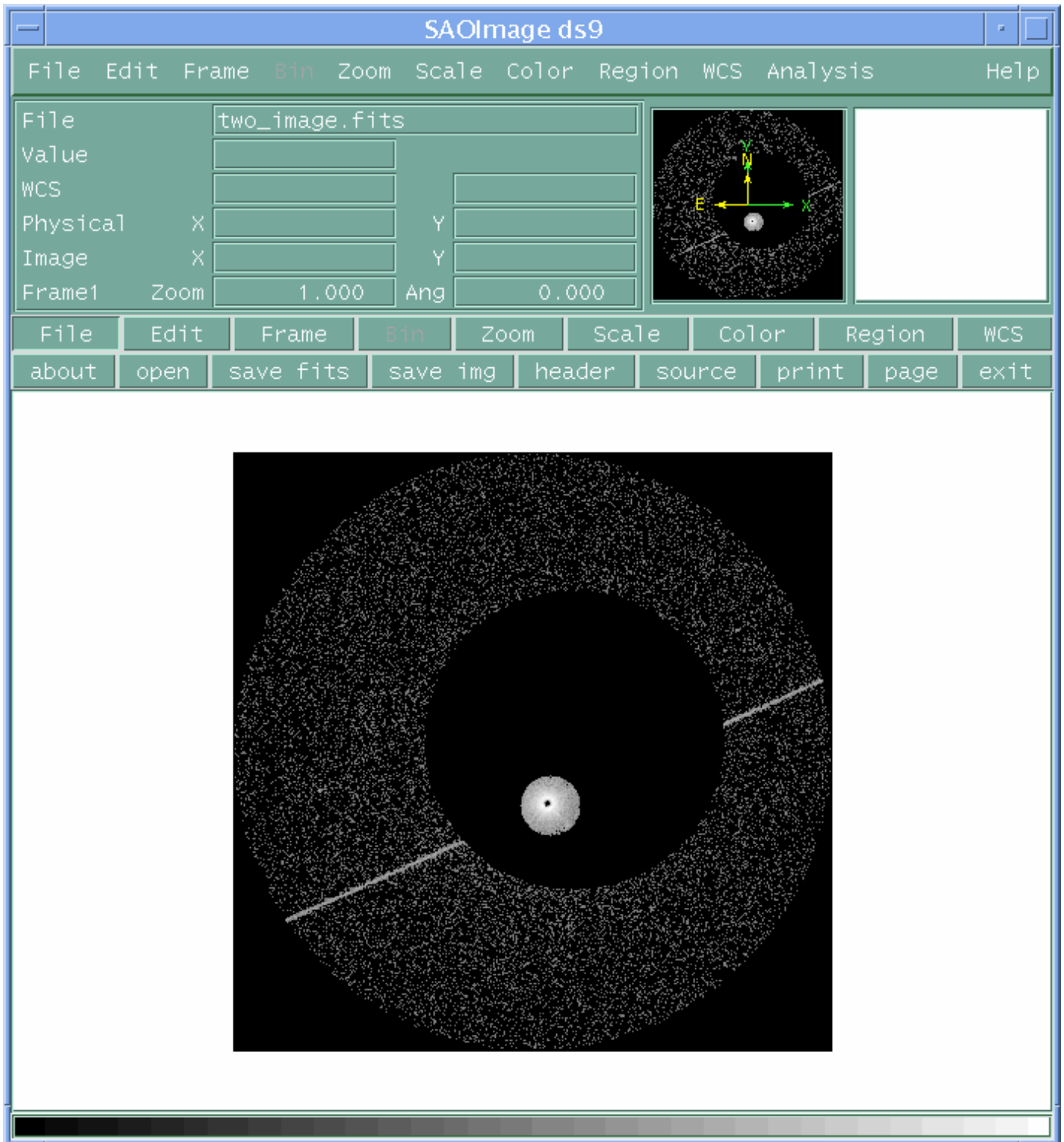# Image 2: Included and excluded regions
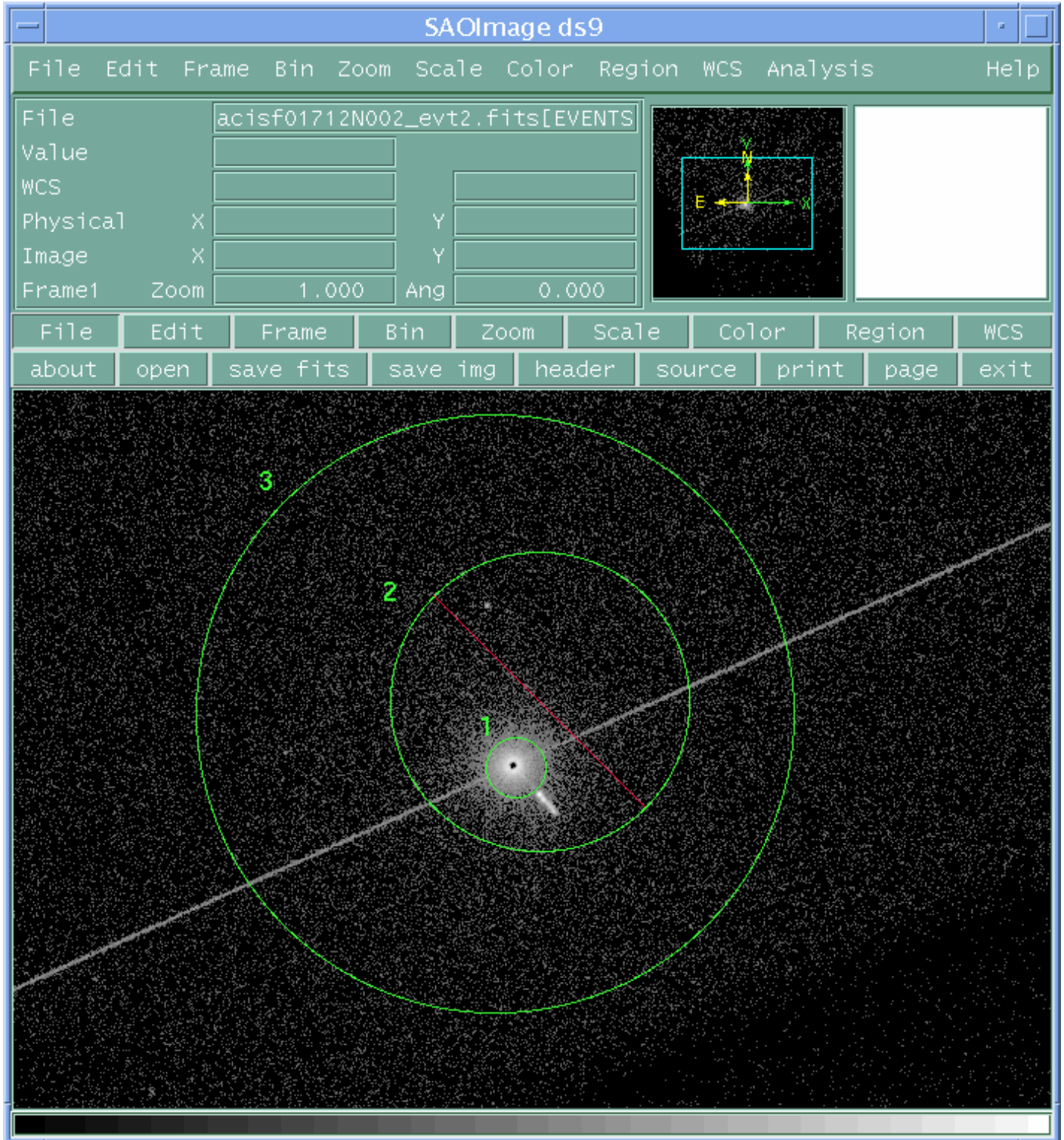
Image 3: Incorrect order of regions

Image 4: Correct order of regions

Image 4: Correct order of regions