

Tutorial: Introduction to IRAF

Kaustubh Vaghmare

In this tutorial, we are going to understand what IRAF is and its basic interface. You will learn how to start an IRAF session and navigate your way across the interface. You will learn how IRAF works in close collaboration with the tool `ds9` to create powerful work flows which make image processing and analysis very easy. We will learn to use a simple but powerful task called `imexamine` which allows a quick examination of any image using IRAF.

1 IRAF

IRAF stands for Image Reduction and Analysis Facility. It is a software developed and maintained by the NOAO (National Optical Astronomical Observatories). At some point of time in professional astronomy, lot of researchers would have developed several independent tools, each specializing in a specific task or operation. IRAF provides a uniform, cohesive and consistent environment for several such tools to be used under one roof.

The number of packages and tasks that you have in IRAF are incredible. There are tools for quick examination of images, construction of RGB composite, manipulation of tables, galaxy surface photometry, field and crowded field photometry, long slit spectroscopy, Echelle spectroscopy etc. It is never going to be possible for one to master every single task within IRAF and you don't need to either. You should be able to just use the tasks that help your specific operations of interest.

IRAF is also designed to be very flexible, in the sense that it is possible to govern almost every aspect of how a task works. If a task has to perform interpolation, you can control the algorithm used for interpolation. Similarly, if the task does some kind of outlier rejection, then it is possible to govern the exact nature of the rejection scheme. And this flexibility is also the source of complexity of IRAF. Open up the possible list of inputs one can provide to a typical program in IRAF and it is huge. Now, here is a golden rule to remember - *defaults for most technical parameters will usually work*. Of course, sometimes they don't. But if you can figure out that they don't work, you are at such an advanced level of knowledge, that you can change them!

Let's start! We will assume you know your Linux/UNIX shell commands and that IRAF has been installed.

2 Starting IRAF

In this section, we will learn how to start IRAF. Note: The `$` at the beginning of every command refers to your shell prompt. YOU SHOULD NOT TYPE IT! Similarly, prompts such as `cl >` or `vocl >` should not be typed.

1. Open a terminal. The exact procedure for this depends on your OS.
2. Launch 'xgterm', a terminal that supports graphical interaction with `ds9`.

```
$ xgterm -fn 10x20 -sbr &
```

3. Now, inside the 'xgterm' window, we will start `ds9` by saying,

```
$ ds9 &
```

4. For IRAF to work properly, it needs a file called `login.cl` in the directory where IRAF is launched. Use `'ls'` command to see if `login.cl` exists in your current directory. IF IT DOES NOT, then we'll create one,

```
$ mkiraf
-- creating a new uparm directory
Terminal types: xgterm,xterm,gterm,vt640,vt100,etc.
Enter terminal type:
```

Type `'xgterm'` and press Enter. You can now use `'ls'` to see that `login.cl` is created.

5. Next, start IRAF by saying,

```
$ cl
```

You will be greeted with a message and some names with dots at their end will appear. A `'dot'` at the end of the name means it is a package. So, `'dataio'`, `'images'`, `'noao'` etc are all packages. A package is a collection of tasks.

6. Next, we enter the `'images'` package and then `'imutil'` by saying,

```
vocl> images
imcoords. imfilter. imfit. imgeom. immatch. imutil. tv.

images> imutil
```

7. You can see a large number of tasks within the `imutil` package. Sometimes, from the name itself, it is obvious to a sufficiently trained astronomer what the task will do but even then, the inputs required and other information will not be known. IRAF provides a powerful in-built help which can be used easily as follows. Say, you want to know what `'imjoin'` does.

```
imutil> help imjoin
```

This displays the in-built help for the `imjoin` task. Notice at the bottom, a bar that says

```
[q=quit,d=downhalf,f|sp=downfull,j|cr=downline,N=next]
```

This is important. It tells you to how to scroll through the help. It is best to train your mind to notice such clues to make overall IRAF experience better. You can press `'q'` and quit. Go ahead, try out seeing the help pages for other tasks. Strangely, you might notice there is no way to scroll back. Use your terminal window's scroll bar for this.

8. Hopefully, by now, you have enough information on your window that you can no longer see what tasks are present in the `imutil` package. Type `'?'` and press Enter to get a list of tasks in current package.
9. You know you are in the `imutils` package because of the prompt which says `imutils >`. To quit current package, we use `'bye'` command

```
imutil> bye
imcoords. imfilter. imfit. imgeom. immatch. imutil. tv.

images> bye
dataio. images. lists. obsolete. proto. stsdas. tables. vo.
dbms. language. noao. plot. softools. system. utilities.

vocl> bye
ERROR: use 'logout' to log out of the CL
```

Notice that `vocl` means we are the root. Saying 'bye' here will produce an error with a friendly advice telling us what to do if we have to leave IRAF.

```
vocl> logout
```

10. Great! You know how to start IRAF, navigate packages, get help and quit IRAF. Close 'xgterm' and 'ds9' windows. Practice starting IRAF on your own a couple of times. Remember: **You don't need to use mkiraf again! This has to be done only once after installing IRAF!**

3 IRAF and ds9 Interaction

As mentioned earlier, ds9 and IRAF can *talk to each other*, allowing information from ds9 to be communicated and analyzed by IRAF and also vice-versa. In this section, we are going to see a demo of this using the task `imexamine`. Let's start!

1. Start IRAF! Forgot how to? Look above! You must have xgterm with IRAF started and also a ds9 window.
2. Now, let us load an image, which comes by default with IRAF. It has a peculiar name because it is inbuilt - we refer it to as 'dev\$pix'. So, let's open the image.

```
vocl> display dev$pix
frame to be written into (1:16) (1): [press Enter key]
z1=35. z2=346.0218
```

ds9 supports loading upto 16 images at a time into what it calls 'frames'. Hence the question from IRAF 'frame to be written into'. (1:16) means that any number between 1 and 16 is a valid input. (1) means that pressing Enter without providing any input will automatically use '1' as input. This is another important part of the IRAF environment. You will see similar messages like this all the time.

3. Let us examine this image.

```
vocl> imexamine
```

You will see that the `vocl` prompt has now disappeared. This means that either xgterm/IRAF is busy or does not have the main control. Now, go to your ds9 window and move the cursor around the image. You will see the cursor blinking. This means that the control is in the hands of ds9 and it is waiting for a command.

4. Move the cursor to any star of your choice. When the cursor is on the star, you can press 'r'. A new window will appear that contains a radial plot of how the brightness of star changes from an estimated centre of the star.

Do you know what these values are, ones at the bottom of the plot window? Well, the help pages of `imexamine` will contain this information somewhere. Find out! That's left as an exercise to be done later.

5. Move your cursor back to the ds9 window, Point the cursor again to the same (or other) star and press 's'. You will now see a surface intensity plot of the star.

WARNING: It is best not to close any plot window when generated. On many OSs, IRAF keeps transmitting new plots to the same window and when it has been closed, no plots appear.

6. Ok, so 'r' and 's' are two commands using `imexamine`. But, how many other commands are there? Should I remember them? If I forget, what can I do? Relax! You can always get a list of commands easily. Let's see! Now while the cursor is blinking on the ds9 window, type '?'. The cursor stops blinking. Now, go to your xgterm window and you will find a big list of possible commands that `imexamine` can handle.

Only look at the 'CURSOR' commands, ignore 'COLON' commands for now.

To get back to ds9, in the xgterm window, press 'q' twice. And go back to ds9, the cursor is blinking again.

7. Let us try a new command now. 'v' is for vector plot. A vector plot means how intensity varies in an image along a line that we indicate. The line is indicated by going to the starting point of the line, pressing 'v' and then going to ending point and pressing 'v' again.

So, point to a location, press 'v'. Point to another location, press 'v' again. Now, check the plot window.

8. Okay, have we had enough of fun? If yes, quit the interactive ds9 mode by saying 'q' when the pointer is on the ds9 window. If no, try out other commands to see what they do.
9. When done, exit the IRAF session.

4 Conclusions

This concludes our introductory tutorial on IRAF. If you want to learn some more features of IRAF such as manipulating parameters etc. please see the tutorial of Basic Image Reduction.

We will conclude with some golden rules that might save a lot of your time while learning IRAF.

- If IRAF misbehaves - hangs, does not give output etc. for a long time - restart your session!
- Always start IRAF from your home directory where the login.cl file is present. If this file is not found, IRAF will warn you saying 'no login.cl' file found. Pay attention to this. You will suffer miserably if you ignore this.
- Just because you see a ds9 window and the cursor is blinking does not mean the commands will work. The ds9 window should be activated. Notice the color of the ds9 window title bar to ensure it is active else your commands will not work.
- Once a plot window is generated, avoid closing it. IRAF tends to misbehave considerably when such a window is closed. For example, no other command that produces plots will work well.