Astronomía Estelar

Práctica 0

Introducción al IRAF

El paquete IRAF (*Image Reduction and Analysis Facility*) es un programa de uso general para la reducción y análisis de datos astronómicos. Fue desarrollado por el grupo de IRAF en el NOAO (National Optical Astronomy Observatories). Esta práctica ofrece un tutorial que introduce al estudiante en el uso de las herramientas más básicas del IRAF, a través de actividades simples y ejercicios exploratorios.
Deberá entregar un informe con lo realizado y lo aprendido.

1. Introducción:

Al trabajar con IRAF, estaremos utilizando dos entornos de líneas de comando: El entorno de UNIX y el entorno de IRAF ("cl"). Para distinguirlos usaremos en los ejemplos "%" para la línea de comandos de UNIX y "cl>" para la línea de comandos de IRAF CL (ambos sin las comillas).

Precaución: Tanto UNIX como IRAF son "case-sensitive", es decir que distinguen entre mayúsculas y minúsculas. Tener cuidado al escribir los nombres de los archivos.

1.1. Configurando el IRAF para su primer uso

IRAF requiere un archivo de configuración (login.cl) y un directorio para guardar los parámetros del usuario (uparm). El comando **mkiraf** creará el login.cl y uparm necesarios en el directorio en el cual se ejecuta.

Para configurar el usuario de IRAF/LINUX, y generar el archivo login.cl y directorio uparm hacer por única vez (es posible que haya que volver a loguearse):

```
% iraflinux_oalp_user.sh


   -- creating a new uparm directory
Terminal types: xgterm,xterm,gterm,vt640,vt100,etc.
Enter terminal type: xgterm
A new LOGIN.CL file has been created in the current directory.
You may wish to review and edit this file to change the defaults.
```

Cuando el script pida un tipo de terminal, poner: `xgterm`.

IRAF tiene varias interfaces gráficas que son útiles para reducir imágenes. Estas sólo funcionarán correctamente si corremos IRAF desde una xgterm. Para abrir la xgterm (que funciona prácticamente igual que una xterm común), escribir xgterm en la línea de comandos. El símbolo & hará que vuelva el prompt en esa ventana:

```
% xgterm &
```

Si queremos que la terminal tenga una barra lateral, que permita subir para ver lo que escribimos más arriba en la pantalla, agregamos una "scroll-bar" ejecutando el comando de esta manera:

```
% xgterm -sb &
```

Siempre se debe correr IRAF desde una xgterm. Algunas de las tareas de IRAF usarán las propiedades especiales de una xgterm para crear ventanas gráficas interactivas que se pueden usar para ver y manipular los datos. Usualmente sólo se necesita una xgterm con IRAF corriendo en esa terminal. Se pueden tener otras terminales comunes abiertas al mismo tiempo, de manera tal de poder ver qué archivos están en el directorio de trabajo.

1.2. Editando el archivo login.cl

Ahora puedo ver el archivo que se acaba de generar utilizando el comando de linux "cat":

```
% cat login.cl
```

Podría editar este archivo, y modificar los parámetros desde aquí si quisiera.
Además, se tuvo que generar un directorio `dev/` en mi home. Es un buffer de memoria.

Ejercicio 1: Antes de ingresar al CL (command language), inspeccione el archivo login.cl, identifique algunas líneas o parámetros.

1.3. Despliegue de imágenes usando DS9

También necesitaremos un programa para desplegar imágenes. El más popular es el DS9. Para abrirlo, ejecutar (desde cualquier terminal):

```
% ds9 -fifo /home/username/dev/imt1 &
```

2. Resumen de los pasos a seguir

Para comenzar a trabajar con IRAF, los pasos a seguir son:

- Abrir una `xgterm`

- Abrir el display de iraf:

  ```
  % ds9 -fifo /home/username/dev/imt1 &
  ```

- En la terminal xgterm que abrimos, tecleamos (desde el home, o donde está guardado el archivo login.cl):

  ```
  % ecl
  ```

  Ya estamos dentro del IRAF para trabajar, y con el SAOImage ds9 abierto para desplegar imágenes.

- Para salir de ecl, escribir:

  `ecl>logout` (ó abreviado `ecl>log`)

3. Paquetes de IRAF - Tareas

IRAF es un sistema integrado por muchos paquetes de aplicaciones. Todas las tareas de IRAF están agrupadas en paquetes. Para usar una tarea, hay que cargar el paquete que contiene esa tarea.

En el login.cl, están los paquetes que se van a cargar automáticamente. También aparecen listados cuando abrimos IRAF por primera vez. Los paquetes son aquellos nombres que terminan con un punto. En mi caso, actualmente son:

```
ctio.       gemini.     language.   obsolete.   stsdas.
dataio.     gmisc.      lists.      plot.       system.
dbms.       guiapps.    nmisc.      proto.      tables.
fitsutil.   images.     noao.       softools.   utilities.
```

Las tareas (aplicaciones) no llevan punto al final.

Para saber que hacen, o como funcionan, utilizamos el comando “help” en la terminal xgterm. Por ejemplo:

```
ecl>help fitsutil
```

También se puede hacer help de una tarea, y éste nos va a decir, en el primer renglón, dentro de que paquete/s está la tarea.

Ejercicio 3: En su caso, ¿Qué paquetes se cargaron automáticamente? Mediante el comando “help” investigue que utilidad tiene cada paquete.

Las imágenes se guardan donde queramos. Dentro del “ecl” podemos utilizar algunos comandos de LINUX, y por ello, podemos cambiar de directorio utilizando el comando “cd” hasta movernos al directorio donde está la imagen.

Las imágenes astronómicas suelen guardarse en un formato especial llamado "Flexible Image Transport System" (FITS). Una imagen en formato FITS suele tener una extensión ".fit". En un mismo archivo FITS está almacenada la imagen, más un conjunto de metadatos que describen esa imagen. Este último se llama *header*, o cabecera.

El formato FITS permite "millones"de escala de grises para guardar distintas valores de intensidad.

Dentro de IRAF funcionan algunos comandos LINUX definidos en el archivo login.cl (se pueden agregar otros a gusto del usuario).

Me puedo mover al directorio donde está la imagen. Ahora puedo mirar el *header* de la imagen. Utilizamos la tarea "imhead":

```
ecl>imhead n3294.fit
```

Para ver un *header* más largo, puedo agregar el parámetro "longheader=yes", también escribiendo de forma simplificada "l+". Así:

```
ecl>imhead n3294.fit l+
```

Todas las tareas de IRAF tienen un conjunto de parámetros que definen la tarea para esa corrida en particular. Hay tareas con muchos parámetros y otras con pocos parámetros. Para listar los parámetros de una tarea, usamos el comando "lpar".

Otra tarea útil se llama "hedit", y sirve para editar cabeceras (*headers*). Para listar los parámetros de la tarea "hedit", por ejemplo, hacemos:

```
ecl>lpar hedit
```

Cada uno de los parámetros del FITS se llama *keyword*.

Se puede editar algún *keyword* del *header*. Utilizamos la tarea "hedit". Por ejemplo, supongamos que tenemos una imagen llamada m74.fit, y en el *header* un campo de nombre title. Si le quiero dar el valor "sky flat" a ese campo, hago:

```
ecl>hedit m74 title "sky flat"
```

Ejercicio 4: Inspeccione la cabecera (*header*) de la imagen "n3294.fit" con *imhead*, identifique algunos *keywords* y realice algún cambio menor en sus valores, con el comando *hedit*.

Para editar algún parámetro de cualquier función, utilizamos la tarea "epar". Así, nuevamente utilizando como ejemplo la tarea hedit:

```
ecl>epar hedit
```

Aparece esto:

```
                                    I R A F
                         Image Reduction and Analysis Facility
PACKAGE = imutil
   TASK = hedit

images  =                          images to be edited
fields  =                          fields to be edited
value   =                          value expression
(add    =                       no) add rather than edit fields
(addonly=                       no) add only if field does not exist
(delete =                       no) delete rather than edit fields
(verify =                      yes) verify each edit operation
(show   =                      yes) print record of each edit operation
(update =                      yes) enable updating of the image header
(mode   =                       ql)
```

Y se puede escribir acá. Si queremos dejar el campo vacío, escribimos "", que es un texto vacío. Una vez que cambiamos algo, ponemos:

```
:go
```

Entonces aplica los cambios y ejecuta la tarea. Alternativamente, si ponemos:

```
:q
```

sale sin grabar. Por otra parte:

```
:wq
```

graba y sale (pero no ejecuta la tarea). Notar que esas acciones son las mismas que se usan al editar un archivo con el programa "vi", que es el editor por defecto del IRAF (como descubrimos al explorar el archivo login.cl).

4. Tarea "display"

La tarea "display" permite visualizar la imagen en el Saoimage:

```
ecl>display n3294.fit
```

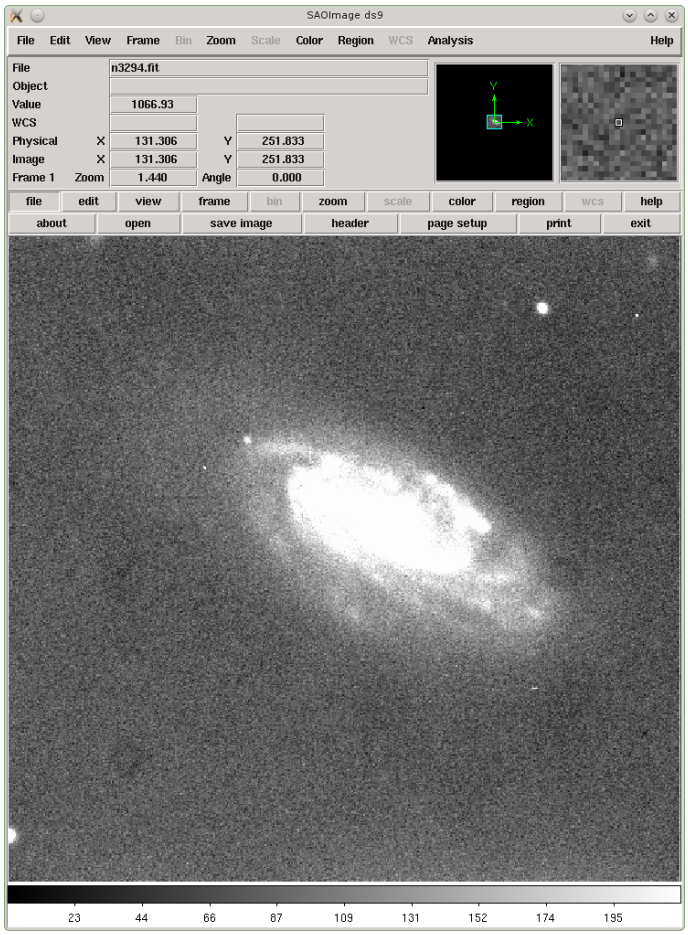
Pregunta en que "frame" escribir. Podemos elegir el "1". Tiene 16 para elegir. Se pueden guardar 16 imágenes a la vez. Se puede indicar directamente en el comando como un parámetro extra del display así:

```
ecl>display n3294.fit 1

%frame to be written into (1:16) (1): 1
z1=1056. z2=1090.002
```

Ejercicio 5: Despliegue la imagen "n3294.fit" con el comando *display*. Pruebe cambiar el despliegue, contraste, aumentos, paletas de colores, etc.

En la ventana de display (que en la tab dice: *SAOImage ds9*) se ve una galaxia espiral.



Cuando desplegamos la imagen, podríamos llegar a pensar que estamos "viendo toda la información" del dato astronómico, pero éste no es el caso. Es importante entender qué es lo que está haciendo el display, y cuales son sus limitaciones.

El rango de despliegue está determinado por los parmetros z_1 y z_2 en "display". Pensemos a la imagen como una matriz de números, cada número representa el brillo en ese pixel. El rango de display, configurado a través de z_1 y z_2, representan los valores mínimos y máximos que se muestran. Si se asigna el color negro para los valores bajos y el color blanco para los valores altos, entonces z_1 corresponde a negro y z_2 corresponde a blanco. Todo lo que sea más débil que z_1 será negro, y todo pixel más brillante que z_2 será blanco. Los que tengan valores intermedios de brillo, tendrán alguna tonalidad de gris. Es importante elegir bien los valores de z_1 y z_2 ya que por ejemplo, si hay detalles en la imagen con brillo por encima de z_2, no se verá en el despliegue elegido (en realidad, se verá igual de blanco que otros objetos menos brillantes pero con brillo $> z_2$), hasta que se ajusten los valores de z_1 y de z_2.

$(z_2 - z_1)$ lo divide en 256 partes, y esa es la escala de grises que muestra el display. La escala es lineal (pero se puede modificar, investigar sobre $Lookup - tables$ en imágenes).

Si cliqueamos con el botón derecho del mouse, y movemos el mouse verticalmente, cambia el contraste (cambia el valor mínimo y máximo, es decir, el rango entre lo más brillante y lo más débil). Si movemos el mouse horizontalmente, cambia el punto de cero de la escala de grises.

Ejercicio 6: Despliegue nuevamente la imagen pero cambiando algunos parámetros de la tarea (por ejemplo: `display n3294.fit 1 zsc- zr- z1=1056 z2=1200`). ¿Qué observa?

Ejercicio 7: Liste el archivo de parámetros del *display* usando *lpar*. Utilice la tarea *help* para más información. Explique brevemente el uso de zscale, zrange, z1, y z2.

Ejercicio 8: La tarea "imexamine" provee algunas herramientas de diagnóstico rápido dentro del IRAF. Una vez ejecutado el comando "imexamine" se verá que el cursor es redondo y parpadea sobre la imagen. Pulsando distintas letras (sobre la imagen) se pueden hacer una gran variedad de tareas. Investigue cuales son algunas de esas tareas leyendo la página: `https://www.noao.edu/kpno/manuals/ice/node31.html`.

Utilizando la tarea *imexamine* analice los siguientes aspectos de la imagen: HD5980_V01_SWO_DC_2013_01_14.fits, y encuentre:

- el cielo promedio de la imagen.
- el FWHM promedio de las estrellas.
- la elipticidad de las estrellas.
- el perfil radial de las estrellas.