

IRAF

Image Reduction and Analysis Facility

Es un software que provee herramientas para el procesamiento y análisis de imágenes astronómicas. Fue desarrollado por NOAO (National Optical Astronomy Observatories), EEUU, en los años 80.

Sobre la estructura de IRAF (muy simplificado)

Tareas (tasks): programas ejecutables.

Paquetes (packages): grupos de tareas relacionadas de acuerdo a su función.

Archivos de parámetros: archivos que contienen parámetros (definidos por el usuario) que serán usados al ejecutar determinadas tareas.

IRAF tiene un set de paquetes básicos pero pueden agregarse paquetes externos desarrollados para aplicaciones específicas (e.g. HST y Gemini tienen sus propios paquetes que funcionan bajo IRAF).

Sólo la primera vez antes de usar IRAF:

```
~$ cd /iraf/iraf/local/oalp/  
~$ ./iraflinux_oalp_user.sh
```

cerrar la terminal y volver a abrirla para activar todo.

Este ejecutable genera los archivos de inicio (login.cl) para cada usuario, dichos archivos contienen definiciones para iniciar el sistema (sólo hay que hacerlo una única vez, los archivos generados serán usados cada vez que se inicie IRAF).

También se generan los directorios /bin, /dev y el directorio uparm el cual se guardarán los valores de los parámetros que se usarán en cada tarea.

1) Abrir el login.cl y mirar que cómo es y qué define.

Sobre cómo entrar en IRAF y algunas consideraciones generales

Desde una terminal abrir una **xgterm** (la *xgterm* contiene una interfase gráfica necesaria para usar IRAF)

```
~$ xgterm &
```

en la *xgterm* abrir el DS9 (del cuál se habla más adelante) y el IRAF con los siguientes comandos:

```
~$ ds9 -fifo dev/imt1 &
```

```
~$ ecl
```

Siempre hay que iniciar el IRAF desde el directorio **/home** de cada uno y luego dentro de IRAF cambiar al directorio de trabajo con el comando `cd`.

Para iniciar el IRAF y el DS9 correctamente, abrir una **xgterm** en el **/home**, iniciar el DS9 (`~$ ds9 -fifo dev/imt1 &`) y el IRAF (`~$ ecl`)

El prompt “`ecl>`” indica que estamos en el entorno del **CL** de IRAF.

CL (Command Language) es la interfase de IRAF con el usuario (sistema operativo para acceder a las tareas de IRAF).

El CL reconoce muchos comandos de UNIX: `pwd`, `mv`, `cd`, `ls`, `&` (para ejecutar los comando en el background), etc

Cuando se inicia IRAF, aparecen listados los paquetes disponibles.

Hay paquetes que se cargan automáticamente cuando se inicia IRAF. Se puede ver cuáles son esos paquetes y una breve descripción de los mismo con el comando:

```
ecl> help
```

2) Ver qué paquetes hay disponibles y para qué sirve cada uno.

Para salir del entorno *CL*

```
ecl> logout
```

Para cargar un paquete hay que tipear su nombre de paquete. e.g.

```
ecl> digiphot
```

Los paquetes tienen el nombre seguido de ".", las tareas solo el nombre, los archivos de parámetros tienen el nombre seguido de "@".

Se pueden tener cargados múltiples paquetes simultáneamente.

Para salir del último paquete cargado:

```
ecl> bye
```

Para saber que paquetes hay disponibles:

```
ecl> ?
```

Para ejecutar cualquier tarea es necesario tener cargado el paquete que la contiene.

Para saber cuál es el paquete que contiene a una determinada tarea:

```
ecl> help {tarea}
```

en el margen superior del help aparecen los paquetes que debemos cargar para poder ejecutar esa tarea, e.g.

```
ecl> help phot
```

se lee al inicio del help: *noao.digiphot.apphot*

entonces se debe cargar el paquete *noao*, el *digiphot* (dentro del *noao*) y el *apphot* (dentro del *digiphot*)... allí aparecerá, entre otras tareas listadas, la tarea *phot* para ejecutar.

El prompt cambia cada vez que se carga un nuevo paquete, indicando cual es el último paquete cargado.

La forma en que se ejecutan las tareas en cada caso depende de algunas especificaciones que le ingresa el usuario. El usuario tiene la posibilidad de configurar un conjunto de parámetros en cada tarea.

Para ver los valores de los parámetros actuales en cada tarea se usa el comando :

```
ecl> lpar {tarea}
```

e.g.

```
ecl> lpar phot
```

Los parámetros pueden ser de dos tipos "obligatorios" (hay que especificarlos CADA vez

que se ejecuta la tarea) y ocultos (estos aparecen entre paréntesis y no es necesario especificarlos todas la veces).

IRAF tiene cargados algunos de los parámetros por "default".

Una forma de modificar los parámetros de las tareas (y también los archivos de parámetros) es usando el comando *epar*

```
ecl> epar {tarea}
```

e.g.

```
ecl> epar phot
```

Dentro del *epar* (que abre el editor "vi")

```
:q sale sin guardar
```

```
:wq sale grabando
```

```
:go sale y ejecuta la tarea
```

También es posible modificar los parámetros por línea de comandos.

Las tareas guardan los parámetros definidos por el usuario si están en *mode=q!* (quit and learn), eso es lo que se almacena en el directorio uparm. Si la tarea es usada nuevamente, los parámetros "obligatorios" serán preguntados pero los parámetros ocultos serán los mismos que los definidos la última vez que se corrió la tarea (a menos que se especifiquen nuevamente).

Para volver los parámetros de las tareas (y los archivos de parámetros) a los valores que tiene IRAF por "default" se usa el comando *unlearn*

```
ecl> unlearn {tarea/archivos de parámetros}
```

e.g.

```
ecl> unlearn phot
```

Hacer la prueba con alguna tarea.

Para ejecutar una tarea, se le definen los parámetros y luego se usa *:go* (desde el *epar*) o se tipea su nombre en el CL. e.g.

```
ecl> phot
```

(si hay parámetros obligatorios que no fueron definidos los irá preguntando antes de ejecutar la tarea)

Generalidades sobre el uso del “help” de IRAF (un help para el “help”)

ecl> ? → lista las tareas/paquetes disponibles en el último paquete cargado.

ecl> ?? → lista todas las tareas/paquetes disponibles.

ecl> help {paquete} → describe brevemente (en una línea) las tareas/paquetes del paquete.

ecl> help {tarea} → describe (con detalle) la tarea.

En general el “help” de una tarea tiene distintas secciones, entre las más importantes están:

NAME: nombre, y en una línea, qué hace al ejecutarla.

USAGE: forma correcta de llamar a la tarea para ejecutarla.

PARAMETERS: define los parámetros de la tarea, dónde se usan, cuál es el valor que tienen por default, etc.

DESCRIPTION: describe (con detalle) lo que hace la tarea.

OUTPUT: describe los archivos de salida generados por la tarea.

ERRORS: describe los posibles mensajes de error y sus explicaciones.

EXAMPLES: describe ejemplos concretos sobre el uso de la tarea.

BUGS: advertencias sobre la tarea.

ecl> refer {palabra}

Busca en los helps de las tareas/paquetes y hace una lista de todos los que contienen “palabra” (es muy útil cuando se desconoce el nombre de la tarea, en cuyo caso no es posible usar el help para buscarla, se puede usar el refer con palabras relacionadas con lo que la tarea ejecuta).

Sobre imágenes en formato “FITS”

En astronomía se trabaja con varios formatos de imágenes (ej. STF, OIF, FITS), cada uno tiene sus especificaciones pero básicamente están compuestos por un archivo de texto con información relevante para la imagen y el archivo propio de la imagen. Nosotros vamos a trabajar, en general, con el formato FITS (Flexible Image Transport System) que es el formato más estándar para el almacenamiento, transporte y análisis de datos científicos.

El formato FITS puede trabajar con imágenes unidimensionales (espectros), bidimensionales (imágenes en 2D) y tridimensionales (cubos de datos) .

El archivo consiste en uno o más (en el caso de los archivos MEF, Multi Extension Fits): Header + Data Units (HDUs)

Cada HDU consiste en un Header en ASCII, seguido de una “Data Unit”, tabla binaria.

En el Header se almacena información relevante para la imagen, con Keywords (palabras claves) y su valor (el último keyword del header es END)

Los Keywords en un Header respetan un orden:

La primera línea (obligatoria para todos los headers) especifica el tamaño y formato del Data Unit

Luego vienen keywords que tienen que ver con otras características de la imagen (tipo de imagen, nombre del objeto, etc), datos del instrumental con que fue adquirida la imagen (telescopio, filtro, temperatura, etc), las condiciones de observación (masa de aire, coordenadas, etc), datos sobre el procesamiento de la imagen, etc.

Para ver un header de una imagen en IRAF hay varias opciones:

```
ec1> imheader {imagen}
```

muestra el header básico de la imagen: dimensión tipo de pixel, nombre del objeto.

```
ec1> imheader {imagen} l+
```

(l+ viene de long+) muestra el header completo de la imagen.

Al usar el comando *imhead* el header se puede ver pero no editar.

Si se desea seleccionar determinados keywords del header para verlos

```
ec1> hselec {imagen} $key1,key2,key3, ... yes
```

Para editar/borrar/agregar Keywords en un header

ecl> hedit {imagen}

A veces es necesario agregar Keywords en los headers pero una buena recomendación tratar de editar lo menos posible los headers, así se evitan muchos errores.

4) Mirar los headers de distintas imágenes (dev\$pix, reflector,Gemini) y comparar la información que tienen. Leer los helps de los distintos comandos descriptos previamente y usarlos probando distintas opciones.

DS9

DS9 (herramienta de visualización y "manipulación" de imágenes FITS (también acepta otros formatos).

DS9 es independiente de IRAF (se puede usar separadamente de IRAF), pero también se puede usar como herramienta de visualización coordinadamente con IRAF.

Para iniciar el DS9 y vincularlo al IRAF, hay que escribir en una xgterm

```
:$ ds9 -fifo dev/imt1 &
```

(en otros lados se abre desde adentro del IRAF directamente, eso depende de cómo este definido)

Advertencia: para usar con IRAF hay que abrir solo un DS9 (cuando se inicia más de uno DS9 IRAF tiene problemas).

IRAF y DS9 (muy básico)

Para desplegar una imagen en el DS9 desde IRAF se utiliza el comando display

```
ecl> display {imagen}
```

(DS9 y IRAF deben estar comunicados de lo contrario aparece un mensaje de error como: "ERROR: Cannot open device (nodelimtool,,512,512)")

Los valores **z1**, **z2** son los límites mínimo y máximo de los valores de los píxeles de la escala que (automáticamente) IRAF le dio a la imagen al desplegarla. De acuerdo al número de cuentas *z*, de cada píxel, estos aparecerán: $z < z1$ en negro, $z1 < z < z2$ en escala de grises y $z > z2$ en blanco (ver la barra con la escala).

En una imagen jpg para el valor de cada píxel hay disponibles 256 niveles de brillo (8 bits = 2^8). En una imagen fits el "rango dinámico" es mayor, para cada píxel hay disponibles 65535 (16 bits = 2^{16}).

En general los programas de visualización de imágenes son de 8 bits (para poder ser desplegados en la pantalla) de modo que deben comprimir el rango dinámico de la imagen al desplegarla.

DS9 tiene varios parámetros para mapear una imagen de 16 bits en 8 bits. La elección de estos parámetros depende fundamentalmente del rango dinámico de la imagen y de qué se desea ver en la imagen.

Es importante setear bien los valores **z1** y **z2**, que son los valores mínimo y máximo respectivamente de la intensidad de los píxeles que se van a usar en el despliegue de la imagen. Y la transformación, con los parámetros **zscale** y **ztrans**. Estos parámetros definen el tipo de transformación que usar entre los niveles de intensidad de la imagen y los niveles de intensidad del despliegue (lineal, algorítmica, etc)

Depende de cómo se despliegue la imagen qué detalles se podrán ver.

5 y 6) Desplegar con el comando display las imágenes dev\$pix y stb.fit y realizar distintas pruebas. (podrían abrir un flat y un bias). Jugar con las distintas opciones del menú: Zoom, Frame, Region, etc. (para jugar con el menú de Scale directamente desde el DS9 hay que abrir la imagen desde el menú del DS9).

7 y 8) Listar los parámetros de la tarea display. Realizar cambios en los parámetros para obtener distintos despliegues de la imagen. (zscale=no, zrange=no, ir cambiando z1 y z2).

Un ejemplo concreto:

En el menú *color*, *aips0* tiene en total 7 colores para distribuir entre *z1* y *z2*, (9 en total), si hacemos

```
ecl> display dev$pix zsc- zr- z1=50 z2=400
```

en forma lineal le asigna 1 color cada 50 cuentas

entonces lo que está en amarillo tendrá un valor de cuentas entre 300-350

Comentarios:

- ✓ En DS9 apretando la ruedita del mouse se centra la imagen donde está el cursor.
- ✓ Manteniendo el botón derecho apretado y moviéndose horizontalmente en la pantalla de DS9 sobre la imagen cambiamos el *z1* y el *z2*. Si se mueve en la dirección vertical la escala se expande (hacia arriba) o se comprime (hacia abajo).
- ✓ La tecla de tabulación cambia entre los distintos frames desplegados.
- ✓ Cuando se despliega una imagen desde IRAF con el comando *display*, en el DS9 se desactiva el menú de *Scale* (porque queda bajo el control de IRAF), para usar el menú de *Scale* desde el DS9 hay que abrir la imagen desde DS9, en *File, Open*.

El comando *imexamine* opera sobre imágenes previamente desplegadas en el DS9.

```
ecl> imexamine {imagen}
```

Permite medir varios parámetros en la imagen de forma interactiva (pero hay que saber qué y cómo está midiendo ... es decir, hay que leer help con detenimiento).

Los comandos más usados en la tarea *imexamine* son:

(Se tipean sobre la imagen desplegada en la pantalla del DS9. El cursor redondo titilando en la imagen indica que la tarea *imexamine* este activa).

? Print help (en la pantalla del IRAF, sale con "q" en la pantalla del IRAF y queda el *imexamine* activo)

- a Aperture sum, moment parameters, and profile fit (lo tira en la pantalla del ecl)
- c Column plot
- e Contour plot
- h Histogram plot
- j Fit 1D gaussian to image lines
- k Fit 1D gaussian to image columns
- l Line plot

m Statistics
image[section] npixels mean median stddev min max
q Quit
r Radial profile plot with fit and aperture sum values
s Surface plot
x Print coordinates
col line pixval [xorigin yorigin dx dy r theta]
, Quick Gaussian/Moffat photometry
. Quick Gaussian/Moffat radial profile plot and fit

9) Lea con detenimiento el help correspondiente a *imexamine* e *implot*.

10) Utilizando *imexamine* estudie:

- a) el cielo promedio de la imagen.
- b) el FWHM promedio de las estrellas.
- c) la elipticidad de las estrellas.
- d) el perfil radial de la estrellas.

11) Utilizando *implot* determine si existe alguna pendiente de intensidad en la imagen y practique como realizar cortes en distintas filas y columnas, como promediar y ampliar filas o columnas.

ADS

(se muestra rapidamente en la clase)

Referencias

<http://astronomy.nmsu.edu/holtz/a535.fall08/supplement/node2.html>

http://www.iac.es/galeria/jap/IRAF_notes/index.html

<http://IRAF.noao.edu/>

<http://IRAF.noao.edu/docs/docmail.html>