Informe de Laboratorio

Cúmulos Abiertos - Taller 1

David Leonardo Paipa León

Universidad de los Andes

Profesor : Alejandro Garcia Varela

05 de Septiembre de 2018

1. Introducción

El ejercicio propuesto introduce ciertas funciones de IRAF y enseña funciones útiles como extraer información de los encabezados de las imágenes, analizar y visualizar imágnes y establecer parámetros para los tasks que se usan usualmente en este tipo de procesos de análisis.

1.1. IRAF

Se trata de software para la reducción y análisis de imágenes desarrollado por el **National Optic Astronomy Observatory** en Estados Unidos (NOAO)[1]. El **Image Reduction and Analysis Facility**, o IRAF por sus siglas en inglés, es fruto de la colaboración de astronomos y programadores con el fin de analizar mejor las imágenes de pixeles con datos proporcionados por detectores tales como CCD's. A pesar de que se usa principalmente en análisis de datos astronómicos, también tiene usos de análisis en diversas ramas de investigación en ciencias[2].

1.2. Extensiones de los archivos

Los archivos cuentan con una cadena de caracteres, usualmente después de un punto, anexada al final del nombre o identificador del archivo. Esta cadena de caracteres **NO** determina ninguna obligación en cuanto al contenido del archivo [1], sino que dicta que procedimientos usa el sistema operativo para extraer información del archivo o ejecutarlo si se trata de una rutina ejecutable.

Para este ejercicio las imágenes tienen formato **.fits** que corresponde a **Flexible Image Transport System**[3]. En este formato las imágenes pueden contener información adicional como encabezados y notas. En este tipo de archivos se pueden almacenar varias imágenes de diferentes procedencias, es decir, en el mismo archivo fits se pueden almacenar imágenes de , por ejemplo, el rango óptico y el rango infrarrojo del mismo

[1] A no ser que el archivo sea creado por un programa que estructura la información dentro del archivo de manera adecuada.

objeto. Para este ejercicio quien está encargado de compilar, procesar y visualizar estas imágenes es **SAOImage DS9**.

1.3. Observaciones

A continuación, algunas observaciones sobre el desrrollo de los ejercicios propuestos y las condiciones que se dieron durante su realización:

- ✓ Pese a seguir las instreucciones proporcionadas por correo a los estudiantes, IRAF tenía problemas con los paths de ciertas funciones como la consola de *XGTERM* y la implementación *ECL*[2]. Re instalé IRAF siguiendo información proporcionada por gente en foros de internet que tuvieron los mismos problemas y ya funciona correctamente.

- ✓ El visor de imágenes SAOImage DS9 no ejecuta correctamente en mi computador desde XGTERM, y por ende es un poco más complicado visualizar imágenes correctamente y ejecutar el task *imexamine*. Solucioné el problema ejecutando DS9 desde la consola de linux directamente. Una vez abierto DS9 , los tasks asociados a visualizar y examinar imágenes corren adecuadamente desde la consola de XGTERM.

- ✓ Exportar imágenes desde DS9 es complicado, puesto que al afectar la posición del visualizador de imágenes (ximtool) también se ven afectadas las imágenes exportadas. Para ello, solo las exporté y edité las imágenes con GIMP [3].

- ✓ Para el task imexamine, cuando se oprimía 'a' con el cursor en la imágen no ocurria nada a menos que en la consola de XGTERM se escribiera de nuevo `imexamine`. es por eso que se usa imexamine después de señalar cada objeto con 'a'.

- ✓ El ejercicio se realizó en un computador HP Folio 9480m con Sistema operativo Ubuntu 18.04.

  También se usó software adicional como GIMP.

1.4. Objetivos

El objetivo principal de este ejercicio era introducir IRAF y sus herramientas en el análisis de imágenes. Para cumplir el objetivo, el ejercicio pide encontrar el desplazamiento relativo entre dos imágenes que están en formato fits usando herramientas proporcionadas por IRAF y DS9.

[2]Embedded Common-Lisp
[3]GNU Image Manipulation Program.

2. Procedimiento y Resultados

A continuación se explica el procedimiento seguido para la ejecución del ejericio y lograr los objetivos propuestos.

2.1. Iniciando

Primero, se inicializa IRAF en el directorio en el que se trabajará. para ello basta con el comando `mkiraf` en la consola de linux. Esto crea el archivo de Login llamado **login.cl** con las especificacionas bajo las cuales se trabajará en IRAF, como por ejemplo el editor de texto por defecto que se usará o la identdad del usuario. Posteriormente, se crea una ventana de SAOImage DS9 con el comando `DS9 &` en la consola de linux también.

Finalmente, se crea una consola de XGTERM con el comando `xgterm &` y en esta consola se ejecuta el comando `ecl` para comenzar a usar iraf y sus funciones. Es importante mencionsr que esto se debe hacer en el mismo directorio que tiene el **login.cl** ya que de lo contrario no funcionará. Tanto para ejecutar DS9 como para abrir la consola de XGTERM se usan los `&` para que la consola de linux no se bloquee.

2.2. Funciones básicas

El ejercicio sugiere practicar ciertos comandos básicos que se usan en el análisis de imágenes. A continuación un breve recuento de lo que se observó:

- ✓ `rfits` : (Task) Sirve para decomprimir los archivos de las imágenes y crea nuevos archivos con la extension `.rfits` .
- ✓ `help/phelp` : (comando) `help` seguido del nombre de cualquier package proporciona información sobre su uso. Para Tasks el comando a usar es `phelp`
- ✓ `lpar` : (comando) `lpar` seguido del nombre de un task muestra lista con los parámetros de este task.
- ✓ `epar` : (comando) edita la lista de parámetros del task.
- ✓ `unlearn` : (comando) Resetea la configuración de parámetros del task que se escriba a continuación a la versión por defecto de estos.
- ✓ `hselect` : (task) Permite buscar palabras clave en los headers de las imágenes.
- ✓ `imheader` : (task) Revela información del header de la imágen mencionada. Es prudente recordar que las imágenes fits permiten almacenar información valiosa en sus encbezados.
- ✓ `hedit` : (Task) Permite adicionar palabras clave a los encabezados de las imágenes.

- ✓ `display` : (task) Permite mostrar la imágen mencionada en el visualizador, que en este caso se trata de DS9. Se puede especificar en qué frame se quiere abrir la imágen.
- ✓ `imexamine` : (task) Es el task encargado del análisis de la imágen. Permite examinar puntos específicos de la imágen y extraer los datos solicitados.
- ✓ `imshift` : (task) genera un corrimiento en X y Y en la imagen proporcionada como parámetro. Los desplazamientos en X y Y también entran como parámetro.
- ✓ `imsum` : (task) Promedia las imágenes proporcionadas como parámetro.
- ✓ `imarith` : (task) permite operaciones aritméticas entre las imagenes proporcionadas.
- ✓ `imstatistics` : (task) Muestra las propiedades estadísticas de las imágenes proporcionadas.

2.3. Análisis de la imágen

Primero, se extrajeron las imágenes *fintro001* y *fintro002* con el comando `rfits fintro* '' '' junk old+` en archivos llamados *im010.fits* y *im011.fits* respectivamente.

Se examina el encabezado de la imágen extrayendo información relevante.

```
|ecl> unlearn imheader
|ecl> imheader im010.fits
|ecl> imheader im011.fits
```

Se obtiene que se trata del objeto **M-92** que corresponde al cúmulo globular **NGC 6341** observado con filtro **V** que corresponde al visual.

Posteriormente, se muestra la imágen *im010.fits* en DS9 en el frame 1 con:

```
|ecl> unlearn display
|ecl> display im010 1
```

Una vez con la imágen en la interfaz se señalan las que serán las 3 estrellas que posteriormente se usarán de referencia para saber el shift de las imágenes. Se señalan con el *pointer* circular las estrellas y posteriormente con el task `imexamine` y presionando 'a' con el cursor sobre estas estrellas se obtienen datos de su posición en la imágen.

```
|ecl> unlearn imexamine
|ecl> imexamine
```

y para cada estrella, después de oprimir 'a' sobre cad una, los parámetros que se obtienen los escribo en un archivo de texto.

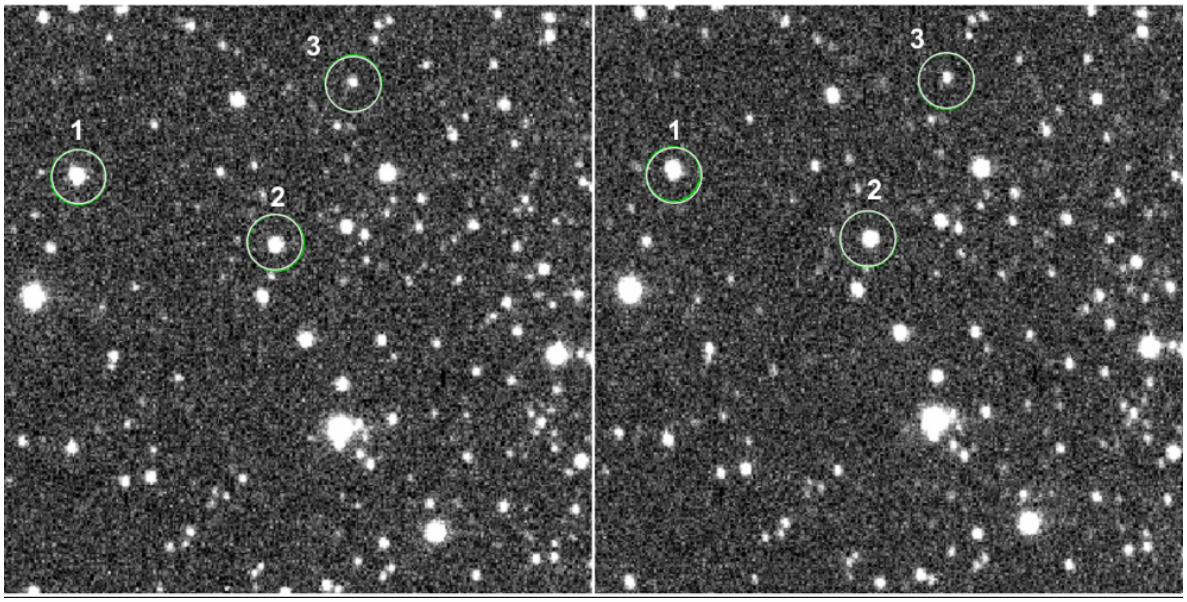
```
|ecl> imexamine > star1.txt
|ecl> imexamine > star2.txt
|ecl> imexamine > star3.txt
```

Después, se oprime 'd' con el cursor sobre la imagen mostrada por DS9. Esto importa otra imagen al mismo frame, pero conserva los pointers. Esto muestra la nueva imagen con los pointers en la misma posición que antes, y de nuevo, almaceno sus datos en archivos de texto.

```
|ecl> unlearn imexamine
|ecl> imexamine
|ecl> imexamine > star1s.txt
|ecl> imexamine > star2s.txt
|ecl> imexamine > star3s.txt
```

Así, se obtienen los datos de posición para cada objeto en las dos imágenes. Las estrellas seleccionadas como star1, star2 y star3 se muestran a continuación:

Figura 1. Imágenes de los objetos 1 , 2 y 3 respectivamente señalados en las imágenes im010.fits (izquierda) y im011.fits (derecha).



y los datos de posición correspondientes en las imágenes para cada objeto que se obtuvieron fueron:

Tabla 1. se muestran los datos de posición obtenidos para cada estrella en cada imágen y los respectivos desplazamientos de la forma im010-im011.

Estrella	**X im010**	**Y im010**	**X im011**	**Y im011**	**Δ X**	**Δ Y**
1	138.00	328.34	138.82	330.08	-0.82	-1.74
2	243.85	289.37	244.05	289.86	-0.2	-0.49
3	281.99	374.13	286.12	374.41	-4.13	-0.28

Al promediar los datos para obtener el shift en X y Y se obtiene el desplazamiento que se debe efectuar sobre im011 para que quede alineada con im010:

Tabla 2. se muestran los promedios de los desplazamientos en X y Y.

	Δ Promedio
X	-1.71
Y	-0.84

Finalmente, se aplica el shift sobre *im011.fits* y se crea una nueva imagen llamada *q011.fits*. Para comparar el resultado se oprime 'Blink' con el fin de sobreponer las imágenes y notar la diferencia. El desplazamiento entre ambas imágenes ahora es casi indetectable.

```
|ecl> imshift im011 s011 -1.71 -0.84
|ecl> display im010 1
|ecl> display q011 2
```

Ahora, se usa aritmética sobre las imágenes *im010.fits* y *q011.fits* para promediar las dos imágenes. El ejercicio sugiere 2 formas de llegar a la imagen promediada, así que se hicieron las dos.

Forma 1 Se usa la siguiente rutina:

```
|ecl> unlearn imsum
|ecl> imsum im010,q011 aver1 pixt=r calct=r option=average v+
```

Obteniendo así una nueva imagen llamada *aver1.fits* que sería el promedio de las dos imágenes según este método.

Forma 2 Se usa la siguiente rutina:

```
|ecl> unlearn imarith
|ecl> imarith im010 + q011 aver2 pixt=r calct=r v+
|ecl> imarith aver2 / 2 aver2
```

Obteniendo así una nueva imagen llamada *aver2.fits* que sería el promedio de las dos imágenes según este método.

La prueba de que ambos métodos arrojan el mismo resultado consta en imprimir la estadistica asociada a ambas imágenes:

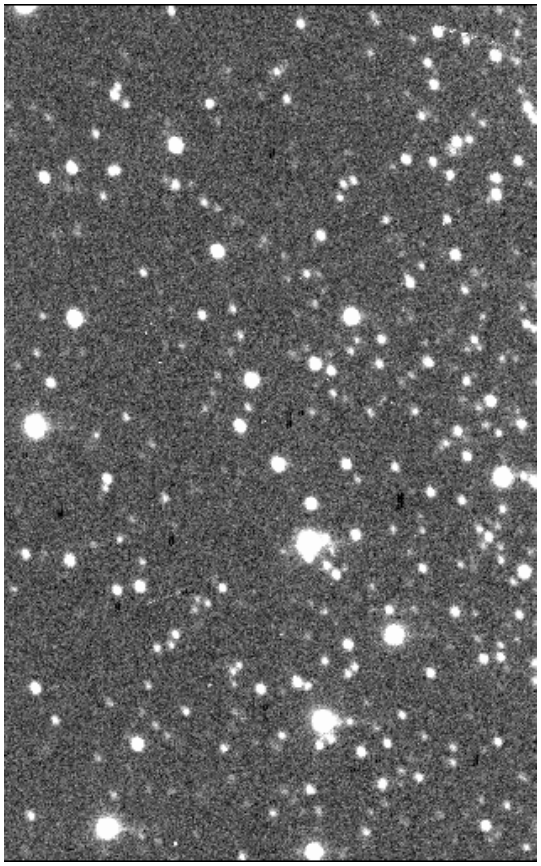
```
|ecl> unlearn imstatistics
|ecl> imstatistics aver*.fits > stats.txt
```

Finalmente se obtiene así en *stats.txt* el siguiente resultado, donde claramente se puede observar que ambas imágenes representan lo mismo:

IMAGE	NPIX	MEAN	STDDEV	MIN	MAX
aver1.fits	161862	46.75	82.33	13.51	7030.
aver2.fits	161862	46.75	82.33	13.51	7030.

La imagen resultante del promedio de las imágenes *im010.fits* y *q011.fits* es:

Figura 2. Imagen que muestra el promedio entre im010 y q011



Como ultimo paso, hay que salir de IRAF antes de cerrar la consola de XGTERM, asi que se usa:

```
|ecl> logout
```

Para finalmente poder cerrar todas las ventanas usadas, como consolas y DS9.

3. Conclusiones

En general, IRAF y DS9 permiten hacer un muy buen análisis de imagen, pues cuenta con certeza numérica y permite tratar archivos de manera rápida y organizada.

Por ejemplo, si se quería que *q011.fits* fuera *im011.fits* con el shift corregido para alinearse con *im010.fits*, se esperaria que la operación *im010.fits - q011.fits* resultara en una imágen con pocos detalles.

Figura 3. Imagen que muestra la resta entre im010 y q011

Se logró determinar el desplazamiento aproximado entre las dos imagenes iniciales y efectuar la corrección sobre la segunda imagen de manera exitosa. No obstante, los valores de desplazamiento para las 3 estrellas tenían una desviación grande, lo que hace que el promedio de los datos sea menos confiable.

Es probable que la estadística en este caso mejore si se tienen en cuenta más estrellas. También hay que revisar la relevancia de la estrella en cuanto a brillo dentro de la imagen pues parece ser que esto afecta la incertidumbre de los datos.

La evidencia de imágenes resltantes muestra que es un buen estimado del desplazamiento. Lo cual deja muchas preguntas abiertas sobre si esta desviación en los datos del desplazamiento es imortante o es una consecuencia directa de la calidad de las imágenes.

A pesar de usar lenguajes de programación poco amigables con el ususario, IRAF es fácil de usar. Lo complicado, por mucho, es el proceso de istalación y configuración de los paths de IRAF. Estos inconvenientes con XGTERM, ECL y DS9 consumieron mucho tiempo del ejercicio.

4. Referencias

1. about National Optic Astronomy Obsevatory., https://www.noao.edu/about-noao.php, Accessed: 2018-09-02.
2. Barnes, J. A Beginner's Guide to Using IRAF., 1.ª ed., August 1993.
3. IAU FITS Working Group., https://fits.gsfc.nasa.gov/iaufwg/iaufwg.html, Accessed: 2018-09-02.