

Introducción a la Astronomía de Rayos X

Guía 06: Análisis de datos de *Chandra*

Introducción

A lo largo de esta guía veremos cómo realizar búsquedas, descargar y procesar observaciones de las cámaras ACIS y HRC del satélite *Chandra*, y cómo obtener los productos científicos a partir de las listas de eventos reprocesadas.

Alternativas para la búsqueda de observaciones

Existen muchas maneras de obtener observaciones de rayos X a través de la red. Damos aquí dos opciones básicas. La primera hace uso de la base de datos de altas energías *HEASARC*, mientras que la segunda corresponde a la base de datos de *Chandra*.

HEASARC → **Browse** →

ingresar nombre de fuente o coordenadas y elegir misiones de rayos-X:

<https://heasarc.gsfc.nasa.gov/cgi-bin/W3Browse/w3browse.pl>

Base de datos de Chandra (CDA)

<https://cxc.cfa.harvard.edu/cda/>

Vista rápida (quick picture)

<https://cda.harvard.edu/pop/>

Webchaser

<https://cda.harvard.edu/chaser/>

Datos de *Chandra*

La estructura y empaquetamiento de los datos son similares a los de *XMM-Newton*, aunque con algunas diferencias. A lo largo de esta guía veremos cómo reprocesar los datos crudos para obtener listas de eventos filtradas y datos de valor científico como imágenes, curvas de luz y espectros.

Crear un directorio específico donde descargar los datos de las observaciones:

Ejemplo: `> mkdir ~/ALGO/FUENTE/`

Depositar allí los datos ODF descargados de HEASARC:

Ejemplo: Descomprimir `> tar -xvf ObsID_6732.tar`

Una vez descomprimidos, podemos pasar a *reprocesar* las listas de eventos con la nueva base de datos de calibración (*CALDB*), usando el script *chandra_repro* de *CIAO*. Para ello será necesario tener inicializados los ambientes de *HEASOFT* (*heainit*) y *CIAO* (*ciaoinit*)

Pararse en el directorio de trabajo:

`> cd ~/ALGO/FUENTE/OBSID`

Ejecutar *ls* y comprobar la existencia de los directorios *primary* y *secondary*:

`> ls`

Lanzar *HEASOFT* y *CIAO* para comenzar a trabajar con los datos de *Chandra*:

`> heainit`

`> ciaoinit`

Ejecutar la tarea para reprocesar la observación:**> chandra_repro**

La salida será un nuevo directorio llamado *repro/* donde estarán almacenadas las listas de eventos reprocesadas. A partir de las listas de eventos, procederemos ahora a extraer imágenes, curvas de luz y espectros.

→ IMÁGENES

La lista de eventos de nivel dos generada en el directorio *repro/*, puede usarse para obtener imágenes usando *ds9*. Para los datos de Chandra, en general, se recomienda utilizar una escala logarítmica y un agrupamiento de 2 o 4 fotones como máximo. La lista de eventos puede ser cargada en un marco RGB y seleccionar los fotones a través de la columna *energy* (en eV) para generar imágenes de falso color.

Para obtener imágenes como proyecciones de la misma lista de eventos a través de tareas específicas de *CIAO*, puede utilizarse *dmcopy* que es el equivalente a *evselect* de *SAS*.

Usando el parámetro [IMAGE] generamos una imagen, en este caso, con un agrupamiento de 4x4:

```
> dmcopy "acis_repro_evt2.fits[events][bin x:::4,y:::4][IMAGE]" acis_repro_img.fits
```

En cambio, en este otro caso, generamos una lista de eventos filtrada en el rango de 0.5 a 7 keV:

```
> dmcopy "acis_repro_evt2.fits[energy=500:7000]" acis_repro_evt2_500_7000.fits opt=all
```

Ayuda/Guía: https://cxc.harvard.edu/ciao/threads/dm_intro/

→ DETECCIÓN DE FUENTES

Para detectar fuentes, podemos utilizar la tarea *wavdetect*:

```
> wavdetect my_input.fits source_list.fits source_cell.fits \
  image.fits bkg.fits expfile=none psffile=mypsfmap.fits
```

donde *my_input.fits* es una imagen Chandra, *mypsfmap.fits* es generado usando, primero:

```
> mkpsfmap my_input.fits mypsfmap.fits energy=1.49 ecf=0.393
```

los productos finales incluyen una lista de fuentes *source_list.fits*, una imagen del mapa de celdas *source_cell.fits* y de fuentes *image.fits* y una imagen del fondo normalizado *bkg.fits*.

Ayuda/Guía: <https://cxc.harvard.edu/ciao/ahelp/wavdetect.html>

→ CURVAS DE LUZ

Para extraer una curva de luz, debemos indicar a CIAO las regiones correspondientes a la fuente y el fondo a utilizar. Para ello, en lugar de introducirlas en una expresión (como hacíamos con *evselect*), podemos utilizar archivos de regiones en formato CIAO generados directamente por **ds9**.

Cargando en **ds9** la lista de eventos reprocesada:

```
> ds9 acis_repro_evt2.fits &
```

Generamos una región para la fuente (por ejemplo, un círculo) y la salvamos en formato **CIAO**, bajo el nombre **src.reg**. Repetimos lo mismo para el fondo, salvándolo bajo el nombre **bkg.reg**.

Con las regiones definidas y guardadas, procedemos a extraer la curva de luz usando **dmextract**, para, por ejemplo, obtener una curva de luz agrupada cada 100 segundos en el archivo **src_lc.fits**:

```
> punlearn dmextract
> pset dmextract infile="acis_repro_evt2.fits[sky=region(src.reg)][bin time>:::100]"
> pset dmextract outfile="src_lc.fits"
> pset dmextract bkg="acis_repro_evt2.fits[sky=region(bkg.reg)]"
> pset dmextract opt="lrc1"
> dmextract
```

Ayuda/Guía: <https://cxc.harvard.edu/ciao/ahelp/dmextract.html>

→ BÚSQUEDA DE VARIABILIDAD

La tarea **glvary** implementa el algoritmo de Gregory-Loredo para la búsqueda de variabilidad en una curva de luz. La tarea genera dos archivos de salida: una tabla con los resultados del estadístico **P** y una curva de luz que incluye los niveles de confianza a $3\text{-}\sigma$, correspondiente al valor óptimo de agrupamiento para la detección de variabilidad. En el caso que el estadístico **P** < **0.5**, la fuente es considerada como no variable, mientras que valores de **P** > **0.9** corresponden a fuentes variables.

Veamos un ejemplo de búsqueda de variabilidad en la región dada por el archivo **src.reg** que corresponde a una fuente ubicada en el **CCD 3** (**ccd_id=3**). Tengan en cuenta que en sus datos, la fuente de interés podría estar ubicada en otro CCD, por lo que habría que modificar este número.

```
> dither_region infile=pcad_asol1.fits region="region(src.reg)" \
    outfile=fracarea.fits

> glvary infile="acis_evt2.fits[sky=region(src.reg),ccd_id=3]" \
    effile="fracarea.fits[cols time,dtf=fracarea]" \
    outfile=gl_prob.fits lcfile=lc_prob.fits
```

Ayuda/Guía: <https://cxc.harvard.edu/ciao/ahelp/glvary.html>

→ ESPECTROS

Para extraer espectros de una región incluida en el archivo **src.reg** (en formato CIAO), considerando un fondo incluido en el archivo **bkg.reg**, puede utilizarse la tarea **specextract**:

```
> punlearn specextract
> pset specextract infile="acisf00459_repro_evt2.fits[sky=region(src.reg)]"
> pset specextract outroot=spec
> pset specextract binspec=16
> pset specextract bkgfile="acisf00459_repro_evt2.fits[sky=region(bkg.reg)]"
> pset specextract weight=no correctpsf=yes
> specextract
```

El parámetro **weight=no** es utilizado para calcular el área efectiva correspondiente a una fuente puntual, acompañado por el parámetro **correctpsf=yes**. En caso de tratarse de una fuente extendida, se utilizan los valores por defecto: **weight=yes** y **correctpsf=no**. El parámetro **binspec** corresponde al agrupamiento mínimo de fotones por cada canal, mientras que **outroot** es el prefijo de salida de todos los archivos que se generan (espectros, matrices y grupo que, en este caso, se denominará: **spec_grp.pi**, y será el archivo que cargaremos en *XSPEC* para ajustar el espectro de la fuente).

Ayuda/Guía: <https://cxc.harvard.edu/ciao/ahelp/specextract.html>