

```

# Reducción de espectros de una sola ranura
#
#*****
#*****
# Previo a cualquier proceso de reducción, con la tarea imheader buscamos,
# en cualquier imagen, el tamaño del CCD y
# los valores de la ganancia y del ruido de lectura.
#
# En este caso la ganancia del CCD es 3.9, la relación señal a ruido es de 4.6 y
# el tamaño de la imagen es [394,576].
#*****
#*****
#
# CORREGIMOS POR OVERSCAN Y TRIMMING
# -----
#
# PASO 1: encontrar la región de overscan de la imagen.
# -----
#
# La región del overscan corresponde a píxeles virtuales que resultan de hacer leer al CCD valores
# adicionales a los que tiene físicamente. En estos valores solo hay ruido y un valor sistemático que
# agrega la electrónica (ese valor es el valor del overscan).
#
# Dicha región se ubica desplegando una imagen con el ds9 y teniendo en cuenta el tamaño de
# la imagen. Es recomendable utilizar un flat.
#
# Para el caso de un flat tomado en CASLEO con el espectrógrafo Boller and Chivens
# podemos ver que la región utilizada del CCD es [1:370,1:576],
# lo que significa que el overscan está establecido en [371:394,1:576]
# En este caso particular vamos a tomar la región que esta en [385:394,1:576]
# porque en la región [379:384,1:290] se observa "algo raro"
# (una columna de píxeles brillantes).
#
# PASO 2: Establecer la región de los bordes que se desean cortar.
# -----
#
# En general es suficiente cortar entre 5 y 10 píxeles de cada lado.
#
#.....
#
# Para corregir por overscan y trimming vamos a utilizar al tarea CCDPROC
#
# Cargamos los paquetes

noao
imred
ccdred

# Primero generamos un archivo que contenga una lista de todas las imágenes (bias,
# flats, objetos y comparaciones) con el comando

```

```
ls *.fit > Todos.lst
```

```
# Ahora generamos un archivo que contenga una lista de todas las imágenes que estarán
# corregidos con el comando
```

```
cp Todos.lst OTtodos.lst
```

```
# Editamos este archivo para que los nombres de los objetos sean de la forma OT***.fits
#
# Ahora con la tarea CCDPRO corregimos por overscan todas las imágenes de manera
# interactiva
#
```

```
ccdproc @Todos.lst output=@OTtodos.lst ccdtype=" " fixpix- overscan+ trim+ zerocor-
darkcor- flatcor- readaxis=line biassec=[385:394,1:576] trimsec=[4:367,4:572]
interactive+ function="legendre" order=3
```

```
# Una vez que corremos la tarea, por pantalla pregunta
# si queremos ajustar el overscan de modo interactivo.
# A lo que debemos contestar que si.
# Luego se despliega una terminal gráfica de iraf, la irafterm.
# Sobre esta terminal vamos a ajustar el polinomio al overscan.
# Lo mas recomendable es utilizar una función de legendre de orden 3.
# El RMS del ajuste debe ser ~0.5 o menor.
```

```
#
# Los parámetros de ajuste se pueden modificar desde la irafterm. Por ejemplo:
# para cambiar el orden del polinomio hay que tipear lo siguiente
# sobre la irafterm
# :o 4
# donde el 4 representa el orden nuevo que le queremos asignar
# al polinomio de ajuste.
```

```
#
# Para ver todas las opciones se tipea ? sobre la irafterm
#
```

```
*****
*****
```

```
#
# CORRECCION POR BIAS
# -----
```

```
#
# PASO 1: Combinar todos los bias (corregidos por overscan).
# -----
```

```
#
# Para esto se utiliza la tarea ZEROCOMBINE.
#
```

```
# PASO 2: Restar el Bias promedio de todas las imágenes.
# -----
```

```
#
# Para esto se utiliza la tarea CCDPROC
```

```

#
#.....
#
# Si es necesario, establecemos los parámetros que vienen por defecto de la tarea zerocombine
unlearn zerocombine

# Primero generamos un archivo que contenga una lista de todos los bias con el comando
ls OTbias*.fits > OTbias.lst

# Ahora PROMEDIAMOS LOS BIAS
zerocombine @OTbias.lst rdnoise=4.6 gain=3.9

# Ahora editamos el archivo OTtodos.lst y borramos los bias de la lista.
# Generamos el archivo que contenga una lista de todas las imágenes que estarán
# corregidos con el comando

cp OTtodos.lst BOTtodos.lst

# Editamos este archivo para que los nombres de los objetos sean de la forma BOT***.fits

# RESTAMOS EL BIAS

ccdproc @OTtodos.lst output=@BOTtodos.lst ccdtype=" " fixpix- overscan- trim- zerocor+
darkcor- flatcor- illumcor- fringe- readcor- scancor- readaxis=line zero=Zero.fits

#*****
#*****
#
# CORRECCION POR FLAT
# -----
#
# PASO 1: Combinar los flats (corregidos por overscan y bias)
#
# Para esto se utiliza la tarea FLATCOMBINE.
#
# PASO 2: Normalizar el Flat promedio
# -----
#
# Esto es necesario para poder corregir por las irregularidades propias del ccd.
# Para esto se utiliza la tarea RESPONSE
#
# PASO 3: Dividir todas las imágenes por el Flat promedio normalizado.
# -----
#
# Para esto se utiliza la tarea CCDPROC
#
#.....

```

```

#
# Primero generamos un archivo que contenga una lista de todos los flats con el comando

ls BOTffl*.fits > BOTflat.lst

# Ahora PROMEDIAMOS LOS FLATS

unlearn flatcombine

flatcombine @BOTflat.lst process- subsets- rdnoise=4.6 gain=3.9

# NORMALIZAMOS EL FLAT
#
# Cargamos los paquetes

twodspec
longslit

#
# Establecemos los parámetros que vienen por defecto la tarea response

unlearn response

response calibrat=Flat.fits normaliz=Flat.fits response=NFlat.fits interactive+

# Una vez que corremos la tarea se despliega la irafterm, en la cual debemos ajustar
# un polinomio a la respuesta del ccd para poder normalizar el Flat.
# Al dividir el Flat por el polinomio, logramos quedarnos con las irregularidades
# del ccd debidas a cuestiones externas al mismo.
# Para lograr un buen ajuste se pueden modificar ciertos parámetros como ser:
# el orden del polinomio, los puntos reyectados, la cantidad de iteraciones
#
# Como lo que estamos tratando de ajustar es la respuesta del ccd, el orden del
# polinomio no debe ser demasiado alto.
# Es decir que el "ruido" no debe ser ajustado.
#
# Ahora editamos el archivo BOTtodos.lst y borramos los flats de la lista.
# Generamos el archivo que contenga una lista de todas las imágenes que estarán
# corregidos con el comando

cp BOTtodos.lst FBOTtodos.lst

# Editamos este archivo para que los nombres de los objetos sean de la forma FBOT***.fits
#
# DIVIDIMOS POR FLAT

ccdproc @BOTtodos.lst output=@FBOTtodos.lst ccdtype=" " fixpix- overscan- trim- zerocor-
darkcor- flatcor+ illumcor- fringe- readcor- scancor- readaxis=line flat=NFlat.fits

#*****

```

```

#*****
#
# EXTRACCION DE LOS ESPECTROS
# -----
#
# PASO 1: Encontrar el espectro, es decir la apertura.
# -----
#
# Esto se puede hacer manualmente examinando un corte eje espacial e indicando
# el pico apropiado con un cursor, o puede hacerse automáticamente si el
# espectro apropiado es el pico mas fuerte presente.
#
# PASO 2: Definir las ventanas de extracción y del fondo del cielo.
# -----
#
# En la practica, esto se realiza especificando el tamaño de la ventana de extracción
# en términos del numero de píxeles a la izquierda del centro del perfil de la apertura,
# y el numero de píxeles a la derecha del mismo.
# De forma similar, la región de fondo del cielo se define en términos de una región
# a la izquierda y a la derecha del centro de perfil.
# Uno puede entonces examinar estas regiones sobre un corte a lo largo del eje espacial,
# y redefinirlas si es necesario.
#
# PASO 3: Trazar el centro del perfil espacial en función del eje de dispersión.
# -----
#
# Aunque supongamos que el eje espacial esta exactamente a lo largo de una fila o
# columna, el espectro no sera exactamente perpendicular al eje espacial (es decir,
# el espectro estelar no es exactamente paralelo a lo que estamos tomando como la teoría).
# En su lugar, el centro exacto del perfil espacial se desplazara ligeramente con la
# ubicación a lo largo del eje de dispersión. Hay por lo menos tres razones para esto:
# (a) las ópticas de cámara introducen distorsiones que serán peores a lo largo del eje
# mas largo (dispersión),
# (b) las redes no se sitúan exactamente en sus celdas,
# y (c) la refracción atmosférica diferencial hará que el extremo azul del espectro
# sea desplazado a lo largo de la ranura mas cerca del cenit que el del extremo rojo
# del espectro. Este ultimo efecto sugiere que podemos esperar que el angulo formado
# por el espectro y el eje de dispersión difieran, a menudo de manera significativa,
# de una exposición a otra.
#
# PASO 4: Sumar el espectro dentro de la ventana de extracción, restando el cielo.
# -----
#
# En cada punto a lo largo del eje de dispersión, los datos dentro de la apertura
# de extracción (centrada espacialmente en base al valor que la traza esta en ese punto)
# se suma, y el fondo del cielo se resta.
#
# Para esto se utiliza la tarea APALL
#
#.....

```

```

#
# Cargamos el paquete

apextract

# Establecemos los parámetros que vienen por defecto en el paquete ccdred

unlearn apall

# EXTRACCION DE LOS ESPECTROS DE CIENCIA
# -----
#
# Ahora generamos un archivo que contenga una lista de todos los espectros que serán
# extraídos con el comando

ls FBOTobj*.fits > FBOTobj.lst
cp FBOTobj.lst EFBOTobj.lst

# Editamos este archivo para que los nombres de los objetos sean de la forma
# EFBOTobj.fits

apall @FBOTobj.lst nfind=1 output=@EFBOTobj.lst backgro=fit weights=variance saturat=16000
readnoi=4.6 gain=3.9

#
# Cuando corremos la tarea, por pantalla nos pregunta:
# Si queremos buscar la apertura,
# Si queremos redimensionar la apertura, y
# Si queremos editar la apertura.
#
# A todas estas preguntas contestamos que sí.
# Luego se despliega el irafterm donde nos muestra un corte espacial de la apertura.
# En esta terminal con las presionando las letras "l" (lower, izquierda) y "u"
# (upper, derecha) podemos redimensionar la apertura.
#
# Recordar que presionando ? sobre el irafterm se accede al menú de ayuda.
#
# Una vez que terminamos de definir la posición y el tamaño de la ranura debemos
# establecer los parámetros del fondo del cielo.
# Para ello presionamos (sobre el irafterm) la letra "b".
# Sobre este nuevo despliegue debemos ajustar las posiciones y los tamaños de las
# ventanas de fondo del cielo y ajustarles un polinomio.
# En general se utiliza un polinomio de chevyshev de orden 2.
# Los parámetros de ajuste se pueden modificar interactivamente en la irafterm.
# Cuando terminamos salimos del ajuste de fondo del cielo con la letra "q".
#
# Una vez definida la apertura y el fondo del cielo hay que pasar al ajuste de la traza.
# Para ello salimos del ajuste de la apertura con la letra "q".
# Luego, sobre la irafterm nos pregunta:
# Si queremos trazar la apertura,

```

```

# Si queremos ajustar la posición de la traza, y
# Si queremos hacer el ajuste de manera interactiva.
#
# A todas estas preguntas contestamos que si.
#
# Generalmente el ajuste de la traza se realiza con un polinomio de legendre de grado 3.
# Lo ideal es que RSM < 0.02.
#
# Una vez ajustada la traza salimos con la letra "q".
# Luego, sobre la irafterm nos pregunta:
# Si queremos escribir la apertura en el directorio DATABASE,
# Si extraemos el espectro de la apertura,
# Si queremos ver el espectro extraído, y
# Si queremos ver el espectro extraído de la apertura.
#
# A todas estas preguntas contestamos que si.
#
#.....
# EXTRACCION DE LOS ESPECTROS DE COMPARACION
# -----
#
# Ahora generamos un archivo que contenga una lista de todos las comparaciones que serán
# extraídas con el comando

ls FBOTcomp*.fits > FBOTcomp.lst
cp FBOTcomp.lst EFBOTcomp.lst

# Editamos este archivo para que los nombres de los objetos sean de la forma EFBOTcomp.fits
# Ademas hay que ver que entre el archivo FBOTobj.lst y el FBOTcomp.lst cada
# comparación se corresponda con su objeto
# (es decir que estén listados en el mismo orden).
# En el caso en el que una comparación se corresponda con 2 objetos distintos,
# debe estar repetido en la lista en su correspondiente orden.
#
# Para extraer los espectros de comparación no es necesario corregirlos por el
# fondo del cielo.
# Vamos a extraerlos con los mismos parámetros que definimos en los espectros de ciencia.

apall @FBOTcomp.lst nfind=1 output=@EFBOTcomp.lst reference=@FBOTobj.lst interact-
find- recente- resize- edit- trace- fittrac- extras- review- bkg- saturat=16000 readnoi=4.6 gain=3.9

#*****
#*****
#
# CALIBRACION EN LONGITUD DE ONDA
# -----
#
# CALIBRACION DE LAS COMPARACIONES
# -----
#

```

```

# PASO 1: Determinar la solución de dispersión.
# -----
#
# Hay que encontrar una transformación para relacionar los píxeles en la dirección de
# dispersión con la longitud de onda.
# Para hacerlo hay que asignarle las longitudes de onda correspondientes a las líneas de la lámpara
# de comparación.
# Una vez identificadas las líneas hay que ajustar un polinomio que relacione los píxeles
# con las longitudes de onda.
#
# Para esto se utiliza las tareas IDENTIFY y REIDENTIFY
#
#.....
#
#
# Vamos a utilizar una lámpara de comparación como prueba para la calibración

imcopy EFBOTcomp01.fits comp_ref.fits

# Hacemos la identificación de las líneas en una lámpara de comparación

identify comp_ref.fits

#
# Cuando corremos la tarea se despliega el espectro de la lámpara.
# Con la letra "m" asignamos las longitudes de onda a las líneas previamente
# identificadas. Conviene hacer esto con cuatro o cinco líneas,
# asegurándonos de tomar al menos una en cada borde. Luego, con la letra "f" ajustamos
# el polinomio y salimos con la "q". Ahora, con la "l" traemos todas las líneas que
# están guardadas en la base de datos del iraf y volvemos a ajustar el polinomio con la
# letra "f".
# El polinomio de ajuste generalmente es mayor o igual a 3 y
# el RMS debe ser, en lo posible, menor a 0.2. Y en el gráfico de velocidades,
# los puntos deben verse distribuidos al azar.
#
# Una vez ajustado el polinomio salimos del ajuste con la letra "q".
# Luego, sobre la xgterm nos pregunta:
# Si queremos guardar las identificaciones en el DATABASE.
#
# A lo que contestamos que si.
#
# Ahora identificamos las líneas de todas las demás lámparas tomando como referencia
# la primera

reidentify reference=comp_ref.fits images=@EFBOTcomp.lst interact+ newap-

#
# CALIBRACION DEL ESPECTRO DE CIENCIA
# -----
#

```



```

# Cargamos el paquete
onedspec

# Establecemos los parámetros que vienen por defecto en el paquete ccdred
unlearn onedspec

# A cada objeto le asignamos su espectro de referencia para la calibración.
refspec @EFBOTobj.lst referen=@EFBOTcomp.lst select=match override+

# Generamos un archivo que contenga una lista de todas los objetos que estarán
# calibrados en longitud de onda con el comando

cp EFBOTobj.lst Wobj.lst

# Editamos este archivo para que los nombres de los objetos sean de la forma Wobj.fits
#
# Aplicamos la calibración.

dispcor @EFBOTobj.lst output=@Wobj.lst

#*****
#*****
#
# CALIBRACION EN FLUJO
# -----
#
# Las que vamos a utilizar en este ejemplo están en el directorio
# /home/iraf/noao/lib/onedstds/spec16cal/
# Para ver los distintos tipos de estrellas estándares hay utilizar la sentencia
#
# page onedstds$README
#
# PASO 1:
# -----
#
# Estimar la cantidad de cuentas por longitud de onda y asignarle el valor de flujo
# correspondiente. Esta información se guarda en un archivo.
#
# Esto lo hacemos con la tarea STAND.
#
# PASO 2: Ajustar la función de sensibilidad como una función de la longitud de onda.
# -----
#
# El ajuste conviene hacerlo de manera interactiva usando el archivo del paso 1.
# Para realizar el ajuste es necesario corregir por extinción atmosférica.
# Para ello se puede utilizar la tabla estándar de extinción que usted haya adoptado,
# o puede tratar de determinar la extinción empírica de sus datos.

```

```

# Puede realizar "cambios grises" de una observación en particular,
# eliminar puntos u observaciones, e interactuar generalmente hasta que tenga un
# ajuste satisfactorio a los puntos.
#
# Esto lo hacemos con la tarea SENSFUNC.
#
# PASO 3: Aplicar la función de sensibilidad a la ciencia.
# -----
#
# Esto lo hacemos con la tarea CALIB
#
#.....
#
# CALIBRACION DE LA ESTRELLA ESTANDAR
# -----
#
# Suponiendo que la estrella estándar que utilizamos corresponde a HR 4468 muestreamos el
# continuo

stand Wstd.fits star_nam=hr4468blue mag=4.68 magband=V
extinct=/home/yael/Observaciones/CASLEO/casleoext
caldir=/iraf/iraf/noao/lib/onedstds/spec16cal/

# donde Wstd.fits corresponde al archivo del espectro correspondiente a la estándar de
# flujo observada.
#
# Cuando corremos la tarea nos pregunta:
# Si queremos editar las bandas.
#
# A lo que contestamos que si.
#
# Es importante que las bandas caigan sobre el continuo, por lo que las que caigan
# sobre las líneas debemos borrarlas con la letra "d"
# Salimos con "q".
#
# Determinamos las funciones de sensibilidad del detector y de extinción.

sens extinct=/home/yael/Observaciones/CASLEO/casleoext graphs="irs"

# Cuando corremos la tarea nos pregunta:
# Si queremos ajustar la apertura interactivamente.
#
# A lo que contestamos que si.
#
# Sobre el irafterm hay que ajustar el polinomio a la función de sensibilidad.

# CALIBRACION DE LA CIENCIA
# -----
#
# Generamos un archivo que contenga una lista de todas los objetos que estarán

```

```
# calibrados en flujo con el comando
```

```
cp Wobj.lst FWobj.lst
```

```
# Editamos este archivo para que los nombres de los objetos sean de la forma FWobj.fits
```

```
calib @Wobj.lst output=@FWobj.lst extinction=/home/yael/Observaciones/CASLEO/casleoext
```

```
#####
```

```
#####
```

```
#
```

```
# NORMALIZAR UN ESPECTRO
```

```
#
```

```
# Para normalizar un espectro se pueden utilizar los comandos de la tarea splot.
```

```
# Primero graficamos el espectro (que puede o no estar calibrado en flujo)
```

```
splot FWobj08.fits
```

```
# Sobre el irafterm presionamos la letra "t". Al pie de la terminal nos lista una serie  
# de opciones, en este caso elegimos la opción NORMALIZE.
```

```
# Si queremos guardar el espectro normalizado como una nueva imagen se presiona  
# la letra "i".
```

```
# Una vez ajustado el polinomio salimos con la letra "q".
```