



Construcción de una base de datos de espectros de alta resolución de estrellas frías: clasificación espectral, parámetros estelares, abundancias

Trabajo Académicamente Dirigido
Departamento de Astrofísica de la UCM
Tutor: David Montes Gutiérrez
Alumno: Jorge Hernández Arce
Fecha: 27 de Septiembre de 2011

Índice de contenidos

<i>Introducción.....</i>	<i>pág 3</i>
<i>Capítulo 1: Procesamiento de espectros con IRAF.....</i>	<i>pág 4</i>
<i>Capítulo 2: Creación de una base de datos con Saada.....</i>	<i>pág 16</i>
<i>Capítulo 3: Espectros.....</i>	<i>pág 18</i>
<i>Referencias.....</i>	<i>pág 20</i>

Construcción de una base de datos de espectros de alta resolución de estrellas frías: clasificación espectral, parámetros estelares, abundancias

Introducción

En esta breve introducción hablaré de los objetivos del trabajo.

El principal objetivo es el procesamiento de espectros Echelle de alta resolución previamente reducidos, mediante el proceso que explicare más adelante con detalle, para que queden en un formato uniforme que permita utilizarlos con objetivos científicos. Como se sabe, el principal interés de los espectros de estrellas es el cálculo de abundancias mediante el cálculo de la anchura equivalente de sus líneas.

Con el formato en el que quedarán los espectros una vez su procesamiento haya concluido podremos determinar dichas abundancias y otros parámetros mediante el uso de programas que calculan de forma automática de anchura equivalente y abundancias.

El procesamiento de los espectros se realiza mediante un proceso largo pero sencillo utilizando la herramienta *splot* de IRAF. En pocas palabras, para procesar un espectro debemos cortar las partes inservibles provocadas por los bordes del espectrógrafo, antes de normalizar el continuo a la unidad mediante una función polinómica para después pegar los ordenes juntos (los espectros vienen en trozos de unos 60 Angstroms llamados ordenes), y finalmente corregirlos de efecto Doppler. Haciendo todo esto ya tendríamos el espectro procesado y listo para ser utilizado.

Un objetivo secundario del trabajo sería la creación de una base de datos de espectros disponibles desde internet. Se ha conseguido aprender a utilizar los programas necesarios, pero la base de datos está aún en proyecto. El programa que se utilizará para realizar esto se llama SaadaDB y ha sido desarrollado por el departamento de Astrofísica de la universidad de Estrasburgo.

Hablemos un poco ahora de los espectros que vamos a procesar. Proceden del Observatorio Hispano-Alemán de Calar Alto, en Almería. Han sido realizados con el espectrógrafo FOCES¹ (Fibre Optics Cassegrain Echelle Spectrograph) del telescopio de 2.2 metros disponible en el observatorio. Se ha usado un detector de CCD de 2048x2048 píxeles. FOCES cubre un rango desde 3600 hasta 10700 Angstroms en más de 100 ordenes. Su dispersión recíproca varía entre 0.03 y 0.07 Å/píxel. Su resolución espectral, que determinamos como la anchura a media altura (FWHM) se encuentra en un rango de entre 0.09 y 0.26 Angstroms. Finalmente, la resolución $R = 40000$ ¹.

¹Para más información sobre FOCES consultar Pfeiffer et al. 1998.

²Para más información sobre los espectros FOCES utilizados consultar Montes et al. 2001a, b; Lopez-Santiago et al. 2006, 2010; Martinez-Arnaiz et al. 2010; Maldonado et al. 2010.

Capítulo 1: Procesamiento de los espectros con IRAF

En este capítulo daré una descripción paso a paso de los pasos que sigo para procesar un espectro en IRAF.

IRAF, acrónimo de *Image Reduction and Analysis Facility*, es una colección de software desarrollado en NOAO (National Optical Astronomy Observatory), es decir el observatorio nacional de astronomía óptica de los Estados Unidos de América. Su función básica es la reducción de distintos tipos de imágenes astronómicas, para lo cual se utilizan distintas herramientas e iteraciones del programa que se pueden encontrar en los distintos paquetes desarrollados por NOAO y otras sedes científicas.

En mi caso, en particular voy a procesar espectros Echelle obtenidos con el espectrógrafo FOCES, la mayoría de ellos en 2005. Para ello utilizare la herramienta “splot” de IRAF dentro del paquete “ONED” desarrollado por NOAO. Es una herramienta que permite visualizar los espectros y a la vez ir corrigiéndolos para que pasen de ser imágenes “raw” a espectros procesados listos para ser utilizados con objetivos científicos.

Lo primero es acceder al programa. Nos situamos en la carpeta donde tengamos instalado el IRAF y tecleamos:

```
xgterm &
```

Esto nos abrirá una terminal tipo xgterm, y añadir “&” nos permitirá seguir utilizando normalmente la terminal que teníamos abierta hasta ahora. En la ventana xgterm, tecleamos:

```
c1
```

Este comando ejecuta IRAF. Antes de comenzar con los espectros, veamos una serie de comandos básicos de IRAF que he ido utilizando a la vez que los más complejos y específicos para espectros que describiré posteriormente. Los comandos que más he utilizado son:

ccdlist: nos saca en pantalla todos los archivos .fits que haya en la carpeta en la que nos encontremos.

w → e → e: con la primera letra entramos en el modo zoom, y con las posteriores letras “e” marcamos los límites de la ventana que queremos visualizar.

w → a: elimina los zooms que hayamos hecho previamente, volviendo a la ventana original.

Barra espaciadora: nos da la posición en longitudes de onda donde se encuentra el cursor.

k: sirve para hacer un ajuste gaussiano de una línea y así obtener datos interesantes como su longitud de onda central o su anchura equivalente.

Ya podemos proceder a visualizar un espectro. Para explicar el procesamiento tomaré como ejemplo el espectro llamado “nHD182488_n2_FOCES_jul05.fits”, que es un espectro tipo seleccionado de forma aleatoria de entre todo el conjunto de espectros que he procesado.

Visualizamos entonces este espectro tecleando:

```
plot nHD182488_n2_FOCES_jul05.fits
```

Esto es lo que vemos:

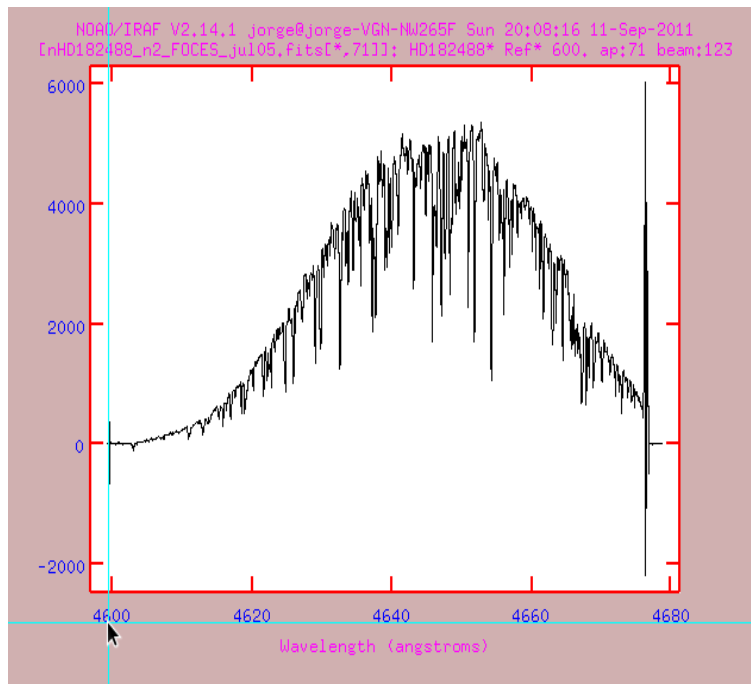


Figura 1

Como se puede apreciar en la figura 1, es el orden número 71 de un total de 111. Podemos ver en la parte derecha de la imagen (también ligeramente en la parte izquierda) un trozo claramente distinto del resto del espectro, y que además produciría la impresión de existencia de líneas de emisión si lo dejará como está. Son trozos como este los que hay que recortar de los ordenes. Recortar ordenes se hace con el comando “scopy”. Dispongo de un “script” con el que puedo hacer estos cortes para todos los ordenes a la vez. Previamente, eso sí, habrá que mirar todos los ordenes uno a uno para saber donde tendremos que hacer el corte. Con esos datos, podemos completar el “script”. En mi caso, el que he utilizado se llama “scopy_terminado.cl” y sigue la forma:

```
scopy ("nHD182488_n2_FOCES_jul05.fits",  
"nHD182488_n2_FOCES_jul05_c.fits", w1=9139., w2=9268.,  
apertures="10", bands=" ", beams="",  
apodulus=0, format="multispec", renumber=no, offset=0,  
clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

```
scopy ("nHD182488_n2_FOCES_jul05.fits",  
"nHD182488_n2_FOCES_jul05_c.fits",          w1=9001.,          w2=9123.,  
apertures="11", bands=" ", beams="",  
apodulus=0,      format="multispec",      renumber=no,      offset=0,  
clobber=yes,  
merge=yes, rebin=yes, verbose=no)
```

Donde $w1$ y $w2$ son los límites que le ponemos en cada orden a la longitud de onda en angstroms, “apertures” es el número de orden, y el resto son parámetros que establecemos para que el programa realice los cortes de la manera deseada. Este “script” se puede aplicar a muchos espectros a la vez, utilizando un @-file.

Un @-file es básicamente un listado de archivos (en nuestro caso espectros) a los que queremos aplicar el mismo proceso. Escribiendo los nombres de los espectros y poniendo “@-nombre del archivo” donde te pide el nombre del espectro, aplicaremos los cambios a todos los espectros de la lista. Será necesario también crear un archivo de salida en el que pondremos los nombres de los espectros una vez modificados. Escribimos, “@-nombre del fichero de salida”, e IRAF nos realizará el proceso para todos los espectros de la lista, generando en el proceso nuevos espectros ya cortados con los nombres que le hayamos especificado.

En mi caso para indicar que les he cortado las partes malas, he añadido una c al final del nombre de cada espectro quedando de la forma:

nHD182488_n2_FOCES_jul05_c.fits

Una vez realizados los cortes, los ordenes se habrán librado de las partes malas, quedando de la siguiente forma:

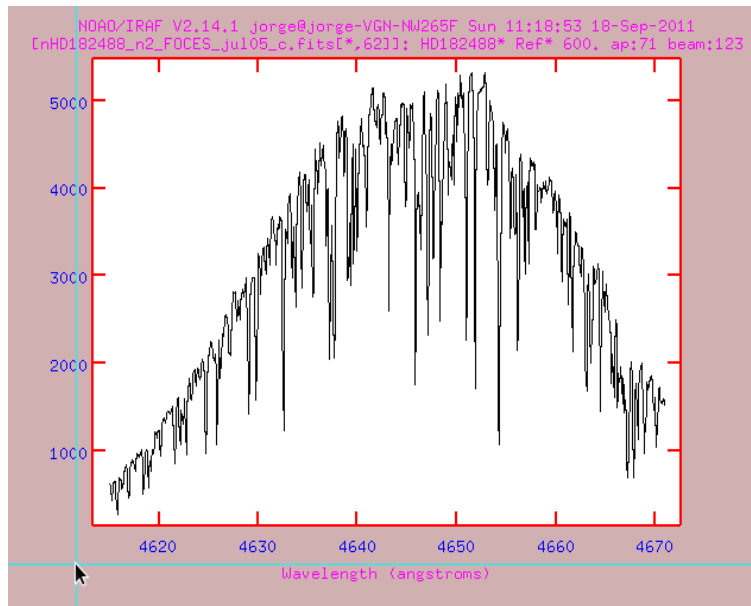


Figura 2

Como puede apreciarse en la figura2, las molestas zonas defectuosas de los bordes han desaparecido, quedándonos solo con las partes útiles. Al analizar los espectros, observamos que hay una serie de ordenes que tienen demasiado ruido como para ser utilizables (del 1 al 10 y del 94 en adelante). El “script” de “scopy” nos elimina también estos ordenes, lo que nos reduce el número de ordenes original de 111 a un total de 84.

Esto facilita enormemente el siguiente proceso, en el que realizaré la normalización de los ordenes mediante el comando de IRAF “continuum”. En la terminal *xgterm* escribimos:

```
epar continuum
```

La herramienta *epar* nos da la opción de determinar los parámetros del comando *continuum* antes de realizar la normalización. Podemos determinar, por ejemplo, los ordenes en los que queremos realizar la normalización, si no son todos.

Esto es lo que IRAF nos devuelve cuando tecleamos lo anterior:

```
input      = nHD182488_n2_FOCES_jul05_c.fits  Input images
output     = nHD182488_n2_FOCES_jul05_n.fits  Output images
(lines    =
           10-94) Image lines to be fit
(bands    =
           1) Image bands to be fit
(type     =
           ratio) Type of output
(replace=
           no) Replace rejected points by fit?
(wavesca=
           yes) Scale the X axis with wavelength?
(logscal=
           no) Take the log (base 10) of both
axes?
(overrid=
           no) Override previously fit lines?
(listonl=
           no) List fit but don't modify any
images?
(logfile=
           logfile) List of log files
(interac=
           yes) Set fitting parameters
interactively?
(sample   =
           *) Sample points to use in fit
(naverag=
           1) Number of points in sample
averaging
(funcio=
           spline3) Fitting function
(order   =
           3) Order of fitting function
(low_rej=
           2.) Low rejection in sigma of fit
(high_re=
           0.) High rejection in sigma of fit
(niterat=
           5) Number of rejection iterations
(grow    =
           1.) Rejection growing radius in
pixels
(markrej=
           yes) Mark rejected points?
(graphic=
           stdgraph) Graphics output device
```



```
(cursor =          ) Graphics cursor input
ask      =          yes
(mode   =          ql)
```

A la derecha se puede leer una descripción sobre que es cada una de las opciones listadas. Especialmente importantes, bajo mi punto de vista, han sido:

override: determina si se vuelven a normalizar ordenes que habían sido previamente normalizados. Como en mis espectros hay varios ordenes que ya estaban normalizados, es importante poner que no.

interac: hay que poner que sí para que IRAF permita cambiar cosas según vamos normalizando orden a orden (cosas como el orden de la función polinómica que usamos para normalizar, por ejemplo).

Cuando ya tenemos todas las opciones configuradas como más nos interese, tecleamos `:go`, e IRAF nos preguntará si queremos normalizar el primer orden. A partir de ahí, tendremos que ir orden a orden realizando varias tareas para asegurarnos de que la normalización es lo más adecuada posible.

Dentro de cada orden, podremos ir cambiando diferentes cosas, escribiendo algunos comandos en la ventana de imagen de IRAF. A continuación, un listado de los comandos que he utilizado:

:o: Con este comando podemos cambiar el orden de la función polinómica que estamos utilizando para realizar el ajuste.

:low: con esto elegimos el número de puntos que deja el ajuste por debajo del continuo.

:high: es la antítesis del anterior, controla el número de puntos que deja el ajuste por encima del continuo.

s: Al teclear una “s”, IRAF nos pide que volvamos a marcar esta tecla. Esto sirve para determinar ciertas zonas que serán las que utilizará el programa para realizar el ajuste. De esta forma, podemos eliminar zonas malas que no hayan podido ser eliminadas con el recorte realizado por *scopy* debido a que se encontraran en el centro de un orden. Defectos de este tipo pueden deberse a rayos cósmicos, por ejemplo.

t: Sirve para borrar las zonas determinadas previamente con “s”.

f: Reajusta la función con todos los cambios que hayamos hecho. Hay que tener cuidado, porque si hacemos los cambios y no pulsamos “f”, no nos los guardará.

Veamos ahora ejemplos de como podemos utilizar esos comandos para mejorar los ajustes de normalización.

Si seguimos usando nuestro ejemplo del orden 71 del espectro nHD182488_n2_FOCES_ju105, con los valores por defecto, la normalización quedaría:

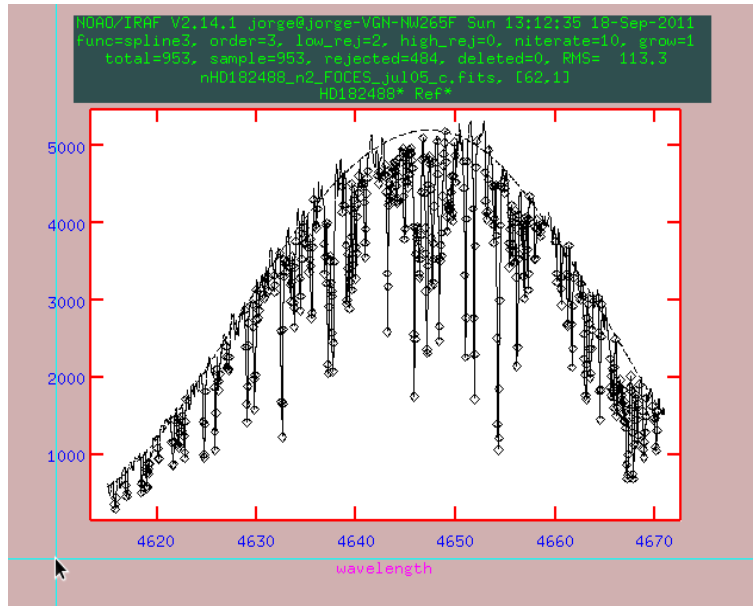


Figura 3

Como puede verse, el ajuste no es muy adecuado en la figura 3, si dejamos los valores por defecto, así que habrá que cambiar cosas. En principio lo que se ve aquí es que hay demasiados puntos por debajo del continuo, por lo que habrá que cambiar el *low_rejection*. Se puede ver que está en 2, si probamos a cambiarlo por 1:

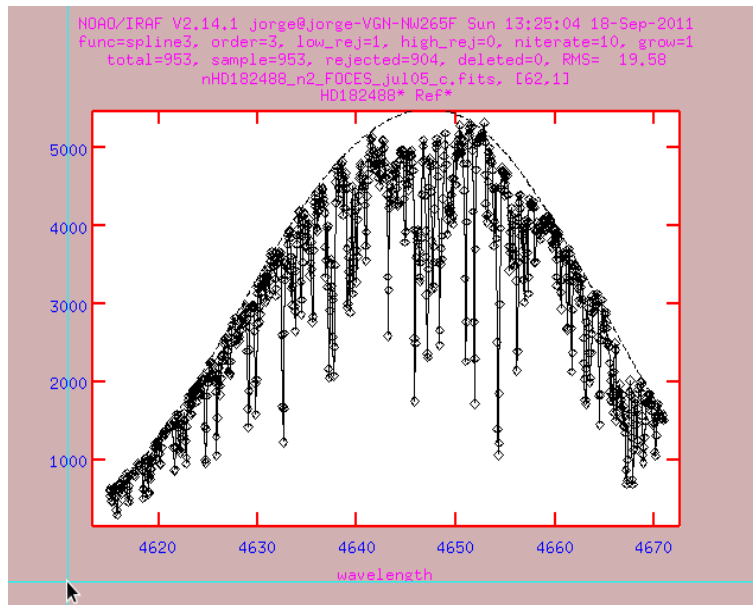


Figura 4

Se puede apreciar una mejora considerable en el ajuste, tanto es así que en este orden en particular no habría que hacer ya más cambios. Para ver ejemplos del comando “s”, recurriremos a otro orden del mismo espectro (figura 5, a continuación).

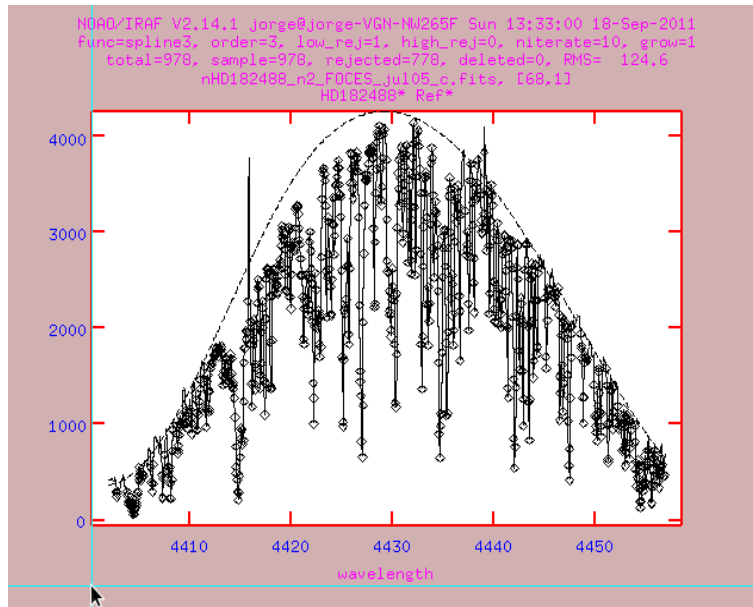


Figura 5

Utilizando el comando s para recortar las zonas de ajuste nos queda:

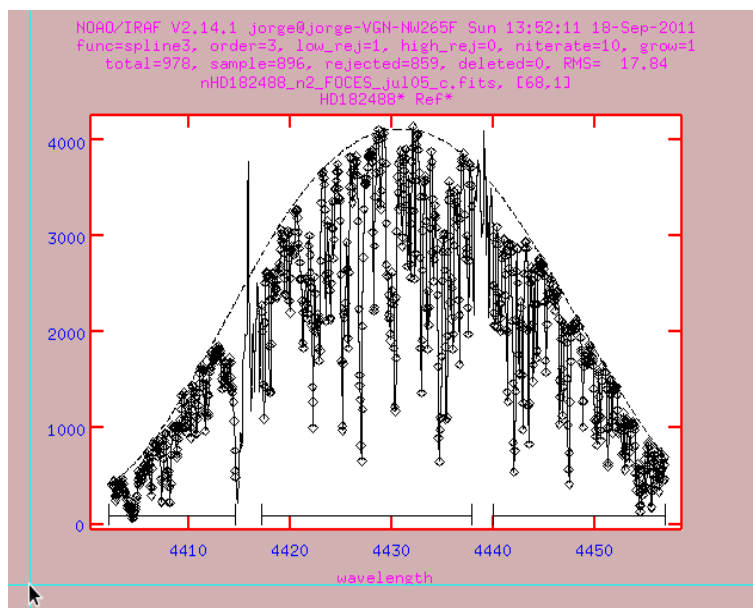


Figura 6

El ajuste ha quedado ya bastante aceptable una vez que hemos eliminado las dos zonas conflictivas que aparecían en el espectro, en torno a 4410 y 4440 Angstroms. Se pueden apreciar en la parte inferior del espectro las zonas que hemos determinado como válidas para el ajuste, aquellas que caen dentro de las líneas horizontales.

Ahora que por fin tenemos normalizados todos los órdenes podemos pasar al siguiente paso, que consiste en pegarlos todos juntos para que así formen un espectro único.

Para ello, usaremos el comando *scombine* de IRAF. De nuevo, lo llamamos tecleando:

```
epar scombine
```

Como en el caso anterior, IRAF nos devuelve una serie de opciones, pero en este caso las he dejado todas por defecto, por lo que no son tan importantes como las opciones de la herramienta *continuum*, y no las citare a continuación.

Este es el espectro que nos queda una vez pegado todo junto:

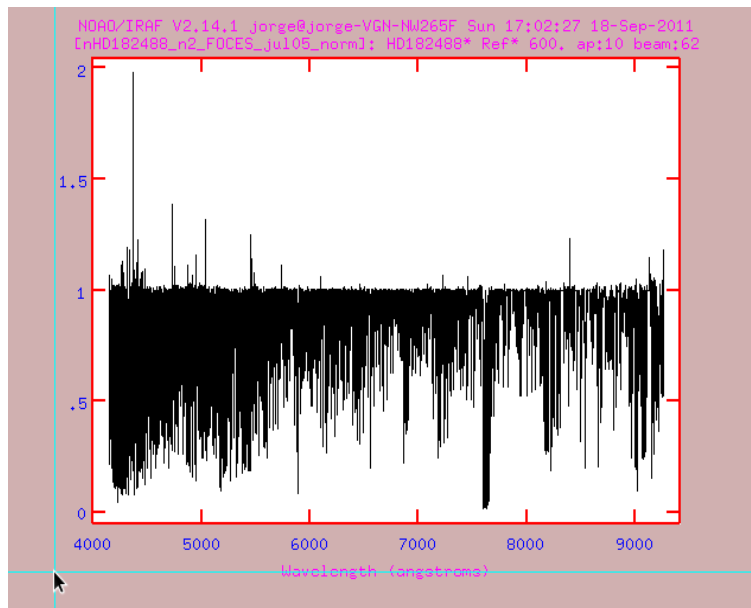


Figura 7

Las líneas que se aprecian por encima del continuo (de valor 1) son los rayos cósmicos y otros efectos que he comentado antes que no se podían eliminar por estar en el centro de los órdenes. Se puede apreciar que tanto los primeros órdenes como sobre todo los últimos (a la izquierda de la imagen, longitudes de onda más cortas) son más propensos a sufrir imperfecciones de estas características. Esto se debe a que, sobre todo en las longitudes de onda cortas, el número de

líneas que aparecen es mucho mayor que en el resto de ordenes, lo que dificulta mucho su normalización y da lugar a estas imperfecciones.

Después de pegar todos los ordenes juntos uno pensaría que el procesamiento del espectro ya ha concluido, pero no será así, ya que todavía queda un ultimo paso muy importante que es corregirlo de efecto Doppler.

Para corregir un espectro de efecto Doppler con IRAF, tenemos que usar dos comandos de manera sucesiva.

El primero es el comando *fxcor*, que, comparando las líneas presentes en nuestro espectro con las de un espectro del Sol (*sunatlas.fits*) nos hallará la velocidad radial de la estrella respecto al Sistema Solar.

Tecleamos en la *xgterm*:

```
epar fxcor
```

IRAF nos responde con las siguientes opciones:

```
objects = nHD182488_n2_FOCES_jul05_p.fits List of object spectra
template= sunatlas List of template spectra
(apertur= *) Apertures to be used
(cursor = ) Graphics input cursor
(continuum= both) Continuum subtract spectra?
(filter = none) Fourier filter the spectra?
(rebin = smallest) Rebin to which dispersion?
(pixcorr= no) Do a pixel-only correlation?
(osample= A5053-5661,A6000-6276,A6330-6864) Object regions to be
correlated ('*')
(rsample= A5053-5661,A6000-6276,A6330-6864) Template regions to
be correlated
(apodize= 0.2) Apodize end percentage
(funcion= gaussian) Function to fit correlation
(width = INDEF) Width of fitting region in pixels
(height = 0.) Starting height of fit
(peak = no) Is height relative to ccf peak?
(minwidt= 3.) Minimum width for fit
(maxwidt= 21.) Maximum width for fit
(weights= 1.) Power defining fitting weights
(backgro= 0.) Background level for fit
(window = INDEF) Size of window in the
correlation plot
(wincent= INDEF) Center of peak search window
(output = salidan61) Root spool filename for output
(verbose= long) Verbose output to spool file?
```

```
(imupdat=          no) Update the image header?
(graphic=          stdgraph) Graphics output device
(interac=          no) Interactive graphics?
(autowri=         yes) Automatically record results?
(autodra=         yes) Automatically redraw fit results?
(ccftype=         image) Output type of ccf
(observa=         lapalma) Observation location database
(continp=         ) Continuum processing parameters
(filtpar=         ) Filter parameters pset
(keywpar=         ) Header keyword translation pset
(mode   =         ql)
```

De todos estos, los más importantes son:

osample: nos da a elegir las regiones de nuestro espectro problema que queremos usar para comparar con el espectro del sol.

rsample: Básicamente lo mismo que el anterior, pero se refiere a las regiones del espectro del Sol que queremos usar.

functio: nos da a elegir el tipo de función que usar para la correlación entre los dos espectros. En nuestro caso hemos usado una gaussiana.

Este comando nos crea al ser ejecutado un archivo de texto en el que figura la velocidad de la estrella. Para el espectro de nHD182488 obtenemos una velocidad relativa al Sol de:

$$V_{REL} = -20.6523 \text{ km/s}$$

Conocido este dato, ya podemos aplicar el último comando que necesitaremos de IRAF, *dopcor*, en el que usaremos la velocidad que acabamos de obtener para desplazar el espectro y así corregir de efecto Doppler las longitudes de onda en las que se centran las líneas que observamos.

Tecleamos en IRAF:

```
epar dopcor
```

Y el nos proporciona las siguientes opciones:

```
input   = nHD182488_n2_FOCES_jul05_p  List of input spectra
output  = nHD182488_n2_FOCES_jul05_fin List of output spectra
redshift=          -20.6523  Redshift or velocity (Km/s)
(isveloc=          yes) Is the redshift parameter a
velocity?
(add     =          yes) Add to previous dispersion
correction?
(dispers=          yes) Apply dispersion correction?
```

```
(flux      =          no) Apply flux correction?  
(factor    =          3.) Flux correction factor (power of  
1+z)  
(apertur=          ) List of apertures to correct  
(verbose=  no) Print corrections performed?  
(mode     =          ql)
```

Al ejecutar correctamente este comando ya tenemos nuestro espectro terminado. Sólo falta comprobar que esta todo correcto. Para ello, usaremos líneas espectrales conocidas.

Si estas líneas aparecen donde deberían, significará que hemos hecho bien la corrección Doppler y habremos terminado de procesar el espectro.

Comprobemos primero la línea de la serie de Balmer $H\alpha$. Conocemos su longitud de onda, que se sitúa en 6562.8 Angstroms. Pues bien, usando la herramienta *plot*:

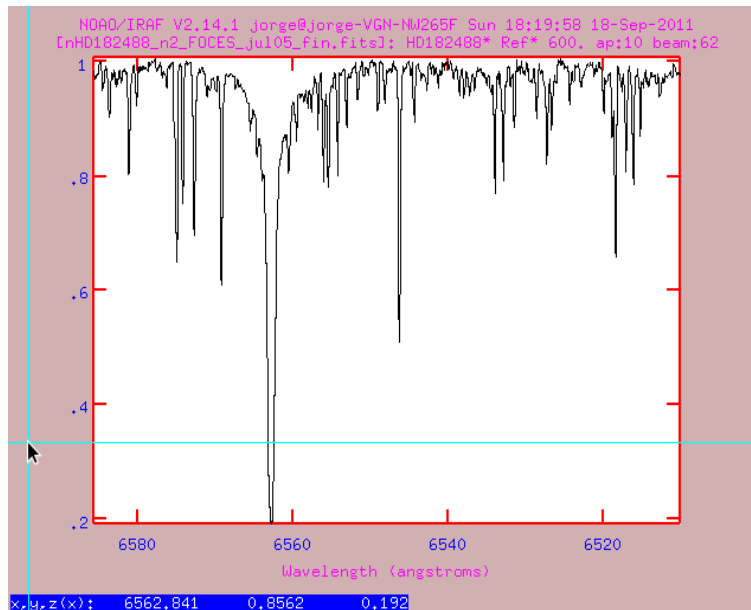


Figura 8

Como se aprecia en la imagen, aparece la línea justo en donde debería. Los números de abajo nos indican la posición aproximada del cursor en el centro de la línea, en Angstroms. Parece que nuestra corrección ha sido un éxito, pero vamos a terminar de comprobarlo usando otras líneas menos conocidas.

Elegimos las líneas del CaII presentes en las longitudes de onda de 8498.1 y 8542.1 Angstroms respectivamente. La imagen que nos queda es la siguiente:

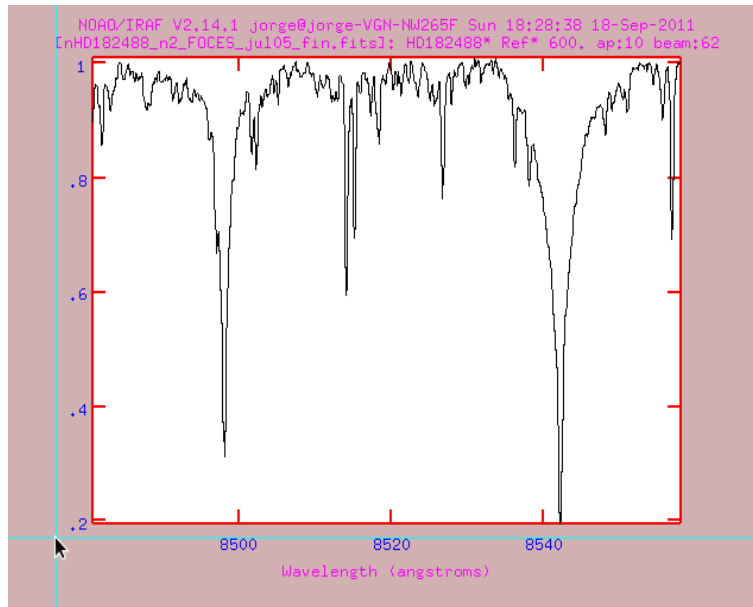


Figura 9

Se puede apreciar que ambas líneas están perfectamente establecidas en sus respectivas posiciones en longitudes de onda (ver figura 9, página anterior).

Tanto este comando como el anterior (*fxcor*) se pueden utilizar con @-files, haciendo que sea mucho más rápido el procesamiento de múltiples espectros.

Con esto hemos comprobado definitivamente que la correlación ha sido correcta, y podemos dar por terminado el procesamiento del espectro, que ya está listo para ser usado con fines científicos.

Capítulo 2: Creación de una base de datos con Saada

Cuando ya tenemos todos nuestros espectros procesados, podemos crear una base de datos de ellos para que sean accesibles desde internet. Para ello vamos a servirnos del programa *SaadaDB*, de la universidad de Estrasburgo, que nos creará una base de datos accesible sin tener que tener instalado el propio programa.

Lo que vamos a detallar a continuación va a ser el proceso que hay que seguir para crear una base de datos con *SaadaDB*.

Antes de empezar, tenemos que tener instalado el programa PostgreSQL, versión 8.0 o superior, ya que es requisito imprescindible para que funcione nuestro programa.

Comencemos pues con los pasos necesarios para crear la base de datos. En primer lugar tenemos que instalar el programa, para lo cual abrimos una terminal y escribimos:

```
% mkdir Database
% cd Database
```

Dentro de esta nueva carpeta, creamos otros cuatro directorios:

```
% mkdir saada
% mkdir saadadb
% mkdir repository
% mkdir apache
```

Cada uno de estas carpetas tendrá un objetivo concreto. La carpeta *saada* es donde instalaremos el programa propiamente dicho. Por su parte *saadadb* servirá para contener la base de datos. En cuanto a *repository*, es el directorio en el que deberemos meter los espectros que van a formar parte de nuestra base de datos. Finalmente, *apache* contendrá la instalación del programa Tomcat de Apache, del que hablaremos más adelante.

Una vez hecho esto, escribimos en la terminal:

```
% java -jar Saada1.6.0.noplatform.jar
```

Esto nos ejecutará el instalador de Saada. Aceptamos el acuerdo de licencia y elegimos como carpeta de destino la carpeta *saada* que hemos creado previamente.

El siguiente paso es instalar el Tomcat. Tomcat es un servidor web que usará Saada para que se pueda acceder a la base de datos a través de internet.

Abrimos un terminal y escribimos:

```
% cd myFirstStepWithSaada/apache/apache-tomcat-6.0/bin
% ./startup.sh
```

Construcción de una base de datos de espectros de alta resolución de estrellas frías: clasificación espectral, parámetros estelares, abundancias

Ya hemos instalado y ejecutado Tomcat. Para comprobar que funciona, vamos a nuestro navegador de internet y en donde hay que poner la url escribimos: <http://localhost:8080>. Si lo hemos hecho correctamente, debería aparecer lo siguiente:

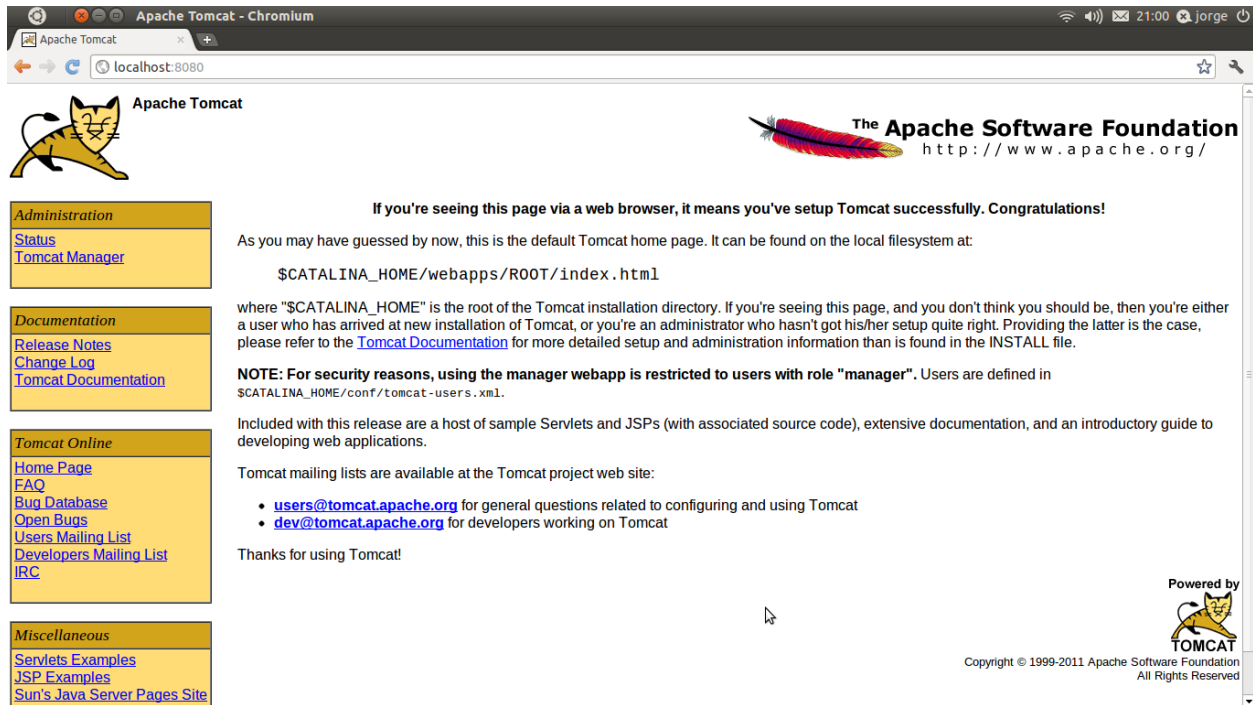


Figura 10

El siguiente paso ya es crear la base de datos propiamente dicha. Para ello tecleamos:

```
% cd Database/saada/Saada1.6.0/bin
```

Esto nos sitúa en la carpeta donde tenemos los archivos necesarios para crear la base de datos. A continuación tecleamos:

```
% ./newsaadadb
```

Al teclear esto se nos abre el interfaz de creación de bases de datos. Primero nos pedirá un nombre para la base de datos, seguido de el directorio donde sea nuestra intención crearla y finalmente el directorio en el que se hallará el repositorio de espectros.

A continuación nos pedirá que elijamos con que sistema queremos que se maneje la base de datos. Saada nos obliga a usar SQLite, por lo que ese es el que usaremos.

Finalmente nos pedirá el directorio dónde hemos instalado el Tomcat.

Con todos estos datos, SaadaDB habrá creado una base de datos y nos proporcionará una URL desde la que podemos acceder a ella.

Hasta aquí he conseguido llegar hasta el momento. Está en proyecto la creación de la base de datos con los espectros procesados.

Capítulo 3: Espectros

Este capítulo final contiene una lista de los 130 espectros que he procesado así como el nombre de la estrella de la que proceden.

<i>Espectro</i>	<i>Estrella</i>
nhd184960_n4_foces05.fits	HD184960
nHD190067_n1_FOCES_jul05.fits	HD190067
nAldebaran1_n3_FOCES_jul05.fits	Aldebaran
nHD190406_n1_FOCES_jul05.fits	HD190496
nAldebaran2_n3_FOCES_jul05.fits	Aldebaran
nHD190470_n1_FOCES_jul05.fits	HD190470
nAldebaran3_n3_FOCES_jul05.fits	Aldebaran
nhd190470_n6_foces05.fits	HD190470
naldebaran_n4_foces05.fits	Aldebaran
nhd191499_n4_foces05.fits	HD191499
naldebaran_n5_foces05_new.fits	Aldebaran
nHD191785_n1_FOCES_jul05.fits	HD191785
naldebaran_n6_foces05.fits	Aldebaran
nHD192263_n2_FOCES_jul05.fits	HD192263
nalphaBoo_n2_FOCES_jul05.fits	Arcturus
nhd195564_n5_foces05_new.fits	HD195564
nalphaBoo_n5_foces05_new.fits	Arcturus
nhd196795_n6_foces05.fits	HD196795
nalphaBoo_n6_foces05.fits	Arcturus
nHD200779_n2_FOCES_jul05.fits	HD200779
ngj625_n4_foces05.fits	GJ625
nhd200968_n5_foces05_new.fits	HD200968
ngsc03469_n5_foces05_new.fits	GSC03469
nHD202575_n2_FOCES_jul05.fits	HD202575
nGSC03469_sin_cuentas.fits	GSC03469
nHD202751_n2_FOCES_jul05.fits	HD202751
nHD10008_n3_FOCES_jul05.fits	HD10008
nhd204587_n6_foces05.fits	HD204587

nHD100696_n1_FOCES_jul05.fits	HD100696
nHD208038_n2_FOCES_jul05.fits	HD208038
nHD100696_n2_FOCES_jul05.fits	HD100696
nHD208313_n1_FOCES_jul05.fits	HD208313
nHD100696_n3_FOCES_jul05.fits	HD100696
nhd210667_n4_foces05.fits	HD210667
nhd100696_n4_foces05.fits	HD100696
nHD214683_n2_FOCES_jul05.fits	HD214683
nhd100696_n5_foces05_new.fits	HD100696
nHD215152_n3_FOCES_jul05.fits	HD215152
nhd100696_n6_foces05.fits	HD100696
nHD215648_n3_FOCES_jul05.fits	HD215648
nhd10476_n4_foces05.fits	HD10476
nHD216259_n3_FOCES_jul05.fits	HD216259
nhd10476_n5_foces05_new.fits	HD10476
nhd217580_n5_foces05_new.fits	HD217580
nHD111395_n2_FOCES_jul05.fits	HD11395
nHD219538_n1_FOCES_jul05.fits	HD219538
nhd115383_n5_foces05_new.fits	HD115383
nhd220182_n4_foces05.fits	HD220182
nhd122120_n4_foces05.fits	HD122120
nHD220221_n1_FOCES_jul05.fits	HD220221
nhd122742_n5_foces05_new.fits	HD122742
nhd220339_n6_foces05.fits	HD220339
nhd124106_n4_foces05.fits	HD124106
nhd222143_n4_foces05.fits	HD222143
nhd124292_n5_foces05_new.fits	HD124292
nHD3651_n3_FOCES_jul05.fits	HD3651
nhd124642_n5_foces05_new.fits	HD124642

nhd3765_n5_foces05_new.fits	HD3765
nHD127244_n2_FOCES_jul05.fits	HD127244
nhd4256_n5_foces05_new.fits	HD4256
nhd128311_n4_foces05.fits	HD128311
nHD4628_n5_FOCES_enero06.fits	HD4628
nHD130004_n2_FOCES_jul05.fits	HD130004
nHD4676_n1_FOCES_jul05.fits	HD4676
nhd131582_n4_foces05.fits	HD131582
nHD4913_n3_FOCES_jul05.fits	HD4913
nHD134083_n2_FOCES_jul05.fits	HD134083
nhd4915_n5_foces05_new.fits	HD4915
nHD136923_n3_FOCES_jul05.fits	HD136923
nhd5996_n6_foces05.fits	HD5996
nHD139323_n1_FOCES_jul05.fits	HD139323
nHD6660_n2_FOCES_jul05.fits	HD6660
nHD141272_n2_FOCES_jul05.fits	HD141272
nHD7439_n3_FOCES_jul05.fits	HD7439
nhd142860_n6_foces05.fits	HD142869
nHD7590_n2_FOCES_jul05.fits	HD7590
nHD145958A_n2_FOCES_jul05.fits	HD145958A
nhd84737_n1_foces02a.fits	HD84737
nHD145958B_n2_FOCES_jul05.fits	HD145958B
nHD8997_n3_FOCES_jul05.fits	HD8997
nhd1461_n4_foces05.fits	HD1461
nHIP103256_n2_FOCES_jul05.fits	HIP103256
nhd147379A_n4_foces05.fits	HD147379A
nHIP115162_n3_FOCES_jul05.fits	HIP115162
nHD148467_n1_FOCES_jul05.fits	HD148467
nhip3418_n6_foces05.fits	HIP3418
nhd148653_n6_foces05.fits	HD148653
nHIP5957_n3_FOCES_jul05.fits	HIP5857
nHD149806_n1_FOCES_jul05.fits	HD149806
nHIP73786_n3_FOCES_jul05.fits	HIP73786
nhd15013_n6_foces05.fits	HD15013
nhip73786_n4_foces05.fits	HIP73786
nHD152391_n3_FOCES_jul05.fits	HD152391
nhip76051_n4_foces05.fits	HIP76051
nhd152391_n5_foces05_new.fits	HD152391
nHIP87579_n1_FOCES_jul05.fits	HIP87579

nHD152555_n3_FOCES_jul05.fits	HD152555
nHIP99764_n2_FOCES_jul05.fits	HIP99764
nhd155712_n6_foces05.fits	HD155712
nnhd100696_n4_foces05.fits	HD100696
nHD1562_n2_FOCES_jul05.fits	HD1562
nREJ1507_n1_FOCES_jul05.fits	REJ1507
nHD159222_n3_FOCES_jul05.fits	HD159222
nrej1507_n4_foces05.fits	REJ1507
nHD160934_n1_FOCES_jul05.fits	HD160934
nrej1507_n5_foces05_new.fits	REJ1507
nHD160934_n3_FOCES_jul05.fits	HD160934
nrej1507_n6_foces05.fits	REJ1507
nhd160934_n4_foces05.fits	HD160934
nXBSJ001749.7_n6_foces05.fits	XBSJ001749.7
nhd160934_n5_foces05_new.fits	HD160934
nHD161198_n1_FOCES_jul05.fits	HD161198
nhd165401_n5_foces05_new.fits	HD165401
nHD16760A_n3_FOCES_jul05.fits	HD16760A
nHD16760B_n3_FOCES_jul05.fits	HD16760B
nhd1690_n4_foces05.fits	HD1690
nHD170493_n1_FOCES_jul05.fits	HD170493
nhd170545_n4_foces05.fits	HD170545
nhd170545_n5_foces05_new.fits	HD170545
nHD173667_n1_FOCES_jul05.fits	HD173667
nHD173818_n1_FOCES_jul05.fits	HD173818
nhd174080_n5_foces05_new.fits	HD174080
nHD178126_n1_FOCES_jul05.fits	HD178126
nhd180161_n4_foces05.fits	HD180161
nhd183870_n4_foces05.fits	HD183870
nHD184385_n1_FOCES_jul05.fits	HD184385
nHD184489_n1_FOCES_jul05.fits	HD184489
n61cygb_n1_foces02b.fits	Cygnus B
nHD182488_n2_FOCES_jul05.fits	HD182488
nHD10476_n2_FOCES_jul05.fits	HD10476

Referencias

- López-Santiago J., Montes D., Crespo-Chacón I., et al. 2006, *ApJ*, 643, 1160
- López-Santiago J., Montes D., Gálvez-Ortiz M.C., et al. 2010, *A&A*, 514, A97
- Maldonado J., Martínez-Arnáiz R.M., Eiroa C., Montes D., et al. 2010, *A&A*, 521, A12
- Martínez-Arnáiz R.M., Maldonado J., Montes D., et al. 2010, *A&A*, 520, A79
- Montes D., López-Santiago J., et al. 2001a, *MNRAS*, 328, 45
- Montes D., López-Santiago J., Fernández-Figueroa M.J., et al. 2001b, *A&A*, 379, 976
- Pfeiffer, M. J., Frank, C., Baumüller, D., Fuhrmann, K., & Gehren, T. 1998, *A&AS*, 130, 381

[-http://iraf.noao.edu/](http://iraf.noao.edu/)

[-http://saada.u-strasbg.fr/saada/spip.php?article32](http://saada.u-strasbg.fr/saada/spip.php?article32)