

Estrazione degli spettri

S. Galletti

L'estrazione degli spettri si fa con la task ***apall*** del package *twodspec*, sottopackage *apextract*. Se lo spettro e' orizzontale dare il comando *dispaxis=1*, se e' verticale *dispaxis=2*. La variabile *dispaxis* rimarra' settata al valore impostato anche se si esce da IRAF.

A questo punto possiamo editare i numerosi parametri di *apall*: nel seguito indichiamo solo quelli che devono essere modificati, lasciando gli altri al loro valore di default. I parametri si modificano con il comando *epar apall*.

- *input*=nome del file con lo spettro
- *output*=nome del file di uscita contenente lo spettro estratto
- Ricordate che per entrambe le variabili si puo' dare invece del nome di un singolo file il nome di una lista che contiene i nomi degli spettri da trattare, facendo precedere la @ alla lista (es. @lista.spettri)
- *interac*=yes
- *find*=yes
- *recente*=yes
- *resize*=yes
- *edit*=yes
- *trace*=yes
- *extras*=yes o no (aggiunge in cubo altre informazioni)
- *lower* = la meta' della larghezza desiderata col segno -
- *upper* = come lower ma col segno +
- *b_sample*= definisce le due striscie ai lati dello spettro su cui calcolare il valore del fondo cielo da sottrarre allo spettro; i valori sono in pixel rispetto al centro dello spettro, come lower ed upper. Piu' le striscie sono larghe migliore e' il rapporto segnale/rumore del cielo, ma bisogna che non ci siano altre stelle nelle striscie

selezionate

- nfind=1
- background=median oppure fit
- pfit=fit1d
- clean=yes
- readnoi= inserire readout noise. Loiano 3.06
- gain=inserire gain. Loiano 2.22

Image Reduction and Analysis Facility

PACKAGE = apextract

TASK = apall

input = **“immaginespettro2D”** List of input images
(output = **“spettroestratto1D”**) List of output spectra
(apertur=) Apertures
(format = multispec) Extracted spectra format
(referen=) List of aperture reference images
(profile=) List of aperture profile images

(interac= yes) Run task interactively?
(find = yes) Find apertures?
(recente= yes) Recenter apertures?
(resize = yes) Resize apertures?
(edit = yes) Edit apertures?
(trace = yes) Trace apertures?
(fittrac= yes) Fit the traced points interactively?
(extract= yes) Extract spectra?
(extras = yes) Extract sky, sigma, etc.?
(review = yes) Review extractions?

(line = INDEF) Dispersion line
(nsum = 10) Number of dispersion lines to sum or median

DEFAULT APERTURE PARAMETERS

(lower = -5.) Lower aperture limit relative to center
(upper = 5.) Upper aperture limit relative to center
(apidtab=) Aperture ID table (optional)

DEFAULT BACKGROUND PARAMETERS

(b_funct= chebyshev) Background function
(b_order= 1) Background function order
(b_sampl= -10:-6,6:10) Background sample regions
(b_naver= -3) Background average or median
(b_niter= 0) Background rejection iterations
(b_low_r= 3.) Background lower rejection sigma
(b_high_= 3.) Background upper rejection sigma
(b_grow = 0.) Background rejection growing radius

APERTURE CENTERING PARAMETERS

(width = 5.) Profile centering width
(radius = 10.) Profile centering radius
(thresho= 0.) Detection threshold for profile centering

AUTOMATIC FINDING AND ORDERING PARAMETERS

nfind = Number of apertures to be found automatically
(minsep = 5.) Minimum separation between spectra
(maxsep = 100000.) Maximum separation between spectra
(order = increasing) Order of apertures

RECENTERING PARAMETERS

(aprecen=) Apertures for recentering calculation

(npeaks = INDEF) Select brightest peaks
(shift = yes) Use average shift instead of recentering?

RESIZING PARAMETERS

(llimit = INDEF) Lower aperture limit relative to center
(ulimit = INDEF) Upper aperture limit relative to center
(ylevel = 0.1) Fraction of peak or intensity for automatic width
(peak = yes) Is ylevel a fraction of the peak?
(bkg = yes) Subtract background in automatic width?
(r_grow = 0.) Grow limits by this factor
(avglimit = no) Average limits over all apertures?

TRACING PARAMETERS

(t_nsum = 10) Number of dispersion lines to sum
(t_step = 10) Tracing step
(t_nlost = 3) Number of consecutive times profile is lost before
(t_funct = legendre) Trace fitting function
(t_order = 2) Trace fitting function order
(t_sample = *) Trace sample regions
(t_naver = 1) Trace average or median
(t_niter = 0) Trace rejection iterations
(t_low_r = 3.) Trace lower rejection sigma
(t_high_r = 3.) Trace upper rejection sigma
(t_grow = 0.) Trace rejection growing radius

EXTRACTION PARAMETERS

(backgro = fit) Background to subtract
(skybox = 1) Box car smoothing length for sky
(weights = none) Extraction weights (none|variance)
(pfit = fit1d) Profile fitting type (fit1d|fit2d)

(clean =	yes) Detect and replace bad pixels?
(saturat=	64000) Saturation level
(readnoi=	3.06) Read out noise sigma (photons)
(gain =	2.22) Photon gain (photons/data number)
(lsigma =	4.) Lower rejection threshold
(usigma =	4.) Upper rejection threshold
(nsubaps=	1) Number of subapertures per aperture

A questo punto lanciare il comando *apall*:

1- guardo dove e' lo spettro

il programma chiede conferma del nome dell'immagine (premere il tasto RETURN)
del numero di aperture (premere RETURN)

chiede se si vogliono modificare le aperture (premere RETURN)

a questo punto si apre la finestra grafica su cui compare il profilo dell'immagine perpendicolare allo spettro, la posizione dello spettro trovata automaticamente con un numero 1 sopra e, sotto, il profilo del fondo cielo le posizioni selezionate per il calcolo delle striscie di cielo da sottrarre.

Se tutto va bene premere **q** (sta per quit), se l'apertura cade su una stella sbagliata cancellarla mettendoci sopra il cursore e dando **d** (delete) e poi mettere il cursore sulla posizione voluta e dare **m** (mark).

2- decido l'apertura

A questo scopo si puo' ingrandire la parte di grafico di interesse col comando **w** (window): mettere il cursore nell'angolo in basso a sinistra del rettangolo che si vuole ingrandire e premere **e**; mettere il cursore nell'angolo in alto a destra e premere di nuovo **e** oppure:

dare **w** poi **z** (zoom) per ingrandire dove e' posizionato il cursore in x e y, oppure **w** poi **x** per ingrandire in x o **w** poi **y** per ingrandire in y.

Per tornare al grafico intero battere **w** poi **a**.

Posso modificare la grandezza dell'apertura utilizzando i comandi *:lower numero*
:upper numero dalla finestra grafica.

3- decido dove prendere il cielo

Dopo che e' stata definita la posizione dell'apertura di estrazione, si passa a controllare che la strazione del cielo sia fatta in modo corretto. Sempre dalla finestra grafica digitare **b** (che sta per background). E' utile riscalare il grafico dando **w** poi **a** e poi andando a ingrandire la parte interessata con **w** poi **z**.

Controllare il fit del cielo sia adeguato in caso positivo digitare **q** e la task ritorna al grafico iniziale, in caso negativo riposizionare il cielo nel modo voluto. Per togliere le regioni selezionate bisogna andarci sopra con il cursore e digitare **t** poi ridare le regioni andando con il cursore dove voglio che prenda le sezioni (una a destra dello spettro e una a sinistra). Mettendomi a sinistra dello spettro digitando **s** poi spostandoni con il cursore di nuovo **s**. Ripetere l'operazione per la parte destra dello spettro.

oppure ridando i limiti con `:b_sample -11:6, 6:12`

ricordarsi alla fine di ridare il comando **f** per rifare il fit.

Finito digitare **q** e la task ritorna al grafico iniziale.

4- Traccio lo spettro

Ridare **q** e il programma chiede se tracciare l'apertura, premere RETURN fino a che compare il grafico dello spettro. In questo grafico, in ascissa c'è la coordinata lungo lo spettro e in ordinata quella ortogonale allo spettro: se lo spettro è ben inseguito la variazione deve essere di pochi pixel: se la linea tratteggiata non segue bene le crocette (tipicamente all'inizio e alla fine dello spettro dove il segnale è peggiore) cancellare le crocette devianti andandoci sopra col cursore e battendo **d**. Premere **f** per ripetere il fit. Si può cambiare l'ordine del fit dando **:o "numero"** (poi ricordarsi di ridare **f**). Quando il fit va bene uscire con **q**.

5- Salvo i parametri usati nel database

Ora chiede se salvare i parametri premere RETURN.

6- riguardo lo spettro monodimensionale

ora chiede se estrarre e rivedere lo spettro, premere sempre RETURN finché compare lo spettro estratto.

Per uscire digitare **q** e viene chiesto se salvare lo spettro, premere RETURN.

Ora lo spettro è monodimensionale.

LAMPADA DI PARAGONE

Estrarre la lampada con gli stessi parametri dello spettro attraverso il comando il linea:

`apall nomelampada out=nomeuscita ref=immagine2Dspettro recen- trace- back- intera-`

Oppure editando la task e inserendo i parametri giusti (dare l'immagine di riferimento, dire di non ricentrare, non ritracciare, non sottrarre il cielo, non fare interattivamente).

PACKAGE = apexextract

TASK = apall

input = "lampada" List of input images
(output = "lampadaestratta1D") List of output spectra
 (apertur=) Apertures
 (format = multispec) Extracted spectra format
(referen= "immaginespettro2D") List of aperture reference images
 (profile=) List of aperture profile images

(interac= **no**) Run task interactively?
 (find = **no**) Find apertures?
 (recente= **no**) Recenter apertures?
 (resize = **no**) Resize apertures?
 (edit = yes) Edit apertures?
 (trace = **no**) Trace apertures?
 (fittrac= yes) Fit the traced points interactively?
 (extract= yes) Extract spectra?
 (extras = yes) Extract sky, sigma, etc.?

(review = yes) Review extractions?

(line = INDEF) Dispersion line
 (nsum = 10) Number of dispersion lines to sum or median

DEFAULT APERTURE PARAMETERS

(lower = -5.) Lower aperture limit relative to center
 (upper = 5.) Upper aperture limit relative to center
 (apidtab=) Aperture ID table (optional)

DEFAULT BACKGROUND PARAMETERS

(b_funct= chebyshev) Background function
 (b_order= 1) Background function order
 (b_sampl= -10:-6,6:10) Background sample regions
 (b_naver= -3) Background average or median

(b_niter=	0) Background rejection iterations
(b_low_r=	3.) Background lower rejection sigma
(b_high_=	3.) Background upper rejection sigma
(b_grow =	0.) Background rejection growing radius

APERTURE CENTERING PARAMETERS

(width =	5.) Profile centering width
(radius =	10.) Profile centering radius
(thresho=	0.) Detection threshold for profile centering

AUTOMATIC FINDING AND ORDERING PARAMETERS

nfind =	1 Number of apertures to be found automatically
(minsep =	5.) Minimum separation between spectra
(maxsep =	1000.) Maximum separation between spectra
(order =	increasing) Order of apertures

RECENTERING PARAMETERS

(aprecen=) Apertures for recentering calculation
(npeaks =	INDEF) Select brightest peaks
(shift =	yes) Use average shift instead of recentering?

RESIZING PARAMETERS

(llimit =	INDEF) Lower aperture limit relative to center
(ulimit =	INDEF) Upper aperture limit relative to center
(ylevel =	0.1) Fraction of peak or intensity for automatic width
(peak =	yes) Is ylevel a fraction of the peak?
(bkg =	yes) Subtract background in automatic width?
(r_grow =	0.) Grow limits by this factor
(avglimi=	no) Average limits over all apertures?

TRACING PARAMETERS

(t_nsum =	10) Number of dispersion lines to sum
(t_step =	10) Tracing step
(t_nlost=	3) Number of consecutive times profile is lost befo
(t_func=	legendre) Trace fitting function
(t_order=	2) Trace fitting function order
(t_sampl=	*) Trace sample regions
(t_naver=	1) Trace average or median
(t_niter=	0) Trace rejection iterations
(t_low_r=	3.) Trace lower rejection sigma
(t_high_=	3.) Trace upper rejection sigma
(t_grow =	0.) Trace rejection growing radius

EXTRACTION PARAMETERS

(backgro=	none) Background to subtract
(skybox =	1) Box car smoothing length for sky
(weights=	none) Extraction weights (none variance)
(pfit =	fit1d) Profile fitting type (fit1d fit2d)
(clean =	yes) Detect and replace bad pixels?
(saturat=	64000.) Saturation level
(readnoi=	3.06) Read out noise sigma (photons)
(gain =	2.22) Photon gain (photons/data number)
(lsigma =	4.) Lower rejection threshold
(usigma =	4.) Upper rejection threshold
(nsubaps=	1) Number of subapertures per aperture

Come al solito, battendo *help nomedeltask* si ottiene la spiegazione completa di tutte le funzioni del task.

Calibrazione in lunghezza d'onda

Il calcolo della trasformazione pixel-lambda si fa con la task *identify* del package *onedspec*. Digitare *oned* per caricare il package.

Vediamo quali sono i parametri da modificare rispetto al default di IRAF:

- images=nome della lampada
- coordli= nome del file hear.tab
- ftype=emission perche' le righe sono in emissione e non in assorbimento
- functio=legendre oppure spline3
- order=1 (si modifica poi interattivamente, ma non con oridine troppo alti)

PACKAGE = onedspec

TASK = identify

images =	"lampadaestratta1D"	Images containing features to be identified
(section=	middle line)	Section to apply to two dimensional images
(databas=	database)	Database in which to record feature data
(coordli=	hear.dat)	User coordinate list
(units =)	Coordinate units
(nsum =	10)	Number of lines/columns/bands to sum in 2D image
(match =	-3.)	Coordinate list matching limit
(maxfeat=	50)	Maximum number of features for automatic identif
(zwidth =	100.)	Zoom graph width in user units
(ftype =	emission)	Feature type
(fwidth =	4.)	Feature width in pixels
(cradius=	5.)	Centering radius in pixels
(thresho=	0.)	Feature threshold for centering
(minsep =	2.)	Minimum pixel separation
(functio=	spline3)	Coordinate function
(order =	1)	Order of coordinate function
(sample =	*)	Coordinate sample regions
(niterat=	0)	Rejection iterations
(low_rej=	3.)	Lower rejection sigma
(high_re=	3.)	Upper rejection sigma
(grow =	0.)	Rejection growing radius
(autowri=	no)	Automatically write to database
(graphic=	stdgraph)	Graphics output device
(cursor =)	Graphics cursor input
crval =		Approximate coordinate (at reference pixel)

cdelt = Approximate dispersion
(aidpars=) Automatic identification algorithm parameters
(mode = ql)

A questo punto lanciare il comando *ident*:
il programma chiede conferma del nome dell'immagine (premere RETURN):
compare lo spettro nella finestra grafica.

1- riconoscere almeno 3-4 righe sparse (non vicine)

Riconoscere le righe (avendo il grafico corrispondente), attivare il cursore sulla finestra (fare click) e selezionare le righe una ad una centrando ogni riga col cursore e battendo **m**. Il programma chiede la lunghezza d'onda corrispondente, scriverla e premere RETURN.

2- fare un fit provvisorio

Quando si sono selezionate tutte le righe volute premere **f**: il programma calcola i coefficienti della relazione e grafica lo spettro con le ascisse in Angstrom. Digitare **j** per vedere il grafico dei residui (pixel/Angstrom).

3- andare avanti ad identificare le righe

Per ritornare alla finestra precedente digitare **q**. Dopo aver identificato almeno 3-4 righe opportunamente sparse e aver fatto il fit posso digitare il comando **l** che mi identifica tutte le righe contenute nel file di riferimento, oppure posso andare avanti a marcarle una a una.

4- rifaccio il fit

Quando si sono selezionate molte righe (almeno una ventina) premere **f**

Nella parte alta della finestra, terza riga, compare l'rms della relazione: questo numero deve essere piccolo altrimenti vuol dire che qualche riga e' stata mal identificata. In questo caso conviene guardare sul grafico quale riga mostra la maggiore deviazione, per capire dove e' stato commesso l'errore, se non si trova conviene uscire dal programma senza salvare e ricominciare da capo. Per cancellare le righe devianti andare con il cursore sopra il punto e digitare **d**. Posso cambiare l'ordine del fit con **:o numero** (fit troppo alti sono fortemente sconsigliati, mai sopra al 3). Cancellate i punti corrispondenti a righe sature.

5- salvo il fit

Se va tutto bene premere **q** e il programma chiede se salvare i parametri: premere RETURN e il programma scrive nella cartella database un file (idnomedelfile) con i parametri salvati. Uscire con **q**.

Per completare la calibrazione bisogna ora usare il task *refspec*, che si trova in **one** che associa la lampada allo spettro. Vediamo i parametri da modificare:

- input=nome del file dello spettro: notare che puo' essere anche una lista di nomi di spettri, in modo da calibrare tutti gli spettri fatti con la stessa configurazione strumentale in un colpo solo
- referen=mettere il nome del file dello spettro usato per identify
- select=nearest
- sort= mettere il campo bianco (barra spaziatrice)
- group= mettere il campo bianco (barra spaziatrice)
- time=no
- overrid=no
- verbose=yes

lanciare il programma e rispondere yes.

Ora occorre lanciare un terzo programma *dispcor*, che assegna effettivamente la relazione lambda-ascissa calcolata allo spettro(i) in esame. Vediamo i parametri da modificare:

- input=lista degli spettri da calibrare
- output=lista degli spettri calibrati: usare nomi diversi da quelli di input in modo da conservare gli spettri originali se qualcosa va male o si vogliono cambiare i parametri
- verbose=yes

Lanciare il programma: con verbose yes il programma dice per ogni spettro le lunghezze d'onda del primo e ultimo pixel, il passo in Angstrom/pixel e il numero totale di pixel. Se la calibrazione e' buona l'rms indicato da identify e' minore del passo, ossia lo scarto tra i centri delle righe effettivi e quelli calcolati dal fit e' minore di un pixel.

Controllare che il passo A/pix siano simili a quello del manuale altrimenti rifare la procedura.

Ora si ha lo spettro calibrato il lunghezza d'onda.

Per identificare lampade con grism gia' fatti si puo' usare il comando: reidentify

> epar reidentify

PACKAGE = onedspec

TASK = reidentify

referenc= "lampadaestratta1D" Reference image
quella gia' identificata con identify
images = "lampadaestratta_nuova1D" Images to be reidentified
(interac= yes) Interactive fitting?
(section= middle line) Section to apply to two dimensional images
(newaps = yes) Reidentify apertures in images not in reference?
(overrid= no) Override previous solutions?
(refit = yes) Refit coordinate function?

(trace = no) Trace reference image?
(step = 10) Step in lines/columns/bands for tracing an image
(nsum = 10) Number of lines/columns/bands to sum
(shift = 0.) Shift to add to reference features (INDEF to sea
(search = 0.) Search radius
(nlost = 0) Maximum number of features which may be lost

(cradius= 5.) Centering radius
(thresho= 0.) Feature threshold for centering
(addfeat= no) Add features from a line list?
(coordli= hear.dat) User coordinate list
(match = -3.) Coordinate list matching limit
(maxfeat= 50) Maximum number of features for automatic identif
(minsep = 2.) Minimum pixel separation

(databas= database) Database
(logfile= logfile) List of log files
(plotfil=) Plot file for residuals
(verbose= yes) Verbose output?
(graphic= stdgraph) Graphics output device
(cursor =) Graphics cursor input

answer = yes Fit dispersion function interactively?
crval = Approximate coordinate (at reference pixel)
cdelt = Approximate dispersion
(aidpars=) Automatic identification algorithm parameters
(mode = ql)

controllare l'RMS se non va bene rifare il fit interattivamente

NB: per utilizzare reidentify bisogna avere il file della lampada gia' identificata e la directory database.

Calcolo dei parametri di righe spettrali

Una riga spettrale ottenuta con bassa risoluzione e' definibile in prima approssimazione da tre parametri: posizione del centro della riga, larghezza a meta' altezza e EW "larghezza equivalente".

Il calcolo di questi parametri e' facilmente ottenibile col task *splot* del package *onedspec*. Questo task puo' fare molte cose, ma noi lo useremo solo per visualizzare gli spettri e calcolare le tre grandezze suddette.

splot si lancia col nome dello spettro da visualizzare. Compare la finestra grafica e si puo' ingrandire il pezzo di spettro desiderato.

Si puo' vedere il valore di un pixel dello spettro mettendo il cursore alla ascissa desiderata e battendo la **barra spaziatrice**: il programma da' la lunghezza d'onda, l'ordinata del cursore e l'intensita' dello spettro alla lunghezza d'onda selezionata.

Scelta una riga, mettere il cursore sul continuo a sinistra e premere **k**, poi metterlo a destra e premere di nuovo **k**: il programma da' il centro della riga, la larghezza equivalente eqw (negativa per righe in emissione) e la larghezza a meta' altezza gfwhm.

Si puo' usare anche **e** per fare la stessa cosa.

Per salvare un grafico bisogna dare dalla finestra grafica

:.snap epsl crea un file sginumero.eps

Altri comandi

:.help per avere help

:mbelow "label" scrive in verticale dove ho il cursore

:mbelow "label" format scrive in orizzontale

sempre dalla finestra grafica si puo' usare T "testo" e scrive "testo" dove c'e' il cursore

Per uscire da *splot* premere q.
