An Introduction to analysis of single dispersion spectra with IRAF

2004년 4월 1일
황 호 성, 이 명 균
galllaxy@astro.snu.ac.kr, mglee@astrog.snu.ac.kr
서울대학교 지구환경과학부 천문학전공

제 1. 절 서론

이 노트는 IRAF(Image Reduction and Analysis Facilities)를 이용하여 분광 관측 data를 처리하고자 하는 사람들의 시작을 위해서 작성되었습니다. 특히 이 노트는 1996년 칠레의 La Silla 에 ESO 1.52m 망원경 B&C+CCD 분광기로 관측한 스펙트럼을 분석하는 것을 설명하고 있다. 자세한 원리와 내용은 original 매뉴얼(이제부터는 TU)을 찾아보고 데이터와 매뉴얼은 아래에서 다운 받으면 된다.

```
http://www.fiz.uni-lj.si/astro/learn.html
```

또한 IRAF 홈페이지 `http://noao.iraf.edu` 에서 많은 것들을 얻을 수 있으니 참고하시기 바랍니다.
여기서 다루지 않는 내용들은 각 manual을 참고하시면 되는데, `IRAF` 기초를 위해서는 `http://astro.snu.ac.kr/~galllaxy`에 있는 IRAF ABC를 참고하시고, `PHOTCAL`을 이용한 표준화 방법은 `http://astro.snu.ac.kr/obscosteam` 에 있는 PHOTCAL TUTORIAL(이명균, 이준협, 김상철, 2001)을, 그리고 `ELLIPSE`를 이용한 은하 표면 측광 방법은 `http://www.kao.re.kr/~sckim/manuals.html`에 있는 ELLIPEE manual을 참고하시기 바랍니다.
수정할 내용이 있으면 언제든지 e-mail을 보내주세요.

제 2. 절 Preprocessing

2.1. 준비 작업

다운 받은 파일을 iraf 작업에 맞게 미리 손봐야 할 필요가 있다. 먼저 파일을 쓰고 지울 수 있도록 모드를 바꿔준다.

```
[astro7] chmod 755 a*
```

iraf에서 읽을 수 있도록 파일명을 .fits로 바꿔준다.

```
[astro7] mv a* a*.fits
```

원래 위 명령은 실행되지 않는다. 따라서 하나씩 mv를 해 주든지 script를 짜서 하든지 알아서 잘 해본다. 이제부터는 iraf에 로그온 해서 실행한다.

2.2. Bits correction

-32768∼+32768 의 original pixel value range를 0∼+65536의 pixel value range로 바꿔주기 위해서 각 pixel마다 32768 값을 더해준다. list_a 와 list_b 등, 뒤에 계속 나올 list 들은 실행 전에 만들어 줘야 한다.

```
cl>imarith @list_a + 32768 @list_b calctype=real pixtype=real
```

2.3. Bias and Dark subtraction

TU 의 파일 목록을 보면 bias 와 dark frame 이 각각 있지만 평균적인 bias 가 176.3 정도라 가정하고 그냥 모든 frame에서 산술적으로 빼 주는 작업만 하고 있다. 또한 dark frame의 경우 질소 냉각 CCD 이기 때문에 빼 주는 작업을 생략했다고 한다. 여기서는 그 값을 그대로 사용하기로 한다.

```
cl>imarith @list_b - 176.3 @list_c calctype=real pixtype=real
```

2.4. Trim the images

bias 빼주기를 해 주었기 때문에 이제 overscan 영역은 필요하지 않다. 따라서 image를 잘라주자. list_c는 TU의 18 page 대로 넣어주면 된다.

```
cl>imcopy @list_d @list_e
```

2.5. Preparing the master flat

우선 모든 flat frame 을 가지고 medflat 을 만든다.

```
cl>imcombine e0001,e0002,e0003,e0004,e0005,e0006 medflat combine=median
```

최종 masterflat 은 pixel 간의 sensitivity 차이만을 갖아야 하므로 wave length dependence 를 없애주기 위해 medflat 을 필요한 것이다.

```
cl>blkavg medflat[20:310,*] avcol_in 291 1
cl>blkrep avcol_in  avcol_out  317  1
cl>imarith medflat / avcol_out masterflat caltype=real pixtype=real
```

2.6. Flat field correction of science frames

list_e 를 수정해서 object file 들만 담도록 한다. 그런 다음 아래와 같은 명령을 실행하면 flat correction 까지 끝났다.

```
cl>imarith @list_e / masterflat @list_g calctype=real pixtype=real
```

물론 TU 에 나온대로 `ccdproc`을 이용해서 전처리를 할 수도 있으나 위 방법이 훨씬 쉬운 것 같다. cosmic ray 제거는 건너뛴다.

제 3. 절 Aperture extraction

이제 apall(noao.twodspec.apextract) task를 이용해서 spectrum 을 꺼내보자. 우선 CCD 양 끝의 값이 별의 spectrum 보다 pixel 값이 높아 별의 스펙트럼을 못찾는 경우가 있다. 따라서 일괄적으로 각 frame 의 양 끝을 잘라내자. 우선 list_g 를 다음과 같은 형태로 만든다.

```
[astro7] vi list_g
  g0001[20:300,*]
  g0002[20:300,*]
  ...

cl> imcopy  @list_g  @list_gtr
```

apall task는 science frame(target)과 calibration frame(CAL 붙은 것) 을 서로 다르게 적용해야 하기 때문에 list를 따로 따로 만들어 준다.(e.g. list_sci(science frame), list_cal(calibration frame))

3.1. Science frame

```
ap> epar apall
PACKAGE = apextract
   TASK = apall
input   =           @list_sci   List of input images
(output =                     ) List of output spectra
(apertur=                     ) Apertures
(format =             onedspec) Extracted spectra format
(lower  =                  -4.) Lower aperture limit relative to center
(upper  =                   4.) Upper aperture limit relative to center
nfind   =                   1   Number of apertures to be found automatically
(llimit =                  -5.) Lower aperture limit relative to center
(ulimit =                   5.) Upper aperture limit relative to center
(backgro=                  fit) Background to subtract
(weights=             variance) Extraction weights (none|variance)
(readnoi=                  4.2) Read out noise sigma (photons)
(gain   =                  1.2) Photon gain (photons/data number)
```

위와 같은 apall task의 parameter 들을 수정해 주는데, help apall 또는 TU manual 등을 반드시 읽어보고 각 parameter의 의미를 이해하도록 노력한다.

실행을 하면 TU 의 Fig. 6 나오는 데 중심의 별이 선택된 지역이 맞는 지 확인하고 b를 눌러 background 범위가 제대로 선택되었는 지 살펴 본 다음 q를 눌러 background mode를 빠져나온다. 이제 다시 q를 누르면 몇 가지 질문들이 차례로 나오는데 계속 Enter 를 누르면 Fig.8 처럼 dispersion 방향을 따라 별의 스펙트럼을 tracing 한 그림이 나온다. + 표시가 크게 벗어난 것들은 그 곳에 cursor를 두고 d를 눌러 fitting에 사용되지 않도록 한다. 이처럼 몇 개의 점을 없애고 f를 누르면 다시 fitting이 되는데 잘 되었다고 생각되면 q를 눌러 다음 단계로 넘어간다. 이제는 Fig. 9와 같이 뽑아진 스펙트럼이 보이게 된다. Fig. 9와 비슷하면 OK 이다.(단 gtr0010.fits 의 경우 peak 가 두 개가 나오는 데 오

른쪽 peak 가 별이니 peak 선택을 옮겨주도록 한다. 그외 커서 조정의 자세한 설명은 TU를 참고하기로 한다.

제 4. 절 Wavelength calibration

4.1. Finding wavelength solution of the first calibration spectrum

이제 calibration frame 에 apall task 를 적용할 차례이다. 주의할 것은 list_cal 에 있는 file들의 순서와 같게 똑같은 target 의 science frame 을 아무거나 한 개씩 매치시켜 list_ref 를 만들어 reference 로 사용한다. output 으로 list_cal_1D도 만들어 준다. 바꿔 주어야 할 parameter 설정은 다음과 같다.

```
ap> epar apall
PACKAGE = apextract
   TASK = apall
input   =           @list_cal   List of input images
(output =       @list_cal_1D) List of output spectra
(referen=          @list_ref) List of aperture reference images
(profile=          @list_ref) List of aperture profile images
(interac=                 no) Run task interactively?
(find   =                 no) Find apertures?
(recente=                 no) Recenter apertures?
(resize =                 no) Resize apertures?
(edit   =                 no) Edit apertures?
(trace  =                 no) Trace apertures?
(fittrac=                 no) Fit the traced points interactively?
(extract=                yes) Extract spectra?
(extras =                 no) Extract sky, sigma, etc.?
(review =                 no) Review extractions?
(backgro=               none) Background to subtract
```

이제 실행하면 calibration frame의 spectrum이 science frame의 spectrum과 같은 위치에서 뽑혀지게 된다. 이제는 calibration frame 의 각 line 들에 실제 파장값을 부여할 차례이다. 여러 calibration frame 중 하나를 선택하여 identify task 를 실행해 보자.

```
ap> epar identify
PACKAGE = onedspec
   TASK = identify
images  = gtr0021_1D.0001.0001  Images containing features to be identified
(order  =                    4) Order of coordinate function
```

위 두 parameter 만 손보면 다른 것은 크게 문제되지 않는다. 이렇게 실행을 하면 TU 의 Fig.11 과 비슷한 그림이 보이는 데 아직 파장 값이 정해진 것이 아니다. 해서 TU 의 Fig.10 과 비교해서 알아차린 line 을 정해서 m을 누르고 그에 해당하는 파장값을 Fig.10 에서 보고 적어주면 된다. 이런 식으로 여러 line 들까지(한 10개 이상) 찾아서 적어주고 f를 누르면 Fig. 12와 같은 fitting 화면이 나온다. fitting line와 점들이 잘 맞지 않는 것 같으면 :order 4(필요하면 더 high order도, 그러나

너무 큰 high order는 좋은 결과를 주지 못한다)를 바꿔주고, fitting에서 너무 크게 벗어난 점들은 d를 눌러 지워주고 다시 f를 눌러 만족할 때 q를 눌러 fitting mode를 빠져 나간다. 다시 q를 눌러 빠져 나가면 identify가 완료되었다. 이제 다른 calibration frame 들도 identify 를 해야하는데 reidentify task를 사용해서 쉽게 할 수 있다.

```
on> reidentify gtr0036_1D.0001.0001 referenc=gtr0021_1D.0001.0001
```

4.2. Apply wavelength calibration to science data

이제 각 science frame 에 calibration frame 의 결과를 적용하자. 각 target 별로 하나의 calibration frame 에 대해 여러 science frame 을 적용한다. 다시 science frame 으로만 이루어진 list_w 를 만든다.

```
cl> hedit gtr0020.0001 refspec1 gtr0021_1D.0001.0001 add+ ver-
cl> dispcor @list_sci @list_w linearize=no
```

제 5. 절 Flux calibration

Flux calibration 을 하기 위해서 준비할 것이 필요한 데 TU 에서는 두 가지 방법을 소개하고 있다. 그런데 여기서는 먼저 나온 방법만 사용하겠다. 우선 TU 의 Table 7 의 내용을 열심히 쳐서(오타 없이 잘 쳐야겠죠!!) eso.set 이라고 저장한다. 그런 다음 w*fits 를 a.lst 로 저장한다. 이제 eso.set script 를 실행한다.

```
cl> cl < eso.set
```

위의 cl 두 개 써 있다고 오타로 생각하면 안 된다. 앞의 cl 은 iraf prompt 라는 의미의 cl 이고 뒤의 cl 은 iraf script 실행할 때 쓰는 cl 이다. 그러면 막 실행이 되면서 여러가지 결과가 잘 나온다. 혹시 에러메시지 비슷한 게 뜰 수도 있으니 읽어보고 잘 처리해 준다.

5.1. The task standard

이제 표준화를 할 차례인데 여러 science frame 중 STD 라고 서 있는 frame 이 표준성 frame 이다. 해서 이 중 하나를 선택해서 standard task 를 실행한다. 고쳐주어야 할 parameter 는 다음과 같다.

```
on> epar standard
PACKAGE = onedspec
   TASK = standard
input   =                w0014  Input image file root name
output  =                  std  Output flux file (used by SENSFUNC)
(extinct=onedstds/ctioextinct.dat) Extinction file
(caldir =     onedstds/ctionewcal) Directory containing calibration data
star_nam=             feige 67     Star name in calibration list
```

결과적으로 std 라는 파일이 하나 생긴다.

5.2. The task sensfunc

이제는 instrumental sensitivity and atmosphere transparency 의 wavelength dependence 를 찾아 보정해주어야 한다. 수정해주어야 할 parameter 는 다음과 같다.

```
on> epar sensfunc
PACKAGE = onedspec
   TASK = sensfunc
standard=                   std  Input standard star data file (from STANDARD)
(extinct=onedstds/ctioextinct.dat) Extinction file
(newexti=          extinct.dat) Output revised extinction file
(graphs =                 srei) Graphs per frame
```

실행하면 TU 의 Fig.16 이 나오는 데 srei 라는 옵션을 주었기 때문에 panel 이 네 개가 나올 것이다. 아무튼 enter 를 열심히 눌러준다.

5.3. The task calibrate

이제 거의 막바지이다. calibration 을 하자. `calibrate task parameter` 를 다음과 같이 실행 한다.

```
on> epar calibrate
PACKAGE = onedspec
   TASK = calibrate
input   =             @list_w   Input spectra to calibrate
output  =             @list_f   Output calibrated spectra
(extinct=          extinct.dat) Extinction file
```

제 6. 절 Additional corrections

6.1. Heliocentric correction

지구의 움직임에 따른 doppler 효과를 제거해 주자. `rvcorrect (noao.astutil)`를 다음과 같이 실행 한다.

```
cl>rvcorrect  ima=@list_f imupdate+ observa=eso
cl>dopcor  @list_f list_h -vhelio isvel+
```

이제 모든 게 끝났다. 각 target 별로 spectrum 을 얻었을 텐데 상훈이형의 Fig.7 - Fig.11 과 비교해 보면 잘 되었는 지 알 수 있다.

제 7. 절 Manuals

아래와 같은 곳에서 Manual들을 얻을 수 있다.

- 한글 :
  - `http://astro.snu.ac.kr/obscosteam/`

    * PHOTCAL - 표준화
    * 그 외
  - http://www.kao.re.kr/~sckim/manuals.html
    * ELLIPSE - 은하 표면 측광
    * ASTROMETRY - 천체의 RA, DEC 구하기
    * FXCOR - 스펙트럼 시선 속도 구하기
    * 그외 다수
  - http://astro.snu.ac.kr/~galllaxy/
    * Preprocessing - image 전처리
    * DAOPHOT-PSF 측광
    * Spectroscopy - 분광 관측 data reduction
    * 그외 조금
- English:
  - http://iraf.noao.edu/iraf/web/docs/docmain.html
    * Everything about iraf

참고문헌

Zwitter T. and Munari U.(1996), *An introduction to analysis of single dispersion spectra with IRAF*
오상훈(2002), *CCD 분광 자료처리(IRAF)*