



Globalny Teleskop Astrofizyczny

Instrukcja korzystania z oprogramowania
do redukcji danych spektroskopowych

Monika K. Kamińska

Instytut Obserwatorium Astronomiczne
Uniwersytet im. Adama Mickiewicza w Poznaniu

Wersja dokumentu: 0.4.0-pl
Data utworzenia: 2017-09-27
Ostatnia modyfikacja: 2017-12-30 (MKK)

this page is intentionally left blank

Wstęp

Niniejsza instrukcja wyjaśnia krok po kroku kolejne etapy redukcji widm typu echelle za pomocą dedykowanego oprogramowania, dostosowanego do danych dostarczanych przez teleskopy projektu GTA (GATS).

Podczas redagowania instrukcji przyjęto założenie, że Użytkownik wykonujący redukcję jest zaznajomiony z pracą w systemie Linux z poziomu linii komend, pakietem IRAF, formatem danych FITS oraz programem ds9 (lub innym, pozwalającym wyświetlić graficzną zawartość pliku FITS) i z grubsza wie, w jaki sposób przebiega redukcja widm typu echelle.

Założono również, że Użytkownik uzyskał już dostęp do serwera redukcyjnego, na którym zainstalowane jest niezbędne oprogramowanie. Niniejszy dokument nie zawiera instrukcji logowania (w tym loginów i haseł) na serwer redukcyjny.

1. Lokalizacja katalogów na serwerze redukcyjnym

DANE

Archiwum obserwacji spektroskopowych z teleskopu **PST1**, z podziałem na poszczególne noce obserwacyjne, znajduje się w katalogu

/dane/spectro/PST1_Raw/

Archiwum obserwacji spektroskopowych z teleskopu **PST2**, z podziałem na poszczególne noce obserwacyjne, znajduje się w katalogu

/dane/spectro/PST2_Raw/

Dane z każdej nocy obserwacyjnej z PST2 dzielone są dodatkowo na podkatalogi odpowiadające poszczególnym kamerom. Dane spektroskopowe znajdują się w podkatalogu **ikon**.

OPROGRAMOWANIE DO REDUKCJI

Aktualne oprogramowanie redukcyjne dla danych z teleskopu **PST1** znajduje się w katalogu

/dane/spectro/Skrypty/IRAF_PST1_[A.B] /

Aktualne oprogramowanie redukcyjne dla danych z teleskopu **PST2** znajduje się w katalogu

/dane/spectro/Skrypty/IRAF_PST2_[C.D] /

(przy czym w miejscu **[A.B]** oraz **[C.D]** znajduje się numer aktualnej wersji oprogramowania).

KATALOGI REDUKCYJNE

Aby uniknąć bałaganu, sugerowanym miejscem do redukcji danych na polaris jest katalog

/dane/spectro/automatic_red/

Struktura podkatalogów jest tu następująca:

- ➔ nazwa obiektu, którego dane będziemy redukować,
 - ➔ data obserwacji w formacie "**YYYY-MM-DD_PST1**" dla zestawów widm z PST1
 - ➔ data obserwacji w formacie "**YYYY-MM-DD**" dla zestawów widm z PST2
 - ➔ pliki z widmami wyłącznie danego obiektu z danej nocy

Przykładowo: **/dane/spectro/automatic_red/epsPer/2016-09-10_PST1/*.fits**

ARCHIWUM REDUKCJI

Archiwum, do którego kopiuje się *wyłącznie poprawnie unormowane widma* (tzn. tylko wynikowe pliki *.fits po redukcji), znajduje się w katalogu

`/dane/spectro/ARCH_RED_GATS/`

W Archiwum redukcji ***nie umieszczamy***:

- plików pośrednich, powstających w trakcie redukcji,
- list plików *.fits,
- jakichkolwiek plików nie związanych z redukcją widm,
- wyników korelacji,
- wykresów,
- publikacji,
- itp.

Struktura katalogów w Archiwum redukcji jest następująca:

- ➔ nazwa obiektu
 - ➔ oznaczenie teleskopu, z którego pochodzą dane ("**PST1**", "**PST2**", ...)
 - ➔ katalog zawierający widma z danej nocy, którego nazwa zawiera:
 - datę w formacie "**YYYY-MM-DD**"
 - oznaczenie trybu redukcji:
 - **IRAF** - dla starych skryptów IRAF-owych
 - **SRP** - dla oprogramowania KK w starszej wersji (obecnie nie używane)
 - **AUTO** - dla aktualnego oprogramowania redukcyjnego
 - ID (inicjały) osoby, która redukowała widma

Przykładowo: `/dane/spectro/ARCH_RED_GATS/20Sco/PST2/2014-02-28_AUTO_MF/*.fits`

2. Przygotowanie do redukcji

Na początku tworzymy katalog roboczy o odpowiedniej ścieżce dla wybranego obiektu i daty obserwacji (data w nazwie katalogu roboczego powinna odpowiadać dacie w nazwie katalogu, z którego pochodzą nasze widma do redukcji). Jeśli widma pochodzą z PST1, dodajemy dodatkowo oznaczenie teleskopu ("**PST1**"), np.:

```
/dane/spectro/automatic_red/epsPer/2016-09-22/  
/dane/spectro/automatic_red/kappaCas/2017-07-05_PST1/
```

Do utworzonego katalogu roboczego kopiujemy z odpowiedniego katalogu z obserwacjami:

- wszystkie widma obiektu (tylko jeden obiekt na dany katalog roboczy!)
- ramki kalibracyjne (***bias*.fits**, ***flat*.fits** oraz widma ***thar*.fits** do flatów i widm naszego obiektu)

oraz pliki ze skryptami i programami w odpowiedniej wersji (patrz: rozdział 1).

UWAGA: w przypadku widm z PST1 upewniamy się, czy w pliku **red01radec.py** nasz obiekt jest na liście (zwracamy tu uwagę na duże i małe litery, podkreślenia w nazwie, itp. - nazwa musi być zgodna z nazwami plików *.fits obiektu w katalogu roboczym) i ma wpisane właściwe współrzędne R.A., Dec. (na epokę J2000.0).

Następnie:

- ◆ upewniamy się, czy wszystkie pliki *.fits poprawnie się skopiowały i nie ma pustych plików
- ◆ jeśli danej nocy wykonane zostały serie ramek kalibracyjnych przed i po obserwacjach, sprawdzamy, czy obie są kompletne (dotyczy to głównie widm z PST2) - jeśli tak, można w czasie redukcji użyć obydwu, można też wybrać tylko jeden zestaw (w tym drugim przypadku zbędne pliki usuwamy z katalogu)
- ◆ upewniamy się, że widma Th-Ar są poprawnie naświetlone
- ◆ przeglądamy (np. przy pomocy ds9) widma obiektu, aby ocenić ich jakość i usunąć ewentualne nieudane/niedoświetlone pliki przed rozpoczęciem redukcji.

Jeśli widma Th-Ar są lekko niedoświetlone, modyfikujemy plik **red06.c1**, ustawiając w nim w dwóch miejscach w funkcjach **ecreidentify** parametr **thresho=60**. Im niższa wartość tego parametru, tym słabsze linie w widmie Th-Ar będą użyte w dopasowaniu - ale nie należy ustawiać go zbyt nisko, aby uniknąć dopasowywania szumu. Przy dobrze naświetlonych widmach Th-Ar można spokojnie ustawić **thresho=500** i dopasowanie powinno być ok.

3. Automatyczna część redukcji

Automatyczna część redukcji odbywa się w pakiecie IRAF. Aby go uruchomić, wpisujemy w katalogu roboczym

```
xgterm &
```

i potwierdzamy klawiszem **ENTER**, a następnie w nowo otwartym terminalu piszemy

```
c1
```

i ponownie klikamy **ENTER**. Następnie po kolei wywołujemy kolejne komendy (od red01 do red07), opisane poniżej.

red01

```
Skrypt red01 wykorzystuje następujące funkcje w IRAF-ie:
    noao -> imred -> astutil -> asthedit
    noao -> imred -> astutil -> setjd
    images -> imutil
oraz następujące dodatkowe skrypty/programy:
    red01.py
    red01changefiles.py
    red01 (red01.c)
    red01.dat
    red01radec.py (w wersji skryptów dla PST1)
```

Na tym etapie:

- nagłówki plików *.fits uzupełniane są o brakujące keywords, które są IRAFowi niezbędne do prawidłowego działania,
- tworzone są listy plików z odpowiednimi nazwami i rozszerzeniami dla poszczególnych etapów redukcji.

UWAGA: Jeśli po przejściu etapu **red01** uruchomiliśmy skrypt z choćby jednego z dalszych etapów, **red01** już nie powtarzamy.

red02

```
Skrypt red02 wykorzystuje następujące funkcje w IRAF-ie:
    noao -> imred -> ccdred -> setinstrument
    noao -> imred -> ccdred -> zerocombine
    noao -> imred -> ccdred -> ccdproc
oraz następujące dodatkowe skrypty/programy/pliki:
    list_bias_slow.dat
    list_bias_fast.dat
```

Na tym etapie następuje uśrednienie biasów i odjęcie uśrednionego poziomego biasu (**BIAS.fits**) od wszystkich pozostałych ramek *.fits w zestawie.

red03

```
Skrypt red03 wykorzystuje następujące funkcje w IRAF-ie:  
noao -> imred -> ccdred -> flatcombine  
images -> imutil -> imcopy  
noao -> imred -> ccdred -> ccdlist  
noao -> imred -> crutil -> crmedian  
oraz następujące dodatkowe skrypty/programy/pliki:  
red03 (red03.c)  
list_flat3.dat  
list_flat120.dat  
list_objects.dat  
list_objects_cosmic.dat
```

Obecnie na obydwu teleskopach projektu GATS wykonujemy dla każdej nocy flaty "krótkie" (nastawione na odpowiednie naświetlenie czerwonej części widma) oraz "długie" (pozwalające doświetlić słabszą, niebieską część widma). Przyjęliśmy takie podejście, ponieważ zdarzało się, że w przypadku doskonale naświetlonych widm obiektu w momencie dzielenia ich przez flat w niebieskiej części widma wprowadzany był dodatkowy, zbędny szum, ostatecznie pogarszający jakość widma gwiazdy.

Etap **red03** obejmuje uśrednianie widm lampy flatowej osobno dla każdego wariantu. Powstaje uśredniony flat dla części czerwonej (**FLAT_A.fits**) oraz niebieskiej (**FLAT_B.fits**). Następnie odbywa się usuwanie kosmików z widm obiektów.

red04

```
Skrypt red04 wykorzystuje następujące funkcje w IRAF-ie:  
noao -> imred -> echelle -> apall  
noao -> imred -> ccdred -> ccdlist  
oraz następujące dodatkowe skrypty/programy/pliki:  
list_reference.dat  
list_objects_cosmic.dat  
list_thar.dat
```

Mając uśrednione pliki FLAT, przystępujemy do przekonwertowania ich zawartości z obrazu 2D do postaci 1D (na tym etapie cały czas mamy do czynienia z widmami "pociętymi" na poszczególne rzędy). Program sprawdza rozmieszczenie i przebieg poszczególnych apertur widma echelle, a następnie odczytuje sygnał z pikseli wzdłuż każdego rzędu widma. Końcowym produktem tego etapu jest rozkład natężeń (w ADU) względem kolejnych pikseli na ramce (liczonych wzdłuż apertury widma).

Jeśli skrypt **red04** się zawiesi albo nie przerobi prawidłowo wszystkich plików obiektu, thar i flat z *.fits -> *.ec.fits, zwykle pomaga uruchomienie go ponownie (warto zapisać sobie, przy którym pliku *.fits wystąpił błąd). Jeśli ponowne uruchomienie również kończy się informacją o błędzie, sprawdzamy, czy problematyczny plik wygląda prawidłowo (jeśli nie, usuwamy go z zestawu) i rozpoczynamy całą redukcję od początku.

Jeśli w nowym przebiegu nadal pojawia się problem z tym samym plikiem *.fits, rozpoczynamy całą redukcję raz jeszcze, uprzednio usuwając problematyczny plik z zestawu - nawet jeśli wydaje się ok (w takiej sytuacji będę wdzięczna za informację, że taki błąd się pojawił i którego pliku dotyczył - przyp. MKK).

red05

```
Skrypt red05 wykorzystuje następujące funkcje w IRAF-ie:  
noao -> onedspec -> sarith  
images -> imutil -> imstatistics  
images -> imutil -> imdelete  
noao -> imred -> ccdred -> ccdlist  
oraz następujące dodatkowe skrypty/programy/pliki:  
list_objects_ec.dat  
list_objects_c.dat
```

Na tym etapie łączymy czerwoną część flatów "krótkich" (FLAT_A) z niebieską częścią flatów "długich" (FLAT_B) - wciąż w postaci osobnych rzędów widma - po czym używamy tej "składanki" do wstępnego unormowania poszczególnych rzędów widma obiektu.

red06

```
Skrypt red06 wykorzystuje następujące funkcje w IRAF-ie:  
noao -> imred -> echelle -> ecreidentify  
oraz następujące dodatkowe skrypty/programy/pliki:  
list_thar_ec.dat  
ecred_output.txt  
reference.ec  
ecreference.ec
```

Przystępujemy do re-identyfikacji linii widm lampy Th-Ar. Odbywa się to w oparciu o pliki **ecreference*.ec**, znajdujące się w katalogu **database** w zestawie skryptów redukcyjnych.

Na tym etapie szczególnie ważne jest zwrócenie uwagi na to, *ile linii w poszczególnych widmach Th-Ar IRAF zdołał zidentyfikować*. Informacja ta jest wyświetlana na końcu działania skryptu **red06**.

Jeśli zidentyfikowanych linii jest mniej niż 800-900, można spróbować zmienić wartość parametru **thresho** w funkcji **ecreidentify** wewnątrz skryptu **red06.c1** na nieco niższą (robi się to w dwóch miejscach w skrypcie, patrz rozdział 2) i spróbować ponownie (w tym celu trzeba wyjść z IRAF-a komendą **"logout"** i uruchomić go na nowo, aby wczytał poprawioną wersję skryptu). Jeżeli nie nastąpiła poprawa, najprawdopodobniej widma Th-Ar są zbyt słabo naświetlone i skala długości fali w poszczególnych rzędach widma nie będzie dostatecznie dobrze określona - w tej sytuacji odpuszczamy redukcję danego zestawu widm (proszę wówczas również o informację, że taka sytuacja miała miejsce i której nocy obserwacyjnej ona dotyczy - słabo naświetlone widma kalibracyjne mogą oznaczać konieczność wymiany lampy Th-Ar na nową :) - przyp. MKK).

red07

```
Skrypt red07 wykorzystuje następujące funkcje w IRAF-ie:  
  noao -> onedspec -> refspectra  
  noao -> imred -> echelle -> dispcor  
  noao -> imred -> ccdred -> ccdlist  
oraz następujące dodatkowe skrypty/programy/pliki:  
  red07.py  
  list_objects_c.dat  
  list_objects_l.dat  
  list_thar_ec.dat  
  list_thar_ec1.dat  
  list_thar_l.dat
```

Po re-identyfikacji linii lampy Th-Ar w oparciu o widma odniesienia, uzyskana skala długości fali przenoszona jest na widma obiektów. Ten etap w razie konieczności może zostać powtórzony wielokrotnie.

4. Normowanie widm

Po zakończeniu etapu **red07** przystępujemy do normowania widma. Odbywa się to osobno dla każdego rzędu widma.

Aby przejść do programu **red08**, w którym następuje dopasowanie funkcji do normowania, wychodzimy z IRAF-a i zamykamy xgterm odpowiednio komendami:

```
logout
exit
```

Jeśli dla widm danego obiektu z innej nocy (z tego samego teleskopu!) wykonywane było już wcześniej wypłaszczanie kontinuum, można przekopiować do aktualnego katalogu roboczego pliki konfiguracyjne z wcześniejszego normowania:

```
envelope_[nazwa_obiektu].ini
lines_[nazwa_obiektu].ini
rows_[nazwa_obiektu].ini
```

Jeśli pliki konfiguracyjne nie zostały skopiowane do katalogu roboczego, zostaną tam utworzone automatycznie (z domyślną zawartością, zwykle nie dopasowaną do konkretnego obiektu) przy pierwszym uruchomieniu programu.

red08

Program **red08** uruchamiamy (we właściwym katalogu roboczym) komendą:

```
./red08 nazwa_obiektu
```

Podana w komendzie **nazwa_obiektu** pozwala programowi zidentyfikować nazwy wymienionych wyżej plików konfiguracyjnych (***.ini**), z których ma korzystać podczas normowania widm w danej sesji.

Plik **envelope_[nazwa_obiektu].ini** (**envelope*.ini**) zawiera ogólne ustawienia programu, w większości edytowalne już w samym programie i zazwyczaj nie ma konieczności do niego zaglądać.

O wiele ważniejszy jest plik **lines_[nazwa_obiektu].ini** (**lines*.ini**), który zawiera listę zakresów widma (w Å), które mają być *wyłączone* z dopasowania. Zaznaczane są w ten sposób głównie linie absorpcyjne (zarówno pojedyncze, jak i całe zakresy - np. "grzebień" linii tellurycznych) i emisyjne (zakładamy, że redukowane widma nie zawierają linii emisyjnych o szerokościach porównywalnych i/lub większych niż długość pojedynczej apertury redukowanego widma).

W trzecim pliku konfiguracyjnym, **rows_[nazwa_obiektu].ini** (**rows*.ini**), określone są zakresy długości fali λ (w Å) dla każdego rzędu widma z osobna. W zależności od różnych czynników zdarza się czasami potrzeba zmiany punktu łączenia niektórych rzędów widm - w takim przypadku modyfikujemy zakresy w tym miejscu.

Kolejne 2 kolumny zawierają domyślne parametry dopasowania, których generalnie się nie zmienia. W skryptach dostosowanych do widm z teleskopu PST2 istnieje możliwość dodania w pliku **rows*.ini** jeszcze dwóch dodatkowych kolumn, modyfikujących parametry zakrzywienia brzegowych fragmentów każdej z apertur widma z osobna.

UWAGA: Zmiany w pliku **rows*.ini** wykonujemy *tylko na pierwszym widmie w zestawie*. Wprowadzanie w tym pliku zmian w trakcie normowania kolejnych widm, zwłaszcza zmiana zakresów długości fali dla poszczególnych rzędów widma, spowoduje na dalszych etapach problem z właściwym połączeniem rzędów dla części widm. Jeśli wystąpi konieczność edycji pliku **rows*.ini** dla któregoś z dalszych widm, najlepiej postąpić zgodnie z procedurą opisaną w rozdziale 6.

Program **red08** składa się z 4 etapów (stage 1-4). Na etapach 1-3 program dokonuje wyboru punktów reprezentatywnych dla kontinuum w widmie i wykonuje próbne dopasowanie w oparciu o parametry wskazane przez użytkownika. Na etapie 4 odbywa się końcowe dopasowanie, którego wynikiem powinno być poprawnie unormowane widmo.

OBSŁUGA INTERFEJSU red08 - okno tekstowe

klawisze funkcyjne

F1	- otwarcie graficznego interfejsu, podgląd aktualnego etapu normowania
F2	- przejście do kolejnego etapu normowania
F3	- powrót do poprzedniego etapu normowania
F8	- uruchomienie automatycznego normowania całego zestawu widm zgodnie z aktualnymi wartościami parametrów (UWAGA: tą opcję uruchamiamy <u>na etapie 1</u> normowania dowolnego widma)
F10	- wyjście z programu red08

ustawienia wyświetlania dla okna graficznego

window size	- rozmiar okna (zwykle liczba między 12 a 16)
window proportion	- proporcja rozmiaru okna w osi y względem osi x
original spectrum	- czy ma być domyślnie wyświetlone widmo wejściowe dla danego etapu (0-nie, 1-tak), można to zmienić później interaktywnie
corrected spectrum	- czy ma być domyślnie wyświetlone widmo po poprawkach dla danego etapu (0-nie, 1-tak), można to zmienić później interaktywnie
corrected median pts	- czy mają być domyślnie wyświetlone uśrednione punkty widma, używane na danym etapie (0-nie, 1-tak), można to później zmienić interaktywnie
autoscale y	- czy ma być domyślnie włączone autoskalowanie w osi y (0-nie, 1-tak), można to później zmienić interaktywnie
starting xmin	- początkowa długość fali do wyświetlenia, można to później zmienić interaktywnie
ending xmax	- końcowa długość fali do wyświetlenia (w programie wyświetla się w tym miejscu dwa razy "starting xmin", to oczywiście drobiazg do poprawki) można to później zmienić interaktywnie

etap 1 (STAGE 1)

Median points nr	- liczba punktów (pikseli) w widmie, które mają być użyte do wyliczenia jednego punktu widma uśrednionego
Bend x left	- piksel, od którego rozpoczyna się wykrzywienie niebieskiej końcówki pojedynczego rzędu widma (jedna wartość dla wszystkich rzędów widma)
Bend x right	- piksel, od którego rozpoczyna się wykrzywienie czerwonej końcówki pojedynczego rzędu widma (jedna wartość dla wszystkich rzędów widma)
Bend coeff left	- współczynnik zakrzywienia rzędów widma na końcach niebieskich (jedna wartość dla wszystkich rzędów widma), ujemny to zakrzywienie w dół, dodatni w górę
Bend coeff right	- współczynnik zakrzywienia rzędów widma na końcach czerwonych (jedna wartość dla wszystkich rzędów widma), ujemny to zakrzywienie w dół, dodatni w górę
Starting row nr	- numer pierwszej apertury widma echelle na ramce, do której dopasowane są kontinua pozostałych rzędów

etap 2 (STAGE 2)

Median points nr	- liczba punktów (pikseli) w widmie, które mają być użyte do wyliczenia jednego punktu widma uśrednionego
Nr of pts	- liczba punktów uśrednionych, do których ma być dopasowany jeden segment łamanej linii reprezentującej wstępne dopasowanie do kontinuum
Factor	- współczynnik decydujący o tym, czy segmenty dopasowanej linii łamanej będą umieszczane centralnie względem punktów widma (0.5), czy odchylone w dół (<0.5) bliżej dolnej obwiedni, czy odchylone w górę (>0.5) bliżej górnej obwiedni
Low reject	- krotność sigmy dopasowania kontinuum, po przekroczeniu której punkty znajdujące się PONIŻEJ linii dopasowania są odrzucane
High reject	- krotność sigmy dopasowania kontinuum, po przekroczeniu której punkty znajdujące się POWYŻEJ linii dopasowania są odrzucane
Minimum reject	- minimalne odchylenie punktu od linii dopasowania, niezbędne, by punkt można było odrzucić
Lines edges file name	- nazwa odpowiedniego pliku lines*.ini

etap 3 (STAGE 3)

Nr of pts	- tak jak na etapie 2
Factor	- tak jak na etapie 2

etap 4 (STAGE 4)

Order	- rząd dopasowania funkcji spline (jedna dla całego widma), która ma skorygować ewentualne błędy powstałe po wstępnej normalizacji na etapie 3
Exclude top	- wartość punktu, powyżej której jest on pomijany przy dopasowaniu funkcji spline; zakładamy, że wstępna normalizacja sprowadziła kontinuum w okolice 1.0, więc zwykle 1.1 wystarcza
Exclude bottom	- wartość punktu, poniżej której jest on pomijany przy dopasowaniu funkcji spline; zakładamy, że wstępna normalizacja sprowadziła kontinuum w okolice 1.0, więc zwykle 0.98 wystarcza
Exclude lines	- czy pomijać linie widmowe (zakresy widma) z pliku lines*.ini
Lines edges file name	- nazwa odpowiedniego pliku lines*.ini
Exclude cosmic	- czy usuwać punkty odstające w górę ponad wartość 1.0 o podaną wielokrotność sigmy
Exclude low-high	- usunąć punkty, które zazwyczaj można uznać za ewidentnie złe (poniżej -1.0 i powyżej 2.0)

OBSŁUGA INTERFEJSU red08 - okno graficzne

nawigacja w oknie graficznym

w, s, a, d	- przesuwanie widma: w górę, w dół, w lewo, w prawo
+, -, =	- zoom in/out w obydwu osiach
l, j	- zoom in/out w osi x
k, i	- zoom in/out w osi y
r	- wyświetla całe widmo
1, 2, 3, 4, 5	- włączenie/wyłączenie wyświetlanych danych (różna w zależności od aktualnego etapu)
q	- zamknięcie okna graficznego i powrót do okna tekstowego

red08 - zalecana procedura postępowania:

- (1) Na wstępie przechodzimy do etapu 4 (stage 4) i oglądamy wynik działania programu przy ustawionych domyślnych parametrach dla pierwszego (dobrze naświetlonego) widma w zestawie. Klawiszem "4" włączamy/wyłączamy autoskalowanie w pionie. Korzystając z klawiszy **w, s, a, d, i, j, k, l** (patrz: ramka "nawigacja w oknie graficznym") przeglądamy stopniowo całe widmo w poszukiwaniu miejsc, gdzie automatyczne normowanie zawiodło.
- (2) Natrafiwszy na miejsce wymagające poprawek w normowaniu, wracamy do etapu 1 (stage 1). Klawiszem "1" włączamy/wyłączamy podgląd "uciętych" końcówek każdego z rzędów widma (tj. fragmentów, które znalazły się poza ustalonymi w pliku **rows*.ini** zakresami długości fali dla każdego z rzędów widma).
- (3) Korygujemy normowanie kontinuum w problematycznym miejscu. W tym celu edytujemy plik **lines*.ini** lub **rows*.ini** (w zależności od potrzeb, patrz niżej) i sprawdzamy wynik, przechodząc na etap 4 (stage 4).
- (4) Przechodzimy stopniowo przez całe widmo, usuwając po kolei błędy dopasowania. Rezultat wprowadzanych zmian sprawdzamy wyświetlając widmo na etapie 4 (stage 4).
- (5) Mając poprawnie unormowane pierwsze widmo w zestawie, można uruchomić automatyczne normowanie pozostałych przy użyciu tych samych parametrów.

Na co szczególnie należy zwrócić uwagę w punkcie (2):

- a) łączenie sąsiednich rzędów widma przypada wewnątrz linii widmowej

Ponieważ wykonujemy obserwacje obiektów różniących się temperaturą (a co za tym idzie liczbą linii w widmach) i o różnych (w dodatku zwykle zmiennych) prędkościach radialnych, zdarza się, że łączenie kolejnych rzędów widma wypadnie wewnątrz linii widmowej. W takiej sytuacji należy wyedytować plik **rows*.ini**, przesuując miejsce łączenia rzędów tak, aby przypadało na możliwie płaski fragment kontinuum widma.

b) jeden z rzędów widma w miejscu łączenia jest wyraźnie mniej zaszumiony, niż drugi

W takiej sytuacji należy wyedytować plik **rows*.ini**, przesuwając miejsce łączenia rzędów tak, aby przypadało na punkt, w którym obydwa sąsiadujące rzędy widma są podobnie zaszumione.

c) "garb" w kontinuum nie przypada na miejsce łączenia rzędów, ale w pobliżu znajdują się linie absorpcyjne

W pliku **lines*.ini** najprawdopodobniej nie zostały uwzględnione znajdujące się w danym obszarze widma linie, które mogą zaburzać dopasowanie, "wypychając" kontinuum w górę. W tej sytuacji należy wyedytować plik **lines*.ini**, korygując istniejące (lub dodając nowe) zakresy długości fali, które program powinien wykluczyć z dopasowania.

d) końcówki jednego lub obydwu sąsiadujących rzędów widma są wygięte w taki sposób, że prawidłowe dopasowanie obydwu rzędów jest niemożliwe

W przypadku widm z PST2 należy wyedytować plik **rows*.ini**, dodając (lub modyfikując) w nim kolumny 6 i 7 - współczynniki wygięcia końcówek dla odpowiednich rzędów widma. Z reguły współczynniki te mieszczą się w przedziale od -0.5 do 0.5 (choć czasami sięgają nawet bezwzględnych wartości rzędu 2.5). Dodatnia wartość oznacza wygięcie końcówki bardziej w górę, ujemna - w dół.

W przypadku widm z PST1 nie ma aktualnie możliwości ustawienia tych parametrów osobno dla każdego rzędu widma. Zamiast tego jest możliwość ustawienia na etapie 1 (stage 1) parametru wygięcia osobno dla niebieskiego (**Bend coeff left**) i dla czerwonego (**Bend coeff right**) końca rzędu widma - przy czym będą to jednakowe wartości stosowane dla wszystkich rzędów. W praktyce dla widm z PST1 jest to zwykle wystarczające.

5. Końcowe etapy redukcji

WAŻNE: Po zakończeniu etapu normowania kontinuum programem **red08** należy w pierwszej kolejności upewnić się, że zakresy długości fali dla poszczególnych rzędów widma, wskazane w plikach **red09** i **red10**, są *dokładnie takie same*, jak w pliku **rows*.ini**. Jeśli występują różnice, edytujemy pliki **red09** i **red10**, dostosowując ich zawartość do zakresów wykorzystywanych na etapie **red08** (nie odwrotnie!!!).

Upewniwszy się, że zakresy długości fali są zgodne, uruchamiamy z powrotem terminal:

xgterm &

i IRAF-a:

cl

red09

```
Skrypt red09 wykorzystuje następujące funkcje w IRAF-ie:
noao -> onedspec -> scopy
images -> imutil -> imdelete
noao -> onedspec -> scombine
oraz następujące dodatkowe skrypty/programy/pliki:
list_objects_b1.dat
list_objects_one.dat
```

Dopiero na tym etapie redukcji oddzielne rzędy każdego z widm obserwowanego obiektu są łączone w jedną całość. Ten skrypt może być uruchamiany wielokrotnie.

red10

```
Skrypt red10 wykorzystuje następujące funkcje w IRAF-ie:
noao -> onedspec -> scopy
images -> imutil -> imdelete
noao -> onedspec -> scombine
oraz następujące dodatkowe skrypty/programy/pliki:
list_thar_l.dat
list_thar_one.dat
```

Etap **red10** łączy w całość widma lampy kalibracyjnej (Th-Ar). Ten skrypt może być uruchamiany wielokrotnie.

red11

```
Skrypt red11 wykorzystuje następujące funkcje w IRAF-ie:
noao -> onedspec -> splot
noao -> onedspec -> bplot
oraz następujące dodatkowe skrypty/programy/pliki:
list_objects_one.dat
```

Skrypt **red11** umożliwia przejrzanie każdego z widm obiektu (w postaci 6 fragmentów) w zakresie od 3900 do 6800 Å. Wykorzystywana do tego funkcja IRAF-a **splot** pozwala w dowolnym momencie obejrzeć całe widmo (dwukrotne kliknięcie "**a**"), lub jego wybrany fragment (wskazujemy myszą i klikamy "**a**" kolejno na lewym i prawym skraju wybranego fragmentu widma, aby go powiększyć).

Na tym etapie można usunąć pominięte przez automat "kosmiki" (j.w. - klikamy "**x**", wskazując kursorem kolejno lewy i prawy brzeg usuwanego fragmentu). Po wprowadzeniu zmian klikamy "**i**", aby automatycznie zapisać zmodyfikowany plik z widmem. Do kolejnego fragmentu widma przechodzimy, wybierając "**q**".

W razie potrzeby skrypt **red11** może (a nawet powinien :)) być uruchamiany wielokrotnie, aby użytkownik mógł upewnić się, że nic nie zostało pominięte.

red12

```
Skrypt red12 wykorzystuje następujące dodatkowe skrypty/programy/pliki:  
red12.py  
list_objects_one.dat
```

Na zakończenie uruchamiamy ostatni skrypt, który zmienia nazwy plików z widmami obiektów (ujednolicając jednocześnie ich format) tak, aby można było wszystkie dane dla wybranego obiektu umieścić we wspólnym katalogu i razem analizować. Jest to szczególnie ważne w przypadku danych z PST1, ponieważ numeracja plików z tego teleskopu każdej nocy rozpoczyna się od "01".

6. Przypadki szczególne :)

Bywa, że redukcja nie idzie tak sprawnie, jak byśmy chcieli. Zdarzają się błędy, problemy, obiekty o nietypowych widmach, zmiany konfiguracji sprzętowej. Poniżej zamieszczam najczęściej spotykane przykłady tego typu przypadków. Niektóre problemy da się dość prosto obejść, inne wymagają sporo pracy, a niektóre są nie do przeskoczenia. Zachęcam do przejrzenia tej listy, może okazać się pomocna. Będę ją uzupełniać w miarę pojawiania się kolejnych ciekawych przypadków.

I. KTÓRYŚ SKRYPT NIE DZIAŁA LUB ZGŁASZA BŁĄD

(1) red01 na końcu raportuje błąd

Jeśli mamy do czynienia z danymi z PST1, należy sprawdzić, czy obiekt został dodany do listy w pliku **red01radec.py**. Jeśli nie, należy uzupełnić plik. Jeśli tak, upewniamy się, że znajdująca się w pliku **red01radec.py** nazwa zgadza się idealnie z oznaczeniem obiektu w nazwach plików *.fits z widmami (zwracamy uwagę na podkreślenia, duże i małe litery, itp.).

(2) red04 przerywa pracę i raportuje błąd

Sprawdzamy, którego pliku *.fits dotyczy problem. Zaglądamy do niego i sprawdzamy, czy wygląda prawidłowo. Jeśli tak, uruchamiamy skrypt red04 raz jeszcze.

Jeśli po ponownym uruchomieniu red04 znowu pojawi się błąd, usuwamy wszystko z katalogu i rozpoczynamy całą redukcję od nowa - możliwe, że na którymś z wcześniejszych etapów coś poszło nie tak.

Jeśli błąd się powtarza, albo plik *.fits rzeczywiście okaże się wadliwy (np. pusty), można uruchomić całą redukcję raz jeszcze, usuwając uprzednio problematyczny plik z katalogu roboczego.

(3) red08 uruchamia się, ale nie chce działać

Możliwe, że plik **envelope*.ini** w katalogu roboczym jest uszkodzony. Usuń go z katalogu roboczego i uruchom program raz jeszcze - plik **envelope*.ini** zostanie utworzony na nowo z domyślnymi parametrami.

Jeśli **red08** nadal nie działa, być może pracujesz na zbyt starych danych z PST1, do których oprogramowanie redukcyjne nie zostało dostosowane. Ma to związek ze zmianą konfiguracji sprzętowej oraz późniejszymi zmianami w procedurze samych obserwacji, które są powiązane ze zmianami w procesie redukcji. Można czekać na poprawki w kodzie **red08**, albo zredukować dane starszymi skryptami, w których normowanie wykonywane jest ręcznie za pomocą IRAF-a (w obydwu przypadkach proszę o informację i służę pomocą w redukcji starszą metodą - przyp. MKK).

II. PROBLEM Z RAMKAMI KALIBRACYJNYMI

- (1) w zestawie danych brakuje plików `*bias*.fits` lub `*flat*.fits`

Można skopiować odpowiedni zestaw plików `*bias*.fits` lub `*flat*.fits` z którejś z sąsiednich nocy obserwacyjnych - zwykle działają bez problemu.

Należy się jednak upewnić, że widma gwiazdowe z danej nocy są ok. Jeśli były naświetlane wyłącznie jako test pracy teleskopu (a nie w ramach normalnej nocy obserwacyjnej), mogą się okazać nieużyteczne.

- (2) w zestawie brakuje plików `*thar*.fits` lub jest tylko pojedynczy plik na całą noc

Najprawdopodobniej były to tylko testy lub pokaz w ramach praktyk studenckich. Danych z tej nocy nie redukujemy do celów naukowych - użycie widm Th-Ar z innej nocy miałyby się z celem.

Wyjątkiem od tej reguły jest dosłownie parę pojedynczych nocy, podczas których testowana była komórka jodowa, jednak aktualne oprogramowanie redukcyjne nie jest dostosowane do redukcji danych z komórką jodową.

- (3) widma Th-Ar są, ale bardzo słabo naświetlone

Nie ma sensu redukować tych danych - bez dobrej skali długości fali widma i tak nie będą użyteczne, a użycie widm Th-Ar z innej nocy miałyby się z celem.

III. PROBLEM Z DANYMI

- (1) na etapie **red08** pierwsze widmo udało się unormować, ale z kolejnymi widmami przy tych samych parametrach automat sobie nie radzi - konieczne są dalsze zmiany w pliku **rows*.ini**

Dotyczy to głównie widm z PST2 z sezonu 2016/2017, przy czym najbardziej jaskrawym przypadkiem jest tu gwiazdka eps Per. W takiej sytuacji redukcja całego zestawu potrwa nieco dłużej, ale da się zrobić:

- a) normujemy pierwsze widmo w zestawie tak dobrze, jak się da i zapuszczamy **red08** dla pozostałych na tych samych parametrach,
- b) przechodzimy kolejno etapy **red09** i **red10** (pamiętając o uprzednim uzgodnieniu zakresów długości fali z plikiem **rows*.ini**),
- c) na etapie **red11** przeglądamy po kolei wszystkie widma i sprawdzamy, które zostały unormowane prawidłowo, a które nie,
- d) przechodzimy etap **red12**

e) edytujemy pliki:

```
list_objects_b1.dat  
list_objects_1.dat  
list_objects_one.dat
```

usuwając z nich nazwy plików odpowiadające poprawnie zredukowanym widmom w zestawie - *pozostawiamy na liście jedynie te widma, które wymagają poprawki.*

f) iterujemy etapy od (a) do (e) tak długo, aż wszystkie widma w zestawie zostaną unormowane poprawnie.