

UNIVERSIDADE DO VALE DO PARAÍBA  
INSTITUTO DE PESQUISA E DESENVOLVIMENTO

SEMINÁRIO DE REDUÇÃO DE DADOS  
**FOTOMETRIA**

\*\*\*\*\*

*Baseado no IRAF TUTORIAL SECTIONS disponível em:  
<http://iraf.noao.edu/tutorials/tutorials.html>*

\*\*\*\*\*

Dr. Will Robson M. Rocha  
Dr. Sergio Pilling

29 de outubro de 2015  
São José dos Campos, SP

## 1. INTRODUÇÃO

**IRAF (Image Reduction and Analysis Facility)** é uma ampla coleção de [softwares](#) para uso geral em redução e análise de dados científicos. O projeto IRAF foi desenvolvido e é mantido pelo [National Optical Astronomy Observatory \(NOAO\)](#), em [Tucson, Arizona](#), nos [Estados Unidos](#). Dentro do IRAF, os softwares estão organizados em "pacotes". O pacote "NOAO", por exemplo, inclui uma coleção de programas para uso geral em processamento de imagens e em aplicações gráficas, bem como um grande número de programas para redução e análise de dados de astronomia óptica.

Pacotes externos também estão disponíveis para uso em análise de dados de fontes específicas como os dados do [Hubble Space Telescope \(HST\)](#), dados XRAY ou EUV, por exemplo.

O IRAF oferece um ambiente completo de programação, o qual inclui uma linguagem de programação em script, chamada *Command Language*, uma interface de programação em IMFORT Fortran, e um ambiente de programação SPP/VOS no qual o sistema IRAF (portável) foi escrito.

O IRAF é gratuito e está disponível para a maioria dos [sistemas operacionais](#). Em ambiente [Windows](#), o IRAF pode ser executado através do [Cygwin](#).

Fonte do texto: <https://pt.wikipedia.org/wiki/IRAF>

## 2. INSTALAÇÃO DO IRAF

Atualmente existem várias formas fáceis de instalar o IRAF. Neste trabalho mostraremos como montar um arquivo .iso e instalar o IRAF de forma fácil e eficiente nos sistemas operacionais linux (Ubuntu).

Inicialmente será necessário salvar no diretório /home/seu\_nome o arquivo IRAF\_Ubuntu.iso que pode ser baixado no link: [http://www.astro.uson.mx/%7Efavilac/downloads/ubuntu-iraf/iso/IRAF\\_Ubuntu.iso](http://www.astro.uson.mx/%7Efavilac/downloads/ubuntu-iraf/iso/IRAF_Ubuntu.iso). Em seguida é só copiar e colar os comandos no terminal:

1. Verificar se você já tem o arquivo IRAF\_Ubuntu.iso em seu computador.

2. `sudo mkdir /media/iso`

```
sudo mount -o loop IRAF_Ubuntu.iso /media/iso
```

```
cd /media/iso
```

3. `sudo sh install.sh`

4. Digitar no diretório /home/seu\_nome para criar o arquivo login.cl:

```
mkiraf ---> Escolher xgterm
```

5. Fechar e reabrir terminal

6. Digitar no terminal:

```
sudo gedit /usr/local/bin/irafshell
```

Inserir no editor de texto gedit:

```
PID=`pidof ds9`  
if [ ! $PID ]; then  
    ds9 &  
fi  
pushd ~/iraf > /dev/null  
xgterm -iconic -geometry 80x24 -sb -title "IRAF" -bg "lemon chiffon" -fg "black" -e "ecl" &  
popd > /dev/null
```

7. Dar permissão para todos os usuários:

```
sudo chmod a+x /usr/local/bin/irafshell
```

8. Fechar e reabrir terminal

```
-> Digitar ecl no local onde está o login.cl  
-> Digitar !xgterm  
-> Digitar irafshell
```

Pronto!!!

### 3 PRIMEIROS COMANDOS NO IRAF

Na presente sequência de instalação do IRAF, duas janelas são abertas ao digitar o comando “irafshell”, que é a janela xgterm com onde serão executados os comandos do IRAF e a janela com o visualizador ds9. A partir de agora vamos ver como executar alguns comandos com o IRAF.

O IRAF trabalha com linha de comando e, portanto, não deve ser problema para usuários do Linux. Para iniciar esse tutorial, verifique se você já tem a pasta *exercices* com as seguintes pastas dentro: *dataio*, *docs*, *intro*, *phot*, *spec*. Caso tenha, verifique inicialmente em qual diretório você está com o comando `pwd`. É comum que se esteja no diretório

```
/home/seu-nome/IRAF
```

uma vez que você foi até lá para abrir o IRAF corretamente devida a presença do arquivo `login.cl`. Se esse for o caso, saia dele com o comando

```
ecl > cd ..
```

e vá até o diretório `exercices`, digitando

```
ecl > cd exercises
```

se essa pasta estiver em `/home/seu-nome/`.

Agora que estamos em `/home/seu-nome/exercises`, entre em `intro` com o comando:

```
ecl > cd intro
```

Para verificar o que tem na pasta, digite:

```
ecl > ls
```

Existem dois arquivos .fits chamados im010.fits e im011.fits que contêm imagens do mesmo campo, porém levemente deslocadas uma da outra. Podemos ver mais informações sobre as imagens digitando, por exemplo:

```
ecl > imhead im010.fits l+
```

Para abri-las no ds9 digitamos o seguinte comando seguido do número 1:

```
ecl > displa im010.fits número (1, 2, 3, ...)
```

Dessa forma, o ds9 abre a imagem alocando o frame 1. Outras imagens podem ser abertas em outros frames, digitando o mesmo comando acima seguido dos números 2, 3, ..., dependendo de quantas imagens você quer abrir no ds9.

Para verificar que as imagens estão deslocadas clicamos em blink na janela do ds9. O processo de correção do campo deslocado é feito usando o comando

```
ecl > imexamine
```

Ao digitar esse comando, um círculo piscante aparece na tela do ds9 indicando um modo interativo de visualização. Algumas informações de cada estrela do campo podem ser obtidas colocando o cursor sobre a estrela de interesse e clicando na tecla “a” do teclado do computador. As informações aparecem na janela do IRAF, como coordenada, fluxo, magnitude e etc. Para sair do modo interativo clicamos em “q” no teclado do computador.

Para corrigir o deslocamento do campo, escolhemos 3 ou mais estrelas brilhantes em cada imagem e anotamos o seu deslocamento. Em seguida fazemos a diferença do deslocamento e por fim a média das coordenadas. No meu caso, obtive como média -0.54 e -1.66. Essa correção é feita no ambiente IRAF digitando:

```
ecl > epar imshift
```

onde deve ser digitado o nome da imagem deslocada e o nome de saída, bem como inseridas as correções desejadas. Para executar as correções digita-se “:go” + ENTER. Para verificar se a correção foi aplicada abrem-se as imagens novamente no ds9 (Ex. Im10.fits e s011.fits – corrigida) e clica-se em blink.

Além dessas correções, é possível fazer cálculos com as imagens através do comando:

```
ecl > imarith im010 / s011 aver2 pixt=r e calct=r
```

ou através do epar imarith e em seguida digitar os comandos.

Finalmente, para sair do IRAF digita-se:

```
ecl > logout
```

#### **4. PROCESSOS DE PRÉ-REDUÇÃO DE DADOS**

A pré-redução de dados consiste em retirar ruídos instrumentais dos dados científicos para melhor análise. As etapas são fazer a correção de overscan e o trim das imagens, bem como a subtração do bias e a divisão pelo flatfield.

## 4.1 Correção de overscan e trim

A região de overscan fica localizada na borda da imagem obtida com o CCD, e aparece devido à sua própria eletrônica quando ele é feito o processo de leitura. Assim, o nível de píxeis na região do overscan dá uma média do sinal introduzido pela leitura do CCD, o que também é chamado de nível de overscan. Dessa forma, após ser computado, ele deve ser removido. A remoção é feita fazendo um corte na imagem e preservando apenas a região de interesse. Esse corte é chamado de TRIM. Acontece que para fazer essa subtração, todas as colunas da região de overscan são agrupadas em uma só coluna, e a ela é ajustada uma função suave (nesse caso Chebyshev). Assim, a função ajustada é subtraída de todas as colunas da imagem.

## 4.2 Subtração do Bias

Após o primeiro passo, precisamos agora subtrair o bias. Ele está associado ao ruído de leitura do CCD. Após as cargas serem armazenadas no poço de potencial, depois da exposição do telescópio, os elétrons são transportados para um amplificador e para um conversor analógico digital, que transforma cada carga em um pacote de dados que vai ser lido pelo computador. Essa leitura é traduzido como um determinado número de contagens. Assim temos um ganho determinado pela razão de elétrons por ADU (Analogical Digital Unity). A questão é que esse ganho não é o mesmo para todos os CCDs, e portanto, isso está associado ao ruído de leitura. Então, são feitas exposições curtas (cerca de 1s) com o obturador da câmera fechado para que seja computado apenas o ruído. Na prática, são feitas em torno de 10 imagens bias, e então é obtida uma imagem com o ruído médio. Esse processo faz com que tenhamos uma relação sinal ruído melhor para a análise de imagens. Dessa forma, as imagens dos campos são subtraídas do ruído médio proveniente do CCD.

## 4.3 Divisão pelo flat field

O último passo então, é remover as variações de ganho pixel à pixel. A necessidade dessa remoção ocorre porque os píxeis do CCD não respondem de maneira uniforme à iluminação incidente. Assim são feitas algumas exposições em uma tela branca uniformemente iluminada nos filtros que está sendo feita a observação. Então as imagens são combinadas da mesma forma como foram as imagens *bias*, mas nos filtros correspondentes. Daí, a imagem em cada filtro é então normalizada. Finalmente, as imagens dos campos são divididas pelas imagens de cada flat correspondente. Portanto, temos uma imagem final subtraída de todos os fatores inseridos pelo CCD.

## 4.4 Pré-redução no IRAF

Para iniciar o processo de pré-redução no IRAF, saia do diretório **intro** e vamos entrar no diretório phot. Nesta pasta encontram-se imagens de bias, flat em dois filtros e imagens de campo em dois filtros. Para saber o que cada uma significa vamos digitar:

```
ecl > imhead m92*.fits
```

O próximo passo não é obrigatório, mas apenas para facilitar a identificação dos dados vamos renomeá-los com os seguintes comandos:

```
ecl > imrename m92001.fits bias.fits  
ecl > imrename m92006.fits flatB.fits  
ecl > imrename m92007.fits flatV.fits
```

```
ecl > imrename m92010.fits M92V_1.fits
ecl > imrename m92011.fits M92V_2.fits
ecl > imrename m92014.fits M92B_1.fits
ecl > imrename m92015.fits M92B_2.fits
```

Em seguida vamos visualizar os arquivos com o comando:

```
ecl > displa bias.fits 1 (exemplo)
```

Agora devemos encontrar a região de overscan e definir as regiões de trim. Para isso digitamos o comando:

```
ecl > implot flatV.fits
```

Será aberta uma nova janela interativa, cujos principais comandos são:

e → expande o plot  
l → mostra as linhas da imagem para uma dada coluna  
c → mostra as colunas da imagem para uma dada linha  
barra de espaço → mostra a posição da linha ou da coluna

Nessa janela escolhemos que a região de trim deve ser entre 1 – 318 para coluna e 2 – 510 para linha. Para a imagem bias.fits escolhemos 335 – 350 para coluna e 2 – 510 para linha. Uma observação antes de continuar é que algumas vezes é preciso combinar as imagens bias e flat porque numa noite de observação várias imagens são feitas. Isso é feito com as tarefas:

**noao** → **imred** → **ccdred** → **zerocombine** (para bias)  
**noao** → **imred** → **ccdred** → **flatcombine** (para flat)

No entanto, nesse tutorial não foi preciso seguir esses passos, pois eles já tinham sido feitos.

Então, após ter os flats médios em cada filtro e os bias médios, as tarefas que nos permitem fazer as correções são: **noao** → **imred** → **ccdred** → **epar ccdproc**. Então introduzimos a imagem que queremos fazer a subtração do bias e a divisão pelo flat, a região que será feito o trim, a imagem bias e a imagem flat no filtro correspondente. Se quisermos que seja feito primeiro o trim, depois o bias e depois o flat separadamente, precisamos digitar *yes* ou *no* em *trim*, *zerocor* e *flatcor*. Finalmente digitamos **:go** e as imagens estão pré-reduzidas.

É importante observar neste processo que o overscan e o trim são aplicadas a todas as imagens, inclusive ao flat field e ao bias. Para isso é necessário mudar na lista “ccdtype” para imagens → object, flat field → flat e bias → zero. Em seguida, o bias é aplicado a todas as imagens, inclusive ao flat field. Posteriormente, os flat fields são aplicados às imagens dos seus respectivos filtros.

## 5. FOTOMETRIA NO IRAF

O estudo de fotometria tem como objetivo calcular as magnitudes das estrelas do campo. Dessa forma, a redução de fotometria é feita em duas partes: (i) cálculo das magnitudes instrumentais e (ii) calibração para o sistema fotométrico padrão.

Durante o procedimento de fotometria iremos usar bastante as informações no *header* das imagens, como por exemplo o tempo de exposição, a identificação do filtro e o *airmass*. Para sabermos o *airmass*, pode-se seguir o seguinte caminho:

```
ecl > astutil
ecl > phelp setairmass → para sabe informações sobre essa tarefa
ecl > unlearn setairmass → limpar possíveis informações contidas nessa tarefa
ecl > setairmass image.fits update- → será perguntado o nome do observatório
```

A informação da massa de ar (*airmass*) precisa ser adicionada ao *header* da imagem através do comando:

```
ecl > setairmass nome_da_imagem.fits
```

Verifique se o comando foi executado acessando novamente o *header* da imagem.

Para iniciar a fotometria por abertura digite os comandos:

```
ecl > digiphot
ecl > apphot
```

Agora é necessário escolher o tamanho do raio de abertura para fazer a fotometria. Esse raio deve ser escolhido como um múltiplo da FWHM da PSF (*Point Spread Function*) das estrelas. Para medir o FWHM das estrelas usamos o *imexamine*. Digite os seguintes comandos:

```
ecl > displa nome_da_imagem 1
ecl > imexamine
```

Ao fazer isso, um círculo piscante vai aparecer na tela do ds9 onde a imagem foi carregada. Agora pode-se clicar em “r” no teclado do computador para saber a FWHM das PSFs. O valor de FWHM é adotado como o último valor na lista abaixo do gráfico. O valor que geralmente é adotado para o cálculo da abertura é de 4 a 5 vezes a FWHM. Outras teclas também fornecem imagens da estrela como por exemplo:

```
s → plot de superfície
e → plot de contorno
c → plot da coluna da imagem
l → plot da linha da imagem
```

Após fazer esse procedimento para uma imagem, o mesmo pode ser feito para as demais imagens:

```
d → pergunta ao usuário o nome da nova imagem a ser carregada
r → faz o plot da PSF
```

Para as imagens da pasta *exercises* o valor de FWHM para o campo de M92 no filtro V foi de 3 pixels, e portanto a abertura pode ser escolhida como 15 pixels.

Agora usaremos a tarefa *qphot* no modo interativo para fazer a fotometria do primeiro campo (M92 em V). Digite os comandos:

```
ecl > unlearn apphot
ecl > epar qphot → em seguida ctrl + d para salvar os parâmetros
```

As informações que podemos inserir para os campos são:

```

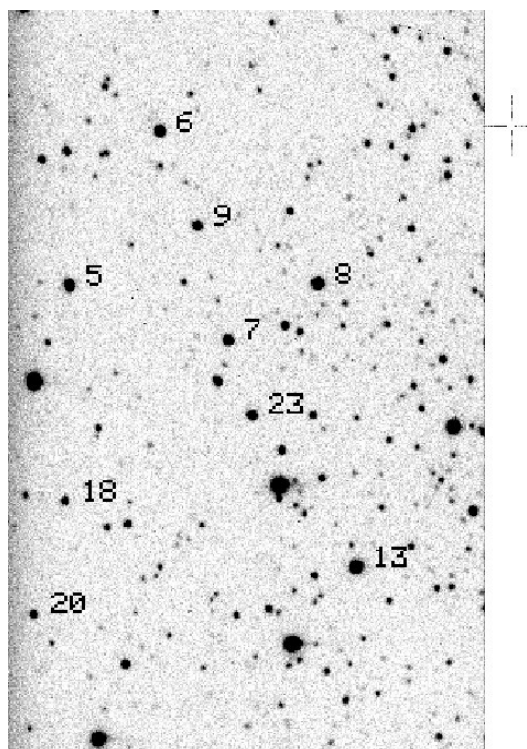
    image = "m92010"      Input image
    cbox = 5.             Centering box width in pixels
    annulus =             Inner radius of sky annulus in pixels
    dannulus =           Width of the sky annulus in pixels
    apertures = "10"     List of photometry apertures
    (coords = "")        Coordinate list
    (output = "default") Results file
    (plotfile = "")      Plot metacode file
    (zmag = 25.)         Zero point of magnitude scale
    (exposure = "exptime") Exposure time image header keyword
    (airmass = "airmass") Airmass image header keyword
    (filter = "filters") Filter image header keyword
    (obstime = "ut")     Time of observation image header keyword
    (epadu = 14.)       Instrument gain
(interactive = yes)     Interactive mode
    (radplots = no)      Plot the radial profiles in interactive mode
    (verbose = no)       Print messages
    (graphics = "stdgraph") Graphics device
    (display = "stdimage") Display device
    (icommands = "")     Image cursor: [x y wcs] key [cmd]
    (gcommands = "")     Graphics cursor: [x y wcs] key [cmd]
    (mode = "ql")

```

Uma vez que os parâmetros foram salvos, digita-se:

```
ecl > qphot nome_da_imagem
```

Podemos escolher cbox=5, annulus=15, dannulus=10 e aperture=10. Ao fazer isso, o círculo piscante está na imagem do ds9. Para fazer a fotometria, posicionamos o cursor em cima da estrela e tecamos na “barra de espaço”. Podemos observar que algumas informações vão aparecer na tela do IRAF, como o centro (x, y), céu, magnitude e erro (se houver). Vamos executar esse comando seguindo a sequência de identificação: 5, 6, 7, 8, 9, 13, 18, 20, 23 conforme a Figura abaixo. Ao final tecele q para sair do modo interativo.





Além de mostrar os parâmetros na tela do IRAF, um arquivo \*.mag.1 foi criado no diretório onde estão sendo feitos os procedimentos. Agora, podemos fazer a fotometria para os outros campos no modo não interativo usando uma lista de coordenadas. Para criá-la pode-se digitar:

```
ecl > txdump nome_da_imagem.mag.1 xcenter,ycenter yes > coords
ecl > type coords
ecl > displa nome_da_imagem 1
ecl > tvmark 1 coords mark=circle radii=10 color=205
```

Agora fazemos:

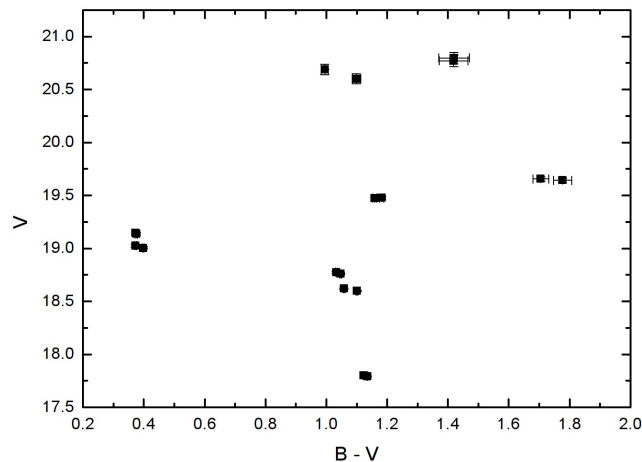
```
ecl > epar qphot
```

e colocamos em image = image1, image2, image3,..., em coords = coords e finalmente :go. Isso deve gerar um arquivo .mag.1 para cada imagem. Para ver o resultado da fotometria pode-se digitar:

```
ecl > txdump *.mag.1 xcenter,ycenter,mag,merr,ifilter yes
```

Note que os filtros são V=60 e B=50.

Os resultados das magnitudes instrumentais podem ser usadas para plotar um diagrama Cor-Magnitude, como o mostrado a seguir:



## 6. CALIBRAÇÕES FOTOMÉTRICAS

Agora podemos transformar as magnitudes instrumentais para o sistema padrão usando as tarefas do pacote **photcal**.

```
ecl > unlearn photcal
ecl > photcal
```

A princípio é necessário criar um arquivo que contém as magnitudes das estrelas padrão, que pode ser feito com a tarefa **mkcatalog**. Entretanto, para o tutorial que estamos seguindo, esse arquivo já existe (stds). Um arquivo associado chamado de fstds.dat existe para descrever o formato de stds.

Agora precisamos criar um arquivo que contém uma lista de imagens medidas com o **qphot**, e agrupadas em um conjunto de dados. Para fazer isso abra um editor de texto (ex. Gedit) chamado *imsets* e digite por exemplo:

```
M92 : m92010 m92014
M92 : m92011 m92015
```

O próximo passo é criar um arquivo que conterá as observações das estrelas padrão. Esta tarefa extrairá as magnitudes instrumentais dos arquivos *.mag.1* e agrupará de acordo com o arquivo *imsets*. Lembre-se que os filtros são identificados nesse tutorial como  $V = 60$  e  $B = 50$ . Para criar o arquivo digite:

```
ecl > epar mknobsfile
```

e altere as linhas necessárias de acordo com a seguinte figura:

```
photfiles = "*.mag*"           The input list of APPHOT/DAOPHOT databases
idfilters = "60,50"           The list of filter ids
imsets = "imsets"            The input image set file
observations = "sobs"         The output observations file
(obsparams = "")              The input observing parameters file
(obscolumns = "2,3,4")        The format of obsparams
(minmagerr = 0.001)           The minimum error magnitude
(shifts = "")                  The input x and y coordinate shifts file
(apercors = "")                The input aperture corrections file
(aperture = 1)                 The aperture number of the extracted magnitude
(tolerance = 8.)               The tolerance in pixels for position matching
(allfilters = yes)             Output only objects matched in all filters
(verify = no)                  Verify interactive user input ?
(verbose = yes)                 Print status, warning and error messages ?
(mode = "ql")
```

Nota: Observe no seu diretório que foi criado um arquivo *sobs*.

O passo seguinte é criar um arquivo de configuração para aplicar as equações de correção de magnitudes ao sistema fotométrico. Para fazer isso usa-se a rotina **mkconfig**. Nesse tutorial esse arquivo já está feito e chama-se *m92fig*.

Agora vamos fazer a calibração para V e B-V usando a tarefa **fitparams** usando o comando:

```
ecl > epar fitparams
```

e ajustá-lo conforme as instruções a seguir.

```

observations = "sobs"           List of observations files
  catalogs = "stds"            List of standard catalog files
    config = "m92fig"          Configuration file
  parameters = "calib"         Output parameters file
  (weighting = "uniform")      Weighting type (uniform, photometric, equations)
  (addscatter = yes)           Add a scatter term to the weights ?
  (tolerance = 3.0000000000000E-5) Fit convergence tolerance
  (maxiter = 15)               Maximum number of fit iterations
  (nreject = 0)                Number of rejection iterations
  (low_reject = 3.)            Low sigma rejection factor
  (high_reject = 3.)           High sigma rejection factor
  (grow = 0.)                  Rejection growing radius
  (interactive = yes)          Solve fit interactively ?
  (logfile = "STDOUT")         Output log file
(log_unmatche = yes)           Log any unmatched stars ?
  (log_fit = no)                Log the fit parameters and statistics ?
  (log_results = no)           Log the results ?
  (catdir = "")                 The standard star catalog directory
  (graphics = "stdgraph")      Output graphics device
  (cursor = "")                 Graphics cursor input
  (mode = "ql")

```

Essa é uma tarefa interativa que permite verificar os ajustes. Os principais comando são:

```

:vshow → mostra os erros e resultados do ajuste
:results → lista de resultados do ajuste
q → vai para o próximo ajuste

```

Para ver a calibração, digite:

```
ecl > page calib
```

O último passo é aplicar a calibração para as magnitudes instrumentais. Isto é feito com a tarefa **invertfit**. Execute essa tarefa com o comando:

```
ecl > epar invertfit
```

e modifique-o de acordo com a figura a seguir.

```

observations = "sobs"           List of observations files
  config = "m92fig"            Configuration file
  parameters = "calib"         Fitted parameters file
  calib = "results"            Output calibrated standard indices file
  (catalogs = "stds")          List of standard catalog files
  (errors = "obserrors")       Error computation type (undefined, obserrors, equ
  (objects = "all")            Objects to be fit (all, program, standards)
  (print = "")                  Optional list of variables to print
  (format = "")                 Optional output format string
  (append = no)                 Append output to an existing file ?
  (catdir = "")                 The standard star catalog directory
  (mode = "ql")

```

Veja os resultados no arquivo *results* no seu diretório. Como exercício, pode-se fazer a fotometria novamente alterando as aberturas para ver se ocorrem mudanças.