

Guia para aprender a usar programas para lidar com dados óticos

Alessandro Marins
7 de junho de 2019

INTRODUÇÃO

Meu objetivo neste texto é construir um guia prático para a utilizações de programas tais como o CFITSIO e Healpix apresentando como instalá-los e fazer exemplos simples destes. Adicional, mas não necessário, descreverei como instalar o ifort, compilador de Fortran da intel. Infelizmente este não é gratuito. Contudo, alunos e docentes vinculados à Universidade de São Paulo (USP) podem obter uma versão gratuitamente¹.

ORIENTAÇÕES PRELIMINARES

Suporei que o leitor acabou de instalar a distribuição Ubuntu. Então, primeiro é preciso criar a senha de super-usuário e instalar alguns programas, via terminal. Para abrir o terminal basta teclar "Ctrl+Alt+T". Em seguida, será necessário instalar alguns programas que serão úteis. Uma linha de comando no terminal será representada por >, ao fim desta, aperte "ENTER".

- > sudo passwd
- > sudo apt-get update && sudo apt-get upgrade
- > sudo apt-get install gfortran
- > sudo apt-get install g++

¹O meu processador é um i7-8550U CPU @ 1.80GHz × 8, com memória RAM de 16GB. Meu sistema é um Ubuntu 18.04.

```

> sudo apt-get install make
> sudo apt-get install cython
> sudo apt-get install python-pip
> sudo apt-get install python-numpy
> sudo apt-get install python-matplotlib
> sudo apt-get install python-scipy
> sudo apt-get install liblapack-dev
> sudo pip install pyfits
> sudo pip install getdist
> sudo pip install astropy
> sudo pip install healpy
> sudo apt-get install gedit
> sudo apt-get install gnuplot
> sudo apt-get install gnuplot-x11
> sudo apt-get install build-essential

```

INSTALANDO O CFITSIO

Para conseguirmos trabalhar com arquivos FITS (Flexible Image Transport System), ou seja, com extensões ".fits", precisaremos instalar um programa (em C e Fortran) que lê e escreve este tipo de arquivo e para isto instalaremos o [CFITSIO](#). Eu organizo meus programas, "Programacao", em um diretório somente de programas, dentro dos "Documentos" e o nome do meu usuário é "cosmo". Com isso, os caminhos (path) aos diretórios que deixarei os programas será: /home/cosmo/Documentos/Programacao/, e cada programa terá sua pasta dentro do diretório "Programacao". Aqui, será "cfitsio", ou seja, a path do cfitsio será: /home/cosmo/Documentos/Programacao/cfitsio. Siga as instruções abaixo, trocando os nomes das "paths" pelas "paths" do seu computador.

```

> cd
> wget http://heasarc.gsfc.nasa.gov/FTP/software/fitsio/c/cfitsio_latest.tar.gz
> mv cfitsio_latest.tar.gz /home/cosmo/Documentos/Programacao/
> cd Documentos/Programacao/
> tar zxvf cfitsio_latest.tar.gz

```

```

> cd cfitsio/
> ./configure --prefix=/home/cosmo/Documentos/Programacao/cfitsio
> make
> make install
> cd
> vi .bashrc

```

Agora, devemos adicionar ao final da bash a path do programa.

```
export LD_LIBRARY_PATH=/home/cosmo/Documentos/Programacao/cfitsio/lib:$LD_LIBRARY_PATH
```

Saia. "Esc", ":wq!". Por fim, atualize a bash.

```
> source .bashrc
```

INSTALANDO O HEALPIX

Entre no site <https://sourceforge.net/projects/healpix/> e baixe a penúltima² versão, aqui a versão é a 3.40. Então vá aos Downloads, onde está o arquivo tar.gz e descompacte-o.

```

> cd Downloads
> unzip Healpix_3.40_2018Jun22.zip
> mv Healpix_3.40 /home/cosmo/Documentos/Programacao/
> cd
> cd Documentos/Programacao/Healpix_3.40
> ./configure

```

O terminal será direcionado à instalação. Primeiro, digite "3"³, para fazer o procedimento via Fortran⁴, em seguida nomeie um sufixo para os diretórios, eu utilizei "_hy". Então siga os procedimentos

```

> Y
> "Enter"
> "Enter"

```

²A versão 3.50 está com problemas e não entra na parte de instalação.

³Aqui estou procedendo para configurar em Fortran, contudo, a depender de como o Healpix for utilizado poderá ser necessário fazer em outra linguagem. Por exemplo, um código que utilizamos é o [FLASK](#) e ele necessita que a configuração do HEALpix seja em C++. Uma forma de contornar isto pode ser usar pipenv ou virtualenv.

⁴caso tenha instalado o ifort, informe o compilador como "ifort"

```
> "Enter"
> "Enter"
> "Enter"
> "Enter"
> /home/cosmo/Documentos/Programacao/cfitsio
> N
> "Enter"
> "Enter"
> "Enter"
> "Enter"
> "Enter"
```

Pronto. Ele voltará a tela inicial de instalação, basta agora apertar "0" e "Enter". Em seguida

```
> make
```

Por fim, basta testar.

```
> make test
```

```

nu3= 0.9600000000000000
Relative skewness= 0.695246330426225
NG map MIN and MAX: -5.621908 5.472547
rms value of the pixels is 0.999680621520008
Computing the values of the a_{lm}
rms value of the a_{lm} is 8.009140860416993E-003
Expected rms value is 7.994739905831941E-003
-----
Number of OpenMP threads in use: 8
Number of CPUs available: 8
-----
sky_ng_sim> Writing sky map to FITS file

Report Card for sky_ng_sim simulation run
-----

Input power spectrum : cl.fits
Multipole range : 0 < l <= 256
Number of pixels : 196608
Gauss. FWHM in arcmin: 30.000
Output map : !test_ngfs.fits
Clock and CPU time [s] : 0.08 0.09

sky_ng_sim> normal completion

process_mask 3.40
*** Mask processing ***
Reading run parameters from prmask.par
parameters not defined in that file will be set to their default value
Parser: mask_file = mask.fits
Parser: hole_min_size = 0 <default>
Parser: hole_min_surf_arcmin2 = 0.000000000000000E+000 <default>
Parser: filled_file = '' <default>
Parser: distance_file = !test_distmask.fits
-----
Number of OpenMP threads in use: 8
Number of CPUs available: 8
-----
process_mask> Converting RING -> NESTED

Report Card for process_mask run
-----

Input binary mask : mask.fits
Total number of pixels : 3072
Input valid pixels : 3044
Effective min. hole size: 0
Filled-in pixels : 0
Output distance file : !test_distmask.fits
Clock and CPU time [s] : 0.01 0.08

process_mask> normal completion
Healpix F90 tests done
success rate: 12/12

```

Figura 1: Parte final do teste do Healpix 3.40.

JUPYTER/JUPYTERLAB

Muitos códigos que utilizo e disponibilizo são escritos em um tipo de arquivo ".ipynb" que chamamos de "notebooks", que são códigos escritos em JUPYTER, um programa que trabalha com linguagens Python, Julia e R, além de suportar escritas que incluem a formatação LaTeX. Ele funciona por blocos, o que facilita, em códigos não muito grandes e complexos, a visualização. Ou seja, muitos códigos em python estarão em "notebooks". Eu costumo usar uma extensão do JUPYTER chamado de JUPYTER LAB. Se eu não me engano, para quem usa Anaconda o JUPYTER já vem instalado, mas o JUPYTER LAB não. Fica a critério do leitor usar ou não o JUPYTER LAB, o necessário é ter um dos dois.

Para instalar o JUPYTER entre neste [link](#), e siga as instruções.

Para instalar o JUPYTER LAB entre neste [link](#), ou simplesmente digite no terminal

```
> pip install jupyterlab
```

EXEMPLOS PRÁTICOS PARA USAR HEALPIX E LIDAR COM ARQUIVOS FITS.

ALGUMAS INFORMAÇÕES SOBRE OS DADOS ÓTICOS

Primeiro, vocês precisarão criar uma conta no servidor do telescópio [Pan-STARRS1](#). Para isto, entrem neste [link](#) e criem. Será necessário terem usuário e senha para que consigam acessar o banco de dados pelo código que disponibilizarei a vocês. Também será necessário que instalem o programa [MAST CasJobs](#),

```
> pip install git+git://github.com/dfm/casjobs@master
```

```
> pip install git+git://github.com/rlwastro/mastcasjobs@master
```

Há dois "notebook" que exemplificam como trabalhar com o servidor: [ps1_dr2_api](#) e [ps1_dr2_query](#). Provavelmente vocês não precisarão utilizá-los, pois o que era necessário eu já fiz no código que será utilizado, mas fica como referências caso surja alguma necessidade.

Caso queiram aprender mais sobre o Pan-STARRS1 (o que seria ótimo!) os artigos recomendados são:

- [The Pan-STARRS1 Surveys](#)
- [The Pan-STARRS Data Processing System](#)
- [Pan-STARRS Pixel Processing: Detrending, Warping, Stacking](#)
- [Pan-STARRS Pixel Analysis : Source Detection and Characterization](#)
- [Pan-STARRS Photometric and Astrometric Calibration](#)
- [The Pan-STARRS1 Database and Data Products](#)
- [Pan-STARRS1: Galaxy Clustering in the Small Area Survey 2](#)

Este último é o artigo que descreve a seleção de galáxias pelo telescópio.

⁵COMPILADOR DE FORTRAN DA INTEL: IFORT

Há programas tais como o CLASS e o CosmoMC que fortemente recomendo a utilização do ifort, ao invés do gfortran. O ifort leva à um ganho substancial de tempo. Na última atualização do CAMB pelo Antony Lewis, agosto de 2018, este recomenda a utilização do ifort e diz:

On linux system now builds with ifort if available (35% faster)

Ao se referir ao *wrapper* do CAMB para python. Abra o terminal e digite "pwd". Ele lhe mostrará a sua *home path*, no meu caso é: **/home/cosmo**. Eu baixei o arquivo da intel direto do site e ele foi direcionado à pasta **Downloads**, então digitei "cd Downloads" e me direcionei à pasta, no meu caso o caminho é **/home/cosmo/Downloads**⁶. Agora precisamos descompactar o arquivo, no meu caso o arquivo **parallel_studio_xe_2019_update3_cluster_edition.tgz**, para isto: Antes de se direcionar a nova pasta, precisamos entrar como super-usuário de forma a fazer uma instalação não-local do compilador. Para isto, basta digitar "su" e fornecer a senha que criamos inicialmente. Agora existirá uma pasta com o mesmo nome do arquivo com extensão **.tgz**, então entre nesta pasta:

```
> cd parallel_studio_xe_2019_update3_cluster_edition.tgz
```

Você pode verificar o que há nesta pasta utilizando o comando "l" ou o "ls". Agora entraremos no módulo de instalação via terminal, digite nesta pasta

```
> ./install.sh
```

Você utilizará aqui o *serial number* fornecido quando baixou o arquivo. A instalação é simples, basta seguir as instruções, inicialmente apertar "Enter", depois, após ler os termos de licença, escrever "accept" e apertar "Enter". Então, digitar "1", consentindo que a intel tenha acesso às suas informações. Então pressionar "Enter" para fornecer a *serial number*, em seguida forneça-a. Ele irá checar esta e prosseguirá a instalação. Pressione "Enter" e "Enter", caso queira fazer a instalação padrão. Depois, leia as informações, pressione "Enter" e "Enter", mantendo o default. Após estes, a instalação será realmente feita, podendo levar alguns minutos (~ 10, 15 min). Por fim, neste processo, basta apertar "Enter". Ele voltará à *path*. Agora teremos de entrar na *bash* do terminal e informar a localização do compilador.

Caso tenha seguido todos os passos até aqui descritos poderá verificar que o compilador foi instalado na pasta **/opt**. Poderá verificar que há nesta a pasta **/intel**, ou seja, o caminho é **/opt/intel**.

Como teremos que incluir os caminhos na **.bashrc** aconselho utilizar o programa Vim. Para isto,

```
> sudo apt install vim7
```

Então, entramos na **.bashrc** via Vim

```
> vim .bashrc
```

Aperte "Enter", vá até o final no arquivo pressione "i", ou o teclado *insert*, e insira (aqui irei pôr a minha versão)

⁵Não é obrigatório ter este compilador. Ele costuma ter uma performance melhor que o gfortran (gratuito) mas que somente será perceptível em trabalhos mais específicos.

⁶Verifique este caminho digitando "pwd", novamente. Digite "ls" para verificar se há o arquivo da intel nesta pasta.

⁷Pode-se utilizar também o *gedit*. A edição então será em um gerenciador de texto.

```
> source /opt/intel/parallel_studio_xe_2019.1.053/compilers_and_libraries_2019/linux/bin/ifortvars.sh  
intel64
```

Então aperte "Esc" e digite ":wq", sairá do Vim. Agora teste se achará o compilador. Atualize a bash digitando

```
> source .bashrc
```

```
> ifort -v
```

Ele deverá fornecer a versão do ifort.

REFERÊNCIAS

- [1] Li, Ming-Hua, et al. "CosmoMC Installation and Running Guidelines."arXiv preprint arXiv:1409.1354 (2014).
- [2] Mos' hafi, Hossein. "A Guide for CosmoMC Installation and Running."arXiv preprint arXiv:1808.05080 (2018).
- [3] <http://prestrellocristiano.blogspot.com/2012/05/instalando-o-compilador-fortran-da.html>