

EXPLORATION D'UN FICHIER FITS GÉNÉRÉ PAR ASTROMETRY

Les fichiers au « format » FITS sont très mystérieux pour de nombreux astronomes amateurs qui ne comprennent pas pourquoi ils ne peuvent pas les afficher dans les logiciels de traitement d'images usuels tels que Gimp (logiciel libre) ou Photoshop (marque commerciale).

Une association loi 1901 telle que l'Association Astronomique de l'Indre est un lieu d'échange de pratiques et de savoir et, même si le sujet est assez complexe, il est pertinent de l'aborder, ne serait-ce qu'à travers un exemple.

Le fichier qui sert de support aux manipulations a été généré par Astrometry.net (en local). Il contient des données intermédiaires du traitement, c'est à dire les coordonnées (X et Y) des « étoiles » (les points plus ou moins clairs sur l'image) dans la photographie traitée.

Les lecteurs qui souhaitent faire les manipulations peuvent le récupérer ici :

<http://lerautal.lautre.net/journal/AAI/Astrometry/cible.axy>

1. Premiers éléments sur les fichiers FITS

Cité depuis l'article de Wikipedia : https://fr.wikipedia.org/wiki/Flexible_Image_Transport_System
« FITS ou Flexible Image Transport System est le format de fichiers le plus communément utilisé en astronomie. »

On peut y mettre à la fois des données permettant d'afficher (et de traiter) une image, mais aussi des mesures de spectres ou toute autre mesures physiques. En fait c'est une sorte de « fourre-tout » qui permet d'enregistrer et de transmettre des données scientifiques. À l'usage ce format s'est révélé tellement souple qu'on l'utilise pour tout autre chose que des images... d'où la déconvenue quand on essaie de les afficher.

Mais alors, comment s'y retrouver quand on n'est pas astronome professionnel ?

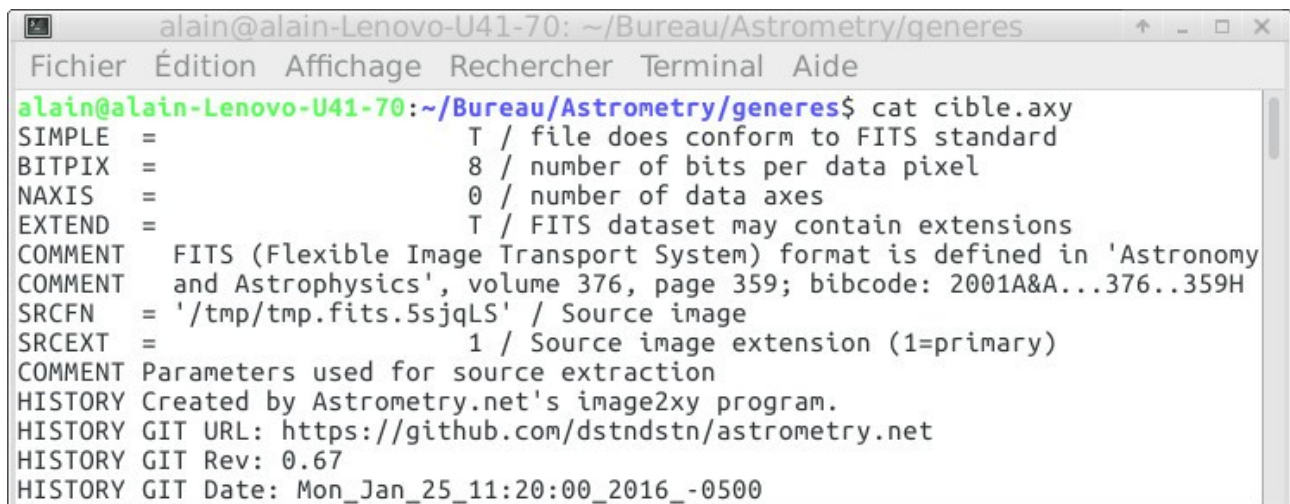
Retenir déjà une chose : un fichier FITS contient toujours au **moins un en-tête** suivi de données.

L'en-tête est obligatoirement au format texte, lisible par tout être humain.

Afficher le contenu du fichier exemple.

Sous Linux, on utilise la commande « cat » suivie du nom de fichier. (équivalent Windows : « type »).
cat cible.axy

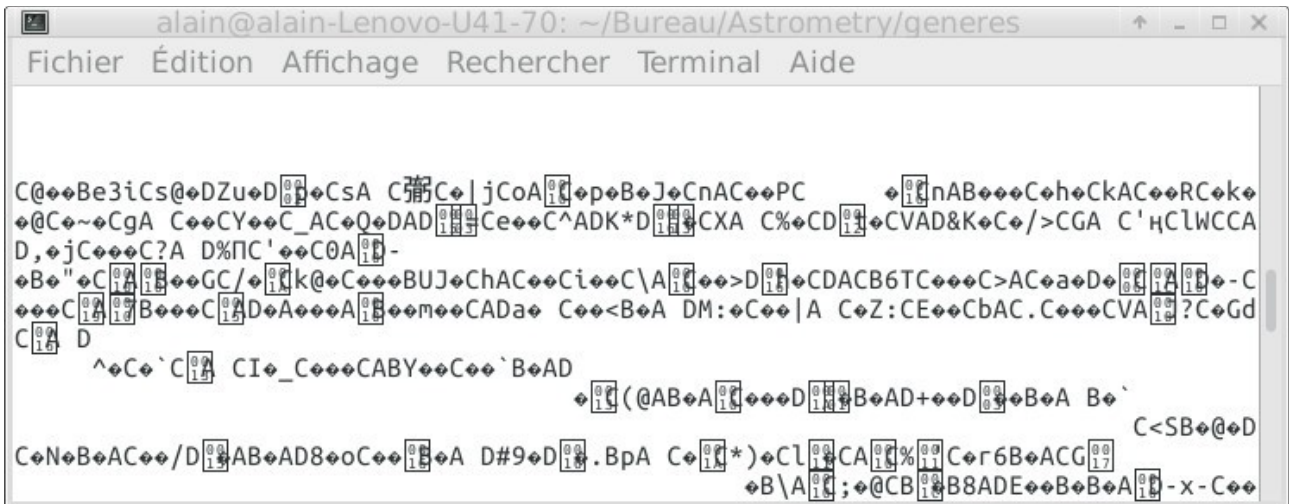
On obtient d'abord l'en-tête (la suite n'est pas listée ici)



```
alain@alain-Lenovo-U41-70: ~/Bureau/Astrometry/generes
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
alain@alain-Lenovo-U41-70:~/Bureau/Astrometry/generes$ cat cible.axy
SIMPLE = T / file does conform to FITS standard
BITPIX = 8 / number of bits per data pixel
NAXIS = 0 / number of data axes
EXTEND = T / FITS dataset may contain extensions
COMMENT FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
SRCFN = '/tmp/tmp.fits.5sjqLS' / Source image
SRCEXT = 1 / Source image extension (1=primary)
COMMENT Parameters used for source extraction
HISTORY Created by Astrometry.net's image2xy program.
HISTORY GIT URL: https://github.com/dstndstn/astrometry.net
HISTORY GIT Rev: 0.67
HISTORY GIT Date: Mon_Jan_25_11:20:00_2016_-0500
```

Celui-ci donne toutes les informations que le créateur de l'image a jugé utile d'inclure : dimension de l'image, lieu de mesure, auteur...

puis les données :



Celles-ci ne sont pas au format texte : elles sont binaires et la commande « cat » ne sait pas lister les contenus. Par défaut elle retourne de « faux caractères » qui ne signifient rien pour un humain.

2. Utiliser un langage de programmation pour accéder aux contenus

Les contenus possibles des fichiers FITS sont tellement variables qu’il n’existe pas d’outil généraliste pour accéder à l’information. Le plus simple est encore d’apprendre un minimum de programmation avec un langage efficace et d’écrire les commandes qui donneront les résultats.

Dans la suite de cet exemple, on utilise le langage Python, mais on pourrait en utiliser d’autres si on en connaît.

Les possibilités du « Python de base » sont étendues grâce à une bibliothèque de fonctions d’accès aux fichiers FITS, gratuite, développée par des bénévoles. Elle s’appelle « pyfits ».

2.1. Remarques sur le code Python présenté ici

Il n’est pas « beau », mais simpliste : il s’adresse à des presque débutants et veut être compris par eux. La fonction « print(...) » permet d’afficher un ou des résultats à l’écran.

2.2. Ouvrir et fermer le fichier cible

Aucun résultat n’apparaîtra... si ce n’est un message d’erreur si le fichier est absent ou si l’on a mal tapé le code.

# -*- coding:Utf-8 -*-	Nécessaire pour la prise en compte des caractères accentués.
from __future__ import print_function	Pour pouvoir utiliser « bien » la fonction print.
import pyfits as pf	Accéder à la bibliothèque pyfits.
nom_fichier = 'cible.axy'	On utilise une variable pour le nom du fichier.
hdulist = pf.open(nom_fichier)	Ouvrir le fichier.
hdulist.close()	Fermer le fichier.

2.3. Information sur les zones de données

On ajoute simplement la ligne suivante :

```
print(hdulist.info())
```

Que l’on précède de quelques commentaires, ce qui donne le code suivant :

```

# -*- coding:Utf-8 -*-
from __future__ import print_function
import pyfits as pf

nom_fichier = 'cible.axy'
hdulist = pf.open(nom_fichier)
print("TITRE : Exploration du fichier ",nom_fichier)
print("-----")
print("-- Informations sur les tables du fichier :")
print(hdulist.info())
hdulist.close() # Fermeture du fichier

```

Il en résulte la listage suivant à l'écran :

```

TITRE : Exploration du fichier cible.axy
-----
-- Informations sur les tables du fichier :
Filename: cible.axy
No.      Name          Type          Cards  Dimensions  Format
0        PRIMARY        PrimaryHDU    34     ()
1        SOURCES         BinTableHDU   39     576R x 4C   [E, E, E, E]
None

```

Nous savons maintenant que le fichier comprend deux zones, marquées 0 et 1.

La première correspond à l'en-tête. La seconde contient les données. Celles-ci contiennent 576 lignes (576R : R = Row , soit ligne en anglais).

Chaque ligne comprend 4 Colonnes, qui sont regroupées entre les signes « [] ».

2.4. Lister l'en-tête

Il suffit d'ajouter les lignes suivantes au programme :

```
print("-- Contenu de l'en-tête : ")
print(hdulist[0].header)
```

La variable `hdulist` contient la totalité des données du fichier. Le « [0] » associé signifie que l'on veut accéder à la zone « 0 ». Le mot « header » désigne l'en-tête.

La capture, page suivante, montre le résultat.

Il se termine par le mot conventionnel « END » qui désigne la fin de l'en-tête.

Mais... l'information listée par la commande « cat », n'est pas complète : il manque la moitié de l'en-tête.

```
-- Contenu de l'en-tête :
SIMPLE = T / file does conform to FITS standard
BITPIX = 8 / number of bits per data pixel
NAXIS = 0 / number of data axes
EXTEND = T / FITS dataset may contain extensions
COMMENT FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
SRCFN = '/tmp/tmp.fits.5sjqLS' / Source image
SRCEXT = 1 / Source image extension (1=primary)
COMMENT Parameters used for source extraction
HISTORY Created by Astrometry.net's image2xy program.
HISTORY GIT URL: https://github.com/dstndstn/astrometry.net
HISTORY GIT Rev: 0.67
HISTORY GIT Date: Mon_Jan_25_11:20:00_2016_-0500
HISTORY Visit us on the web at http://astrometry.net/
NEXTEND = 1 / Number of extensions
LONGSTRN= 'OGIP 1.0' / The OGIP long string convention may be used
COMMENT This FITS file may contain long string keyword values that are
COMMENT continued over multiple keywords. This convention uses the '&'
COMMENT character at the end of the string which is then continued
COMMENT on subsequent keywords whose name = 'CONTINUE'.
IMAGEW = 906 / image width
IMAGEH = 648 / image height
ANRUN = T / Solve this field!
ANVERUNI= T / Uniformize field during verification
ANVERDUP= F / Deduplicate field during verification
ANAPPL1 = 3.97350993377 / scale: arcsec/pixel min
ANAPPU1 = 15.8940397351 / scale: arcsec/pixel max
ANTWEAK = T / Tweak: yes please!
ANTWEAKO= 2 / Tweak order
ANSOLVED= 'generes/cible.solved' / solved output file
ANMATCH = 'generes/cible.match' / match output file
ANRDLS = 'generes/cible.rdls' / ra-dec output file
ANWCS = 'generes/cible.wcs' / WCS header output filename
ANCORR = 'generes/cible.corr' / Correspondences output filename
END
```

Mystère...

2.5. Lister d'autres éléments sur la structure des données.

Le code (presque) entier est listé ci-dessous. Les lignes qui commencent par un # sont des commentaires utiles pour la re-lecture du code et ne sont pas exécutées.

```
# -*- coding:Utf-8 -*-
from __future__ import print_function
import pyfits as pf

nom_fichier = 'cible.axy'
hdulist = pf.open(nom_fichier)
print("TITRE : Exploration du fichier ",nom_fichier)
print("-----")
print("-- Informations sur les tables du fichier :")
print(hdulist.info())
print("-- Contenu de l'en-tête : ")
print(hdulist[0].header)
# Chargement des données dans un conteneur
scidata = hdulist[1].data
# Chargement des informations sur les colonnes du tableau
cols = hdulist[1].columns

hdulist.close() # Fermeture du fichier fit
print("-- Information sur la structure du tableau : ")
print(scidata.shape)
print("-- Types de contenus : ")
print(scidata.dtype.name)
print("-- Exemple de contenu d'une ligne du tableau : ")
print(scidata[0])
print("-- Noms des colonnes : ")
print(cols.names)
print("-- Information plus complète sur les colonnes : ")
print(cols.info())
```

Commentaires sur le listage obtenu en plus.

-- Information sur la structure du tableau :	Message généré par print(...)
(576,)	Le tableau contient 576 lignes
-- Types de contenus :	Message généré par print(...)
record128	Définit une taille d'enregistrement.
-- Exemple de contenu d'une ligne du tableau :	Message généré par print(...)
(192.96024, 57.300205, 243.0, 7.0)	Première ligne. Une ligne contient quatre éléments (les quatre colonnes annoncées)
-- Noms des colonnes :	Message généré par print(...)
['X', 'Y', 'FLUX', 'BACKGROUND']	Coordonnées de l'étoile dans l'image, intensité lumineuse, intensité du fond.
-- Information plus complète sur les colonnes :	Message généré par print(...)
name:	
['X', 'Y', 'FLUX', 'BACKGROUND']	Noms des colonnes.

(Le listage a été coupé ici.)

2.6. Lister les données

Il suffit d'ajouter ces deux lignes à la fin du code :

```
for i in range(0,576):  
    print(i, ";", scidata[i])
```

Cela va générer les 576 lignes de contenu dont voici les quatre premières.

```
0 ; (192.96024, 57.300205, 243.0, 7.0)  
1 ; (873.84271, 521.76074, 243.0, 10.0)  
2 ; (455.27744, 472.97198, 239.0, 9.0)  
3 ; (278.87976, 115.14592, 238.0, 8.0)  
4 ; (455.38525, 137.94174, 238.0, 8.0)
```

Signification de chaque ligne :

numéro de ligne puis coordonnées de l'étoile dans l'image, intensité lumineuse, intensité du fond.

3. En guise de conclusion

Les explications données ici sont minimalistes. Le programme Python a été peu commenté.

L'intention était de donner une idée, même un peu superficielle, de l'exploitation d'un fichier FITS quitte à être bref.