



## SIRIL 1.3 Notes (Décembre 23)

The screenshot displays the SIRIL 1.2.0 software interface. The main window shows a star field image with a bright star in the center. The console window on the right contains the following log entries:

```
11:41:46: Rejet des pixels dans le canal #1 : 0.209% - 0.385%
11:41:46: Rejet des pixels dans le canal #2 : 0.259% - 0.390%
11:41:47: Empilement avec rejet fini. 155 images ont été empilées.
11:41:47: Intégration de 155 images sur 155 de la séquence :
11:41:47: Combinaison ..... moyenne
11:41:47: Normalisation en entrée ... additive + mise à l'échelle
11:41:47: Normalisation en sortie ... activée
11:41:47: Rejet des pixels ..... winsorized sigma clipping
11:41:47: Paramètres de rejet ..... bas=3.000 haut=3.000
11:41:47: Création d'une carte de rejet ... non
11:41:47: Pondération ..... désactivée
11:41:47: Égalisation RVB ..... désactivée
11:41:47: Estimation du bruit (canal : #0) : 2.364 (3.607e-05)
11:41:47: Estimation du bruit (canal : #1) : 2.250 (3.434e-05)
11:41:47: Estimation du bruit (canal : #2) : 2.193 (3.347e-05)
11:41:47: Fichier FITS enregistré : fichier ../result.fit, 3 canal(aux), 5496x3672 pixels, 32bits
11:41:47: Séquence empilée avec succès.
11:41:47: Temps d'exécution: 2 min 13 s
11:41:47: Exécution de la commande : cd
11:41:47: Définir le répertoire de travail à 'C:\0_KWORK_2023\Francis\Astro Traitements SIRIL M31 CALC oct 23'
11:41:47: Exécution de la commande : close
11:41:47: Définir le répertoire de travail à 'C:\0_KWORK_2023\Francis\Astro Traitements SIRIL M31 CALC oct 23'
11:41:47: L'exécution du script s'est terminée avec succès.
11:41:47: Temps d'exécution total: 12 min 57 s
11:55:03: Lecture FITS : fichier result.fit, 3 canal(aux), 5496x3672 pixels, 32bits
11:55:08: Le mode d'affichage auto-ajustement utilise une table de 16 bits
11:56:43: Annuler : Recadrage (x=2572, y=2927, w=90, h=90)
```

The interface also shows a command line at the bottom with the text "Ligne de commande" and a red "Arrêter" button. The status bar at the bottom left indicates "10% fwhm : 4.32 px, r : 0.46 L : 25px H : 19px ratio : 1.3158".

Document réalisé par Mickaëlle B & Francis R.

## Avant Propos

- tutos de SIRIL :

<https://siril.org/fr/tutorials/>

<https://www.webastro.net/noctua/astrophotographie/tutorial-siril-complet-pour-apn-fichiers-raw-et-camera-fichiers-fits-r185/>

- doc sur le flow manuel:

-page sur les forums

<https://www.astronomie-va.com/forum/viewtopic.php?p=143128#p143128>

-vidéos

-Sirilic :

<https://astroslachonet.wordpress.com/2023/07/31/utilitaire-conv2siril/>

et scripts SIRIL :

<https://siril.readthedocs.io/fr/stable/Scripts.html#using-scripts>

<https://gitlab.com/free-astro/siril-scripts>

## Partie 1: Le Pré-traitement

Le pré-traitement consiste prendre les images brutes pour :

- les reformater au format .fits (qui est le format des images de référence des logiciels astro),
- Puis retirer les différents bruits inhérents au capteur avec le setup avec les DOF (Dark-Offset(bias)-Flat).
- Et finalement aligner et empiler ces images pour en réaliser une.

L'image résultante est une image linéaire qui fera ensuite l'objet de retraitement divers : retrait de gradient, réduction de bruit, ... Ce sera la partie traitement.

SIRIL offre deux modes de pré-traitement :

- Une méthode automatique avec des scripts
- Une méthode manuelle pas à pas.

SIRIL rend disponible quelques scripts disponibles à l'installation. D'autres scripts sont téléchargeables sur Gitlab.

Présentation des points abordés à la première session par Philippe :

- Session 1 : le prétraitement des images
  - Le principe de prétraitement des images
  - Les formats des images
  - Présentation rapide de l'interface de SIRIL
  - Le mode automatique et les scripts
  - Le mode manuel
    - Créer une séquence d'images lors de la conversion
    - Une logique de traitement des séquences avec SIRIL :
      - Conversion, Calibration, Alignement, Graphique, Empilement
    - Traiter les darks et le offsets
    - Traiter les flats
    - Traiter les images (lights)
      - Calibration = corriger des darks et des flats
      - Aligner les images
      - Trier les images
      - Empiler les images

## A) Méthode automatique

Je m'appuierai sur le tutoriel officiel de SIRIL : <https://siril.org/fr/tutorials/tuto-scripts/>

Et les documents réalisés par Philippe F. pour les ateliers de prise en main de SIRIL pour la SAR en Octobre 23.

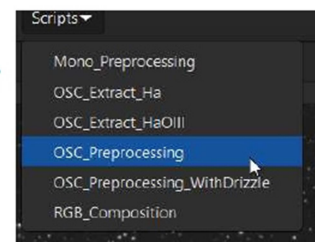
### 1) Principes et organisation

## Prétraitement avec script

- Utiliser un script pour un prétraitement automatique
  - Faire simple
  - Aller vite pour avoir une idée de la série
  - Pour faire des traitements spécifiques répétés
- Ranger les images dans 4 répertoires en respectant bien les noms (biases, darks, flats, lights)

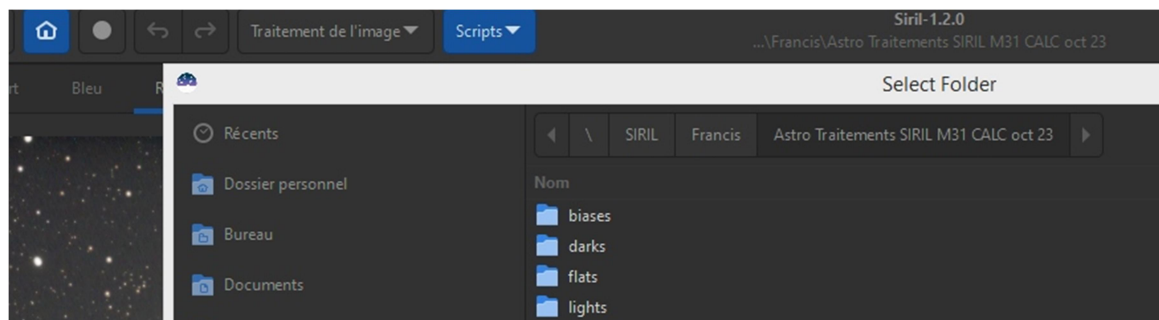
Nom	Modifié le	Type	Taille
biases	30/06/2023 21:04	Dossier de fichiers	
darks	15/07/2023 15:29	Dossier de fichiers	
flats	15/07/2023 14:49	Dossier de fichiers	
lights	15/07/2023 14:45	Dossier de fichiers	

- Aller dans le menu script et sélectionner OSC\_Preprocessing pour une image couleur



Source : Atelier de prise en main de SIRIL – SAR – Philippe F. – Octobre 23

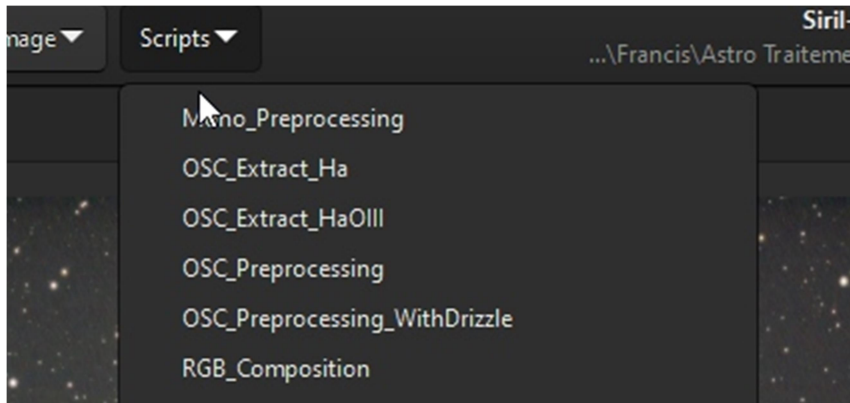
La méthode automatique (script) fonctionne sur la base d'une architecture de sous-répertoire bien précise. Ensuite, il faut définir le répertoire qui contient ces répertoires comme répertoire de travail : cliquez sur la petite maison bleue en haut à droite et aller chercher le répertoire contenant les 4 sous-répertoires : biases, darks, flats et lights.



*Notes : il est possible de remplacer les offsets réalisés par la caméra par des offsets synthétiques (générés par SIRIL).*

## 2) Les scripts natifs

Les scripts natifs sont disponibles dès l'installation



Descriptifs :

- **Mono\_Preprocessing.ssf** : script pour le prétraitement des appareils photos numériques ou caméras astro monochromes, utilise les biais, les flats et les darks, aligne et empile les images. Pour l'utiliser : mettez vos fichiers (RAW ou FITS) dans les dossiers nommés `lights`, `darks`, `flats` et `biases` (dans le dossier de travail par défaut de Siril), puis lancez le script.
- **OSC\_Preprocessing.ssf** : même script que ci-dessus mais pour les appareils photos numériques ou caméras astro couleur. Pour l'utiliser : mettez vos fichiers (RAW ou FITS) dans les dossiers nommés `lights`, `darks`, `flats` et `biases` (dans le dossier de travail par défaut de Siril), puis lancez le script.
- **OSC\_Extract\_Ha.ssf** : script pour le prétraitement des appareils photos numériques ou caméras astro couleur, à utiliser avec le filtre Ha ou le filtre à double bande. Ce script extrait la couche Ha de l'image couleur. Pour l'utiliser : mettez vos fichiers (RAW ou FITS) dans les dossiers nommés `lights`, `darks`, `flats` et `biases` (dans le dossier de travail par défaut de Siril), puis lancez le script.
- **OSC\_Extract\_HaOIII.ssf** : même script que ci-dessus, mais extrait les couches Ha et OIII de l'image couleur. Pour l'utiliser : mettez vos fichiers (RAW ou FITS) dans les dossiers nommés `lights`, `darks`, `flats` et `biases` (dans le dossier de travail par défaut de Siril), puis lancez le script. Vous pouvez aussi utiliser le menu `Traitement de l'image` puis `Composition RVB` et mettre le résultat Ha dans le canal rouge et le résultat OIII dans les couches verte et bleue pour obtenir une image HOO.

### Astuce

Pour les possesseurs de filtres SII ou dualband SII-OIII, les mêmes scripts s'appliquent. En effet, il est impossible pour un capteur couleur de voir la différence entre Ha (656.3 nm) et SII (671.6 nm), tous deux de couleur rouge.

- **RGB\_Composition.ssf** : Ce script ajouté dans la version 1.2 aligne des images monochromes avec l'alignement globale, les recadre en utilisant leur zone commune et prend les trois premières images pour créer une image en couleur. Les images d'entrée doivent être placées seules dans un répertoire et nommées `R.fit` (ou avec l'extension configurée), `G.fit` et `B.fit`. Le résultat est nommé `rgb.fit`. Assurez-vous de supprimer le répertoire `process` entre chaque exécution.

Source : <https://siril.readthedocs.io/fr/stable/Scripts.html#troubleshooting>

### 3) Scripts additionnels

#### Obtenir plus de scripts

Il y a tout un tas de scripts qui ne sont pas fournis avec l'installation de Siril. Cependant, nous avons mis en place un dépôt gitlab pour eux. Chacun est libre de s'inscrire et de proposer de nouveaux scripts. Nous les accepterons en fonction de leur pertinence : la langue utilisée doit être l'anglais.

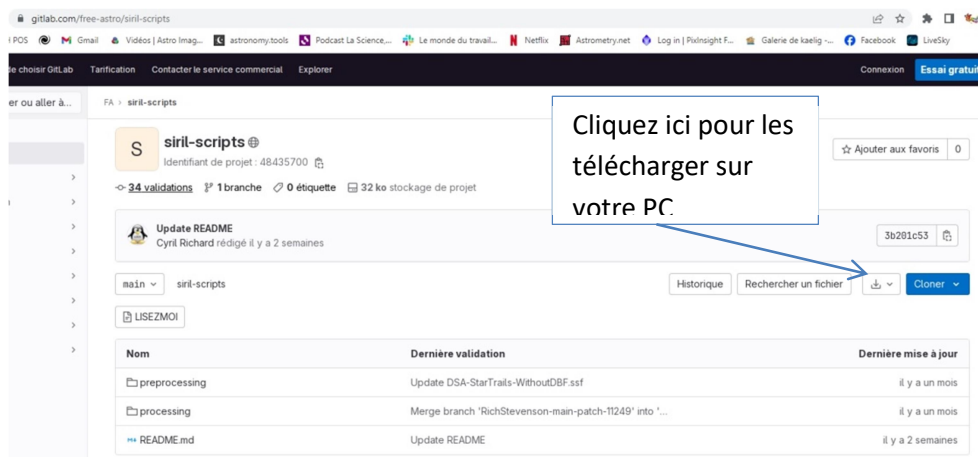
Veillez vous référer à cette adresse pour parcourir et télécharger les scripts :

<https://gitlab.com/free-astro/siril-scripts>.

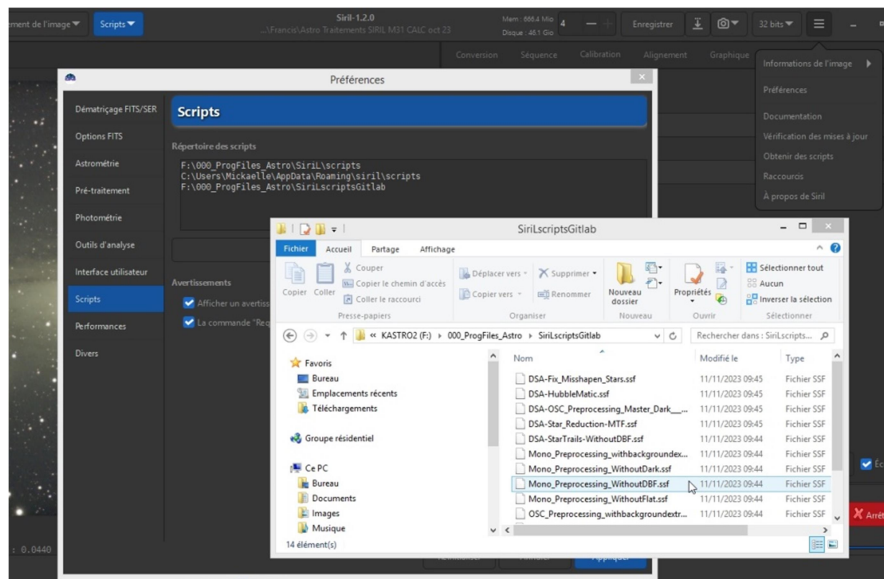
#### ⚠ Avertissement

Gardez toutefois à l'esprit que ces scripts ne sont pas nécessairement maintenus par les utilisateurs qui les ont téléchargés et qu'ils peuvent ne pas être à jour. Cela dit, amusez-vous bien.

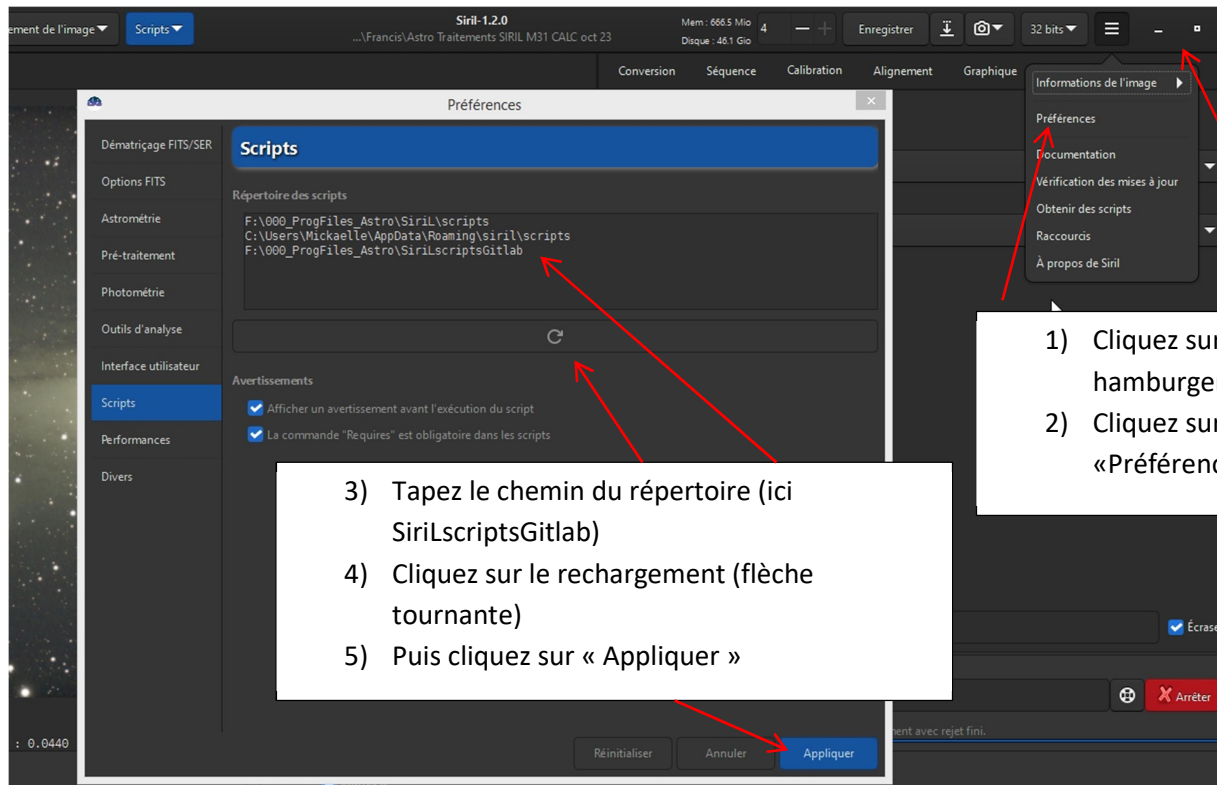
Cliquer sur le lien pour accéder au dépôt Gitlab, puis sur l'onglet « téléchargement »



Vous n'avez plus qu'à décompresser le fichier zip (clic droit souris > Extraire), puis placer le contenu dans un répertoire spécifique. Il faudra ensuite définir le chemin de ce répertoire dans Siril (préférences > script)



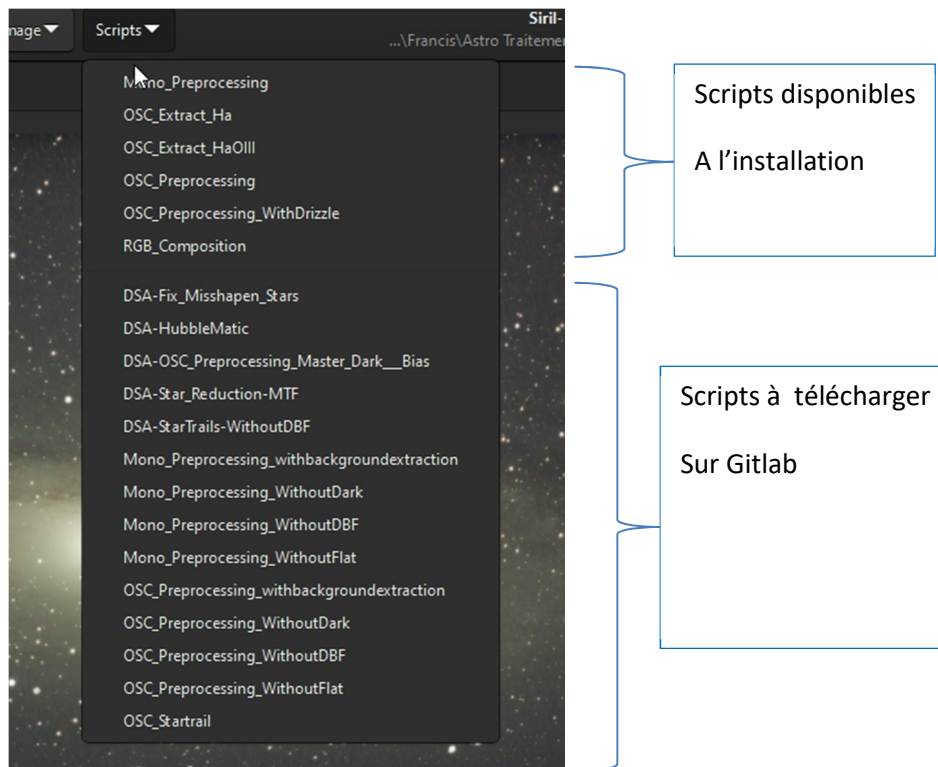
Pour ce faire :



Si vous allez regarder le menu console, vous devriez avoir ces messages :

```
17:11:37: Recherche de scripts dans : "F:\000_ProgFiles_Astro\Siril\scripts" ...
17:11:37: Chargement du script Mono_Preprocessing
17:11:37: Chargement du script OSC_Extract_Ha
17:11:37: Chargement du script OSC_Extract_HaOIII
17:11:37: Chargement du script OSC_Preprocessing
17:11:37: Chargement du script OSC_Preprocessing_WithDrizzle
17:11:37: Chargement du script RGB_Composition
17:11:37: Recherche de scripts dans : "F:\000_ProgFiles_Astro\SirilscriptsGitlab" ...
17:11:37: Chargement du script DSA-Fix_Misshapen_Stars
17:11:37: Chargement du script DSA-HubbleMatic
17:11:37: Chargement du script DSA-OSC_Preprocessing_Master_Dark__Bias
17:11:37: Chargement du script DSA-Star_Reduction-MTF
17:11:37: Chargement du script DSA-StarTrails-WithoutDBF
17:11:37: Chargement du script Mono_Preprocessing_withbackgroundextraction
17:11:37: Chargement du script Mono_Preprocessing_WithoutDark
17:11:37: Chargement du script Mono_Preprocessing_WithoutDBF
17:11:37: Chargement du script Mono_Preprocessing_WithoutFlat
17:11:37: Chargement du script OSC_Preprocessing_withbackgroundextraction
17:11:37: Chargement du script OSC_Preprocessing_WithoutDark
17:11:37: Chargement du script OSC_Preprocessing_WithoutDBF
17:11:37: Chargement du script OSC_Preprocessing_WithoutFlat
17:11:37: Chargement du script OSC_Startrail
```

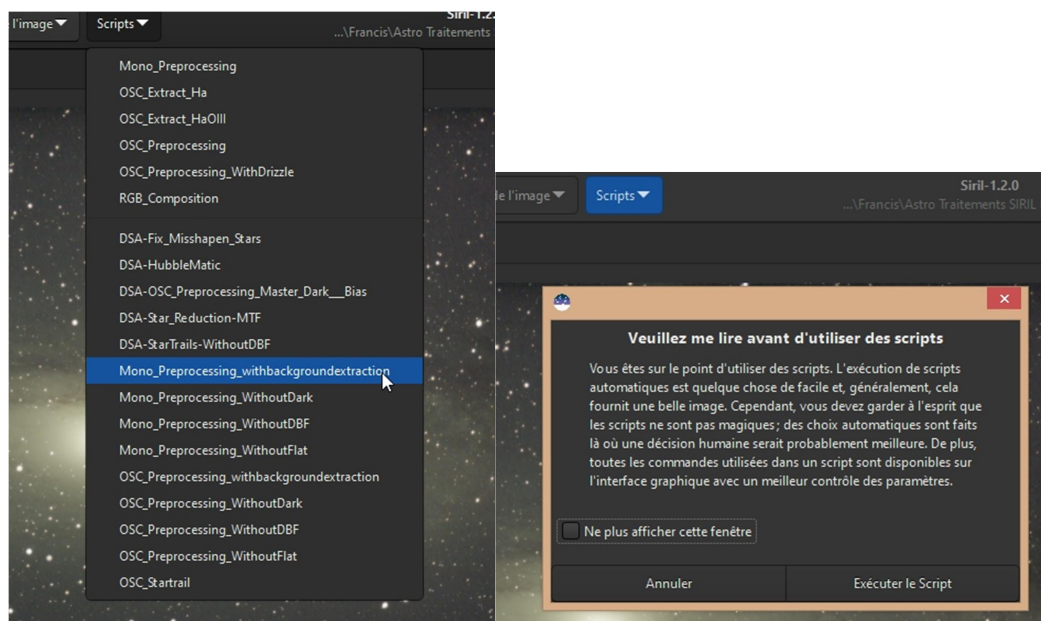
Maintenant, quand vous cliquez sur le menu script, vous devriez obtenir une liste plus substantielle :



#### 4) Mise en œuvre

Une fois les données placées dans les différents répertoires, vérifier que toutes les images sont bien des brutes (RAW ou Fits). Il peut arriver que des fichiers .jpeg y trainent (par exemple si on utilise l'Asiair), il faut les supprimer. Il ne faut qu'un seul type de fichiers.

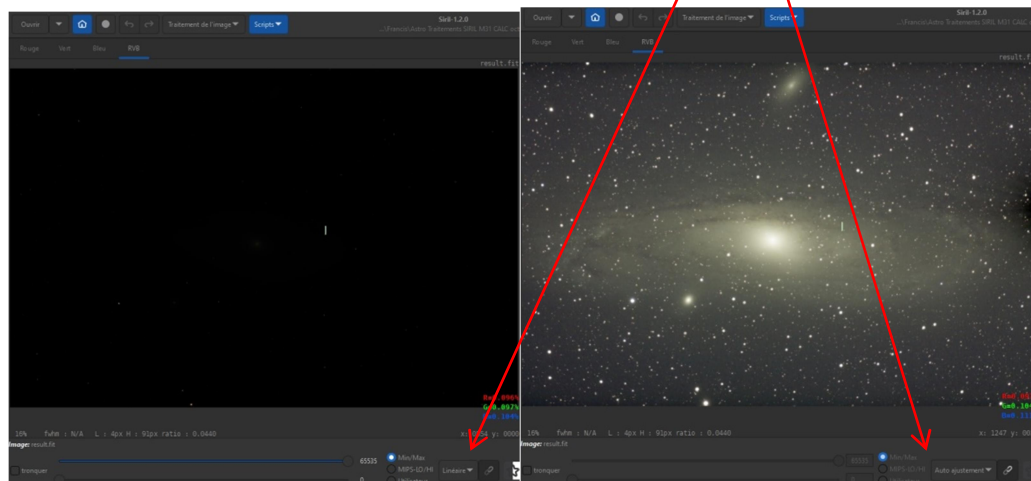
Sélectionner le script qui vous intéresse puis « Executer le script », laissez tourner (allez prendre un café) :



Ensuite, il faut aller dans la console et attendre la fin de l'exécution du script

```
18:13:07: Nous avons 24 blocs parallèles de taille 459 (+0) pour l'empilement.
18:13:07: Débute l'empilement...
18:18:17: Rejet des pixels dans le canal #0 : 0.242% - 0.364%
18:18:17: Rejet des pixels dans le canal #1 : 0.209% - 0.385%
18:18:17: Rejet des pixels dans le canal #2 : 0.259% - 0.390%
18:18:18: Empilement avec rejet fini. 155 images ont été empilées.
18:18:18: Intégration de 155 images sur 155 de la séquence :
18:18:18: Combinaison ..... moyenne
18:18:18: Normalisation en entrée ... additive + mise à l'échelle
18:18:18: Normalisation en sortie ... activée
18:18:18: Rejet des pixels ..... winsorized sigma clipping
18:18:18: Paramètres de rejet ..... bas=3.000 haut=3.000
18:18:18: Création d'une carte de rejet ... non
18:18:18: Pondération ..... désactivée
18:18:18: Égalisation RVB ..... désactivée
18:18:19: Estimation du bruit (canal : #0) : 2.364 (3.608e-05)
18:18:19: Estimation du bruit (canal : #1) : 2.250 (3.433e-05)
18:18:19: Estimation du bruit (canal : #2) : 2.194 (3.347e-05)
18:18:19: Fichier FITS enregistré : fichier ../result.fits, 3 canal(aux), 5496x3672 pixels,
32bits
18:18:19: Séquence empilée avec succès.
18:18:19: Temps d'exécution: 9 min 17 s
18:18:19: Exécution de la commande : cd
18:18:19: Définir le répertoire de travail à 'G:\SIRIL\Francis\Astro Traitements SIRIL M31
CALC oct 23'
18:18:19: Exécution de la commande : close
18:18:20: Définir le répertoire de travail à 'G:\SIRIL\Francis\Astro Traitements SIRIL M31
CALC oct 23'
18:18:20: L'exécution du script s'est terminée avec succès.
18:18:20: Temps d'exécution total: 26 min 05 s
18:21:49: Lecture FITS : fichier result.fits, 3 canal(aux), 5496x3672 pixels, 32bits
```

Il faut ensuite charger le fichier généré (ici : result.fits) en cliquant sur « Ouvrir » (en haut à gauche). On a une image noire qui s'affiche (ce qui est normal vu que l'on est en présence d'une image linéaire). Il faut aller en bas à droite pour changer de linéaire à Auto-Ajustement



Il faut maintenant traiter cette image : retirer du bruit, ajuster la couleur des étoiles, faire ressortir les détails de l'image ... et étirer l'histogramme pour rendre l'image visible (on parle d'image non-linéaire).

Si vous avez des soucis avec les scripts, c'est ici : <https://siril.readthedocs.io/fr/stable/Scripts.html#troubleshooting>

## B] Méthode manuelle

On peut suivre ce tuto pas à pas : <https://siril.org/fr/tutorials/tuto-manual/>

La méthode automatique permet d'appréhender le résultat possible. On peut très bien traiter cette image issue du script car elle nous convient. Mais l'astram est rarement satisfait par un résultat rapide et perfectible :D Pour exploiter au mieux les images récoltées, il faudra passer par une méthode manuelle.

- Le mode manuel
  - Créer une séquence d'images lors de la conversion
  - Une logique de traitement des séquences avec SIRIL :
    - Conversion, Calibration, Alignement, Graphique, Empilement
  - Traiter les darks et le offsets
  - Traiter les flats
  - Traiter les images (lights)
    - Calibration = corriger des darks et des flats
    - Aligner les images
    - Trier les images
    - Empiler les images

*Note : en fin de ce chapitre, j'aborderai SIRILIC qui permet de générer des scripts adaptés à son setup.*

## 1) Principe du pré-traitement

Le calcul des images est fait par la formule suivante :

$$\frac{\text{image} - \text{dark} - \text{offset}}{\text{flat}}$$

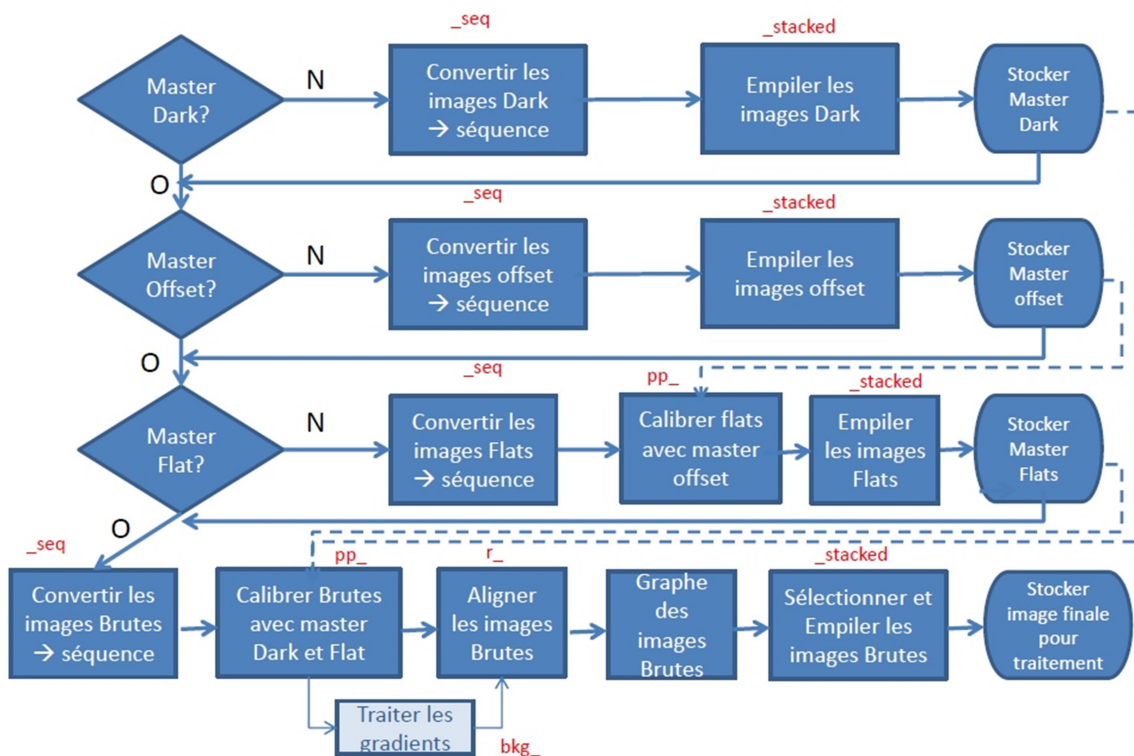
Par principe, les offsets sont contenus dans toutes les images que nous avons réalisées : les brutes bien sûr, mais aussi dans les darks et les flats. Il faudra donc enlever les offsets des darks et des flats pour utiliser ces images-outils. Mais analysons cette formule.

$$\frac{\text{image} - \text{dark} - \text{offset}}{\text{flat}} = \frac{\text{image} - (\text{dark} + \text{offset})}{\text{flat}}$$

Nous constatons qu'il suffit de soustraire les darks **NON TRAITÉS** (qui contiennent donc également les offsets) pour retirer en une fois le dark **ET** les offsets : gain de travail et de temps !

Philippe a un excellent transparent qui décrit les différentes étapes pour réaliser le traitement manuel. Les scripts réalisent le même flow de façon automatique.

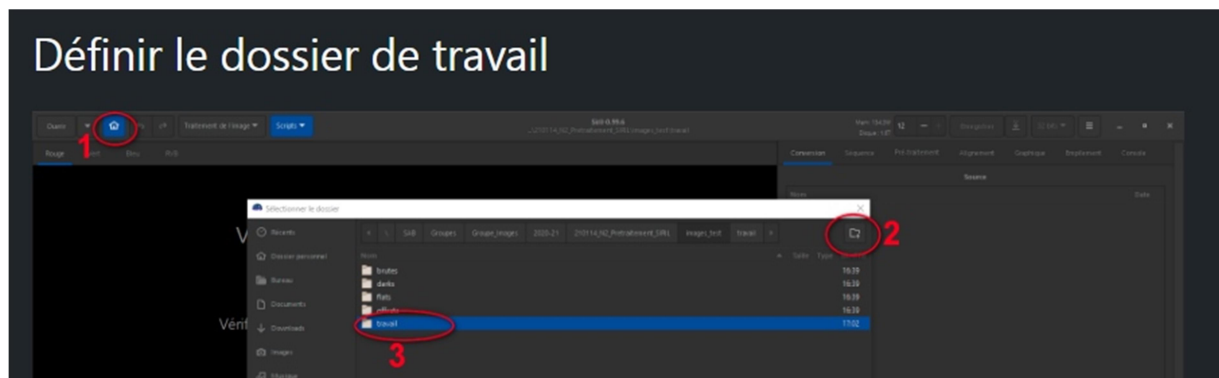
## Séquence pour traiter les images



Pour pré-traiter les images, le mode opératoire proposé est le suivant :

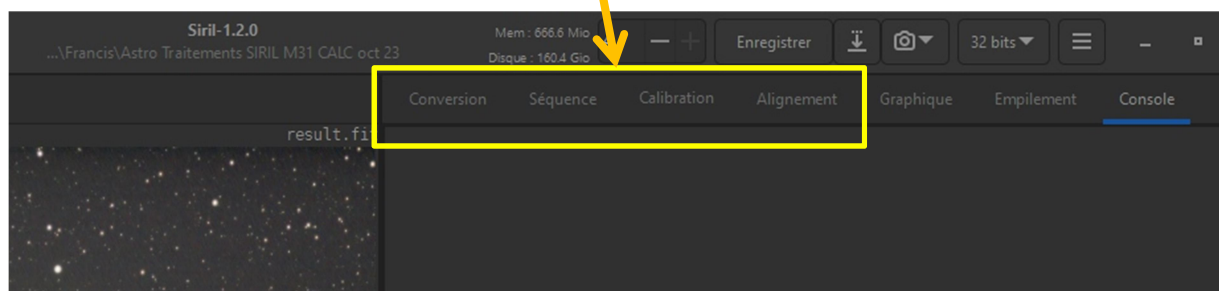
- 1) Définir le dossier de travail.
- 2) Convertir les fichiers.
- 3) Créer l'offset (ou biais) maître.
- 4) Créer le flat maître, en soustrayant l'offset maître.
- 5) Créer le dark maître, sans soustraire l'offset maître.
- 6) Calibrer les images, avec flat et dark maîtres.
- 7) Traiter les gradients.
- 8) Aligner les images.
- 9) Empiler les images.

On va d'abord créer un répertoire de travail, par exemple : process\_manuel et on le définit comme répertoire de travail (la petite maison bleue).




Le pré-traitement va être réalisé en plusieurs étapes dans les 4 onglets :

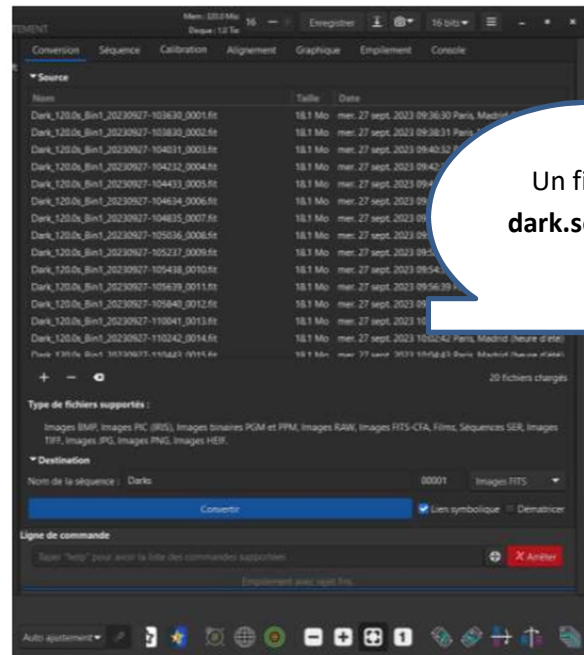
- Conversion et création d'une séquence
- Sélection de la séquence
- Calibration de la séquence
- Alignement des images light calibré



## 2) Réalisation du master dark

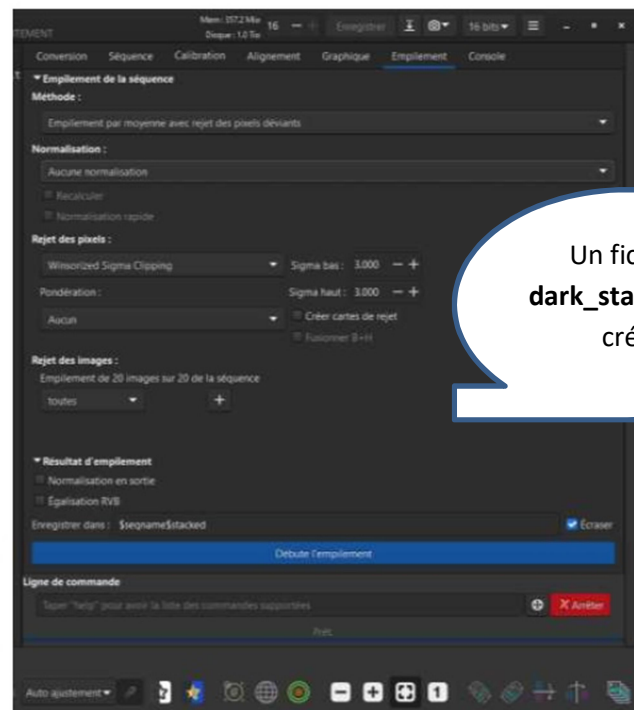
# Générer séquence Darks

- Aller sur l'onglet conversion
- Vider une liste pré-existante le cas échéant en cliquant sur 
- Aller chercher les fichiers darks dans le bon répertoire avec +
- **Donner un nom à la séquence**  
**Attention : le nom de la précédente séquence n'est généralement pas effacé = risque d'écrasement!!!**
- Pas de dématricage des images couleurs
- Le lien symbolique coché accélère le traitement (si on a donné les droits en mode développeur)




# Empiler les darks pour créer le Master\_Dark

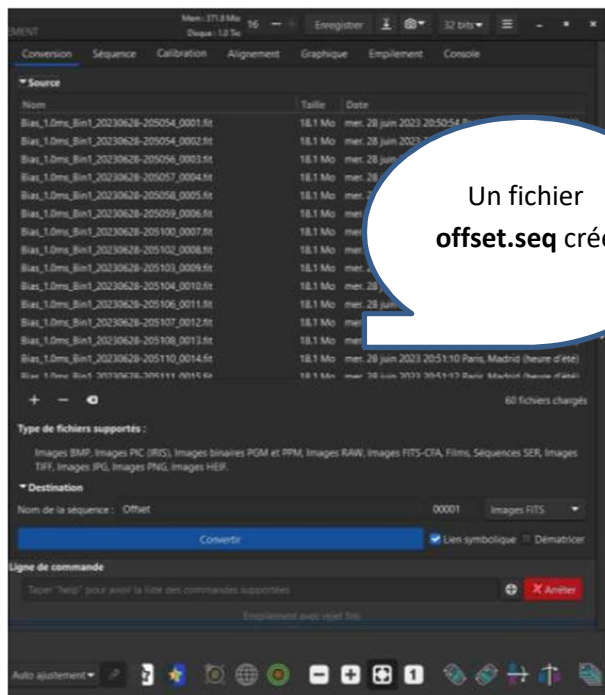
- Aller sur l'onglet empilement
- Méthode : empilement par moyenne avec rejet des pixels déviants
- **Aucune normalisation**
- Winsorized Sigma Clipping
- Aucune pondération
- -> création d'un fichier *nomseq\_stacked.fit*



### 3) Réalisation du master Offset (biases=offset)

## Générer séquence Offset (ou biases)

- Aller sur l'onglet conversion
- Vider une liste pré-existante le cas échéant en cliquant sur 
- Aller chercher les fichiers offsets dans le bon répertoire avec +
- **Donner un nom à la séquence**  
**Attention : le nom de la précédente séquence n'est généralement pas effacé = risque d'écrasement!!!**
- Pas de dématricage des images couleurs
- Le lien symbolique coché accélère le traitement (si on a donné les droits en mode développeur)

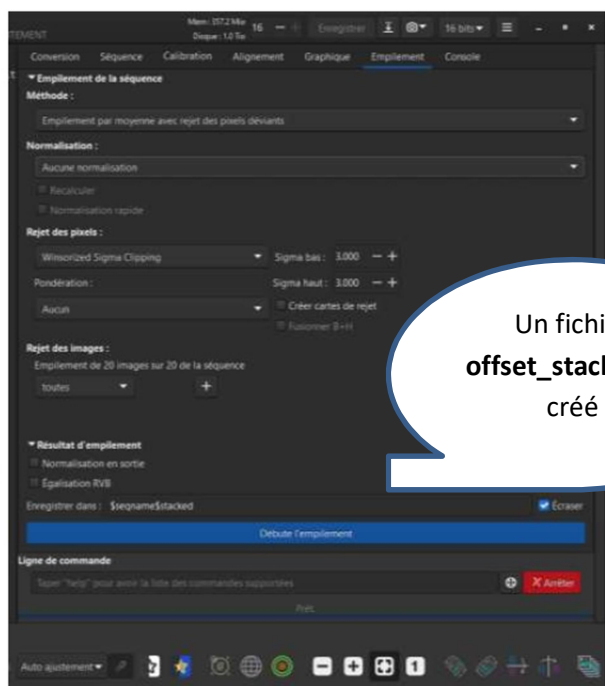


## Empiler les offsets pour créer le Master\_Offset

- Aller sur l'onglet empilement
- Méthode : empilement par moyenne avec rejet des pixels déviants
- **Aucune normalisation**
- Winsorized Sigma Clipping
- Aucune pondération
- -> création d'un fichier *nomseq\_stacked.fit*
- Option : Clic droit statistiques sur l'image du master\_offset noter la valeur moyenne pour la création d'un offset synthétique (cf plus loin).


Ex : 2797

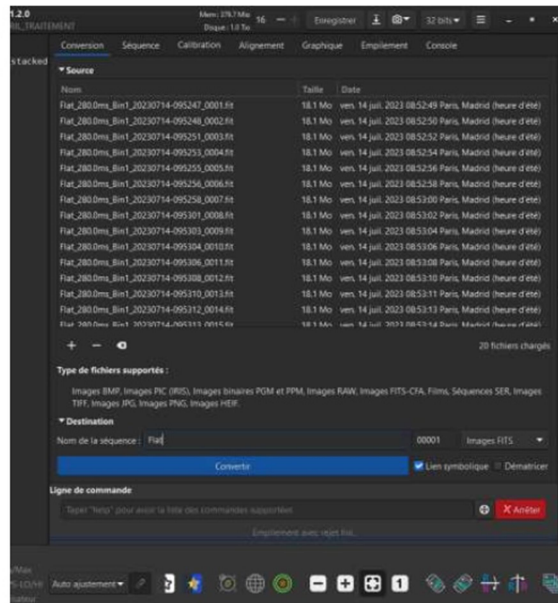
Offset_stacked			
Réels normalisés [0, 1]			
Taille de la sélection en pixel : (1846, 1724)			
	Rouge	Vert	Bleu
mean	2797.0	2797.2	2797.2
médiane	2797.0	2797.2	2797.2
sigma	0.9	0.9	0.9
avgDev	0.7	0.7	0.7
MAD	0.5	0.5	0.5
sqrt(BWNV)	0.8	0.8	0.8
min	2777.1	2744.6	2780.9
max	2814.2	2817.1	2815.9



#### 4) Réalisation du master flat

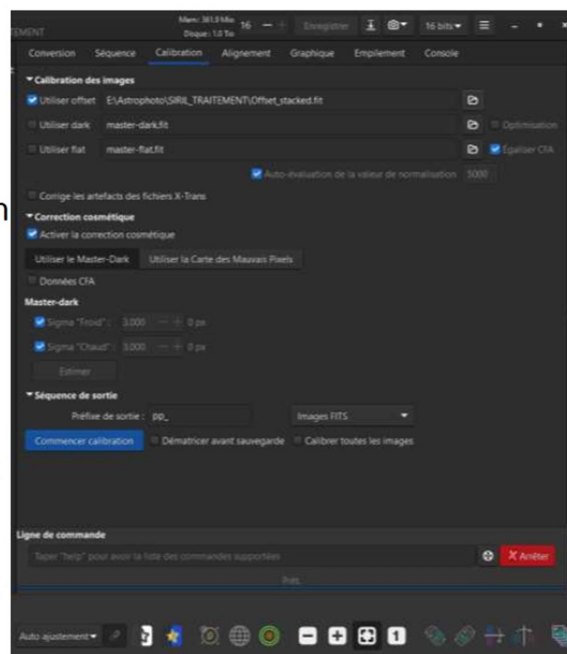
## Générer séquence Flats

- Vider une liste pré-existante le cas échéant en cliquant sur 
- Aller chercher les fichiers flat dans le bon répertoire avec +
- Donner un nom à la séquence  
Attention : le nom de la précédente séquence n'est généralement pas effacé = risque d'écrasement!!!
- **Pas de dématricage des images couleurs**
- Le lien symbolique coché accélère le traitement (si on a donné les droits en mode développeur)



## Calibration des flats

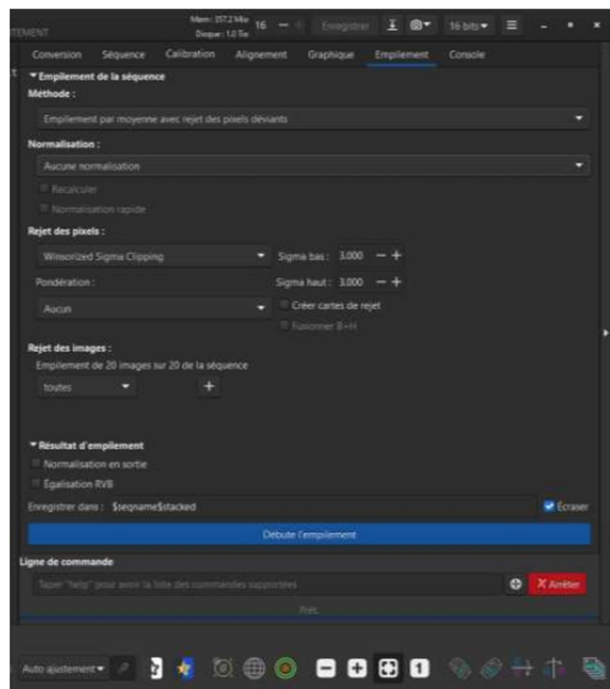
- Aller sur l'onglet calibration
- Utiliser le master offset qui vient d'être créé  
NB : on peut remplacer le lien vers ce fichier par l'offset synthétique en mettant = 2797 (valeur notée avant sur le master offset)
- La calibration va créer des images avec le préfixe pp\_



*Note : « REMARQUE: Nous recommandons de cocher le bouton "Egaliser CFA" pour les images couleurs afin de préserver l'équilibre des canaux original. » (source : <https://siril.org/fr/tutorials/tuto-manual/>)*


# Empiler les flats pour créer le Master\_Flat

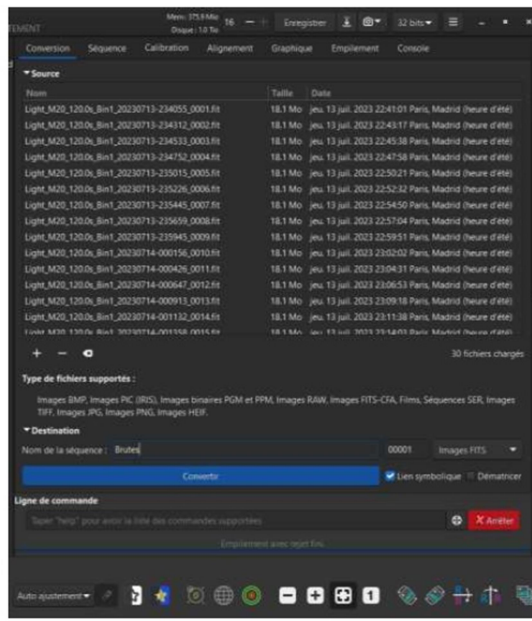
- Aller sur l'onglet empilement
- Méthode : empilement par moyenne avec rejet des pixels déviants
- **Normalisation multiplicative**
- Winsorized Sigma Clipping
- Aucune pondération
- -> création d'un fichier *nomseq\_stacked.fit*
- Vérifier dans la console que le % de rejet des pixels ne dépasse pas 1% (idéalement 0.1-0.5%) sinon augmenter la valeur des sigmas bas/haut



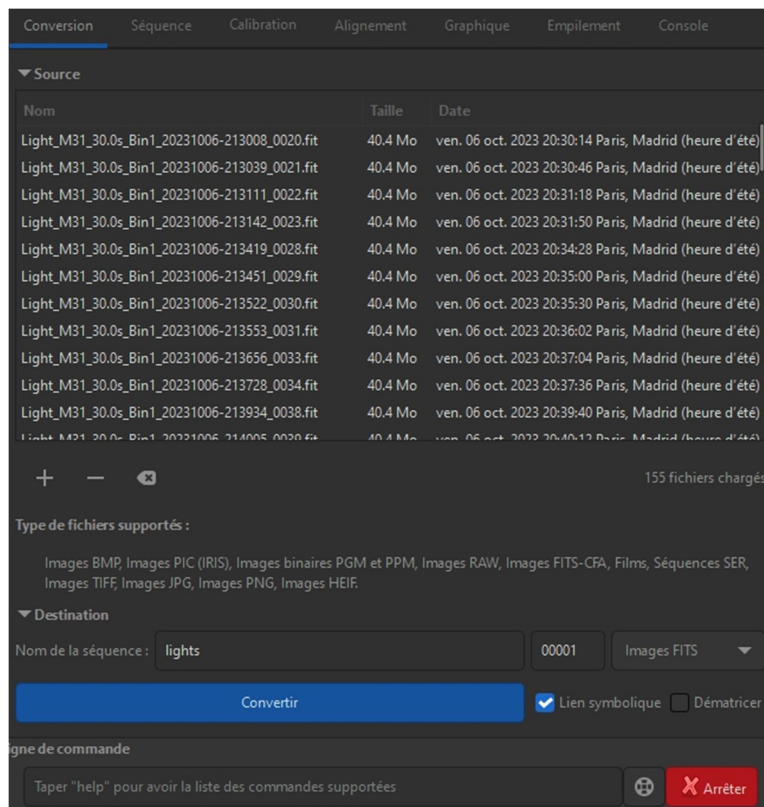
## 5) Pré-traitement des lights (brutes)

### 5a) *génération de la séquence des brutes (lights)*

- Aller sur l'onglet conversion
- Vider une liste pré-existante le cas échéant en cliquant sur 
- Aller chercher les fichiers bruts dans le bon répertoire avec +
- **Donner un nom à la séquence**  
**Attention : le nom de la précédente séquence n'est généralement pas effacé = risque d'écrasement!!!**
- Pas de dématricage des images couleurs
- Le lien symbolique coché accélère le traitement (si on a donné les droits en mode développeur)

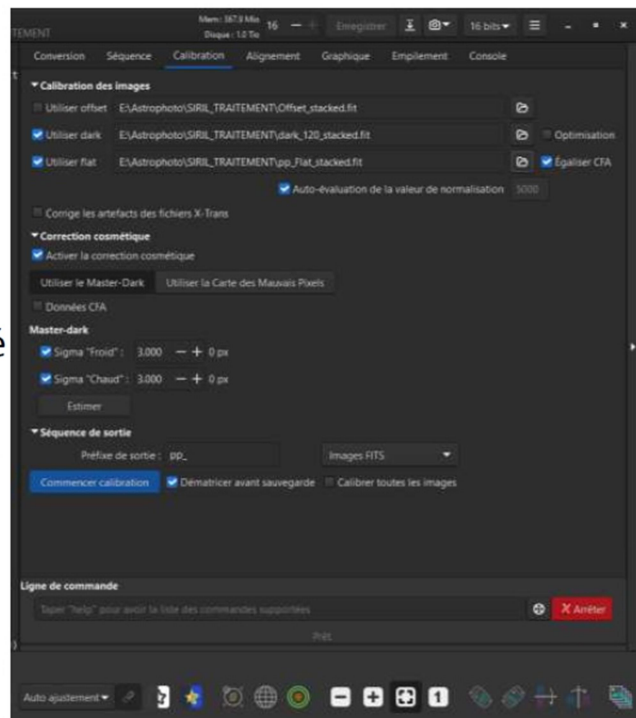


Pour notre image de M31 :



## 5b) calibration des brutes

- Aller sur l'onglet calibration
- On n'utilise pas de master offset ici
- Utiliser le master dark et le master flat qui viennent d'être créés  
NB : le master flat a un préfixe pp\_ car il a été calibré par le master offset
- **Cocher dématricer avant sauvegarde pour les images couleur !!!**
- La calibration va créer des images avec le préfixe pp\_

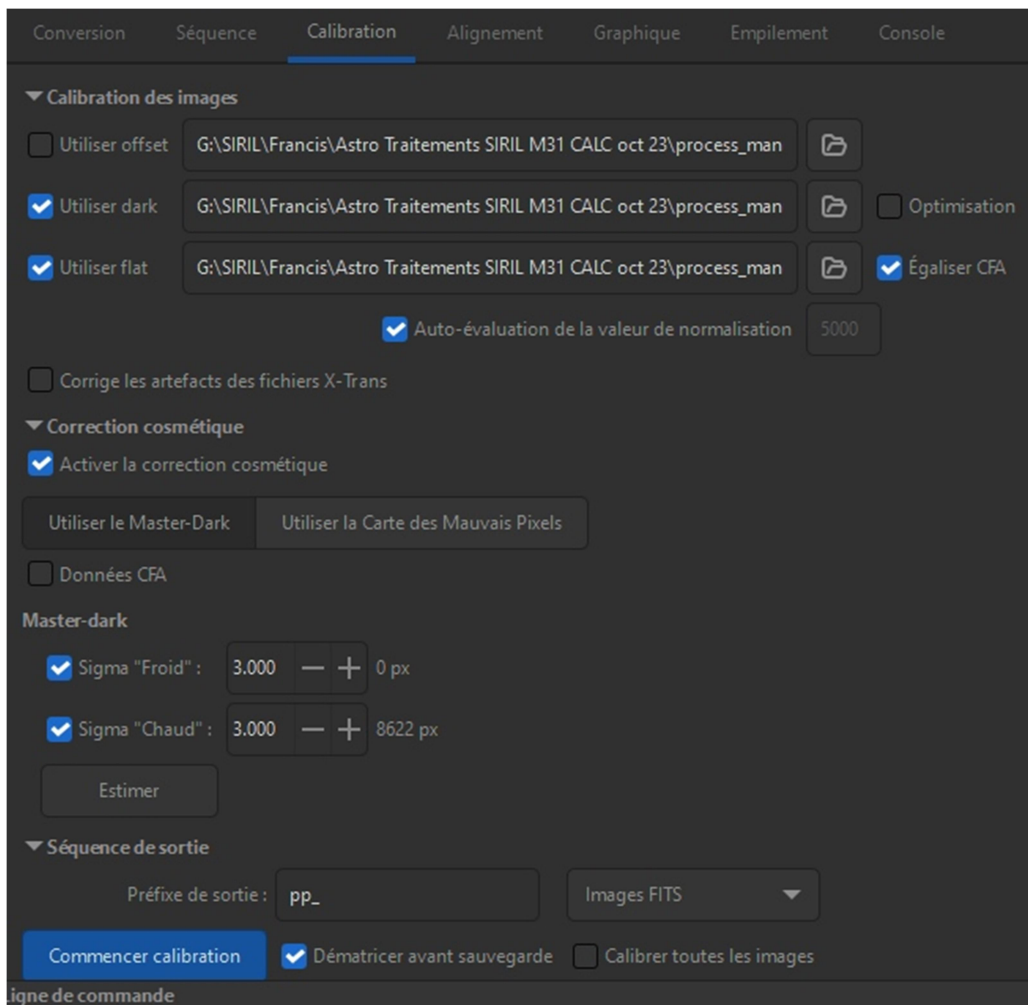


Prétraitement des LIGHT avec les Masters DARK et FLAT, alignement puis empilement.

- Ecran Conversion
  - Idem que pour les autres après avoir trié préalablement les brutes
  - Donner un nom de séquence : **brute\_<objet>\_<date>\_F<focale>\_S<durée>\_G<gain>**
- Ecran Calibration
  - **Calibration des images**
    - On n'utilise pas les offsets, car ils sont déjà dans les DARK
    - Dans « Utiliser dark », charger le master DARK : **master\_dark\_S<durée>\_G<gain>.fit**
    - Dans « Utiliser flat », charger le master FLAT : **master\_flat\_<XXX>\_F<focale>\_G<gain>.fit**
    - « Egaliser CFA » coché (rend le flat neutre)
  - **Correction cosmétique**
    - Cocher : Activer la correction cosmétique
    - Utiliser le Master-Dark
    - Cocher Données CFA
  - Master-dark  
Cliquer sur « Estimer » et modifier les limites si en rouge
  - Cocher Dématricer avant sauvegarde (sinon images en N&B)
  - Cliquer sur **Commencer calibration**  
=> Résultat : **pp\_brute\_<objet>\_<date>\_F<focale>\_S<durée>\_G<gain>**

Source : Web, Process\_SIRIL.pdf

## Pour notre image M31, configuration

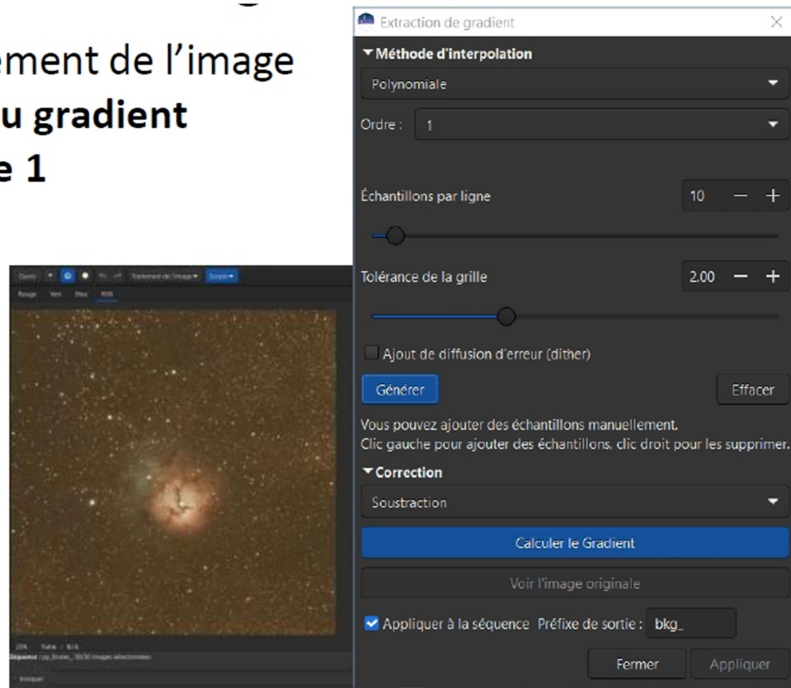


### 5c) Option : extraction du gradient

L'outil a été mis à jour dans la version 1.2. Pour en savoir plus sur le retrait du gradient :

<https://siril.org/fr/tutorials/gradient/>

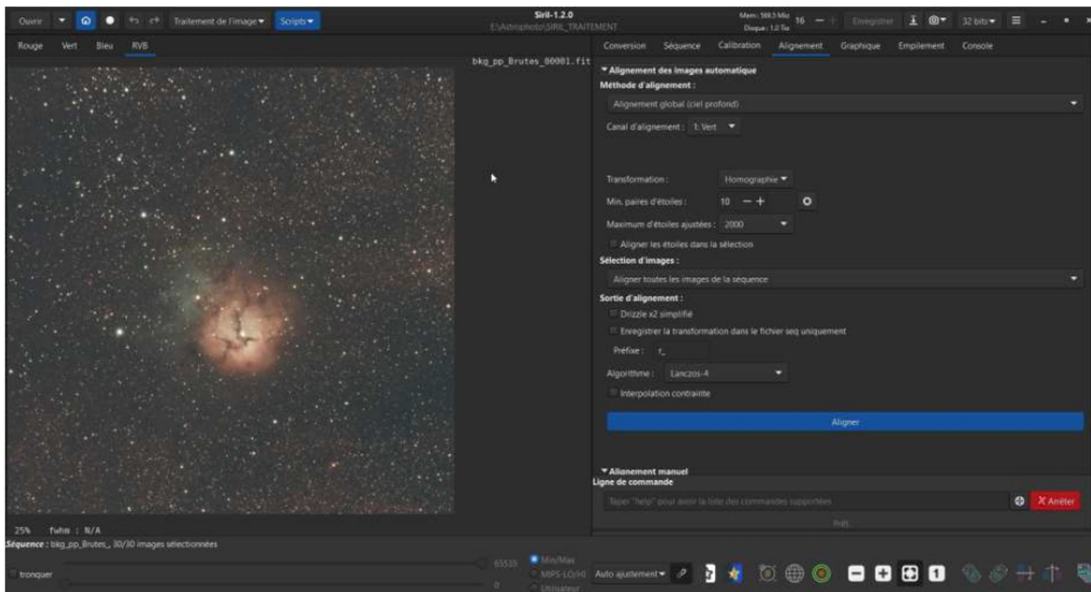
- Cliquer onglet traitement de l'image
- Choisir extraction du gradient
- Polynomiale d'ordre 1
- Générer la grille
- Appliquer à toute la série
- Calculer le gradient
- Et appliquer
- Crée les fichiers préfixe bkg\_



- Extraction du gradient
  - Possibilité d'extraire le gradient  
Ici, on veut supprimer le gradient sur les images individuelles (sur toute la séquence)  
Appliquer la méthode Polynomiale d'ordre 1
  - Correction = Soustraction (on utilisera Division si on veut compenser un problème de FLAT)  
Pas nécessaire de cliquer sur calculer le gradient car on va le calculer pour chaque image  
Générer seulement une dizaine d'échantillons (peut être fait manuellement)
  - Cliquer sur **Appliquer**  
=> Résultat : **bkg\_pp\_brute\_<objet>\_<date>\_F<focale>\_S<durée>\_G<gain>**

## 5d) Alignement des images

- Aller sur l'onglet alignement et aligner ciel profond

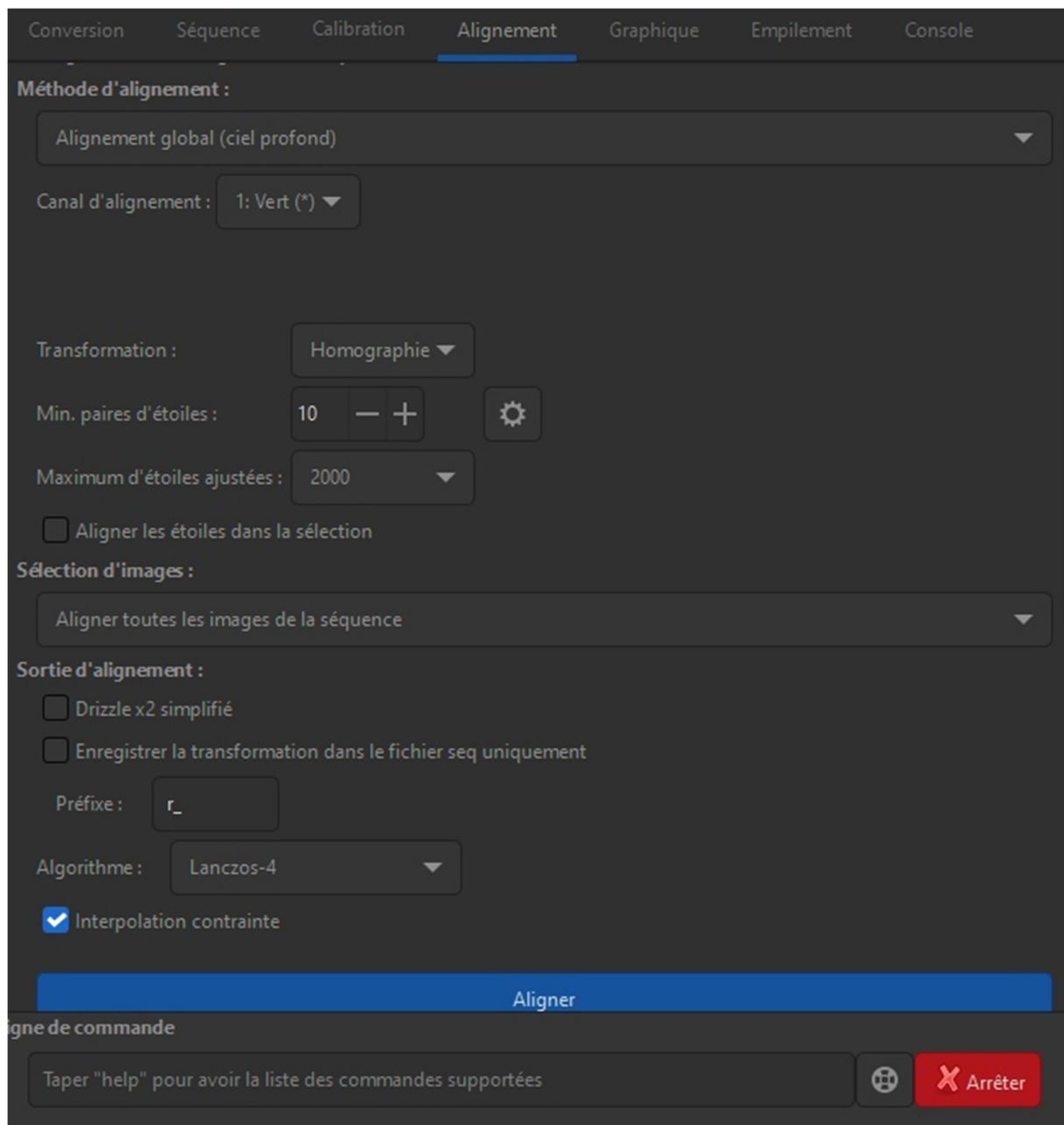


## Les différentes méthodes d'alignement en fonction du type d'objet

- Ciel profond
  - Alignement sur 1-2-3 étoiles
  - Alignement global
  - Alignement des étoiles en deux passes
- Surface planétaire
  - Alignement par motif d'image (disque entier)
  - Alignement Kombat (surface planétaire ou disque entier)
- Alignement comète ou astéroïde
- Ne pas réaligner
  - Applique alignement existant

- Ecran Alignement
  - Charger la séquence (bkg\_pp\_brute\_<objet>\_<date>\_F<focale>\_S<durée>\_G<gain>seq)
  - **Méthode d'alignement**
    - Alignement global (Ciel Profond)
    - Choisir le canal vert
    - Transformation : Homographie
  - **Sélection d'images**
    - Cliquer sur la liste des images et choisir comme image de référence la meilleure en FWHM
  - **Sortie d'alignement**
    - Cocher Drizzle \*2 si sous-échantillonnage : FWHM < 2 pixels
    - Algorithme : Bicubique
  - Cliquer sur **Aligner**  
=> Résultat : **r\_bkg\_pp\_brute\_<date>\_F<focale>\_S<durée>\_G<gain>**

Pour notre image M31, configuration



## 5e) Graphique pour trier les images

Puis on vient trier les images dans l'onglet Graphique. Il s'agit d'éliminer les images avec des FWHM trop hautes

### Les graphiques pour trier les images

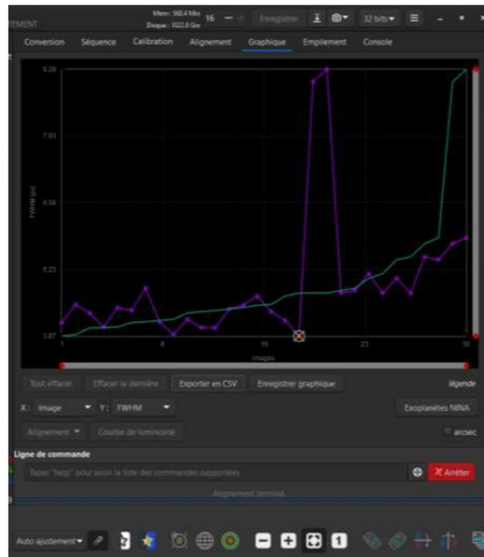
Après l'alignement on va sur l'onglet graphique pour analyser les images

on peut faire des graphiques X – Y avec les différentes variables en particulier

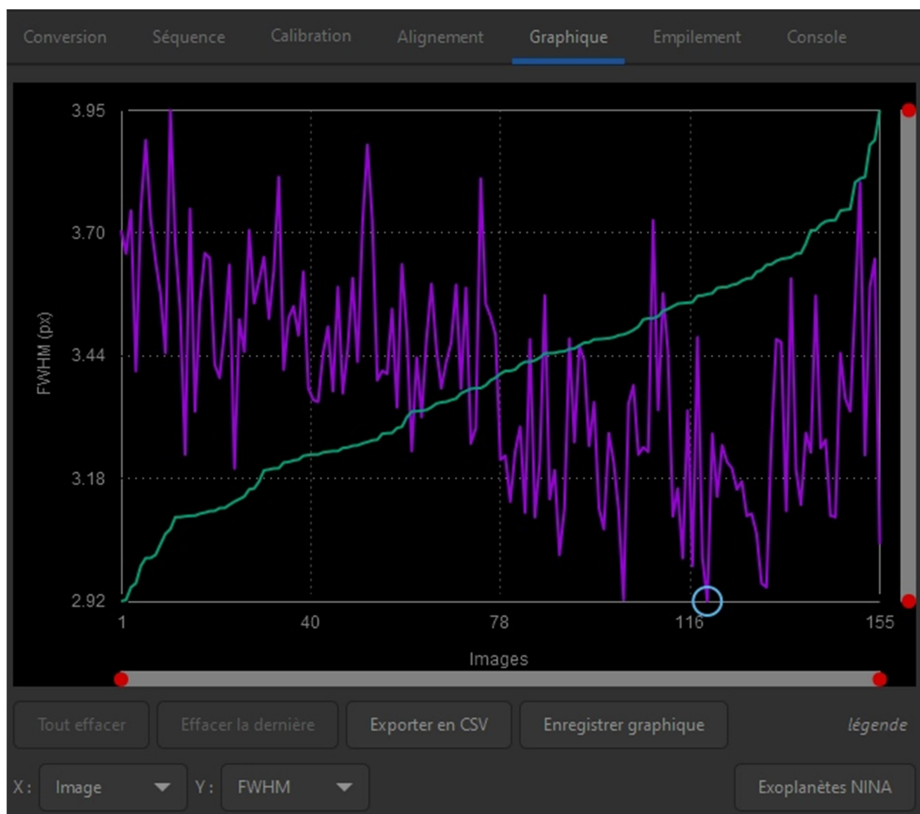
- Image (le N° de l'image)
- Fond de ciel
- FWHM (Taille des étoiles)
- wFWHM (FWHM pondérée)
- Rondeur des étoiles

On va éliminer les images de mauvaise Qualité avant de faire l'empilement

- Clic droit sur les points
- Encadrer la zone de points



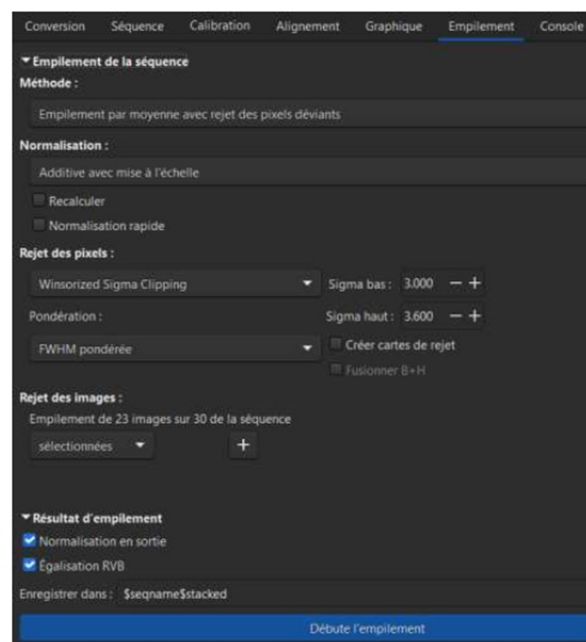
Pour notre image de M31



## 5f) empilement des images

Puis on passe à l'étape finale de l'empilement des images calibrées, alignées et triées

- Aller sur l'onglet empilement
- Méthode : empilement par moyenne avec rejet des pixels déviants
- **Normalisation additive avec mise à l'échelle**
- Winsorized Sigma Clipping
- On peut faire une pondération, par exemple sur la taille des étoiles
- On peut cocher normalisation en sortie et Egalisation RVB pour les images couleur (meilleur équilibre des couleurs)
- -> création d'un fichier nomseq\_stacked.fit
- Vérifier dans la console que le % de rejet des pixels ne dépasse pas 1% (idéalement 0.1-0.5%) sinon augmenter la valeur des sigmas bas/haut



- Ecran Empilement
  - **Empilement de la séquence**
    - Par moyenne avec rejet des pixels déviants
  - **Normalisation**
    - Additive avec mise à l'échelle (image étalée sur toute la plage dynamique)
  - **Rejet des pixels**
    - Méthode de rejet
      - Si < 30 images : Winsorized Sigma Clipping
      - Si 30 <> 90 images : Linear Fit Clipping
      - Si > 90 images : Test de déviation extrême
    - Cocher éventuellement « Pondération » (met un poids statistique sur chaque image – les meilleures en le plus)
  - **Rejet des images**
    - Choisir Sélectionnées
    - Autres critères surtout pour le *Lucky imaging*
  - **Résultat d'empilement**
  - Cocher « Normalisation en sortie »
  - Cocher Egalisation RVB  
*Fournie une image équilibrée par rapport à la dominante verte causée par le filtre de Bayer*
  - Cliquer sur **Débuter l'empilement**  
=> Résultat : **r\_bkg\_pp\_brute\_<date>\_F<focale>\_S<durée>\_G<gain> stacked.fit**

*Note : « L'option Egalisation RVB vous permettra d'avoir une image déjà équilibrée en sortie d'empilement. Cela évite la dominante verte bien connue causée par le filtre de Bayer en imagerie couleur. » (source : <https://siril.org/fr/tutorials/tuto-manual/>)*

M31, Configuration d'empilement :

Conversion   Séquence   Calibration   Alignement   Graphique   **Empilement**   Console

**Méthode :**

Empilement par moyenne avec rejet des pixels déviants ▼

**Normalisation :**

Additive avec mise à l'échelle ▼

Recalculer

Normalisation rapide

**Rejet des pixels :**

Test de déviation extrême généralisé de Student ▼   ESD taux : 0.300 — +

Pondération :   ESD Importance : 0.050 — +

FWHM pondérée ▼    Créer cartes de rejet

Fusionner B+H

**Rejet des images :**

Empilement de 155 images sur 155 de la séquence

sélectionnées ▼   +

▼ **Résultat d'empilement**

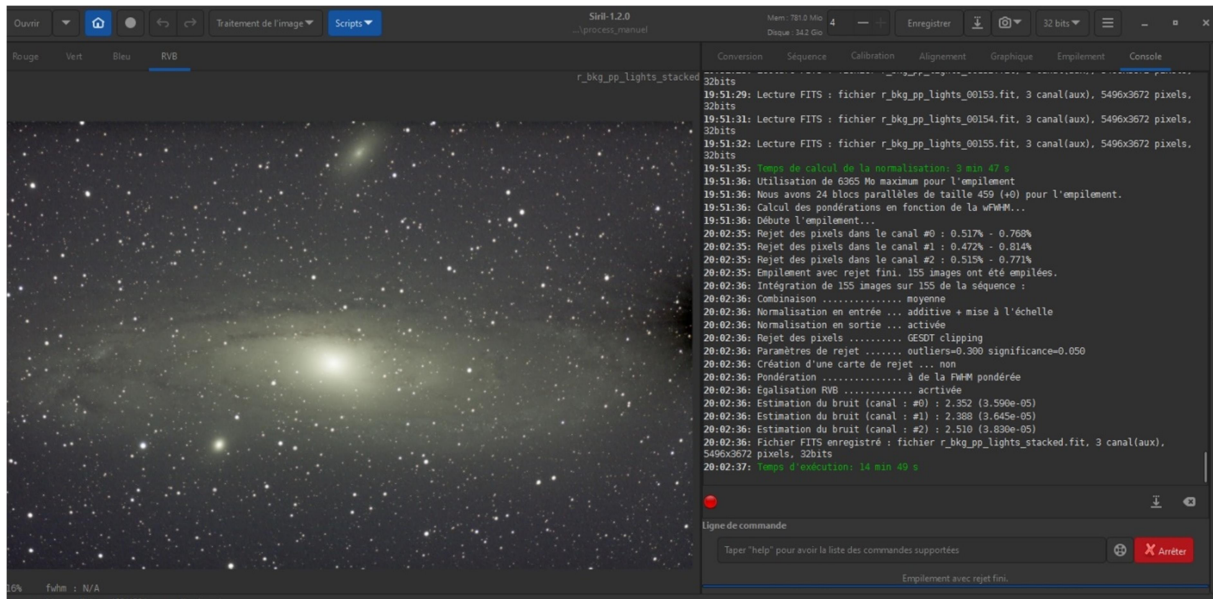
Normalisation en sortie

Égalisation RVB

Enregistrer dans : Sseqname\$stacked    Écraser

## 5g) image finale

On obtient le fichier final obtenu avec le pré-traitement manuel : fichier r\_bkg\_pp\_lights\_stacked.fit



Il faut vérifier le log de la console, et notamment le rejet des pixels qui doit rester en dessous de %.

```
32bits
19:51:35: Temps de calcul de la normalisation: 3 min 47 s
19:51:36: Utilisation de 6365 Mo maximum pour l'empilement
19:51:36: Nous avons 24 blocs parallèles de taille 459 (+0) pour l'empilement.
19:51:36: Calcul des pondérations en fonction de la wFWHM...
19:51:36: Début de l'empilement...
20:02:35: Rejet des pixels dans le canal #0 : 0.517% - 0.768%
20:02:35: Rejet des pixels dans le canal #1 : 0.472% - 0.814%
20:02:35: Rejet des pixels dans le canal #2 : 0.515% - 0.771%
20:02:35: Empilement avec rejet fini. 155 images ont été empilées.
20:02:36: Intégration de 155 images sur 155 de la séquence :
20:02:36: Combinaison ..... moyenne
20:02:36: Normalisation en entrée ... additive + mise à l'échelle
20:02:36: Normalisation en sortie ... activée
20:02:36: Rejet des pixels ..... GESDT clipping
20:02:36: Paramètres de rejet ..... outliers=0.300 significance=0.050
20:02:36: Création d'une carte de rejet ... non
20:02:36: Pondération ..... à de la FWHM pondérée
20:02:36: Égalisation RVB ..... activée
20:02:36: Estimation du bruit (canal : #0) : 2.352 (3.590e-05)
20:02:36: Estimation du bruit (canal : #1) : 2.388 (3.645e-05)
20:02:36: Estimation du bruit (canal : #2) : 2.510 (3.830e-05)
20:02:36: Fichier FITS enregistré : fichier r_bkg_pp_lights_stacked.fit, 3 canal(aux),
5496x3672 pixels, 32bits
20:02:37: Temps d'exécution: 14 min 49 s
```

Note : « Siril empile les images choisies, et crée le fichier dont on a défini le nom. Comme précédemment, vérifier que le taux de rejet se trouve dans une fourchette 0.1 à 0.5 %. Si les valeurs sont aberrantes, reprendre l'empilement en changeant les valeurs des Sigmas bas/haut. Quand on augmente un facteur sigma, on abaisse le taux de rejet ! » (source : <https://siril.org/fr/tutorials/tuto-manual/>)

## C) SIRILIC : Pre-traitement avec interface graphique

On peut récupérer des scripts existants et les modifier pour les adapter à ses besoins. Autant sous linux, c'est facile, autant sous Windows, c'est très vite la galère.

Heureusement, un outil avec interface graphique a été développé par Colmic : SIRILIC

Il permet de créer un script adapté à ses images (et à son setup). Une fois le script optimisée pour son setup, on le range et on le réutilise les fois suivantes.

### Qu'est ce que Sirilic ?

Il s'agit d'un outil logiciel graphique interactif permettant d'automatiser le prétraitement des images astronomiques à l'aide du logiciel **Siril**. Il crée les fichiers maîtres (**Offset, Flat, Dark** et **Dark-Flat**) associés à une séquence d'images, calibre chaque image brute, les aligne et les empile. L'amélioration par rapport à **Siril** est qu'il ne nécessite pas un ensemble prédéfini de fichiers d'entrée, et prend également en charge le traitement de plusieurs sessions ou les acquisitions de filtres. Il affiche graphiquement le processus actuel pour chaque couche.

### Pourquoi Sirilic ?

Au départ, **Sirilic** a été créé pour construire facilement des scripts personnalisés en quelques clics et quelles que soient les sources d'images (*APN, CCD mono, etc ...*) et les options choisies. Pour aider les débutants, un affichage graphique a été ajouté. Il montre les opérations de script basées sur les options cochées.

### Que fait Sirilic exactement ?

Il fait principalement 4 choses :

- Organiser le répertoire de travail **Siril** en le structurant en sous-dossiers
- Copier ou lier les fichiers **Brute, Offset, Dark** ou **Flat** dans les sous-dossiers
- Générer un script **Siril** adapté aux fichiers présents et aux options choisies
- Lancer les traitements sur les images: **Siril** est lancé en mode script en tâche de fond

Il traite indifféremment les images issues d'APN, de camera CCD / CMOS couleur et monochrome.

Source : <https://siril.org/fr/docs/sirilic/>

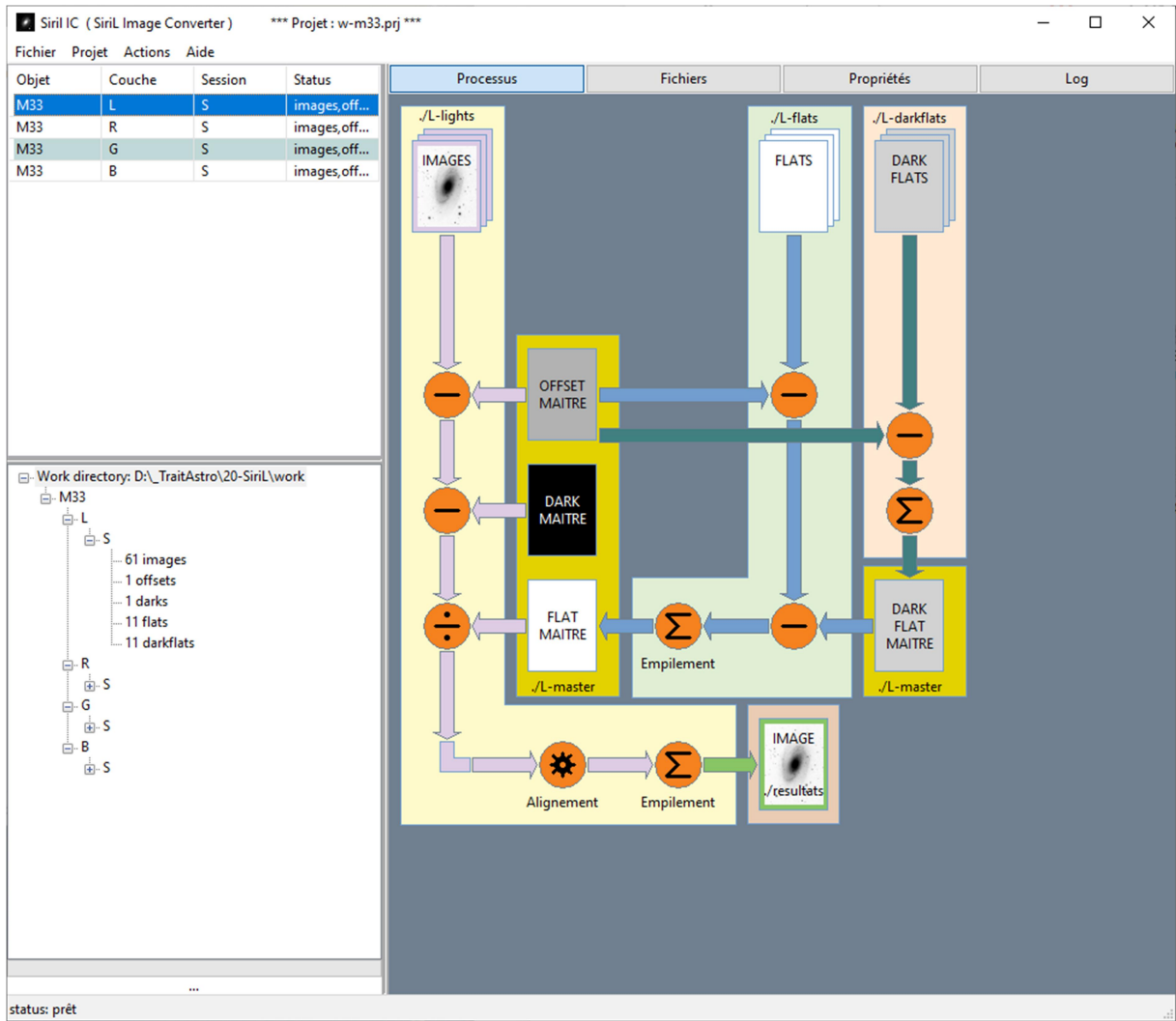
Autres liens utiles :

<https://astroslachonet.wordpress.com/2023/07/31/utilitaire-conv2siril/>

<https://www.webastro.net/forums/forum/150-logiciel-sirilic/>

Vidéos :

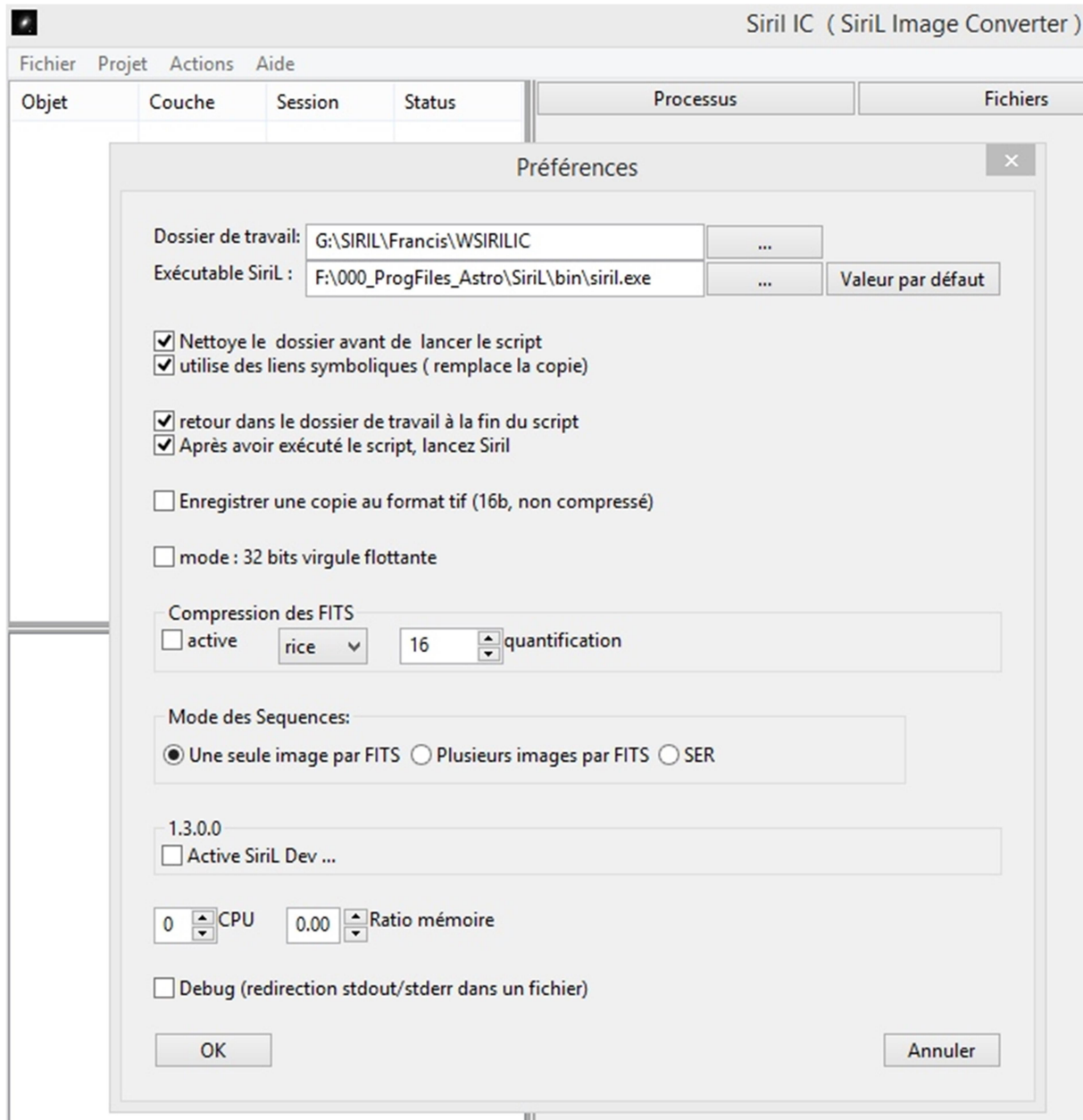
tutos : <https://www.youtube.com/watch?v=XnFcl7Lt6nY>



## 1) Avant-propos

Avant de commencer à travailler, il faut créer un répertoire de travail pour notre projet avec Sirilic.

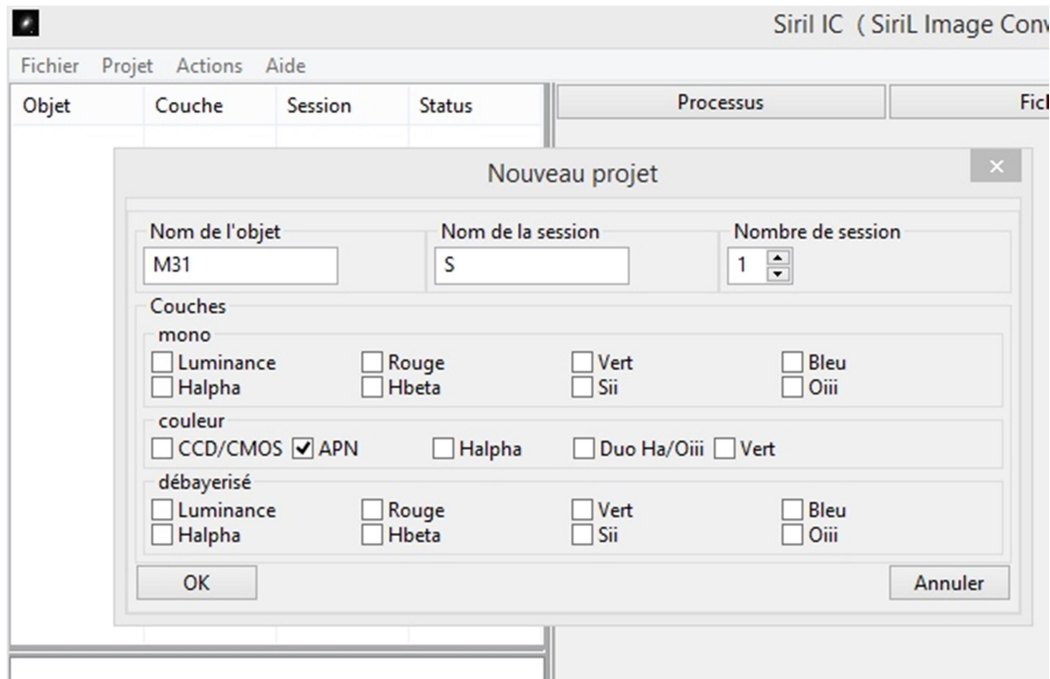
Il faut ensuite indiquer le chemin de ce répertoire et celui de l'exécutable de SIRIL dans les préférences (ouvrir Sirilic, Fichier > Préférences) . On peut se référer à la page de Sirilic (<https://siril.org/fr/docs/sirilic/>)



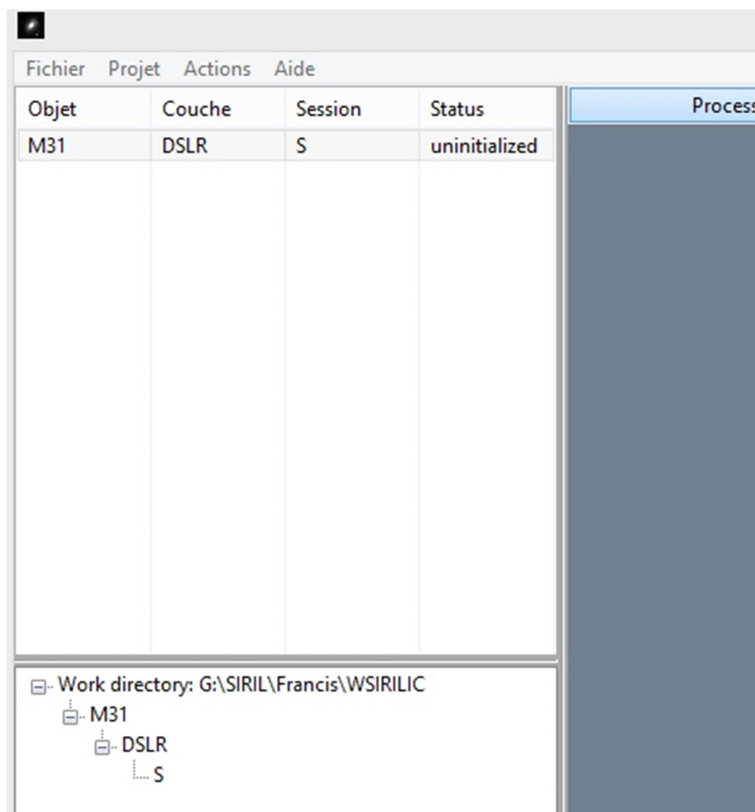
## 2) Création d'un projet

Pour créer un projet : Fichier > Nouveau.

Prendre le temps de bien renseigner les différentes couches et validez.



Le projet est créé, une arborescence apparait à gauche

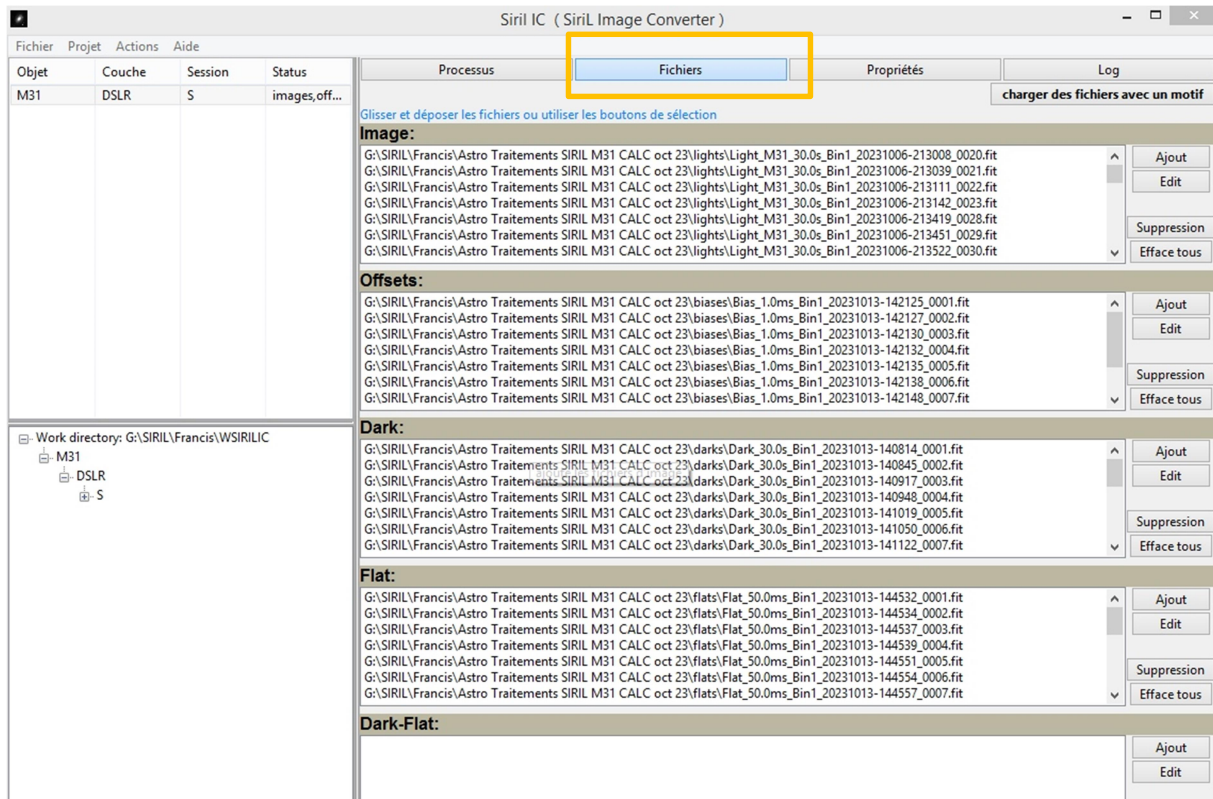


Nous allons pouvoir ajouter nos images et les DOFs à notre projet

### 3) Ajout des images et DOFs pour chaque couche

Sélectionner la ligne nommée M31, puis cliquer sur l'onglet Fichiers dans la zone à droite.

Pour notre image de M31 :



### 4) Configuration de chaque couche du projet

L'étape suivante consiste à définir les diverses options et paramètres pour chaque type d'image. Il faut aller sur l'onglet Propriétés. Ce qui est intéressant ici, c'est quand fonction des choix, le synoptique dans l'onglet « Processus » évolue. De ce fait, on peut rapidement se rendre compte si l'on a fait une erreur (par exemple une double soustraction de l'offset).

On fera comme dans la partie manuelle, on fera la soustraction de l'offset au niveau des flats.

Note importante : si l'on veut obtenir le même résultat qu'en mode manuel, il faut avoir les paramètres définis de la même manière. Il ne faut donc vérifier chaque paramètre.

IMAGE	OFFSETS	DARK	FLAT	DARK FLAT
-------	---------	------	------	-----------

propriétés identiques pour toutes les couches

### Prétraitement

Soustrait l'offset  correction Fuji X

Correction cosmétique

sigma chaud: 3.0  sigma froid: 3.0  Bad Pixel Map:  ...

### Empilement

Egalisation RVB  Optim.Noir

Empilement	Type de réjection	Weighting:	Normalisation
mean <input type="text"/>	Winsorized <input type="text"/>	Haut: 3.000 <input type="text"/> Bas: 3.000 <input type="text"/>	WFWHM <input type="text"/> addscale <input type="text"/>

Rejecting Filter

FWHM:  WFWHM:  Background:  Rondeur:  Qualité:  Nombre d'étoiles:

IMAGE	OFFSETS	DARK	FLAT	DARK FLAT
-------	---------	------	------	-----------

propriétés identiques pour toutes les couches

### Prétraitement

copie de la librairie par SirIL

### Empilement

Empilement	Type de réjection	Normalisation
mean <input type="text"/>	Winsorized <input type="text"/>	Haut: 3.000 <input type="text"/> Bas: 3.000 <input type="text"/> no <input type="text"/>

Processus	Fichiers	Propriétés	Log
-----------	----------	------------	-----

IMAGE	OFFSETS	DARK	FLAT	DARK FLAT
-------	---------	------	------	-----------

propriétés identiques pour toutes les couches

### Prétraitement

copie de la librairie par SirIL  Soustrait l'offset

### Empilement

Empilement	Type de réjection	Normalisation
mean <input type="text"/>	Winsorized <input type="text"/>	Haut: 3.000 <input type="text"/> Bas: 3.000 <input type="text"/> no <input type="text"/>

Processus	Fichiers	Propriétés	Log
-----------	----------	------------	-----

IMAGE	OFFSETS	DARK	FLAT	DARK FLAT
-------	---------	------	------	-----------

propriétés identiques pour toutes les couches

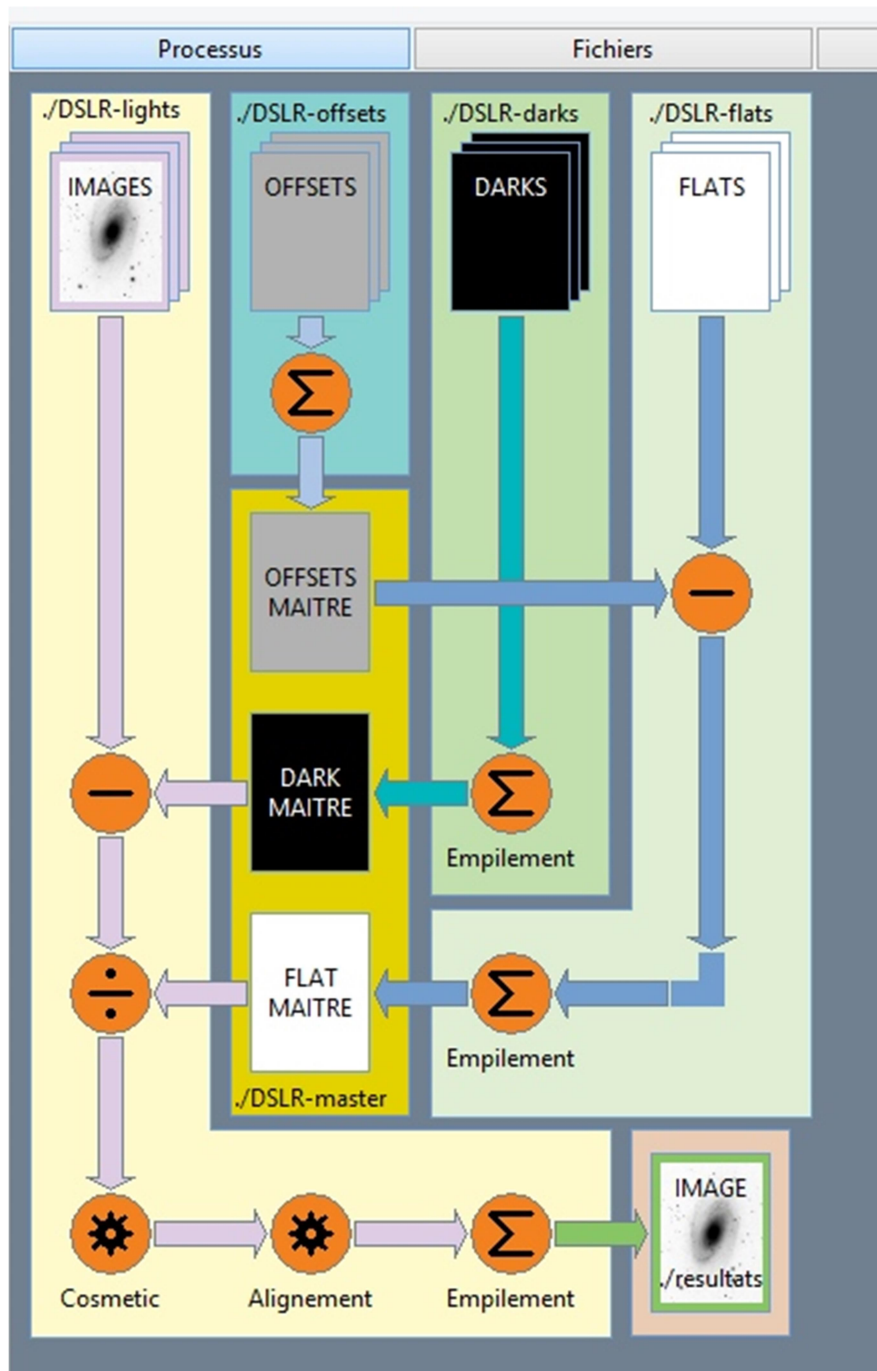
### Prétraitement

copie de la librairie par SirIL  Soustrait l'offset

### Empilement

Empilement	Type de réjection	Normalisation
mean <input type="text"/>	Winsorized <input type="text"/>	Haut: 3.000 <input type="text"/> Bas: 3.000 <input type="text"/> mul <input type="text"/>

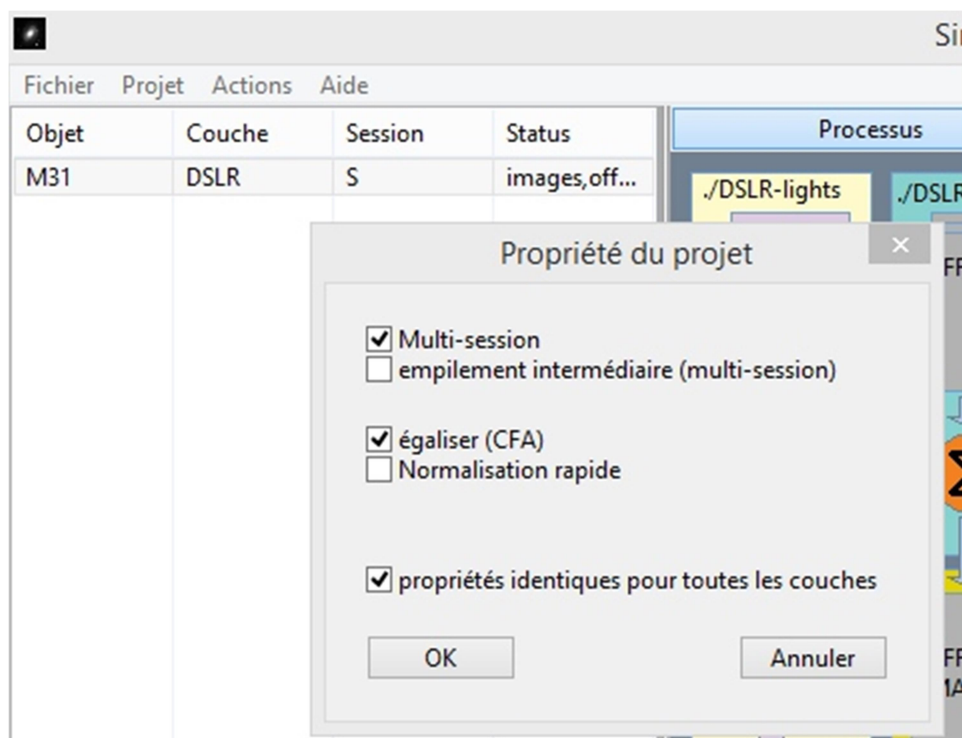
On pourra vérifier graphiquement dans l'onglet processus que le cheminement est logique. Cette représentation graphique permet de mieux comprendre ce qui se passe.



On va maintenant définir les propriétés globales du projet.

## 5) Paramètres globaux du projet

Aller dans le menu Projet > Edition des propriétés



Dans notre exemple (APN), le paramètre *égalise (CFA)* est coché.

Sirilic peut traiter des images du même objet prises sur plusieurs nuits dans les mêmes conditions : avec à peu près le même cadrage, les mêmes dimensions d'image, le même filtre utilisé le cas échéant, mais comment la caméra a été déplacée entre les deux nuits les flats doivent être refaits, éventuellement la température est différente et un ensemble de nouveaux darks doit également être fait ... L'option *Multi-session* n'a aucun effet sur le traitement lorsqu'il n'y a qu'une seule session. Le tableau suivant explique l'effet de la combinaison de *Multi-Session* et *Empilement intermédiaire*.

Multi-session	Empilement intermédiaire	Description
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Les couches avec plusieurs sessions sont regroupées, alignées et empilées mais pas d'empilement sur les sessions de cette couche
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dans les couches qui ont plusieurs sessions, Sirilic aligne, empile chaque session de cette couche et enregistre les résultats intermédiaires. Ensuite, il regroupe les sessions en une, puis aligne et empile
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Dans la couche à plusieurs session, Sirilic aligne, empile chaque session de cette couche et enregistre les résultats. En résumé, il ne regroupe pas les sessions

Avant de poursuivre, il est intéressant de sauvegarder le projet en avec *Fichier* et *Sauvegarder*.

## 6) Structuration des dossiers et copie des images

On va finaliser l'image du prétraitement en 2 étapes :

Etape 1 : structuration et mise en forme de la base de données

Etape 2 : Exécution du « script » (configuration que nous avons mise en place aux étapes précédentes) pour finaliser une image résultante de toutes les images prises.

Note : cette image est souvent peu flatteuse car les opérations de traitement n'ont pas été réalisées et aussi parce que l'image est étirée artificiellement.

Le processus est représenté par un diagramme à six étapes en forme de chevrons :

- Création du projet (Ajout de couches)
- Ajout des images, flats, darks de chaque couche
- configuration de chaque couches
- Paramètres globaux du projet
- Création des dossiers, Copie des images
- Création des scripts, Lancement des scripts

Le projet étant configuré, il faut maintenant :

- Construire l'arborescence des répertoires dans le dossier de travail
- Nettoyer les fichiers d'un précédent traitement avant la copie
- Copier les fichiers ou créer des liens sur les images

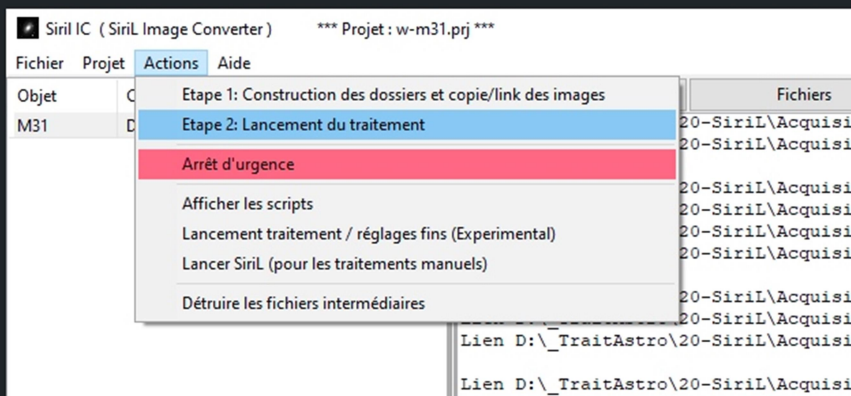
Cliquer sur l'étape 1 dans le menu **Actions**.

Construction des dossiers et copie des images

On peut observer le déroulement des opérations dans la fenêtre de log

Pour l'instant, nous avons préparé le « script » a exécuté et nous avons reformaté les données pour réaliser la dernière étape : exécuter notre « script » et sortir une image finale de pré-traitement.

C'est l'étape finale : il faut cliquer sur l'étape 2 du menu **Actions**.

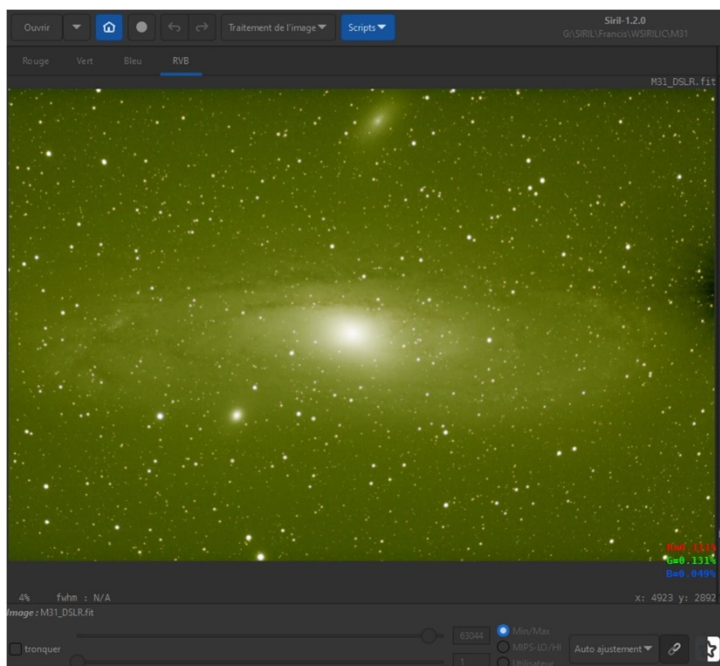


Exécution du script *Siril*

Le déroulement du traitement des images par Siril s'affiche dans la fenêtre de log. Il se fait en plusieurs étapes :

- Création d'un script *sirilic-part1.ssf* dans le sous-dossier *script* du dossier de travail
- Lancement du script *sirilic-part.ssf* :
  - Il crée les fichiers maîtres et pré-traite toutes les images.
  - Dans le cas d'une couche avec une seule session, il aligne et empile.
  - Pour les couches ayant plusieurs sessions:
    - Sirilic regroupe toutes les sessions d'une couche en une seule (Merge)
    - Il aligne et empile les sessions fusionnées.
- Pour une caméra CCD mono, il aligne les couches L, R, V, B, Ha... Ainsi elle sont prêtes à être combinées sous Siril pour former la photo couleur finale.
- Le résultat du traitement est stocké directement dans le sous-dossier *objet* (dans notre cas, c'est *M31*) à la racine du dossier de travail de **Siril**.

Si l'option *Après avoir exécuté le script, lancer Siril* est cochée dans les préférences, alors Siril est lancé automatiquement en fin de script.



Nous avons une image linéaire qu'il va falloir traiter et délinéarisée pour la rendre plus présentable.

## Partie 2 : Le traitement

Le traitement est la partie la plus compliquée et la plus intéressante du travail. C'est celle qui va permettre d'extraire l'image finale de toute cette gange photométrique.

Dans la phase de pré-traitement, tout est fortement structuré et balisé, on joue sur quelques leviers pour adapter le process à nos sources et c'est tout.

Dans la phase de traitement, on a une boîte à outils et il va falloir les essayer, sélectionner ceux qui sont pertinents pour notre image. Il ne faudra pas oublier de ne pas trop forcer sur les différents outils pour garder une image naturelle et aussi pouvoir utiliser les autres outils dans la suite du traitement.

Conseil : conserver votre image à chaque étape avec un nom différent. On rajoute souvent un « \_ » avec les initiales ou trois lettres correspondant à l'étape. Ainsi il vous sera beaucoup plus facile de revenir sur vos pas.

Commencez par sauver l'image linéaire issue du post-traitement sous un nom compréhensible. C'est votre image initiale de référence. Si vous vous loupez ou si vous perdez le fil, ne perdez pas de temps à essayer de corriger et/ou annuler vos erreurs. Repartez tout de suite de cette image.

Commençons par un état des lieux de l'image obtenue

### 1) Analyse de l'image obtenue

La première étape va consister à faire l'état des lieux de l'image obtenue en pré-traitement. Positionner la visualisation sur auto-ajustement.

## Table des matières

Avant Propos .....	2
Partie 1: Le Pré-traitement.....	3
A] Méthode automatique.....	4
1) Principes et organisation.....	4
2) Les scripts natifs .....	5
3) Scripts additionnels .....	6
4) Mise en œuvre.....	8
B] Méthode manuelle.....	10
1) Synoptique traitement des images en mode manuel .....	<b>Erreur ! Signet non défini.</b>
2) Réalisation du master dark.....	13
3) Réalisation du master Offset (biases=offset) .....	14
4) Réalisation du master flat .....	15
5) Pré-traitement des lights (brutes).....	17
6) SIRILIC : Pre-traitement avec interface graphique .....	27