



ÉCOLE NATIONALE  
DES SCIENCES  
GÉOGRAPHIQUES

Ecole Nationale des  
Sciences  
Géographiques



# Surveillance de l'Environnement Assistée par Satellite pour l'Océan Indien



Unité Mixte de  
Recherche  
Espace pour le  
Développement

Stage de fin d'études  
Cycle des Ingénieurs diplômés de l'ENSG 3<sup>ème</sup> année

Septembre 2017

# **Automatisation de l'acquisition et du traitement des images Sentinel-2 pour le calcul d'indices de végétation aidant à la prévention des pics de paludisme à Madagascar**



Charlotte Wolff

Maitre de stage : Christophe Revillion

Non confidentiel    Confidential IGN    Confidential Industrie    Jusqu'au ...

ÉCOLE NATIONALE DES SCIENCES GÉOGRAPHIQUES  
6-8 Avenue Blaise Pascal - Cité Descartes - 77420 Champs-sur-Marne  
Téléphone 01 64 15 31 00 Télécopie 01 64 15 31 07

## **Jury**

### **Président de jury :**

Serge Botton

### **Commanditaire :**

Institut Pasteur de Madagascar

### **Encadrement de stage :**

Christophe Revillion et Vincent Herbreteau (*UMR Espace-Dev*)

### **Responsable pédagogique du cycle Ingénieur :**

Serge Botton, IGN/ENSG/DE/DPTS

### **Tuteur du stage pluridisciplinaire :**

Marc Poupée, IGN/ENSG/DIAS

© ENSG

## **Stage de fin d'étude du xxx au xxx**

**Diffusion web :**  Internet  Intranet Polytechnicum  Intranet ENSG

### **Situation du document :**

Rapport de stage de fin d'études présenté en fin de 3<sup>ème</sup> année du cycle des Ingénieurs

**Nombres de pages :** 45 pages dont 6 d'annexes

**Système hôte :** L<sup>A</sup>T<sub>E</sub>X

### **Modifications :**

EDITION	REVISION	DATE	PAGES MODIFIEES
1	0	09/2012	Création

# **Remerciements**

écrire les remerciements

## Résumé

La thèse réalisée par Florian Girond portait sur le développement d'une application web pour la surveillance des pics de paludisme à Madagascar à partir de données environnementales et de données fournies par des médecins présents sur le terrain. Le projet TOSCA Sentinel-2, financé par le CNES, a pour objectif de compléter cette application avec des données issues des images multi-spectrales des satellites Sentinel-2 dont la résolution atteint 10m et dont la fréquence d'acquisition est de 10 jours actuellement et devrait atteindre bientôt 5 jours.

Le stage réalisé avec l'UMR Espace-Dev, au sein des locaux du SEAS-OI à Saint-Pierre de la Réunion a donc pour objectif de tester et de comparer les différentes chaînes d'acquisition et de traitement des images Sentinel-2, permettant de les corriger de l'effet de l'atmosphère et de créer un masque de nuages. Les chaînes testées sont Sen2cor, MAJA et MUSCATE. Sen2cor donne des résultats moins bons que les deux autres chaînes mais ils restent suffisants pour l'application que l'on veut en faire, d'autant que la chaîne de traitement Sen2cor est plus facile à prendre en main.

Les images téléchargées et traitées sont ensuite rangées dans une base de données et des images d'indices -de végétation dans un premier temps- sont calculées et rangées dans la base de données.

Ces indices sont ensuite utilisés pour créer des fichiers de séries temporelles facilement utilisables par l'Institut Pasteur de Madagascar pour aider à prédire les pics de paludisme sur le territoire malgache. Le produit attendu a été défini lors d'un séjour de cinq jours à l'institut situé à Antananarivo, à Madagascar.

L'automatisation a été réalisé sous un environnement Linux, avec des scripts codés en bash ou en Python. Des bibliothèques telles que Orpheo ToolBox et GDAL ont été utilisées.

**Mots clés :** Télédétection, Images multi-spectrales, Sentinel-2, Sen2cor, PEPS, projet TOSCA Sentinel-2, Paludisme

## Résumé

The PhD thesis written by Florian Girond deals with the development of a web-based application for Malaria Surveillance in Madagascar using environmental data and data from healthcare providers in the field. The TOSCA Sentinel-2 project, funded by the CNES, is aimed at enriching this application with data from Sentinel-2 satellites multispectral images. Their resolution is 10m and their acquisition frequency is of 10 days.

The internship with the UMR Espace-Dev, took place in the SEAS-OI premises, in Saint-Pierre de la Réunion. The goal was to test and compare different acquisition and processing tools for Sentinel-2 images in order to correct them from the atmospheric effects and create cloud masks. The three tested processing tools are Sen2cor, MAJA and MUSCATE. Sen2cor is less efficient than MAJA and MUSCATE but sufficient enough for our use. Furthemore Sen2cor is easier to use.

The downloaded and corrected images are then collected in a database and index images -of vegetation first- are computed and collected in the database in a second step.

These index are used to create time series files that the Pasteur Institut of Madagascar can easily use to predict malaria's peaks throughout Madagascar. The expected results have been discuss during a journey in Antananarivo (Madagascar).

This automatization has been done in a Linux environment, with scripts written in Python and bash. Libraries such as Orpheo ToolBox and GDAL have been used.

**Key words :** Remote sensing, Multispectral images, Sentinel-2, Sen2cor, PEPS, TOSCA Sentinel-2 project, Malaria

# Table des matières

<b>Glossaire et sigles utiles</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
<b>1 Chapitre 1</b>	<b>5</b>
1.1 Les objectifs du stage . . . . .	5
1.2 Les organismes d'accueil . . . . .	6
1.3 La gestion de projet . . . . .	7
<b>2 Chapitre 2</b>	<b>8</b>
2.1 Caractéristiques des images Sentinel-2 . . . . .	8
2.2 Acquisition des images Sentinel-2 . . . . .	9
2.3 Logiciels de prétraitement des images . . . . .	10
2.4 Comparaison et choix de la chaîne de prétraitement . . . . .	14
2.5 Création du masque des nuages . . . . .	17
<b>3 Chapitre 3</b>	<b>19</b>
3.1 Organisation de la base de données . . . . .	19
3.2 Acquisition et traitement des données déjà disponibles . . . . .	21
3.3 Automatisation pour l'acquisition des données futures . . . . .	21
3.4 Ajout de nouveaux indices . . . . .	21
<b>4 Chapitre 4</b>	<b>23</b>
4.1 Format du fichier CSV demandé par l'Institut Pasteur de Madagascar . . . . .	23
4.2 Développement du code . . . . .	24
<b>Conclusion</b>	<b>28</b>
<b>A Déroulement du stage</b>	<b>33</b>
<b>B Comparaison des chaînes de prétraitement sur l'île de La Réunion</b>	<b>36</b>
<b>C Algorithmes de création de la base de données</b>	<b>38</b>
<b>D Algorithmes de création du fichier .csv des séries temporelles</b>	<b>40</b>
<b>E Organisation finale du dossier</b>	<b>43</b>

# Glossaire et sigles utiles

**ENSG** École Nationale des Sciences Géographiques

**SEAS-OI** Surveillance de l'Environnement Assistée par Satellite pour l'Océan Indien

**Télédétection** Discipline scientifique regroupant l'ensemble des connaissances et des techniques pour l'observation, l'analyse et l'interprétation de l'environnement à partir d'images obtenues à l'aide de plates-formes aéroportées, spatiales, terrestres ou maritimes

**UMR Espace-Dev** Unité Mixte de Recherche, Espace pour le Développement

**IPM** Institut Pasteur de Madagascar

**CESBIO** Centre d'Etudes Spatiales de la BIosphère

**Theia** Pôle de données et de services surfaces continentales du CNES

**Sentinel-2** Série de satellites d'observation de la Terre de l'Agence spatiale européenne développée dans le cadre du programme Copernicus

**PEPS** Plateforme d'Exploitation des Produits Sentinel, développé par le CNES

**Sen2cor** Sentinel 2 Correction, outil de pré-traitement des images Sentinel-2 pour corriger des effets de l'atmosphère et la détection des nuages, développé par l'ESA

**MAJA** MACCS ATCOR Joint Algorithm, chaîne de détection des nuages et de correction atmosphérique, adaptée au traitement des séries temporelles d'images, développé par le CNES

**MUSCATE** MUlti Satellite, multi-CApteurs, pour des données multi-TEmporelles, chaîne d'acquisition d'images corrigées grâce à des séries temporelles d'images, développé le CNES et le CESBIO, au sein du pôle Téhia

**UTM** Universal Transverse Mercator, type de projection cylindrique, conforme à la surface de la terre, découpant cette dernière en 120 zones

**WGS 84** World Geodetic System 1984

**NDVI** Normalized Difference Vegetation Index

**SNAP** Sentinel Application Platform, outils de traitement des images Sentinel, développé par l'ESA

**OTB** Orpheo ToolBox, outils de traitements des images de télédétection, développé par le CNES

# Introduction

La thèse réalisée par Florian Girond portait sur le développement d'une application web pour la surveillance des pics de paludisme à Madagascar à partir de données environnementales et de données fournies par des médecins présents sur le terrain. Le projet TOSCA Sentinel-2, financé par le CNES, a pour objectif de compléter cette application avec des données issues des images multi-spectrales des satellites Sentinel-2 dont la résolution atteint 10m et dont la fréquence d'acquisition est de 10 jours actuellement et devrait atteindre bientôt 5 jours.

Le stage réalisé avec l'UMR Espace-Dev, au sein des locaux du SEAS-OI à Saint-Pierre de la Réunion a donc pour objectif de tester et de comparer les différentes chaînes d'acquisition et de traitement des images Sentinel-2, permettant de les corriger de l'effet de l'atmosphère et de créer un masque de nuages. Les chaînes testées sont Sen2cor, MAJA et MUSCATE. Sen2cor donne des résultats moins bons que les deux autres chaînes mais ils restent suffisants pour l'application que l'on veut en faire, d'autant que la chaîne de traitement Sen2cor est plus facile à prendre en main.

Les images téléchargées et traitées sont ensuite rangées dans une base de données et des images d'indices -de végétation dans un premier temps- sont calculées et rangées dans la base de données.

Ces indices sont ensuite utilisés pour créer des fichiers de séries temporelles facilement utilisables par l'Institut Pasteur de Madagascar pour aider à prédire les pics de paludisme sur le territoire malgache. Le produit attendu a été défini lors d'un séjour de cinq jours à l'institut situé à Antananarivo, à Madagascar.

L'automatisation a été réalisé sous un environnement Linux, avec des scripts codés en bash ou en Python. Des librairies telles que Orpheo ToolBox et GDAL ont été utilisées.

# Chapitre 1

## Déroulement du stage

### 1.1 Les objectifs du stage

**Le projet TOSCA Sentinel** Le stage réalisé se déroule dans le cadre du projet TOSCA Sentinel-2 for Malaria Surveillance, financé par le CNES. L'objectif principal de ce projet est d'utiliser les informations fournies par les satellites Sentinel-2 afin d'aider à la surveillance du paludisme en Afrique. Ce projet s'applique principalement dans deux pays : L'Afrique du Sud et Madagascar. Il s'agit de deux pays connaissant une récente augmentation du nombre de cas de paludisme. En Afrique du Sud notamment, le taux de mortalité a considérablement diminué ( $>75\%$ ) depuis 2000 mais le paludisme persiste, avec des réintroductions régulières dans les zones où l'environnement y est favorable.

Ce projet s'inscrit dans la continuité de la thèse de Florian Girond [[thèse](#)] basée à l'Institut Pasteur de Madagascar. Elle portait sur le développement d'une application web pour l'analyse et la surveillance des pics de paludisme à Madagascar. L'application a pour objectif de signaler lorsqu'un pic de paludisme va être observé dans certaines zones de Madagascar afin d'améliorer l'efficacité des interventions des organismes de santé publique. Pour prédire ces pics de paludisme, l'application prend en compte :

- Des messages envoyés par des médecins sur le terrain, renseignant les cas de paludisme
- L'intégration en temps quasi-réel d'indicateurs favorables au développement du paludisme (indices de végétation, pluviométrie, etc.).

Ces informations sont stockées dans une base de données.

Le stage a pour but d'avancer sur le deuxième point, en alimentant le système de surveillance du paludisme de l'IPM avec des séries temporelles d'indicateurs favorables au développement du paludisme. Pour le stage, on s'intéresse au NDVI (Normalized Difference Vegetation Index) mais le code doit être facilement réutilisable pour intégrer d'autres indices. Lors du stage, j'ai donc dû automatiser :

- Le processus d'acquisition des images satellitaires Sentinel-2 au-dessus de l'île de Madagascar,
- Les corrections atmosphériques pour pouvoir avoir des images comparables entre elles,
- La détection des nuages, en mettant une valeur d'indice fixée (= 99) si il y a présence d'un nuage au dessus du pixel,
- Le stockage des images multispectrales et des images avec les indices dans une base de données sur laquelle on pourra ensuite faire des requêtes spatio-temporelles.

L'application développée par Florian Girond utilise déjà des séries temporelles d'indices de végétation issus du système américain MODIS (Moderate-Resolution Imaging Spectroradiometer), mis à disposition par l'IRI via le site : <http://iri.columbia.edu>. Le projet TOSCA Sentinel-2 a également

pour vocation de voir en quoi les images Sentinel-2 peuvent améliorer ces séries temporelles.

**Les images Sentinel-2 et les chaînes de traitement associées** Les images Sentinel-2 sont des produits récents et de nouvelles chaînes de traitement pour ces données ont vu le jour autour de ces données. Ce stage a été également l'occasion de tester ces chaînes de traitement et de les comparer. Les résultats de la comparaison intéressent l'UMR Espace-Dev, notamment pour choisir lesquelles utilisées dans le futur, pour quelles applications et pour connaître les avantages et inconvénients de chaque chaîne.

### 1.2 Les organismes d'accueil

**SEAS-OI** Le stage s'est déroulé dans les locaux de SEAS-OI à Saint-Pierre. Il s'agit d'une plate-forme d'acquisition d'images satellitaires et de traitement d'images haute résolution, qui couvre toutes les îles de l'Océan Indien. Elle a été mise en service en juillet 2012. Le site est géré par l'IRD, l'Université de la Réunion et la région Réunion.

**L'UMR Espace-Dev** Les encadrants du stage sont Christophe Révillion et Vincent Herbreteau. Ils travaillent tous les deux à l'UMR Espace-Dev. L'UMR Espace-Dev a des implantations à la maison de la télédétection de Montpellier, en Nouvelle-Calédonie, en Martinique, en Guyane et à la Réunion. Ses objectifs concernent des recherches sur les dynamiques spatiales telles que l'évolution des sociétés, les risques liés aux maladies émergentes en fonction de paramètres environnementaux, la vulnérabilité des territoires aux changements globaux.

**L'Institut Pasteur de Madagascar** L'IPM est un établissement scientifique privé, membre du réseau international des Instituts Pasteur dont le siège est à Paris. Il a pour mission de contribuer à la prévention et au traitement des maladies ainsi qu'au développement économique par des activités de recherche, de formation et de santé publique.

Dans le cadre de l'intégration de mon stage dans le projet Tosca S2 Malaria, j'ai eu l'occasion de rencontrer les membres de l'IPM à Antananarivo, du 11 au 15 juin 2017.

Cette mission a été l'occasion de voir dans quelles conditions travaillent les membres de la cellule SIG de l'IPM et de voir comment fonctionne l'application développée par Florian Girond. Les différentes réunions ont permis de mieux cibler leurs besoins et de définir la forme des livrables qu'ils attendent.

Ainsi, leur débit internet ne permettant pas le téléchargement d'images, le livrable devra être un fichier .csv avec les séries temporelles des NDVI, pour différents sites épidémiologiques. Ces sites sont fournis dans un fichier shapefile. Ce fichier .csv, beaucoup plus léger, pourra être facilement téléchargeable par l'IPM pour l'insérer dans la base de données du système de surveillance Tosca Sentinel.

### 1.3 La gestion de projet

Le stage a débuté le 18 avril 2017, pour une durée de 5 mois, jusqu'au 18 septembre 2017. Lors de mon arrivé à l'institut, les objectifs de stage ont été rapidement fixés, afin d'établir un calendrier prévisionnel du stage.

Lors de la première semaine de stage, j'ai donc réalisé une première recherche bibliographique sur les images Sentinel-2, les chaînes d'acquisition et de traitement associées, sur les bases de données et sur la programmation en bash. Cela m'a permis d'avoir une vue globale sur les données, les techniques et les méthodes qui seront abordées tout au long du stage. Suite à cela, j'ai réalisé le Gantt prévisionnel du stage.

Puis lors du stage, j'ai réalisé un carnet de bord permettant de connaître l'avancement du stage et de réaliser en fin de stage un Gantt réel du déroulement du stage. (Annexe A)

## Chapitre 2

# Acquisition et choix du pré-traitement des images

*Avant d'automatiser le processus de calcul des indices de végétation sur les images, il a fallu trouver les chaînes de traitement permettant :*

- D'acquérir les images Sentinel-2
- De corriger les images de l'effet de l'atmosphère
- De détecter et masquer les nuages

### 2.1 Caractéristiques des images Sentinel-2

Sentinel-2 est une série de satellites d'acquisition d'images de la surface de la Terre. Ils ont été développés par l'Agence Spatiale Européenne (*ESA*). Les deux premiers satellites sont :

- **Sentinel-2A**, lancé le 23 juin 2015
- **Sentinel-2B**, lancé le 7 mars 2017

Ils s'inscrivent dans le projet *Copernicus* qui consiste en l'observation de l'évolution des terres émergées au travers d'une Constellation de satellites (Sentinel 1 à 5). Les images acquises par Sentinel-2 ont des spécifications proches de celles acquises par les satellites des missions SPOT et Landsat mais avec une meilleure résolution spectrale et une répétitivité des acquisitions plus importante. (Table 2.1)

Les différentes bandes des images Sentinel-2 sont présentées Figure 2.1.

Caractéristiques	Sentinel-2	Spot	Landsat
Mise en orbite	2015	2002	1999
Période entre 2 acquisitions	5 jours	26 jours	16 jours
Meilleure résolution disponible	60 m	10 m	30 m
Taille des tuiles	290 km	60 km	185 km
Nombre de bandes	13	4	6

TABLE 2.1 – Caractéristiques des différentes missions d'observation de la surface de la Terre

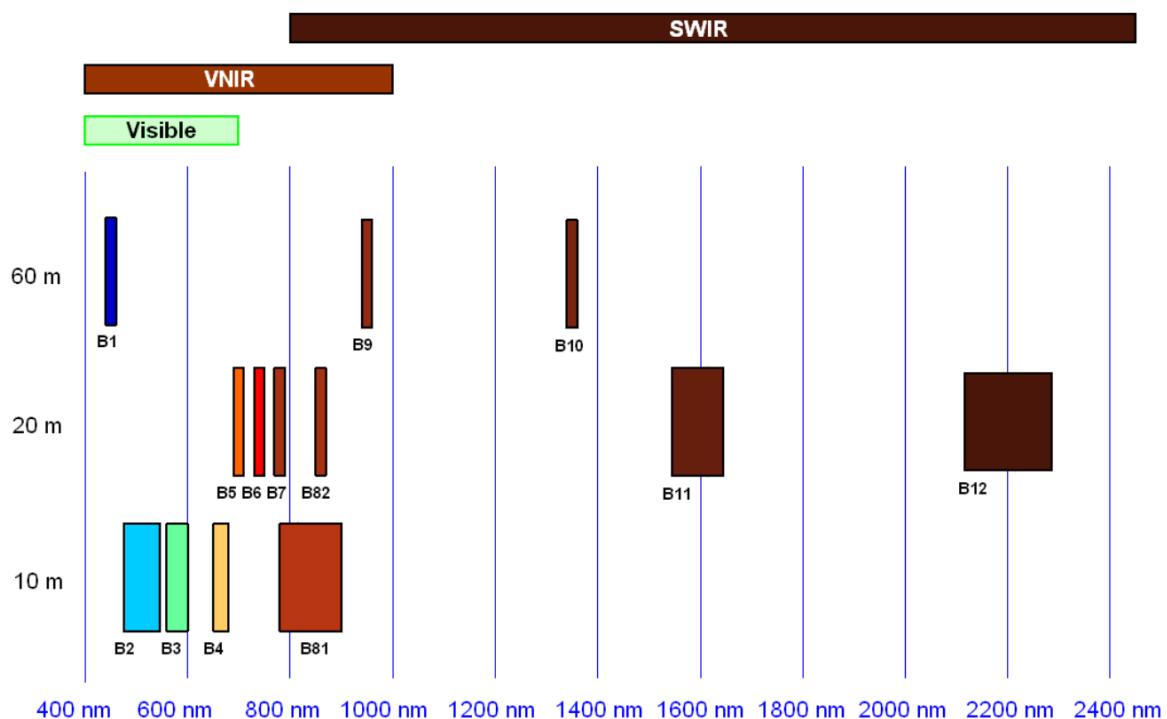


FIGURE 2.1 – Les différentes bandes spectrales au différentes résolutions des images sentinel-2

## 2.2 Acquisition des images Sentinel-2

Les images Sentinel-2 sont disponibles sur plusieurs plateformes de diffusion dont celle de l'ESA,

PEPS et Amazone. Le site :

<https://sentinel.esa.int/web/sentinel/missions/sentinel-2/acquisition-plans> permet d'accéder au planning des acquisitions de Sentinel-2 semaines après semaines, via *Google Earth*.

Olivier Hagolle du CESBIO a développé un script en Python 2.7 permettant de télécharger automatiquement un ensemble d'images Sentinel-2, en indiquant :

- Latitude et longitude de la dalle à télécharger
- La date d'acquisition de la première dalle à télécharger
- La date d'acquisition de la dernière dalle à télécharger

Un exemple de ligne de commande est :

```
python ./peps_download.py -c S2ST -lon 55 -lat -22 -a peps.txt -d 2017-01-01 -f 2017-04-01
```

Cette de ligne de commande télécharge les images Sentinel-2 au-dessus de la Réunion, pour les acquisitions entre le premier janvier et le premier avril 2017. Pour les images antérieures à 2017, la commande *S2ST* doit être remplacée par *S2*.

Ce code télécharge toutes les tuiles des images passant au-dessus de La Réunion. Cependant, certaines tuiles ne sont pas intéressantes. Lors du stage, j'ai donc modifié le code Python de Olivier Hagolle, afin de pouvoir ajouter un paramètre indiquant le nom des tuiles qui nous intéressent et que l'on souhaite télécharger. Ainsi, la ligne de commande :

```
python ./peps_download.py -c S2ST -lon 55 -lat -22 -a peps.txt -d 2017-01-01 -f
```

2017-04-01 -t T40KCB, T40KBB

ne va télécharger que les tuiles nommées T40KCB et T40KBB. Ce code ne fonctionne que pour les données après 2017.

Les données sont alors stockées dans un dossier zippé qui, après décompression porte l'extension SAFE. Un exemple de titre de dossier est :

*S2A\_MSIL1C-20170102T063502\_N0204-R134\_T40KCB\_20170102T063503.SAFE*

- Tous les dossiers téléchargés portent le préfixe : S2A\_MSIL1C
- T40KCB est le nom de la tuile
- 20170102 est la date d'acquisition de l'image

Le répertoire contenant les bandes de l'image multi-spectrale est GRANULE/IMG\_DATA. On y retrouve les 13 bandes et une couche d'extension TCI. A noter également que les données téléchargées par Sentinel-2 contient également un premier masque des nuages. Le masque des nuages porte le nom *MSK\_CLOUDS\_B00.gml*, dans le répertoire GRANULE/QI\_DATA.

## 2.3 Logiciels de prétraitement des images

Une fois les images Sentinel-2 téléchargées, ces dernières doivent être traitées pour les corriger de l'effet de l'atmosphère (Top of Canopy). Cela permettra d'obtenir des images comparables entre-elles. Le prétraitement permet également d'obtenir un masque des nuages, qui permettra d'extraire les zones ennuagées lors du calcul des indices de végétation.

Une deuxième étape du stage a donc été de tester les différentes chaînes de traitement disponibles afin de les comparer et de choisir celle qui donne les meilleurs résultats pour notre problématique. J'ai testé les trois chaînes de traitement suivantes :

- **Sen2cor**, développé par l'ESA.
- **MAJA**, développé par le CNES, le CESBIO et le DLR (Deutsche Zentrum für Luft- und Raumfahrt).
- **MUSCATE**, développé par le CNES et le CESBIO dans le cadre du pôle Théia.

### 2.3.1 Sen2cor [sen2cor]

Sen2cor s'utilise en ligne de commande Linux, directement sur les images téléchargées. Il s'agit d'un fichier bash à lancer en indiquant le nom du dossier contenant les images Sentinel-2 à corriger.

A la fin du traitement, un fichier avec le préfixe *S2A\_MSIL2A\_* est créé. Le sous-dossier GRANULE/IMG\_DATA contient les images corrigées de l'effet de l'atmosphère pour les différentes bandes et les différentes résolutions. De plus, deux images portant les suffixes *\_AOT* et *\_WVP* indiquent les corrections atmosphériques et les corrections dues à la vapeur d'eau qui ont été apportées sur les images.

Sen2cor crée également une image de classification *\_SCL*. Les classes sont ombres, nuage, végétation, sol/désert, eau, neige.

Un masque des nuages et un masque neigeux sont également créés dans le dossier GRANULE/QI\_DATA, respectivement avec le suffixe *\_CLD* et *\_SNW*. Le masque des nuages donne pour chaque pixel de

l'image la probabilité que le pixel soit un nuage.

**La correction atmosphérique** La correction atmosphérique se fait en utilisant des modèles de l'atmosphère présents dans une *Look-Up Table*(LUT). La correction à faire due à la présence des aérosols (AOT) dans l'atmosphère utilise l'algorithme DDV (Dense Dark Vegetation) qui fait des calculs de corrélation entre les bandes 12 (infrarouge de faible longueur d'onde) et les bandes rouge et bleue de l'image. Pour faire tourner l'algorithme, l'image doit contenir des zones avec des sols sombres et des zones d'eau. En sortie de l'algorithme, on obtient l'image \_AOT.

**La correction de la teneur en vapeur d'eau** L'algorithme utilisé se nomme APDA (Atmospheric Pre-corrected Differential Absorption) et utilise les bandes 8a et 9 pour calculer l'absorption de la vapeur d'eau.

**La détection des cirrus** Les cirrus sont détectés à partir de la bande 10 (1400nm) des images. Il s'agit d'une fine bande spectrale à une longueur d'onde absorbant fortement la vapeur en eau. La bande va ainsi absorber le signal réfléchi au sol mais recevoir le signal dû à la présence des cirrus. En corrélant cette bande à d'autres bandes spectrales, on peut alors enlever les cirrus des images.

**La correction topographique** Sen2cor permet également de corriger de l'effet de la topographie car celle-ci crée notamment des ombres dans les vallées. En rentrant un paramètre supplémentaire, il est possible de corriger cela, à partir d'un MNT de la zone d'acquisition de l'image. Nous avons comparé les résultats obtenus avec et sans la correction de la topographie sur les images au-dessus de la Réunion. Les résultats obtenus ne sont pas satisfaisants : On constate que l'ajout du MNT entraîne l'apparition d'artefacts au niveau des vallées.(Figure 2.2)

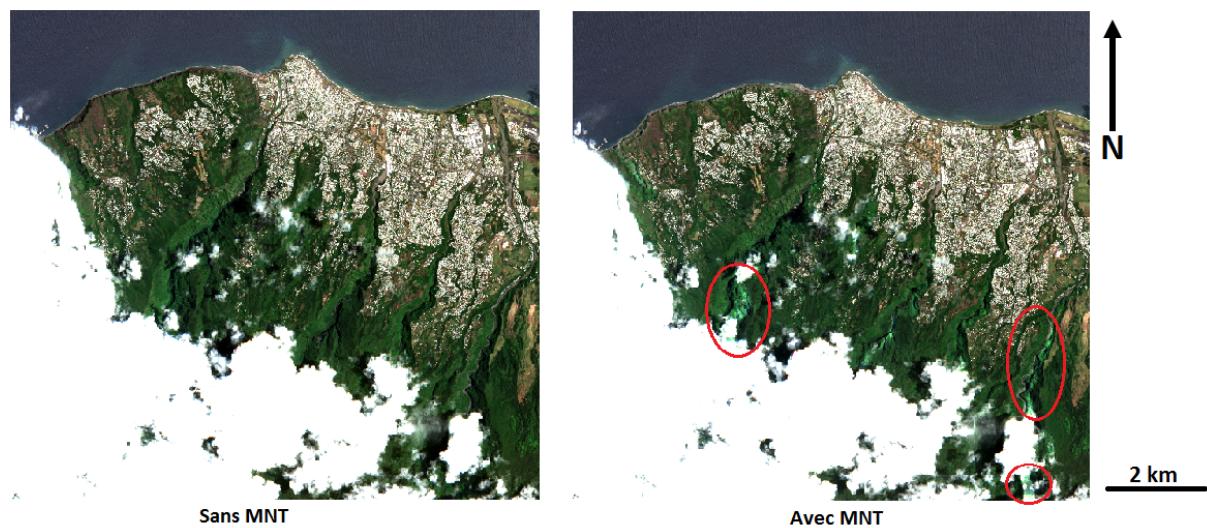


FIGURE 2.2 – Comparaison du résultat de la correction des images avec Sen2cor, avec et sans la correction de la topographie. En rouge, les artefacts lors de l'ajout d'un MNT

### 2.3.2 MAJA

MAJA, développée par le CNES, CESBIO et le DLR, est une chaîne de traitement pour toutes les images satellitaires, pas uniquement les images Sentinel. Il ne peut pas être utilisé sur UBUNTU.

Pour l'utiliser, j'ai dû installé sur l'ordinateur une machine virtuelle avec CentOS.

L'utilisation de MAJA s'effectue en deux étapes. Dans un premier temps, il faut télécharger les MNT et les données de vapeurs d'eau de l'atmosphère directement sur internet. Il faut ensuite créer un fichier texte portant le nom de la tuile de la zone et y enregistrer des informations sur la tuile tel que ses dimensions et la résolution. Puis un code python nommé *conversion\_format\_maja.py* permet de convertir les deux fichiers dans un format spécifique à MAJA.

Il est ensuite possible de lancer le traitement avec MAJA. Un fichier *param.txt* contient les informations sur les chemins d'accès aux différents fichiers nécessaires pour le traitement. Puis un code python nommé *lance\_maja.py* lance le traitement.

La chaîne de traitement MAJA utilise une méthode multi-spectrale et multi-temporelle pour l'estimation de l'AOT et la détection des nuages. Cela permettra d'affiner le masque des nuages ; les bâtiments et l'eau détectés avec Sen2cor ne le sont pas ici.

**La correction atmosphérique [maja2]** La correction de l'absorption faites par les molécules atmosphériques se fait avec un algorithme nommé SMAC (Simplified Model for Atmospheric Correction). L'algorithme utilise les informations sur les concentrations en ozone, oxygène et vapeur d'eau, provenant des métadonnées des images et de données météorologiques externes.

**Méthode multi-temporelle et multi-spectrale de détection des nuages et correction de l'AOT [maja1]** Le principe est de comparer l'image avec les images anciennes (multi-temporelle) et de comparer les valeurs des réflectances à différentes longueurs d'onde des pixels de l'image (multi-spectrale).

Pour la partie multi-spectrale, on compare l'image à corriger et une image composite, obtenue avec l'ensemble des images précédentes. Cette image composite est sans nuage. Pour comparer les images, on fait alors deux hypothèses :

- La réflectance à la surface du sol varie peu dans le temps.
- L'AOT varie peu avec la distance.

Pour corriger de l'effet de la présence des aérosols, on utilise un modèle atmosphérique dont il faut trouver les paramètres. On utilise la méthode du filtre de Kalman pour trouver, à partir de la nouvelle photo et des photos anciennes, les paramètres du modèle.

Pour la partie multi-spectrale, on suppose qu'il existe une relation linéaire reliant les bandes bleue, rouge et proche infrarouge à la surface du sol. On utilise les moindres carrés pour trouver les paramètres du modèle.

MAJA utilise à la fois le multi-spectral et le multi-temporel. La fonction coût à minimiser prend en compte à la fois les modèles temporel et spectral. On obtient alors une estimation des AOT pour l'image et un masque des nuages.

L'une des contraintes de MAJA est qu'il faut donc une profondeur historique des images pour l'utiliser. Ces images doivent être de relativement bonne qualité (peu de nuages).

**Correction topographique** De même que pour Sen2cor, MAJA corrige les images des ombres dues à la topographie. Pour cela, il faut donc télécharger les MNT de la zone où se trouve la photo à corriger.

### 2.3.3 MUSCATE [MUSCATE]

MUSCATE est une chaîne de traitement qui utilise le même algorithme de traitement que MAJA. Les images sont traitées par le CNES et le portail de Théia distribue ensuite les images corrigées sur son site. Nous n'avons donc pas accès à la chaîne de traitement pour la modifier. Un code python développé par Olivier Hagolle permet de récupérer directement les images sur le site pour les dates qui nous intéressent.

Il suffit par exemple taper la ligne de commande suivante :

```
python ./theia_download.py -t 'T31TCJ' -c SENTINEL2 -a config_theia.cfg -d 2016-09-01  
-f 2016-10-01
```

Cette ligne de commande va télécharger toutes les images de la tuile T31TCJ pour les dates entre le premier septembre 2016 et le premier octobre 2016.

Le deuxième inconvénient de MUSCATE est que les images corrigées ne sont pas disponibles de suite après l'acquisition des images Sentinel-2. Il y a un délai de quelques jours entre l'acquisition des images et leur traitement et intégration sur le site.

## 2.4 Comparaison et choix de la chaîne de prétraitement

Lors de mon stage, j'ai téléchargé un ensemble de jeu de données afin de comparer les trois chaînes de traitement et choisir laquelle sera utilisée pour l'automatisation des corrections. Quatre zones d'étude ont été choisies pour comparer les masques de nuage créés : La Réunion et 3 zones sur Madagascar (Figure 2.4).

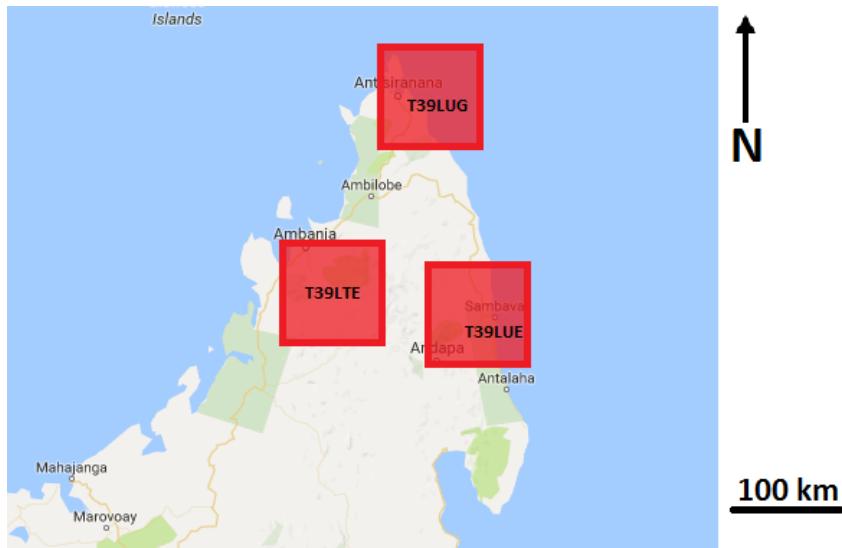


FIGURE 2.3 – Zones d'étude choisies pour la comparaison des chaînes de traitement

Pour chaque zone d'étude, l'image est corrigée à la fois avec Sen2cor et MAJA et l'image corrigée avec MUSCATE est téléchargée. Les résultats des différentes chaînes de traitement ont des tailles différentes.(Table 2.2) Les images sont ensuite intégrées dans QGIS, ainsi que le masque de nuage initial fourni lors du téléchargement de l'image Sentinel-2. Des dates avec peu de nuages ont été choisies. Les masques de nuages ainsi que les résultats des corrections atmosphériques ont été comparés.

Chaîne de traitement	Non-traité	Sentinel-2	MAJA	Muscate
Poids	800 Mo	1.4 Go	3.6 Go	3.6 Go

TABLE 2.2 – Poids des résultats en sortie des différentes chaînes de traitement

### 2.4.1 Comparaison des masques de nuages

**Sen2cor** Sen2cor détecte presque tous les nuages mais le masque ne les englobe pas entièrement. Il faudrait faire une dilatation du masque pour prendre en compte les nuages en entier. De plus, Sen2cor ne faisant pas de multi-temporel, les toits clairs, les plages et certaines zones d'eau, qui ont une signature spectrale proche de celle des nuages, sont considérés comme des nuages. Les cirrus ne sont également pas détectés. En revanche, le masque fourni par les images Sentinel-2 détecte ces cirrus. Il serait donc intéressant de fusionner ces 2 masques pour avoir également le masque de cirrus.

Un autre inconvénient à Sen2cor est qu'il ne détecte pas les ombres des nuages.

Pour utiliser Sen2cor dans l'automatisation, il serait donc intéressant de dilater le masque des nuages pour qu'il prenne en compte les nuages en entier.

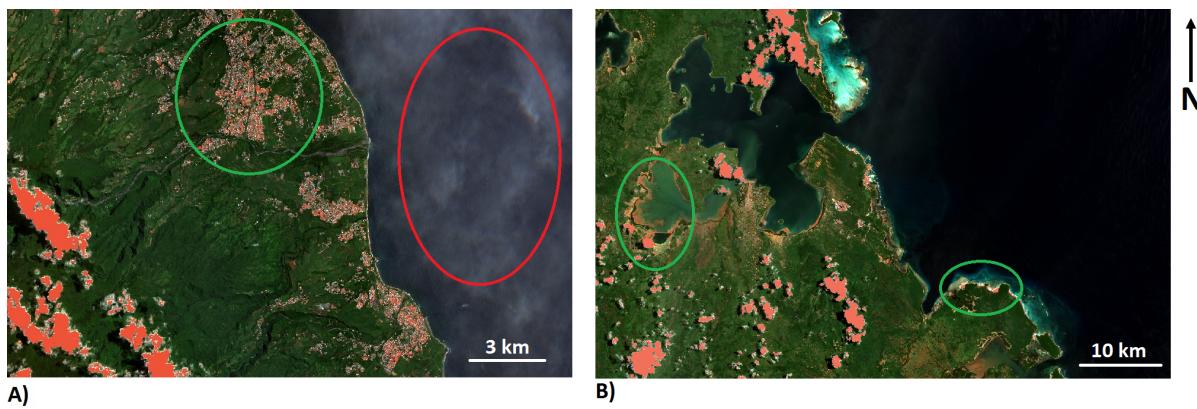


FIGURE 2.4 – Résultats avec le masque issu de Sen2cor. En vert, des zones détectées comme étant des nuages (toits de maisons, sable). En rouge, les cirrus ne sont pas détectés.

**MAJA** Avec MAJA, les masques sont plus gros qu'avec Sen2cor : pour les nuages détectés, le masque les englobe entièrement. Mais, MAJA ne détecte pas tous les nuages, contrairement à Sen2cor. Néanmoins, comme la méthode est multi-spectrale, les cirrus sont détectés mais les zones habitées et les plages ne sont pas considérés comme des nuages.

**MUSCATE** Les résultats que l'on obtient avec MUSCATE sont similaires à MAJA, si ce n'est que le masque est moins large autour des nuages détectés. Il détecte également une partie des ombres des nuages contrairement aux deux autres chaînes de traitement. Mais il détecte comme des nuages ou des ombres de nuages des zones qu'il ne devrait pas détecter.(Figure 2.5)

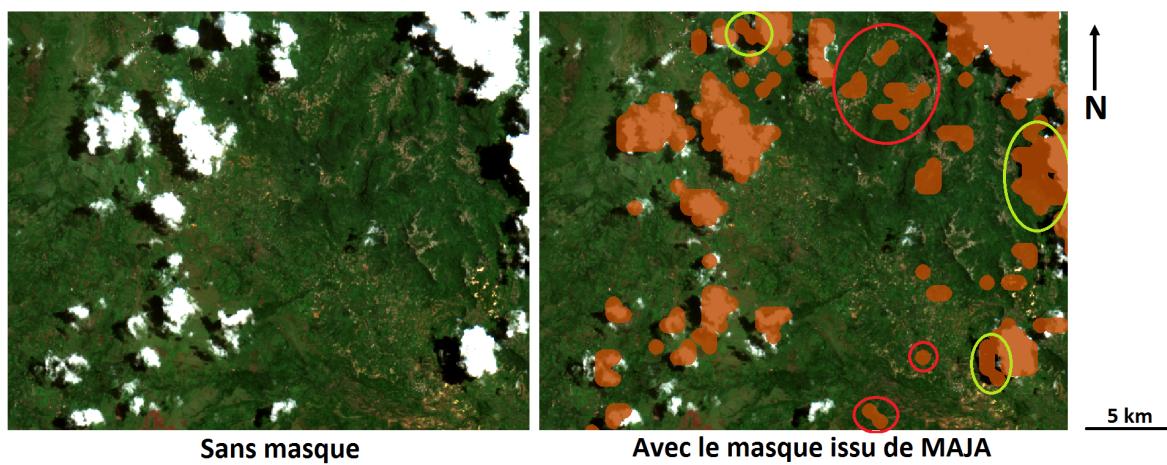


FIGURE 2.5 – Résultat avec le masque issu de MUSCATE. En vert, les ombres des nuages détectés et en rouge des erreurs.

La Table 2.3 résume les avantages et inconvénients de chaque chaîne de traitement et l'Annexe B compare les chaînes de traitement en images.

#### 2.4.2 Comparaison des corrections atmosphériques

Pour pouvoir réaliser des séries temporelles des NDVI, les images sur une même zone doivent pouvoir être comparées les unes avec les autres. Par exemple, les pixels correspondant à des bâtiments

	<b>Avantages</b>	<b>Inconvénients</b>
Sen2cor	<ul style="list-style-type: none"> <li>-Déetecte tous les nuages</li> <li>-Fonctionne sur tous les systèmes d'exploitation Linux</li> <li>-Pas besoin de profondeur historique</li> </ul>	<ul style="list-style-type: none"> <li>-Ne détecte pas les ombres des nuages</li> <li>-Masque les toits clairs, les plages et zones d'eau</li> <li>-Ne détecte pas les cirrus</li> </ul>
MAJA	<ul style="list-style-type: none"> <li>-Déetecte les cirrus</li> <li>-Englobe entièrement les nuages détectés</li> </ul>	<ul style="list-style-type: none"> <li>-Ne fonctionne pas sur UBUNTU</li> <li>-Nécessite une base de données préalable pour la profondeur historique</li> <li>-Préparation de fichiers .txt en amont</li> </ul>
MUSCATE	<ul style="list-style-type: none"> <li>-Déetecte les cirrus</li> <li>-Déetecte les ombres des nuages</li> <li>-Masque similaire à MAJA mais moins large</li> </ul>	<ul style="list-style-type: none"> <li>-Pas d'accès au code</li> <li>-Données non accessibles le jour de l'acquisition</li> <li>-Déetecte des zones fausses</li> </ul>

TABLE 2.3 – Comparaison des chaînes de prétraitement

doivent avoir des valeurs proches sur les différentes bandes pour les différentes dates. Pour cela, il faut donc corriger les images des effets de l'atmosphère pour avoir des valeurs similaires. Les différentes chaînes de traitement effectuent également cette correction. Les résultats obtenus ont donc également été comparés.

Suite à la comparaison des chaînes de traitement, la chaîne Sen2cor a été choisie pour l'automatisation des traitements.

## 2.5 Creation du masque des nuages

Nous avons donc choisi d'utiliser le masque de nuage fourni en sorti de la chane de traitement de Sen2cor. En utilisant QGis, j'ai ensuite ralise des tests pour amliorer le masque des nuages et obtenir finalement un fichier raster qui pour chaque pixel de l'image donne l'information sur la presence d'un nuage :

- 0 s'il n'y a pas de nuage
- 99 s'il y a un nuage

L'algorithme de traitement pour obtenir ce raster est present dans la Figure 2.6. Cet algorithme a et automatise avec une fonction Python que j'ai nomm *createCLDMask.py*.

Dans un second temps, l'image contenant pour chaque pixel la valeur du NDVI est cree en utilisant les bandes 4 (rouge) et 8 (infrarouge) issues du traitement de Sen2cor et la bibliotheque OTB. Le calcul de l'image se fait facilement en ligne de commande, avec la commande suivante :  
`otbcli_BandMath -il bande_4.jp2 bande_8.jp2 -out NDVI.tif -exp "ndvi(im1b1,im2b1)"`

On rappelle que la formule de l'indice NDVI est la suivante :

$$-1 < \frac{\text{bande}8 - \text{bande}4}{\text{bande}8 + \text{bande}4} < 1$$

Une fois ces deux images raster crees, elles sont combines afin d'avoir une valeur de NDVI unique-  
ment sur les pixels qui ne sont pas cachs par les nuages. Ceux cachs par les nuages prennent la  
valeur 99. Cela s'effectue grce  la bibliotheque GDAL et la ligne de commande suivante :

`gdal_calc.py -A NDVI.tif -B CLDmask -outfile=ndvi_with_mask.tif -calc="A*(B==0)+99*(B>0)"`

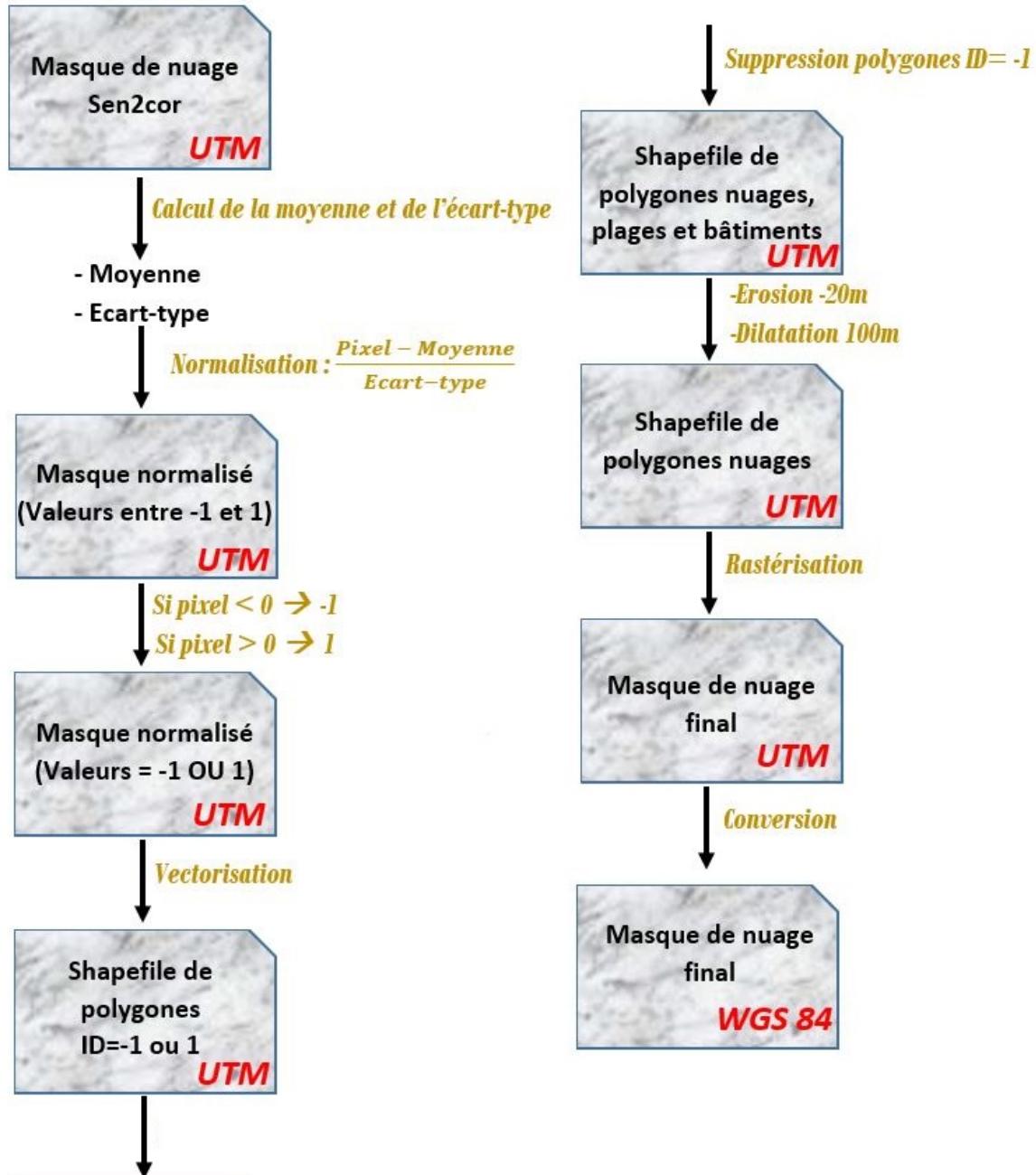


FIGURE 2.6 – Chaîne de traitement pour obtenir le masque de nuage utilisé. En sortie, les pixels portent la valeur 99 en présence de nuage et 0 sinon.

# Chapitre 3

## Automatisation de l'acquisition et du prétraitement des images

*L'automatisation du prétraitement des images utilise des scripts se lançant avec des lignes de commande Linux. La phase d'acquisition et de traitement des images a donc été codée en bash.*

### 3.1 Organisation de la base de données

La première question qui s'est posée au moment de l'automatisation a été de choisir comment ranger les données téléchargées et traitées afin qu'elles soient facilement requêtables. De plus, le code et la base de données avec les images doivent être contenus dans des dossiers différents pour pouvoir facilement accéder aux données tout en ne touchant pas aux fichiers de code.

En ce qui concerne le code, il a été séparé en trois parties et par conséquent en trois répertoires distincts :

- **PREPARATION** Un code pour acquérir, traiter et ranger les images déjà acquises par Sentinel-2
- **ACQUISITION** Un code qui se lance tous les jours pour récupérer les nouvelles images, acquises la veille, les traiter et les ranger
- **NEW\_INDEX** Un code qui permettra de calculer de nouveaux indices sur les images déjà acquises et traitées

Le dossier contenant le projet contient également un fichier *parametres.txt*. Ce fichier renseigne les longitudes et latitudes de la zone d'intérêt, ainsi que le nom des tuiles qui intéressent l'utilisateur. Il contient également le chemin vers le dossier *DATA* contenant la base de données à remplir avec les images. Enfin, un fichier *ReadMe.txt* présente le contenu de chaque répertoire, comment faire fonctionner le code et comment ajouter de nouveaux indices. Les dossiers *PREPARATION* et *ACQUISITION* contiennent également un répertoire *ACQUISITION* dans lequel seront stockées les images téléchargées grâce à PEPS mais pas encore traitées. Une fois les images traitées et rangées correctement dans la base de données, le dossier *ACQUISITION* est vidé.(Figure 3.1)

Le dossier *DATA* contient la base de données créée, avec deux sous-répertoires. Le premier contient pour chaque tuile et chaque date :

- Toutes les bandes corrigées avec Sen2cor,
- Le masque de nuages initial issu de Sentinel-2,
- Le masque de nuages issu de Sen2cor et modifié avec une fonction Python *createCLDMask.py*.

Le second sous-répertoire contient pour chaque tuile les images de NDVI pour les différentes dates. Le nom de chaque image de NDVI contient le nom de la tuile et la date. (Figure 3.2)

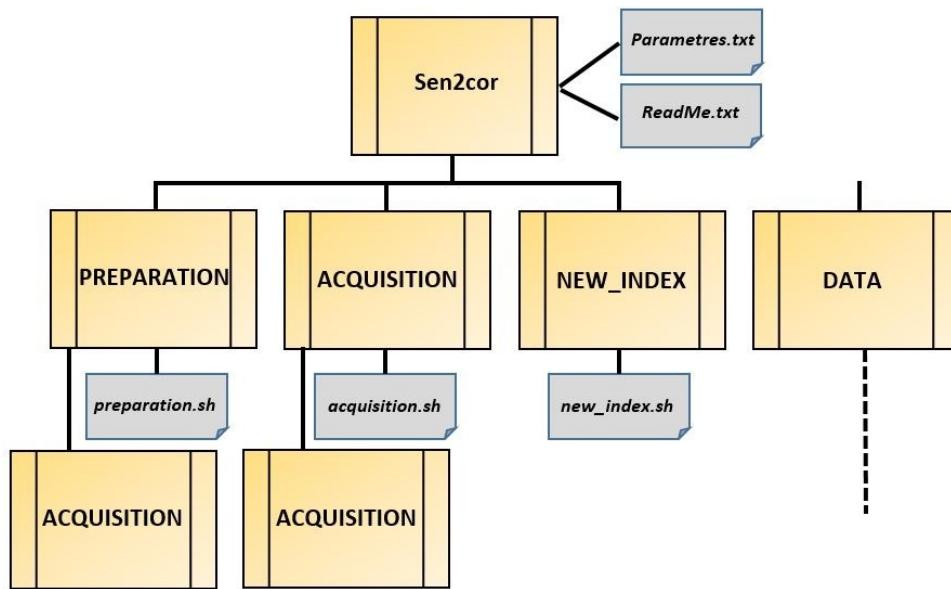


FIGURE 3.1 – Organisation du dossier du projet. Le dossier DATA contenant la base de données peut être contenu dans le dossier du projet ou dans un dossier séparé.

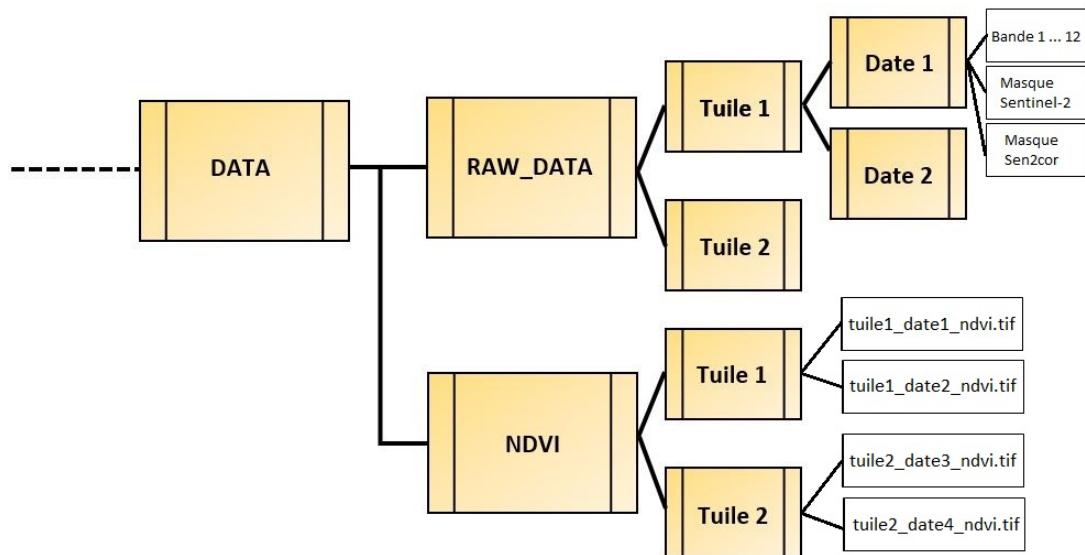


FIGURE 3.2 – Organisation du dossier DATA contenant la base de données requérable

## 3.2 Acquisition et traitement des données déjà disponibles

La première étape de l'automatisation consiste à récupérer, traiter et stocker les images déjà acquises par Sentinel-2. Cela s'effectue grâce aux codes du répertoire *PREPARATION*. Il contient le code *preparation.sh* à lancer. Ce code se compose de 4 grandes étapes (Annexe C) :

- Récupération dans le fichier *parametres.txt* des informations sur la zone d'intérêt et création des dossiers avec le nom des tuiles
- Téléchargement, traitement et stockage des images acquises avant décembre 2016. La fonction *treatment2016()* codée en bash est appelée.
- Téléchargement, traitement et stockage des images acquises après décembre 2016. La fonction *treatment2017()* codée en bash est appelée.
- Si les étapes précédentes ont pris plus d'une journée, de nouvelles images ont pu être acquises. S'il y en a, on télécharge, traite et stocke ces nouvelles images. La fonction *treatment2017()* codée en bash est appelée.

De plus, j'ai codé un script Python *createCLDMask.py* qui permet de créer le masque de nuage Sen2cor. Ce masque est également stocké dans *RAW\_DATA*. Les images NDVI calculées et stockées ne prennent pas en compte les nuages et le masque des nuages. Il ne sera utilisé que lors de la création du fichier .csv avec les séries temporelles des NDVI.

Une fois les images téléchargées, traitées et rangées dans la base de données, le code crée également dans la base un dossier *Water\_Mask*, contenant pour chaque tuile un masque indiquant pour chaque pixel si il est sur terre ou en mer. Cela sera utile lors de l'écriture du fichier .csv avec les séries temporelles des NDVI, car on ne veut prendre en compte ni les pixels sous un nuage, ni en mer.

## 3.3 Automatisation pour l'acquisition des données futures

Une fois avoir développé un code permettant de récupérer les images déjà disponibles, j'ai créé un deuxième code, stocké dans *ACQUISITION*, nommé *acquisition.sh* qui sera lancé **tous les jours ? ? ?** grâce à un Cron. Il s'agit d'un programme permettant au système Linux d'exécuter automatiquement des scripts et des commandes à une date et une heure définies à l'avance ou suivant un cycle prédéfini (dans notre cas, tous les jours).

Le programme ressemble à celui présenté plus haut pour l'acquisition des images déjà disponibles : Il télécharge toutes les images acquises la veille et y applique l'algorithme de correction Sen2cor. La fonction *Treatment2017()* présentée dans la partie précédente est ensuite appliquée à chaque image pour obtenir le masque des nuages et l'image des NDVI.

## 3.4 Ajout de nouveaux indices

Les scripts développés ont pour vocation d'être repris et enrichis par l'ajout de nouveaux indices, autres que le NDVI. L'ajout des nouveaux indices doit être facile, et sans à avoir à retélécharger toutes les images. Cela a été pris en compte :

- Les différentes bandes des images et le masque des nuages issu de Sen2cor sont stockés dans *DATA/RAW\_DATA*
- Le dossier *NEW\_INDEX* contient le code *new\_index.sh* permettant de créer les images avec le nouvel indice et de les stocker dans un dossier correspondant, pour toutes les images de toutes les tuiles, à toutes les dates

- Dans les codes *preparation.sh* et *acquisition.sh*, le calcul des images avec les indices se fait sous la forme d'une fonction, écrite au début du code bash. Il suffit de compléter cette fonction en rajoutant le calcul de nouveaux indices, en suivant l'exemple du NDVI.

# Chapitre 4

# Requête dans la base de données

#### **4.1 Format du fichier CSV demandé par l’Institut Pasteur de Madagascar**

L'un des objectifs de mon déplacement à l'Institut Pasteur de Madagascar était de définir le format du fichier avec les séries temporelles des indices de végétation. Pour faciliter le téléchargement à l'Institut Pasteur, le fichier doit être léger. Le fichier à créer est un fichier.csv dont le format est présenté Figure 4.1.

Semaine	Site 1	Site 2	Site 3	Site 4	Site 5	Site 6	Site 7	Site 8
05/01/2015								
12/01/2015								
19/01/2015								
26/01/2015								
02/02/2015								
09/02/2015								

Valeurs des indices pour chaque site pour chaque semaine

FIGURE 4.1 – Format du fichier csv à créer

L’Institut Pasteur nous a fourni une couche shapefile avec 34 sites épidémiologiques sur Madagascar. Le fichier .csv créé doit fournir une valeur de NDVI pour chaque site et chaque semaine. La valeur de NDVI est une valeur moyenne sur 5km autour du site, en ne prenant pas en compte les valeurs sous les nuages et dans la mer.

On calcule alors pour chaque image NDVI de la base de données une nouvelle image NDVI\_MEAN dont chaque pixel renseigne sur la valeur moyenne de NDVI 5 km autour de ce point et qui ne prend pas en compte les valeurs sous les nuages et dans les mers. On utilise pour cela deux commandes, l'une gdal et l'autre otb. (Figure 4.2)

Les images Sentinel téléchargées sont en UTM 38 ou 39S sur Madagascar. Les images brutes NDVI calculées ainsi que le masque de nuages sont donc également en UTM 38 ou 39S. Or le masque des mers et le shapefile des sites épidémiologiques fournis par Madagascar sont tous les deux fournis en WGS84. Nous avons donc choisi de convertir et de stocker dans la base de données :

- Les images NDVI en WGS84,
  - Les masques de nuages en UTM et en WGS84. (Tableau 4.1)

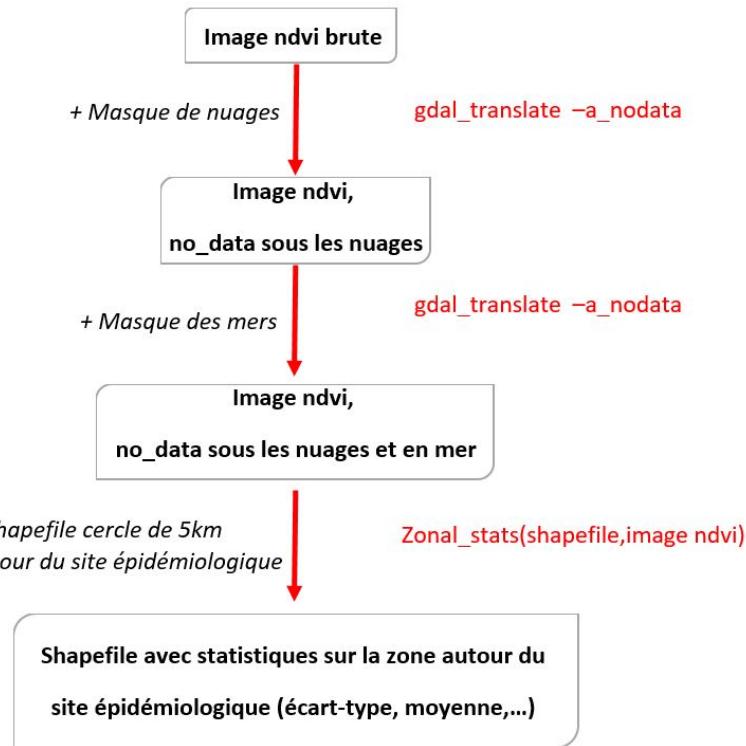


FIGURE 4.2 – Etapes pour créer l'image dont chaque pixel donne la valeur moyenne de NDVI sur 5km, en ne prenant pas en compte les nuages et les mers.

Pour convertir les images, on utilise la ligne de commande :

```
gdalwarp -ot Float32 -t_srs EPSG:4326 -r near -of GTiff -co COMPRESS=DEFLATE -co PREDICTOR=1 -co ZLEVEL=6 ndvi_UTM.tif ndvi_WGS84.tif
```

De plus, la valeur du NDVI choisie pour compléter le fichier .csv pour chaque site doit respecter certaines règles. Pour une semaine épidémiologique donnée et pour chaque site :

- **S'il y a une image pour le lundi de cette semaine** : On choisit cette image pour chercher la valeur du NDVI pour ce site,
- **S'il y a une image pour un autre jour de la semaine** : On cherche la valeur de NDVI sur cette image pour le site et on calcule la moyenne suivante :

$$ndvi_{final} = \frac{(ndvi_{semaine précédente} * joursemaine) + (ndvi_{semaine} * (7 - joursemaine))}{7} \quad (4.1)$$

- **S'il n'y a pas d'image pour cette semaine** : On reprend la valeur du NDVI de la semaine précédente pour ce site.

## 4.2 Développement du code

### 4.2.1 La base de données avec le masque des mers

Pour ne pas prendre en compte les NDVI au niveau de la mer, on utilise un masque. Il est issu de l'application *OpenStreet Map*. Il s'agit d'un shapefile de polygones dessinant les limites de Ma-

Nom	Type	Dossier	Projection
Bandes Sentinel-2	Raster	RawData	UTM38S ou 39S
Masque sen2cor	Raster	RawData	UTM38Sou39S et WGS84
Masque des mers	Shapefile (polygones)	waterMask	WGS84
Images NDVI	Raster	NDVI	WGS84
Sites épidémiologiques	Shapefile (points)		WGS84

TABLE 4.1 – Bilan des projections utilisées pour les différentes images et les shapefiles de la base de données

dagascar. Or nous avons les images NDVI pour chaque tuile Sentinel-2. Pour appliquer un masque de mer aux images NDVI, il faut redécouper le shapefile à la taille de chaque tuile.

J'ai donc développé un script Python *createWaterMask.py* permettant de créer dans la base de données un dossier *waterMask* qui contiendra pour chaque tuile un fichier shapefile dont les polygones correspondent à de la Terre. Ces shapefiles sont convertis en raster dont les pixels valent :

- 0 si c'est de la terre,
- 99 si c'est de la mer.

Pour découper le shapefile à la taille des tuiles, j'ai utilisé un deuxième shapefile qui renseigne la position de chaque tuile Sentinel-2. Pour découper les polygones et créer le dossier dans la base de données, j'ai utilisé la commande *ogr2ogr*.

Le code Python se lance en ligne de commande : `python createWaterMask.py parametres.txt` Le fichier *parametres.txt* renseigne le chemin d'accès au fichier shapefile des polygones du masque des mers que l'on veut découper à la taille des tuiles.

### 4.2.2 Ecriture du fichier .csv

Tout comme pour l'acquisition et le traitement des images Sentinel-2, l'écriture du fichier .csv se fait en deux temps.

- **Phase de préparation** : Le fichier .csv est créé et complété avec les séries temporelles des NDVI pour la période entre le début de l'acquisition des images et la date de la semaine actuelle.
- **Phase d'automatisation** : S'utilise avec un Cron. Chaque semaine, le fichier est complété des valeurs de NDVI pour la semaine en cours.

Les deux étapes ont été codées en Python et rangées dans un dossier CREATE\_CSV (Annexe D). Les deux étapes possédant des parties de code en commun, un troisième fichier Python est créé dans lequel sont écrites les fonctions appelées ensuite dans les deux autres scripts. (Table 4.2)

**Le code de préparation** Pour lancer le code permettant de créer le fichier .csv pour les semaines déjà disponibles, il faut tout d'abord renseigner le fichier *parametres.txt* avec les informations suivantes :

- **Index name** : Le nom de l'indice dont on veut la série temporelle, tel qu'il est indiqué dans la base de données
- **Starting date** : Date à partir de laquelle on veut la série temporelle
- **Stopping date** : Date jusqu'à laquelle on veut la série temporelle
- **Path to DATA folder** : Chemin d'accès vers la base de données

Nom	Utilité
params.txt	Informations sur les chemins d'accès aux données
csv_preparation.py	Code pour l'étape de préparation
csv_automation.py	Code pour l'étape d'automatisation
csvFunction.py	Regroupe les fonctions communes aux 2 codes Python précédents

TABLE 4.2 – Fichier et scripts Python pour la création et la mise à jour du fichier des séries temporelles de NDVI

- **Path to out csv file** : Chemin d'accès vers le fichier csv de sortie
- **Path to shapefile with epidemiologic sites** : Chemin d'accès vers le fichier contenant les points des sites épidémiologiques. Chaque point correspond à un site sur lequel on veut la valeur moyenne de l'indice 5km autour de ce site

La ligne de commande bash à lancer est alors :

```
python csv_preparation.py chemin/parametres.txt
```

Le code récupère tout d'abord les informations dans le fichier *parametres.txt* grâce à une fonction Python *readParams.py*. Puis, pour faciliter le traitement, il crée dans la base de données un dossier *Sites* contenant pour chaque site épidémiologique un shapefile avec le polygone cercle dont le centre est le site épidémiologique et de rayon 5km.

Le fichier .csv est ensuite créé et la première ligne écrite renseigne le nom de chaque site épidémiologique. Le nom est un code à trois lettres.

On peut alors commencer à remplir la série temporelle avec les indices en utilisant une boucle *while* : À partir de la première semaine épidémiologique renseignée dans le fichier des paramètres ("Starting date") et tant que la semaine à traiter n'est pas la dernière demandée ("Stopping date"), on calcule pour chaque site, la valeur moyenne de l'indice considéré en suivant les consignes de calcul citées plus haut.

Lorsqu'une image est trouvée pour une semaine épidémiologique et pour un site donné, une nouvelle image de l'indice est calculée pour la tuile correspondante en utilisant l'algorithme présenté dans la Figure 4.2. Cette image porte l'extension \_MEAN.tif. Avant de créer l'image, le script vérifie si elle existe déjà. Puis on utilise la fonction statistique de zone pour extraire la moyenne de l'indice dans le polygone autour du site épidémiologique. Pour faire cela, j'ai codé une fonction *find\_indexName.py*. Si un site est entièrement dans les nuages, la fonction *find\_indexName.py* renvoie *None*. Dans ce cas, on garde la valeur de l'indice obtenue la semaine précédente pour ce site.

Une fois le fichier renseigné pour toutes les semaines épidémiologiques, les images créées portant l'extension \_MEAN.tif sont supprimées. L'Annexe D présente les algorithmes développés pour créer le fichier .csv des séries temporelles.

**Le code d'automatisation** Le code d'automatisation permet tout d'abord de mettre à jour la dernière ligne du fichier .csv. En effet, si le code est lancé par exemple chaque mardi, on a déjà une valeur pour chaque site pour cette semaine. Mais une image peut être acquise pour cette semaine après le mardi. La semaine d'après, l'indice doit alors être recalculé en faisant la moyenne de l'indice de la semaine précédente et de cette semaine.

Puis le code calcule les indices pour les différents sites pour la semaine suivante et ajoute une ligne au fichier.csv.

La ligne de commande à lancer est la suivante :

```
python csv_automation.py chemin/parametres.txt
```

Le script récupère tout d'abord le chemin d'accès au fichier .csv grâce à la fonction codée, nommée *readParams.py*. Puis pour chaque site, il recalcule l'indice considéré pour la dernière semaine du fichier et la met à jour. Il calcule ensuite également l'indice pour la semaine suivante et écrit la nouvelle ligne dans le fichier.

Comme pour l'étape de préparation, le code crée la nouvelle image portant l'extension \_MEAN.tif pour calculer la moyenne sans prendre en compte les mers et les nuages. Puis à la fin du script, ces fichiers sont supprimés.

**Les fonctions codées** Comme les scripts *csv\_automation.py* et *csv\_preparation.py* possèdent des parties de code similaires, pour ne pas avoir de redondance dans les codes, ces parties ont été codées sous forme de fonctions (Table 4.3). Elles sont regroupées dans un fichier Python portant le nom *csvFunction.py*. Pour les importées ensuite dans les deux scripts Python, la ligne *import csvFunction as fct* est ajoutée au début de chaque script.

Nom	Utilité	Sorties
removeMeanFile	Supprime les fichiers _MEAN	0
find_indexName	Crée le fichier _MEAN	Valeur moyenne de l'indice
getCloudMask	Cherche la moyenne 5km autour du site	
readParams	Récupère le chemin du masque nuage	Chemin du masque nuage
buffer	Lit le fichier des paramètres	Liste des informations du fichier
createBDDpoints	Crée pour un site le shapefile de dilatation	0
	Crée les shapefiles polygones pour les sites	0

TABLE 4.3 – Liste des fonctions codées pour la création et la mise à jour des fichiers.csv

# **Conclusion**

écrire la conclusion ici

# Bibliographie

- [1] Girond FLORIAN. "Mise en place d'un système d'information géographique pour la détection précoce et la prédition des épidémies de paludisme à Madagascar". Thèse de Doctorat en Géographie. Université de La Réunion, Faculté de Ecole Doctorale Lettres et Sciences Humaines, June 2017.
- [2] Olivier HAGOLLE et al. "A Multi-Temporal and Multi-Spectral Method to Estimate Aerosol Optical Thickness over Land, for Atmospheric Correction of FORMOSAT-2, VENuS, LANDSAT and SENTINEL-2 images". In: *Remote Sensing of Environment* 7 (Mar. 2015). ISSN: 2072-4292.
- [3] Olivier HAGOLLE et al. "A multi-temporal method for cloud detection, applied to FORMOSAT-2, VENuS, LANDSAT and SENTINEL-2 images". In: *Remote Sensing of Environment* (Mar. 2010).
- [4] Olivier HAGOLLE et al. "Correction of aerosol effects on multi-temporal images acquired with constant viewing angles : Application to Formosat-2 images". In: *Remote Sensing of Environment* 112 (Aug. 2007).
- [5] Marc LEROY et al. "MUSCATE : Operational Production Atmospheric Corrections and Monthly Composites Sentinel". 2016. URL: [http://seom.esa.int/S2forScience2014/files/03\\_S2forScience-MethodsII\\_LEROY.pdf](http://seom.esa.int/S2forScience2014/files/03_S2forScience-MethodsII_LEROY.pdf).
- [6] *Sen2cor Configuration and User Manual*. esa. Nov. 2016. URL: [step.esa.int/thirdparties/sen2cor/2.3.1/\[L2A-SUM\]S2-PDGS-MPC-L2A-SUM\[2.3.0\].pdf](http://step.esa.int/thirdparties/sen2cor/2.3.1/[L2A-SUM]S2-PDGS-MPC-L2A-SUM[2.3.0].pdf).

# Table des figures

2.1	Les différentes bandes spectrales au différentes résolutions des images sentinel-2 . . . . .	9
2.2	Comparaison du résultat de la correction des images avec Sen2cor, avec et sans la correction de la topographie. En rouge, les artefacts lors de l'ajout d'un MNT . . . . .	11
2.3	Zones d'étude choisies pour la comparaison des chaînes de traitement . . . . .	14
2.4	Résultats avec le masque issu de Sen2cor. En vert, des zones détectées comme étant des nuages (toits de maisons, sable). En rouge, les cirrus ne sont pas détectés. . . . .	15
2.5	Résultat avec le masque issu de MUSCATE. En vert, les ombres des nuages détectés et en rouge des erreurs. . . . .	15
2.6	Chaîne de traitement pour obtenir le masque de nuage utilisé. En sortie, les pixels portent la valeur 99 en présence de nuage et 0 sinon. . . . .	18
3.1	Organisation du dossier du projet. Le dossier DATA contenant la base de données peut être contenue dans le dossier du projet ou dans un dossier séparé. . . . .	20
3.2	Organisation du dossier DATA contenant la base de données requetable . . . . .	20
4.1	Format du fichier csv à créer . . . . .	23
4.2	Etapes pour créer l'image dont chaque pixel donne la valeur moyenne de NDVI sur 5km, en ne prenant pas en compte les nuages et les mers. . . . .	24
A.1	Gantt prévisionnel . . . . .	34
A.2	Gantt réel . . . . .	35
B.1	Masques obtenus avec les différentes chaines de traitement . . . . .	37
C.1	Algorithme mis en place pour le téléchargement, le traitement et le stockage des images Sentinel-2 déjà disponibles, codé en bash . . . . .	38
C.2	Fonction de traitement des images acquises. La fonction treatment2017() ressemble à treatment2016() mais l'architecture des dossiers est différente, codé en bash . . . . .	39
D.1	Algorithme mis en place pour créer et remplir le fichier .csv des séries temporelles d'un indice pour les semaines déjà disponibles, codé en Python . . . . .	41
D.2	Algorithme pour la mise à jour chaque semaine du fichier .csv des séries temporelles d'un indice, codé en Python . . . . .	42
E.1	Structure du fichier parametres.txt pour créer la base de données . . . . .	43
E.2	Structure du fichier parametres.txt pour créer le fichier .csv avec les séries temporelles des indices . . . . .	43
E.3	Organisation générale du dossier avec les codes, réalisé lors du stage . . . . .	44
E.4	Organisation générale de la base de données . . . . .	45

# Liste des tableaux

2.1	Caractéristiques des différentes missions d'observation de la surface de la Terre . . . . .	8
2.2	Poids des résultats en sortie des différentes chaînes de traitement . . . . .	14
2.3	Comparaison des chaînes de prétraitement . . . . .	16
4.1	Bilan des projections utilisées pour les différentes images et les shapefiles de la base de données . . . . .	25
4.2	Fichier et scripts Python pour la création et la mise à jour du fichier des séries temporelles de NDVI . . . . .	26
4.3	Liste des fonctions codées pour pour la création et la mise à jour des fichiers.csv . . . . .	27
D.1	Liste des librairies utilisées dans le script Python développé . . . . .	40

# Annexes

A	Déroulement du stage . . . . .	33
B	Comparaison des chaînes de prétraitement sur l'île de La Réunion . . . . .	36
C	Algorithmes de création de la base de données . . . . .	38
D	Algorithmes de création du fichier .csv des séries temporelles . . . . .	40
E	Organisation finale du dossier . . . . .	43

## **Annexe A**

### **Déroulement du stage**

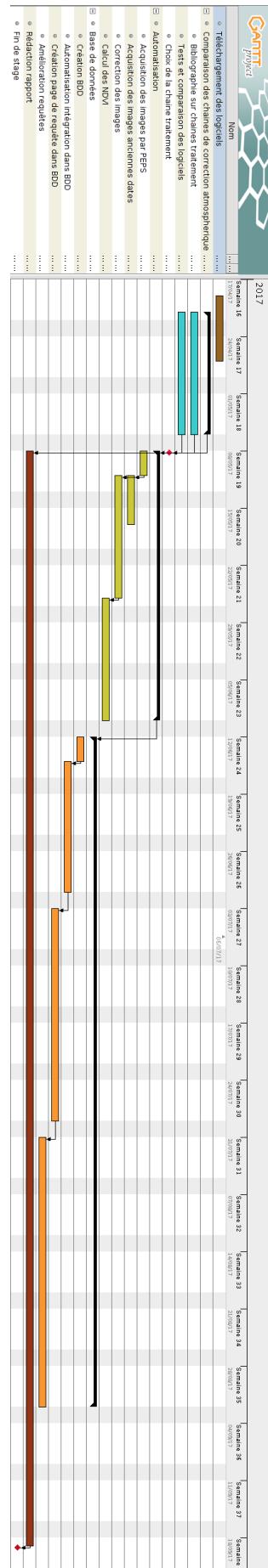


FIGURE A.1 – Gantt prévisionnel

Rapport de stage, Charlotte Wolff, Septembre 2017

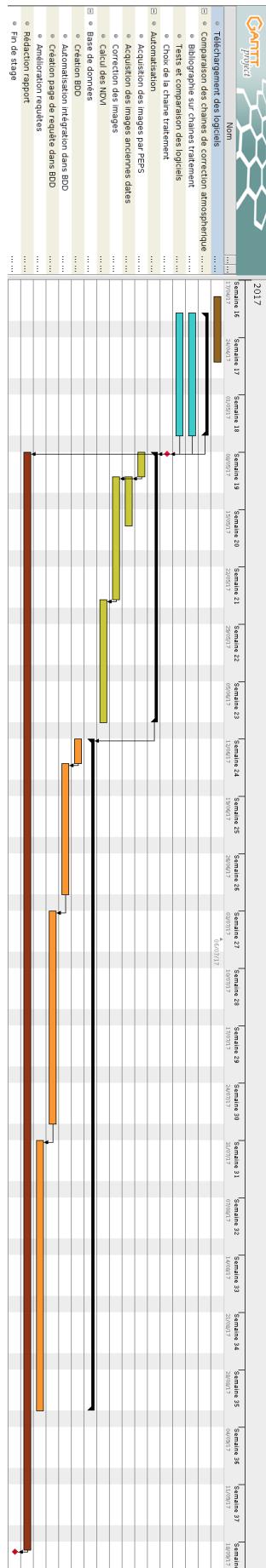


FIGURE A.2 – Gantt réel

Rapport de stage, Charlotte Wolff, Septembre 2017

## **Annexe B**

# **Comparaison des chaînes de prétraitement sur l'île de La Réunion**

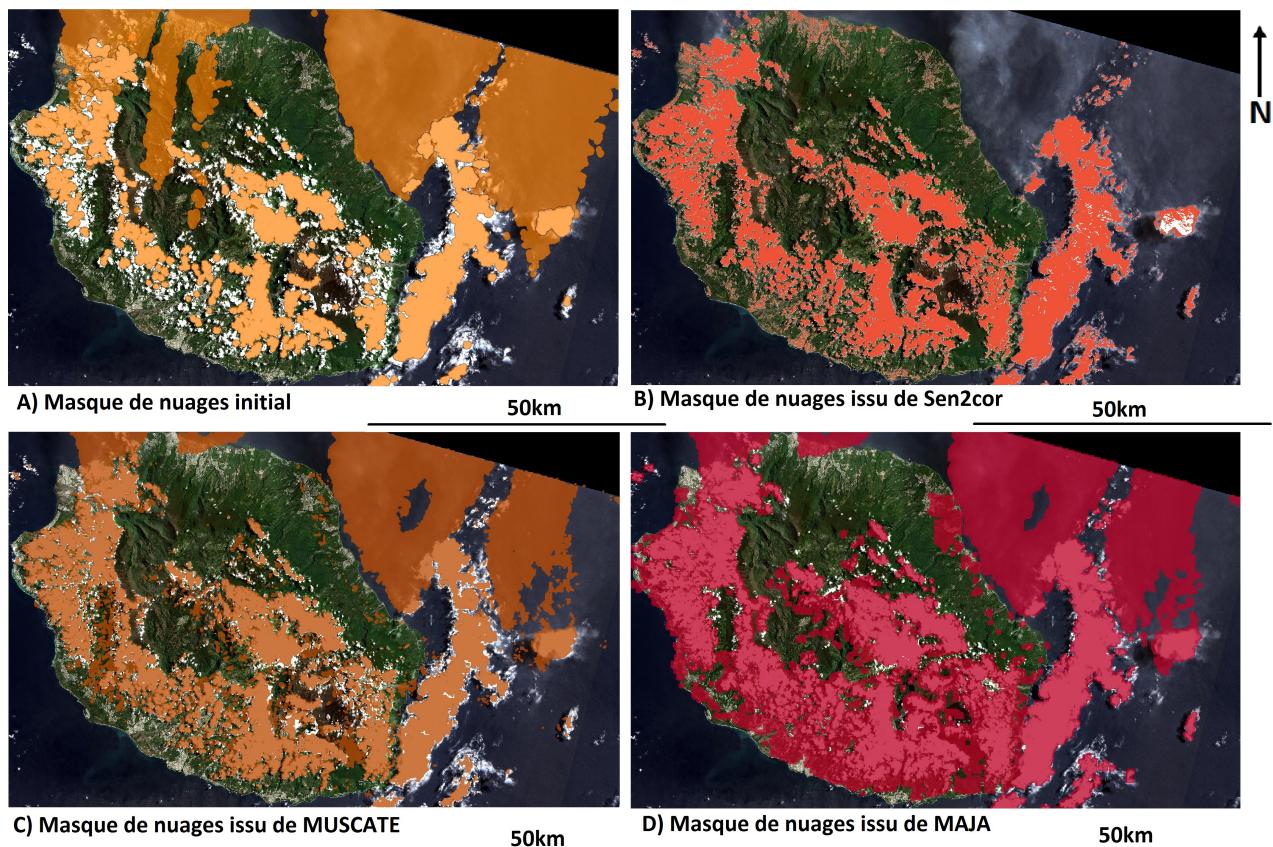


FIGURE B.1 – Masques obtenus avec les différentes chaînes de traitement

## Annexe C

# Algorithmes de création de la base de données

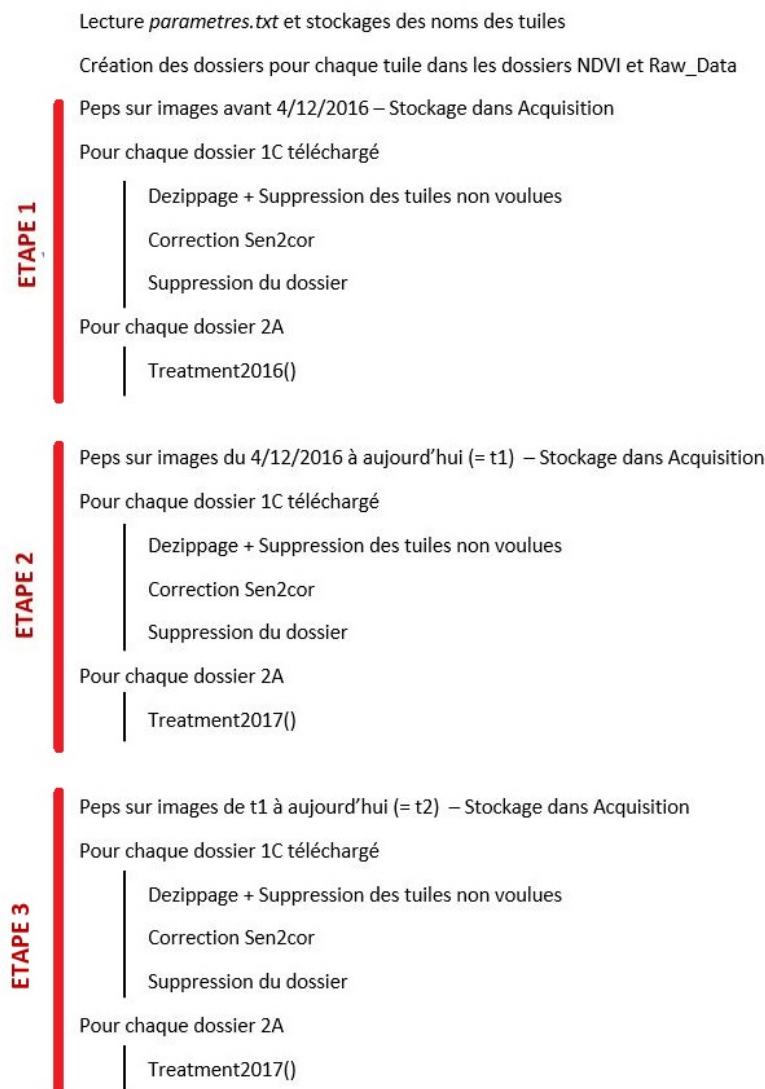


FIGURE C.1 – Algorithme mis en place pour le téléchargement, le traitement et le stockage des images Sentinel-2 déjà disponibles, codé en bash

Treatment2016()

- Recherche et copie des bandes et du masque de nuage initial
- Collage dans DATA/RawData/tuile/date
- Création du masque sen2cor et copie dans DATA/RawData/tuile/date → createMask.py
- Calcul de l'image NDVI et copié dans DATA/NDVI/tuile
- Suppression du dossier

FIGURE C.2 – Fonction de traitement des images acquises. La fonction treatment2017() ressemble à treatment2016() mais l'architecture des dossiers est différente, codé en bash

## Annexe D

# Algorithmes de création du fichier .csv des séries temporelles

Nom du module	Utilité
csv	Créer le fichier.csv et le lire
datetime	Faire les traitements sur les dates
gdal et ogr de osgeo	Faire des traitement sur les images
glob	Lister des fichiers
zonal_stats de rasterstat	Obtenir des statistiques (moyenne et écart-type) sur un shapefile

TABLE D.1 – Liste des librairies utilisées dans le script Python développé

```

Lecture du fichier des paramètres → readParams()
Pour chaque site épidémiologiques, création d'un shapefile → createBDDPoints()
Création du fichier csv et écriture de la première ligne avec le nom des sites
previousWeek = [dateDébut]

Pour chaque site épidémiologique :
    Liste des images NDVI sur la bonne tuile pour toutes les dates
    Pour chaque date de la liste des images :
        Si dateImage dans semaine épidémiologique ou semaine suivante :
            Récupération masque nuage → getCLDMask()
            Calcul du NDVI moyen 5km autour du site → find_indexName()
            previousWeek.append(valeur NDVI)

    Ecriture de la ligne dans le fichier csv
    Week = previousWeek
    dateSemaine = dateDébut + 7 jours

Tant que dateSemaine < date Fin :
    c = 1
    Week[0] = dateSemaine
    Pour chaque site épidémiologique :
        Liste des images NDVI sur la bonne tuile pour toutes les dates
        Pour chaque date de la liste des images :
            Si date == dateSemaine:      # C'est un lundi
                Récupération masque nuage → getCLDMask()
                Calcul du NDVI moyen 5km autour du site → find_indexName()
                Si valeur != None :      # Pas dans les nuages
                    Week[c] = valeur
            Si date > dateSemaine et dans même semaine :      # Dans semaine mais pas lundi
                Récupération masque nuage → getCLDMask()
                Calcul du NDVI moyen 5km autour du site → find_indexName()
                Si valeur != None :      # Pas dans les nuages
                    NDVI =  $\frac{\text{previousWeek}[c] * \text{jour_semaine} + (7 - n^{\circ}\text{jour_semaine}) * \text{valeur}}{7}$ 
                    Week[c] = NDVI
            c += 1
    previousWeek = Week
    dateSemaine += 7
    Ecriture de la ligne
Suppression des images _MEAN.tif créées → removeMeanFile()

```

FIGURE D.1 – Algorithme mis en place pour créer et remplir le fichier .csv des séries temporelles d'un indice pour les semaines déjà disponibles, codé en Python

Lecture du fichier des paramètres et récupération fichier .csv → `readParams()`

Lecture des 2 dernières lignes fichier .csv, ligne\_1 et ligne\_2

`datePreviousWeek = date ligne_1`

`dateWeek = datePreviousWeek + 7`

`ligne_0 = ligne_1`

`c = 0`

Pour chaque site épidémiologique :

```

    c += 1
    Liste des images NDVI sur la bonne tuile pour toutes les dates
    Pour chaque date de la liste des images :
        Si date == datePreviousWeek : # C'est un lundi
            Récupération masque nuage → getCLDMask()
            Calcul du NDVI moyen 5km autour du site → find_indexName()
            Si valeur != None :      # Pas dans les nuages
                ligne_1[c] = valeur
        Si date > datePreviousWeek et dans même semaine : # Dans la semaine mais pas lundi
            Récupération masque nuage → getCLDMask()
            Calcul du NDVI moyen 5km autour du site → find_indexName()
            Si valeur != None :      # Pas dans les nuages
                NDVI =  $\frac{(ligne_2[c] + n^{\circ}jour_{semaine} + (7 - n^{\circ}jour_{semaine}) * valeur)}{7}$ 
                ligne_1[c] = NDVI
    Pour chaque date de la liste des images :
        Si date == dateWeek : # C'est un lundi
            Récupération masque nuage → getCLDMask()
            Calcul du NDVI moyen 5km autour du site → find_indexName()
            Si valeur != None :      # Pas dans les nuages
                ligne_0[c] = valeur
        Si date > datePreviousWeek et dans même semaine : # Dans la semaine mais pas lundi
            Récupération masque nuage → getCLDMask()
            Calcul du NDVI moyen 5km autour du site → find_indexName()
            Si valeur != None :      # Pas dans les nuages
                NDVI =  $\frac{(ligne_1[c] + n^{\circ}jour_{semaine} + (7 - n^{\circ}jour_{semaine}) * valeur)}{7}$ 
                ligne_0[c] = NDVI

```

Mise à jour de la semaine précédente

Ajout de la nouvelle semaine

Suppression des images \_MEAN.tif créées → `removeMeanFile()`

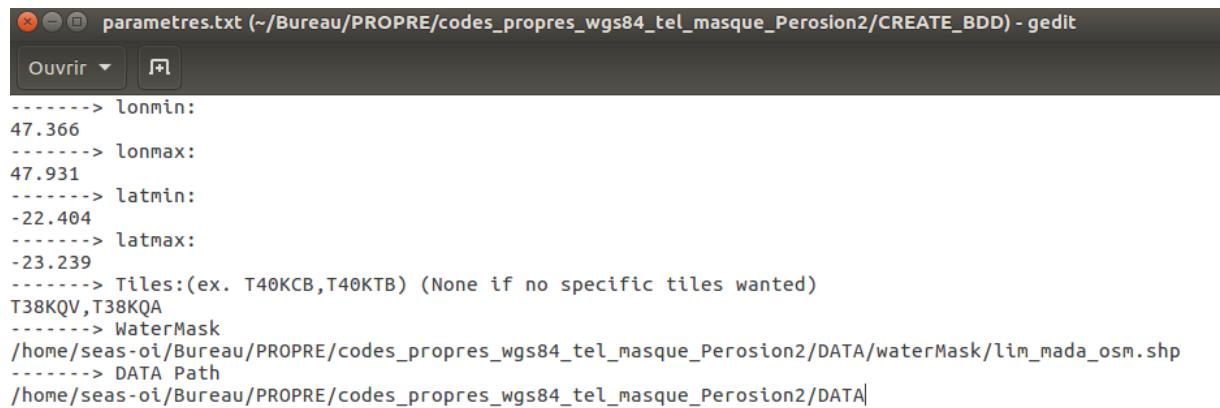
Suppression dernière ligne fichier .csv

Ecriture ligne\_1 et ligne\_0

FIGURE D.2 – Algorithme pour la mise à jour chaque semaine du fichier .csv des séries temporelles d'un indice, codé en Python

## Annexe E

# Organisation finale du dossier



```
parametres.txt (~/Bureau/PROPRE/codes_propres_wgs84_tel_masque_Perision2/CREATE_BDD) - gedit
Ouvrir ▾ Fichier
-----> lonmin:
47.366
-----> lonmax:
47.931
-----> latmin:
-22.404
-----> latmax:
-23.239
-----> Tiles:(ex. T40KCB,T40KTB) (None if no specific tiles wanted)
T38KQV,T38KQA
-----> WaterMask
/home/seas-oi/Bureau/PROPRE/codes_propres_wgs84_tel_masque_Perision2/DATA/waterMask/lim_mada_osm.shp
-----> DATA Path
/home/seas-oi/Bureau/PROPRE/codes_propres_wgs84_tel_masque_Perision2/DATA|
```

FIGURE E.1 – Structure du fichier parametres.txt pour créer la base de données



```
parametres.txt (chachou /media/charlotte/chachou/codes_propres_wgs84_tel_masque_Perision2/CREATE_CSV) - gedit
Ouvrir ▾ Fichier Enregistrer
-----> "Index name"
ndvi
-----> "Starting date"
20161219
-----> "Stopping date"
20170402
-----> "Path to DATA folder"
/home/seas-oi/Bureau/PROPRE/codes_propres_wgs84_tel_masque_Perision2/DATA
-----> "Path to out csv file"
/home/seas-oi/Bureau/PROPRE/codes_propres_wgs84_tel_masque_Perision2/DATA/serie_temporelles.csv
-----> "Path to points shapefile"
/home/seas-oi/Bureau/PROPRE/codes_propres_wgs84_tel_masque_Perision2/DATA/sites/sites_epil.shp
```

FIGURE E.2 – Structure du fichier parametres.txt pour créer le fichier .csv avec les séries temporelles des indices

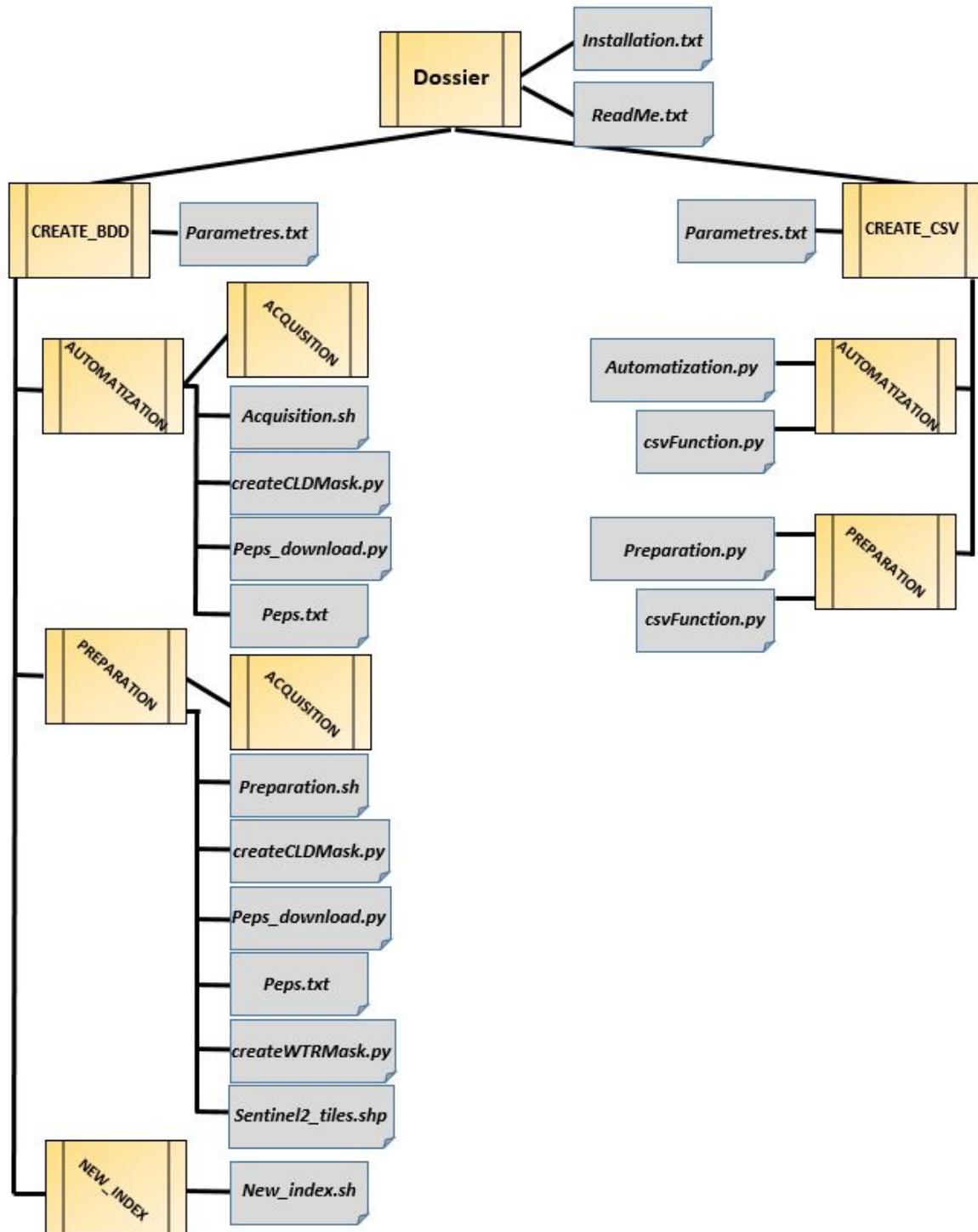


FIGURE E.3 – Organisation générale du dossier avec les codes, réalisé lors du stage

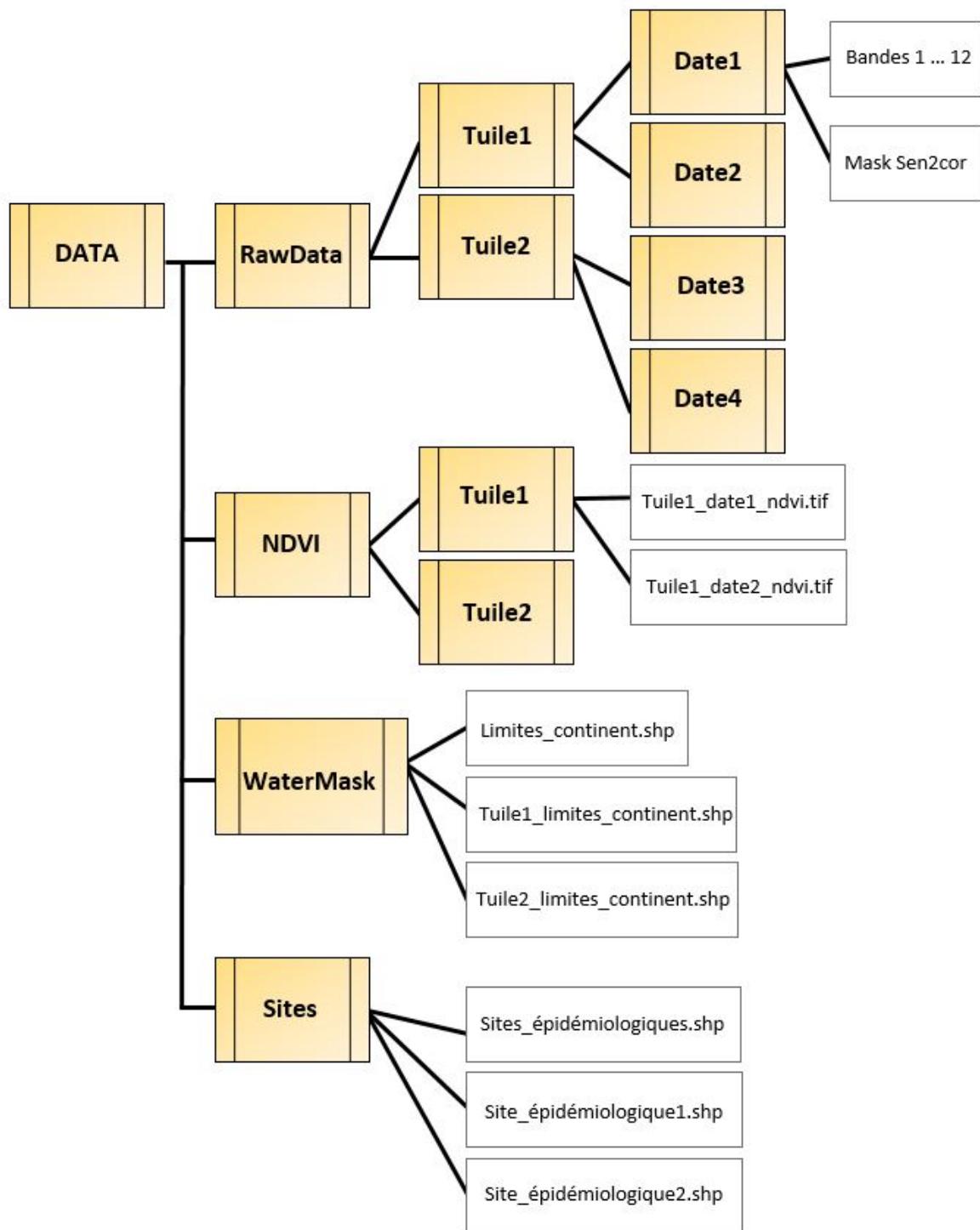


FIGURE E.4 – Organisation générale de la base de données