



**HAL**  
open science

# 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

## ► To cite this version:

Charlotte Wolff. 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. Traitement du signal et de l'image [eess.SP]. 2017. hal-04199716

HAL Id: hal-04199716

<https://hal.univ-reunion.fr/hal-04199716>

Submitted on 8 Sep 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

**ENSG**  
Géomatique

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 Révillion (Univ. Réunion - UMR Espace-Dev)

Non confidentiel    Confidentiel IGN    Confidentiel 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 :**

Projet S2-Malaria (APR 2017 CNES - TOSCA), UMR Espace-Dev, Institut Pasteur de Madagascar

**Encadrement de stage :**

Christophe Révillion (Univ. Réunion - UMR Espace-Dev) et Vincent Herbreteau (IRD - UMR Espace-Dev)

**Responsable pédagogique du cycle**

Jacques BEILIN, cycle Ingénieur, filière Photogrammétrie, Positionnement et Mesures de Déformations

**Tuteur du stage pluridisciplinaire :**

Marc Poupée, IGN/ENSG/DIAS

© ENSG

## Stage de fin d'étude du 17 avril au 17 septembre 2017

**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 :** 61 pages dont 4 d'annexes

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

**Modifications :**

EDITION	REVISION	DATE	PAGES MODIFIEES
1	0	08/2017	Création

# Remerciements

Je tiens tout particulièrement à remercier mes maîtres de stage, Christophe Révillion et Vincent Herbreteau pour leur disponibilité et tout ce que j'ai appris à leurs côtés, ainsi que l'équipe de l'UMR Espace-DEV, à la station SEAS-OI de Saint-Pierre de la Réunion, qui m'a accueillie à Saint-Pierre de La Réunion.

Un grand merci également à Fanjasoa Rakotomanana et Bienvenue Rahoilijaona de l'Institut Pasteur de Madagascar qui m'ont accueillie à Antananarivo et m'ont aidé à définir les besoins et les contraintes à respecter.

Enfin, je remercie Florian Girond pour ses nombreux conseils et le temps qu'il a pris pour répondre à mes questions.

## Résumé

Dans des pays africains en voie de développement, tels Madagascar ou l'Afrique du Sud, le taux de mortalité lié aux maladies infectieuses est en constante diminution. Pourtant, le paludisme persiste. Pour gérer les épisodes de pics épidémiques, les autorités sanitaires développent des systèmes de surveillance basés sur l'acquisition en temps réel de données sur les cas de paludisme recensés par des agents présents sur le terrain et corrélées à des données environnementales acquises par satellite. L'Institut Pasteur de Madagascar a développé notamment un système de détection précoce et de prédiction des épidémies de paludisme à partir du système de surveillance sentinelle des maladies à potentiel épidémique et de données environnementales à basse résolution. Les données Sentinel-2, dont la résolution atteint 10m et dont la fréquence d'acquisition est de 5 jours, permettrait de compléter cette application.

Mon stage de fin d'études réalisé avec l'UMR Espace-Dev, au sein des locaux de 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 qui restent suffisants pour l'utilisation que l'on veut en faire, d'autant que la chaîne de traitement Sen2cor est plus facile à prendre en main et ne nécessite pas de profondeur historique des images pour fonctionner, comme c'est le cas avec MAJA et MUSCATE.

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 à leur tour dans la base de données.

Ces indices sont 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, capitale de Madagascar.

Puis dans un deuxième temps, les images des indices ont été intégrées au catalogue en ligne du SEAS-OI pour pouvoir dans le futur télécharger à distance les images traitées si besoin.

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

**Mots clés :** Télédétection, Images multi-spectrales, Sentinel-2, Sen2cor, surveillance, paludisme

## Résumé

In African developing countries such as Madagascar and South Africa, fatality rate due to infectious diseases constantly reduces. However, malaria persists. To handle epidemiologic pic periods, health authorities develop monitoring systems based on real-time acquisition of malaria cases in the field, which are correlated with satellite environmental data. The Pasteur Institute of Madagascar developed an early warning detection system for malaria with the help of environmental information and data from healthcare providers in the field. Sentinel-2 data, whose resolution reaches 10m and acquisition frequency 5 days, can be an effective support to this application.

My end-of-course 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 and compared processing tools are Sen2cor, MAJA and MUSCATE. Sen2cor is less efficient than MAJA and MUSCATE but sufficient enough for our use. Furthermore Sen2cor is easier to use and does not need images from the past to work as it is the case for MAJA or MUSCATE.

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

These index are used to create time series files that the Pasteur Institute of Madagascar can easily use to predict malaria peaks throughout Madagascar. The expected results have been discussed during a trip in Antananarivo (Madagascar). In a second step, the index images have been integrated in the online catalog of SEAS-OI. The aim is to be able to download images when needed.

This automatization was done in a Linux environment, with scripts written in Python and bash (Linux). Libraries such as Orpheo ToolBox and GDAL were used.

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

# Table des matières

<b>Glossaire et sigles utiles</b>	<b>4</b>
<b>Introduction</b>	<b>7</b>
<b>1 Chapitre 1</b>	<b>8</b>
1.1 Les objectifs du stage . . . . .	8
1.2 Les organismes d'accueil . . . . .	10
1.3 La gestion de projet . . . . .	11
<b>2 Chapitre 2</b>	<b>12</b>
2.1 Caractéristiques des images Sentinel-2 . . . . .	12
2.2 Acquisition des images Sentinel-2 . . . . .	13
2.3 Logiciels de prétraitement des images . . . . .	14
2.4 Comparaison et choix de la chaîne de prétraitement . . . . .	17
2.5 Création du masque des nuages . . . . .	22
<b>3 Chapitre 3</b>	<b>24</b>
3.1 Organisation de la base de données . . . . .	24
3.2 Acquisition et traitement des données déjà disponibles . . . . .	26
3.3 Automatisation pour l'acquisition des données futures . . . . .	27
3.4 Ajout de nouveaux indices . . . . .	27
<b>4 Chapitre 4</b>	<b>29</b>
4.1 Format des séries temporelles pour répondre aux besoins de l'IPM . . . . .	29
4.2 Développement du code . . . . .	31
4.3 Résultats et comparaison avec les données MODIS . . . . .	34
4.4 Produits complémentaires . . . . .	35
<b>Conclusion</b>	<b>38</b>
<b>A Présentation de l'application web développée par Florian Girond</b>	<b>44</b>
<b>B Déroulement du stage</b>	<b>47</b>
<b>C Organisation du dossier créé par le traitement de Sen2cor</b>	<b>49</b>
<b>D Comparaison des chaînes de prétraitement sur l'île de La Réunion</b>	<b>51</b>
<b>E Algorithmes de création de la base de données</b>	<b>53</b>
<b>F Algorithmes de création du fichier .csv des séries temporelles</b>	<b>55</b>
<b>G Organisation finale du dossier</b>	<b>58</b>

**H Algorithmes de création des images Quicklook pour l'intégration au catalogue Seas-Oi 61**

# Glossaire et sigles utiles

**CENTOS** Community ENTERprise Operating System, système d'exploitation GNU/Linux basé sur la distribution RHEL (Red Hat Enterprise Linux)

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

**CNES** Centre National d'Etudes Spatiales

**Cron** Programme Linux permettant l'exécution automatiquement de scripts et de commandes à une date et une heure précise

**CSV** Comma-separated values, format informatique ouvert dont les données tabulaires sont des valeurs séparées par des virgules

**DLR** Deutsche Zentrum für Luft- und Raumfahrt, centre allemand pour l'aéronautique et l'astronautique

**ENSG** École Nationale des Sciences Géographiques

**ESA** European Space Agency

**GDAL** Geospatial Data Abstraction Library, bibliothèque libre pour la lecture et le traitement d'images géographiques

**GeoTIFF** Format d'image donnant des informations de géoréférencement à une image TIFF, portant l'extension .tiff

**IPM** Institut Pasteur de Madagascar

**JPEG 2000** Norme de compression d'images produite par Joint Photographic Experts Group, portant l'extension .jp2

**LANDSAT** Premier programme d'observation de la Terre à des fins civiles, développé par la NASA

**MNT** Modèle numérique de terrain

**MODIS** Moderate-Resolution Imaging Spectroradiometer, série de satellites d'observation de la Terre, lancé par la NASA

**NASA** National Aeronautics and Space Administration, agence gouvernementale américaine en charge du programme spatial civil américain

**NDVI** Normalized Difference Vegetation Index

**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 Théia

**NDVI** Normalized Difference Vegetation Index, indice de végétation souvent utilisé dans les études des études environnementales

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

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

**QGIS** Logiciel SIG libre multiplate-forme

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

**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

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

**SIG** Système d'Information Géographique

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

**SPOT** Satellite Pour l'Observation de la Terre, série de satellites de télédétection développée par le CNES

**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

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

**UBUNTU** Système d'exploitation GNU/Linux basé sur la distribution Linux Debian

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

**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

# Introduction

Le programme Copernicus est un programme développé par l'ESA (European Space Agency) qui a permis notamment de lancer deux satellites d'observation de la Terre, Sentinel-2 A (2015) et Sentinel-2 B (2017). L'objectif de ces satellites est de fournir des images à une très bonne résolution spatiale (10 mètres) et à une fréquence élevée (tous les cinq jours) dans le monde entier. Ceci représente un gain important par rapport aux images MODIS ou LANDSAT par exemple. Les images Sentinel-2 ont pour vocation à être utilisées dans de vastes domaines d'application, dont celui de la santé notamment.

Le stage que j'ai réalisé au sein de l'UMR Espace-Dev, à la station SEAS-OI à Saint-Pierre de la Réunion devait ainsi répondre à deux principaux objectifs, à savoir tester et comparer différentes chaînes d'acquisition et de traitement des images Sentinel-2 d'une part et créer des séries temporelles d'indices -de végétation pour commencer- à partir de ces images d'autre part. Ces séries, combinées à d'autres paramètres, devront aider des pays en voie de développement tels que Madagascar et l'Afrique du Sud à prédire des pics de paludisme.

Ce rapport présente dans un premier temps le contexte et le déroulement du stage. Puis les résultats obtenus avec les trois principales chaînes de traitement testées que sont Sen2cor, MAJA et MUSCATE sont décrits. Enfin, la base de données et les séries temporelles obtenues sont présentées ainsi que les algorithmes et les scripts utilisés pour les créer.

# Chapitre 1

## Déroulement du stage

### 1.1 Les objectifs du stage

**Le projet Sentinel-2 Malaria** Le stage réalisé se déroule dans le cadre du projet Sentinel-2 for Malaria Surveillance, financé par le CNES (Appel à Projets de Recherches 2017, TOSCA). 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 : Madagascar et l'Afrique du Sud. A Madagascar, le paludisme reste une importante cause de morbidité et de mortalité. En Afrique du Sud notamment, le taux de mortalité lié aux maladies infectieuses a considérablement diminué (>75%) depuis 2000 mais le paludisme persiste, avec des réintroductions régulières dans les zones où l'environnement y est favorable. [9] [11]

Ce projet s'inscrit dans la continuité des travaux de thèse de Florian Girond [2] à l'Institut Pasteur de Madagascar. Elle portait sur le développement d'un outil de détection précoce de paludisme à Madagascar. L'application a pour objectif de signaler quelques semaines à l'avance, 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 :

- Les notifications de cas de paludisme envoyés par SMS par un réseau "*sentinelle*" de médecins à travers le pays,
- L'intégration en temps quasi-réel d'indicateurs favorables au développement du paludisme (indices de végétation NDVI issus d'images MODIS, estimation de la pluviométrie à partir d'images satellitaires, etc). En effet, on sait que des pics de paludisme apparaissent dans certaines régions d'Afrique quelques semaines après des pluies importantes, temps pendant lequel les moustiques, vecteurs du paludisme, se développent. [1]

Ces informations sont stockées dans une base de données. (Annexe A)

Ce stage a pour but d'avancer sur l'intégration d'indicateurs favorables au développement du paludisme, en alimentant le système de surveillance du paludisme de l'IPM avec des séries temporelles d'indicateurs issues des images multispectrales Sentinel-2. Pendant 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) s'il y a présence d'un nuage,
- Le stockages des images multispectrales, des images d'indices et du masque de nuages dans une base de données sur laquelle on pourra ensuite faire des requêtes spatio-temporelles pour créer des séries temporelles.

L'enjeu de l'automatisation est de simplifier l'utilisation des données satellitaires Sentinel-2 dans les pays en voie de développement comme Madagascar. En effet, ces pays n'ont pas toujours les ressources humaines et les moyens techniques (limitation des bandes passantes, mémoire de stockage restreinte, ...) nécessaires à l'utilisation des données satellitaires. C'est pourquoi la base de données est externalisée sur l'île de La Réunion et seuls des fichiers de séries temporelles sont envoyés.

L'application développée par Florian Girond utilise déjà des séries temporelles d'indices de végétation issues du système américain MODIS (Moderate-Resolution Imaging Spectroradiometer), mis à disposition par l'IRI via le site : <http://iri.columbia.edu> [8]. Le projet Sentinel-2 Malaria a également pour vocation de voir en quoi les images Sentinel-2 peuvent améliorer ces séries temporelles. Il s'agit d'un projet sur deux ans. (Figure 1.1)

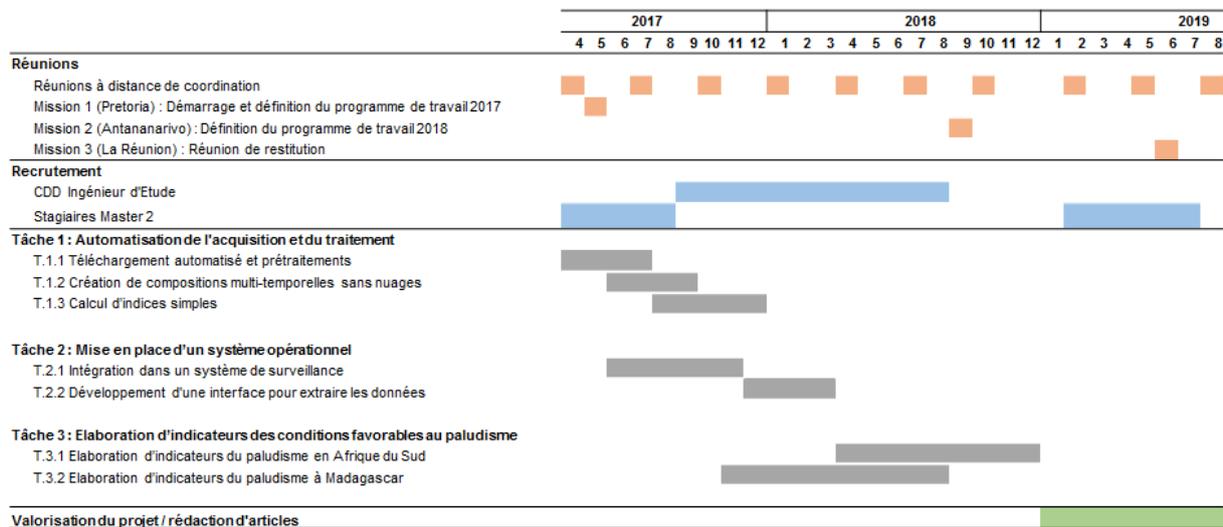


FIGURE 1.1 – Déroutement du projet Tosca Sentinel-2 sur deux ans

**Les images Sentinel-2 et les chaînes de traitement associées** Les images Sentinel-2 sont des produits récents et de nouvelles chaînes de traitement 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 utiliser dans le futur, pour quelles applications et pour connaître les avantages et inconvénients de chaque chaîne. Les chaînes testées sont Sen2cor, MAJA et MUSCATE.

## 1.2 Les organismes d'accueil

**SEAS-OI** (Surveillance de l'Environnement Assistée par Satellite pour L'Océan Indien : <http://www.seas-oi.org>) Le stage s'est déroulé dans les locaux de la station SEAS-OI à Saint-Pierre de La Réunion. Il s'agit d'un centre de compétences en télédétection et d'une plate-forme d'acquisition et de traitement d'images satellitaires, qui couvre toutes les îles du sud-ouest de l'Océan Indien. Elle a été mise en service en juillet 2012 avec quatre tutelles : l'Université de La Réunion, l'Institut de Recherche pour le Développement, la Région Réunion et L'État.

**L'UMR Espace-Dev** (IRD, Univ. Antilles, Univ. Guyane, Univ. Montpellier, Univ. Réunion : <http://www.espace-dev.fr>) Les encadrants du stage sont Christophe Révillion (Univ. Réunion) et Vincent Herbreteau (IRD) 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 face aux changements globaux.

**L'Institut Pasteur de Madagascar** (<http://www.pasteur.mg>) L'IPM est un établissement scientifique privé malgache à bon non lucratif et reconnu d'utilité publique du Réseau international des instituts Pasteur, placé sous le haut patronage du gouvernement de la République malgache. 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. L'IPM a mis en place une cellule SIG au sein de son unité d'Épidémiologie afin de réaliser des analyses spatiales et de prendre en compte les composantes géographiques et environnementales de épidémiologie des maladies suivies.

Afin de prévoir l'utilisation de mes travaux par la Cellule SIG de l'IPM, j'ai eu l'occasion d'aller en mission à Antananarivo et de rencontrer les membres de l'IPM, du 11 au 15 juin 2017. Cette mission a été l'occasion de rencontrer les membres de la cellule SIG de l'IPM et de voir comment fonctionne le système d'alerte précoce développé par Florian Girond. Les différentes réunions ont permis de mieux cibler leurs besoins et de définir la forme des livrables attendus.

Ainsi, leur débit internet ne permettant pas le téléchargement d'images satellitaires brutes Sentinel-2, le livrable doit être un fichier .csv avec les séries temporelles des NDVI, pour différents sites épidémiologiques. Ces sites sont fournis dans un fichier shapefile. Le fichier .csv, beaucoup plus léger, pourra être facilement téléchargeable par l'IPM pour l'insérer dans la base de données du système de surveillance Sentinelle.

## 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 arrivée à la station SEAS-OI, 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 ont été abordées tout au long du stage. Suite à cela, j'ai réalisé un diagramme de Gantt prévisionnel du stage.

Puis lors du stage, j'ai réalisé un carnet de bord permettant d'avoir une vue globale de l'avancement du stage et de réaliser en fin de stage un Gantt réel du déroulement du stage. (Annexe [B](#))

## 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 temporelle. (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	10 m	2,5 m	30 m
Taille des tuiles	290 km	60 km	185 km
Nombre de bandes	13	4	6
Type d'acquisition	Systématique	Programmation	Programmation

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

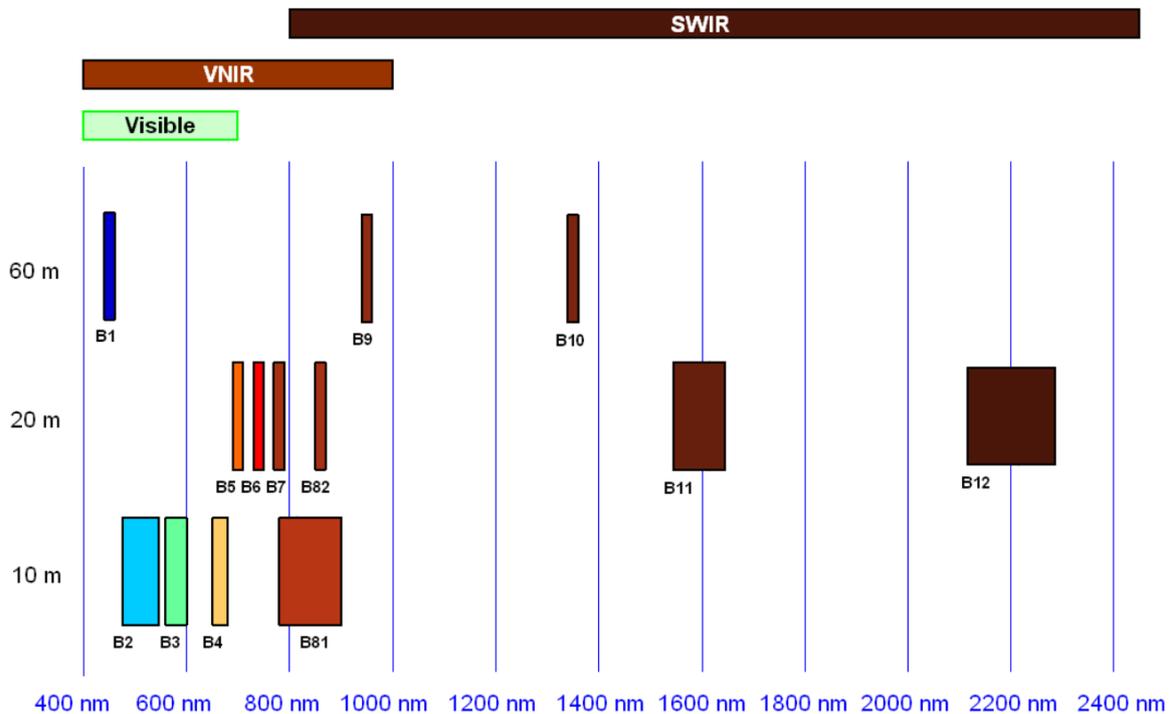


FIGURE 2.1 – Bandes spectrales et résolutions spatiales des images Sentinel-2

## 2.2 Acquisition des images Sentinel-2

Les images Sentinel-2 de niveau 2A (images orthorectifiées) sont disponibles gratuitement sur plusieurs plateformes de diffusion dont celle de l'ESA, celle du CNES (plateforme d'Exploitation des Produits Sentinel - PEPS) et le cloud d'Amazon (Amazon Web Service). Le site : <https://sentinel.esa.int/web/sentinel/missions/sentinel-2/acquisition-plans> permet d'accéder au planning des acquisitions de Sentinel-2 semaine après semaine, 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 :

- L'emprise (latitudes et longitudes) 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 ligne de commande télécharge les images Sentinel-2 de l'île 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 de La Réunion. Cependant, certaines tuiles ne sont pas intéressantes. Lors du stage, j'ai donc modifié ce code Python, 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, car avant, le format du dossier téléchargé est différent.

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 que le dossier téléchargé 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). Cette première étape est importante pour permettre la comparaison des images. Le prétraitement génère également un masque des nuages, qui permettra d'extraire et ne pas prendre en compte les zones nuageuses 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 (Deutsches Zentrum für Luft- und Raumfahrt),
- **MUSCATE**, développé par le CNES et le CESBIO dans le cadre du pôle Théia (Le pôle de données et de services surfaces continentales Théia : <https://www.theia-land.fr/fr>).

### 2.3.1 Sen2cor [10]

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, nuages, 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. (Annexe C)

**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 détecter et enlever les cirrus des images.

**La correction topographique** Sen2cor permet également de corriger les effets de la topographie qui 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. Les résultats obtenus avec et sans la correction de la topographie sur les images de La Réunion ont été comparées. Les résultats obtenus avec la correction topographique ne sont pas satisfaisants : On constate que l'ajout du MNT entraîne l'apparition d'artefacts au niveau des vallées. (Figure 2.2)

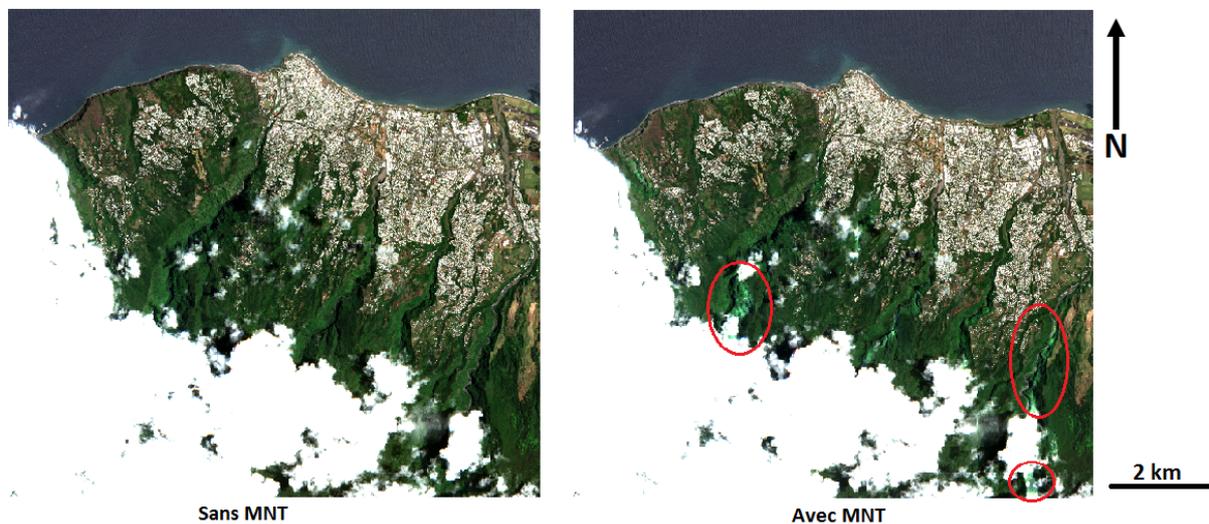


FIGURE 2.2 – Comparaison du résultat de la correction des images avec la chaîne de traitement Sen2cor, avec et sans la correction de la topographie. En rouge, les artefacts lors de l'ajout d'un MNT

### 2.3.2 MAJA

La chaîne de traitement MAJA, développée par le CNES, CESBIO et le DLR, est s'applique à toutes les images satellitaires, pas uniquement les images Sentinel. Elle ne peut pas être utilisée sur UBUNTU. Pour l'utiliser, j'ai dû installer sur l'ordinateur une machine virtuelle avec le système d'exploitation 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 [4]** 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 [5]** 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-temporelle, l'image à corriger et une image composite, obtenue avec l'ensemble des images précédentes, sont comparées. 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, un modèle atmosphérique dont il faut trouver les paramètres est utilisé. La méthode du filtre de Kalman est utilisée 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. Les moindres carrés sont utilisé 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, c'est-à-dire avec 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 l'image à corriger.

### 2.3.3 MUSCATE [7]

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 du Pole Théia distribue ensuite les images corrigées sur son site (<https://theia.cnes.fr>). Nous n'avons donc pas accès à la chaîne de traitement pour la modifier. Un code Python développé par Olivier Hagolle au CESBIO permet de récupérer directement les images sur le site pour les dates, l'emprise et les tuiles 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 le suivant : 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, leur traitement et l'intégration sur le site.

## 2.4 Comparaison et choix de la chaîne de prétraitement

Lors de mon stage, j'ai téléchargé un ensemble de jeux 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 nuages créés : La Réunion et 3 zones sur Madagascar (Figure 2.3).

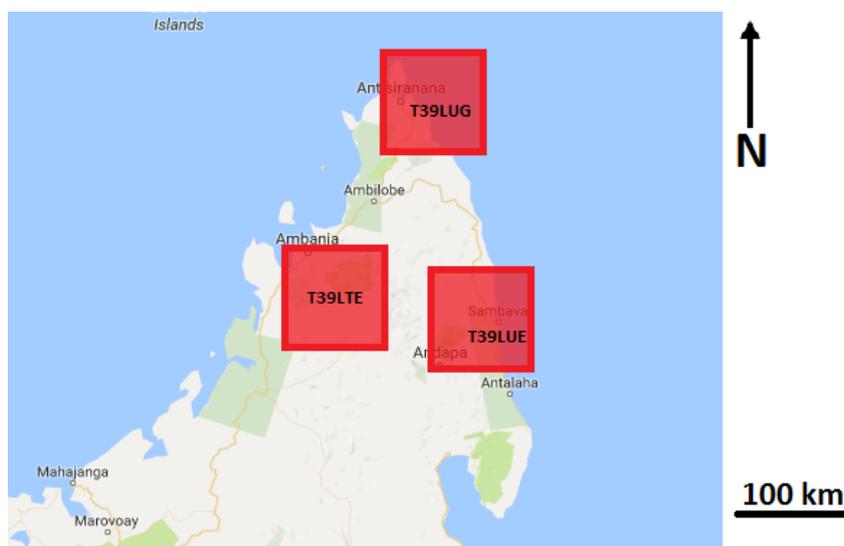


FIGURE 2.3 – Zones d'étude choisies pour la comparaison des chaînes de traitement

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 nuages initial fourni lors du téléchargement de l'image Sentinel-2. Des dates avec peu de nuages ont été

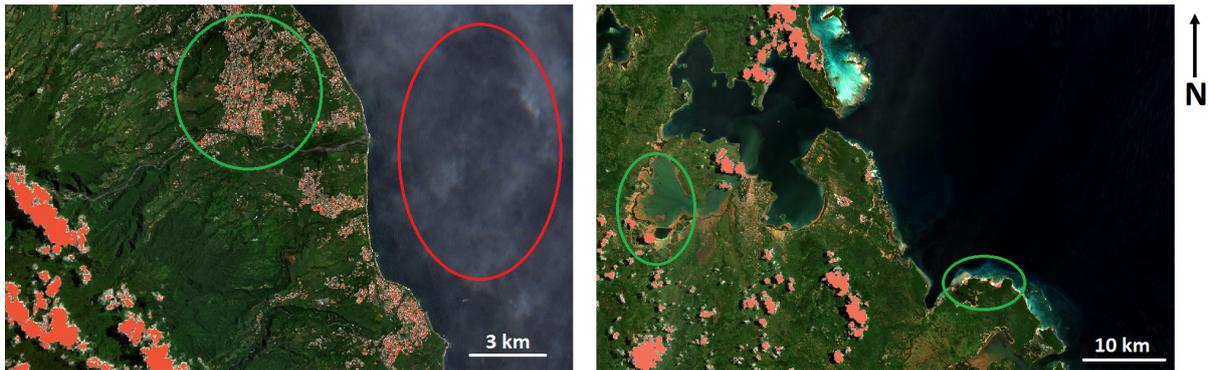


FIGURE 2.4 – Résultats avec le masque issu de Sen2cor. En vert, des zones détectées comme étant des nuages (toits de maisons, sable). En rouge, les cirrus ne sont pas détectés.

choisies pour avoir des informations sur les corrections atmosphériques qui aient un sens. Les masques de nuages ainsi que les résultats des corrections atmosphériques ont été comparés.

Chaîne de traitement	Non-traité	Sentinel-2	MAJA	MUSCATE
<b>Poids</b>	800 Mo	1.4 Go	3.6 Go	3.6 Go
<b>Poids d'une image spectrale</b>	[2-70]Mo	[3-200]Mo	environ 70 Mo	environ 70 Mo
<b>Poids du masque nuages</b>	90 ko	600 ko	120 Mo	120 Mo

TABLE 2.2 – Poids des résultats en sortie des différentes chaînes de traitement. L'essentiel du poids de chaque dossier est porté par les images des bandes spectrales

### 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 correctement. (Figure 2.4)

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 :

- Éroder légèrement le masque des nuages pour enlever une parties des toits et des plages,
- Dilater ensuite le masque des nuages pour qu'il prenne en compte les nuages en entier. (Figure 2.5)

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

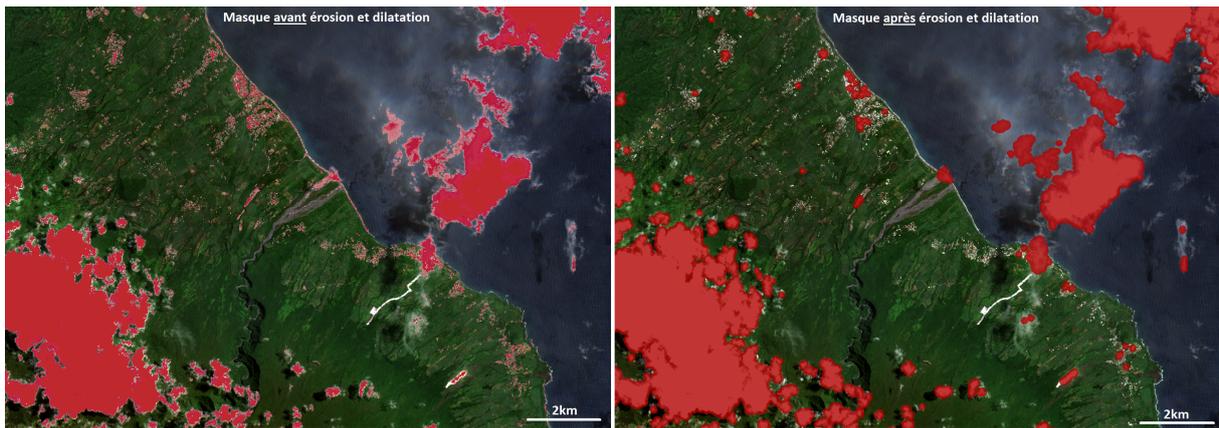


FIGURE 2.5 – Comparaison du masque issu de Sen2cor, avant et après traitement d'érosion et de dilatation. L'érosion permet d'éliminer une partie des bâtiments et des plages et la dilatation d'obtenir un masque englobant entièrement les nuages.

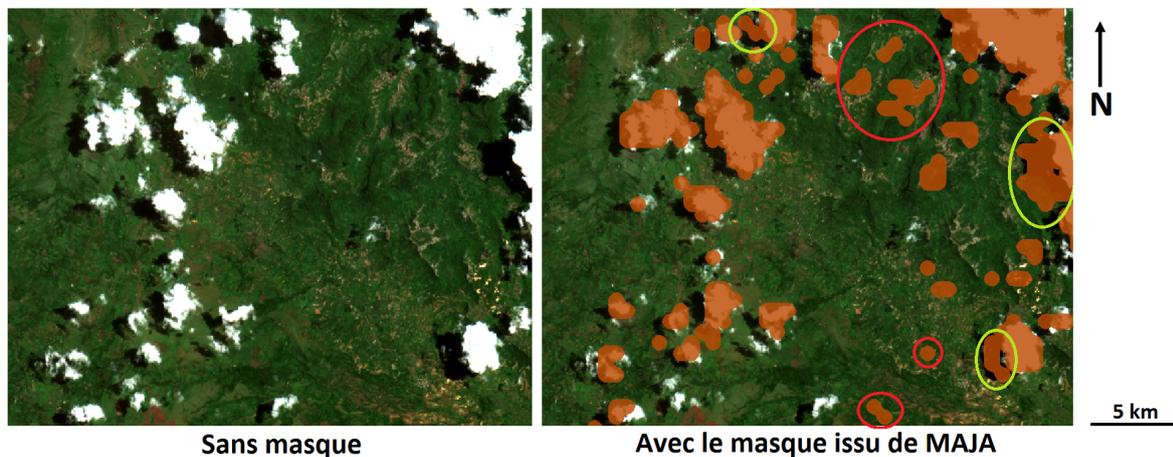


FIGURE 2.6 – Résultat avec le masque issu de MUSCATE. En vert, les ombres des nuages détectés et en rouge des erreurs.

**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 2.6)

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

	<b>Avantages</b>	<b>Inconvénients</b>
Sen2cor	-Détecte tous les nuages -Fonctionne sur tous les systèmes d'exploitation Linux -Pas besoin de profondeur historique	-Ne détecte pas les ombres des nuages -Masque les toits clairs, les plages et zones d'eau -Ne détecte pas les cirrus
MAJA	-Détecte les cirrus -Englobe entièrement les nuages détectés	-Ne fonctionne pas sur UBUNTU -Nécessite une base de données préalable pour la profondeur historique -Préparation de fichiers .txt en amont
MUSCATE	-Détecte les cirrus -Détecte les ombres des nuages -Masque similaire à MAJA mais moins large	-Pas d'accès au code -Données non accessibles le jour de l'acquisition -Détecte des zones fausses

TABLE 2.3 – Comparaison des chaînes de prétraitement

## 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 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.

Les images NDVI sur La Réunion et les 3 zones de Madagascar ont été calculées à partir :

- Des bandes avant correction atmosphérique
- Des bandes après correction avec Sen2cor
- Des bandes après correction avec MAJA
- Des bandes après correction avec MUSCATE

Puis l'histogramme des valeurs des pixels a été tracé en utilisant QGIS pour pouvoir comparer les résultats. (Figure 2.7)

Sur la zone au niveau de La Réunion, les histogrammes des valeurs de pixels sont similaires pour les 3 chaînes de traitement.

Suite à la comparaison des chaînes de traitement, la chaîne Sen2cor a été choisie pour l'automatisation des traitements. En effet, le déploiement de cette chaîne de traitement est plus simple que les deux autres chaînes, les images facilement disponibles et ne demandant pas de préparation en amont pour son utilisation. Les résultats obtenus, que ce soit en terme de masque de nuages ou de correction atmosphérique, sont satisfaisant pour l'utilisation et l'intégration dans le système de surveillance épidémiologique.

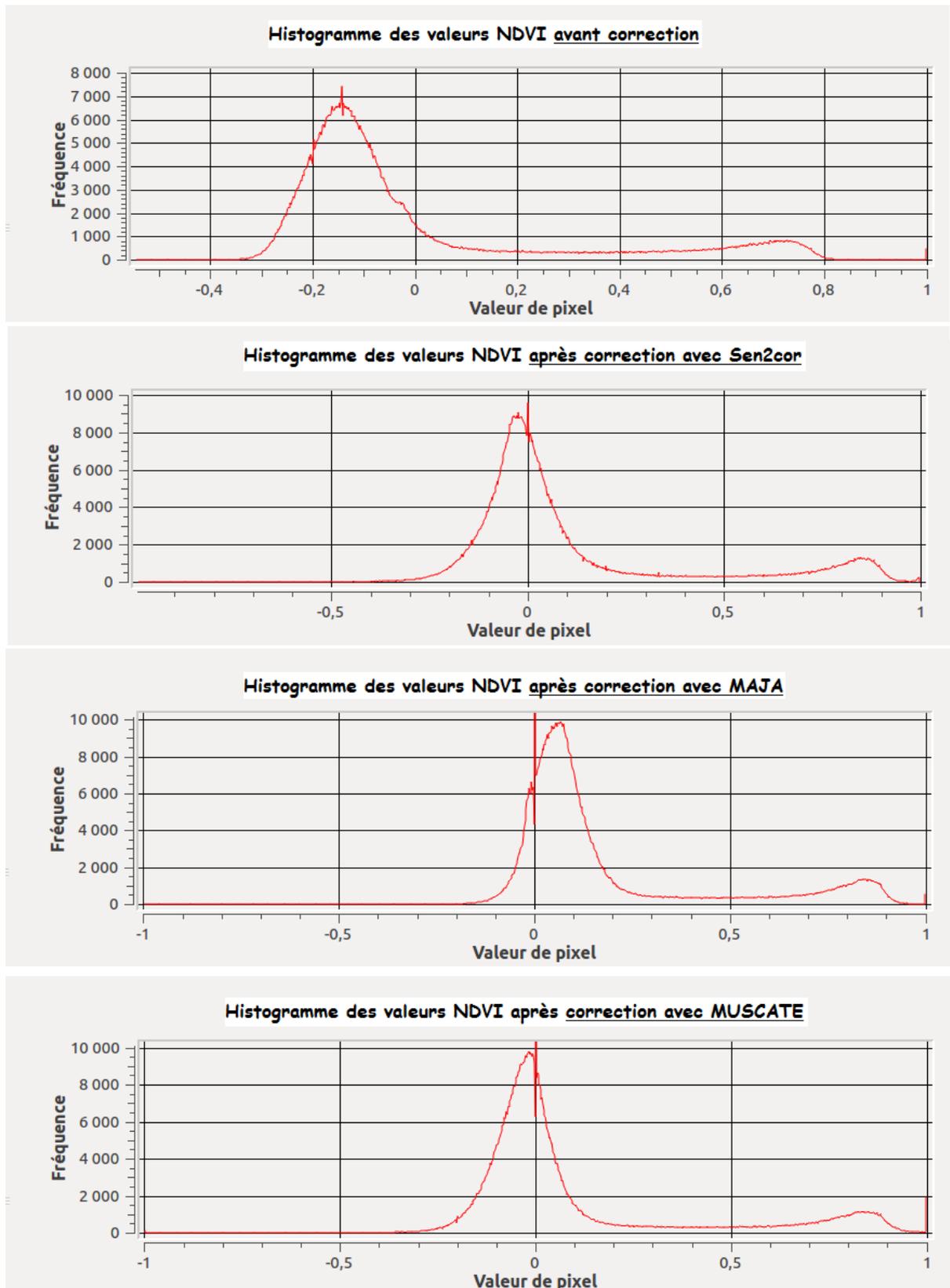


FIGURE 2.7 – Comparaison des histogrammes des valeurs des pixels sur les images NDVI au dessus de La Réunion, avant traitement et avec les trois chaînes de traitement testées

## 2.5 Création du masque des nuages

Nous avons donc choisi d'utiliser le masque de nuages fourni en sortie de la chaîne de traitement de Sen2cor. En utilisant QGIS, j'ai ensuite réalisé des tests pour améliorer le masque des nuages et obtenir finalement un fichier raster qui pour chaque pixel de l'image donne l'information sur la présence ou l'absence d'un nuage :

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

Les objectifs du traitement des masques de nuages est de diminuer le nombre d'artefacts (zones d'habitations et plages) pris pour des nuages grâce à un processus d'érosion puis d'obtenir des masques qui englobent entièrement les nuages grâce à un processus de dilatation.

L'algorithme de traitement pour obtenir ce raster est présenté dans la Figure 2.8. Cet algorithme a été automatisé avec une fonction Python que j'ai nommé *createCLDMask.py*.

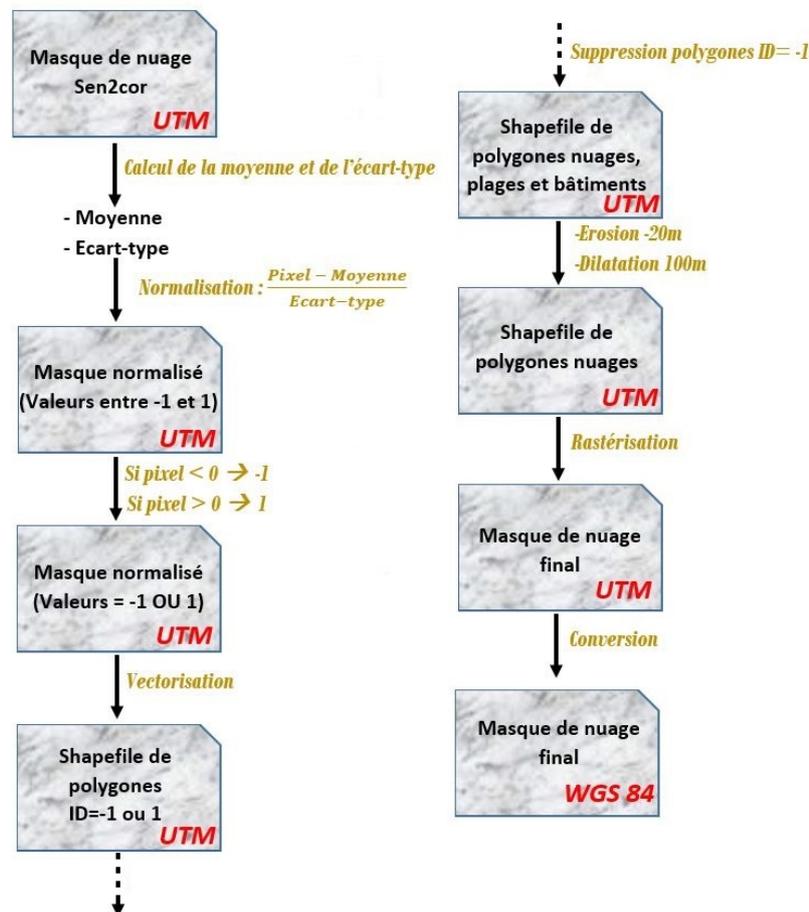


FIGURE 2.8 – Chaîne de traitement pour obtenir le masque de nuages utilisé. En sortie, les pixels portent la valeur 99 en présence de nuage et 0 sinon.

Dans un second temps, l'image contenant pour chaque pixel la valeur du NDVI est créée en utilisant les bandes 4 (rouge) et 8 (infrarouge) issues du traitement de Sen2cor et la bibliothèque Orpheo ToolBox (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 :

$$-1 < \frac{\text{bande8} - \text{bande4}}{\text{bande8} + \text{bande4}} < 1$$

Une fois ces deux images raster créées, elles sont combinées afin d'avoir une valeur de NDVI uniquement sur les pixels qui ne sont pas cachés par les nuages. Ceux cachés par les nuages prennent la valeur 99. Cela s'effectue grâce à la bibliothèque 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)"
```

## 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 et indépendants 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 dans la base de données
- **NEW\_INDEX** Un code qui permettra de calculer de nouveaux indices sur les images déjà acquises et traitées

Le dossier avec 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)

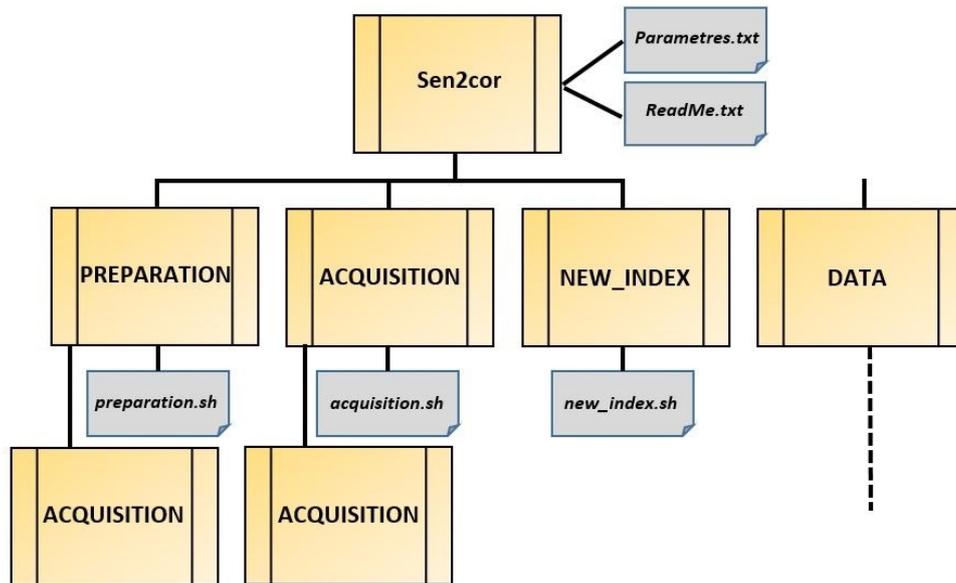


FIGURE 3.1 – Organisation du dossier du projet. Le dossier DATA avec la base de données peut être contenue dans le dossier du projet ou dans un dossier séparé.

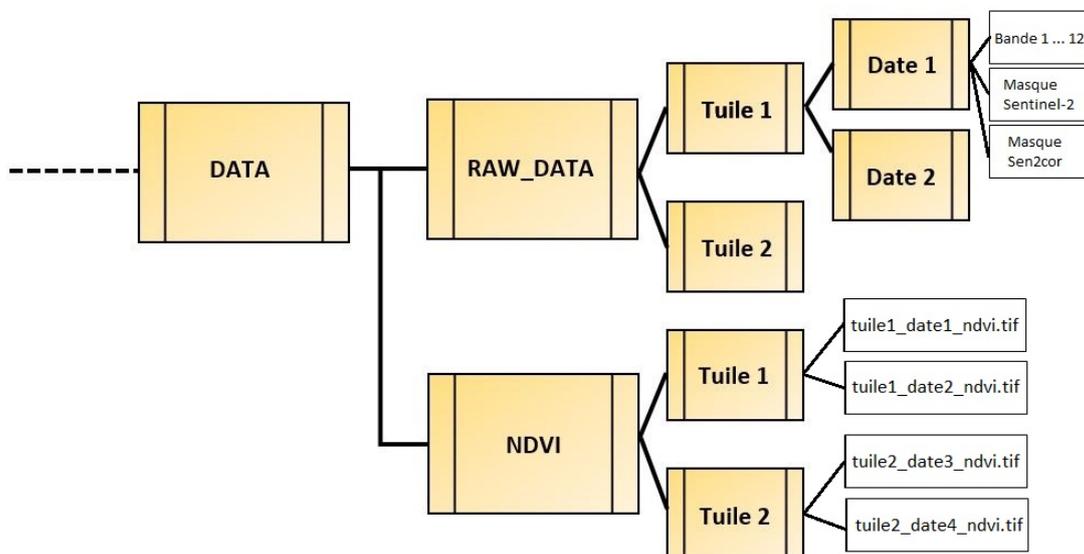


FIGURE 3.2 – Organisation du dossier DATA contenant la base de données requêteable

## 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 E) :

- 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 nuages 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. Ce dernier 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 s'il est sur terre ou en mer. Cela sera utile lors de l'écriture du fichier des séries temporelles des NDVI, car on ne veut prendre en compte ni les pixels sous un nuage, ni ceux en mer.

Une des contraintes qu'il a fallu prendre en compte pour le stockage dans la base de données concerne le poids de chaque image. En effet, pour une tuile données il y a une acquisition tous les cinq jours, soit 71 acquisitions par an. Cela correspond à 2 420 acquisitions par an pour les 34 tuiles au-dessus des sites épidémiologiques de Madagascar. Pour gagner de la mémoire, j'ai pris les décisions suivantes :

- Le masque des nuages est codé en bits
- Les images NDVI sont codées en *Int16* (entier à 16 bytes). Lors du calcul, les valeurs sont multipliées par 10000 avant de les convertir en *Int16* (cela est également le cas avec les données NDVI issues de MODIS). Pour gagner encore de la mémoire, il est possible de convertir les images en *jpeg2000* mais cela dégrade également la qualité de l'image NDVI. (Table 3.1)

Image	Extension	Poids
Bandes spectrales corrigées avec Sen2cor	.jp2 (Float16)	Entre 4 et 100Mo
Masque issu de Sen2cor	.tif (Byte)	3Mo
Image d'indice	.tif (Int16)	250Mo
Image d'indice	.jp2 (Int16)	60Mo

TABLE 3.1 – Poids des différentes images contenues dans la base de données

### 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 pourra être 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 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 à être repris et enrichis par l'ajout de nouveaux indices, autres que le NDVI. L'ajout des nouveaux indices doit être facile, et sans 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 alors de compléter cette fonction en rajoutant le calcul de nouveaux indices, en suivant l'exemple du NDVI.

## Chapitre 4

# Requêtes dans la base de données

### 4.1 Format des séries temporelles pour répondre aux besoins de l'IPM

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'IPM, 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								

Date du lundi de chaque semaine épidémiologique

Valeurs des indices pour chaque site pour chaque semaine

FIGURE 4.1 – Format du fichier .csv avec les séries temporelles à créer

L'IPM 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 des pixels 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 la valeur de NDVI et prend la valeur *no\_data* s'il est situé en mer ou sous un nuage. On utilise pour cela deux commandes, l'une GDAL et l'autre OTB. Puis la commande *zonal stat* de la librairie *rasterstat* permet d'obtenir la valeur moyenne de NDVI 5km autour du point d'intérêt. (Figure 4.2)

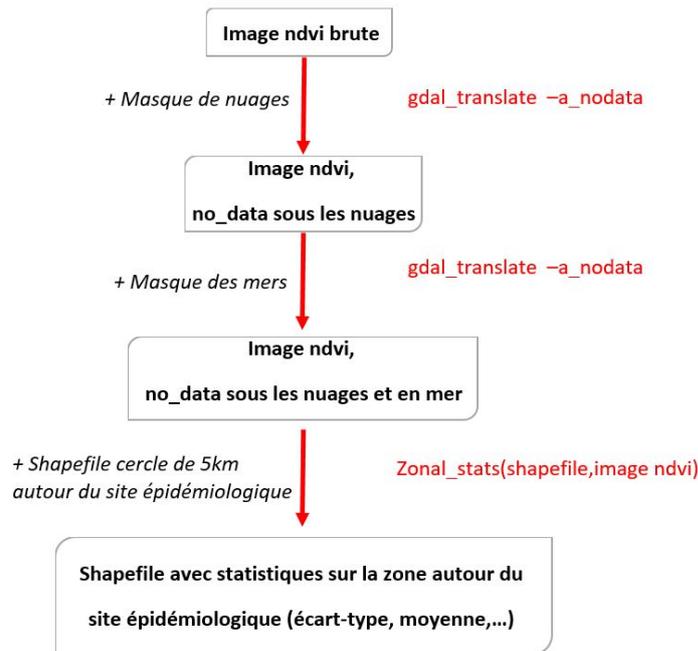


FIGURE 4.2 – Etapes pour obtenir la valeur moyenne de NDVI 5km autour du site épidémiologique considéré

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, non projetés. 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)

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 épidémilogiques	Shapefile (points)		WGS84

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

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

```
gdalwarp -ot Int16 -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 :

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

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

Si le site est entièrement dans les nuages, on reprend également la valeur de NDVI de la semaine précédente.

## 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 des mers. Il est issu de la base de données *OpenStreetMap*. Il s'agit d'un shapefile de polygones dessinant les limites de Madagascar. 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 des séries temporelles

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 de la semaine en cours.

Les deux étapes ont été codées en Python et rangées dans un dossier `CREATE_CSV` (Annexe F). Les deux étapes possédant des parties de code en commun, un troisième fichier Python est créé contenant les fonctions appelées ensuite dans les deux autres scripts. (Table 4.2)

Nom	Utilité
params.txt	Informations sur les chemins d'accès aux données
csv_preparation.py	Code pour l'étape de préparation
csv_automatization.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

**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
- **Path to out csv file** : Chemin d'accès et nom du fichier des séries temporelles en 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 sentinelles. 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 5km 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 F 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_automatization.py chemin/parametres.txt
```

Le script récupère tout d'abord le chemin d'accès au fichier .csv grâce à la fonction *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\_automatization.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 importer 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 <i>_MEAN</i>	0
find_indexName	Crée le fichier <i>_MEAN</i>	Valeur moyenne de l'indice
	Cherche la moyenne 5km autour du site	
getCloudMask	Récupère le chemin du masque nuages	Chemin du masque nuage
readParams	Lit le fichier des paramètres	Liste des informations du fichier
buffer	Crée pour un site le shapefile de dilatation	0
createBDDpoints	Crée les shapefiles polygones pour les sites	0

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

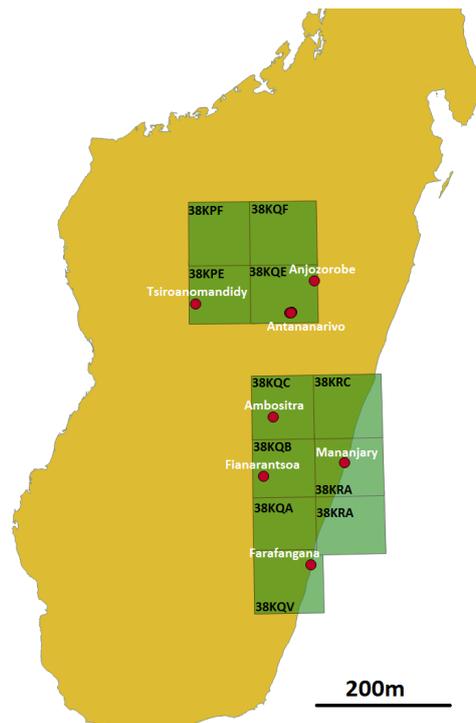


FIGURE 4.3 – Choix des tuiles à télécharger en priorité pour tester les séries temporelles

### 4.3 Résultats et comparaison avec les données MODIS

L'IPM m'a fourni trois emprises sur Madagascar à tester en priorité, situées au niveau des villes de Farafangana, de Manangary et d'Ankazobe. (Figure 4.3) J'ai donc téléchargé l'ensemble des images sur ces trois zones et j'ai créé les images NDVI associées, pour l'année 2017. J'ai ensuite utilisé le code écrit pour faire les séries temporelles des indices NDVI au niveau de différents sites épidémiologiques fournis par l'IPM. La cellule SIG de l'IPM m'a également fourni pour ces sites les séries temporelles calculées à partir d'images MODIS afin de pouvoir comparer les résultats. (Figure 4.4)

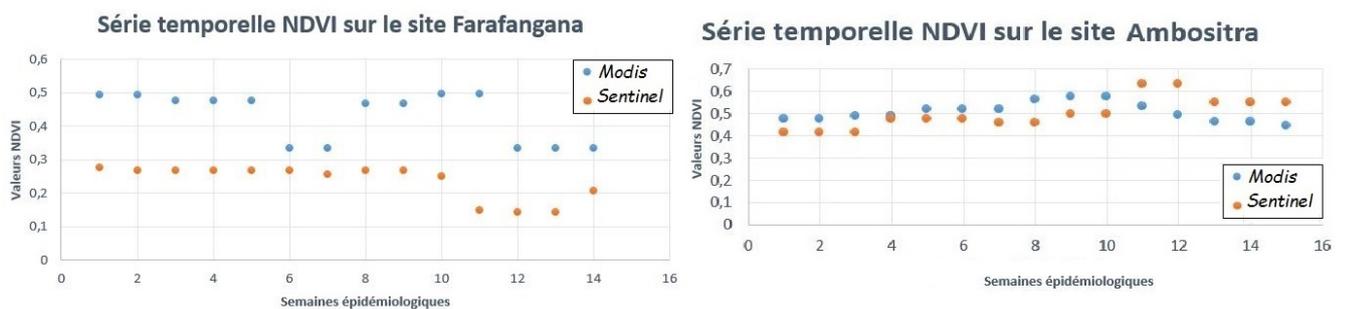


FIGURE 4.4 – Comparaison des séries temporelles des NDVI MODIS et Sentinel sur deux sites épidémiologiques de Madagascar

La comparaison n'a pu se faire que sur 16 semaines. Cela est insuffisant pour tirer des conclusions sur l'apport des données Sentinel par rapport aux données MODIS. Cependant, il semble que les deux produits aient des tendances similaires, Sentinel présentant des valeurs de NDVI plus élevées que MODIS. Les valeurs MODIS sur le site de Farafangana en semaines 6 et 7 semblent quant à elles aberrantes. Une des perspectives de mon travail est donc de faire une comparaison plus poussée entre les données MODIS et Sentinel pour en tirer des tendances spatiales et temporelles. Un article

est par ailleurs envisagé pour la suite du projet TOSCA, avec des comparaisons statistiques sur de gros volumes de données.

## 4.4 Produits complémentaires

En plus des séries temporelles pour les différents sites épidémiologiques, les futurs utilisateurs peuvent souhaiter télécharger les images des indices pour une tuile et une date en particulier. La seconde partie du travail de requête dans la base de données a donc été d'intégrer ces images dans le catalogue en ligne de SEAS-OI.

Connecté au catalogue en ligne, l'utilisateur peut choisir la donnée qu'il veut télécharger, la date et l'emprise. (Figure 4.5) Une fois la recherche lancée, la carte en ligne présente un aperçu de l'image que l'on va télécharger. Cette image, nommée *quickLook* est une image en trois bandes qu'il faut créer et stocker. (Figure 4.6)

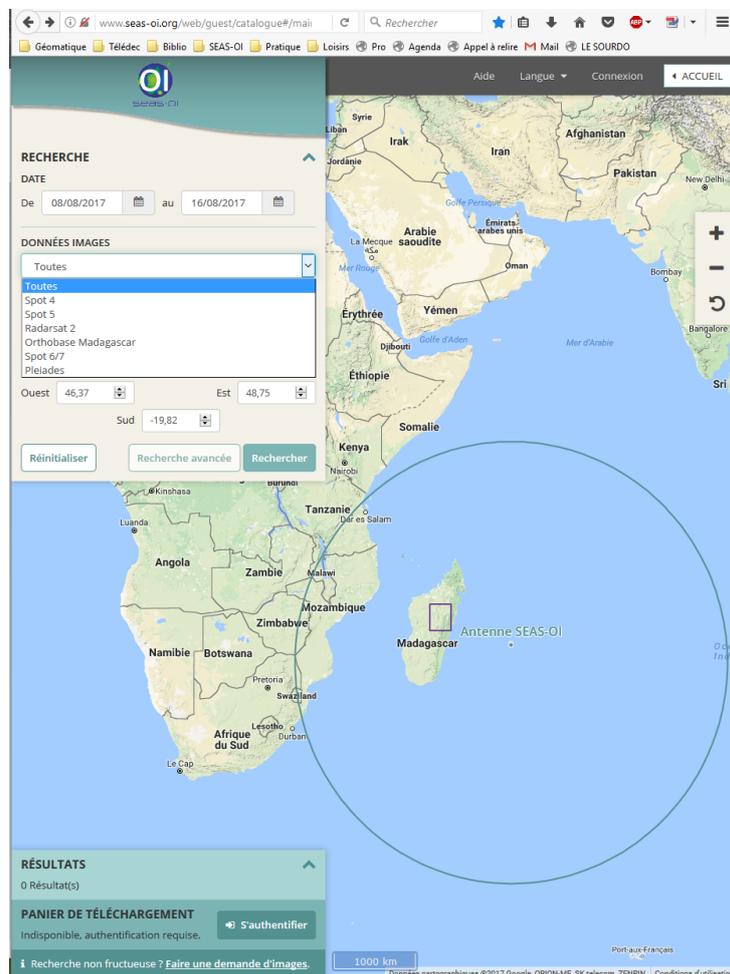


FIGURE 4.5 – Présentation du catalogue en ligne de SEAS-OI. L'utilisateur choisit la donnée qu'il veut télécharger et une image quicklook de prévisualisation permet de voir ce qu'il va télécharger

Dans une dernière partie de stage, j'ai donc automatisé la création d'un dossier contenant pour toutes les images d'indice calculées, les images quicklook correspondantes. Le code a été écrit en Python, avec les bibliothèques GDAL et OTB.

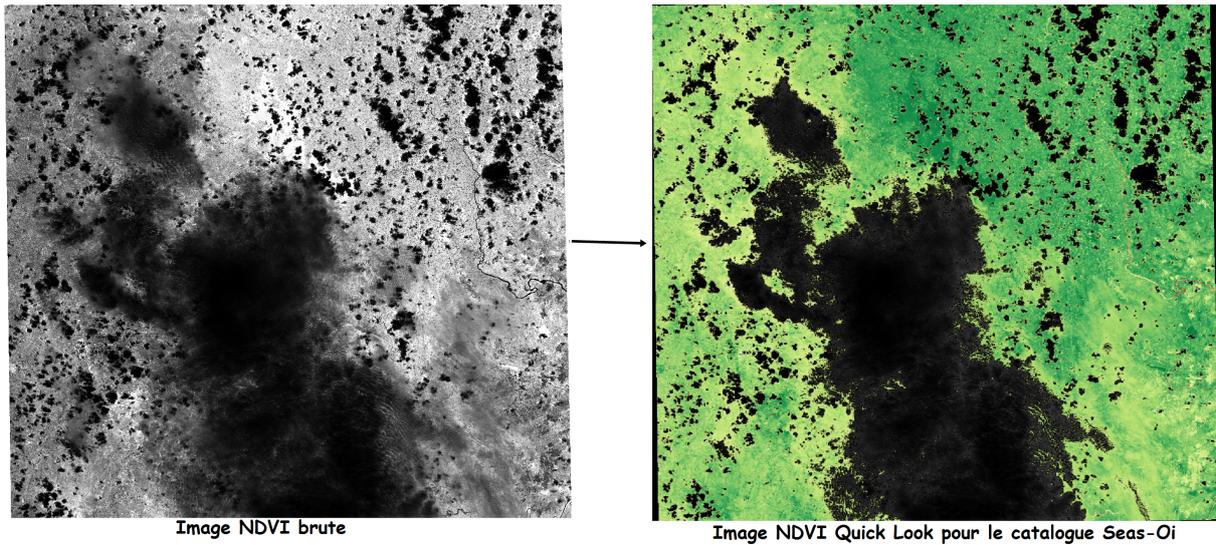


FIGURE 4.6 – Image QuickLook créée à partir de l’image NDVI brute. Les pixels situés sous des nuages sont masqués.

Un premier script nommé *createQL.py* permet de créer pour une image donnée le quicklook correspondant. Le second script, *createAllQL.py* fait appel au premier script et permet de créer et stocker tous les quicklook de toutes les images d’un indice donné. (Annexe H)

RdYIGn.lut			
1	0	165	0 38
2	1	166	1 38
3	2	168	3 38
4	3	170	5 38
5	4	172	7 38
6	5	174	9 38
7	6	176	11 38
8	7	178	13 38
9	8	180	15 38
10	9	182	16 38
11	10	184	18 38
12	11	186	20 38
13	12	188	22 38
14	13	190	24 38
15	14	192	26 38
16	15	194	28 38
17	16	196	30 38
18	17	198	32 38

Valeur NDVI  
 Valeur canal rouge  
 Valeur canal vert  
 Valeur canal bleu

FIGURE 4.7 – Format du fichier Look-Up Table utilisé pour convertir chaque valeur NDVI en une couleur. Pour avoir des valeurs positives de NDVI nécessaires à la création de la table, on additionne 10 000 à toutes les valeurs de NDVI.

Pour convertir l'image NDVI, qui ne possède qu'une bande, en une image à trois bandes, on utilise la fonction *colorMapping* de OTB. Un fichier *Look-Up Table* permet de convertir chaque valeur NDVI en une composition de trois valeurs (Rouge, Vert, Bleu). Cette table est obtenue à partir d'un script Python qui fait appel à la commande *cm.RdYIGn* de la bibliothèque Matplotlib. (Figure 4.7)

Puis les pixels situés sous les nuages sont colorés en noir en utilisant le masque de nuages. Enfin, la résolution de l'image est réduite pour rendre l'image moins lourde et pouvoir facilement la charger sur le site. Le téléchargement permet d'obtenir un dossier zippé contenant à la fois l'image NDVI brute et le masque des nuages associé.

# Conclusion

Ainsi, l'objectif principal du stage, à savoir automatiser une chaîne complète d'acquisition et de traitement d'images Sentinel-2 pour créer une base de données d'images d'indices environnementaux, a été rempli. Cette base de données est requêtable, permettant ainsi de créer des fichiers de séries temporelles des différents indices. Ces fichiers ont notamment pour vocation à être utilisés dans le système de détection précoce des pics de paludisme développé à l'IPM. Les contraintes techniques ont été prises en compte dans les choix de développement des codes (réutilisation facile des codes, minimisation du poids des images, etc.). De plus, la chaîne de traitement a été codée de manière à pouvoir facilement enrichir la base de données avec de nouveaux indices environnementaux.

En fin de stage, les programmes codés ont été installés sur un serveur dédié, dans les locaux de SEAS-OI à Saint-Pierre de La Réunion, afin de lancer la production prochainement. Les images au-dessus des 34 sites épidémiologiques donnés par l'IPM vont être traitées en priorité puis les images sur l'ensemble de Madagascar. A terme, l'objectif est d'obtenir au-moins une base de données pour Madagascar et pour La Réunion, avec des images d'indices intégrées également dans le catalogue de SEAS-OI.

Dans un deuxième temps, dans le cadre du projet Sentinel-2 Malaria, les séries temporelles des indices vont être analysées afin de comparer avec les données MODIS et voir ce que peut apporter des données Sentinel pour un système de prévention et de détection précoce des pics de paludisme.

Ce stage de cinq mois aura été très enrichissant car s'intègre pleinement dans un travail d'ingénieur. En effet, en amont j'ai participé à des réunions afin de définir les besoins et le livrable répondant au mieux à ces besoins. Puis le développement du produit m'a contraint à faire des choix techniques (chaîne de traitement, organisation de la base de données) et à échanger régulièrement avec les commanditaires du projet.

# Bibliographie

- [1] Stephen J CONNOR. "An online operational rainfall-monitoring resource for epidemic malaria early warning systems in Africa". In: *Malaria Journal* (2005). URL: <http://www.malariajournal.com/content/4/1/6>.
- [2] Florian GIROND. "Mise en place d'un système d'information géographique pour la détection précoce et la prédiction 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.
- [3] Florian GIROND et al. "Analysing trends and forecasting malaria epidemics in Madagascar using a sentinel surveillance network : a web-based application". In: *Malaria Journal* (Février 2017).
- [4] 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.
- [5] 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).
- [6] 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).
- [7] 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).
- [8] *MODIS Vegetation Index User's Guide (MOD13 Series), Version 3.00*. The University of Arizona. Juin 2015.
- [9] R. Snow S. HAY O. Ox. "Urbanization, malaria transmission and disease burden in Africa". In: *Europe PMC Funders Group* 3 (Avril 2017), pp. 81–90.
- [10] *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).
- [11] "Towards malaria elimination in the MOSASWA (Mozambique, South Africa and Swaziland) region". In: *Malaria Journal* 15 (2016), pp. 1–419.

# Table des figures

1.1	Déroulement du projet Tosca Sentinel-2 sur deux ans . . . . .	9
2.1	Bandes spectrales et résolutions spatiales des images Sentinel-2 . . . . .	13
2.2	Comparaison du résultat de la correction des images avec la chaîne de traitement Sen2cor, avec et sans la correction de la topographie. En rouge, les artefacts lors de l'ajout d'un MNT . . . . .	15
2.3	Zones d'étude choisies pour la comparaison des chaînes de traitement . . . . .	17
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. . . . .	18
2.5	Comparaison du masque issu de Sen2cor, avant et après traitement d'érosion et de dilatation. L'érosion permet d'éliminer une partie des bâtiments et des plages et la dilatation d'obtenir un masque englobant entièrement les nuages. . . . .	19
2.6	Résultat avec le masque issu de MUSCATE. En vert, les ombres des nuages détectés et en rouge des erreurs. . . . .	19
2.7	Comparaison des histogrammes des valeurs des pixels sur les images NDVI au dessus de La Réunion, avant traitement et avec les trois chaînes de traitement testées . . . . .	21
2.8	Chaîne de traitement pour obtenir le masque de nuages utilisé. En sortie, les pixels portent la valeur 99 en présence de nuage et 0 sinon. . . . .	22
3.1	Organisation du dossier du projet. Le dossier DATA avec la base de données peut être contenue dans le dossier du projet ou dans un dossier séparé. . . . .	25
3.2	Organisation du dossier DATA contenant la base de données requêtable . . . . .	25
4.1	Format du fichier .csv avec les séries temporelles à créer . . . . .	29
4.2	Etapes pour obtenir la valeur moyenne de NDVI 5km autour du site épidémiologique considéré . . . . .	30
4.3	Choix des tuiles à télécharger en priorité pour tester les séries temporelles . . . . .	34
4.4	Comparaison des séries temporelles des NDVI MODIS et Sentinel sur deux sites épidémiologiques de Madagascar . . . . .	34
4.5	Présentation du catalogue en ligne de SEAS-OI. L'utilisateur choisit la donnée qu'il veut télécharger et une image quicklook de prévisualisation permet de voir ce qu'il va télécharger . . . . .	35
4.6	Image QuickLook créée à partir de l'image NDVI brute. Les pixels situés sous des nuages sont masqués. . . . .	36
4.7	Format du fichier Look-Up Table utilisé pour convertir chaque valeur NDVI en une couleur. Pour avoir des valeurs positives de NDVI nécessaires à la création de la table, on additionne 10 000 à toutes les valeurs de NDVI. . . . .	36
A.1	Imprime écran de l'interface web de l'outil d'alerte précoce des épidémies de paludisme de l'Institut Pasteur de Madagascar. Quatre algorithmes statistiques sont disponibles pour détecter les potentiels pics de paludisme. . . . .	45
A.2	Présentation du fonctionnement du système d'alerte précoce du paludisme . . . . .	46
B.1	Gantt prévisionnel et Gantt réel . . . . .	48

C.1	Présentation de l'organisation générale d'un dossier d'images Sentinel traitées avec la chaîne de traitement Sen2cor . . . . .	50
D.1	Masques obtenus avec les différentes chaînes de traitement . . . . .	52
E.1	Algorithme mis en place pour le téléchargement, le traitement et le stockage des images Sentinel-2 déjà disponibles, codé en bash . . . . .	53
E.2	Fonction de traitement des images acquises, codée en bash. La fonction treatment2017() ressemble à treatment2016() mais l'architecture des dossiers est différente. . . . .	54
F.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 . . . . .	56
F.2	Algorithme pour la mise à jour chaque semaine du fichier .csv des séries temporelles d'un indice, codé en Python . . . . .	57
G.1	Structure du fichier parametres.txt pour créer la base de données . . . . .	58
G.2	Structure du fichier parametres.txt pour créer le fichier .csv avec les séries temporelles des indices . . . . .	58
G.3	Organisation générale du dossier avec les codes, réalisé lors du stage . . . . .	59
G.4	Organisation générale de la base de données . . . . .	60
H.1	Algorithme mis en place pour le téléchargement, le traitement et le stockage des images Sentinel-2 déjà disponibles, codé en bash . . . . .	61

# Liste des tableaux

2.1	Caractéristiques des différentes missions d'observation de la surface de la Terre . . . .	12
2.2	Poids des résultats en sortie des différentes chaînes de traitement. L'essentiel du poids de chaque dossier est porté par les images des bandes spectrales . . . . .	18
2.3	Comparaison des chaînes de prétraitement . . . . .	20
3.1	Poids des différentes images contenues dans la base de données . . . . .	26
4.1	Bilan des projections utilisées pour les différentes images et les shapefiles de la base de données . . . . .	30
4.2	Fichier et scripts Python pour la création et la mise à jour du fichier des séries temporelles de NDVI . . . . .	32
4.3	Liste des fonctions codées pour pour la création et la mise à jour des fichiers .csv . .	33
F.1	Liste des librairies utilisées dans le script Python développé . . . . .	55

# Annexes

A	Présentation de l'application web développée par Florian Girond . . . . .	44
B	Déroulement du stage . . . . .	47
C	Organisation du dossier créé par le traitement de Sen2cor . . . . .	49
D	Comparaison des chaînes de prétraitement sur l'île de La Réunion . . . . .	51
E	Algorithmes de création de la base de données . . . . .	53
F	Algorithmes de création du fichier .csv des séries temporelles . . . . .	55
G	Organisation finale du dossier . . . . .	58
H	Algorithmes de création des images Quicklook pour l'intégration au catalogue Seas-Oi . . . . .	61

## **Annexe A**

# **Présentation de l'application web développée par Florian Girond**

**Alert Thresholds**

**Algorithms:**

- Percentile
- MinSan
- C-SUM
- RDT+fever Indicator

**About algorithm**

An alert is triggered when a week malaria cases exceeds selected value (below) of percentile of the whole chronological series of a site. The 90th percentile is the value such that 90% of the time series values are below it (and therefore 10% are above).

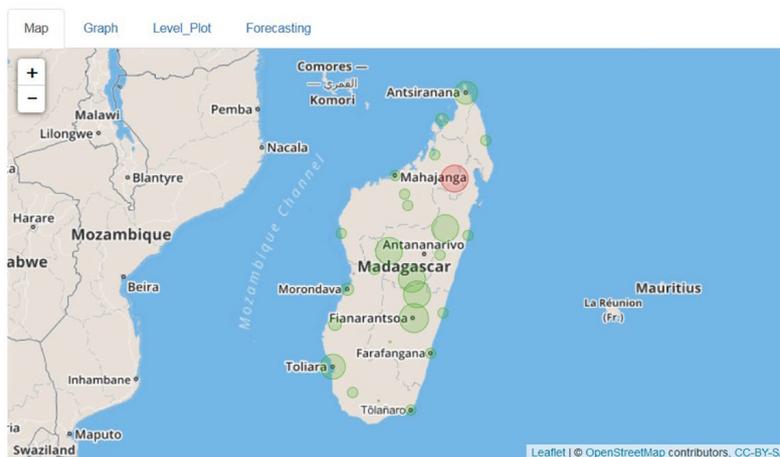
**Centile Value:**

1 50 100

This option is intended to improve the specificity of the alert system

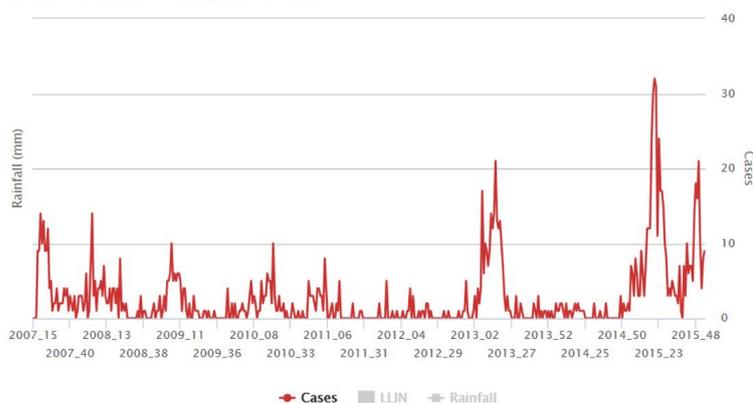
**Number of consecutive weeks above threshold:**

- 1
- 2
- 3
- 4



- Circle width is proportional to percentile value (click on map and see below)
- Green circle: The number of reported malaria cases is below the threshold
- Red circle: The number of reported malaria cases is above the threshold
- Hand icon: Click on a site to see malaria time series

cases of Tsiroanomandidy , tdd 2016\_01



Institut Pasteur de Madagascar  
USAID FROM THE AMERICAN PEOPLE  
Surveillance Sentinelle Madagascar (beta)

FIGURE A.1 – Imprime écran de l'interface web de l'outil d'alerte précoce des épidémies de paludisme de l'Institut Pasteur de Madagascar. Quatre algorithmes statistiques sont disponibles pour détecter les potentiels pics de paludisme.

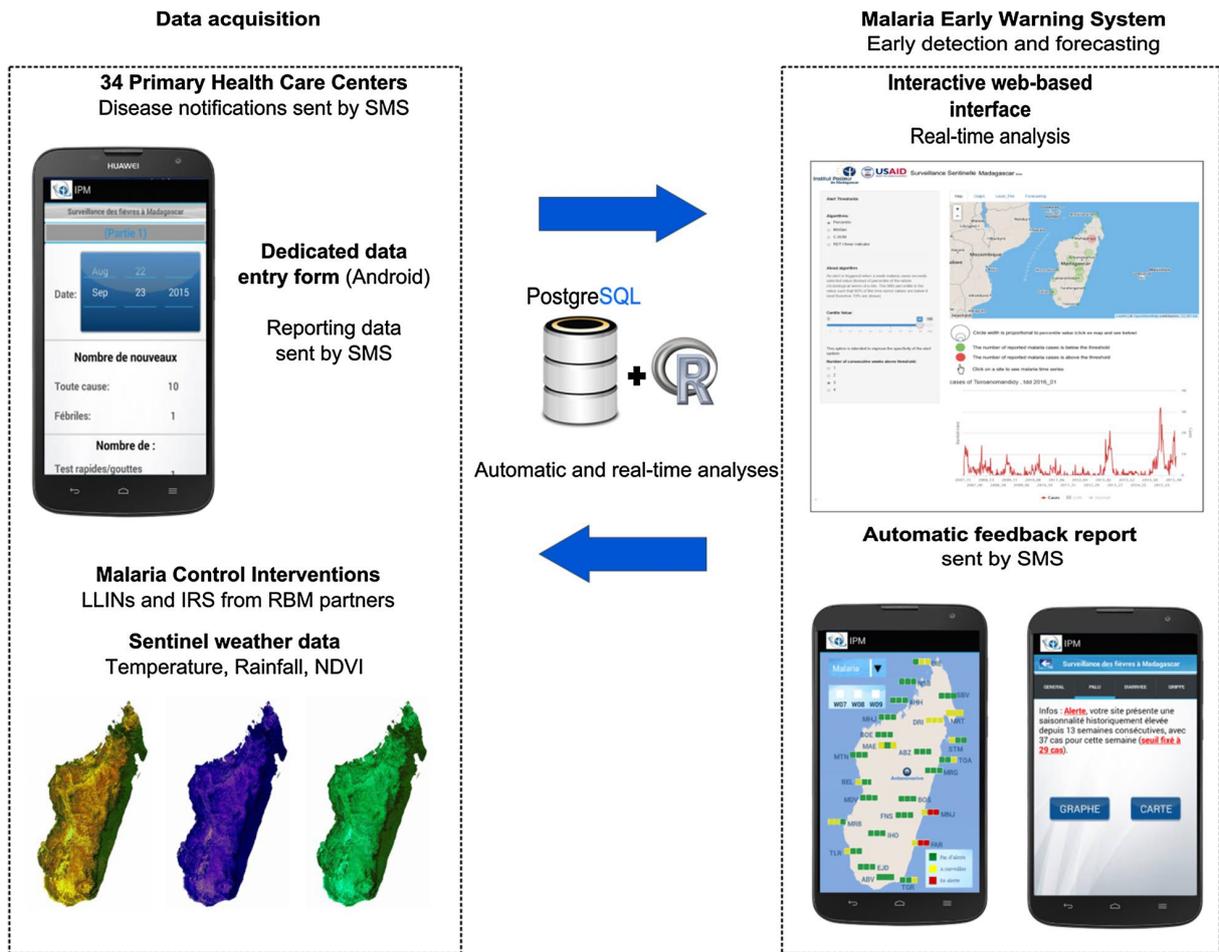
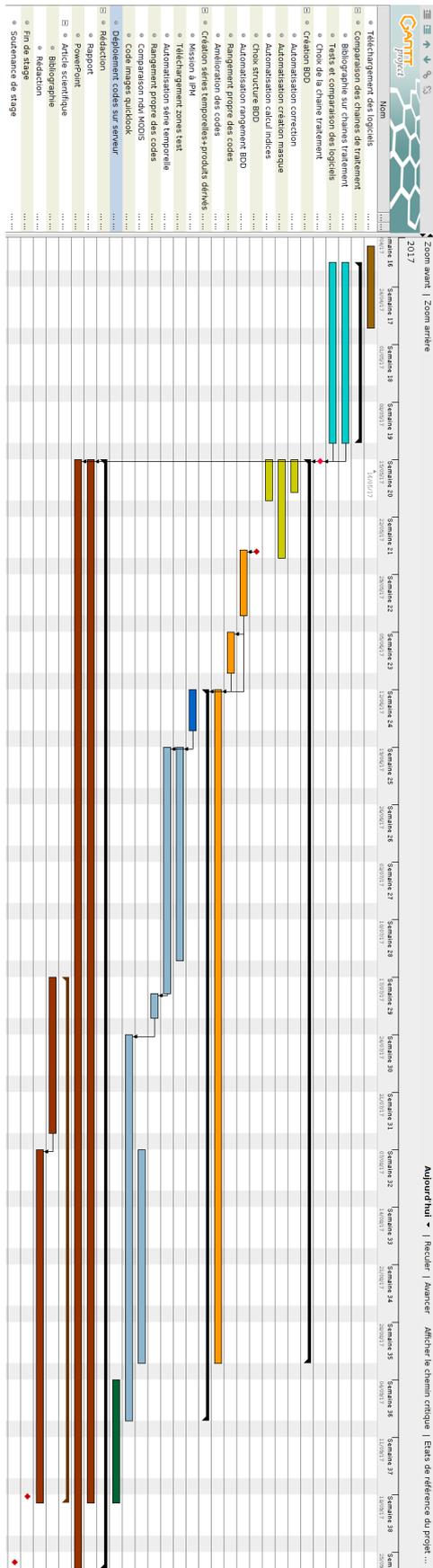


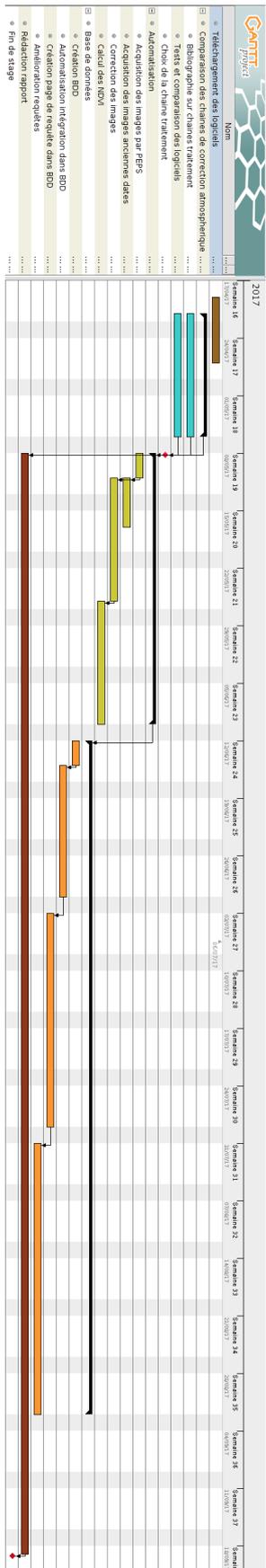
FIGURE A.2 – Présentation du fonctionnement du système d'alerte précoce du paludisme

**Annexe B**

# **Déroulement du stage**



## Gantt Réel



## Gantt Prévisionnel

FIGURE B.1 – Gantt prévisionnel et Gantt réel

## **Annexe C**

# **Organisation du dossier créé par le traitement de Sen2cor**

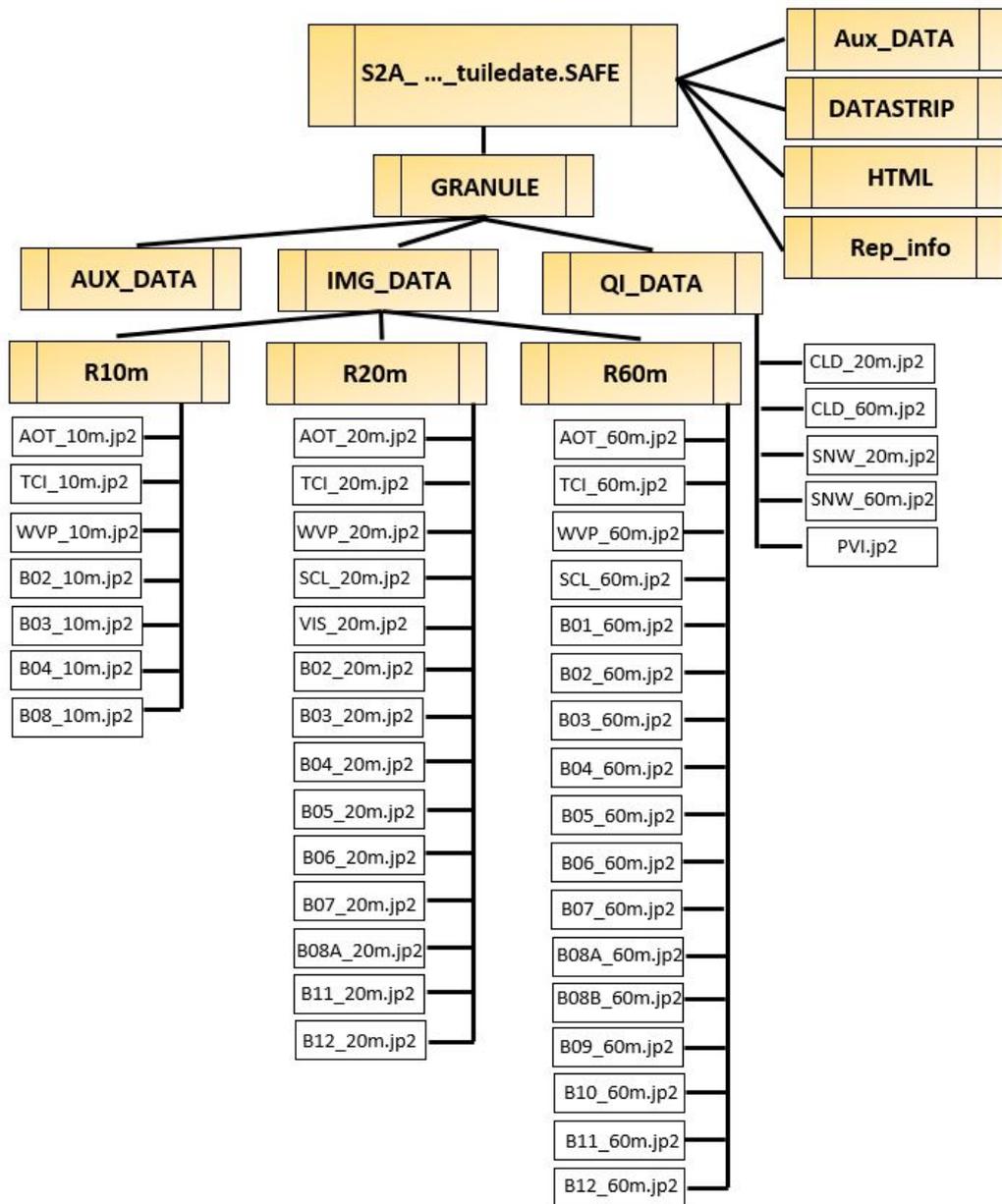


FIGURE C.1 – Présentation de l'organisation générale d'un dossier d'images Sentinel traitées avec la chaîne de traitement Sen2cor

## **Annexe D**

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

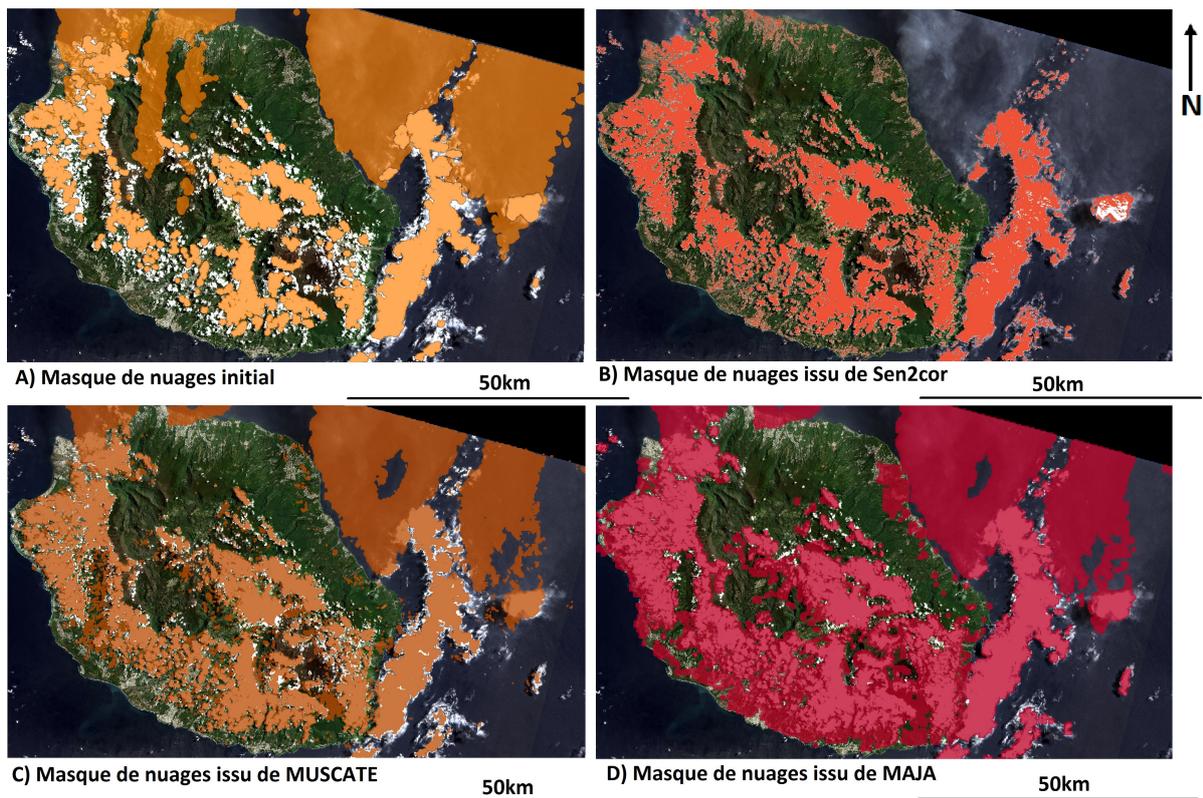


FIGURE D.1 – Masques obtenus avec les différentes chaînes de traitement

# Annexe E

## Algorithmes de création de la base de données

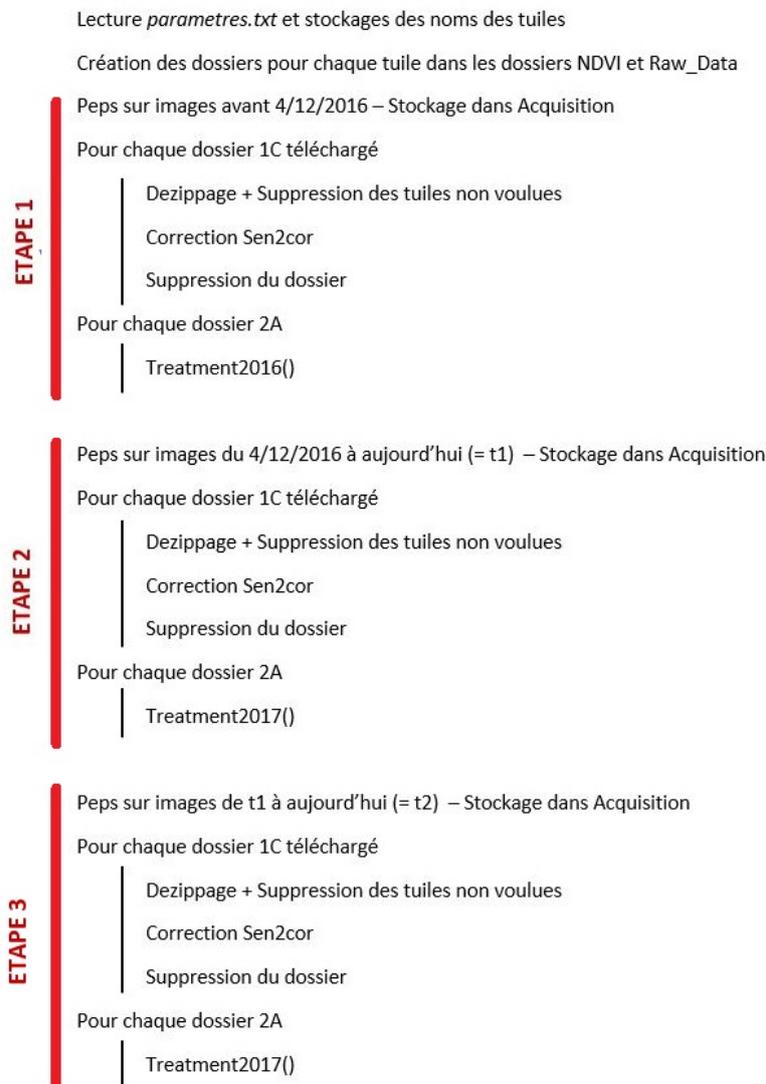


FIGURE E.1 – Algorithme mis en place pour le téléchargement, le traitement et le stockage des images Sentinel-2 déjà disponibles, codé en bash

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 copie dans DATA/NDVI/tuile
- Suppression du dossier

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

## Annexe F

# 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 F.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]

Première semaine
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

Toutes les autres semaines
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
                    
$$NDVI = \frac{previousWeek[c] + n^{\circ} jour semaine + (7 - n^{\circ} jour semaine) * valeur}{7}$$

                    Week[c] = NDVI
        c += 1
    previousWeek = Week
    dateSemaine += 7
    Ecriture de la ligne
Suppression des images _MEAN.tif créées → removeMeanFile()

```

FIGURE F.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
                
$$NDVI = \frac{ligne\_2[c] + n^{\circ} jour\_semaine + (7 - n^{\circ} jour\_semaine) \cdot valeur}{7}$$

                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
                
$$NDVI = \frac{ligne\_1[c] + n^{\circ} jour\_semaine + (7 - n^{\circ} jour\_semaine) \cdot valeur}{7}$$

                ligne_0[c] = NDVI

Suppression des images _MEAN.tif créées → removeMeanFile()
Suppression dernière ligne fichier .csv
Ecriture ligne_1 et ligne_0

```

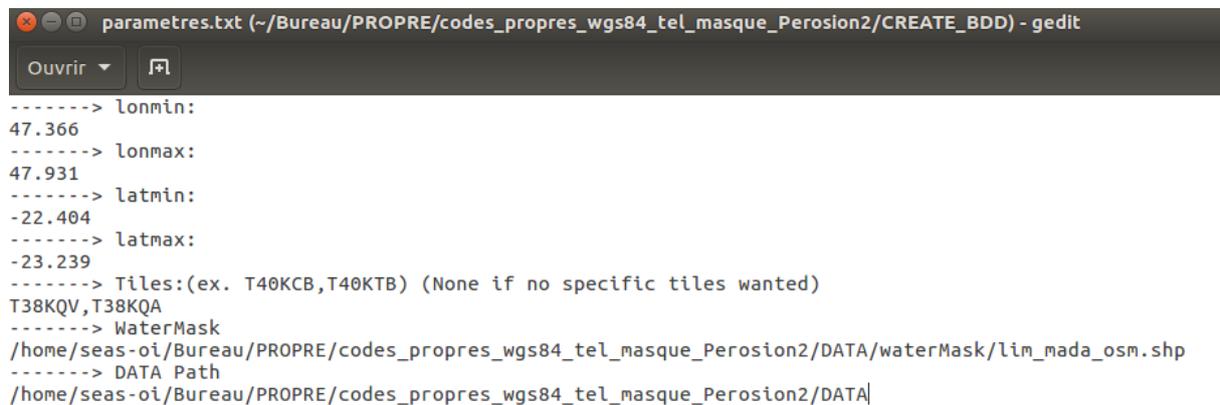
Mise à jour de la semaine précédente

Ajout de la nouvelle semaine

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

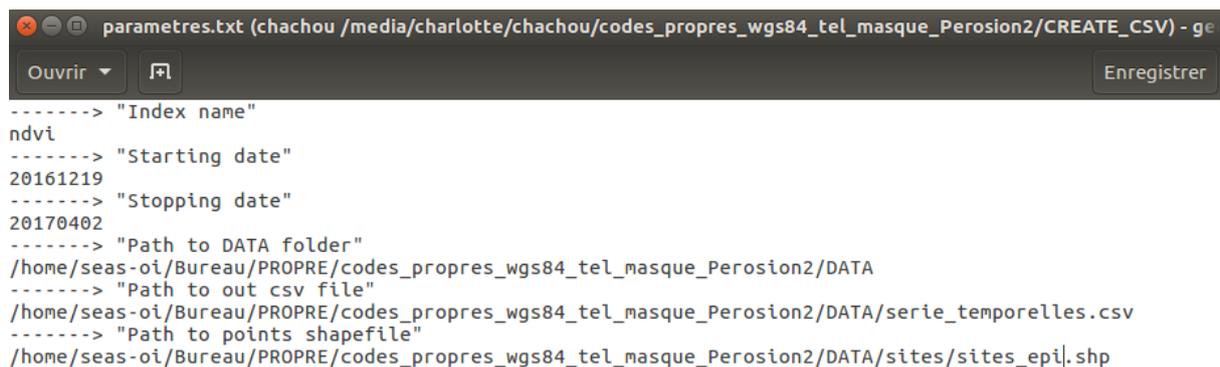
## Annexe G

# Organisation finale du dossier



```
-----> 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_Perosion2/DATA/waterMask/lim_mada_osm.shp
-----> DATA Path
/home/seas-oi/Bureau/PROPRE/codes_propres_wgs84_tel_masque_Perosion2/DATA|
```

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



```
-----> "Index name"
ndvt
-----> "Starting date"
20161219
-----> "Stopping date"
20170402
-----> "Path to DATA folder"
/home/seas-oi/Bureau/PROPRE/codes_propres_wgs84_tel_masque_Perosion2/DATA
-----> "Path to out csv file"
/home/seas-oi/Bureau/PROPRE/codes_propres_wgs84_tel_masque_Perosion2/DATA/serie_temporelles.csv
-----> "Path to points shapefile"
/home/seas-oi/Bureau/PROPRE/codes_propres_wgs84_tel_masque_Perosion2/DATA/sites/sites_epi.shp
```

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

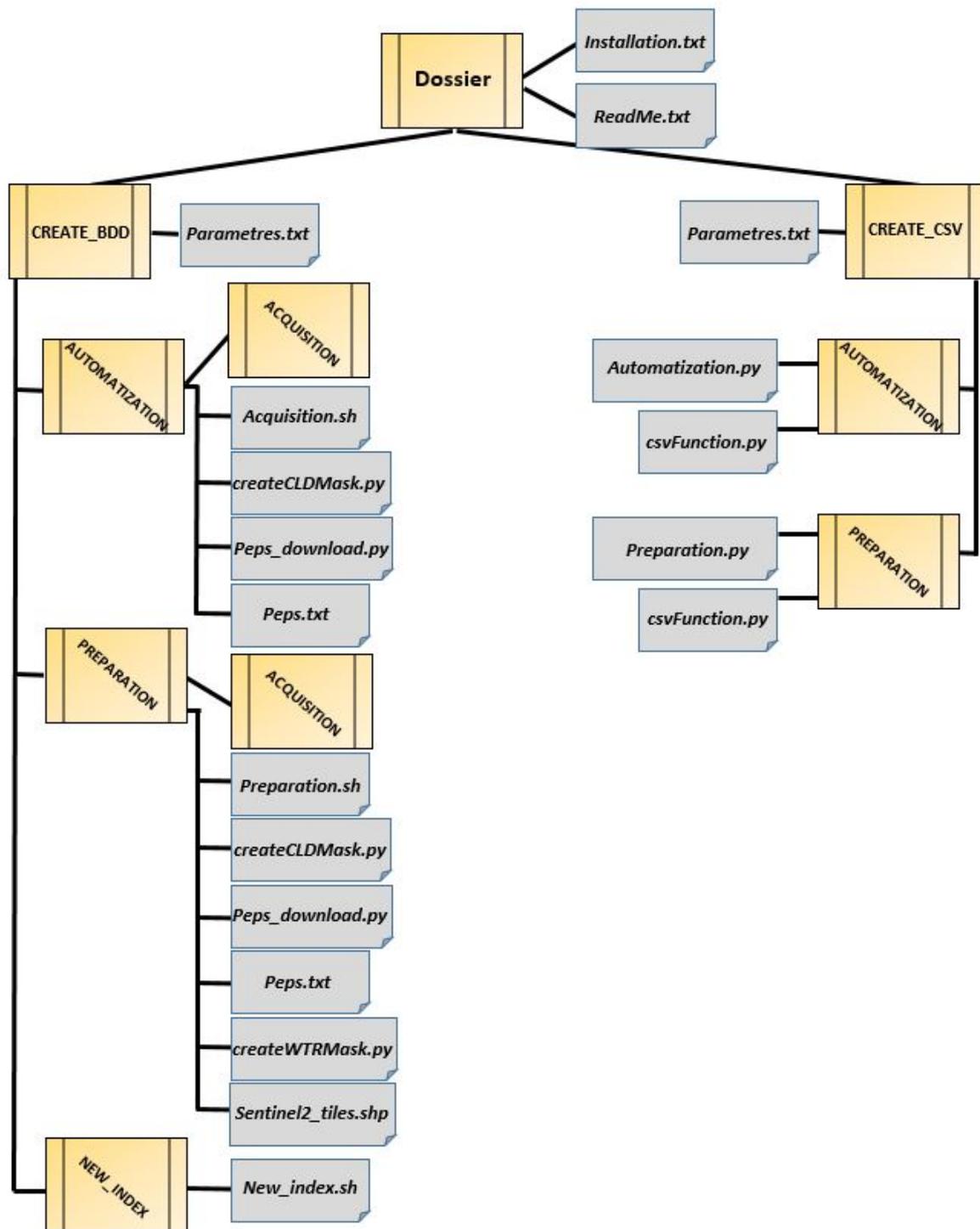


FIGURE G.3 – Organisation générale du dossier avec les codes, réalisé lors du stage

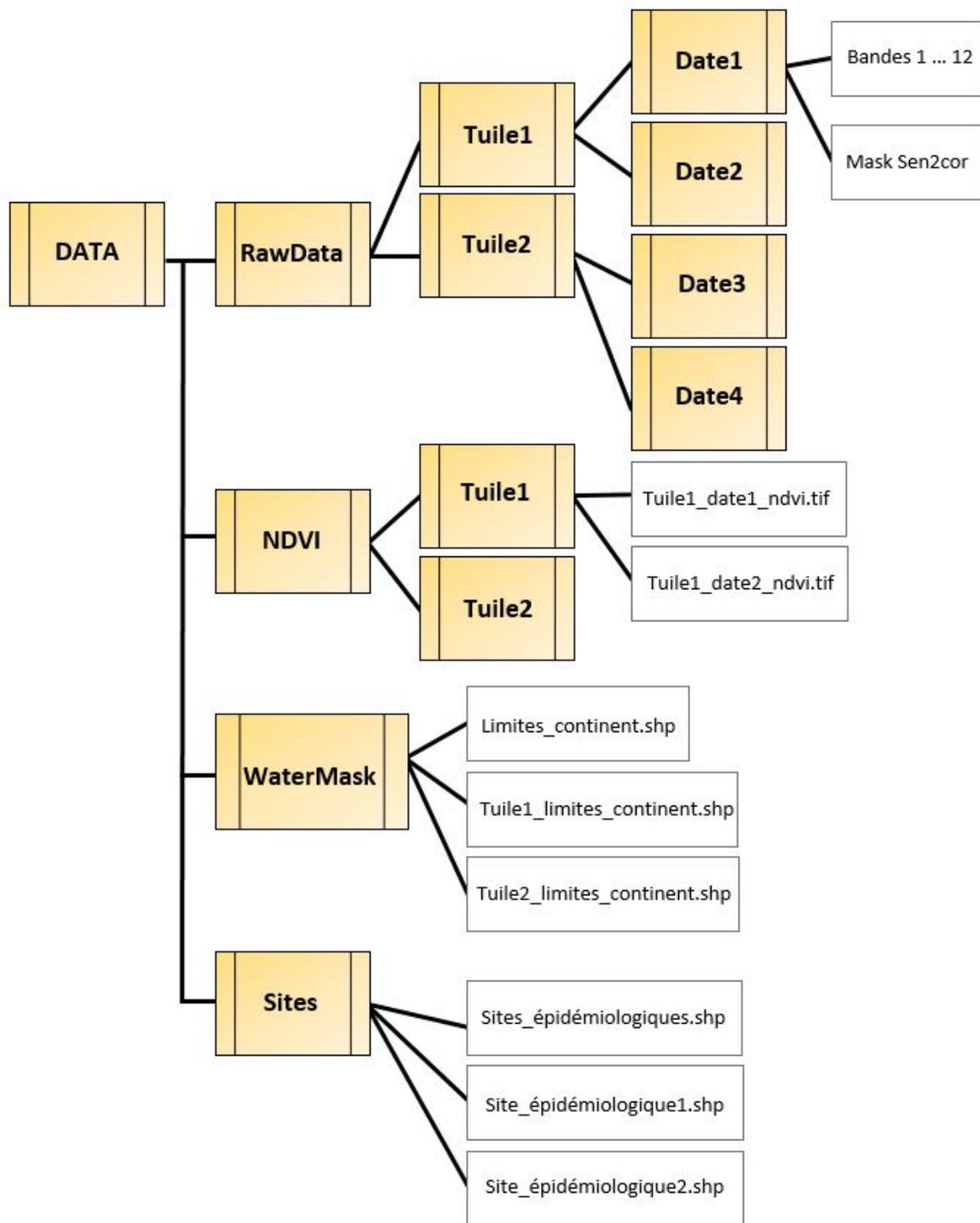


FIGURE G.4 – Organisation générale de la base de données

## Annexe H

# Algorithmes de création des images Quicklook pour l'intégration au catalogue SEAS-OI

```
def createColorImage(image2color,CLDmask,out):  
    #Recalibrage de l'image entre [0 ; 255]  
    gdal_calc.py -A image2color --outfile=resized.tif --calc=(128/10000*(A+10000))  
    # Conversion image en 3 bandes [Rouge, Vert, Bleu]  
    otbcli_ColorMapping -in resized.tif -method custom -method.custom.lut LUT.txt -out Colorized_summer.tif  
    # Nuages colorés en noir  
    gdal_calc.py -A Colorized_summer.tif -B CLDmask --outfile=cloud.tif --calc=A*(B==0)+1*(B>0) --allbands=ALLBANDS  
    # Bords de l'image colorés en noir  
    gdal_calc.py -A cloud.tif -B image2color --outfile=border.tif --calc=A*(B>=0)+1*(B<0) --allbands=ALLBANDS  
    # Changement de resolution  
    Gdal_translate --outsize 1000 1000 -r bilinear border.tif out  
    Suppression des fichiers intermédiaires créés  
    Return 0  
  
def createAllQuickLook(DataPath,indexName,out):  
    Tuiles = liste des dossiers dans DataPath/indexName  
    Pour chaque tuile de tuiles :  
        Images = liste des dossiers dans DataPath/indexName/tuile  
        Pour chaque image de images :  
            createColorImage(image, masque nuage correspondant, out)  
            Return 0
```

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