

## Partie II

# Analyse de séries temporelles de NDVI

## 5 Objectifs

L'indice de végétation par différence normalisé (NDVI), utilisé pour la première fois par Rouse *et al.* (1974), est un indicateur numérique basé sur les réflectances du spectre visible et de l'infra-rouge. Il est utilisé dans le cadre d'analyses d'images satellites, permettant d'évaluer la quantité de végétation verte présente en un lieu donné. Il peut servir à estimer les rendements agricoles (Benedetti & Rossini 1993), à évaluer la dégradation des terres à plusieurs échelles (Yengoh *et al.* 2015), à cartographier les pâturages en fonction de leur biomasse (Edirisinghe *et al.* 2011), etc. L'un des objectifs de LiMES est d'offrir à l'utilisateur une vision rapide et réaliste de l'évolution de la couverture végétale sur une période et un site donné. Pour cela, plusieurs facteurs sont à prendre en compte : **i)** gestion des données manquantes, **ii)** automatisation des processus, **iii)** offre variée d'output complémentaires, **iv)** mise en forme des résultats. Un script Python a donc été créé à cet effet, dont le seul input est un dossier contenant les différentes scènes sur lesquelles seront basées les analyses. Ceux-ci doivent à leur tour contenir les bande de réflectances 4 (rouge) et 5 (infra-rouge) nécessaires au calcul du NDVI, ainsi que le masque de nuages issu de la fonction Fmask (voir section 4, p. 25).

## 6 Comblement des valeurs manquantes

Une fois l'image 'nettoyée', il reste la question du traitement des données manquantes issues du masque. Selon les sites et les saisons, celles-ci peuvent être importantes et il peut être nécessaire de les combler. En résumé, il existe trois approches pour le remplissage des données manquantes (Weiss *et al.* 2014).

La première est basée sur l'information spatiale et consiste à utiliser une seule image source (ex. Rossi *et al.* 1994). Les vides peuvent être reconstruit par 'krigeage' ou par interpolation traditionnelle en utilisant d'autres régions de la même image. Différents algorithmes (recherche des plus proches voisins, algorithme bilinéaire, interpolation bilinéaire, interpolation bicubique, etc.) peuvent être utilisés à cet effet, bien qu'en général, cette approche produit des résultats insatisfaisants (Jabar *et al.* 2014).

La seconde approche est basée sur l'information temporelle et requiert plusieurs images sources. Une analyse de Fourier peut être réalisée sur plusieurs images afin de modéliser une série temporelle d'observations pixelliques permettant de combler les données manquantes (ex. Roerink *et al.* 2000). Aussi, d'autres algorithmes (fonction gaussienne, filtre de Savitzky-Golay, etc.) permettent de lisser des séries temporelles selon plusieurs paramètres (ex. Kandasamy *et al.* 2013). D'autres méthodes consistent simplement à remplacer une donnée manquante par une donnée existante provenant d'autres images (ex. Surya & Simon 2015) ou

d'une seule image de référence (ex. Jin *et al.* 2013).

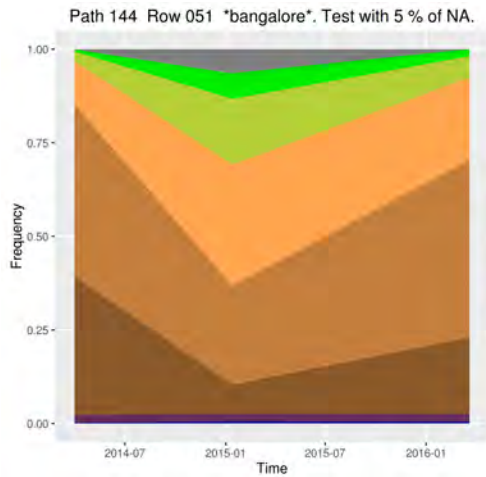
Finalement l'approche mixte combine l'information spatiale et temporelle. Par exemple Kang *et al.* (2005) ont développé un algorithme utilisant une interpolation temporelle lorsque l'information spatiale n'est pas suffisamment précise pour combler les vides.

Plusieurs facteurs dépendant du site, de l'image, de la rapidité de calcul souhaitée, de l'analyse à effectuer, etc. sont à prendre en compte pour choisir l'une ou l'autre méthode. Par exemple, l'utilisation d'images multi-temporelles doit être considérée avec précaution pour le calcul du NDVI, compte tenu des variations inter-annuelles de ce dernier (ex. Gamon *et al.* 2013; Olusegun & Adeyewa 2013). Aussi, une méthode simple d'interpolation spatiale pourrait s'avérer suffisante pour une image où les nuages sont de tailles réduites.

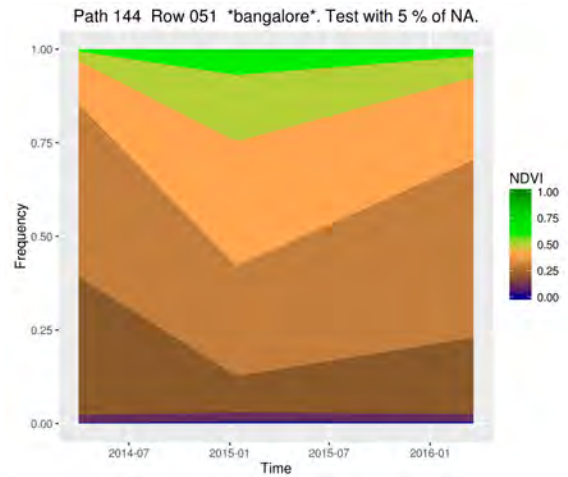
Voici une liste non-exhaustive des différents programmes/fonctions disponibles associés à la problématique des données manquantes dans le cadre d'analyses d'images satellites.

- *r.fillnulls* (GRASS) comble les données manquantes en utilisant une interpolation spline des pixels avoisinants.
- Le script Python *gdal\_fillnodata.py* de la librairie GDAL comble les données manquantes par interpolation des pixels avoisinants.
- *t.rast.gapfill* (GRASS) utilise une interpolation linéaire entre plusieurs couches d'une série temporelle et comble les données manquantes de chaque couche.
- La fonction *TSGFlinear* du package 'greenbrown' (R) comble les données manquantes d'une série temporelle par interpolation linéaire.
- La fonction *sgolayfilt* du package 'signal' (R) permet de lisser les données en exécutant un filtre de Savitzky-Golay sur une série temporelle.
- La fonction *approxNA* du package 'raster' (R) estime la valeurs des données manquantes par interpolation d'une série temporelle.

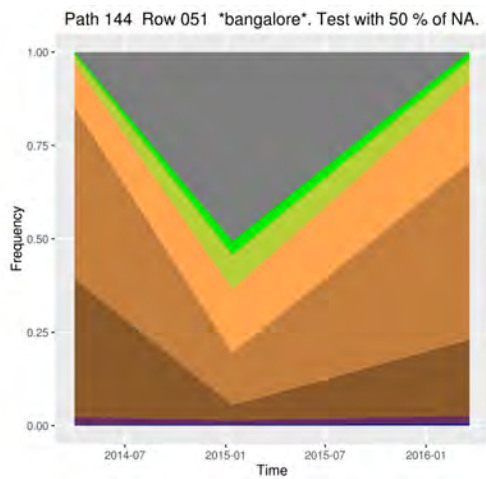
Le choix de la méthode permettant de combler les données manquantes dépend de la nature des inputs (une seule image ou série temporelle), de leur qualité et des objets d'analyses postérieures (constants ou fluctuants) et de la praticité de la méthode. Par exemple, si la taille des patchs de données manquantes est importante, une interpolation spatiale ne serait pas recommandée. De même, si l'objet d'analyse fluctue en fonction des saisons (ex. NDVI) un lissage des données sur une série temporelle aurait tendance à sous-estimer les variations intra-annuelles. Dans le cadre d'analyses de NDVI sur des séries temporelles, tel que présenté dans le projet LiMES, la fonction R *approxNA* du package 'raster' représente un solution adaptée pour les raisons mentionnées ci-dessus. En effet, elle effectue une interpolation linéaire entre les données les plus proches d'un point de vue temporel et permet de conserver le caractère oscillatoire de l'indice si les images ne sont pas trop éloignées les unes des autres. La fiabilité des images issus de l'interpolation dépend de la quantité de données existantes pour chaque pixel. Quoiqu'il en soit l'interpolation linéaire peut potentiellement provoquer une sous-estimation des variations (Figure 25).



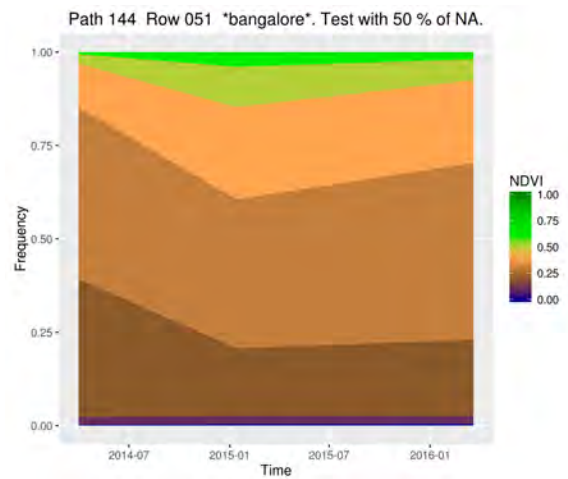
(a) Sans interpolation. 5% de données manquantes.



(b) Interpolation. 5% de données manquantes.



(c) Sans interpolation. 50% de données manquantes.



(d) Interpolation. 50% de données manquantes.

FIGURE 25 – Graphiques en aires de la proportion des différentes classes de valeurs de NDVI illustrant l'effet de l'interpolation linéaire en fonction du pourcentage de données manquantes (5% et 50%).

# 7 Script

## 7.1 Description

Un script Python intégrant certaines fonctions R offre à l'utilisateur la possibilité d'obtenir une série d'output complémentaires ainsi qu'une fiche de résumé en format PDF lui permettant d'appréhender l'évolution du NDVI au cours du temps sous plusieurs angles. Les output offerts à l'utilisateur incluent entre autre une représentation de la déviation standard des valeurs de NDVI permettant de mesurer la variabilité intra-annuelle de la végétation (ex. Hall-Beyer 2012), une représentation des coefficients de pente significatives issus de régressions linéaires effectuées sur chaque pixel (ex. Hartemink *et al.* 2008) ainsi qu'un graphique en aires illustrant l'évolution de la proportion des différentes classes de valeurs de NDVI au cours du temps. Toutes les analyses sont effectuées une première fois sans traiter les données manquantes, et une deuxième fois en leur assignant une valeur de NDVI via une interpolation linéaire tenant compte de la distance temporelle entre chaque scène. Ceci offre à l'utilisateur une certaine transparence et lui permet de définir, le cas échéant, la prudence avec laquelle les résultats complets (sans données manquantes) doivent être interprétés. La liste exhaustive des output issus de ce script est la suivante :

1. Un fichier ENVI multibande des valeurs de NDVI correspondant à chaque scène (par défaut)
2. Un raster (TIF) et une image (PNG) représentant les valeurs de déviation standard pour chaque pixel (argument optionnel -v)
3. Un raster (TIF) et une image (PNG) représentant les valeurs significatives de coefficient de pente (p-value < 0.05) issus des régressions linéaires réalisées sur chaque pixel (argument optionnel -s).
4. Un raster (TIF) et une image (PNG) représentant les différences de valeurs de NDVI entre la dernière et la première scène calculées sur chaque pixel (argument optionnel -d).
5. Un graphique (PNG) illustrant l'évolution de la distribution des valeurs de NDVI au cours du temps associé à un un fichier texte (TXT) contenant les données utilisées pour générer le graphique (argument optionnel -p).
6. Un graphique en aires (PNG) illustrant l'évolution de la proportion de différentes classes de valeurs de NDVI au cours du temps associé à un un fichier texte (TXT) contenant les données utilisées pour générer le graphique (argument optionnel -a).
7. Un raster (TIF) et une image (PNG) où la valeur de chaque pixel correspond au pourcentage du nombre de fois où celui-ci a été masqué, ceci permettant d'évaluer la fiabilité des données pour différentes zones des rasters en output (argument optionnel -m).
8. Un fichier PDF résumant l'analyse effectuée (données, résultats, etc.) et incluant les output précédents (par défaut).

## 7.2 Structure et processus

Le script est configuré de manière à être exécuté depuis le dossier LiMES contenant les différents scripts (*LiMES\_batcher*, *LiMES\_grasser*, etc.) et requiert l'exécution préalable du script *LiMES\_grasser* afin d'obtenir pour chaque scène les réflectances des bandes 4 et 5 ainsi que les masques de nuages issus de la fonction *Fmask*. L'argument principal à fournir lors de l'exécution du script est le nom du site sur lequel l'analyse du NDVI va être réalisée. Par défaut, le script identifie les couches à traiter en fonction des suffixes *T4*, *T5* et *Fmask*. Si ces derniers diffèrent, une modification du code permet de laisser à l'utilisateur la possibilité de définir les suffixes correspondant aux différents rasters (lignes de code commentées). Une fois les différentes scènes identifiées et les différentes variables établies, une boucle s'effectue sur les différentes scènes, laquelle permet d'appliquer les masques et d'effectuer les calculs de NDVI. Ceci réalisé, un *RasterStack* (collection de rasters ayant la même extension et la même résolution) est créé. Les différents output optionnels sont créés en fonction des choix de l'utilisateur d'abord une première fois sans traiter les données manquantes, puis une deuxième fois en ayant réalisé une interpolation linéaire afin de combler ces dernières. Le script a été pensé de manière à automatiser au maximum les processus et le format des output, notamment le format des graphiques où les titres, les labels et les valeurs de l'axe des abscisses sont obtenus par le nom des différentes scènes (eux-même contenant l'année et le jour julien de celle-ci). Tous les output ainsi que la fiche de résumé en format PDF sont exportés. Le script complet figure dans l'Annexe E.1.

Le module Python *rpy2* importé dans le script principal (*ndvi.py*) permet d'intégrer des fonctions R dans le langage Python. Il permet aussi de compiler des fonctions R pures (possible grâce à la fonction *SignatureTranslatedAnonymousPackage*). Ceci est d'autant plus utile que cette dernière permet de profiter des subtilités de R et de contourner certains obstacles liés aux différences de syntaxe. C'est pour cette raison qu'un script auxiliaire (*RFunctions.py*) a été élaboré contenant différentes fonctions codées en langage R (voir Annexe E.2). Aussi, le module Python *pylatex* importé dans le script principal (*ndvi.py*) permet de créer une interface entre Python et L<sup>A</sup>T<sub>E</sub>X et ainsi de générer des fichiers PDF. Le code permettant cela est écrit dans un script auxiliaire (*ndviPDF.py*) figurant dans l'Annexe E.3. Les ordigrammes structurel (Figure 26) et fonctionnel (Figure 27) résument le processus d'automatisation des analyses.

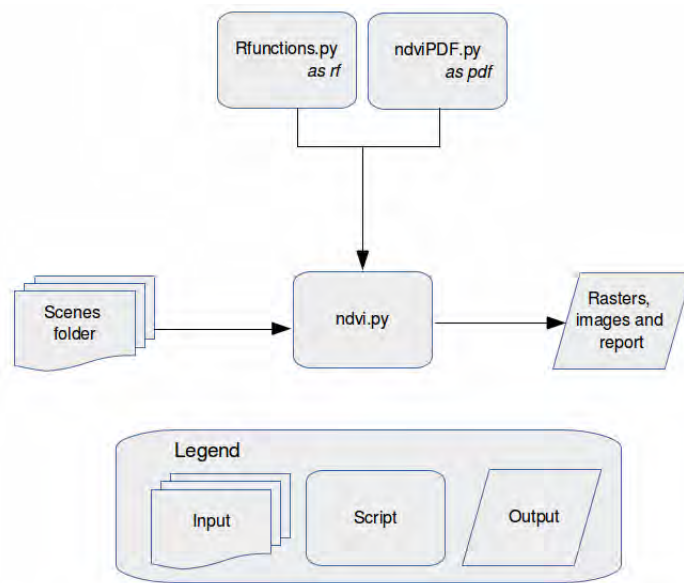


FIGURE 26 – Ordigramme structurel du processus d’automatisation des analyses effectuées sur des séries temporelles de NDVI.

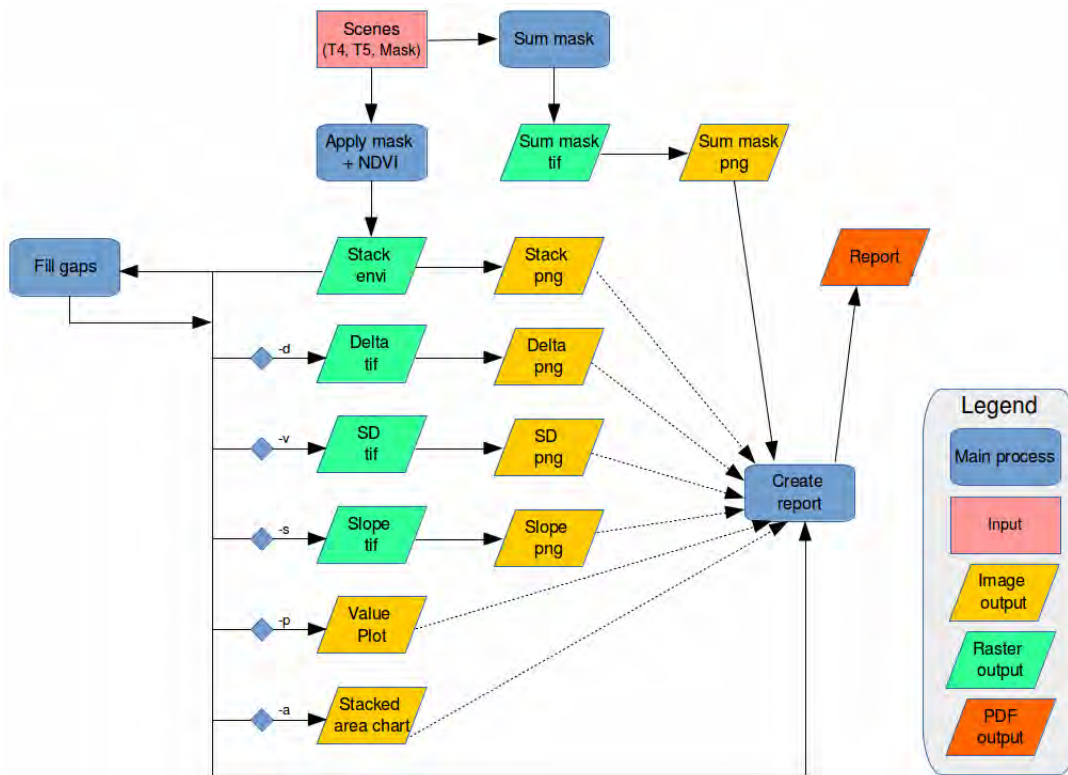


FIGURE 27 – Ordigramme fonctionnel du processus d’automatisation des analyses effectuées sur des séries temporelles de NDVI.

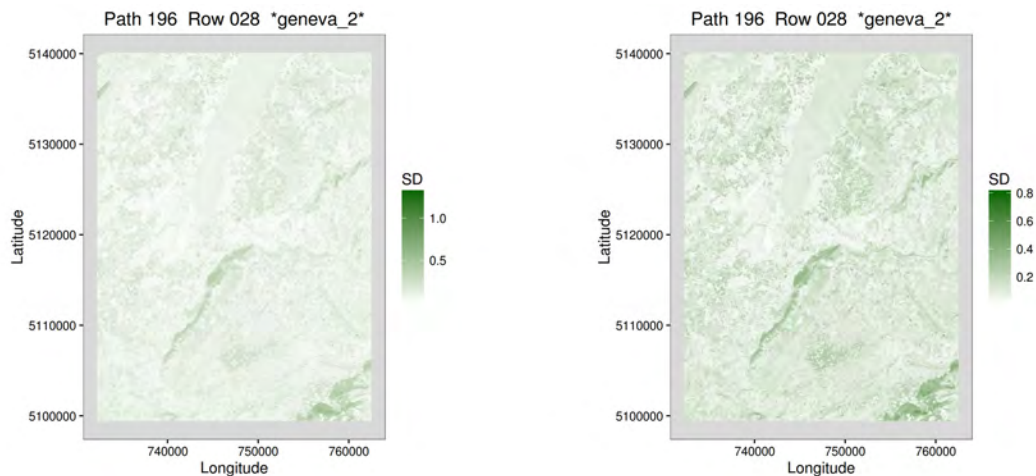
## 7.3 Exemple

### 7.3.1 Exécution

Le script *ndvi.py* a été exécuté sur une série de 8 scènes du site *geneva\_2* aux couvertures nuageuses variables (période avril 2013 - novembre 2015) (La liste des différentes scènes utilisées figure dans la Section 1 de la fiche de résumé de l’analyse, Annexe E.4). Tous les arguments possibles ont été ajoutés de manière à obtenir une vision d’ensemble des différents output possibles. Le temps total nécessaire à l’exécution du script a été de 3’648 secondes (1.4 million de pixels) et les temps d’exécution de chaque analyse figurent dans l’Annexe E.4<sup>4</sup>.

### 7.3.2 Déviation standard des valeurs de NDVI

L’argument optionnel (-v) permet d’obtenir un raster représentant le degré de variation du NDVI au cours du temps. Une déviation standard des valeurs de NDVI pour chaque pixel est calculée sur la base des 8 scènes. Ceci permet entre autre d’identifier les zones où la végétation n’est pas stable, comme notamment les zones agricoles (Figure 28).



(a) Données manquantes non-traitées. (b) Données manquantes comblées par interpolation linéaire.

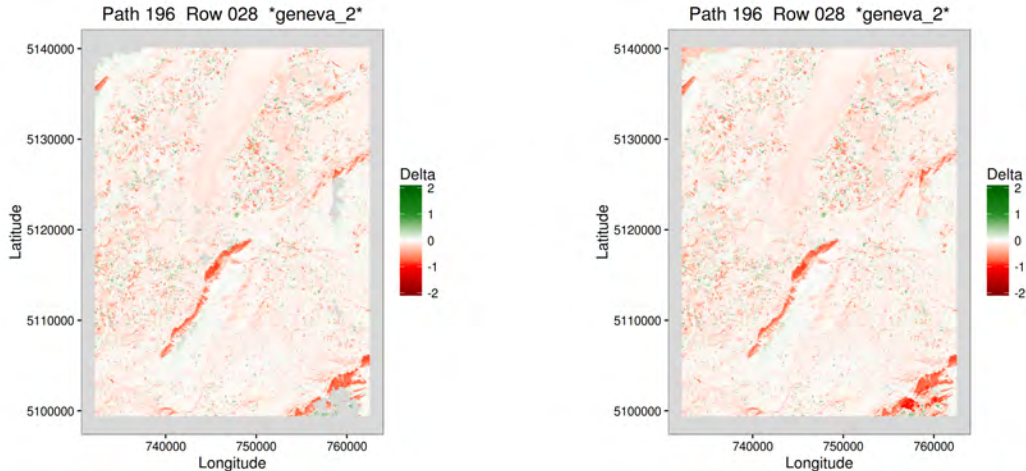
FIGURE 28 – Déviation standard des valeurs de NDVI pour le site de *geneva\_2*.

Le raster identifie les pixels où le NDVI varie, notamment de manière saisonnière (ex. les champs de culture). De même, les zones qui ne présentent pas de variation (zones urbaines) sont clairement identifiées. Il est intéressant de voir que le Lac Léman présente un certain degré de variation, probablement dû au cycle de croissance des algues. Il est à noter que la face orientée nord-ouest du Salève présente des valeurs de déviation standard relativement élevées. Ceci est probablement causé par l’ombrage, lequel oscille et fausse les valeurs de NDVI.

4. Les temps d’exécution sont relatifs et dépendent des performances de l’ordinateur

### 7.3.3 Changements potentiels sur le long terme

Un autre argument optionnel (-d) permet d'obtenir un raster représentant les différences observées entre la première et la dernière scène afin de pouvoir identifier les possibles changements sur le long terme (urbanisation, désertification, etc.) (Figure 29).



(a) Données manquantes non-traitées. (b) Données manquantes comblées par interpolation linéaire.

FIGURE 29 – Différences des valeurs de NDVI entre la première et la dernière scène pour le site de *geneva\_2*.

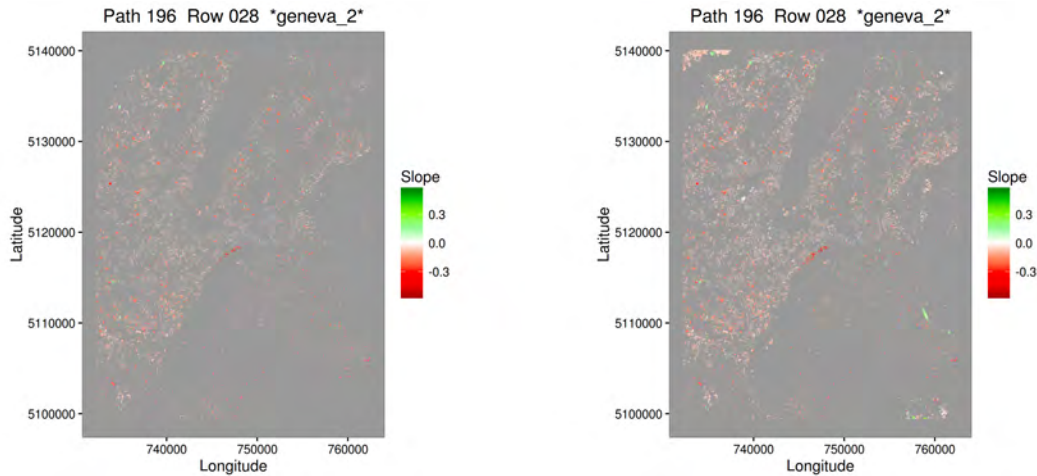
Dans notre cas, ceci n'a pas beaucoup de sens étant donné que la première scène et la dernière scène ne correspondent pas à la même période de l'année (avril vs novembre). Par ailleurs, Genève et ses alentours ne représentent probablement pas le meilleur exemple de changements substantiels en terme de végétation.

### 7.3.4 Coefficients significatifs de pente

Un autre argument optionnel (-s) permet d'obtenir un raster représentant les coefficients significatifs de pente ( $p$ -value  $< 0.05$ ) issus des régressions linéaires effectuées sur les valeurs de chaque pixel. Ceci offre la possibilité d'appréhender les tendances pour chaque zone du site et d'identifier les possibles changements sur le long terme (urbanisation, désertification, etc.) de manière relativement fiable (Figure 30). Les valeurs de pente sont relatives à 365 jours.

Globalement, les valeurs de NDVI ont eu tendance à légèrement diminuer entre avril 2013 et novembre 2015. Les zones rouges représentent probablement, pour la plupart, des zones nouvellement urbanisées.





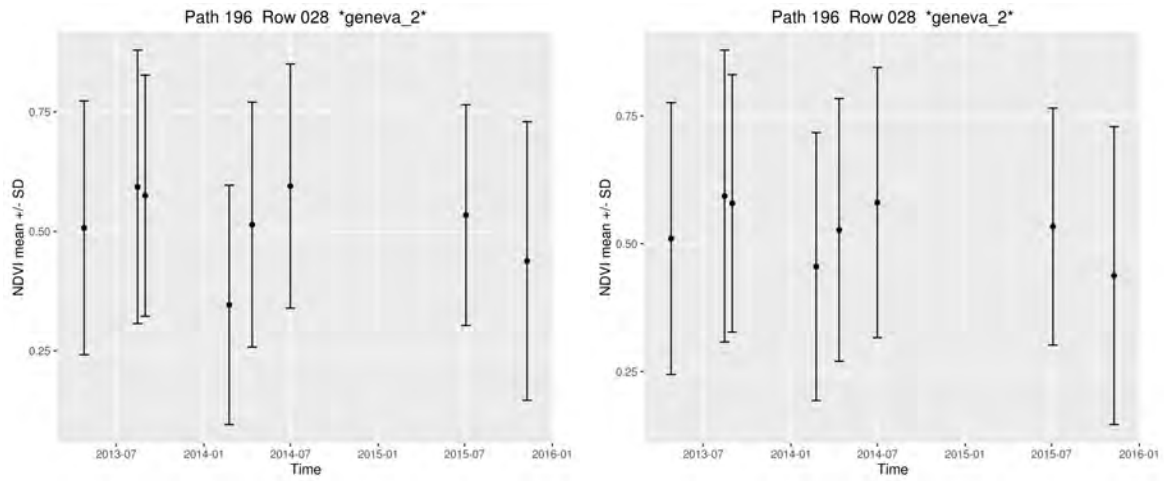
(a) Données manquantes non-traitées. (b) Données manquantes comblées par interpolation linéaire.

FIGURE 30 – Coefficients significatifs de pente ( $p$ -value  $< 0.05$ ) extraits de la régression linéaire effectuée sur chaque pixel pour le site de *geneva\_2*.

### 7.3.5 Évolution de la distribution des valeurs de NDVI

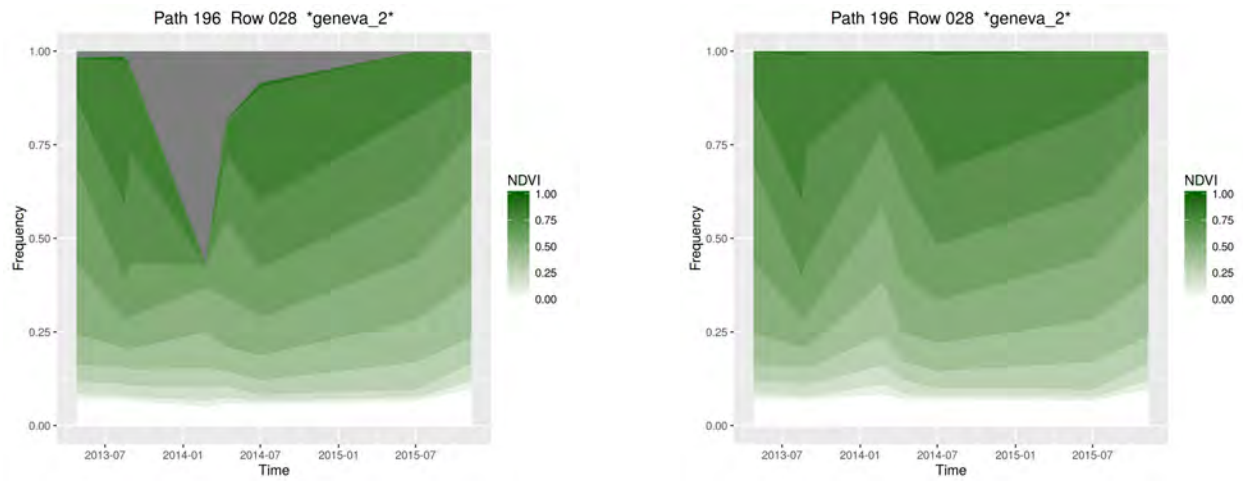
L'argument optionnel (-p) permet la création d'un graphique illustrant l'évolution de la distribution des valeurs de NDVI, afin d'observer les tendances globales potentielles sur un site donné (Figure 31). L'espacement sur l'axe des abscisses permet d'appréhender la dimension temporelle de l'analyse en général. Pour cet exemple, les scènes présentant le moins de nuages possible ont été retenues, ce qui fait que l'observation n'est pas régulière dans le temps. Cependant, il est intéressant de noter que les oscillations saisonnières des valeurs de NDVI sont clairement illustrées par ce graphique.

Une autre argument optionnel (-a) permet d'illustrer l'évolution de la distribution des valeurs de NDVI par un graphique en aires, représentant la proportion de 10 classes de valeurs (Figure 32). Ce graphique permet d'observer l'évolution d'une classe de valeurs de NDVI déterminée au cours du temps. Il est à noter que la proportion des classes de valeurs élevées (végétation importante) oscille avec le temps (à cause des saisons), tandis que la proportion des classes de valeurs plus faibles tend à être plus constante. Etant donné que le graphique offre une vision continue des valeurs, il est important que les intervalles soient **i)** relativement court et **ii)** relativement constants.



(a) Données manquantes non-traitées. (b) Données manquantes comblées par interpolation linéaire.

FIGURE 31 – Evolution de la distribution des valeurs de NDVI pour le site de *geneva\_2*.



(a) Données manquantes non-traitées. (b) Données manquantes comblées par interpolation linéaire.

FIGURE 32 – Graphiques en aires de la proportion des classes de valeurs de NDVI pour le site de *geneva\_2*

### 7.3.6 Fiabilité des résultats

Finalement, avec l'argument optionnel (-m) on obtient un raster permettant d'estimer le poids de l'interpolation ou des données manquantes dans les différents rasters en output. En effet, la valeur de chacun de ses pixels représente le pourcentage du nombre de fois où ceux-ci ont été masqués à cause de la présence de nuages ou de leurs ombres (Figure 33).

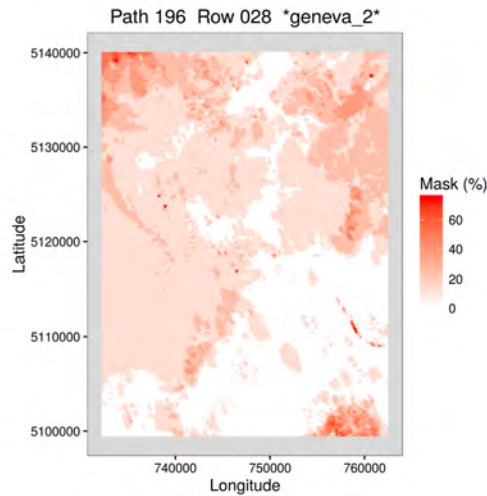


FIGURE 33 – Pourcentage du nombre de fois où un masque est appliqué.

Dans cet exemple, plus les zones sont foncées, plus l'interprétation concernant l'évolution des valeurs de NDVI devrait être réalisée avec prudence.

### 7.3.7 Résumé PDF

Une fiche de résumé en format PDF est automatiquement générée à la fin du processus dans laquelle figurent un tableau des données initiales, les différents output sous forme d'images et de graphiques, ainsi que la liste des output générés par le script et leur emplacement. Le document PDF automatiquement généré pour l'exemple utilisé ici se trouve dans l'Annexe E.4.

### 7.3.8 Discussion et conclusion

Cet exemple a permis de voir les différents output complémentaires produits par le script *ndvi.py*, lesquels décrivent tous, pour un site donné, un aspect spécifique de l'évolution du NDVI au cours du temps. L'interprétation des résultats doit prendre en compte la qualité initiale des images, et le raster où la valeur de chaque pixel correspond au pourcentage du nombre de fois où celui-ci a été masqué peut être utilisé afin d'estimer la fiabilité des données en output. La génération automatique d'une fiche de résumé en format PDF permet d'obtenir un aperçu rapide des analyses effectuées et des résultats. Le choix des output dépend des besoins de l'utilisateur et c'est ainsi que, par exemple, le raster illustrant les variations entre la première et la dernière scène n'a de sens que si celles-ci correspondent à la même saison. Aussi, étant donné que certains output requièrent un temps de calcul relativement important

(ex. le raster représentant les coefficients significatifs de pente issus des régressions linéaires effectuées sur chaque pixel) il est recommandé, en fonction de la taille des images, de ne produire que les output répondants aux besoins de l'utilisateur. Certains traitements ont pris plus de 10 fois plus de temps pour le site de *dubti* (environ 14 millions de pixels) que pour le site de *geneva\_2* (environ 1.4 million de pixels). Aussi, selon la taille des images et leur nombre, la mémoire RAM nécessaire peut-être relativement importante. Lors d'un test plus conséquent (série de 14 images de 5.3 millions de pixels chacune), un espace d'échange (swap) a dû être créé afin de pallier le manque de mémoire (8 Go installé) (Annexe E.5). Évidemment, le temps de calcul dépend des performances de l'ordinateur utilisé, mais une optimisation du script pourrait être envisagée, notamment en parallélisant certains processus (par exemple les analyses tenant compte de l'interpolation et celles n'en tenant pas compte).

Finalement, il a été vu comment, selon les besoins de l'utilisateur, de la qualité des images initiales et des performances des ordinateurs à disposition, l'automatisation des processus permet d'obtenir facilement des output complémentaires offrant la possibilité d'appréhender le comportement du NDVI pour un site donné, d'un point de vue spatial et temporel.

## Conclusion personnelle

L'expérience de ce stage au sein du GRID a rempli toutes mes attentes de tout point de vue (académique, professionnel et personnel). J'ai découvert Linux et le shell Unix Bash, j'ai approfondi mes connaissances de GRASS et du langage Python, et j'ai découvert toute la flexibilité des solutions open sources permettant de combiner différents langages et applications. J'ai aussi découvert un des aspects potentiels et positifs de l'open source, celui de la communication et de la solidarité. En effet, l'auteur principal de Zhu *et al.* (2015) et développeur de la fonction Fmask a répondu à plusieurs de mes questions, notamment au sujet du problème lié aux bandes TIRS. Aussi, après avoir insisté auprès de l'éditeur du package R 'raster', Robert J. Hijmans, au sujet d'une erreur de la fonction *approxNA* exécutée sur des très grandes images, celui-ci l'a corrigée (version 2.5-12). J'ai aussi eu du plaisir à observer comment mon travail a pris forme avec le temps et en fonction des différentes solutions trouvées. Il faudra cependant être attentif aux innovations et évolutions relatives au domaine de la télédétection. Par exemple, il est prévu qu'une version C de Fmask soit utilisée comme nouvel algorithme pour l'identification des nuages de la bande QA<sup>5</sup>, ce qui impliquerait une économie de temps conséquente liée au pré-traitement des images. Finalement, l'atmosphère de travail au sein du GRID m'a paru très agréable et motivante, et m'a donné envie de continuer d'acquérir de l'expérience dans le domaine de la géomatique en lien avec l'environnement.

---

5. New Collection 1 products. <http://landsat.usgs.gov//landsatcollections.php>, consulté le 28 juin 2016