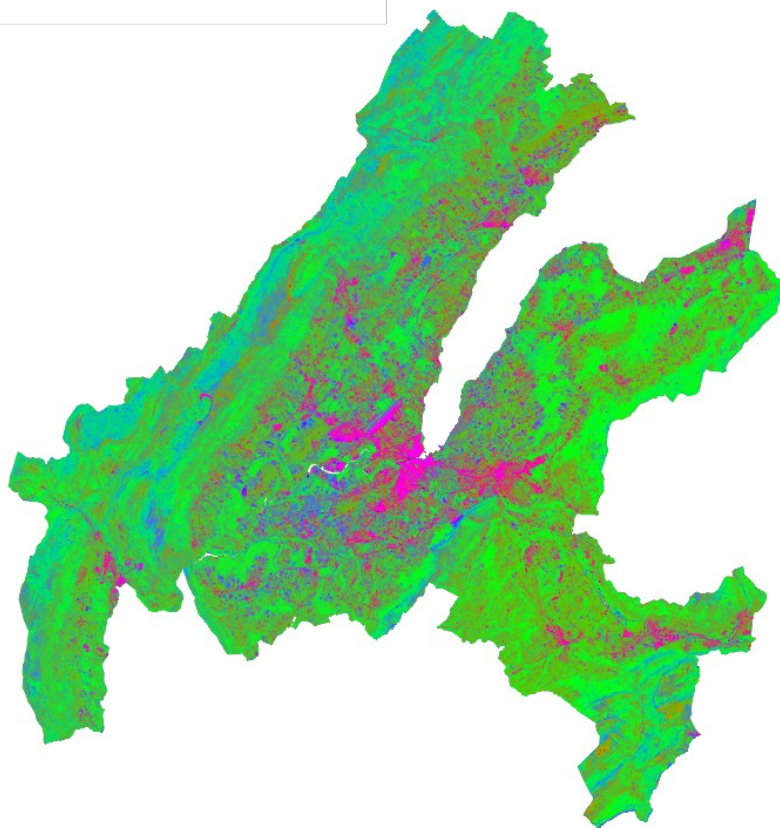


ESTIMATION DU FRACTIONAL COVER DANS LE GRAND GENEVE 1993 - 2023



Mémoire Certificat en Géomatique 2023 – 2024

Macarena Ramos López

Table of Contents

RESUMÉ.....	3
1 INTRODUCTION.....	4
1.1 Le Grand Genève.....	4
1.2 Couverture du sol.....	5
1.3 Télédétection.....	5
1.4 Fractional Cover.....	6
1.5 L'interprétation du modèle.....	7
2 CONCEPTS THEORIQUES.....	9
2.1 Les données Landsat.....	9
2.1.1 Landsat 5 et 7.....	9
2.1.2 Landsat 7 et 8/9.....	9
2.2 Le modèle du JRSRP.....	11
2.2.1 Les données.....	11
2.2.2 Le modèle de démixage.....	12
2.3 Démixage linéaire.....	13
2.3.1 Le modèle de mixicité linéaire.....	14
2.3.2 Dédution des Endmembers.....	16
2.3.3 Inversion des observations du terrain.....	16
2.3.4 Transformation des variables et validation croisée.....	17
2.3.5 Unmixing methodology.....	18
3 DONNEES.....	20
3.1 Le Swiss Data Cube.....	20
4 METHODOLOGIE.....	21
5 RESULTATS.....	24
5.1 Analyse visuelle.....	24
5.2 Time-series.....	27
5.3 Saisonnalité.....	28
5.4 Tendances.....	29
5.5 Animation 1993 -2023.....	30
6 CONCLUSION.....	31
ANNEXES.....	32
BIBLIOGRAPHIE.....	61

RÉSUMÉ

L'estimation du Fractional Cover de la végétation photosynthétique (PV), non photosynthétique (NPV) et de sol nu (BS) constitue un outil particulièrement intéressant pour la gestion des ressources naturelles et la modélisation du cycle du carbone, entre autres. Le Joint Remote Sensing Research Program (JRSRP) et Digital Earth Australia (DEA) ont développé un algorithme qui permet son calcul. Le modèle a été calibré et validé à partir des données de terrain collectées en Australie et appliqué ensuite en Australie et en Afrique avec des données Landsat et Sentinel, ainsi qu'à l'échelle globale avec des données MODIS. L'objectif de ce travail est d'explorer la validité de ce modèle pour le calcul du FC dans la région du Grand Genève à partir des images Landsat.

Pour cela nous avons travaillé avec le Swiss Data Cube et analysé les images Landsat des 30 dernières années. Ensuite la moyenne mensuelle et annuelle de chacune de 3 catégories du FC a été calculée. A partir de ces résultats nous avons généré des séries temporelles, évalué les tendances et la saisonnalité pour chaque type de couverture. Nous observons une diminution des fractions de PV et de NPV, tandis que la fraction de BS augmente. L'analyse visuelle confirme la pertinence du FC pour détecter des changements tels que l'urbanisation ou la sécheresse.

L'algorithme du JRSRP semble bien fonctionner sur le territoire genevois. Les valeurs de FC obtenues correspondent aux unités du paysage, permettant de distinguer les zones urbanisées, forestières, agricoles ou rocheuses. Cependant, ces résultats demeurent qualitatifs et nécessitent une validation sur le terrain pour que ces estimations puissent avoir une valeur quantitative.

1 INTRODUCTION

1.1 Le Grand Genève

Dans ce travail nous allons estimer le FC pour la région du Grand Genève. Le Grand Genève¹ est une agglomération transfrontalière entre France et Suisse, composée de trois entités principales, à savoir le canton de Genève, le district de Nyon (canton de Vaud) et le pôle métropolitain du Genevois français. Cette région est constituée d'une superficie (hors lac) de 1 996 km².

Il nous semblait pertinent de ne pas limiter notre étude à des frontières institutionnelles, mais plutôt de l'élargir à la réalité économique et géographique naturelle du terrain. L'espace occupé par le Grand Genève est celui intégrant la cuvette formée par la topographie (Salève, Jura, Voirons, Vuache) et le réseau hydrographique affluent du Rhône. Les partenaires du Grand Genève travaillent ensemble dans les domaines de la transition écologique, la mobilité et l'aménagement du territoire, et de ce fait il est nécessaire d'avoir des données pour l'ensemble de cette région plutôt que par canton ou pays.

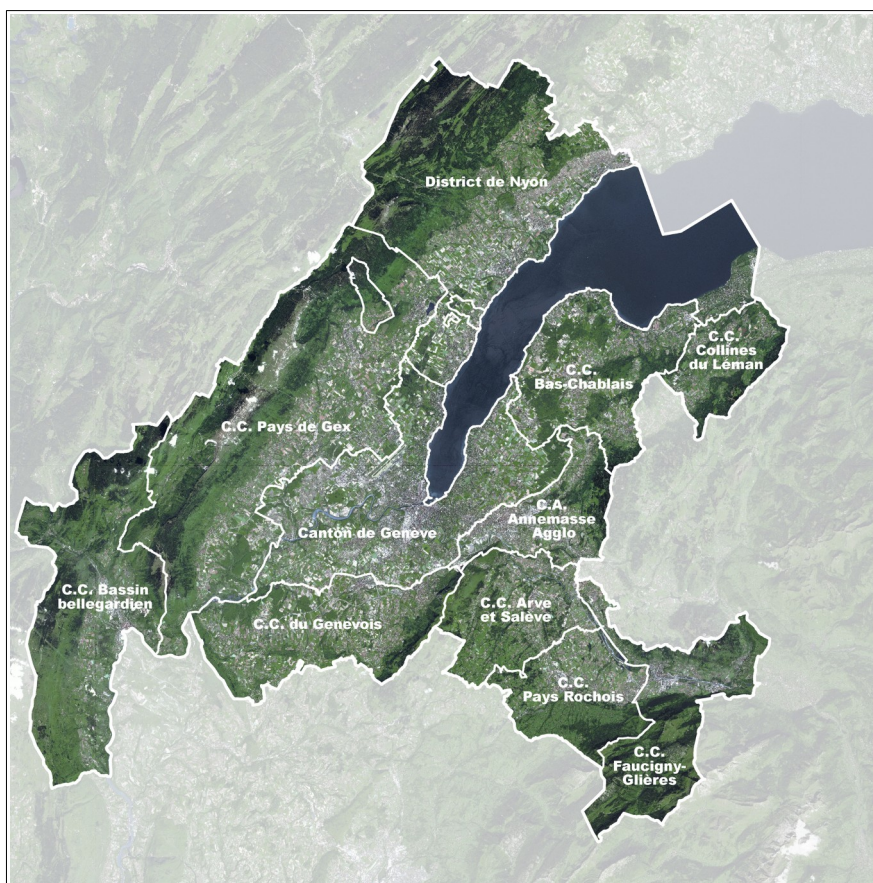


Figure 1: L'agglomération du Grand Genève. Source : www.grand-geneve.org

1 fr.wikipedia.org

1.2 Couverture du sol

La couverture du sol fait référence aux matériaux vivants et non-vivants (végétation photosynthétique et non photosynthétique) qui sont en contact avec la surface du sol. Son rôle est crucial dans le paysage, influant directement l'infiltration de l'eau, le ruissellement et l'érosion. De manière indirecte, elle a des répercussions significatives sur les plans économique, notamment dans le secteur agricole, et écologique, touchant à la biodiversité, à la productivité des écosystèmes, autant qu'à notre bien-être.

Les interventions humaines ou les changements naturels, tels que la météorologie et le climat, exercent une influence sur la couverture du sol. Avoir des données cohérentes et compréhensibles sur la couverture du sol, tant dans l'espace que dans le temps, est essentielle pour la prise de décisions politiques éclairées, ainsi que pour la surveillance de l'évolution de notre environnement afin de préserver et gérer de manière durable les différents écosystèmes tant humains que naturels. La télédétection offre une solution efficace et rentable pour surveiller à long terme les changements dans la couverture du sol sur de vastes étendues.

Si l'on exclut les régions permanentement recouvertes de neige et de glace, la surface terrestre peut être décrite en termes simples par la couverture fractionnelle de feuilles vertes qui fixent le carbone par photosynthèse (végétation photosynthétique - PV), le matériel végétatif non vert qui est principalement constitué de matériaux structurels végétaux tels que la cellulose et la lignine (végétation non photosynthétique - NPV), et le sol et la roche nus (sol nu - BS) (Hill et Guerschman, 2021). Les travaux de Small (2004) et de Scarth et al. (2010) prouvaient déjà qu'un espace spectral de démixage tridimensionnel est déjà suffisant pour modéliser la réflectance de la surface dans une grande majorité des cas.

La dynamique de ces fractions est influencée par les schémas saisonniers de précipitations et de température, se traduisant par des modèles variés de croissance, de sénescence, de dormance et de régénération à travers le monde (Foley et al., 2000).

1.3 Télédétection

Il existe des nombreuses méthodes pour analyser la quantité de végétation photosynthétique en télédétection, comme le Normalized Difference Vegetation Index (NDVI) ou le Leaf Area Index (LAI). Grâce au spectre d'absorption de la végétation, qui reflète fortement la bande du proche infra-rouge dû à la présence de chlorophylle, il est relativement facile d'identifier la fraction de PV par rapport aux fractions de NPV et BS. Cependant, la quantification de la végétation non photosynthétique et des surfaces de sol nu est plus complexe. L'un des défis posés par l'extraction d'informations sur la couverture de la NPV à partir de données de télédétection est la similarité spectrale avec le sol, où la variance spectrale de ces deux composantes se chevauche (Okin, 2013).

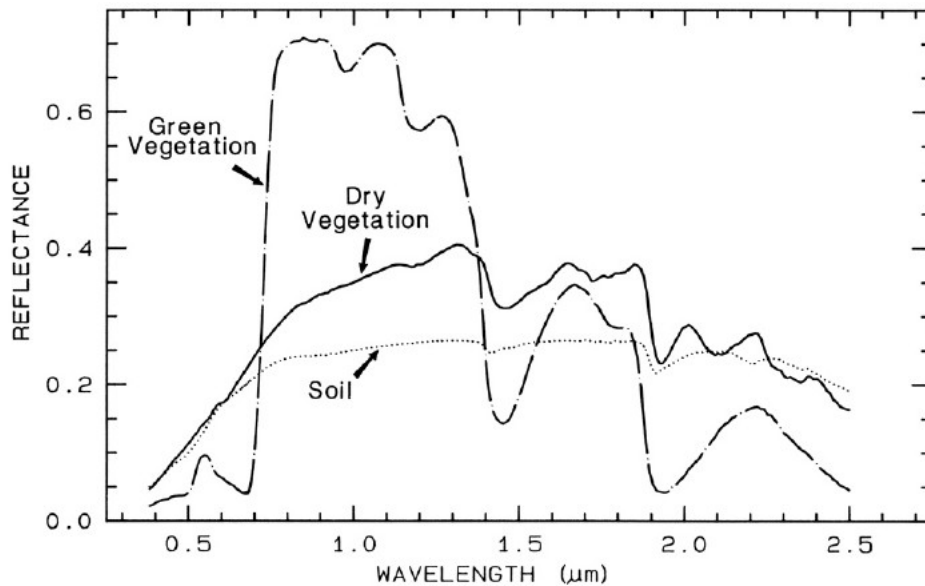


Figure 2: Signature spectrale PV, NPV et BS. Source: CLARK, R. 1999, *Spectroscopy of Rocks and Minerals, and Principles of Spectroscopy, Remote Sensing for the Earth Sciences*.

La végétation non photosynthétique joue un rôle écologique important dans le contrôle de l'absorption du carbone (C), de l'eau et des nutriments (Nagler et al., 2003, Li et al., 2015). La NPV est également reconnue pour être un indicateur pertinent des conditions de combustible et du stress dû à la sécheresse (Roberts et al., 2003, 2006). Les sols nus sont également un élément crucial des écosystèmes terrestres, avec des influences importantes nettement dans les phénomènes d'érosion et d'albédo (Balling, 1988).

Estimer les proportions de ces deux éléments permettrait de mieux comprendre la dynamique des écosystèmes et d'améliorer leur gestion. Ces analyses revêtent une importance particulière pour la gestion des zones d'intervention humaine, telles que les zones agricoles ou d'élevage. Le FC permet une analyse plus approfondie et une comparaison des effets que les différents types de gestion ont sur la couverture des sols.

1.4 Fractional Cover

L'analyse Fractional Cover est une modélisation de la mixité, signifiant que nous ne produisons pas des images classifiées avec des catégories, mais plutôt des séries de cartes qui fournissent la concentration des différents types de couverture du sol dans l'image. Nos résultats sont des données quantitatives et non thématiques. Cette approche est particulièrement intéressante lorsque la taille de nos pixel au sol est comparable ou plus grande que la taille naturelle des unités de couverture du sol (Settle & Drake, 1993).

Initialement, l'indice d'absorption de la cellulose (CAI) a été proposé comme méthode pour calculer le NPV (Nagler et al., 2003), mais cette approche repose sur plusieurs bandes spectrales relativement étroites dans l'infrarouge à ondes courtes (SWIR) qui sont généralement fournies par l'imagerie hyperspectrale. La bande correspondant au calcul du CAI est absente dans les images

multispectrales et cet approche ne permettait pas de calculer la proportion de NPV avec les images de MODIS ou Landsat. Une version améliorée utilisant des images multispectrales fut développée par Guerschman et al. en 2009. Ils ont constaté qu'une combinaison de NDVI et un ratio de bandes de réflectance de MODIS pouvaient être empiriquement calibrés par rapport aux valeurs CAI pour produire des séries temporelles de la couverture NPV qui s'accordaient avec les données de terrain. En 2010 Scarth et al. développèrent un modèle de démixage spectral linéaire pour les données Landsat, rendant possible le calcul du FC à une meilleure résolution.

1.5 L'interprétation du modèle

Le FC produit une image raster à 3 bandes, une pour chaque catégorie de couverture du sol.

- Végétation photosynthétique (PV): feuilles, herbe, cultures
- Végétation non photosynthétique (NPV): branches, herbe sèche et végétation morte)
- Sol un (BS): sol ou roche.



Avec ces informations, nous pouvons obtenir une image en faux couleurs du FC avec le PV dans la bande verte, le NPV dans la bleu et le BS dans la rouge. Les pixels ainsi représentés ne correspondent pas à une seule catégorie comme dans une carte de classification, sinon que nous montrent la proportion de chaque catégorie qu'ils contiennent. Les différentes combinaisons de couverture du sol possibles sont représentés par les matisse dans les mélanges de ces trois couleurs.

Les applications du FC sont nombreuses dans les domaines de l'environnement et l'agriculture, nettement dans la:

- surveillance de l'érosion de sols
- modélisation de processus de la surface du sol
- gestion de la terre
- études de la végétation
- estimation de la charge en combustible d'un sol
- modélisation des écosystèmes
- mapping de la couverture du sol

Actuellement il existent des données du FC au niveau mondial obtenues des images MODIS à une résolution de 250m. Des études à plus petite échelle, à partir des images Landsat et Sentinel ont été faits en Australie et en Afrique. Actuellement un projet est mené par l'Université de Genève pour étudier le FC dans le Canton de Vaud.

Avoir un historique des données du FC au niveau national et régional permettrait à la Suisse et/ou à Genève une meilleure compréhension des dynamiques de ses écosystèmes et systèmes de production en analysant non seulement la végétation active, mais aussi la végétation non photosynthétique et les proportions de sol nu. L'objet de ce travail est d'appliquer le modèle développé par Scarth à des images Landsat des 30 dernières années dans la région du Grand Genève, et d'analyser visuellement la validité des valeurs de FC obtenus. Néanmoins les valeurs obtenues nous permettront seulement d'analyser la tendance du FC. Pour que les estimations produites puissent avoir une valeur quantitative, il serait nécessaire une validation du modèle avec des données de terrain.

2 CONCEPTS THÉORIQUES

2.1 Les données Landsat

2.1.1 Landsat 5 et 7

Le satellite Landsat 5 TM a été lancé en mars 1984 et le Landsat 7 ETM+ en avril 1999 atteignant son orbite opérationnelle à la fin du mois de juin 1999. Le cycle de répétition de 16 jours du Landsat 5 TM et du Landsat 7 ETM+ étaient décalé de 8 jours l'un par rapport à l'autre (Arvidson et al., 2006).

Contrairement au Landsat 7 ETM+, le Landsat 5TM n'a pas d'enregistreur de données embarqué et a donc utilisé le système TDRSS (Tracking and Data Relay Satellite System) pour transmettre aux archives Landsat de l'USGS les données collectées en dehors des États-Unis. L'antenne TDRSS de Landsat 5 TM est tombée en panne en 1987 et la transmission des données s'est limitée à la transmission directe et en temps réel en bande X. Seules les données Landsat 5 TM captées dans la ligne de visée d'une station terrestre américaine ont été copiées dans les archives de l'USGS. Les stations terrestres internationales non américaines collectaient des données Landsat 5 TM via la transmission en bande X, mais seule une partie de ces données a été copiée dans les archives de l'USGS (Chander et al., 2007). Après le lancement de Landsat 7 ETM+ en 1999, certaines stations terrestres internationales ont transféré leur réception de Landsat 5 TM à Landsat 7 ETM+. Lorsque le système de correction de la ligne de balayage de Landsat 7 ETM+ est tombé en panne en mai 2003, plusieurs stations terrestres internationales ont cessé d'acquérir des données Landsat 7 ETM+ et sont passées à Landsat 5 TM. En effet, en mai 2003, le correcteur de la ligne de balayage de Landsat 7 ETM+ est tombé en panne, réduisant de 22 % les données utilisables dans chaque scène Landsat ETM+ (Markham et al., 2004).

Les résultats présentés par Kovalsky et Roy (2013) démontrent l'utilité de combiner les flux de données Landsat 5 TM et Landsat 7 ETM+ afin de tirer parti de leurs différents modèles d'acquisition et d'atténuer l'impact délétère de la défaillance de la ligne de balayage de Landsat 7 ETM+ en 2003.

En plus de fournir un nombre mensuel moyen d'acquisitions plus élevé, la combinaison de capteurs offre une plus grande couverture d'acquisition, avec un pourcentage de 1,7 % à 14,4 % plus élevé d'emplacements terrestres mondiaux acquis mensuellement par rapport à la prise en compte des données Landsat 7 ETM+ seules.

2.1.2 Landsat 7 et 8/9

Les instruments TM (*Thematic Mapper*) et ETM+ (*Enhanced Thematic Mapper*) utilisés à bord des satellites Landsat 5 et 7 échantillonnent les mêmes régions du spectre. Bien que présentant des fonctions de réponse spectrale relatives légèrement différentes, ils sont souvent considérés comme équivalents. Le capteur OLI (*Operational Land Imager*), équipant les satellites Landsat 8 et

9, a été conçu pour être en continuité avec ETM+ et TM, mais avec certaines améliorations. Notamment, les bandes équivalentes dans les longueurs d'onde du proche infrarouge ont été considérablement réduites, visant à éviter certaines caractéristiques d'atténuation atmosphérique. Cela devrait permettre d'obtenir des mesures moins impactées par les variations atmosphériques, particulièrement celles dues à la vapeur d'eau.

Par conséquent, les capteurs Landsat 8/9 OLI présentent des courbes de réponse spectrale relatives différentes de celles des capteurs Landsat 5 TM et Landsat 7 ETM+. De légères variations dans les mesures de réflectance peuvent donc se produire.

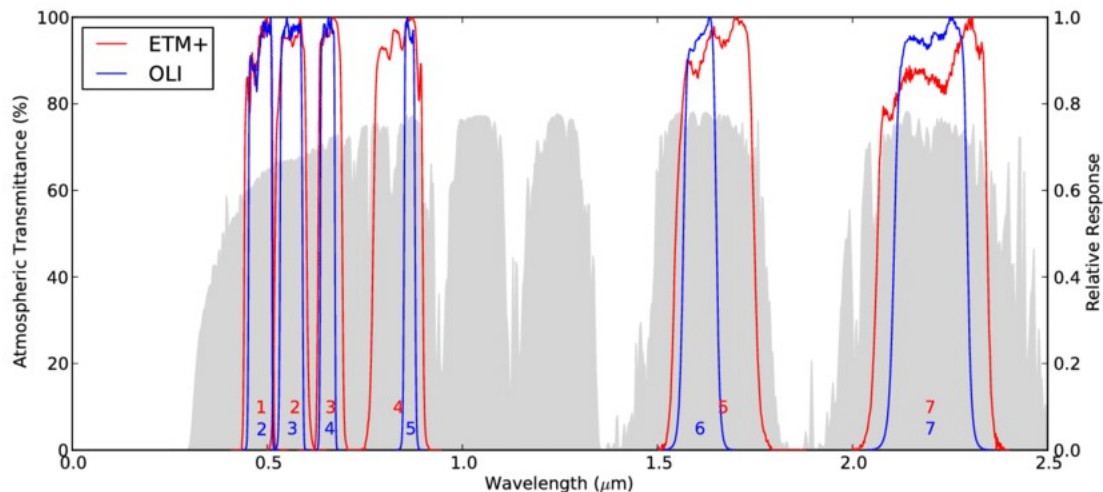


Figure 4: Réponses spectrales Landsat ETM et OLI. Source: Flood (2014). L'arrière-plan gris correspond à la transmittance atmosphérique pour les mêmes longueurs d'onde, conformément à l'axe de gauche.

En 2014, Flood, N. a mené une étude à grande échelle sur les différences entre ETM+ et OLI dans le paysage australien. Les disparités observées dans les réflectances mesurées par ces capteurs dépendent non seulement des réponses spectrales, mais également des pixels cibles. Il était donc essentiel d'évaluer ces différences dans la pratique.

L'analyse a été réalisée en termes de réflectance au sommet de l'atmosphère et en surface, ainsi qu'en termes de paramètres biophysiques modélisés à partir des spectres de réflectance respectifs. Trois modèles ont été analysés, incluant celui du FC que nous utilisons dans notre étude.

Les résultats de cette étude ont révélé de légères différences entre les capteurs, pouvant être amplifiées par la modélisation d'un paramètre biophysique. Bien que ces différences varient considérablement d'une bande à l'autre, la différence moyenne peut atteindre jusqu'à 6 % de la réflectance au sommet de l'atmosphère, et la réflectance de la surface peut être impactée jusqu'à 2 %. Ces disparités résultent des effets atmosphériques qui interagissent avec le changement de réponse spectrale de l'instrument. La correction atmosphérique est capable de prendre en compte ces différences directement.

Il a également été démontré qu'une partie de ces différences semble être systématique et peut donc être éliminée de manière fiable par des équations de régression. Les tests présentés dans

l'étude de Flood indiquent que la différence de réflectance entre les capteurs peut effectivement se traduire par une variation dans la variable biophysique modélisée. De plus, un ajustement systématique de la réflectance entraînera une amélioration de la continuité de la variable modélisée. Toutefois, ces résultats ne sont valables qu'en moyenne pour tous les pixels dans l'ensemble du paysage, et une composante plus localisée (probablement par pixel) de la différence entre les capteurs subsiste et n'est pas prise en compte par ce type de procédure.

Il est donc recommandé de prédire la réflectance ETM+ à partir de la réflectance OLI avant d'appliquer des modèles biophysiques à nos images. Ceci revêt une importance particulière lorsque les modèles ont été ajustés à des données historiques de terrain coïncidant avec l'imagerie ETM+, comme c'est le cas pour le FC. Dans l'étude de Flood, le modèle FC de Scarth a montré des différences allant jusqu'à 7 %, réduites à moins de 1 % en ajustant les réflectances.

2.2 Le modèle du JRSRP

Scarth et al (2010) ont développé le premier modèle de démixage spectral à être appliqué de manière systématique à l'imagerie multispectral à l'échelle continentale et mondiale. Ce modèle repose sur l'analyse de nombreuses données de terrain qui ont permis le calcul de *endmembers* synthétiques ainsi que des estimateurs nécessaires au modèle. L'algorithme permet d'obtenir à partir des images Landsat/MODIS des cartes du FC avec des résolutions spatiales respectives de 30 mètres et 500 mètres.

L'algorithme utilise un modèle perceptron multicouche (MLP) pour calculer les proportions de végétation photosynthétique, non-photosynthétique et de sol nu pour chaque image à partir de la réflectance de surface. Tensorflow a été utilisé pour entraîner le modèle MLP en utilisant la réflectance de surface de Landsat TM, ETM+ et OLI ainsi qu'une collection d'environ 4000 observations sur le terrain dans différents écosystèmes avec des environnements homogènes et hétérogènes.

2.2.1 Les données

Les observations sur le terrain utilisés pour l'entraînement et la validation de l'algorithme couvrent une grande variété de types de végétation, de sol et de climat à travers l'Australie. Elles furent collectées entre 1997 et 2018 en suivant la procédure décrite par Muir et al. (2011). Dans les régions de végétation naturelle, des mesures ont été prises tous les mètres sur trois transects de 100m qui se croisent au centre et se situent à 60° d'intervalle les uns des autres donnant une forme d'étoile. Des données ont été recueillies pour les trois strates de végétation, à savoir inférieure, intermédiaire et supérieure, si présents. Pour les zones agricoles semés en files parallèles seulement deux transects de 100m, placés à 45° de lignes de culture, ont été recensés.

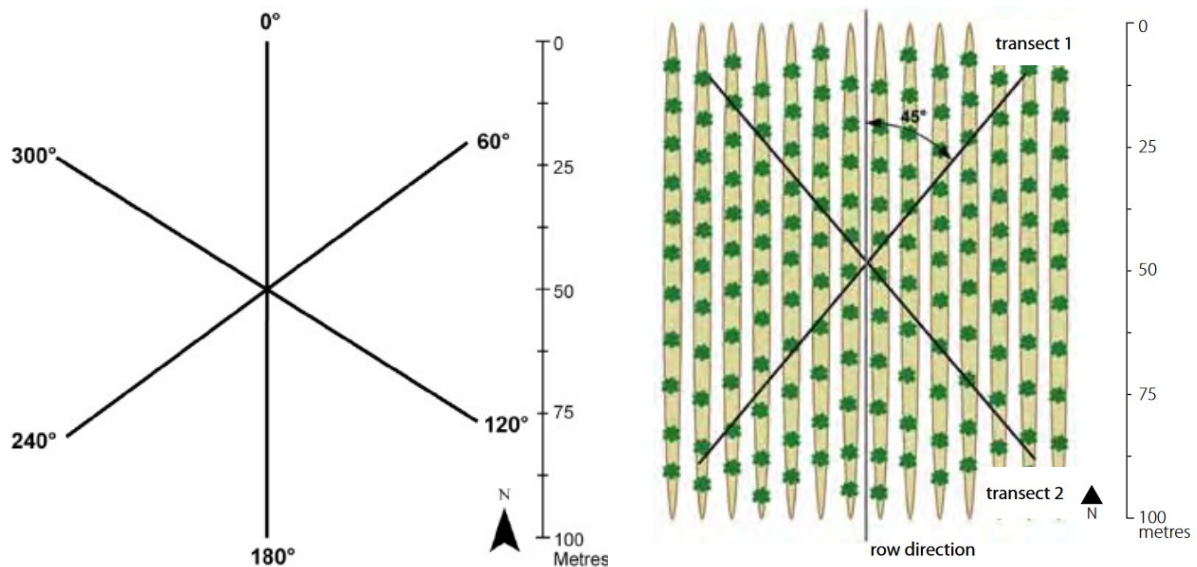


Figure 5: Disposition des transects dans les zones naturelles et agricoles. Source: Muir et al (2011).

La fraction de couverture du sol par strate est calculé en prenant le nombre des comptages de chaque type de couverture du sol et en le divisant par le nombre total de comptages (300 ou 200). Afin de comparer les observations sur le terrain avec les estimations dérivées des images satellites, la couverture plane exposée a été calculée en fusionnant la couche inférieure avec les couches intermédiaires et supérieures lorsque présentes.

Concernant les données satellite, une fois effectuées les correction géométriques, radiométriques, et le nettoyage des images pour éliminer les nuages, les ombres et les zones d'eau, le spectre des bandes 2 à 7 des satellites Landsat TM+ et Landsat ETM+ a été extrait pour des fenêtres de 3 x 3 pixels autour des sites où les données sur le terrain ont été collectées. Les images étaient affectées par l'erreur de la ligne de balayage de Landsat 7 n'ont pas été retenues. Seules les images ayant moins de 60 jours d'écart entre le moment des mesures sur le terrain et la date d'acquisition de l'image satellite ont été conservées. Une pondération a été appliqué en fonction du nombre des jours séparant la date d'acquisition de l'image de la date de la collecte de données sur le terrain. Une deuxième pondération a été ajouté pour prendre en considération l'hétérogénéité des sites. Plus le site est hétérogène, plus la certitude avec laquelle on peut établir une valeur moyenne de couverture diminue (Korb et al., 2003).

Le script développé par la DEA tient en compte les différences entre les capteurs TM+ de Landsat 7 et OLI de Landsat 8 et 9 et un facteur d'ajustement de la bande spectrale est appliqué aux données Landsat 8 et 9 pour les rendre plus proches de la réflectance mesurée par Landsat 7, comme décrit en détail dans Flood (2014). Cependant, il subsistera toujours une petite différence par pixel qui pourrait être une source de variabilité indésirable.

2.2.2 Le modèle de démixage

Le modèle démixage présenté ici se base sur un régression multiple des données de terrain par rapport aux image satellites afin de déduire la valeur des pixels purs (*endmembers*). Par pixel pur

on entend un pixel avec 100% de PV, NPV ou BS. Afin de modéliser certaines des non linéarités existantes, des bandes Landsat transformées logarithmiquement et de termes interactifs ont été ajoutées aux équations de régression. Bien que l'ajout de termes interactifs améliore le RMSE du modèle, des problèmes d'*over-fitting* peuvent survenir à un certain point. Un total de 55 bandes spectrales furent obtenus, avec des fortes corrélations entre plusieurs d'entre elles. Pour éviter les effets d'*overfitting* et garantir un modèle extrapolable à d'autres régions, une validation croisée a été utilisée pour déterminer le sous-espace optimal ainsi que la contrainte de somme égal à 1 optimale. Le RMSE minimal correspond à l'utilisation des 8 valeurs, ce qui est significativement supérieur aux 5 bandes originales des images Landsat.

Les *endmembers* synthétiques obtenus sont ensuite utilisés dans un modèle contraint de démixage par moindres carrés non négatifs (NNLS, *non negative least squares*) pour calculer les proportions de chaque *endmember* à partir des données spectrales des pixels.

Afin d'estimer la précision du modèle, les résultats de l'algorithme FC ont été comparés à 1565 sites de terrain qui n'ont pas été utilisés pour former le modèle FC. Sur la base de la comparaison avec ces données de terrain indépendantes, le produit FC présente une erreur quadratique moyenne globale (RMSE) de 11,9 %. Les marges d'erreur varient pour les trois couches différentes :

- RMSE du PV : 11,9%,
- RMSE du NPV : 17,1%
- RMSE du BS : 14,6%.

L'impact de l'humidité du sol sur la précision du produit FC et la similarité entre certains *endmembers* du sol nu et ceux de la végétation non photosynthétique peuvent conduire à l'instabilité du modèle. Les types/couleurs de sol non inclus dans les données d'entraînement du modèle peuvent également être sujets à des erreurs. Cependant, Guerschman et al. (2015) avait conclu qu'il n'y avait pas d'évidences significatives montrant que la couleur du sol ou l'humidité de celui-ci influenceraient la performance du modèle.

La diversité naturelle de l'Australie qui comprend des écosystèmes tels que des forêts tropicales, savanes, prairies tempérées, zones semi-arides et arides, forêts tempérées d'eucalyptus et forêts tropicales tempérées, ainsi que des vastes zones dédiées à l'agriculture ou à l'élevage, rend possible l'application du modèle dans le reste du globe (Hill et Guerschman, 2020). Cependant, les analyses ont été faites dans le continent australien et une calibration et validation sur le territoire genevois serait conseillé si le FC veut être utilisé avec une valeur quantitative.

2.3 Démixage linéaire

Le principe du *linear mixture modelling* ou modèle de démixage linéaire, est celui où les sorties d'un processus de combinaison linéaire sont mesurables, mais où les paramètres d'intérêt sont en fait les entrées. En télédétection par satellite, cette situation se produit lorsque la zone élémentaire de la surface terrestre imagée (un pixel) contient plusieurs composants différents de la

couverture fondamentale, par exemple, l'eau, le sol nu, l'herbe, la forêt, donnant ainsi un spectre mesuré qui est une combinaison des spectres des différents composants individuels.

La classification en 3 types de couverture (PV, NPV, BS) utilisée dans le FC, permet l'utilisation des techniques de démixage (*unmixing*). Dans ces techniques, la réflectance de chaque pixel est considérée comme une combinaison linéaire des proportions de surface de chaque type de couverture (Garcia-Haro et al, 1996). La définition de Keshava (2003) explique la procédure du *spectral unmixing* comme la décomposition de le spectre mesuré pour un pixel mixte en une collection de plusieurs spectres constitutifs ou *endmembers*, et ses correspondantes fractions ou abondances qui nous donnent les proportions de chaque *endmember* présent dans le pixel.

Le terme *endmember* implique que le pixel est pur, c'est à dire constitué d'une seule catégorie de couverture. Dans notre étude, 100% de sol nu, de végétation non photosynthétique ou de végétation photosynthétique. Ce terme de *endmember* est emprunté aux études géochimiques, et pour nous il désigne la réponse qui serait reçue en l'absence de bruit par un pixel composé d'une seule de nos catégories d'étude.

Il s'agit donc d'estimer pour chaque pixel de l'image, la proportion couverte par chacune de nos catégories à partir de l'observation multispectral. Pour cela, traditionnellement, on calcule un estimateur, une fonction de l'information multivariable pour ce pixel, à savoir la réflectance pour les bandes spectrales de notre image satellite. Nous cherchons à trouver un modèle qui nous permettrait de prédire quel serait le signal produit pour un pixel avec un certain mélange donnée de couverture du sol.

Cette fonction dépendra dans une situation idéal du spectre des types de couverture qui composent notre image. Cependant, la contamination atmosphérique et les mauvaise calibrations impliquent que nos données ne peuvent pas être simplement réduites aux valeurs de réflectance. Par conséquent, les réflectances de laboratoire ont une valeur limitée et il est nécessaire d'avoir des données de terrain sur les proportions de couverture du sol pour parvenir à calculer l'estimateur Settle (1993).

2.3.1 Le modèle de mixité linéaire

Ce modèle se base dans l'hypothèse selon laquelle chaque photon qui arrive au capteur à interagit avec un seul type de couverture de sol. On assume qu'il n'y a pas de réflexion multiple significative entre les différents types de couverture du sol. Dans ces conditions, l'énergie reçue par le capteur serait égale à la somme des énergies reçues par chaque type de composant de la couverture de la surface analysée. Chaque composant aura une réflectance caractéristique selon le type de couverture et serait proportionnel à la surface qu'il occupe. La somme de ces proportions doit être égal à 1.

Par exemple, si 25 % d'un pixel contient la catégorie A, 25 % la catégorie B et 50 % la catégorie C, le spectre de ce pixel est une moyenne pondérée de 0,25 fois le spectre de la catégorie A, plus 0,25 fois le spectre de la catégorie B, plus 0,5 fois le spectre de la catégorie C. Ainsi, étant donné le

spectre résultant (les données d'entrée) et les spectres des *endmembers*, le démixage spectral linéaire² résout les valeurs d'abondance de chaque *endmember* pour chaque pixel.

Sous l'hypothèse du mélange linéaire, la valeur du signal attendu pour un pixel mélangé peut être définie comme suit:

$$f_1\mu_1 + f_2\mu_2 \dots + f_c\mu_c = Mf \quad (\text{eq 1})$$

Dans cette équation, le vecteur μ_i représente le signal attendu pour des pixels qui contiendraient uniquement un type de couverture (i). Les quantités de ces vecteurs sont appelées les *spectres des endmembers* et les catégories elles-mêmes les *endmembers*. f_i dénote les proportions de chacun de nos *endmembers*.

A cette étape, le modèle linéaire peut être défini comme:

$$X = M F \quad (\text{eq 2})$$

Où X est une matrice n x b (nombre d'échantillons de terrains par nombre de bandes spectrales), F est une matrice n x c (nombre des catégories ou *endmembers*) et M est une matrice c x b des spectres des *endmembers*

La valeur des spectres des *endmembers*, au nombre de trois dans notre cas, définit le plan dans lequel se situent les mélanges de nos trois catégories. Tous les mélanges possibles, c'est à dire tous ceux qui n'ont pas des valeurs négatives, vont remplir le volume enfermé dans le polyèdre qui a comme sommets les points correspondants aux spectres des *endmembers* (Settle, 1993).

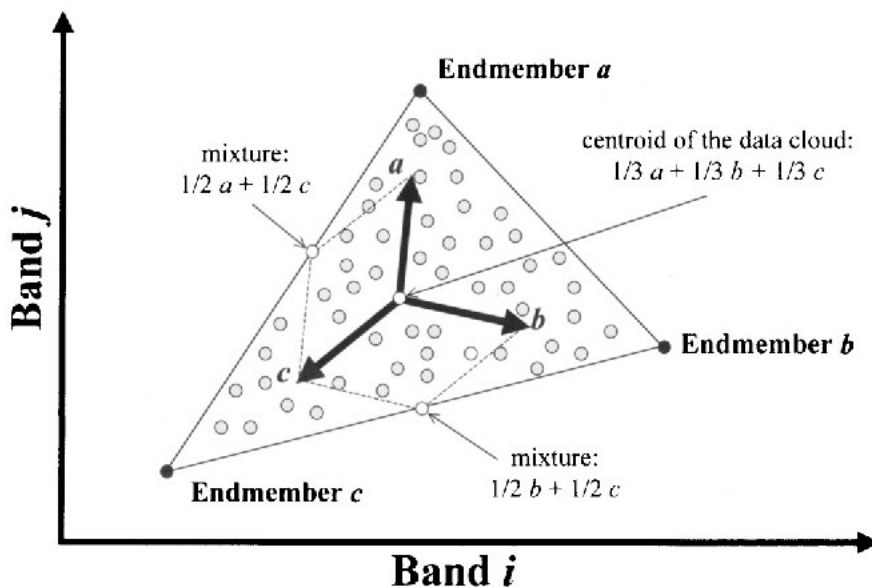


Figure 6: Espace spectral. Source: Goretta et Gomez, *Data Handling in Sciences and Technology*, vol. 30.

Afin d'estimer les proportions de chaque *endmember* (F) nous devons disposer des réflectances de nos données de terrain (X) ainsi que de la matrice des spectres des *endmembers* (M). Comme mentionné précédemment, l'utilisation des mesures de laboratoire n'est pas idéale pour

2 <https://www.nv5geospatialsoftware.com/docs/LinearSpectralUnmixing.html>

déterminer le spectre des *endmembers*. Par conséquent, il est nécessaire de déduire la matrice M des données de terrain.

2.3.2 Dédution des Endmembers

L'**estimateur classique** se base sur un modèle de mélange physique linéaire, l'un des modèles de démixage les plus couramment utilisés. Il estime le signal produit par un pixel mélangé en se basant sur un ensemble d'*endmembers*. L'hypothèse fondamentale de ce modèle est que les spectres des *endmembers* sont linéairement indépendants, ce qui implique que la matrice M (représentant les spectres des *endmembers*) est de plein rang et inversible.

En d'autres termes, chaque *endmember* contribue de manière linéaire et indépendante au spectre mesuré d'un pixel mélangé. Cela est exprimé mathématiquement par l'équation 2 :

$$X = M F \quad (\text{eq 2})$$

La valeur de M n'est pas connue mais nous pouvons obtenir X avec des données de terrain.

Settle (1996) était arrivé à la conclusion que la différence entre le modèle classique ou inverse pour estimer les proportions de couverture du sol était inférieur à l'erreur de prédiction si la mixité était linéaire et si les signatures étaient bien séparées. (3)

L'**estimateur inverse** repose sur une régression linéaire des fractions de couverture du sol par rapport au signal multispectral. Il cherche le meilleur modèle, qui soit une fonction linéaire du signal spectral, et qui permet de déterminer les proportions de chaque *endmember*. Pour cela nous devons définir un opérateur inverse A tel que :

$$A M = I \quad (\text{eq 3})$$

Où A est la matrice c (*endmembers*) x b (bandes spectrales) et I est la matrice d'identité b x b.

Cette définition permet l'inversion suivante :

$$F = X A \quad (\text{eq 4})$$

La matrice A est calculé à partir des données d'entraînement en utilisant des techniques de régression à plusieurs variables (Kalivas, 1999). Cette approche nécessite un jeu des données d'entraînement suffisamment varié pour tenir en compte l'hétérogénéité du paysage. De plus, il est aussi nécessaire que la matrice X (observations multispectrales x sites du terrain) ait un rang suffisant à l'estimation de A. Si ces conditions sont remplies, et si on assume une mixité linéaire, l'estimateur inverse peut être considérée comme une forme régularisée de l'estimateur classique. Cette approche permet l'inclusion de termes interactives pour modéliser les non-linéarités du système.

2.3.3 Inversion des observations du terrain

L'inversion des observations du terrain dans le contexte du modèle de démixage implique l'estimation de l'opérateur inverse A à partir des matrices des proportions des *endmembers* sur le

terrain (F) et des données multispectrales (X). L'opérateur inverse peut être obtenu par la solution des moindres carrés, comme indiqué dans l'équation suivante (Lawson, 1995) :

$$A = (X^T X)^{-1} X^T F \quad (\text{eq 5})$$

La méthode des moindres carrés est une procédure statistique qui vise à minimiser la somme des décalages entre les points de données et la courbe ajustée, offrant ainsi le meilleur ajustement pour l'ensemble des données.

Dans certaines situations, afin de modéliser les non-linéarités du modèle, des variables transformées seront utilisées. Dans ce contexte, une décomposition des valeurs singulières tronquées (TSVD, *Truncated Singular Values Decomposition*) sera utilisée pour éliminer les valeurs moins significatives et réaliser l'inversion dans un sous-espace inférieure plus restreint (Xu, 1998).

En définissant X^+ comme la pseudo-inverse de Moore-Penrose de X, nous pouvons obtenir une approximation de A par le biais des moindres carrés en inversant l'équation 4 de la manière suivante.

$$A \approx (X^+ F) \quad (\text{eq 6})$$

La pseudo-inverse de Moore-Penrose est une technique d'algèbre linéaire utilisée pour approximer l'inverse de matrices non inversibles. Elle est souvent utilisé pour trouver une solution de "meilleur ajustement" (moindres carrés) à un système d'équations linéaires dépourvu de solution.

Ensuite on peut estimer M en calculant la pseudo-inverse de Moore-Penrose de A, comme indiqué dans l'équation suivante :

$$M \approx A^+ \approx (X^+ F)^+ \quad (\text{eq 7})$$

Il est crucial de noter que cette approche peut devenir mal posé si les données d'entraînement ne couvrent pas suffisamment la gamme de couvertures du sol et leur variabilité spectrale. D'où l'importance de bien choisir le sous-espace pour l'inversion (Eldén, 2004).

Il convient de souligner que ce résultat est différent de l'inversion directe de l'équation 2 :

$$M \approx X F^+ \quad (\text{eq 8})$$

Car, dans ce cas, nous assumons que les données multispectral sont la variable indépendant.

Cette estimation de endmembers peut ensuite être utilisée dans un modèle de démixage contraint (*constrained unmixing model*), ce qui a des avantages importantes quand il est nécessaire d'extrapoler les résultats.

2.3.4 Transformation des variables et validation croisée

L'utilisation des méthodes de démixage linéaire simple peut être compliquée par la présence d'effets non linéaires, principalement dus à la réflexion multiple et à la transmission de surfaces. Pour tenir en compte les non-linéarités légères il est possible d'intégrer dans le modèle des transformations logarithmiques des bandes de réflectance de la surface et des termes interactifs. Ces transformations sont appliquées pour rendre les données asymétriques afin qu'elles se

conformement approximativement à la normalité. Dans le cadre du FC, les transformations logarithmiques de chaque bande, les produits de chaque bande avec les autres, les produits des transformations logarithmiques de chaque bande avec les autres, ainsi que les ratios normalisées de toutes les combinaisons de bandes ont été pris en compte.

Cependant, l'utilisation de variables transformées peut entraîner des modèles sujets à un sur-ajustement (*overfitting*), c'est-à-dire des modèles qui s'ajustent très bien aux mesures sur le terrain mais qui perdent leur pertinence en dehors de la zone de calibration, entraînant ainsi de grandes erreurs d'estimation. (Salvador and Pons, 1998). Pour éviter ces problèmes, des approches telles que la régression par étapes, où seul le sous-espace de variables offrant la meilleure explication est retenu (Grossman et al, 1996) sont adoptées. Scarth et son équipe, utilisent la méthode de troncation du sous-espace, dont laquelle la valeur de troncation est choisie au point où l'erreur quadratique moyenne (RMSE, *root mean square erreur*) est la plus faible, à l'aide d'une approche de validation croisée ten-fold (*ten-fold cross-validation*).

La validation croisée³ est une méthode permettant d'évaluer les performances d'un modèle prédictif en le testant sur des nouvelles données. Pour ce faire, une partie des données du jeu d'entraînement est mise de côté et non utilisée pour l'entraînement initial, mais plutôt pour tester et valider le modèle ultérieurement.

La technique K-Folds permet d'assurer que toutes les observations du jeu de données original sont incluses tant dans l'ensemble d'entraînement que dans l'ensemble de test. Pour ce faire, le jeu de données est séparé de manière aléatoire en K plis, puis le modèle est ajusté en utilisant K-1 plis. La validation du modèle est réalisée en utilisant le K-fold restant. Ce processus est répété jusqu'à ce que chaque K-fold ait été utilisé au sein de l'ensemble d'entraînement. La moyenne des scores enregistrés constitue la métrique de performance globale du modèle.

2.3.5 Unmixing methodology

Il existe plusieurs manières de spécifier l'équation de l'estimateur classique pour résoudre le problème inverse de récupération des fractions des *endmembers* à partir des informations spectrales du pixel.

Le (*linear spectral unmixing*) démixage spectral linéaire⁴ offre deux options de contrainte : le démixage sans contrainte ou le démixage avec contrainte partielle. Dans la méthode sans contrainte, les abondances peuvent prendre des valeurs négatives et ne sont pas contraintes à une somme égale à l'unité.

Dans leur travail Scarth et al. introduisent les contraintes suivantes dans les fractions:

- la somme des fractions doit être égale à 100 %
- les fractions ne doivent pas être négatives.

Des solutions directes sont disponibles pour l'équation sans contrainte et pour celle avec contrainte de la somme égale à un. Ils résolvent le problème des moindres carrés non négatifs

³ <https://datascientest.com/cross-validation>

⁴ <https://www.nv5geospatialsoftware.com/docs/LinearSpectralUnmixing.html>

(NNLS) en utilisant la stratégie itérative des ensembles actifs proposée par Lawson and Hanson (Lawson (1995)). Pour résoudre les deux contraintes, Scarth utilise l'algorithme NNLS avec une stratégie de pondération pour la contrainte de la somme à un qui optimise l'erreur des moindres carrés (Heinz and Chang, 2001). Cela résulte dans la modification de l'équation 1 de la façon suivante:

$$[X, \lambda] = [M, \lambda 1^T] F \quad (\text{eq 9})$$

Où λ est la pondération pour la contrainte de la somme égale à 1, avec une valeur fixée à 0.2 en base à l'étude de Scarth et al (2010). Et $1^T = [1 \ 1 \ \dots \ 1]$: un vecteur $c+1$ de uns.

Bien que leur étude ait révélé des différences minimales dans l'erreur quadratique moyenne (RMSE) entre le modèle sans contrainte et le modèle avec contrainte, il est crucial de noter que ces résultats indiquent que le modèle s'adapte bien aux données sur le terrain. Cependant, lors de l'extrapolation du modèle, il était anticipé que le modèle avec contrainte surpasse le modèle sans contrainte.

3 DONNÉES

3.1 Le Swiss Data Cube

Nous allons travailler avec le Swiss Data Cube qui fournit des données pour toute la Suisse depuis 1984. Les données sont actualisées quotidiennement et elles ont été transformées dans un format spatio-temporel cohérent et co-registrées, ce qui permet de passer directement à la phase d'analyse. Le Swiss Data Cube a été créé dans le but de fournir les données nécessaires pour aboutir des analyses environnementales plus rapidement et avec moins des connaissances scientifiques. Dans une période où les problèmes environnementaux doivent être détectés, analysés et résolus urgemment, le SDC vise à permettre des réponses plus efficaces aux problèmes nationaux. La technologie des EODCs (Observation Data Cubes) apparaît comme une solution technique pour gérer, accéder et analyser les données «EO».

Le SDC est construit sur la base du Open Data Cube software. Ce projet open source permet de cataloguer des grandes quantités de données EO, offre une interface de programmation en Python et permet également de retracer l'origine des données pour des contrôles de qualité ou des mises à jour. Le Swiss Data Cube permet d'accéder aux données provenant de l'imagerie optique de Sentinel-2, Landsat 5,7,8 et l'imagerie radar de Sentinel 1.

3.2 Les données satellite

Notre étude s'étalant de 1993 à 2023, nous utiliserons les images provenant des satellites Landsat. Malgré leur résolution spatiale et temporelle plus faible par rapport aux images des satellites Sentinel, les données de Landsat ont l'avantage de remonter suffisamment loin dans le temps pour nous permettre d'observer l'évolution du Fractional Cover sur une plus longue période. Concrètement, nous avons utilisé des images de Landsat 5 pour la période de 1993 à 2012 et Landsat 9 pour la période entre 2014 et 2023. Pour Landsat 7, nous avons retenu les images antérieures à l'erreur de la ligne de balayage, c'est à dire celles correspondant à la période de 1999 à 2003. Ensuite les images de 2013 ont été utilisées pour obtenir des images du FC de cette année, malgré l'erreur car nous ne disposons pas de suffisamment des données. Celles datant de la période 2003 - 2022, ont été triées manuellement et les images sans nuages ont été sélectionnées et traitées avec QGIS pour corriger l'erreur de balayage. Ces corrections, n'ayant pas été validées, les images seront uniquement utilisées dans le graphique de visualisation de l'évolution du FC afin de réduire le nombre de données manquants, mais elles ne seront pas utilisées pour le calcul des moyennes annuelles ni des tendances.

La zone du Grand Genève est contenu dans les scènes Landsat avec le Path/Row suivant : 196/028.

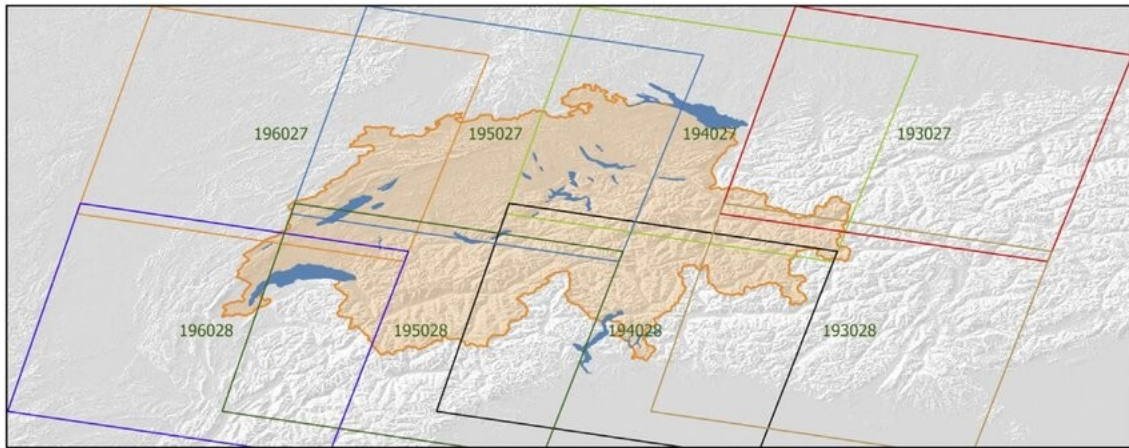


Figure 7: Landsat scenes coverage over Switzerland. Source : Building an Earth Observations Data Cube: lessons learned from the Swiss Data Cube (SDC) on generating Analysis Ready Data (ARD)

Les images Landsat que le SDC nous fournit sont des *Analysis Ready Data (ARD)*. C'est à dire des images calibrées radiométriquement et géométriquement, avec les corrections atmosphériques nécessaires et possédant des descriptions des métadonnées. Le SDC télécharge les données Landsat avec la plus grande qualité des données possible correspondant à la réflectance de surface. Ces données sont ensuite corrigées pour les perturbations atmosphériques telles que le scattering des aérosols ou les fins nuages, permettant ainsi la comparaison entre plusieurs images et dates. Les corrections pour Landsat 5 et 7TM sont effectuées avec l'algorithme *Landsat Ecosystem Disturbance Adaptive Processing System (LEDAPS)* alors que pour Landsat 8 et 9 OLI c'est l'algorithme *Land Surface Reflectance Code (LaSRC)* qui est utilisé. Les deux modèles utilisent des données auxiliaires du climat de MODIS et un modèle de transfert radiatif pour évaluer les conditions atmosphériques sur une scène donnée (Chatenaux et al, 2021).

4 MÉTHODOLOGIE

Nous allons travailler avec l'interface Jupyterlab qui propose plusieurs bibliothèques et extensions utiles dans le traitement des données volumineux. L'interface de Jupyterlab utilise le langage de programmation Python et permet de se connecter au SDC pour acquérir des données, les traiter et les analyser en utilisant Python.

La première étape consiste à préparer nos données. Une fois les images téléchargées depuis le SDC pour notre région d'étude et aux dates sélectionnées, nous procéderons au nettoyage des images en enlevant les pixels couverts par des nuages. Pour ce faire, nous utiliserons la fonction "*clean*" disponible sur le SDC, laquelle utilise la bande QA_PIXEL. Cette dernière fournit des informations sur l'assurance de la qualité du pixel (*pixel_qa*), indiquant l'état d'un pixel (plein, vide, eau, ombre d'un nuage, neige, nuage (oui/non), confiance en nuage et confiance en nuage de cirrus (uniquement pour Landsat 8). Il faudra ensuite mettre à l'échelle avec la fonction "*scale*" pour restreindre les valeurs à une plage valide

Une fois que nos images sont prêtes, nous appliquerons l'algorithme du calcul du FC développé par Scarth à nos images Landsat. Le script utilisé (annexe I) est une adaptation de le script de Bruno

Chatenoux pour le calcul du FC à partir d'images Sentinel. Le produit qu'on obtient se compose de 3 couches:

- PV: la fraction de végétation verte
- NPV: la fraction de végétation non-verte
- BS: la fraction de sol nu

Il peut arriver que la somme de ces trois composantes ne soit pas égale à 100 %. Les différences sont généralement faibles et ne sont pas arrondies afin de préserver d'éventuels indicateurs saisonniers utiles. Cependant, les *outliers* ont été saturés. Les valeurs négatifs ont été considérés comme égaux à 0 et ceux dépassant le 100% comme égaux à 100%.

Les valeurs obtenues seront normalisées afin d'améliorer la qualité et la comparaison entre elles avant de calculer les moyennes par mois et par année. Ainsi, nous obtenons des images raster par mois et par année pour chacune de nos variables (PV, NPV et BS). Pour le traitement de ces raster nous utiliserons le logiciel QGIS.

Une sélection manuelle des images de Landsat 7 affectées par l'erreur de balayage n'ayant pas de couverture nuageuse a été effectuée. Ces images ont été ensuite traitées avec l'outil "*gap filling*" de QGIS pour interpoler des valeurs pour les pixels manquants. Il aurait été plus efficace de sélectionner les bandes sans nuages au moment du nettoyage des images dans le script, ce qui aurait réduit considérablement le temps de calcul du FC et les étapes ultérieures. Dans notre étude, nous avons décidé de conserver toutes les images afin de récupérer un maximum d'information, même si celle-ci n'est parfois que partielle. Il est important d'être conscient de ce choix, lors de l'analyse des résultats.

Les produits FC n'ont pas de masquage de l'eau, ce qui signifie que des valeurs erronées pour la végétation verte au-dessus de l'eau peuvent apparaître. Celles-ci doivent être ignorées et non prise en considération pour l'analyse. En utilisant l'outil *Extractor* de QGIS, nous avons calculé la moyenne et la déviation standard de chaque image en sélectionnant la région du Grand Genève sans le lac Léman ni les rivières qui s'y trouvent. L'outil *Extractor* produit un fichier csv avec la moyenne et la déviation standard pour chaque date. A partir de ce fichier, nous procéderons à l'analyse des séries temporelles du FC entre 1993 et 2023 dans l'interface de Jupyterlab (script dans l'annexe II).

En raison de la couverture nuageuses, il y a beaucoup des données manquants total ou partiellement, en particulier entre novembre et février. Seules les images avec un $FC = 100\% \pm 10\%$ ont été retenus pour l'analyse de la série temporelle. Une série temporelle⁵ est constitué de plusieurs composants, à savoir, la tendance, la saisonnalité et les résidus.

- La tendance indique la direction générale des données de la série temporelle sur une longue période, pouvant être croissante, décroissante ou horizontale.

5 <https://www.kaggle.com/code/prashant111/complete-guide-on-time-series-analysis-in-python#5.-Visualize-the-Time-Series->

- La composante saisonnière présente une tendance qui se répète en termes de moment, direction et ampleur.
- La variation irrégulière représente les fluctuations imprévisibles dans les données de la série temporelle une fois que les tendances et les variations cycliques sont éliminées.

Nous allons procéder à une décomposition ETS pour séparer les différentes composantes d'une série temporelle. Le terme ETS signifie erreur, tendance et saisonnalité. Pour ce faire, nous utiliserons *seasonal_decompose* de statsmodels , avec un modèle de série temporelle additive (script dans l'annexe III).

$$\text{Valeur} = \text{Niveau de base} + \text{Tendance} + \text{Saisonnalité} + \text{Erreur}$$

L'objectif de cette partie de l'étude étant d'identifier la saisonnalité du FC, nous avons limité la décomposition aux 10 dernières années (2013-2023), afin de simplifier la visualisation. Pour ces 10 années dû à la présences des données manquantes, spécialement pour les mois d'hiver, nous effectuerons une interpolation par le technique mathématique du spline pour estimer ces valeurs. Ce type d'interpolation estime les valeurs manquantes en ajustant une ligne flexible et incurvée à travers les points de données.

Afin de visualiser l'évolution que chacune de nos variables (PV, NPV, BS) a suivie pendant les 30 dernières années, la tendance depuis 1993 a été calculé (script dans l'annexe IV). Pour le calcul de la tendance nous n'avons pas fait recours à des techniques d'interpolation, préférant calculer les tendances avec les moyennes annuelles (donnée dans l'annexe V). La tendance a été calculé avec la fonction *polyfit* de la bibliothèque numpy qui permet d'obtenir les paramètres de modélisations polynomiales. Nous utiliserons un polynôme de degré 1 qui utilise l'équation:

$$y = m x + c \quad (\text{eq 10})$$

Où m est égal à la pente de notre ligne et c est le point d'intersection entre la ligne et l'axe y .

Comme résumé visuelle du travail réalisé, des cartes annuelles du FC ont été élaborées. Les raster correspondants à chacune de nos variables d'étude ont été combinés dans une composite RGB avec QGIS. Ces cartes permettent un comparaison rapide des changements majeurs inter-annuels, facilitant par exemple la détection de processus telles que l'urbanisation ou les changements dans la végétation liés à la météorologie. Avec l'aide du logiciel GIMP une animation gif a été créée à partir de ces cartes.

5 RÉSULTATS

5.1 Analyse visuelle

Dans un premier temps, les résultats obtenus ont été comparés avec les images de la région. Les estimations faites du FC dans le Grand Genève nous permettent de distinguer clairement les principales unités du paysage. Les zones construites, appartenant à la catégorie BS s'affichent en rouge sur la carte, tandis que les zones végétalisées varient entre le bleu et le vert en fonction du type de végétation. Il est possible de distinguer les zones de forêt dense et clairsemé, ainsi que les zones agricoles dans lesquelles prédomine la végétation non photosynthétique. Ceci peut s'expliquer par l'absence de strates supérieures de végétation qui puissent cacher la NPV existante.

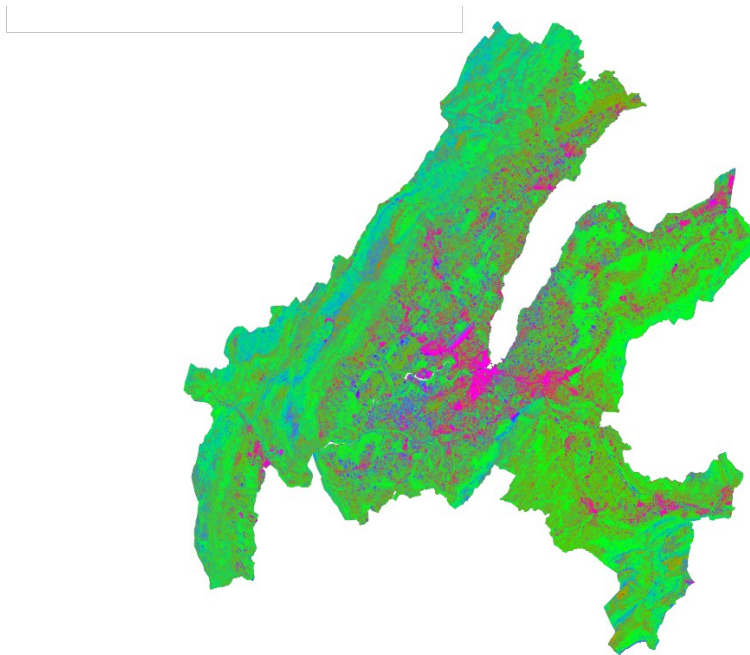


Figure 8: Fractional Cover dans le Grand Genève, moyenne 2023

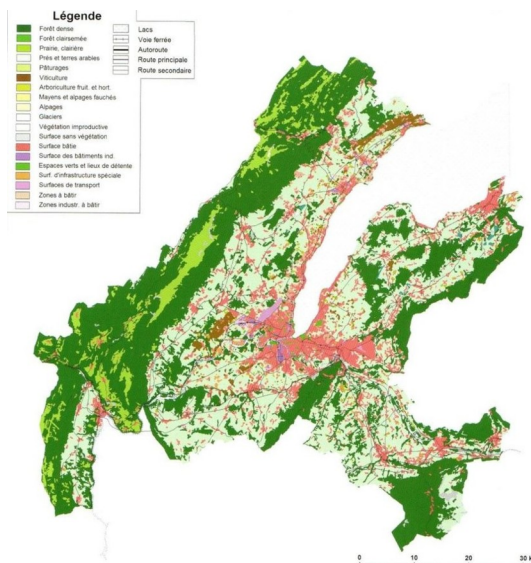


Figure 9: Carte d'occupation du sol du Grand Genève Source : Atlas du Grand Genève : état des lieux pour un progrès durable, Genève.

Les images du FC permettent également de détecter visuellement les changements dans le paysage, comme par exemple le processus d'urbanisation dans la région du Chablais au bord du Léman entre 1993 et 2023. On constate comment les parcelles agricoles en bleu en 1993 se réduisent et deviennent des zones construites en rouge en 2023.

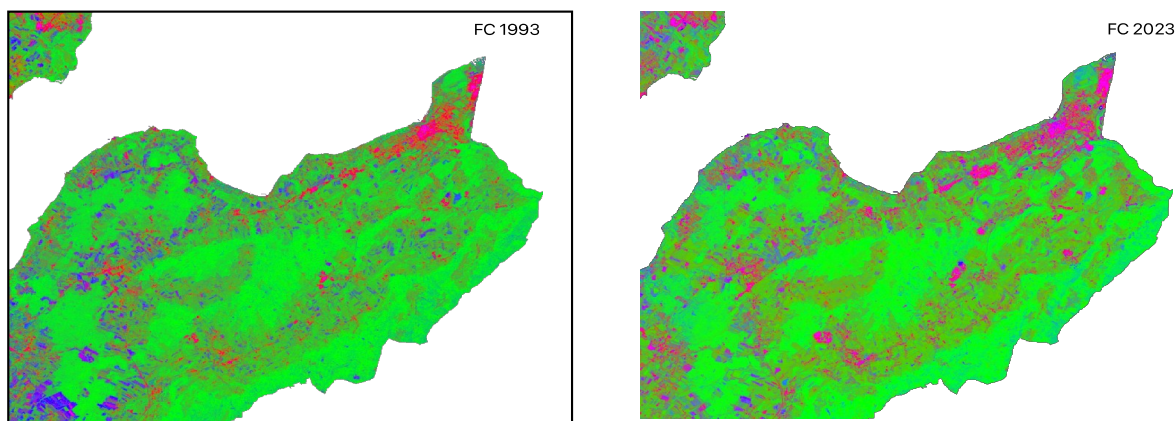


Figure 10: Fractional Cover région Chablais -1993 et 2023

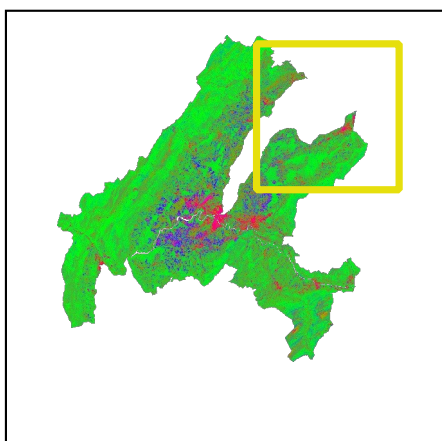


Figure 11: Emplacement de la zone d'exemple sur l'urbanisation

Les dynamiques des écosystèmes liées aux changements météorologiques sont également reconnaissables. Par exemple, les images de la figure 10 montrent les changements dans la région du Jura, avec une prédominance de la couleur verte (PV) en 2016, année caractérisée par des précipitations plus tôt élevées pendant toute la première moitié de la saison, et une forte augmentation de la couleur bleu (NPV) en 2020, année particulièrement sèche et chaude. Cependant, il faut rester attentif au fait que pour certaines années, dû à la couverture nuageuses, des données pour une saison entière peuvent manquer, ce qui pourrait biaiser les résultats. Dans ce travail notre objectif se limite à montrer le potentiel d'utilisation du FC pour la détection de changements et nous n'allons pas faire de distinction entre les moyennes annuelles. Dans un vraie étude des changements il serait nécessaire de s'assurer que la moyenne annuelle inclut des mesures de toutes les saisons. La moyenne saisonnière serait probablement plus adapté à la comparaison des changements liés à la météorologie.

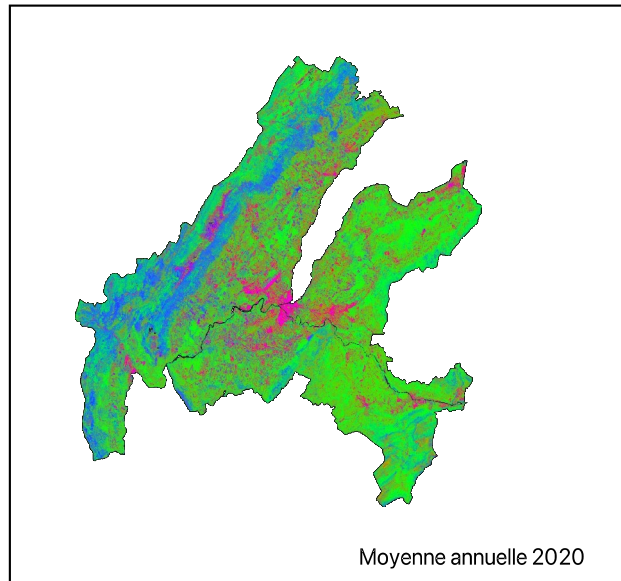
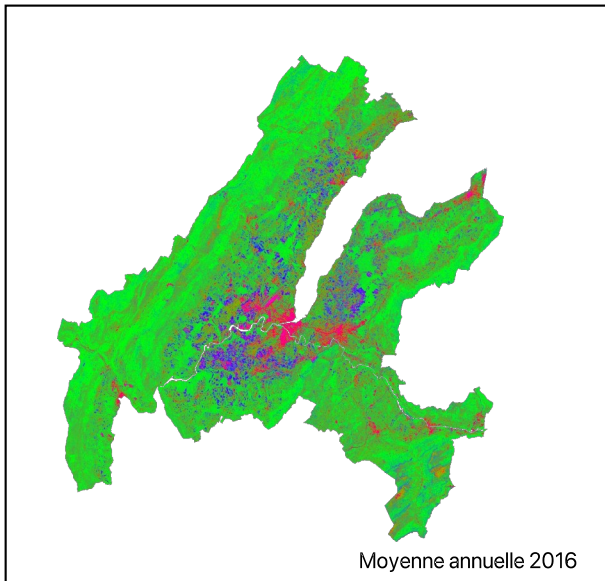


Figure 12: Fractional Cover Grand Genève moyenne, 2016 et 2020

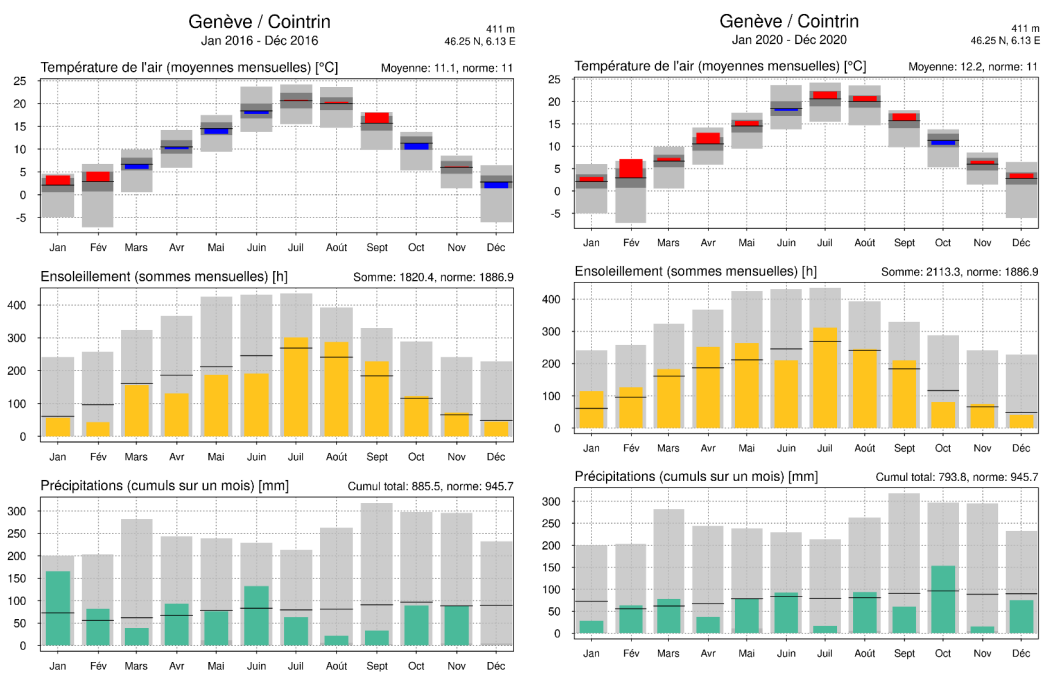


Figure 13: Evolution température, ensoleillement et précipitations à la station de Cointrin ,2016 et 2020.
Source: www.meteosuisse.admin.ch

5.2 Séries temporelles

Le calcul de moyennes zonales pour la région du Grand-Genève nous a permis d'obtenir de séries temporelles par indice pour la période 1993 et 2023. Les résultats dans la figure 14 montrent que le PV est le type de couverture prédominante dans la région. Nous constatons aussi que le BS est la catégorie qui expérimente le moins de fluctuations.

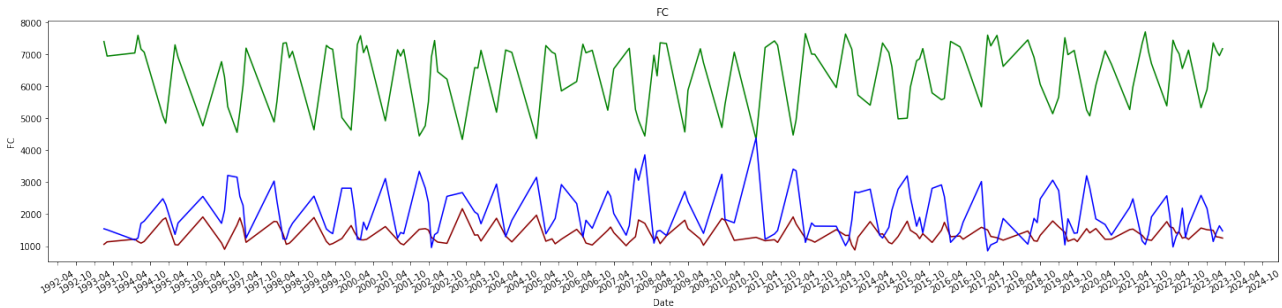


Figure 14: Evolution du FC entre 1993 et 2023

La déviation standard ou écart type a été calculée pour chaque image. Ces valeurs ont été intégrés dans les graphiques suivants (figures 13) pour visualiser la variabilité entre nos données. Les données de Landsat 7 sans nuages, après un traitement de gap filling avec le logiciel QGIS, apparaissent dans les graphiques mais ne seront pas prises en compte pour l'analyse de tendances, car les corrections effectués n'ont pas été validées.

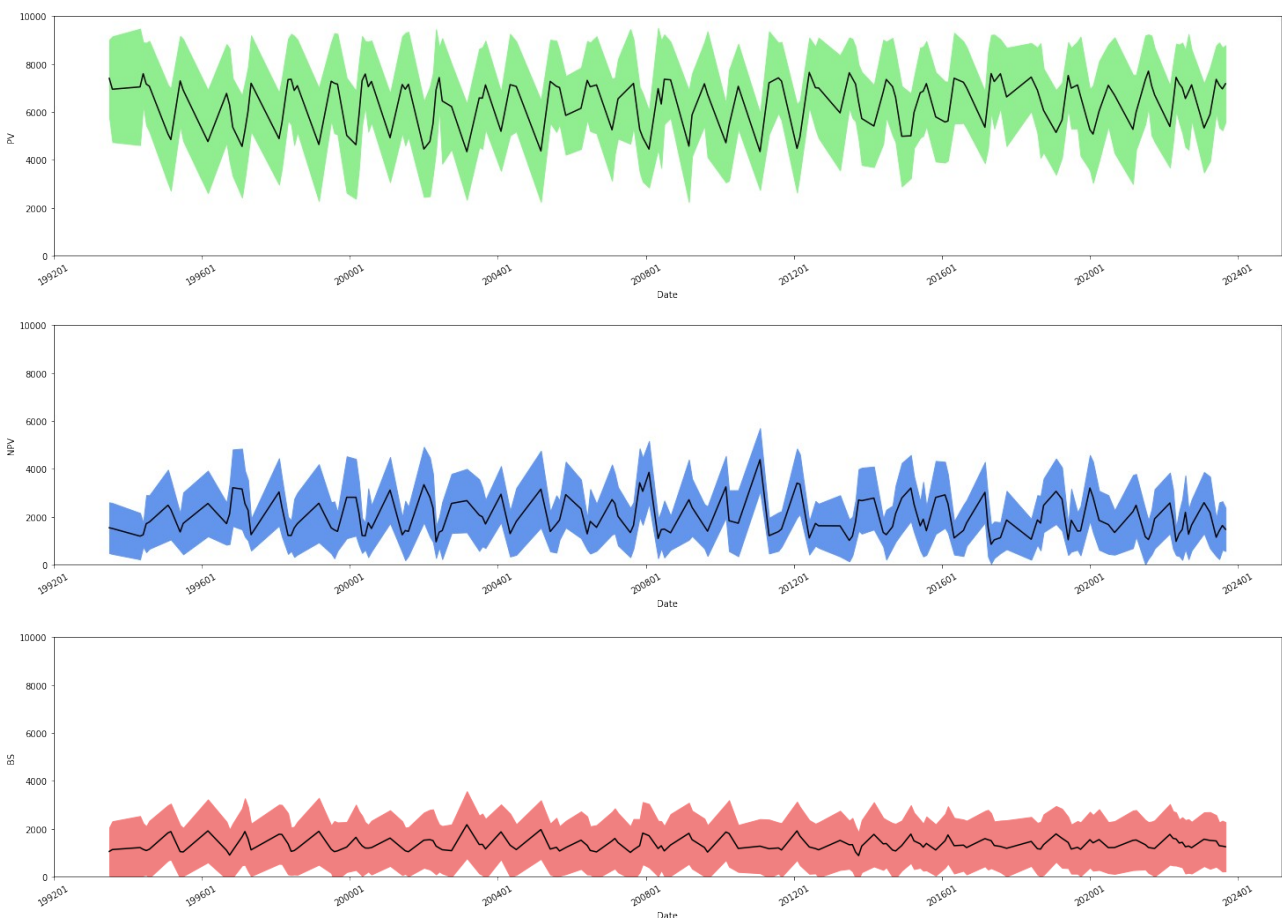


Figure 15: Evolution et déviation standard du PV, NPV et BS

Nous observons que la région du Grand Genève est dominée par une végétation photosynthétique, correspondant avec la réalité du paysage. Un tiers du territoire est composé d'espaces agricoles (champs, pâtures, vignes, etc.) et plus de 50% est couvert de forêts⁶. Les données montrent également l'existence de patrons saisonniers qui se répètent dans l'évolution du Fractional Cover.

5.3 Saisonnalité

La décomposition des données du FC dès 10 dernières années met en évidence le patrons saisonniers liés aux phases de croissance et dormance de la végétation.

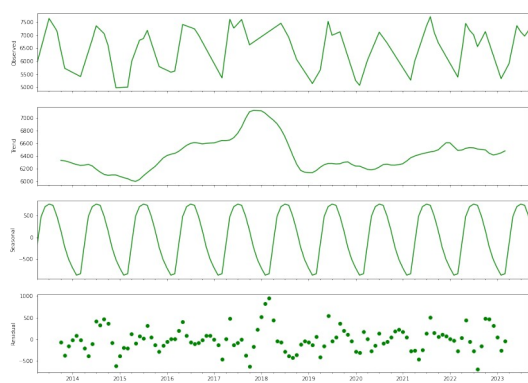


Figure 16: Décomposition PV - 2013-2023

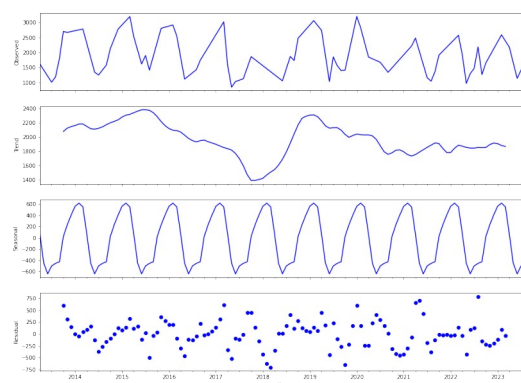


Figure 17: Décomposition NPV - 2013 - 2023

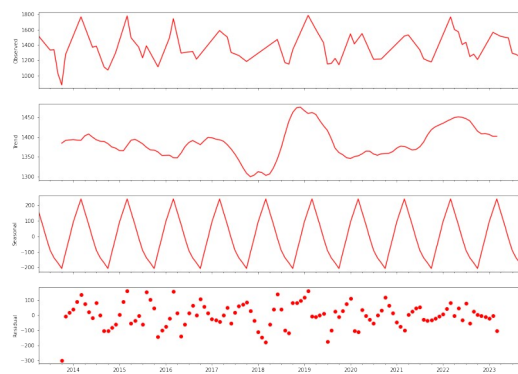


Figure 18: Décomposition BS - 2013 - 2023

Les estimations du FC coïncident avec la saisonnalité attendu dans nos régions. La fraction de PV diminue à la fin de l'été, atteignant son minimum en hiver avant de remonter au printemps et d'atteindre son maximum en 'été. La fraction de NPV suit une évolution pratiquement inverse à celle du PV, avec des valeurs maximales en automne-hiver et des valeurs minimales en printemps-été. La fraction du BS est celle qui fluctue le moins, suivant une dynamique saisonnière avec des valeurs maximales en hiver, quand le PV est à son minimum et les sols sont plus visibles depuis les satellites, et ses valeurs minimales en été quand le PV est à son maximum.

⁶ <https://www.grand-geneve.org/>

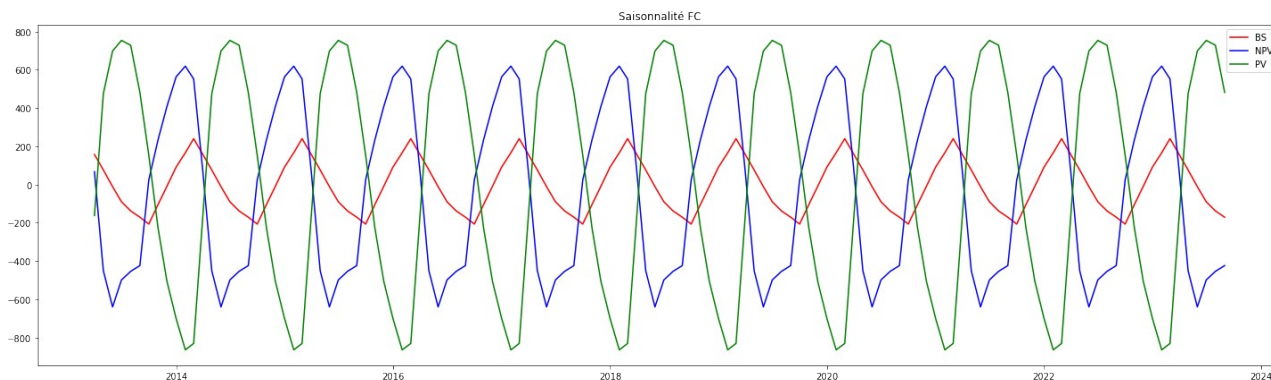


Figure 19: *Patrons saisonnières FC - 2013 - 2023*

5.4 Tendances

Concernant les tendances, même en éliminant la saisonnalité, il n'est pas évident de déceler une tendance claire. Il y a certainement d'autres cycles interannuelles qui devraient être pris en considération. De plus, beaucoup des données sont manquantes pour les mois d'hiver, surtout si nous enlevons les données Landsat 7 ayant l'erreur de balayage.

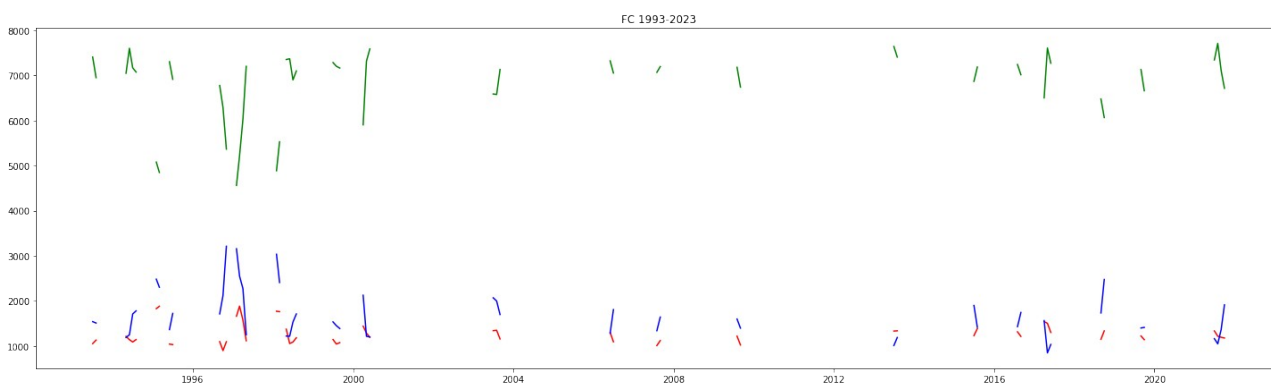


Figure 20: *Données Landsat 5 et 8/9 - 1993-2013*

Pour pallier ce manque des données à certains périodes, la tendance a été calculé avec les moyennes annuelles. Une autre possibilité qui pourrait être exploré serait d'obtenir la moyenne saisonnière chaque trois mois. Nous rappelons que pour certains années les données de plusieurs mois sont manquants dû à la couverture nuageuse, ce qui peut introduire des valeurs erronées dans l'analyse, une étude plus approfondie des tendances est conseillé.



Figure 21: Tendances du PV, NPV, BS - 1993 - 2023

Dans notre analyse du FC nous constatons :

- une légère diminution des surfaces de PV avec une pente de la tendance = -0.4636
- une diminution du NPV avec une pente = -8.7376
- une augmentation du BS avec une pente = 5.8488.

Les tendances des séries temporelles du FC sont cohérentes avec l'évolution que le territoire du Grand Genève connaît, marquée par une augmentation des zones bâties au détriment des zones agricoles. La région⁷ connaît une tendance à la baisse des surfaces agricoles utiles, en accord avec les données de l'OFS pour la Suisse et de l'Insee pour la partie française du Grand Genève.

- Genève a vu sa surface agricole se réduire de 30 ha de SAU/an entre 2000 et 2019. Le District de Nyon a connu une réduction de 4 ha de SAU/an pour la même période.
- Les surfaces agricoles utiles de la Haute-Savoie se sont réduites de 1219 ha de SAU/an.
- Dans l'Ain la réduction est de 709 ha de SAU/an pour l'Ain entre 2000 et 2010.
- Les données pour la France englobent l'ensemble du département et non pas seulement la partie appartenant au Grand Genève.

5.5 Animation 1993 -2023

Finalement des cartes avec les moyennes annuelles ont été élaborées afin de créer une animation gif de l'évolution du FC ces 30 dernières années. Les cartes utilisées pour cette animation sont présentées dans l'annexe VI.

⁷ [Agriculture - Grand Genève \(grand-geneve.org\)](http://grand-geneve.org)

6 CONCLUSION

Bénéficier des archives avec l'estimation du Fractional Cover (FC) pour la région du Grand Genève, voir idéalement l'ensemble de la Suisse, constituerait une valeur ajoutée significative aux données actuellement disponibles dans le domaine de la recherche scientifique. Les données du FC offrent la possibilité d'étudier de manière plus approfondie les dynamiques écologiques de la région en permettant un analyse rapide des fractions de PV, NPV et BS.

Actuellement il existe déjà des données du FC au niveau mondial estimés à partir des données MODIS . Bien que utiles pour l'étude des dynamiques à l'échelle des écorégions, la résolution spatiale de 250m de ces données, ne suffit pas pour des analyses plus fines à l'échelle locale, comme c'est le cas pour le Grand Genève.

Notre étude semble valider l'utilisation de l'algorithme du JRSRP dans la région du Grand Genève, avec des résultats cohérentes par rapport aux dynamiques saisonnières et à l'évolution du paysage au cours des 30 dernières années. Cependant, une validation avec des données de terrain demeure nécessaire pour conférer une valeur quantitative aux estimations du FC. Le travail de validation en cours dans le canton de Vaud pourrait bientôt valider l'utilisation de l'algorithme de la DEA pour le calcul du FC au niveau de la Suisse. L'optimisation du modèle avec des nouveaux *endmembers* basés sur des données suisses pourrait considérablement améliorer son efficacité, mais obtenir de telles données de terrain est une tâche laborieuse qui nécessite idéalement d'être effectué sur une période prolongée.

Il serait judicieux de reproduire le travail ici présenté avec les données Sentinel et de vérifier leur validité. L'algorithme que nous avons utilisé permet également d'estimer le FC à partir des données Sentinel, qui offrent une meilleure résolution spatiale (10 m) et temporelle, 6 jours au lieu de 16 pour Landsat. Ces avantages pourraient contribuer à réduire le nombre d'images fortement nuageuses et combler les lacunes d'informations à certains périodes de l'année. Une meilleure résolution spatial nous permettrait en plus de réaliser des études plus détaillé des changements dans la couverture de sol.

Cette étude représente une première étape prometteuse dans l'implémentation des cartes du FC pour le Grand Genève, ouvrant la voie à des investigations plus approfondies et à des applications variées dans le domaine de la gestion environnementale et de la planification du territoire.

ANNEXES

ANNEXE I - SCRIPT CALCUL FC

FC_chunks_landsat

January 21, 2024

1 Estimation du Fractional Cover dans le Grand Genève

1.1 Background

Pour calculer le FC nous avons adapté le script de Bruno Chatenoux (fc3_monthly_tif_maker.ipynb) pour appliquer le script développé par le Joint Remote Sensing Research Program aux images Landsat.

1.2 Description

Le FC permet de calculer les fractions de PV, NPV et BS à partir des images multispectrales. Dans ce script nous allons

1. Load des images depuis le SDC et définir les chunks
2. Nettoyage: on applique clean et scale
3. Calculer le FC
4. Sauvegarder les fichiers tif (moyenne mensuelle par bande: PV, NPV, BS)

1.2.1 Import dependencies

Commencer avec iPython magic commands.

```
[1]: # IMPORT DEPENDENCIES

# reload module before executing code
%load_ext autoreload
%autoreload 2

# define modules locations (you might have to adapt define_mod_locs.py)
%run sdc-notebooks/Tools/define_mod_locs.py
```

1.2.2 Load packages

Importer les Python packages nécessaires.

```
[2]: #standard python packages

import warnings
```



```

#monthly tif
import gc
import glob

import geopandas as gpd

#script chunk landsat
#from shapely.geometry import Polygon

from osgeo import gdal
from datetime import datetime
from dateutil import rrule

from odc.ui import with_ui_cbk

#from sdc_tools.sdc_plotting import rgb
#from sdc_tools.sdc_utilities import lsc2_loadcleanscale, write_geotiff_from_xr
#monthly tif sentinel
#from sdc_tools.sdc_utilities import create_scl_clean_mask,
↳write_geotiff_from_xr
from sdc_tools.sdc_utilities import lsc2_qa_clean, lsc2_ds_scale,
↳write_geotiff_from_xr

#import functions fc3
import os
import numpy as np
import xarray as xr

```

1.2.3 Connect to the datacube

Se connecter au datacube pour accéder aux données du SDC.

```

[3]: # connect to DC
import datacube
dc = datacube.Datacube()

```

```

[4]: products = dc.list_products()
products

```

```

[4]:
name \
name
arealstatistik          arealstatistik
astgtmv003              astgtmv003
corinelc_europe        corinelc_europe
corinelc_switzerland   corinelc_switzerland
GE1                    GE1

```

landsat_etm_c2_l2	landsat_etm_c2_l2
landsat_ot_c2_l2	landsat_ot_c2_l2
landsat_tm_c2_l2	landsat_tm_c2_l2
RapidMapping	RapidMapping
s2l2	s2l2
swissalti3d	swissalti3d
swissimagedop10	swissimagedop10

description \

name	
arealstatistik	Arealstatistik products from Federal Statistic...
astgtmv003	Global Elevation Data from TERRA satellite's A...
corinelc_europe	CORINE Land Cover Europe v2020_20u1 products f...
corinelc_switzerland	CORINE Land Cover Switzerland 1990 products fo...
GE1	GeoEye 1 imagery, provided by Swisstopo and co...
landsat_etm_c2_l2	USGS Landsat 7 Collection 2 Level-2 Surface Re...
landsat_ot_c2_l2	USGS Landsat 8-9 Collection 2 Level-2 Surface ...
landsat_tm_c2_l2	USGS Landsat 4-5 Collection 2 Level-2 Surface ...
RapidMapping	Special flight imagery of 13 August 2022, prov...
s2l2	Sentinel-2a and Sentinel-2b imagery, processed...
swissalti3d	swissALTI3D from swisstopo.
swissimagedop10	SWISSIMAGE 10 provided by SWISSTOPO

license default_crs \

name		
arealstatistik	None	EPSG:2056
astgtmv003	None	EPSG:4326
corinelc_europe	None	EPSG:3035
corinelc_switzerland	None	EPSG:2056
GE1	None	None
landsat_etm_c2_l2	None	None
landsat_ot_c2_l2	None	None
landsat_tm_c2_l2	None	None
RapidMapping	None	None
s2l2	None	None
swissalti3d	None	EPSG:2056
swissimagedop10	None	None

default_resolution

name	
arealstatistik	(-100, 100)
astgtmv003	(-0.00027777777777778, 0.00027777777777778)
corinelc_europe	(-100, 100)
corinelc_switzerland	(-250, 250)
GE1	None
landsat_etm_c2_l2	None
landsat_ot_c2_l2	None

landsat_tm_c2_l2	None
RapidMapping	None
s2l2	None
swissalti3d	(-2, 2)
swissimagedop10	None

1.3 Calcul

1.3.1 Import des fonctions et données

```
[5]: from fractionalcover3 import unmix_fractional_cover
      from fractionalcover3 import data
```

1.3.2 Définition de la fonction fc3

Stack pour aligner bands sur une seule dimension. Garder coordonnées et attributs. Définir les bandes qui seront créés. Enlever la bande bleu. ### Définition des autres fonctions Saturation de valeurs ≥ 1 ou ≤ 0

```
[6]: def fc3(ds_t):
      bands = ['blue', 'green', 'red', 'nir', 'swir_1', 'swir_2']
      fractions = unmix_fractional_cover(np.stack([ds_t[band].values for band in
      ↪bands])),
                                          fc_model=data.get_model())

      fc3_ds_t = xr.Dataset(coords = {'x': ds_clean.x.values, 'y': ds_clean.y.
      ↪values},
                            attrs=ds_clean.attrs)
      fc3_ds_t["bs"] = xr.DataArray(fractions[0], dims=["y", "x"])
      fc3_ds_t["pv"] = xr.DataArray(fractions[1], dims=["y", "x"])
      fc3_ds_t["npv"] = xr.DataArray(fractions[2], dims=["y", "x"])
      return fc3_ds_t.where(~ds_t.blue.isnull())

def saturate_var_max(ds):
    vars = ds.data_vars
    for var in vars:
        mask = fc3_ls_ds[var] >= 1
        fc3_ls_ds[var] = xr.where(mask, 1, fc3_ls_ds[var])
    for v in vars:
        if v != var:
            fc3_ls_ds[v] = xr.where(mask, 0, fc3_ls_ds[v])

def saturate_var_min(ds):
    vars = ds.data_vars
    for var in vars:
        mask = fc3_ls_ds[var] <= 0
        fc3_ls_ds[var] = xr.where(mask, 0, fc3_ls_ds[var])
```

```
def delete_file(fp):
    if os.path.isfile(fp):
        os.remove(fp)
```

1.4 Load data

Définir période et AOI (Grand Genève)

```
[71]: # %load ../utils/config_cell.txt
# Configuration

product = 'landsat_etm_c2_l2'
measurements = ['blue', 'green', 'red', 'nir', 'swir_1', 'swir_2', 'QA_PIXEL']

#CH1903+
longitude = (2462923.93, 2539056.35)
latitude = (1085136.47, 1158288.61 )
crs = "epsg:2056"

time = ('2011-01', '2014-12')

output_crs = 'epsg:32632'
#resolution = -30.0, 30.0
res_ls = 30
#resolution = (-res_ls, res_ls) script fc3_TS_processing
```

```
[8]: #aoi_gpkg = 'swissBOUNDARIES3D_1_4_TLM_KANTONSGBIET_Vaud.gpkg'
aoi_shp = 'AGGLO_PERIMETRE_AVEC_LAC.shp'
```

```
[9]: # Load gpkg, get bounding box and crs epsg code
# and define AoI by extending bounding box by 1 resolution

gdf = gpd.read_file(aoi_shp)
extent = gdf.total_bounds # Returns [minx, miny, maxx, maxy]
epsg_code = gdf.crs.to_epsg()

#longitude = (extent[0] - res_s2, extent[2] + res_s2)
#latitude = (extent[1] - res_s2, extent[3] + res_s2)

longitude = (extent[0] - res_ls, extent[2] + res_ls)
latitude = (extent[1] - res_ls, extent[3] + res_ls)
```

1.4.1 Définition des chunks

Afin de ne pas saturer la mémoire nous allons travailler par chunks

```
[15]: # chunk

import itertools

cols, rows = 3, 4

x_chunk_size = (longitude[1] - longitude[0] + res_ls) / cols
y_chunk_size = (latitude[1] - latitude[0] + res_ls) / rows

x_ranges = [((longitude[0] - res_ls / 2) + x_chunk_size * chunk_number,
             (longitude[0] - res_ls / 2) + x_chunk_size * (chunk_number + 1))
            for chunk_number in range(cols)]
y_ranges = [((latitude[0] - res_ls / 2) + y_chunk_size * chunk_number,
             (latitude[0] - res_ls / 2) + y_chunk_size * (chunk_number + 1))
            for chunk_number in range(rows)]

chunks = [{ 'longitude': pair[0], 'latitude': pair[1]} for pair in
          ↪list(itertools.product(x_ranges, y_ranges))]
```

1.4.2 Calcul

Par chunks, nous allons: Nettoyer et mettre à l'échelle les données, calculer le FC, saturer les outliers, normaliser, calculer la moyenne mensuelle, normaliser et sauvegarder les données obtenues par mois et variable (PV, NPV, BS).

```
[72]: # MAIN PROCESS

delete_file("fc3_monthly.log")

#####
#corriger format dates

for dt in rrule.rrule(rrule.MONTHLY, dtstart=datetime.strptime(time[0],
↪'%Y-%m'), until=datetime.strptime(time[1], '%Y-%m')):
    # log processing
    msg = f"{datetime.now().strftime('%Y-%m-%d %H:%M')} | processing {dt.
↪strftime('%Y-%m')}}"
    print(msg)
    with open("fc3_monthly.log", "a") as f:
        print(msg, file=f)

# process by chunks
for i in range(len(chunks)):
    # load
    with warnings.catch_warnings():
        warnings.filterwarnings("ignore", category=Warning)
        ds = dc.load(longitude = chunks[i]['longitude'],
```

```

        latitude = chunks[i]['latitude'],
        crs = f"epsg:{epsg_code}",
        product = product,
        time = (dt.strftime('%Y-%m'),dt.strftime('%Y-%m')),
        measurements = measurements,
        output_crs = f"epsg:{epsg_code}",
        resolution = (-res_ls, res_ls))

if len(ds) == 0:
    continue
# mask
clean_mask = lsc2_qa_clean(ds.QA_PIXEL)
ds_clean = ds.drop_vars('QA_PIXEL').where(clean_mask)
del ds
gc.collect()

# scale
ds_clean = lsc2_ds_scale(ds_clean, product)

# compute fc3 for each time
fc3_ls_ds = ds_clean.groupby('time').apply(fc3)

del ds_clean
gc.collect()

# saturate outliers
saturate_var_max(fc3_ls_ds)
saturate_var_min(fc3_ls_ds)

# normalize scenes
total = fc3_ls_ds["pv"] + fc3_ls_ds["npv"] + fc3_ls_ds["bs"]
fc3_ls_ds["pv"] /= total
fc3_ls_ds["npv"] /= total
fc3_ls_ds["bs"] /= total

# summarize per month
fc3_ls_ds = fc3_ls_ds.resample(time="1M").median()

# normalize monthly summary
total = fc3_ls_ds["pv"] + fc3_ls_ds["npv"] + fc3_ls_ds["bs"]
fc3_ls_ds["pv"] /= total
fc3_ls_ds["npv"] /= total
fc3_ls_ds["bs"] /= total

# save chunk
for v in fc3_ls_ds.data_vars:
    write_geotiff_from_xr(tif_path = f"tmp_{v}_{i}.tif",

```

```

dataset = (fc3_ls_ds[v].isel(time = 0) *
↳10000) \
                                                .astype('int16') \
                                                .rename({'y':
↳'latitude', 'x':'longitude'}) \
                                                .to_dataset(name = v),
crs = f"epsg:{epsg_code}",
compr = 'DEFLATE',
no_data = -9999)

del fc3_ls_ds
gc.collect()

# merge chunks
for v in ['pv', 'npv', 'bs']:
    v_files = glob.glob(f"tmp_{v}*.tif")
    gdal.BuildVRT("merged.vrt", v_files)
    gdal.Translate(f"fc3_{v}_m{dt.strftime('%Y%m')}.tif",
                  "merged.vrt", creationOptions=["COMPRESS=Deflate"])
    os.remove("merged.vrt")
for vf in v_files:
    os.remove(vf)

```

2024-01-16 22:40 | processing 2011-01

```

/opt/development/conda-envs/jovyan/odc-dev/lib/python3.9/site-
packages/xarray/core/duck_array_ops.py:171: RuntimeWarning: invalid value
encountered in cast
    return data.astype(dtype, **kwargs)
/opt/development/conda-envs/jovyan/odc-dev/lib/python3.9/site-
packages/xarray/core/duck_array_ops.py:171: RuntimeWarning: invalid value
encountered in cast
    return data.astype(dtype, **kwargs)
/opt/development/conda-envs/jovyan/odc-dev/lib/python3.9/site-
packages/xarray/core/duck_array_ops.py:171: RuntimeWarning: invalid value
encountered in cast
    return data.astype(dtype, **kwargs)
/opt/development/conda-envs/jovyan/odc-dev/lib/python3.9/site-
packages/xarray/core/duck_array_ops.py:171: RuntimeWarning: invalid value
encountered in cast
    return data.astype(dtype, **kwargs)
/opt/development/conda-envs/jovyan/odc-dev/lib/python3.9/site-
packages/xarray/core/duck_array_ops.py:171: RuntimeWarning: invalid value
encountered in cast
    return data.astype(dtype, **kwargs)

```

```
/opt/development/conda-envs/jovyan/odc-dev/lib/python3.9/site-
packages/xarray/core/duck_array_ops.py:171: RuntimeWarning: invalid value
encountered in cast
    return data.astype(dtype, **kwargs)
/opt/development/conda-envs/jovyan/odc-dev/lib/python3.9/site-
packages/xarray/core/duck_array_ops.py:171: RuntimeWarning: invalid value
encountered in cast
    return data.astype(dtype, **kwargs)
/opt/development/conda-envs/jovyan/odc-dev/lib/python3.9/site-
packages/xarray/core/duck_array_ops.py:171: RuntimeWarning: invalid value
encountered in cast
    return data.astype(dtype, **kwargs)
/opt/development/conda-envs/jovyan/odc-dev/lib/python3.9/site-
packages/xarray/core/duck_array_ops.py:171: RuntimeWarning: invalid value
encountered in cast
    return data.astype(dtype, **kwargs)
/opt/development/conda-envs/jovyan/odc-dev/lib/python3.9/site-
packages/xarray/core/duck_array_ops.py:171: RuntimeWarning: invalid value
encountered in cast
    return data.astype(dtype, **kwargs)
/opt/development/conda-envs/jovyan/odc-dev/lib/python3.9/site-
packages/xarray/core/duck_array_ops.py:171: RuntimeWarning: invalid value
encountered in cast
    return data.astype(dtype, **kwargs)
```

1.5 Information additional

Ce notebook est inspiré du template pour notebooks de la DEA.

ANNEXE II - SCRIPT SERIE TEMPORELLE

time-series_date-std_ALL

January 21, 2024

Time series - PV, NPV, BS - mean and standard deviation

```
[5]: # IMPORT DEPENDENCIES

# reload module before executing code
%load_ext autoreload
%autoreload 2

# define modules locations (you might have to adapt define_mod_locs.py)
%run sdc-notebooks/Tools/define_mod_locs.py
```

The autoreload extension is already loaded. To reload it, use:

```
%reload_ext autoreload
```

```
[6]: import warnings

import xarray as xr
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from matplotlib.dates import DateFormatter

import csv

from dateutil.parser import parse
```

```
[7]: %matplotlib inline
```

```
[98]: file = pd.read_csv("FC_93-23_ALL_nan.csv")
```

```
[99]: file.head()
```

```
[99]:
```

	time	fid	bs_mean	bs_std	npv_mean	npv_std	\
0	199307.0	1.0	1054.775291	1001.426236	1541.767800	1059.451323	
1	199308.0	1.0	1132.368496	1181.174668	1510.916887	1062.569578	
2	199405.0	1.0	1215.246040	1315.950739	1191.283207	968.164109	

```

3 199406.0 1.0 1141.751071 1090.906754 1250.556405 497.291966
4 199407.0 1.0 1091.561499 1016.294777 1712.214020 1190.962899

```

```

      pv_mean      pv_std      TL FC
0 7401.772724 1610.654016 9998.315814
1 6946.447855 2206.153834 9589.733238
2 7044.125526 2427.042397 9450.654773
3 7596.966003 1304.057357 9989.273479
4 7162.915931 1722.495738 9966.691451

```

```
[100]: file.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 366 entries, 0 to 365
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   time        154 non-null    float64
1   fid         149 non-null    float64
2   bs_mean     154 non-null    float64
3   bs_std      154 non-null    float64
4   npv_mean    154 non-null    float64
5   npv_std     154 non-null    float64
6   pv_mean     154 non-null    float64
7   pv_std      154 non-null    float64
8   TL FC      154 non-null    float64
dtypes: float64(9)
memory usage: 25.9 KB

```

```
[101]: # convert column to datetime pandas
file['time'] = pd.to_datetime(file['time'], format="%Y%m")
```

```
[102]: file.head()
```

```

[102]:      time  fid      bs_mean      bs_std      npv_mean      npv_std  \
0 1993-07-01  1.0 1054.775291 1001.426236 1541.767800 1059.451323
1 1993-08-01  1.0 1132.368496 1181.174668 1510.916887 1062.569578
2 1994-05-01  1.0 1215.246040 1315.950739 1191.283207  968.164109
3 1994-06-01  1.0 1141.751071 1090.906754 1250.556405  497.291966
4 1994-07-01  1.0 1091.561499 1016.294777 1712.214020 1190.962899

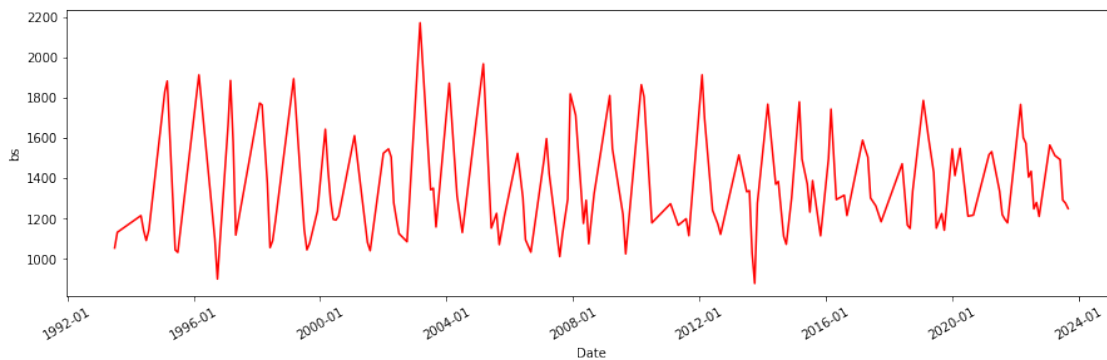
      pv_mean      pv_std      TL FC
0 7401.772724 1610.654016 9998.315814
1 6946.447855 2206.153834 9589.733238
2 7044.125526 2427.042397 9450.654773
3 7596.966003 1304.057357 9989.273479
4 7162.915931 1722.495738 9966.691451

```

```
[103]: file.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 366 entries, 0 to 365  
Data columns (total 9 columns):  
#   Column      Non-Null Count  Dtype  
---  ---  
0   time        154 non-null   datetime64[ns]  
1   fid         149 non-null   float64  
2   bs_mean     154 non-null   float64  
3   bs_std     154 non-null   float64  
4   npv_mean   154 non-null   float64  
5   npv_std    154 non-null   float64  
6   pv_mean    154 non-null   float64  
7   pv_std     154 non-null   float64  
8   TL FC      154 non-null   float64  
dtypes: datetime64[ns](1), float64(8)  
memory usage: 25.9 KB
```

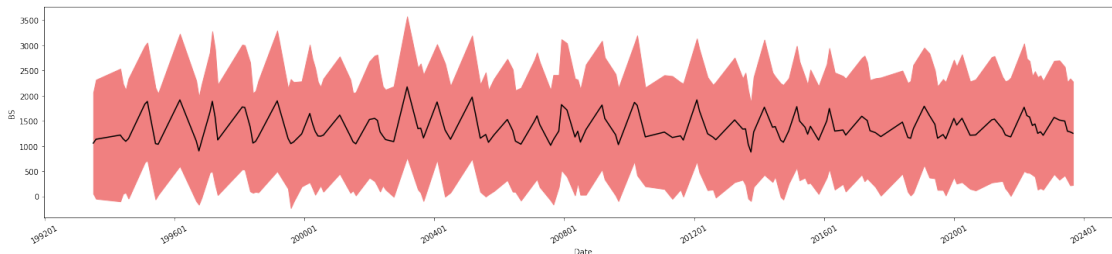
```
[111]: # Plot  
plt.figure(figsize=(15, 4.2))  
#x = range(len(file['time']))  
plt.plot(file['time'], file['bs_mean'], color = "red")  
#plt.xticks(x, file['time'])  
plt.xticks(rotation="30")  
  
#plt.gca().xaxis_date()  
# tick every 3 months  
#plt.gca().xaxis.set_major_locator(mdates.MonthLocator(bymonth=(1,7)))  
#plt.gca().xaxis.set_minor_locator(mdates.MonthLocator())  
# format as "%Y%m%d"  
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter("%Y-%m"))  
  
plt.xlabel('Date')  
plt.ylabel('bs')  
plt.show()
```



```
[146]: # Plot
plt.figure(figsize=(25, 5))
#x = range(len(File['time']))
plt.plot(file['time'], file['bs_mean'], "k-")
plt.fill_between(file['time'], file['bs_mean']-file['bs_std'],
    ↪file['bs_mean']+file['bs_std'], color = "lightcoral")
#plt.xticks(x, file['time'])
plt.xticks(rotation="30")

#plt.gca().xaxis_date()
# tick every 3 months
#plt.gca().xaxis.set_major_locator(mdates.MonthLocator(bymonth=(1,7)))
#plt.gca().xaxis.set_minor_locator(mdates.MonthLocator())
# format as "%Y%m%d"
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter("%Y%m"))
#plt.ylim((0,10000))

plt.xlabel('Date')
plt.ylabel('BS')
plt.savefig('bs_st_93-23.png', bbox_inches='tight')
plt.show()
```

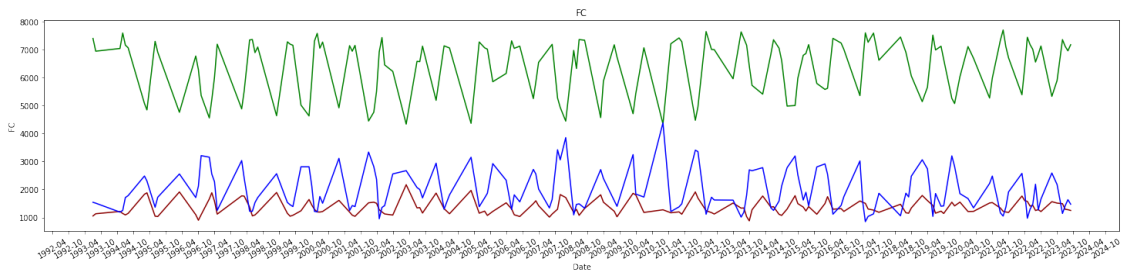


```
[127]: fig, ax = plt.subplots(figsize=(25, 5))
ax.plot(file['time'], file['bs_mean'], color='darkred')
#plt.fill_between(file['time'], file['bs_mean']-file['bs_std'],
    ↪file['bs_mean']+file['bs_std'], color = "lightcoral")
ax.plot(file['time'], file['pv_mean'], color='green')
#plt.fill_between(file['time'], file['pv_mean']-file['pv_std'],
    ↪file['pv_mean']+file['pv_std'], color = "lightgreen", alpha=0.8)
ax.plot(file['time'], file['npv_mean'], color='blue')
#plt.fill_between(file['time'], file['npv_mean']-file['npv_std'],
    ↪file['npv_mean']+file['npv_std'], color = "cornflowerblue", alpha=0.7)
plt.xticks(rotation="30")
ax.set(xlabel="Date", ylabel="FC",
```

```
title="FC")

# Format the x axis
ax.xaxis.set_major_locator(mdates.MonthLocator(interval=6))
ax.xaxis.set_major_formatter(DateFormatter("%Y-%m"))

plt.savefig('3bands_93-23.png', bbox_inches='tight')
plt.show()
```



[]:

ANNEXE IV - SCRIPT SAISONNALITE

gapfilling_season_3FC

January 23, 2024

Estimation de la saisonnalité

```
[1]: # IMPORT DEPENDENCIES

# reload module before executing code
%load_ext autoreload
%autoreload 2

# define modules locations (you might have to adapt define_mod_locs.py)
%run sdc-notebooks/Tools/define_mod_locs.py
```

```
[37]: import pandas as pd
from pandas import read_csv
from matplotlib import pyplot as plt
from numpy import polyfit
df = read_csv('FC_13-23_ALL.csv')
```

```
[38]: # convert column to datetime pandas
df['time'] = pd.to_datetime(df['time'], format="%Y%m")
```

```
[39]: # set the Date column be the index of our dataset
df = df.set_index('time')
```

```
[40]: df.shape
```

```
[40]: (126, 8)
```

```
[41]: nul_data = pd.isnull(df['bs_mean'])

# print only the data, Customers = NaN
df[nul_data]
```

```
[41]:
```

	fid	bs_mean	bs_std	npv_mean	npv_std	pv_mean	pv_std	\
time								
2013-05-01	1	NaN	NaN	NaN	NaN	NaN	NaN	
2013-06-01	1	NaN	NaN	NaN	NaN	NaN	NaN	
2013-12-01	1	NaN	NaN	NaN	NaN	NaN	NaN	
2014-01-01	1	NaN	NaN	NaN	NaN	NaN	NaN	

2014-02-01	1	NaN	NaN	NaN	NaN	NaN	NaN
...
2022-11-01	1	NaN	NaN	NaN	NaN	NaN	NaN
2022-12-01	1	NaN	NaN	NaN	NaN	NaN	NaN
2023-01-01	1	NaN	NaN	NaN	NaN	NaN	NaN
2023-03-01	1	NaN	NaN	NaN	NaN	NaN	NaN
2023-05-01	1	NaN	NaN	NaN	NaN	NaN	NaN

TL

```
time
2013-05-01 #VALEUR !
2013-06-01 #VALEUR !
2013-12-01 #VALEUR !
2014-01-01 #VALEUR !
2014-02-01 #VALEUR !
...
2022-11-01 #VALEUR !
2022-12-01 #VALEUR !
2023-01-01 #VALEUR !
2023-03-01 #VALEUR !
2023-05-01 #VALEUR !
```

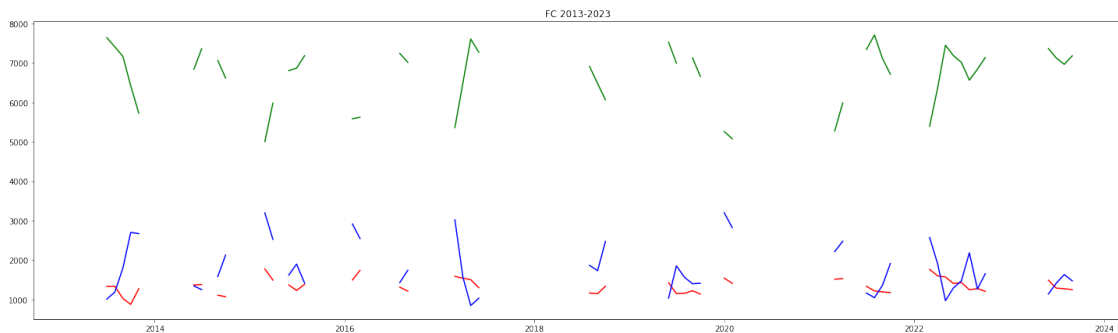
[61 rows x 8 columns]

```
[42]: plt.rcParams['figure.figsize']=(25,7)

# plots our series
plt.plot(df['bs_mean'], color='red')
plt.plot(df['npv_mean'], color='blue')
plt.plot(df['pv_mean'], color='green')

plt.title('FC 2013-2023')

#plt.savefig('FC_13-23.png', bbox_inches='tight')
plt.show()
```

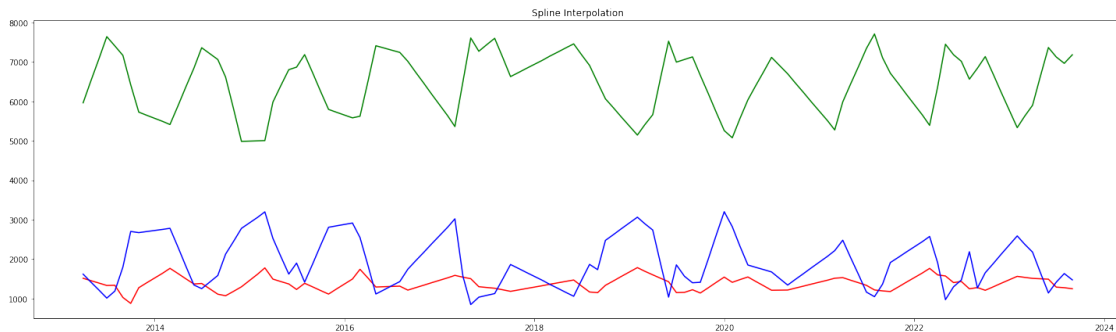


```
[43]: plt.rcParams['figure.figsize']=(25,7)

# on our data, impute the missing values using the interpolation techniques and
↳ specifically, the linear method
df['bs_mean_Spline']= df['bs_mean'].interpolate(option='spline')
df['npv_mean_Spline']= df['npv_mean'].interpolate(option='spline')
df['pv_mean_Spline']= df['pv_mean'].interpolate(option='spline')

# plot the complete dataset
plt.plot(df['bs_mean_Spline'], color='red')
plt.plot(df['npv_mean_Spline'], color='blue')
plt.plot(df['pv_mean_Spline'], color='green')

plt.title('Spline Interpolation')
#plt.savefig('Interpolation_13-23', bbox_inches='tight')
plt.show()
```



```
[44]: df['bs_mean_Spline']
```

```
[44]: time
2013-04-01    1515.745240
2013-05-01    1454.904694
2013-06-01    1394.064148
2013-07-01    1333.223602
2013-08-01    1338.574778
...
2023-05-01    1502.781316
2023-06-01    1492.556792
2023-07-01    1292.069996
2023-08-01    1277.260125
2023-09-01    1250.295091
Name: bs_mean_Spline, Length: 126, dtype: float64
```



```
[45]: nul_data = pd.isnull(df['bs_mean_Spline'])
```

```
#print only the data, Customers = NaN  
df[nul_data]
```

```
[45]: Empty DataFrame
```

```
Columns: [fid, bs_mean, bs_std, npv_mean, npv_std, pv_mean, pv_std, TL,  
bs_mean_Spline, npv_mean_Spline, pv_mean_Spline]  
Index: []
```

```
[46]: from statsmodels.tsa.seasonal import seasonal_decompose  
from dateutil.parser import parse
```

```
# Additive Decomposition
```

```
additive_decomposition_bs = seasonal_decompose(df['bs_mean_Spline'],  
↳model='additive', period=12)
```

```
additive_decomposition_npv = seasonal_decompose(df['npv_mean_Spline'],  
↳model='additive', period=12)
```

```
additive_decomposition_pv = seasonal_decompose(df['pv_mean_Spline'],  
↳model='additive', period=12)
```

```
# Plot
```

```
plt.rcParams.update({'figure.figsize': (16,12)})
```

```
#additive_decomposition_bs.plot()
```

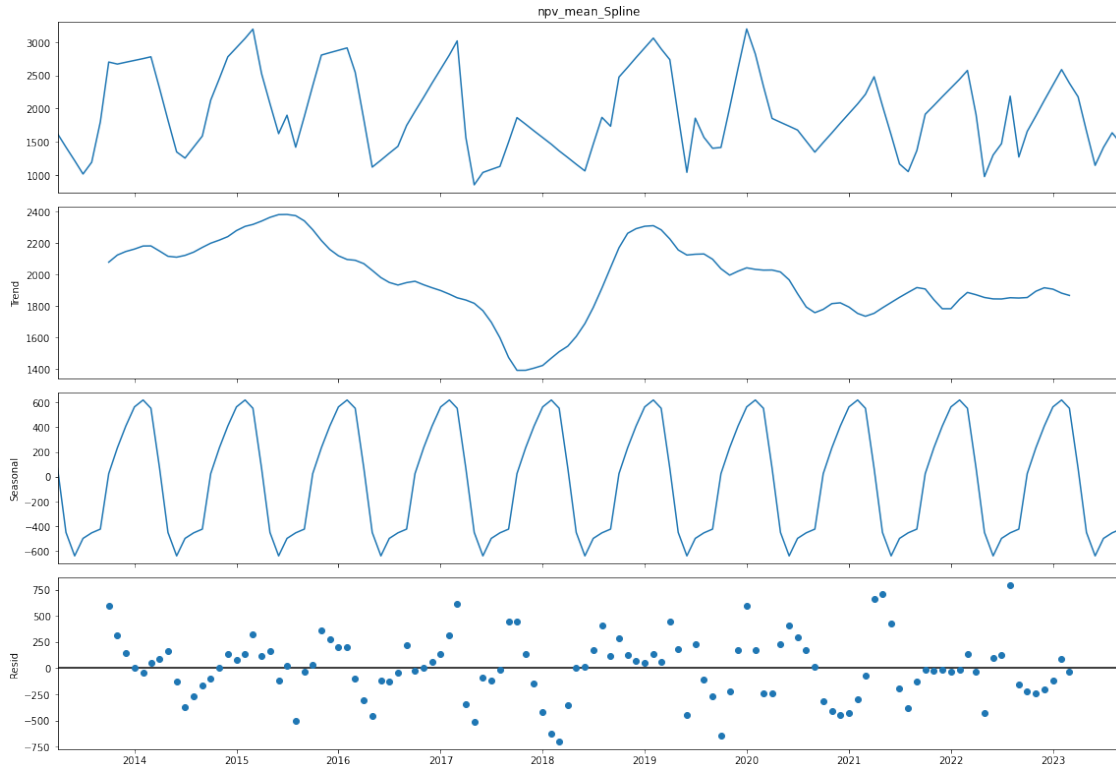
```
additive_decomposition_npv.plot()
```

```
#additive_decomposition_pv.plot()
```

```
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
```

```
#plt.savefig('decomposition_bs_13-23.png', bbox_inches='tight')
```

```
plt.show()
```

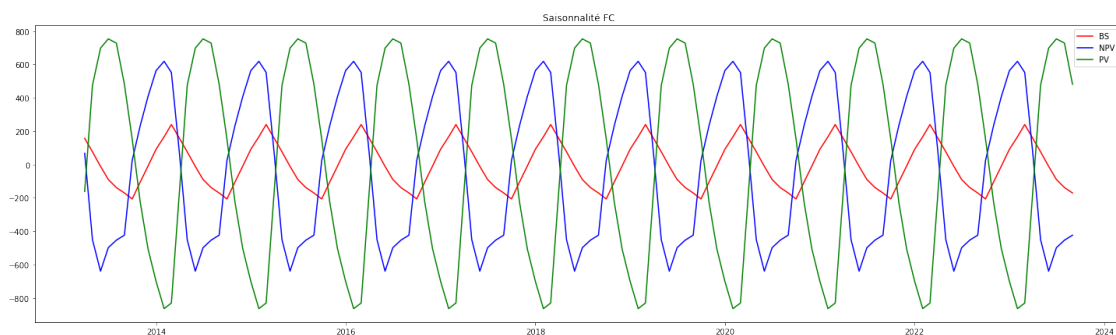


```
[47]: plt.rcParams['figure.figsize']=(25,7)
```

```
# plot the seasonality
plt.plot(additive_decomposition_bs.seasonal, color='red', label="BS")
plt.plot(additive_decomposition_npv.seasonal, color='blue', label="NPV")
plt.plot(additive_decomposition_pv.seasonal, color='green', label="PV")

plt.title('Saisonnalité FC')
plt.legend()

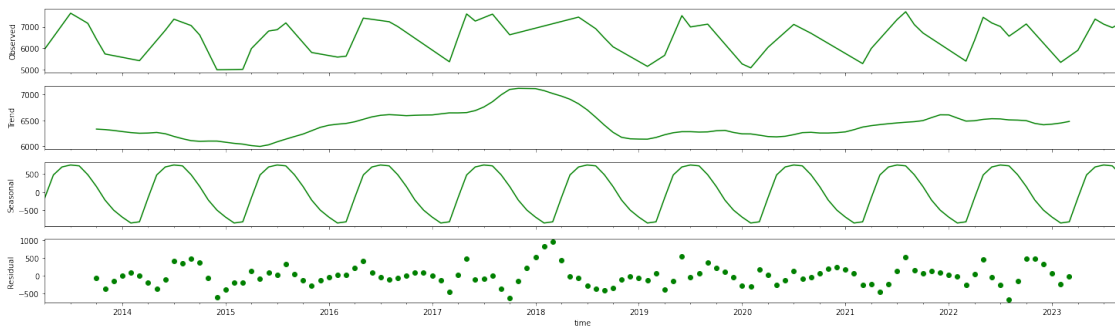
#plt.savefig('Saisons_13-23', bbox_inches='tight')
plt.show()
```



```
[48]: fig, axes = plt.subplots(4, 1, sharex=True)

additive_decomposition_pv.observed.plot(ax=axes[0], legend=False, color='g')
axes[0].set_ylabel('Observed')
additive_decomposition_pv.trend.plot(ax=axes[1], legend=False, color='g')
axes[1].set_ylabel('Trend')
additive_decomposition_pv.seasonal.plot(ax=axes[2], legend=False, color="g")
axes[2].set_ylabel('Seasonal')
additive_decomposition_pv.resid.plot(ax=axes[3], legend=False, color='g',
    ↪marker='o', linestyle="none")
axes[3].set_ylabel('Residual')

fig.savefig('pv_decomposition_13-23.png')
```

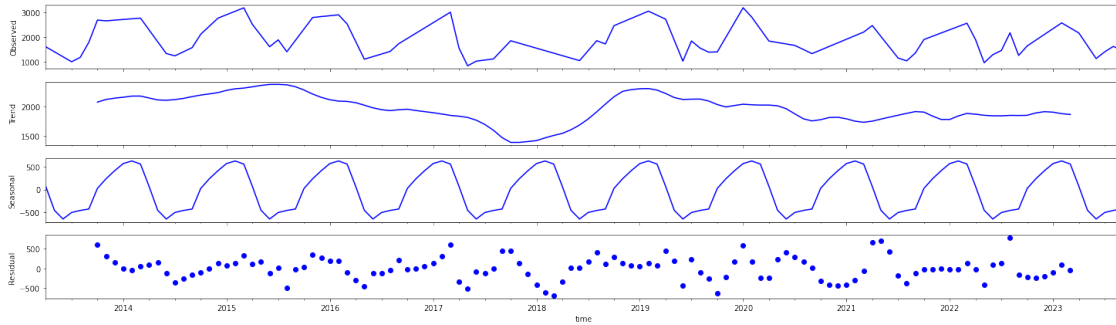


```
[49]: fig, axes = plt.subplots(4, 1, sharex=True)

additive_decomposition_npv.observed.plot(ax=axes[0], legend=False, color='b')
axes[0].set_ylabel('Observed')
additive_decomposition_npv.trend.plot(ax=axes[1], legend=False, color='b')
axes[1].set_ylabel('Trend')
additive_decomposition_npv.seasonal.plot(ax=axes[2], legend=False, color="b")
axes[2].set_ylabel('Seasonal')
additive_decomposition_npv.resid.plot(ax=axes[3], legend=False, color='b',
    ↪marker='o', linestyle="none")
axes[3].set_ylabel('Residual')

#fig.savefig('npv_decomposition_13-23.png')
```

```
[49]: Text(0, 0.5, 'Residual')
```

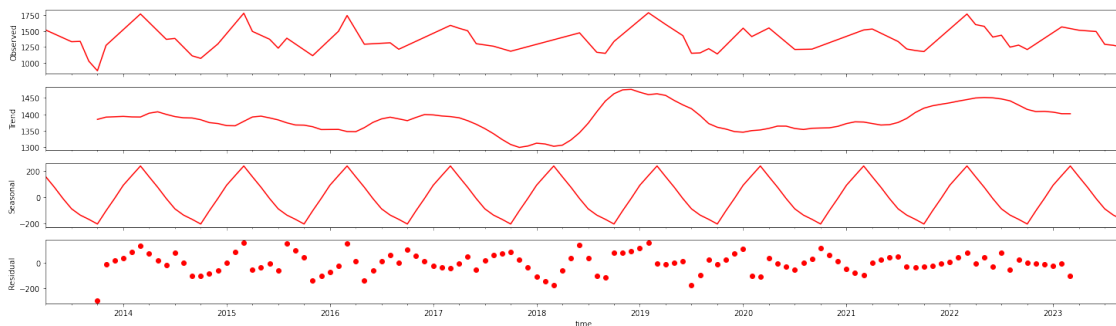


```
[50]: fig, axes = plt.subplots(4, 1, sharex=True)

additive_decomposition_bs.observed.plot(ax=axes[0], legend=False, color='r')
axes[0].set_ylabel('Observed')
additive_decomposition_bs.trend.plot(ax=axes[1], legend=False, color='r')
axes[1].set_ylabel('Trend')
additive_decomposition_bs.seasonal.plot(ax=axes[2], legend=False, color='r')
axes[2].set_ylabel('Seasonal')
additive_decomposition_bs.resid.plot(ax=axes[3], legend=False, color='r',
    ↪marker='o', linestyle="none")
axes[3].set_ylabel('Residual')

#fig.savefig('bs_decomposition_13-23.png')
```

```
[50]: Text(0, 0.5, 'Residual')
```



```
[ ]:
```

ANNEXE III - SCRIPT TENDANCE

time-series-trend-year

January 21, 2024

```
[ ]: SCRIPT YEAR TENDANCE - PV, NPV, BS
```

```
[1]: # IMPORT DEPENDENCIES

# reload module before executing code
%load_ext autoreload
%autoreload 2

# define modules locations (you might have to adapt define_mod_locs.py)
%run sdc-notebooks/Tools/define_mod_locs.py
```

```
[2]: import warnings

import xarray as xr
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from matplotlib.dates import DateFormatter

import csv

from dateutil.parser import parse
```

```
[3]: %matplotlib inline
```

```
[83]: file = pd.read_csv("years-def.csv")
```

```
[84]: file.head()
```

```
[84]:
```

	time	fid	bs_mean	bs_std	npv_mean	npv_std	pv_mean	\
0	1993	1	1115.662724	999.548267	1495.945557	730.055280	7386.878624	
1	1994	1	1115.521105	1009.553085	1505.272994	631.006951	7377.706449	
2	1995	1	1246.426261	963.845893	1766.705553	667.309208	6983.598878	
3	1996	1	1258.406920	1017.388452	2097.493226	695.581230	6642.527584	
4	1997	1	1323.910316	1046.065134	1719.139483	587.764142	6955.405577	

```

      pv_std Unnamed: 8
0  1357.617731      NaN
1  1328.282294      NaN
2  1303.164720      NaN
3  1308.094058      NaN
4  1288.321255      NaN

```

```
[85]: file.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   time        31 non-null    int64
1   fid         31 non-null    int64
2   bs_mean     31 non-null    float64
3   bs_std      31 non-null    float64
4   npv_mean    31 non-null    float64
5   npv_std     31 non-null    float64
6   pv_mean     31 non-null    float64
7   pv_std      31 non-null    float64
8   Unnamed: 8  2 non-null     object
dtypes: float64(6), int64(2), object(1)
memory usage: 2.3+ KB

```

```
[86]: # convert column to datetime pandas
file['time'] = pd.to_datetime(file['time'], format="%Y")
```

```
[87]: file.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   time        31 non-null    datetime64[ns]
1   fid         31 non-null    int64
2   bs_mean     31 non-null    float64
3   bs_std      31 non-null    float64
4   npv_mean    31 non-null    float64
5   npv_std     31 non-null    float64
6   pv_mean     31 non-null    float64
7   pv_std      31 non-null    float64
8   Unnamed: 8  2 non-null     object
dtypes: datetime64[ns](1), float64(6), int64(1), object(1)

```

memory usage: 2.3+ KB

```
[88]: # displaying the datatypes
display(file.dtypes)
```

```
time          datetime64[ns]
fid           int64
bs_mean       float64
bs_std        float64
npv_mean      float64
npv_std       float64
pv_mean       float64
pv_std        float64
Unnamed: 8    object
dtype: object
```

```
[89]: file.head()
```

```
[89]:
```

	time	fid	bs_mean	bs_std	npv_mean	npv_std	\
0	1993-01-01	1	1115.662724	999.548267	1495.945557	730.055280	
1	1994-01-01	1	1115.521105	1009.553085	1505.272994	631.006951	
2	1995-01-01	1	1246.426261	963.845893	1766.705553	667.309208	
3	1996-01-01	1	1258.406920	1017.388452	2097.493226	695.581230	
4	1997-01-01	1	1323.910316	1046.065134	1719.139483	587.764142	

	pv_mean	pv_std	Unnamed: 8
0	7386.878624	1357.617731	NaN
1	7377.706449	1328.282294	NaN
2	6983.598878	1303.164720	NaN
3	6642.527584	1308.094058	NaN
4	6955.405577	1288.321255	NaN

```
[90]: x = np.arange(file['time'].size)
```

```
[105]: fit = np.polyfit(x, file['pv_mean'], deg=1)
print ("Slope : " + str(fit[0]))
print ("Intercept : " + str(fit[1]))
```

```
Slope : -0.46357752103571404
Intercept : 7052.024864151101
```

```
[108]: #Fit function :  $y = mx + c$  [linear regression ]
fit_function = np.poly1d(fit)

plt.figure(figsize=(20,5))

#Linear regression plot
```

```

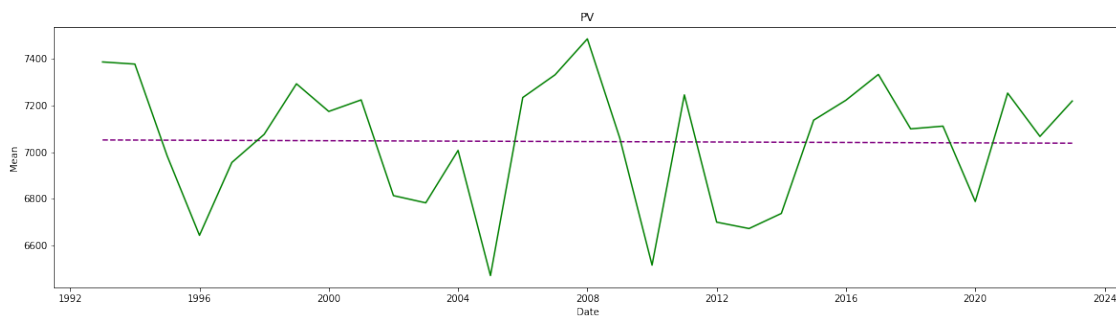
plt.plot(file['time'], fit_function(x), color='purple',
         linestyle='dashed')#Time series data plot
plt.plot(file['time'], file['pv_mean'], color='green')

#ax = plt.gca()
#ax.set_ylim([0, 10000])

plt.xlabel('Date')
plt.ylabel('Mean')
plt.title('PV')

#plt.savefig('bs_trend_years.png', bbox_inches='tight')
plt.show()

```



```

[102]: fit = np.polyfit(x, file['npv_mean'], deg=1)
print ("Slope : " + str(fit[0]))
print ("Intercept : " + str(fit[1]))

```

Slope : -8.737641022109594
Intercept : 1738.4227572738093

```

[104]: #Fit function : y = mx + c [linear regression ]
fit_function = np.poly1d(fit)

plt.figure(figsize=(20,5))

#Linear regression plot
plt.plot(file['time'], fit_function(x), color='purple',
         linestyle='dashed')#Time series data plot
plt.plot(file['time'], file['npv_mean'], color='blue')

#ax = plt.gca()
#ax.set_ylim([0, 10000])

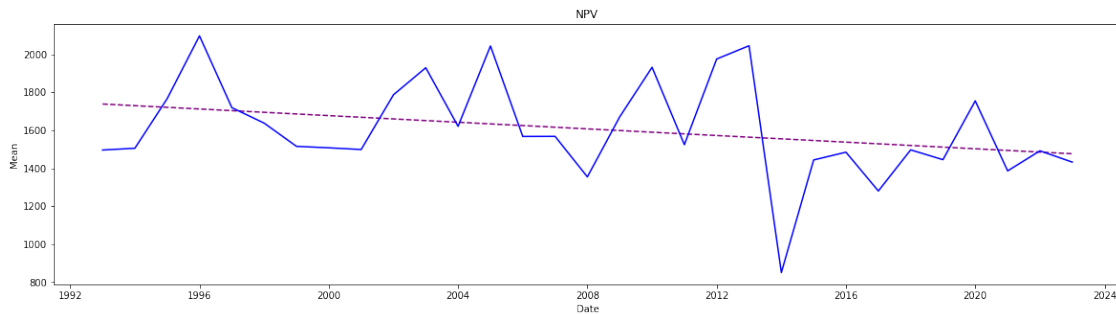
plt.xlabel('Date')

```



```
plt.ylabel('Mean')
plt.title('NPV')

#plt.savefig('bs_trend_years.png', bbox_inches='tight')
plt.show()
```



```
[96]: fit = np.polyfit(x, file['bs_mean'], deg=1)
print ("Slope : " + str(fit[0]))
print ("Intercept : " + str(fit[1]))
```

Slope : 5.848862080808633
Intercept : 1212.5376264049878

```
[101]: #Fit function : y = mx + c [linear regression ]
fit_function = np.poly1d(fit)

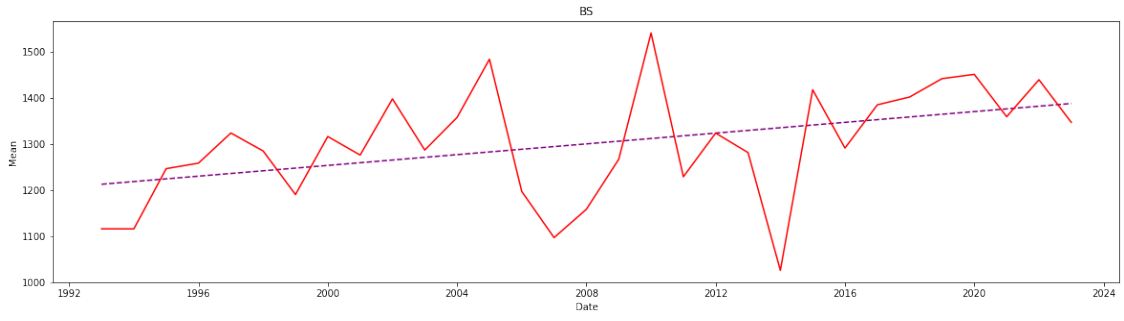
plt.figure(figsize=(20,5))

#Linear regression plot
plt.plot(file['time'], fit_function(x), color='purple',
         ↪linestyle='dashed')#Time series data plot
plt.plot(file['time'], file['bs_mean'], color='red')

#ax = plt.gca()
#ax.set_ylim([0, 10000])

plt.xlabel('Date')
plt.ylabel('Mean')
plt.title('BS')

#plt.savefig('bs_trend_years.png', bbox_inches='tight')
plt.show()
```



[]:

ANNEXE IV – Données moyenne et déviation standard annuelles, 1993-2023, Grand Genève.

time	fid	bs_mean	bs_std	npv_mean	npv_std	pv_mean	pv_std	TL
1993	1	1115.6627235781	999.548266760816	1495.94555686263	730.055280487485	7386.87862408463	1357.61773121626	9998.48690452536
1994	1	1115.52110461398	1009.55308452143	1505.27299374341	631.00695099389	7377.70644908844	1328.28229370943	9998.50054744583
1995	1	1246.42626092459	963.845893231512	1766.70555335028	667.309208092229	6983.59887813209	1303.16472026946	9996.73069240696
1996	1	1258.40692040332	1017.38845204575	2097.49322583872	695.581229901641	6642.52758389228	1308.09405774432	9998.42773013432
1997	1	1323.91031619693	1046.06513425314	1719.13948304276	587.764141880189	6955.40557656977	1288.32125490036	9998.45537580946
1998	1	1284.79011030841	1006.31784739995	1637.56979945694	617.110463175945	7076.12731793183	1359.35395849943	9998.48722769718
1999	1	1190.13238548334	985.782996181349	1515.30509864662	676.123710815055	7293.0631936709	1352.30453158666	9998.50067780086
2000	1	1316.33007340165	1037.95117195062	1507.61316060689	621.776475292274	7174.42172022463	1342.98684407842	9998.36495423317
2001	1	1275.92318830281	976.809066567493	1498.53684588823	712.028789272613	7224.00009595578	1325.14938037619	9998.46013014682
2002	1	1398.01997464414	942.332603945129	1787.11231080983	731.17347777124	6813.03506821687	1282.5010974067	9998.16735367084
2003	1	1286.82229985186	1201.38527026313	1929.54506459137	1139.9771227603	6782.07861856265	1926.94970690435	9998.44598300588
2004	1	1357.7243231157	1076.49724406778	1620.76951738316	743.893790902668	7007.07788305253	1516.37684320321	9985.57172355139
2005	1	1484.22603425169	1046.83815438512	2044.3661509647	903.157253974389	6469.84569224175	1591.76954922424	9998.43787745814
2006	1	1196.84829753172	978.900028106407	1567.43486616658	591.923582013391	7234.17251265191	1285.54450302335	9998.45567635021
2007	1	1096.56152372348	1035.20501021423	1567.88756426661	795.580444965866	7331.76115746208	1485.58809401353	9996.21024545217
2008	1	1158.09206504535	985.195297322485	1354.41543874197	589.314381820768	7485.98024351948	1289.05300460761	9998.4877473068
2009	1	1266.89801303641	1028.73586270052	1671.17007754766	801.428788648711	7060.37818209963	1538.31445574184	9998.44627266837
2010	1	1541.2667385128	1054.55982830079	1931.95058548868	820.843335482995	6514.67534764734	1380.30277237087	9987.89267164882
2011	1	1228.8159346342	1091.13771734137	1524.02244097927	653.198162835616	7245.61678791645	1447.9588441235	9998.45516352992
2012	1	1323.73244269471	980.372852072221	1975.58133768695	724.880631797172	6699.14190954719	1351.10905462359	9998.45568992885
2013	1	1281.25311098146	949.570311777795	2045.45816124283	756.062319745985	6671.71256310111	1290.73499957593	9998.4238353254
2014	1	1025.29064281773	2299.75476184327	850.136276218672	2154.66196984033	6736.60084024675	3408.27269636116	8612.02775928315
2015	1	1417.84948611606	985.832875553649	1443.63614744783	649.11617715236	7137.01509674108	1335.9043713978	9998.50073030497
2016	1	1291.14148136715	1013.52993413488	1484.4872322234	850.015307094028	7222.70866693432	1442.99104798109	9998.33738052487
2017	1	1385.14012123922	1004.47557759481	1280.05574306663	645.895741188709	7333.28655609724	1277.31518715108	9998.48242040309
2018	1	1402.12748992804	992.249230812105	1496.7006057435	702.10430465363	7099.67224673671	1395.67085613273	9998.50034240825
2019	1	1442.0010871971	1010.70425431003	1445.09886613761	691.455608825339	7111.33788972418	1321.59739266472	9998.43784305889
2020	1	1451.38110060375	1006.51721863255	1754.86132171846	911.878622179522	6787.83715625086	1334.84892635694	9994.07957857307
2021	1	1359.31094515991	1008.27537081597	1385.95769300577	670.215395003296	7253.20041903708	1270.16646964777	9998.46905720276
2022	1	1439.66399861317	984.112065025327	1491.68279508326	619.444764934065	7067.15398096545	1317.28697428877	9998.50077466188
2023	1	1347.11709185186	968.918344461821	1432.19048625592	569.631259727776	7219.18898309956	1216.30059461095	9998.49656120734

ANNEXE VI – CARTES MOYENNE ANNUELLE FC



BIBLIOGRAPHIE

- Acevedo-Reyes, S., Pons, L. (1998). Spatial interpolation methods for environmental data analysis.
- Balling Jr, R. C. (1988). The climatic impact of a Sonoran vegetation discontinuity.
- Brossin, L., (2017). Mémoire Certificat Complémentaire en Géomatique, Swiss Data Cube: Observations de la Terre pour le monitoring de l'environnement en Suisse dans le temps et dans l'espace Programme des Nations Unies pour l'Environnement GRID Genève.
- Chatenoux., B., Richard, J-P., Small, D., Roeoesli, C., Wingate V., Poussin C., Rodila D., Peduzzi, P., Steinmeier, C., Ginzler, C., Psomas, A., Shaepman, M.E., Giuliani, G., (2021). The Swiss data cube, analysis ready data archive using earth observations of Switzerland.
- Chander, G., Markham, B. L., Barsi, J. A., (2007). Revised Landsat-5 Thematic Mapper Radiometric Calibration.
- Eldén, L.. (2004). Partial least-squares vs. Lanczos Bidiagonalization – I: Analysis of a projection method for multiple regression.
- Flood, N. (2014). Continuity of reflectance data between Landsat-7 ETM+ and Landsat-8 OLI, for both top-of-atmosphere and surface reflectance: A study in the Australian landscape.
- Foley, J. A., Levis, S., Heil Cosat, M., Cramer, W., Pollard, D., Incorporating (2000). Incorporating dynamic vegetation cover within global climate models.
- García-Haro, F. J., Gilabert , M. A., Melià, J. (1996) Linear spectral mixture modelling to estimate vegetation amount from optical spectral data.
- Grossman, M. A., Spain, J. A., Breckenridge, J. P., & Goodchild, M. F. (1996). An introduction to the development of spatial interpolators for use in geographic information systems.
- Guerschman, J. P., Hill, M. J., Renzullo, L. J., BarrettD. J., Marks, A. S., Botha, E. J. (2009). Estimating fractional cover of photosynthetic vegetation, non-photosynthetic vegetation and bare soil in the Australian tropical savanna region upscaling the EO-1 Hyperion and MODIS sensors.
- Guerschman, J.P., Scarth, P. F., McVicar, T. R., Renzillo, L. J., Malthus, T. J., Stewart, J. B., Rickards, J. E., Trevithick, R. (2015). Assessing the effects of site heterogeneity and soil propoerties when unmixing photosynthetic vegetation, non-photosynthetic vegetation and bare soil fractions from Landsat and MODIS data.
- Heinz, D. C., Fully, C-I-C., (2001) .Constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery.
- Hill, M. J., Guerschman, J. P. (2020). The MODIS Global Vegetation Fractional Cover Porduct 2001-2018 : Characteristics of Vegetation Fractional cover in Grasslands and Savanna Woodlands.
- Hill, M. J., Guerschman, J. P. (2021). Global trends in vegetation fractional cover: Hotspots for change in bare soil and non-photosynthetic vegetation.

Kalivas, J. H. (1999). Interrelationships of multivariate regression methods using eigenvector basis sets.

Keshava, N. (2003). A Survey of Spectral Unmixing Algorithms.

Korb, J., Covington, W. W., Fulé, P. (2003). Sampling Techniques Influence Understory Plant Trajectories After Restoration: An Example from Ponderosa Pine Restoration.

Kovalskyy *,V., Roy, D.P. (2013). The global availability of Landsat 5 TM and Landsat 7 ETM + land surface observations and implications for global 30 m Landsat data product generation.

Krause, C., Dunn, B., Bishop-Taylor, R., Adams, C., Burton, C., Alger, M., Chua, S., Phillips, C., Newey, V., Kouzoubov, K., Leith, A., Ayers, D., Hicks, A., DEA Notebooks contributors 2021. Digital Earth Australia notebooks and tools repository. Geoscience Australia, Canberra.

Lawson, C. L., Hansons, R. J. (1995). Solving Least Squares Problems.

Li, X.; Zheng, G.; Wang, J.; Ji, C.; Sun, B.; Gao, Z. (2015). Comparison of Methods for Estimating Fractional Cover of Photosynthetic and Non-Photosynthetic Vegetation in the Otindag Sandy Land Using GF-1 Wide-Field View Data.

Markham, B. L., Storey, J. C., Williams, D. L., Irons, J. R. (2004). Landsat sensor performance: history and current status .

Muir, J., Schmidt, M., Tindall, D., Trevithick, R., Scarth, P. and Stewart, J.B. (2011). Guidelines for field measurement of fractional ground cover: a technical handbook supporting the Australian Collaborative Land Use and Management Program.

Nagler, P., Inoue, Y., Glenn, E., Russ, A., Daughtry, C. (2003). Cellulose absorption index (CAI) to quantify mixed soil-plant litter scenes.

Okin, G. S., Clarke, K. D., Lewis, M. M. (2013). Comparison of methods for estimation of absolute vegetation and soil fractional cover using MODIS normalized BRDF-adjusted reflectance data.

Okin, Gregory. (2007). Relative spectral mixture analysis — A multitemporal index of total vegetation cover. Remote Sensing of Environment.

Peiliang, Xu, (1998). Truncated SVD methods for discrete linear ill-posed problems.

Plaza, A., Martínez, P., Plaza, J. (2022). Spatial/spectral endmember extraction by multidimensional morphological operations.

Roberts, D. A., & Smith, T. R. (2006). A comparison of spatial interpolation methods for large-area environmental modeling: A review.

Roberts, D. A., Dennison, P. E., Gardner, M. E., Hetzel, Y., Ustin, S. L., Lee, C. T. (2003). Evaluation of the Potential of Hyperion for Fire Danger Assessment by Comparison to the Airborne Visible/Infrared Imaging Spectrometer.

Scarth, P., Roder, A. and Schmidt, M. (2010). Tracking grazing pressure and climate interaction - the role of Landsat fractional cover in time series analysis. *Proceedings of the 15th Australasian Remote Sensing & Photogrammetry Conference*.

Schmidt, M., Denham, R. and Scarth, P. (2010), Fractional ground cover monitoring of pastures and agricultural areas in Queensland. *Proceedings of the 15th Australasian Remote Sensing & Photogrammetry Conference*.

Settle, J. J., Drake, N. A., (1993). Linear mixing and the estimation of ground cover proportions.

Settle, J. J. (1996). On the relationship between spectral unmixing and subspace projection.

Small, C., (2014). The Landsat ETM+ spectral mixing space

Stephen H. (2003), Linear Mixture Modelling Solution Methods for Satellite Remote Sensing, College of Aeronautics Report 9603.

Sutton, A., Fisher, A., Metternicht, G. (2022). Assessing the Accuracy of Landsat Vegetation Fractional Cover for Monitoring Australian Drylands

Vermeulen, L.M., Munch, Z., Palmer, A. (2021). Fractional vegetation cover estimation in southern African rangelands using spectral mixture analysis and Google Earth Engine

Williams, Darrel & Goward, Samuel & Arvidson, Terry. (2006). Landsat: Yesterday, Today, and Tomorrow. *Photogrammetric Engineering & Remote Sensing*.

Sites web

<https://datacube-core.readthedocs.io/en/latest/data-access-analysis/advanced-topics/data-loading.html>

<https://datascientest.com/cross-validation>

<https://www.dea.ga.gov.au/>

<https://www.digitalearthafrica.org/>

<https://natural-resources.canada.ca/maps-tools-publications/satellite-imagery-air-photos/remote-sensing-tutorials/>

<https://portal.tern.org.au/metadata/TERN/f0c32576-9ad7-4c9c-9aa9-22787867e28b>

<https://www.fr.wikipedia.org>

<https://www.grand-geneve.org/>

<https://www.kaggle.com/code/prashant111/complete-guide-on-time-series-analysis-in-python#5.-Visualize-the-Time-Series->

<https://www.meteosuisse.admin.ch>

<https://www.nv5geospatialsoftware.com/docs/LinearSpectralUnmixing.html>

<https://www.nv5geospatialsoftware.com/docs/LinearSpectralUnmixing.html>

Algorithme

Joint Remote Sensing Research Program (2022): Seasonal fractional cover - Landsat, JRSRP algorithm Version 3.0, Australia coverage. Version 1.0. Terrestrial Ecosystem Research Network. (Dataset). <https://portal.tern.org.au/metadata/TERN/0997cb3c-e2e2-45be-ac82-f5e13d24331c>