Rapport de stage assistant ingénieur : Ajout d'un module de visualisation de données satellites à la plateforme urban-sprawl

Valentin GUIMONT
Inria Grenoble Rhône Alpes

14/02/2020 - 10/03/2020

Contents

Introduction

Ce sujet consistait à compléter une API existante appelée urbansprawl permettant d'obtenir différentes informations sur l'étalement urbain. Ces informations sont dédiées aux élus et aux planificateurs urbains pour les aider à prendre en compte les impacts socio-économiques et environnementaux de leurs décisions. Au début du stage, cette API permettait entre autres de visualiser des indices liés à l'étalement urbain sur une zone ciblée. L'objectif était de visualiser en plus des images satellite de la zone en question, et ce dans différents spectres afin d'obtenir éventuellement d'autres informations pertinentes. Dans un souci d'accessibilité et de reproductibilité, les images devaient provenir de plateformes ouvertes.
Les tâches à réaliser étaient donc :

- Récupérer les images satellites de la zone.
- En tirer des visualisations dans différents spectres indiquant des informations liées à l'étalement urbain.
- Afficher les visualisations avec le reste des indices déjà affichés.

1 Contexte

Le stage s'est déroulé dans les locaux d'Inria Grenoble-Rhônes-Alpes, à Montbonnot St Martin. Je faisais alors partie de l'équipe Steep. L'API urbansprawl, qui est le sujet du stage, est un projet développé par différents chercheurs de l'équipe Steep. Au début du stage, cette API pouvait générer une page html permettant de visualiser les bâtiments, le réseau routier et 3 grandeurs liées à l'étalement urbain : la dispersion, la mixité du territoire (land use mix), et l'accessibilité (cf. Gervasoni et al. 2017 [1]). Définie qualitativement, la dispersion est la mesure de la densité des bâtiments dans une zone urbaine, et de la place qu'ils occupent. La mixité du territoire correspond à la proximité, voire à la mixité, de zone résidentielles et d'installations de loisirs. Et l'accessibilité mesure la connectivité de différents types de lieux, en terme de réseau routier.
En prenant l'exemple de Grenoble, on pouvait obtenir ces plans :

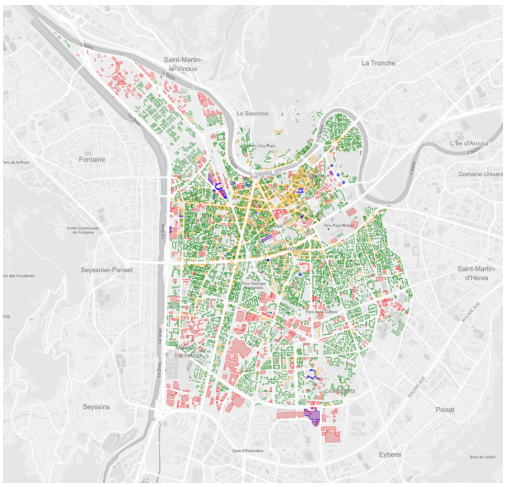

Figure 1: Bâtiments de Grenoble

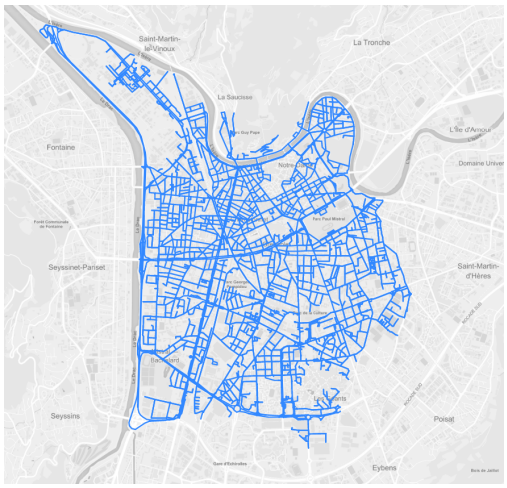

Figure 2: Réseau routier de Grenoble

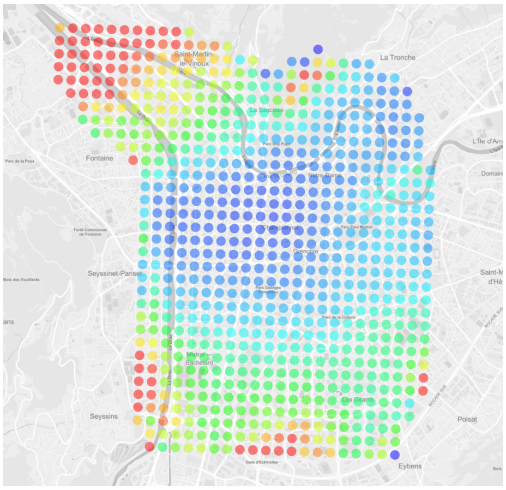

Figure 3: Mesure de dispersion de Grenoble

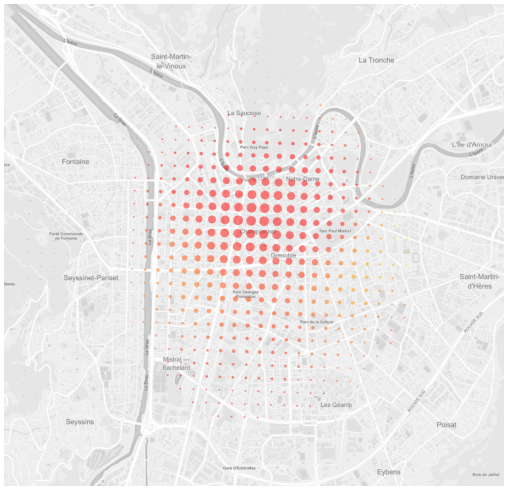

Figure 4: Mesure de 'land use mix' de Grenoble

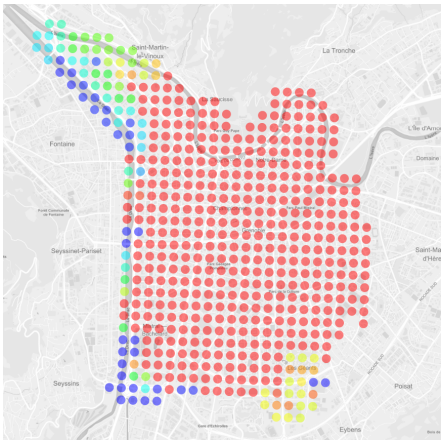

Figure 5: Mesure d'accessibilité de Grenoble

Ces indices sont calculés à partir de données provenant de la plateforme ouverte OpenStreetMap, sont récupérées avec l'API osmnx, et affichées avec le package folium. L'objectif du stage est d'utiliser des données provenant de satellites à plateforme ouverte (Sentinel pour l'Europe, Landsat pour les Etats-Unis ...) afin d'obtenir des informations sur l'étalement urbain, et d'afficher ces informations.

2 Problématique

2.1 Problème à résoudre

L'étalement urbain est relié à de nombreux impacts socio-économiques et environnementaux négatifs. Au cours des dernières décennies, les zones urbaines ont énormément augmenté en population et en taille, et de plus en plus de zones naturelles sont utilisées par les villes. Le besoin d'une approche durable à l'urbanisation se fait donc de plus en plus urgente. L'étalement urbain est cependant un phénomène particulièrement dur à quantifier, et différentes façons de le mesurer vont mener à différents résultats. Certains travaux déjà tentés assignaient une seule valeur à une région entière ou se basaient sur des données privées ou commerciales. De telles approches sont évidemment d'utilité limitée pour les législateurs et les planificateurs urbains. Des données publiques telles que le recensement sont parfois utilisées pour caractériser le développement de l'urbanisation, mais ces informations manquent de précision et de granularité.

C'est avec ce problème en tête qu'a été conduit le projet urbansprawl. Développé par des membres de l'équipe Steep de l'Inria Grenoble-Rhônes-Alpes, il se veut une réponse à cette problématique en proposant une solution open-source à haute granularité basée sur des données libres. Cette API mermet d'obtenir la dispersion, l'accessibilité et la mixité de l'usage de la zone urbaine voulue. Ces valeurs sont calculées à partir de données provenant de la plateforme participative libre OpenStreetMap. Cette méthode présente l'avantage de fournir des données homogènes géo-localisées couvrant le monde entier.

Dans ce cadre là, l'objectif du stage est d'utiliser les ressources mises à disposition par des programmes satellites ouverts comme Sentinel ou Landsat afin d'obtenir d'autres informations sur l'étalement urbain sur une zone donnée. Par exemple, Copernicus est la plateforme officielle permettant d'accéder aux images des satellites européens comme Sentinel. Il existe cependant des API et des applications facilitant la récupération de ces images. Les données en question sont sous la forme de paquets d'images sur différentes bandes de fréquence. Le programme Sentinel-2, par exemple, propose des images sur 13 bandes allant du visible à l'infrarouge proche. Ces images peuvent être manipulées pour détecter différentes grandeurs intéressantes dans le cadre de l'étude de l'étalement urbain. Il est par exemple possible de calculer un indice de végétation du sol à partir des bandes 4 et 8 de Sentinel-2. Le but final du stage est de pouvoir visualiser ces indices sur une zone donnée en même temps que les autres gradeurs calculées à partir d'OpenStreetMap (accessibilité, dispersion et mixité d'usage).

Pour atteindre ces objectifs, il est nécessaire de pouvoir récupérer les images d'un zone donnée en un temps raisonnable, pouvoir manipuler ces images pour créer les indices et les visualisations qui nous intéressent et enfin les afficher en même temps que les autres indices.

2.2 Etat de l'art des solutions existantes

Il existe plusieurs moyens de récupérer des images satellites.

Copernicus

La plateforme officielle européenne pour les images de Sentinel est Copernicus, qui met à disposition le service OpenAccessHub (https://scihub.copernicus.eu/dhus/#/home). Ce service permet de télécharger des images prises par les satellites des programmes Sentinel-1, 2, et 3. Ces images peuvent être filtrées en fonction de critères dépendant des capacités du satellite. Par exemple il est possible de filtrer les images de Sentinel-2 par couverture de nuage et celles de Sentinel-1 par polarisation. Copernicus propose aussi les API OpenSearch et OpenData, qui sont des normes URI permettant d'effectuer des requêtes à la vase de données DataHub et récupérer les produits sous la forme de pages XML.

Sentinelsat

Sentinelsat est le wrapping Python des API OpenSearch et OpenData précédemment mentionnées. C'est une interface permettant de récupérer et enregistrer de plusieurs manières des données des satellites Sentinel. Cette API ne permet cependant que de récupérer les métadonnées des images satellites.

CNES (Centre National d'Etudes Spatiales)

Le CNES propose aussi une application pour récupérer les données des programmes Sentinel appelée PEPS (Plateforme d'Exploitation des Produits Sentinel : https://peps.cnes.fr/rocket/#/search?ma Elle founit le même type de service que OpenAcessHub : il est possible de récupérer des images des programmes Sentinel sur une zone donnée. PEPS propose cependant d'autres critères de sélection plus techniques comme le sens de l'orbite du satellite ou le mode de capture. La plateforme n'offre cependant pas d'API et les requêtes sont plus lentes que sur OpenAcessHub.

SeDAS (Sentinel Data Access Service)

SeDAS est une application permettant de récupérer toutes les images prises par les satellites Sentinel sur une zone donnée pendant une période donnée : https://geobrowser.satapps.org/. Les produits sont sous format .jp2 et contiennent les images de chaque bande de fréquence. Ce service n'offre cependant pas d'API.

AWS

Amazon Web Service est en lien avec plusieurs services et applications permettant d'obtenir des données satellite.

- Earth Observation System Land Viewer (https://eos.com/landviewer/), gérée par Earth Observing System est une application similaire à Earth Observation Browser permettant de récupérer les images de la zone souhaitée sous format .geojson. L'application limite cependant les téléchargements à 10 images par jour.

- Earth Observation Browser est une application gérée par Sinergise, qui est reliée à AWS : https://apps.sentinel-hub.com/eo-browser/. Elle permet de récupérer des images des programmes Sentinel, Landsat, Envisat-Meris, Modis, Proba-V et Gibs. Elle offre aussi des visualisations prédéfinies, par exemple des indices de végétation ou d'humidité et permet de créer des visualisations personnalisées en assemblant les bandes spectrales souhaitées en une image rgb. Les images récupérées sont sous format jp2.

- Sentinel Playground (https://apps.sentinel-hub.com/sentinel-playground/) est une autre application produite par Sinergise ayant des fonctionnalités similaires à Earth Observation Browser et permet la récupération d'images sous format .jpg. Elle fonctionne grâce à l'API Python SentinelHub (https://sentinelhub-py.readthedocs.io/en/latest/examples/ogc_request. Cette API permet entre autres d'effectuer des requêtes à travers un compte SentinelHub et ainsi de récupérer des images des satellites Sentinel-1, Sentinel-2, Landsat, MODIS et DEM. Les images sont récupérées sous forme d'array numpy.

2.3 Solutions choisies

Au vu de sa facilité d'utilisation et de l'étendue de ses possibilités, l'API SentinelHub est choisie pour effectuer des requêtes d'images satellites.

SentinelHub fournit les données sous la forme d'array numpy. Les indices et visualisations sont créés en effectuant des opérations sur les pixels des images correspondant aux différentes bandes de fréquences. Ces opérations sont facilement faites en local après la requête avec les méthodes de la librairie numpy. SentinelHub offre cependant un moyen d'entrer des evalscript (morceaux de code javascript) dans la requête pour que les opérations de transformations soient déja effectuées à la sortie de la requête. Cela implique de faire une requête par visualisation voulue (une requête pour l'indice de végétation, une pour l'indice d'humidité ...), mais cette méthode reste plus rapide à l'exécution que de faire les manipulation localement.
Le site IndexDataBase ([2]) fournit toutes les visualisations et indices fabriquables à partir des bandes de fréquence des images d'un satellite. On va utiliser les plus significatifs, à savoir : vraie couleur, infrarouge, infrarouge proche, agriculture, géologie, bathymétrie, et indices de végétation et d'humidité.

Concernant l'affichage des images satellites, la librairie folium, déja utilisée par urbansprawl pour afficher les indices d'étalement urbain, permet l'ajout d'une couche contenant une image sous forme d'un taleau de pixels, on utilise donc cette technologie.

Pour résumer, les images sont donc récupérées par une requête SentinelHub sous la forme d'array numpy. Un evalscript est utilisé afin d'avoir la visualisation qu'on désire en sortie de requête. On l'affiche ensuite en ajoutant une couche folium contenant l'image à la carte folium déja utilisée pour afficher les données d'OpenStreetMap.

3 Conception

3.1 Architecture de la solution

L'API urbansprawl fonctionne de base de cette manière pour afficher les indices d'étalement urbain d'une région : La région voulue est au préalable décrite en un fichier .json la caractérisant soit par son nom, soit par les coordonnées terrestres la délimitant (sa "bounding box"). Lors de la construction des fichiers html finaux, 2 fichiers sont exécutés : build_data.py et build_html.py . Le fichier build_data.py va lire le fichier .json caractérisant la zone urbaine

choisie et récupérer les données OpenStreetMap correspondantes. Le programme va ensuite calculer les indices d'étalement urbain vu précédemment : la dispersion, l'accessibilité et la mixité d'usage. Ces données seront encodées en un fichier .geojson . Ensuite, lors de l'éxécution de build_html.py, le fichier .geojson va être lu et les données vont être ajoutées en couches à une carte, qui va être utilisée par la librairie folium pour générer le fichier html final.

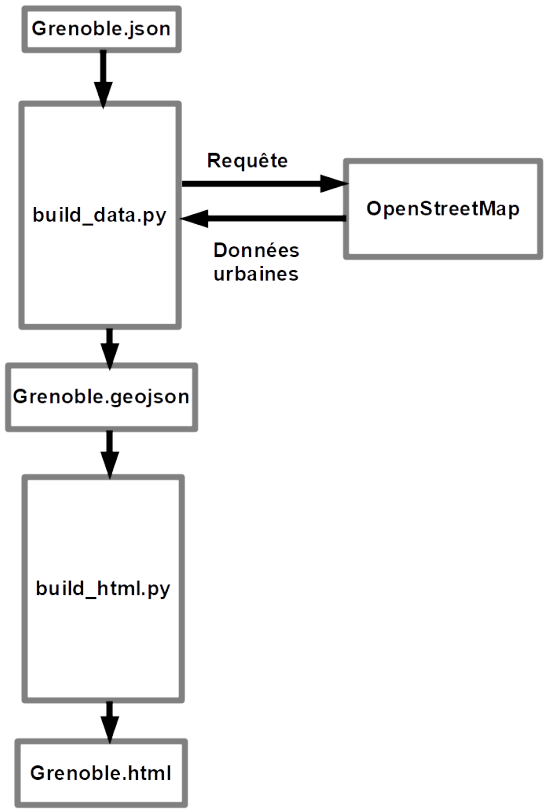

Figure 6: Construction du fichier html avant le stage

La récupération des images satellites, la fabrication des différentes visualisations, et l'ajout des couches folium contenant les images sont toutes gérées par un package que j'ai créé appelé plot_sentinel. Le package est maintenant utilisé dans build_html.py pendant la création de la carte folium. Il prend en argument la bounding box de la zone et la carte et y ajoute les couches folium contenant les visualisations satellites de la zone.

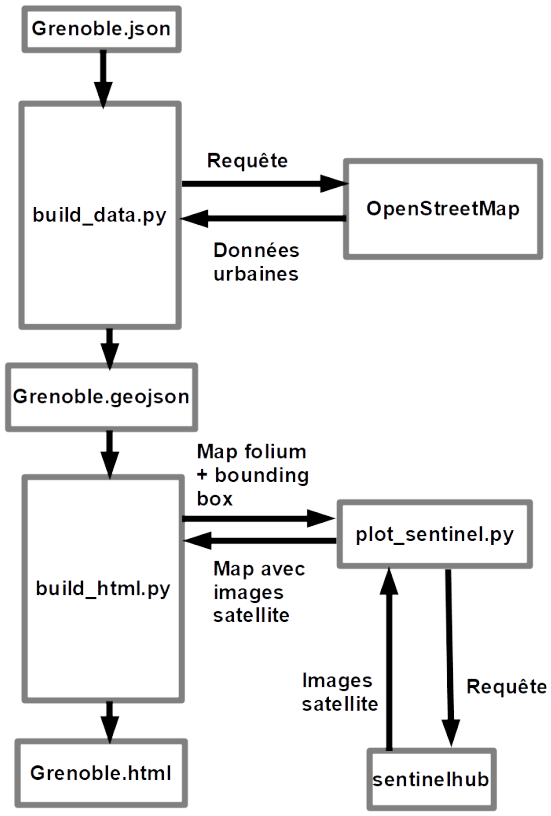

Figure 7: Construction du fichier html avec images satellites

3.2 Tests

Le calcul et l'affichage des données d'une ville avec urbansprawl demande de définir la zone voulue avec un fichier .json. La zone peut-être définie avec les coordonnées terrestres d'un point et la distance entre ce point et les bords de la zone. Si on veut récupérer les données d'une ville entière, on peut indiquer le nom et l'adresse de la ville voulue.
Une batterie de fichiers .json pour différentes villes a ainsi été préparée. La fonction d'affichage de données satellites a été testée sur tous les fichiers tests. Une couverture de tests exhaustive est irréalisable car elle demanderait de récupérer les données de la planète entière. Ci-dessous les résultats de certains fichiers-tests pour les images satellites en vraie couleur.

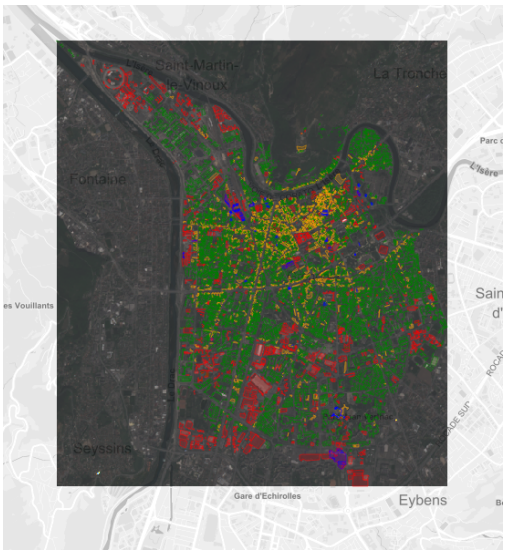

Figure 8: Image en vraie couleur de Grenoble

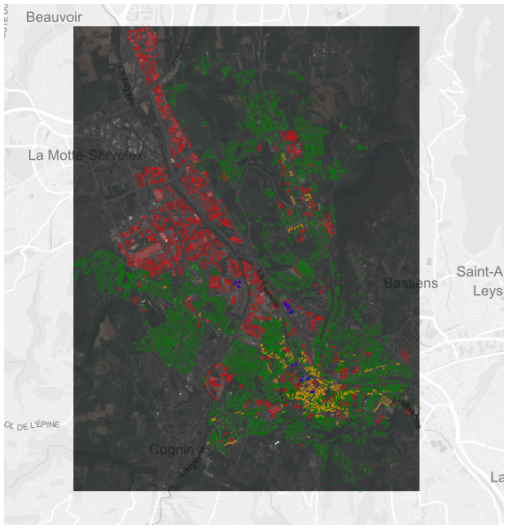

Figure 9: Image en vraie couleur de Chambery

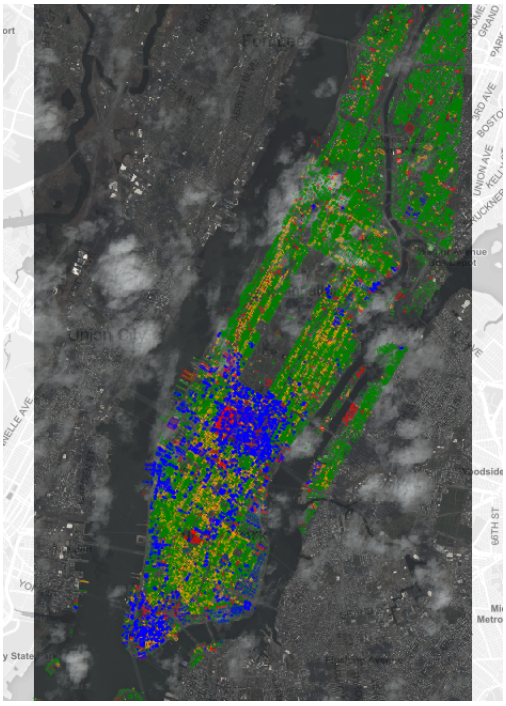

Figure 10: Image en vraie couleur de Manhattan

Figure 11: Image en vraie couleur de l'Isle d'Abeau

3.3 Résultats obtenus

L'affichage des images satellites est fonctionnelle est les différents indices (végétation, humidité ...) obtenus sont en accord avec les visualisations d'autres applications utlisant des données satellites. On peut également voir qu'ils sont en accord avec ce à quoi on peut

s'attendre par intuition (taux d'humidité très haut et végétation très basse sur les fleuves, indice de végétation très bas en ville sauf dans les parcs etc ...). Ci-dessous les indices de végétation et d'humidité de quelques villes-test.

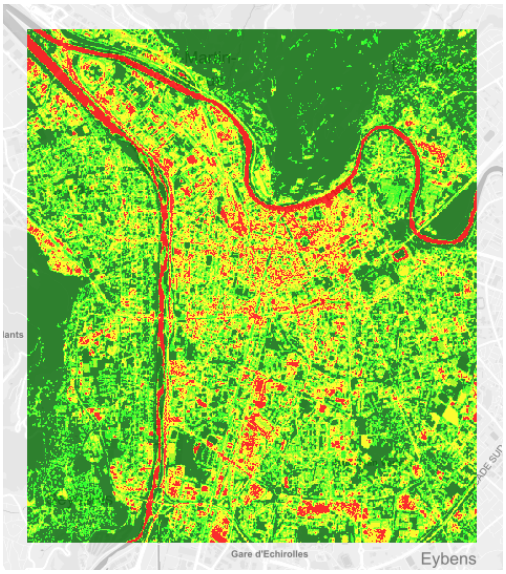

Figure 12: Indice de végétation de Grenoble

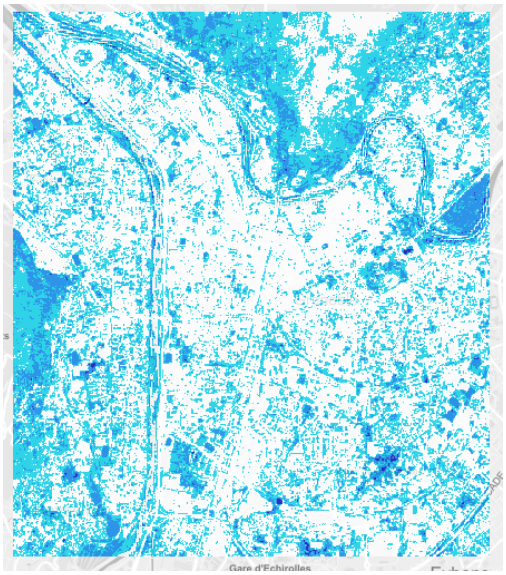

Figure 13: Indice d'humidité de Grenoble

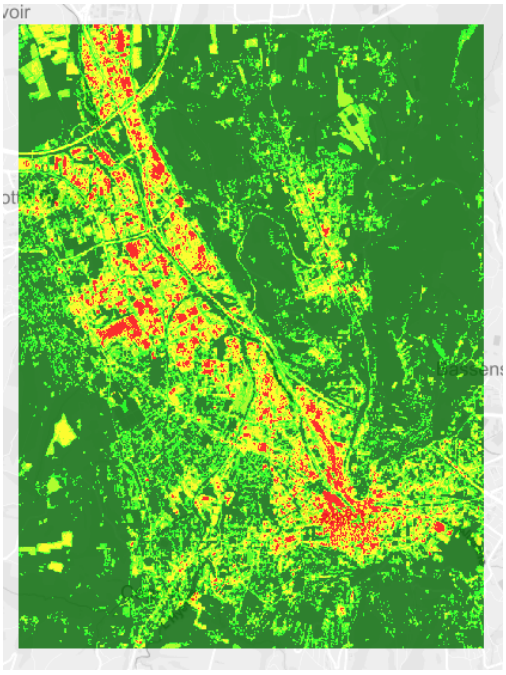

Figure 14: Indice de végétation de Chambery

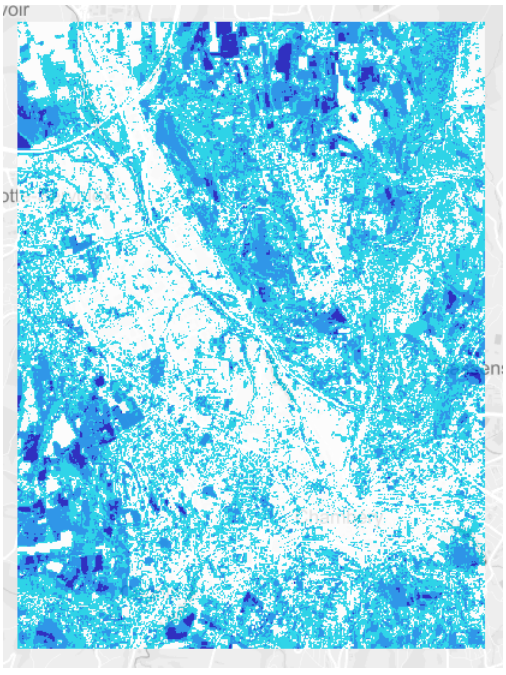

Figure 15: Indice d'humidité de Chambery

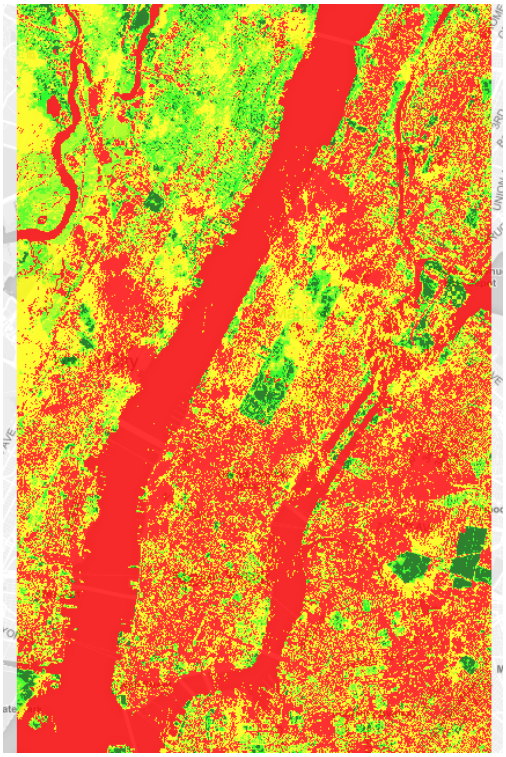

Figure 16: Indice de végétation de Manhattan

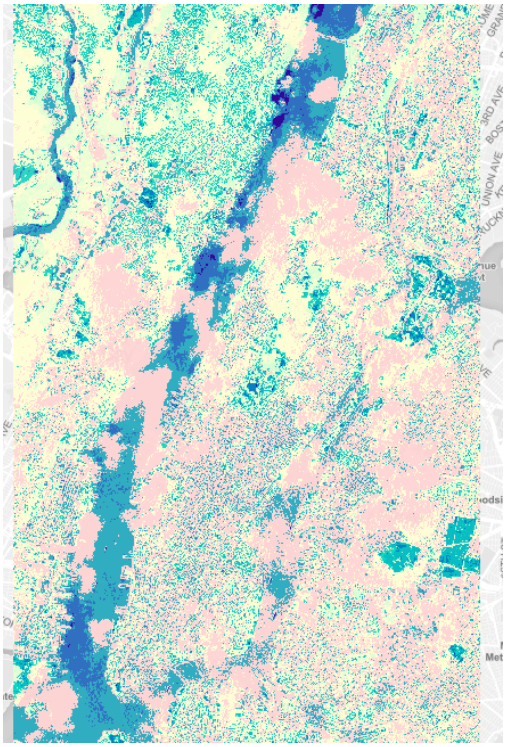

Figure 17: Indice d'humidité de Manhattan

3.4 Difficultés rencontrées

Installation

L'installation des dépendences des API SentinelHub et urbansprawl a été la source de plusieurs problèmes de système. Certaines bibliothèques nécessaires au fonctionnement d'urbansprawl (e.g. osmnx, permettant la communication avec les données OpenStreetMap) se sont révélées difficiles à installer normalement. Ces complications ont conduit à l'utilisation d'un container isolé grâce à la technologie docker. Un dockerfile avait été conçu pour l'API mais des mises à jour de certaines librairies avaient provoqué des erreurs de compatibilité. Il a donc fallu identifier les versions des librairies utilisées à l'origine et les figer à la construction. Cela fut l'occasion pour moi de m'éduquer sur le concept de container et sur la technologie docker en général.

Affichage des images

Les objets affichés originellement par urbansprawl étaient des données Geopandas. Une fois le programme capable de récupérer les images Sentinel, j'ai donc tenté de voir dans quelle mesure une image sous forme de tableau de pixels pouvait être assimilée à un objet Geopandas. Je me suis penché sur plusieurs solutions à ce problème, nottament sur le package contextily. Cependant l'installation de ce package s'est révélée incompatible avec d'autres librairies et ses fonctionnalités moins pertinentes que l'utilisation d'une couche folium.

Confinement

Le confinement national, mis en force le 17 mars, soit le lundi de ma 5 semaine de stage a beaucoup ralenti ma progression. L'obligation de passer au télétravail a dû demander des efforts de mise en place et m'obligeait à utiliser une machine moins puissante que les fixes de l'inria. Cela a porté le temps de certaines exécutions à 30 minutes et a considérablement diminué ma vitesse de travail.

4 Bilan personnel

Ce stage a été l'occasion pour moi d'exercer mes compétences dans un milieu non seulement professionnel mais aussi académique. Ce premier contact avec le monde de la recherche publique m'a permis d'apprécier le type de travail et les conditions de travail dans ce milieu. Le travail d'ajout de fonctionnalités à une plateforme créée par d'autres m'a demandé de m'approprier rapidement une API pré-existante. Plus précisément, ce stage a été une opportunité d'effectuer un véritable travail de documentation en faisant un état de l'art des méthodes de récupération d'images satellite. De plus, j'ai pu appréhender et manipuler de nombreuses technologies qui m'étaient alors étrangères : docker, folium, leaflet, geopandas, GeoJson ...

5 Conclusion

L'objectif d'ajout d'un module de visualisation de données satellites à la plateforme urbansprawl a donc été rempli en utilisant l'API SentinelHub pour la récupération et la manipulation des images, et le package folium pour leur affichage. Le module ainsi installé permet la visualisation d'images satellites de la zone voulue en même temps que les autres indices d'étalement urbain. Cet objectif a été atteint ajoutant une étape supplémentaire à l'exécution via un nouveau package sans modifier ou perturber l'API urbansprawl. Ce module permettra à urbanpsrawl d'afficher des données satellites indiquant des informations pertinentes sur l'étalement urbain.

References

[1] Luciano Gervasoni, Martì Bosch, Serge Fenet, Peter Sturm. Calculating spatial urbansprawl indices using open data. 15th International Conference on Computers in Urban Planning and Urban Management, Jul 2017, Adelaide, Australia. 2017, ¡http://www.unisa.edu.au/cupum¿. ¡hal-01535469¿

[2] https://www.indexdatabase.de/