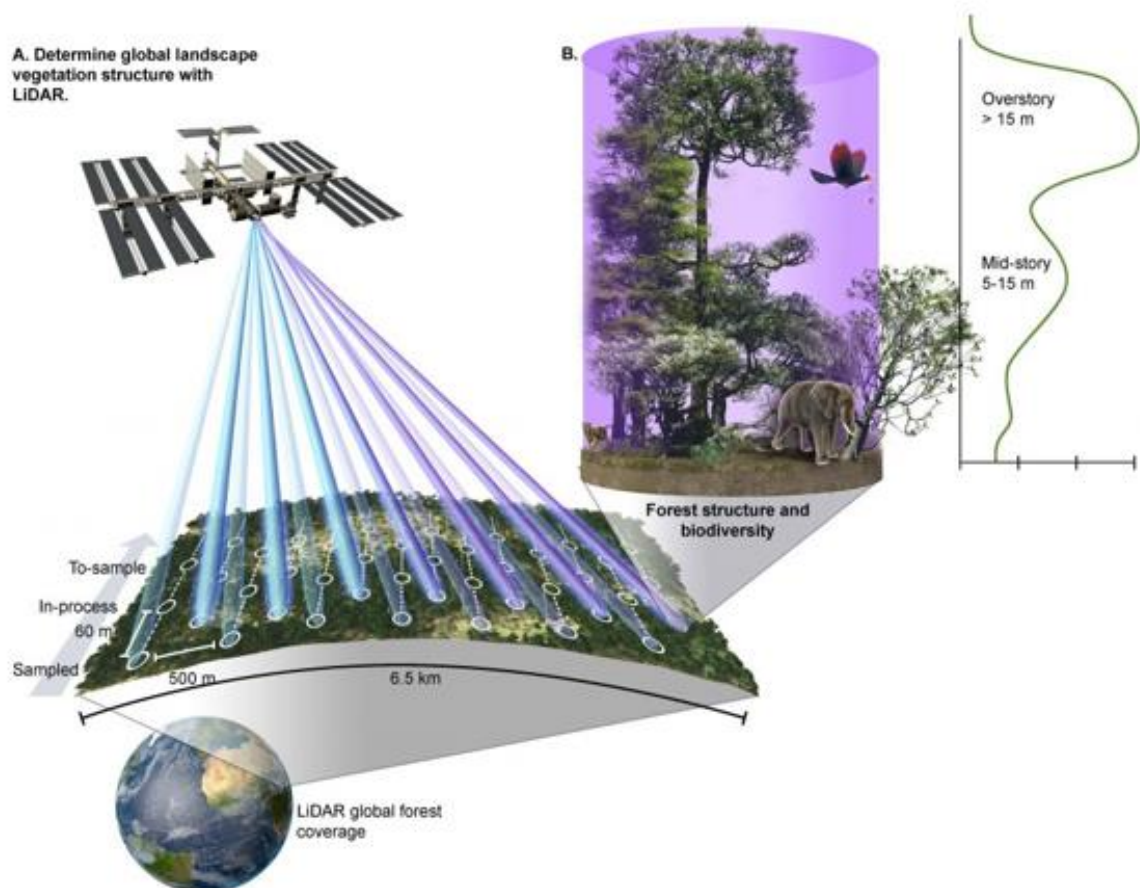


Rapport de stage Aero 4

Conception d'une plateforme de téléchargements et de visualisation de données LiDAR



Remerciements

Je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué au bon déroulement de ce stage d'Aero 4 à l'INRAE Montpellier.

Tout d'abord, je remercie chaleureusement mon maître de stage, Monsieur Nicolas Baghdadi, directeur de recherche, pour m'avoir accueilli au sein du service MathNum et pour la confiance qu'il m'a accordée tout au long de cette expérience. Sa patience, sa pédagogie, sa disponibilité et son encadrement bienveillant ont été des éléments précieux dans la réussite de ce stage.

Je souhaite également remercier l'ensemble des personnes travaillant à la Maison de la Télédétection pour leur accueil chaleureux et leur esprit de collaboration, qui m'ont permis de travailler dans un cadre idéal.

Également, je tiens à remercier Manizheh Rajabpoor et Yasser Nasrallah pour avoir pris le temps pour tester mon projet et pour leurs conseils constructifs grâce à leur esprit critique et leur disponibilité.

Un grand merci également à mon référent pédagogique, Monsieur Richard Salvetat, pour le suivi de ce stage et son encadrement académique.

Enfin, je tiens à remercier mes proches pour leur soutien constant, ainsi que toutes les personnes qui, de près ou de loin, ont contribué à l'accomplissement de cette expérience.

Fiche de synthèse

Fiche de synthèse		Joudy CHAHMI - Aéro 4
Sujet de stage	Objectifs	
Conception d'une plateforme de téléchargements et de visualisation de données LiDAR.	<ul style="list-style-type: none"> • Concevoir un logiciel exécutable sur n'importe quel PC, pouvant donner accès à l'utilisateur aux données de la NASA, les traiter, les visualiser. • Concevoir un manuel d'utilisation pour un utilisateur non formé au logiciel. 	
Client principal	Outils utilisés	
Le logiciel n'a pas pour vocation à être commercialisé. Il est mis à disposition des salariés de l'INRAE.	<ul style="list-style-type: none"> • HDFView • Excel • PowerPoint • Anaconda • Spyder • Librairies principales : folium, PyQt, hdf5, pandas 	
Etudes réalisées		
<ul style="list-style-type: none"> • Etude du LiDAR embarqué sur l'ISS (largeur faisceau, fréquence, quadrillage). • Etude de l'architecture des bases de données générées par la NASA. • Formation en ligne sur les langages CSS/HTML (OpenClassrooms). • Etude des librairies folium et PyQt. • Conception de fonctions pour accéder au téléchargement d'une zone d'étude (Google Earth Pro) et aux données LiDAR (NASA Earth Data Search) ; générer une compilation de toutes les bases de données en un fichier .csv et un fichier .h5 ; filtrer les données selon différents critères ; diviser la bases de données si trop grande ; afficher sur une carte le faisceau d'études ainsi que des informations importantes telles que sa forme d'onde et ses coordonnées géographiques. • Mise en forme du logiciel pour le rendre facile d'utilisation, ergonomique et accueillant. • Conception d'un manuel d'utilisation en anglais permettant à un utilisateur lambda de pouvoir utiliser n'importe quelle fonctionnalité du logiciel. 		
Résultats	Explications des écarts possibles	
<ul style="list-style-type: none"> • Logiciel et manuel livré à 2 semaines avant la fin du stage. • Quelques problèmes d'affichage, d'ergonomie et d'optimisation restent résoudre. 	<ul style="list-style-type: none"> • Absence de gestion de processus (threading) rend le logiciel lent car plusieurs tâches ne peuvent pas se réaliser simultanément. • Manque de temps est la cause principale du problème car il aurait fallu modifier chaque fonction. • Manque d'expériences qui a probablement impacté l'aspect ergonomique du logiciel. 	
Difficultés rencontrées	Travaux à poursuivre	
<ul style="list-style-type: none"> • Plus grande difficulté : se mettre à la place d'un utilisateur qui découvre le logiciel. • Création de l'exécutable qui fut une première pour moi (processus long). 	<ul style="list-style-type: none"> • Ajout de gestion de processus. • Ajout d'un onglet laissant le choix à l'utilisateur de choisir ses données. 	

Table des matières

Introduction	5
I. Leader de la recherche en agriculture et en agronomie durable.....	6
1.1 Historique de l’institut (fusion de deux instituts français).....	6
1.2 L’agriculture et l’agronomie en France et en Europe	7
1.3 Statut juridique	8
1.4 Engagements RSE (Responsabilité Sociétale des Entreprises)	9
1.5 Science ouverte	10
1.6 Gouvernance	11
1.7 Département MATHNUM et la Maison de la télédétection	13
II. Développement d’un logiciel de traitement de données LiDAR	14
2.1 Projet GEDI et utilisation des données	14
2.2 Cahier des charges	16
2.3 Réalisation du projet	16
2.4 Rétrospective sur les difficultés rencontrées et possibilités d’améliorations.....	21
III. Conclusion	22
IV. Table des illustrations	23
V. Annexes.....	24
VI. Bibliographie/Sitographie.....	31
VII. Abréviations	32
VIII. Glossaire	34
IX. Résumé anglais et français.....	35

Introduction

Dans le cadre de ma formation à l'IPSA en AERO 4 option Systèmes Embarqués et Télécommunications (SET), j'ai eu l'opportunité d'effectuer un stage d'assistant ingénieur à l'INRAE (Institut National de la Recherche pour l'Agriculture, l'Alimentation et l'Environnement) à Montpellier, du 17 juin 2024 au 16 septembre 2024. Ce stage avait pour objectif de concrétiser mes connaissances acquises depuis le début de mes études en mettant en pratique mes compétences tout en découvrant le monde professionnel, ce qui fut une première pour moi.

L'INRAE est un grand institut de recherche, regroupant des ingénieurs, des chercheurs et des techniciens dans différents domaines tels que l'agronomie, l'énergie etc... Pour ma part, j'ai intégré, sur le site de la Maison de la Télédétection, le département MathNum, dans un espace de coworking occupé essentiellement de stagiaires, travaillant dans différentes branches de la télédétection : données LiDAR, traitement de l'image, données métrologiques.

Ce stage m'a offert une expérience concrète et enrichissante ; j'ai pu enfin avoir un aperçu du travail réel d'un ingénieur, de la complexité que ce métier réserve mais aussi de la satisfaction du travail accompli. J'ai eu comme missions la conception d'un logiciel de traitement de données ainsi que la rédaction d'un manuel d'utilisation. J'ai pu échanger très régulièrement en anglais avec une ingénieure non francophone qui fut la « bêta-testeuse » de mon projet, ce qui m'a permis de perfectionner mon anglais. J'ai également dû me documenter et me familiariser avec le LiDAR du projet GEDI qui fut le centre de mon projet. Bien que j'aie pu acquérir de nombreuses compétences en programmation durant mon cursus, il était important pour moi de rajouter à mon bagage des connaissances supplémentaires telles que le CSS et l'HTML.

Ce rapport présentera, dans sa première partie, la structure de l'Institut en général, son mode de fonctionnement et la contribution de mon département à la recherche agronomique et environnementale. La seconde partie de ce rapport mettra en lumière les missions effectuées, les compétences que j'ai pu acquérir ou mettre en œuvre ainsi que mon retour d'expérience.

I. Leader de la recherche en agriculture et en agronomie durable

1.1 Historique de l'institut (fusion de deux instituts français)

L'INRA (Institut National de la Recherche Agronomique) est un institut français fondé en 1946, à la suite de la Seconde Guerre mondiale. À cette époque, la France se trouvait face à des défis majeurs en matière de production agricole, nécessaires pour répondre aux besoins alimentaires de la population en pleine reconstruction. L'objectif premier de la création de l'INRA était de moderniser les pratiques agricoles, d'améliorer la productivité et de promouvoir des méthodes scientifiques pour mieux gérer les ressources naturelles.

Durant ses premières années, l'institut s'est principalement concentré sur la recherche en agronomie, génétique végétale, amélioration des rendements agricoles et santé des plantes. Les enjeux économiques et sociaux liés à l'agriculture, comme l'autosuffisance alimentaire et la gestion durable des sols, figuraient également au cœur de ses préoccupations.

Au fil des années, les travaux de l'INRA ont permis des avancées significatives, notamment dans la sélection des espèces végétales et animales, mais aussi dans la mécanisation de l'agriculture et la mise en place de nouvelles techniques de culture.

Dans les années 1980, l'INRA a élargi ses domaines de recherche pour inclure l'étude des interactions entre l'agriculture et l'environnement, en réponse à une prise de conscience croissante des enjeux écologiques. L'institut a commencé à étudier l'impact des pratiques agricoles sur les écosystèmes, la biodiversité et le climat. Les chercheurs de l'INRA se sont également intéressés à l'agroécologie, en cherchant à concilier production agricole et respect de l'environnement.

Les années 1990 ont marqué une nouvelle phase de modernisation avec l'introduction des biotechnologies dans la recherche agronomique, telles que le développement des OGM (organismes génétiquement modifiés), bien que ce sujet ait suscité des débats au sein de la société française.

L'IRSTEA (Institut national de recherche en sciences et technologies pour l'environnement et l'agriculture), initialement connu sous le nom de CEMAGREF (Centre national du machinisme agricole, du génie rural, des eaux et forêts), est quant à lui fondé en 1981. Sa création visait à soutenir l'agriculture à travers la recherche appliquée, en particulier dans le domaine du machinisme agricole, et à répondre aux besoins croissants en ingénierie rurale. À cette époque, la France cherchait à moderniser son agriculture tout en développant une meilleure gestion des ressources naturelles, notamment l'eau et les forêts.

Dès ses premières années, le CEMAGREF a élargi ses activités pour inclure la gestion des écosystèmes forestiers, des ressources hydriques et la prévention des risques naturels (inondations, glissements de terrain, etc.). L'institut s'est rapidement imposé comme un acteur clé dans la recherche environnementale, notamment en ce qui concerne la gestion durable des territoires.

Les Trente Glorieuses (1945-1975) représentent une période de croissance économique rapide en France, marquée par des transformations profondes dans tous les secteurs, y compris l'agriculture.

On remarque donc que l'INRA est fondé à l'aube de cette période-là et que l'IRSTEA est quant à lui fondé après. En effet, l'INRA a pour vocation de répondre à différents enjeux dû à une forte croissance économique (trente glorieuses) et démographique (baby-boom). L'IRSTEA intervient alors pour répondre aux conséquences environnementales de cette période : écologie, gestion de l'eau et aménagement du territoire dû à une exode rurale importante.

Dans les années 1990, avec la montée des préoccupations environnementales, l'institut a mis l'accent sur les interactions entre agriculture et environnement, et a progressivement orienté ses recherches vers une approche plus globale de la gestion des ressources naturelles.

L'INRAE est donc fondé en 2020 à la suite de la fusion entre l'INRA et l'IRSTEA.

1.2 L'agriculture et l'agronomie en France et en Europe

Selon le ministère de l'économie français, en 2022, le secteur lié à l'agriculture et à l'agronomie représentait 7% du PIB et 3% d'emplois été liés à ce secteur. Cependant, on note une nette régression du poids de l'agriculture depuis la fin de la seconde guerre mondiale (Figure 1).

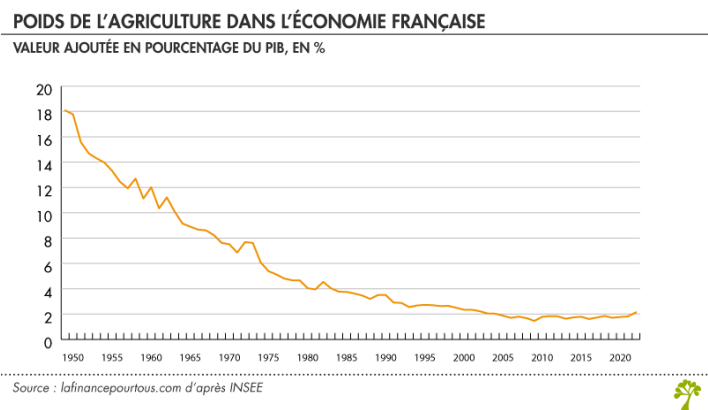


Figure 1: Poids de l'agriculture dans l'économie française (Source : INSEE)

Toutefois, la France garde la première place dans ce secteur à l'échelle européenne. Selon le ministère de l'agriculture, en 2019, avec 77,0 milliards d'euros (soit environ 18% du total de l'UE), la France affichait la plus forte production agricole totale parmi les États membres (Figure 2).



Figure 2 : Palmarès des produits agricoles et agroalimentaires français (Source : ministère de l'Agriculture et de l'Alimentation)

On note que la France figure à la première place en Europe dans de nombreux domaines et qu'elle est première exportatrice mondiale de pomme de terre, semences agricoles, malt, vins...

On peut également souligner que la France est première exportatrice d'eau minérale naturelle, richesse convoitée par de nombreux pays et qui deviendra au 21^{ème} siècle un enjeu mondial.

Aujourd'hui, la France est membre de la PAC (Politique Agricole Commune), partenariat européen visant à soutenir les agriculteurs par des subventions, des prix garantis, à stimuler la production pour répondre aux besoins alimentaires de l'Europe mais également à relever les défis environnementaux liés à l'agriculture.

1.3 Statut juridique

L'INRAE est un établissement public à caractère scientifique et technologique (EPST) français sous la tutelle conjointe du ministère de l'Enseignement supérieur et de la Recherche (MESR) et de celui chargé de l'Agriculture et de la Souveraineté alimentaire (MASA).

Selon le ministère de la fonction publique, l'INRAE a pour missions « de réaliser, d'organiser et de coordonner, sur son initiative ou à la demande de l'État, tous travaux de recherche scientifique et technologique dans les domaines de l'agriculture, de l'alimentation, de la forêt, de l'environnement, de l'eau, de la biodiversité, de la bioéconomie, de l'économie circulaire, de la gestion durable des territoires et des risques dans les champs de compétence précités ». (Décret n°2019-1946). Son décret fondateur règlemente aussi le statut des fonctionnaires de l'INRAE (Décret n°2019-1946).

Le décret fondateur énumère également 8 types d'actions :

1. Produire, publier et diffuser les connaissances scientifiques résultant de ses travaux de recherche et d'expertise ;
2. Organiser l'accès libre aux données scientifiques et aux publications (sauf dispositions ou clauses contraires) ;
3. Concourir à l'élaboration de la politique et de la stratégie nationale et européenne de recherche ;
4. Apporter son concours à l'enseignement supérieur et à la formation à la recherche et par la recherche dans ses domaines de compétence ;
5. Établir et mettre en œuvre des partenariats avec les établissements d'enseignement supérieur et de recherche ;
6. Contribuer à la conception d'innovations technologiques et sociales via la valorisation de ses compétences, de ses savoir-faire et des résultats de la recherche ;
7. Contribuer au développement de la capacité d'expertise scientifique et technologique, conduire des expertises et contribuer aux activités de normalisation pour répondre aux enjeux du développement durable en appui aux politiques publiques, aux fondations et associations reconnues d'utilité publique ;
8. Participer aux débats qui portent sur la place de la science et de la technologie dans la société.

Selon le journal The Conversation : « L'institut se positionne parmi les tout premiers organismes de recherche au monde en sciences agricoles et alimentaires, en sciences du végétal et de l'animal, et se classe 11^{ème} mondial en écologie-environnement ».

1.4 Engagements RSE (Responsabilité Sociétale des Entreprises)

L'INRAE se fixe un agenda pour 2030 pour « relever les défis de l'agriculture, de l'alimentation et de l'environnement de demain ».

Cet objectif se fonde en une stratégie en 5 orientations scientifiques et 3 orientations politiques :

- Répondre aux enjeux environnementaux et gérer les risques associés (climat, biodiversité, adaptation du vivant, risques : faire face à des défis globaux et des risques multiples et soutenir la transition écologique) ;
- Accélérer les transitions agroécologiques et alimentaires (une transformation systémique, de la production à la consommation, en intégrant les enjeux économiques et sociaux) ;
- Une bioéconomie basée sur une utilisation sobre et circulaire des ressources (l'enracinement dans les territoires) ;
- Favoriser une approche globale de la santé (connecter santé humaine, animale, végétale et environnementale) ;
- Mobiliser la science des données et les technologies du numérique au service des transitions (exploration de la complexité, intégration des échelles, transformation des systèmes, opportunités et dangers) ;
- Placer la science, l'innovation et l'expertise au cœur de nos relations avec la société pour renforcer notre culture de l'impact ;
- Être un acteur engagé dans les sites universitaires en France et un leader dans les partenariats européens et internationaux ;
- Responsabilité sociale et environnementale : une priorité collective.

La politique RSE de l'INRAE se distingue par son sens profond et son identité marquée. L'institut se positionne comme un acteur clé dans la préservation de l'environnement, tout en étant un employeur engagé. De plus, il se veut ouvert et transparent dans ses actions.

L'Association pour le Développement des Activités Sociales d'INRAE (ADAS) est une organisation dédiée à améliorer le bien-être social des employés de l'institut. Elle a pour objectif principal de proposer des services et des activités sociales pour les salariés. Elle permet de renforcer l'engagement social et humain envers ses employés en proposant un cadre de travail plus solidaire et convivial. Parmi tous ses services, l'ADAS propose des aides sociales et financières, organise des événements sportifs et culturels afin d'améliorer la cohésion entre collaborateurs ou bien organise des vacances pour les familles.

1.5 Science ouverte

Pour tous les nouveaux arrivants, qu'ils soient stagiaires ou cadres, l'INRAE met en avant des outils de collaboration tels que GIT ou Apache et promeut le principe de science ouverte. Cette approche encourage la collaboration, aussi bien à l'échelle nationale qu'internationale, grâce au partage de données collaboratives. Cela permet de renforcer l'efficacité des recherches et de favoriser l'innovation à travers une dynamique de coopération scientifique (Figure 3).



Figure 3 : Source : « science-ouverte.inrae.fr »

1.6 Gouvernance

L'INRAE est constitué de 14 départements de recherche (Figure 4) :

- ACT : Action, transitions et territoires
- AGROECOSYSTEM : Agroécosystèmes
- ALIM-H : Alimentation humaine
- AQUA : Ecosystèmes aquatiques, ressources en eau et risques
- BAP : Biologie et amélioration des plantes
- ECODIV : Ecologie et biodiversité
- ECOSOCIO : Economie et sciences sociales
- GA : Génétique animale
- **MATHNUM : Mathématiques et numérique**
- MICA : Microbiologie et chaîne alimentaire
- PHASE : Physiologie animale et systèmes d'élevage

- SA : Santé animale
- SPE : Santé des plantes et environnement
- TRANSFORM : Aliments, produits biosourcés et déchets

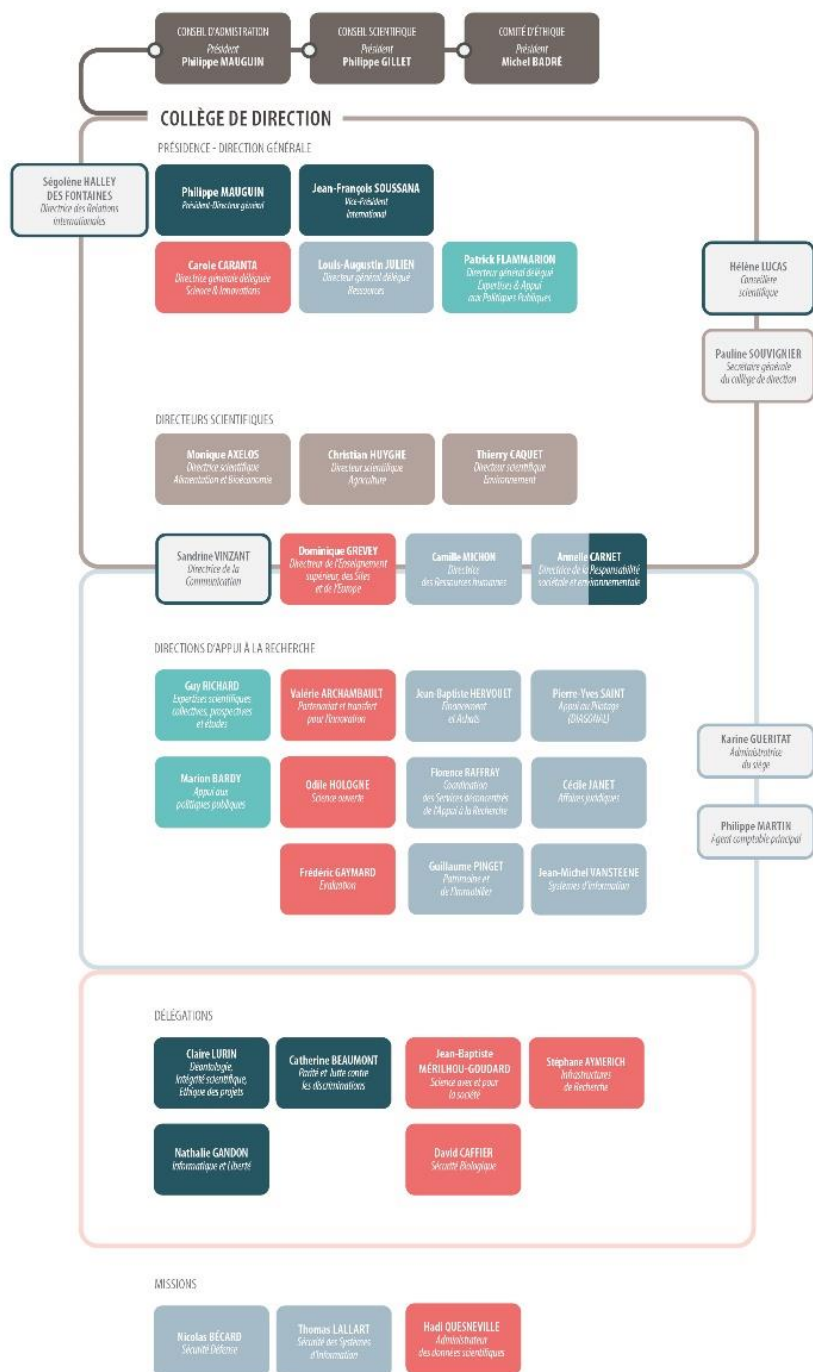


Figure 4 : Organigramme de la gouvernance de l'INRAE (Source : INRAE)

L'INRAE possède 18 centres en France avec un centre d'administration situé à Antony en région parisienne (Figure 5).

Chaque centre est constitué de différents départements et chaque centre est spécialisé dans un ou plusieurs domaines de recherche. Par exemple, le centre Occitanie-Montpellier construit son identité autour de ses recherches en agriculture, alimentation et environnement. Le centre se caractérise également par son ouverture vers les pays du sud et sa spécificité méditerranéenne.

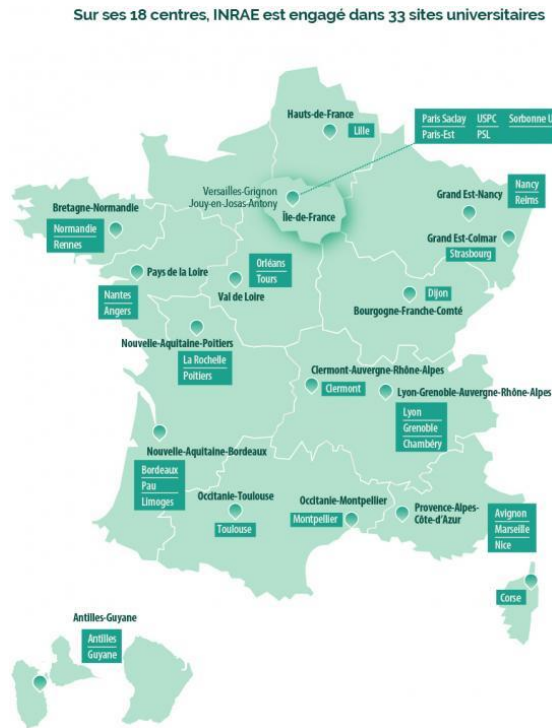


Figure 5 : Répartition des centres INRAE en France (Source : INRAE)

1.7 Département MATHNUM et la Maison de la télédétection

La Maison de la télédétection est situé sur le campus Montpellier-Agropolis. Ce bâtiment regroupe plusieurs organismes de recherche tels que CIRAD ou INRAE. Ces équipes sont dédiées à la télédétection et plus largement à l'information spatialisée. Elles sont organisées en deux unités mixtes de recherche, l'UMR TETIS et l'UMR Espace-Dev.

Le département MATHNUM d'INRAE est quant à lui situé à l'UMR TETIS.

La Maison de la télédétection est un lieu de travail collaboratif avec des salles de conférences, des salles de cours pour les étudiants d'Agro Paris Tech et des espaces de coworking.

Pour mon stage que je présenterai plus en détail par la suite, j'ai été affecté dans un espace « stagiaire » dédié à l'intégration de plusieurs étudiants stagiaires provenant de différentes écoles ou universités.

II. Développement d'un logiciel de traitement de données LiDAR

2.1 Projet GEDI et utilisation des données

GEDI est un LiDAR installé depuis 2018 sur la Station Spatiale Internationale. Il est donc de la famille des « Spaceborne LiDAR » ou LiDAR spatiaux. Cet instrument, conçu par la NASA, a pour premier objectif d'observer la structure verticale des forêts (Figure 6).

Un LiDAR est un système de télédétection qui émet plusieurs faisceaux lasers et récupère les faisceaux réfléchis depuis un objet. La mesure du temps de retour des faisceaux permet alors de cartographier l'objet cible avec précision.

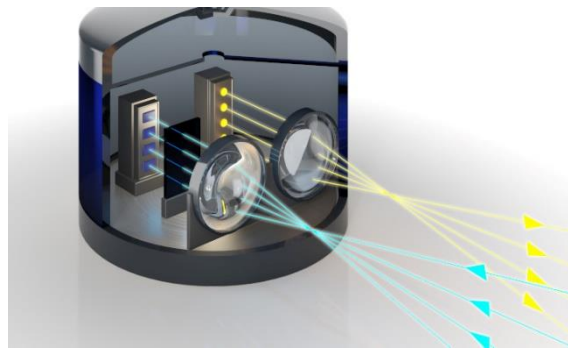


Figure 6 : Schéma du fonctionnement d'un LiDAR (Source : SCHOTT)

Il existe plusieurs types de LiDAR. GEDI est quant à lui un LiDAR dit à ondes complètes (Full-Waveform LiDAR), il capture l'intégralité des ondes réfléchies permettant d'avoir une forme d'ondes continues et non discrètes (LiDAR pulsés). Cette technique permet d'obtenir davantage d'informations sur le retour laser et une meilleure cartographie de la cible.

GEDI a une fréquence de 242 tirs lasers par seconde et une longueur d'onde de 1064 nm (infrarouge proche) adapté à l'étude de la végétation. En effet, la plupart des feuilles absorbent le rouge et le bleu mais réfléchissent le vert donc l'infrarouge proche est la plus adaptée des longueurs d'onde (Figure 7).

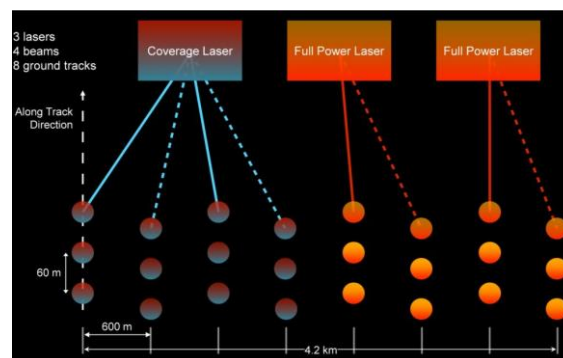


Figure 7 : Schéma de la couverture des faisceaux laser du LiDAR GEDI (Source : NASA)

Ce système utilise 8 faisceaux (Beam en anglais) créés par un diviseur de faisceau (Figure 7) : 4 faisceaux de puissances élevées et 4 de puissances faibles sont générés et permettent un balayage dense de la surface terrestre. En raison de la divergence du faisceau, on peut noter que son empreinte est d'environ 25 m de diamètre au sol et chaque faisceau est séparé de 600 m et pour un même faisceau, chaque impulsion est séparée de 60 m (Figure 7).

La donnée recueillie par le LiDAR GEDI est la forme d'onde du retour du faisceau. Cette donnée est appelée « donnée brute ». En effet, la NASA récupère cette donnée afin de l'exploiter et d'obtenir des métriques tel que l'angle d'attaque du faisceau, le nombre de pics etc...

La forme d'onde permet également de mesurer la hauteur et la densité des végétaux étudiés (Figure 8).

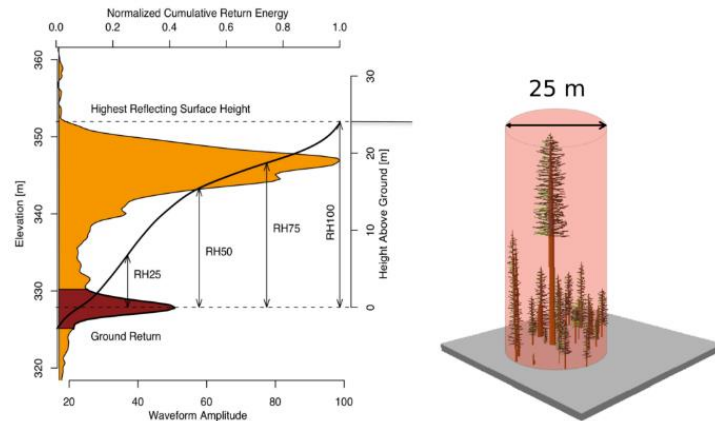


Figure 8 : Etude de la forme d'onde du LiDAR (Source : NASA)

À ces données, différents algorithmes sont ajoutés afin d'obtenir davantage de métriques. Ces algorithmes conçus par la NASA sont au nombre de 6 et permettent d'avoir des données précises sur la zone étudiée.

Les données sont accessibles sur le site NASA EarthData sous la forme de fichier HDF5. Pour obtenir ces bases de données, il faut au préalable dessiner sa zone (sur Google Earth Pro) en format .kml puis choisir quel niveau de données on veut exploiter.

Il existe 4 niveaux de données accessibles : L1A, L1B, L2A, L2B. Chaque niveau de bases de données est différent, la donnée brute est stockée dans les bases de données L1A (non localisé) et L1B (localisé).

Durant ma première semaine de stage, j'ai donc dû m'approprier les caractéristiques du LiDAR GEDI, la structure des bases de données ainsi que le processus de téléchargement de ces bases de données.

2.2 Cahier des charges

Après m'être documenté (voir bibliographie) sur le projet GEDI, j'ai dû élaborer en concertation avec mon maître de stage un cahier des charges pour mener à bien le développement du logiciel.

Malgré la grande autonomie dont j'ai pu bénéficier, j'ai dû respecter dans le cadre du logiciel un certain nombre de points :

- Pouvoir récupérer les données depuis le site NASA EarthData ;
- Pouvoir traiter les données selon différents critères (Filtrage/Séparation des algorithmes/ Division de la base de données en plus petites bases de données) ;
- Pouvoir obtenir à la fin du traitement, des données au format .csv et .h5 ;
- Pouvoir afficher sur une carte les empreintes lasers et les données associées ;
- Créer un numéro d'identification (ID) pour différencier chaque faisceau ;
- Fournir, avec le logiciel, des librairies à installer ;
- Créer un manuel d'utilisation en anglais.

J'ai dû également m'appuyer sur une base de code Python qui m'a été fournie par une collaboratrice de mon maître de stage.

A ce cahier des charges, j'ai ajouté au logiciel quelques points pertinents :

- Création d'une interface graphique afin d'éviter de former un opérateur à l'utilisation d'un terminal ;
- Création d'un exécutable autonome (compatible sur n'importe quel système d'exploitation et compatible sans avoir besoin d'installer Python ou des librairies) ;
- Utilisation de Git qui est un logiciel de gestion de versions décentralisé.

Afin de garantir un respect continu de ce cahier des charges, je faisais le point toutes les semaines sur l'avancé du logiciel.

2.3 Réalisation du projet

Initialement, le projet d'élaborer une plateforme de téléchargement de données LiDAR a été pensé pour faciliter le travail des collaborateurs de l'INRAE. En effet, ses données sont utilisées pour des calculs de prédiction, des mesures ou même pour entraîner un modèle.

Afin d'entraîner un modèle de prédictions, un grand jeu de données est nécessaire.

Il est donc indispensable d'automatiser le processus de génération et de récupérations de données.

Le traitement de données ou « data processing » fait partie de ce que l'on appelle le cycle de vie des données (Figure 9).

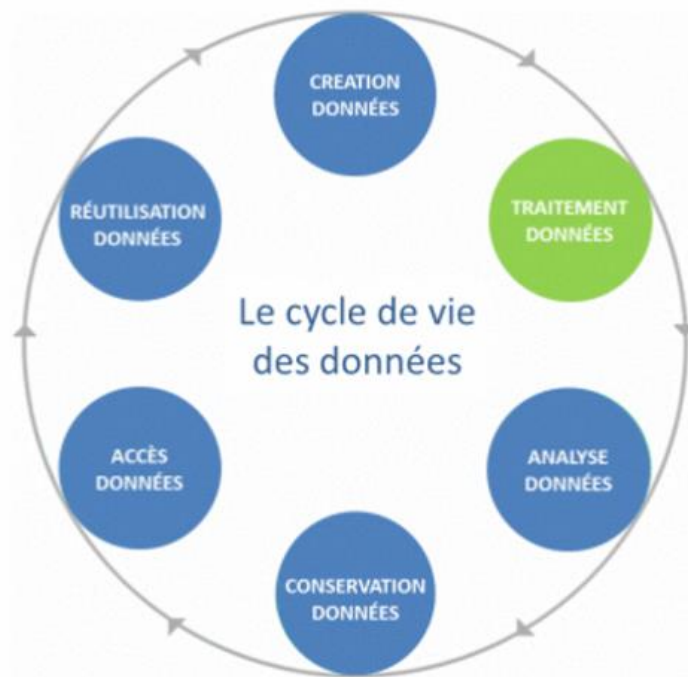


Figure 9 : Cycle de vie des données (Source : inist.fr)

Le traitement de données intervient entre la création et l'analyse. Mon rôle est donc de permettre à un utilisateur de l'INRAE d'exploiter ses données directement avec un format adapté.

Pour mener à bien la conception du logiciel, j'ai divisé mon travail en quatre parties :

- 1- La conception de l'interface graphique ;
- 2- La conception de la carte de visualisation de données ;
- 3- Le traitement des données ;
- 4- La conception du manuel d'utilisation.

Dans le domaine de la programmation, la conception des interfaces graphique est appelée « Frontend » et la conception du code fonctionnel est appelée « Backend ».

La conception conjointe de ces deux domaines de la programmation est alors appelée « Full Stack ».

Pour le Frontend, j'ai dû réaliser un code en HTML (HyperText Markup Language) et CSS (Cascading Style Sheets). J'ai donc appris à utiliser ces deux langages de programmation sur Open Classroom qui est un site internet de cours disponibles librement. HTML permet de coder la « structure » de l'interface, CSS quant à lui permet de « styliser » le contenu.

Afin d'intégrer l'interface graphique à mon code Python, j'ai utilisé le module PyQt qui permet de lier à Qt. Qt est une librairie qui permet de créer des interfaces graphiques.

Pour cela, il est nécessaire de créer une fenêtre principale avec une esthétique commune qu'on appelle « main window » et différents onglets qu'on appelle « layout ».

Pour la fenêtre principale, je définis une police particulière, un fond, la forme des onglets ... (voir annexe 1)

Chaque onglet possède son propre style et ses propres boutons. Pour certains onglets, il est nécessaire d'ajouter une entrée texte pour l'utilisateur. On définit alors une variable de type `QLineEdit()` qu'on ajoute au bouton afin d'en récupérer la valeur et l'utiliser par la suite. Ce texte peut être un nombre, une option, un chemin d'accès pour un dossier... (voir annexe 2). De plus, en choisissant le mode `QVBoxLayout()` ou `QHBoxLayout()`, on peut organiser chaque onglet en imbriquant des éléments verticalement ou horizontalement.

Avec le module PyQt, j'ai donc pu réaliser des boutons qui sont liés à des fonctions de traitement de données, des onglets, des boutons pour quitter un processus, des bars de progression etc...

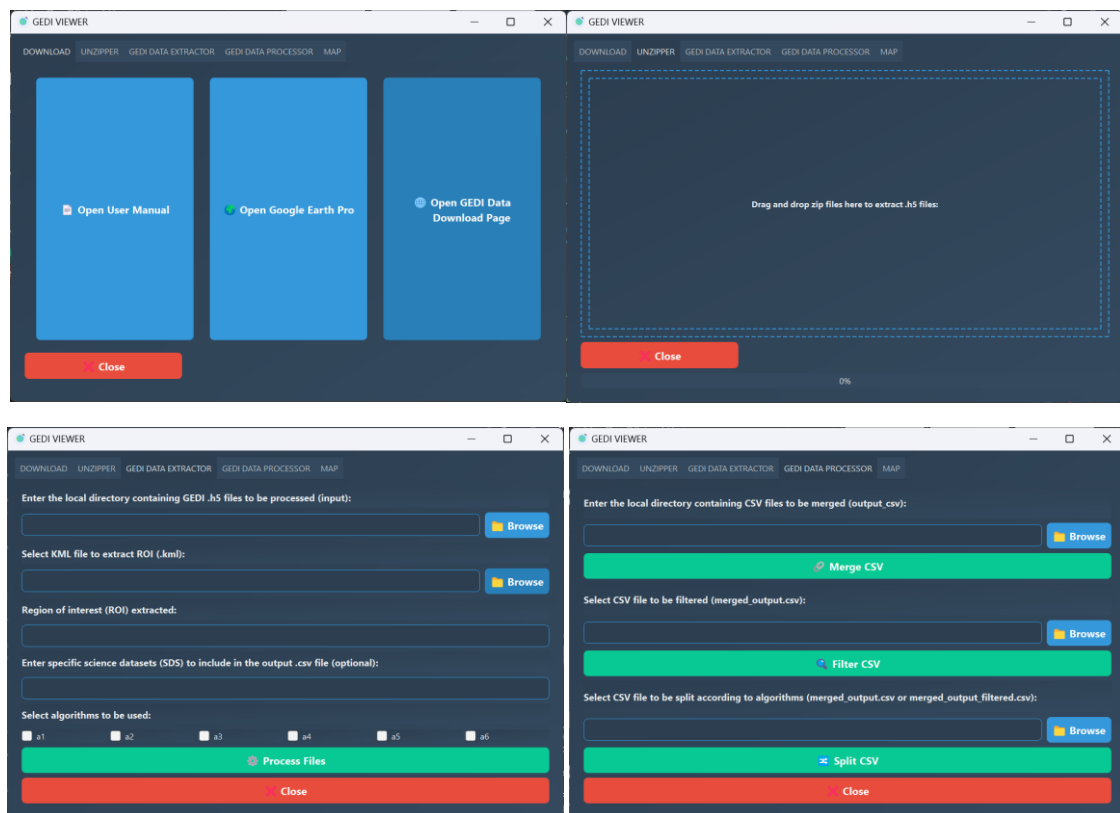


Figure 10 : Interface utilisateur

Comme on peut le voir (Figure 10), l'interface est composée de 5 onglets. Le premier onglet permet d'accéder au manuel d'utilisation, à Google Earth Pro pour sélectionner la zone d'étude et au site de téléchargement des données. Le deuxième onglet permet de glisser et déposer les fichiers de la NASA (initialement en format compressé) afin de rassembler toutes les bases de données dans un et même dossier. Le troisième et quatrième onglet sont les onglets de traitement de données. Ils permettent de créer un fichier au format .csv, un fichier au format .h5, sélectionner les algorithmes à utiliser, filtrer et séparer les données.

L'onglet glisser/déposer (Drag and Drop) permet de rassembler toutes les données compressées et de les transférer en un seul dossier. Une fonction `search_granule()` est aussi implémentée pour rechercher un fichier manquant dans le téléchargement (voir annexe 3).

Afin d'automatiser l'entrée de fichiers, j'ai implémenté différentes fonctions (voir annexe 4). Ces fonctions me permettent de passer d'un type QLineEdit() à une chaîne de caractères (str) exploitable. La définition de la région d'étude (ROI) est également automatisée, l'utilisateur a seulement besoin de rentrer son fichier .kml et la fonction `extract_roi_from_kml()` calcule directement les points aux extrémités de cette zone. Le ROI ou « Region Of Interest » permettra par la suite de vérifier si les données qu'on étudie appartiennent bien à la zone de recherche.

Différentes fonctions m'ont été fournies pour concaténer les données et les organiser en plusieurs fichiers .csv et un fichier .h5. Cependant, afin de filtrer les données par la suite, j'ai ajouté quelques données supplémentaires au fichier final .h5 telles qu'un numéro d'identification pour chaque donnée, le SNR (Signal Noise Rate), le VA (Variation Angle) (voir annexe 5).

Suite aux conseils de mon maître de stage, j'ai dû filtrer des données ne respectant pas certains critères tels que le SNR nul. Le SNR (Signal Noise Rate) est le rapport entre le signal du laser et son bruit. Si la valeur du SNR est nulle, alors on peut directement conclure que le signal n'est pas exploitable.

La formule théorique du SNR est $SNR_{dB} = 10 \cdot \log_{10} \left(\frac{P_{signal}}{P_{bruit}} \right)$

Dans notre cas, nous possédons comme données l'amplitude maximale du signal (maxamp), sa moyenne (mean) et son écart-type (stddev). Le SNR devient alors :

$$SNR_{dB} = 10 \cdot \log_{10} \left(\frac{maxamp - mean}{stddev} \right)$$

Le 5^{ème} et dernier onglet nous permet d'ouvrir la carte de visualisation des empreintes lasers (Figure 11).

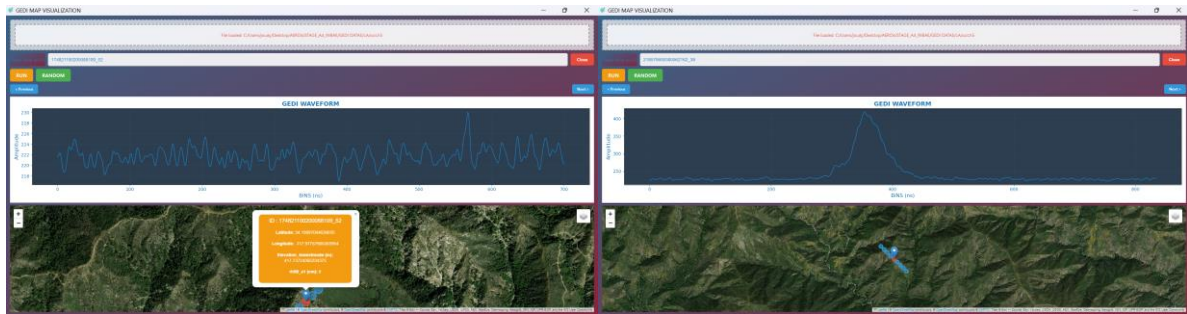


Figure 11 : Visualisation des empreintes lasers

On peut remarquer que sur le signal de gauche, le SNR est de 0 car le signal est bruité contrairement au signal de droite où l'on observe un pic. Le pic correspond à une amplitude spontanée du signal lors de la rencontre d'un obstacle. L'épaisseur de cette cloche permet alors de déterminer l'épaisseur verticale de la couche forestière à ce point. Ce panneau permet de naviguer sur les différents points de la zone d'étude, le point d'intérêt (en rouge) est entouré des points suivants et précédents (en bleu). La taille des points sur la carte n'est pas anodine ; elle correspond, en effet, au diamètre de 25 m du faisceau laser comme mentionné précédemment.

Il faut également souligner la présence d'un numéro d'identification ou « ID » dans les informations du faisceau. Ce numéro n'est pas initialement présent dans les bases de données et doit être défini par une fonction tel que le numéro commence par le numéro du tir laser « shot » suivi du numéro du faisceau de ce shot.

Afin de faciliter l'ouverture de cet onglet qui demande beaucoup de ressources, j'ai pris la décision de séparer l'ouverture de la carte avec l'application en créant un bouton « open MAP » qui permet d'ouvrir la carte sur une nouvelle fenêtre.

On a donc au total deux codes Python : le code principal APP.py contenant le corps et la grande partie des fonctions et MAP.py qui contient le code de la fenêtre de la carte.

Enfin, pour respecter le cahier des charges, j'ai dû créer un exécutable autonome avec Pyinstaller. Il s'agit d'un compilateur de code Python permettant de transformer un script Python en exécutable. Cela permet d'utiliser le script sans avoir besoin de le lancer avec un environnement de développement. De plus, Pyinstaller permet d'ajouter les librairies nécessaires au script afin d'éviter à l'utilisateur de les installer soit même sur son PC. Pour cela, il faut installer toutes les librairies nécessaires dans un environnement tel qu'Anaconda, installer pyinstaller avec la commande 'conda install pyinstaller' puis compiler tout le code avec la commande 'pyinstaller APP.py'.

À cette étape, un dossier dist, build et un fichier .spec sont installés. L'exécutable se trouve dans le dossier dist. Pour faire le lien entre l'application principale APP.py, MAP.py et les librairies associées, il est nécessaire de les spécifier dans le fichier .spec (voir annexe 6).

Une fois le fichier .spec est mis à jour, on peut recompiler en utilisant la commande 'pyinstaller APP.spec'.

Afin de rendre l'application plus accueillante, j'ai ajouté un README au dossier du logiciel pour guider l'utilisateur vers l'emplacement de l'exécutable. J'ai également ajouté une icône au format .ico pour reconnaître le logiciel.

La dernière mission de ce stage consiste à développer un manuel d'utilisation. Ce manuel qui est en format .pptx est accessible depuis le premier onglet du logiciel. Il permet donc de rendre son utilisation plus simple.

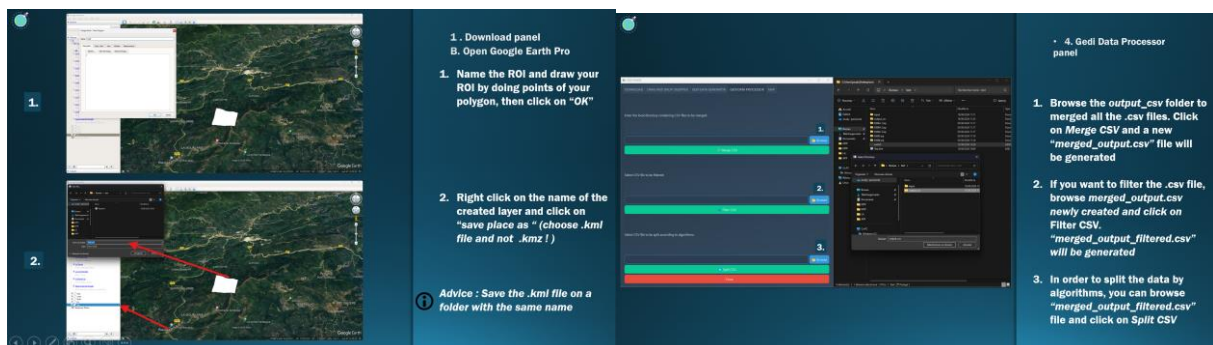


Figure 12 : Aperçu du manuel d'utilisation

Après avoir réalisé le logiciel et le manuel, deux collaborateurs de l'INRAE (Manizheh et Yasser) ont dû tester mon code du début à la fin en suivant le tutoriel du manuel.

Manizeh qui a de l'expérience avec les données GEDI a pu avoir un vrai sens critique sur les résultats finaux et m'a permis de résoudre plusieurs bugs. Yasser a, quant à lui, joué le rôle de « bêta testeur » ; grâce à son regard novice du projet, il a pu se mettre à la place d'un utilisateur lambda qui n'a aucune connaissance de ce programme. Yasser a pu finir de traiter les données en toute autonomie, ce qui nous a permis de conclure que le logiciel et son manuel étaient simples et implémentables, ce qui était le but recherché.

On peut donc dire que le cahier des charges a été respecté en temps et en heure.

2.4 Rétrospective sur les difficultés rencontrées et possibilités d'améliorations

Durant ce stage, j'ai pu rencontrer quelques difficultés à créer l'interface graphique et l'exécutable autonome. En effet, mon manque d'expériences en HTML et CSS m'a rendu la tâche difficile car il a fallu apprendre tout un nouveau langage de programmation. J'ai pu surmonter cette difficulté en consultant des cours et des aides de qualité sur internet. La deuxième difficulté a été de compiler le script Python en un exécutable. Il fallait ajouter toutes les bibliothèques avec les versions compatibles et s'il venait à en manquer une, l'exécutable ne pouvait pas se lancer.

Il a été aussi difficile de mettre régulièrement à jour le code car le processus était long. En effet, lorsque mon maître de stage ou Manizeh relevaient un bug ou demandaient à ajouter une fonctionnalité, il fallait résoudre le problème, tester le code puis re créer l'exécutable et enfin retester le logiciel final. Un seul bug revenait donc à répéter ces 4 étapes.

De mon point de vue, deux améliorations auraient pu être ajoutées à ce projet. Un système de processus (threading en anglais) aurait pu être utilisé afin d'utiliser plusieurs fonctionnalités du logiciel en même temps. Cette gestion des processus aurait pu également rendre le logiciel moins lent. La partie récupération de données depuis le site de la NASA aurait pu être automatisée par une méthode de « Web Scraping ». En traitement des données, le « Web Scraping » consiste à extraire automatiquement les données que l'on souhaite à partir d'un site internet. Un script permet donc d'accéder au site en question, et pourrait permettre de choisir directement les bases de données seulement avec une zone d'étude renseignée.

III. Conclusion

En conclusion, ce stage a été une étape cruciale dans mon parcours académique et professionnel. Il m'a permis d'acquérir de solides compétences techniques, notamment en Python et dans d'autres langages de programmation, tout en me familiarisant avec les enjeux complexes du projet GEDI. De plus, il a été une opportunité d'améliorer mes compétences linguistiques en anglais, grâce aux interactions avec des stagiaires internationaux.

Sur le plan personnel, j'ai découvert la richesse du travail en équipe et l'importance de la collaboration dans le domaine de l'ingénierie. J'ai également pris conscience des exigences élevées de rigueur et de précision qu'impose ce métier, ainsi que des délais serrés auxquels il faut souvent faire face. Même si j'aurais souhaité avoir plus de temps pour perfectionner certains aspects du projet, cette expérience m'a permis de mieux comprendre les défis pratiques que rencontrent les ingénieurs au quotidien.

Enfin, ce stage a renforcé ma motivation à poursuivre dans cette voie et à m'améliorer constamment. Il m'a ouvert des perspectives intéressantes, notamment en matière d'intégration de technologies avancées telles que l'intelligence artificielle et l'automatisation, qui pourront à l'avenir optimiser les processus et la gestion des données. Je suis convaincu que les compétences et les enseignements tirés de cette expérience seront des atouts précieux pour ma carrière future, et je me réjouis de pouvoir les mettre en pratique dans mes prochains projets professionnels.

IV. Table des illustrations

Figure 1 : Poids de l'agriculture dans l'économie française (Source : INSEE).....	7
Figure 2 : Palmarès des produits agricoles et agroalimentaires français (Source : ministère de l'Agriculture et de l'Alimentation)	8
Figure 3 : Source : « science-ouverte.inrae.fr ».....	11
Figure 4 : Organigramme de la gouvernance de l'INRAE (Source : INRAE)	12
Figure 5 : Répartition des centres INRAE en France (Source : INRAE).....	13
Figure 6 : Schéma du fonctionnement d'un LiDAR (Source : SCHOTT)	14
Figure 7 : Schéma de la couverture des faisceaux laser du LiDAR GEDI (Source : NASA)	14
Figure 8 : Etude de la forme d'onde du LiDAR (Source : NASA)	15
Figure 9 : Cycle de vie des données (Source : inist.fr)	17
Figure 10 : Interface utilisateur	18
Figure 11 : Visualisation des empreintes lasers.....	19
Figure 12 : Aperçu du manuel d'utilisation	20

V. Annexes

Table des annexes

ANNEXE 1	Implémentation de la fenêtre principale
ANNEXE 2	Exemple d'implémentation d'un onglet
ANNEXE 3	Implémentation du glisser/déposer et de la recherche de données manquantes
ANNEXE 4	Implémentation de fonctions d'entrées de fichiers
ANNEXE 5	Implémentation de numéro d'identification, du SNR et du VA
ANNEXE 6	Exemple de fichier .spec

Annexe 1

Implémentation de la fenêtre principale

```

main_window.setStyleSheet("""
QWidget {
    background: qlineargradient(x1: 0, y1: 0, x2: 1, y2: 1,
        stop: 0 #2c3e50, stop: 1 #34495e);
    color: #ecf0f1;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
QProgressBar {
    border: none;
    border-radius: 8px;
    background-color: #34495e;
    text-align: center;
    color: #ecf0f1;
}
QProgressBar::chunk {
    background-color: #3498db;
    width: 8px;
    margin: 0.5px;
}
QLabel {
    font-size: 18px;
    color: #ecf0f1;
    font-weight: bold;
}
QLineEdit {
    padding: 8px;
    font-size: 16px;
    border-radius: 8px;
    border: 1px solid #3498db;
    background-color: #2c3e50;
    color: #ecf0f1;
}
QTabWidget::pane {
    border: none;
}
QTabBar::tab {
    background-color: #34495e;
    color: #bdc3c7;
    padding: 10px;
    border: none;
    border-bottom: 2px solid #2c3e50;
}
QTabBar::tab:selected {
    background-color: #2c3e50;
    color: #ecf0f1;
}
QTabBar::tab:hover {
    background-color: #3b4b5b;
}
""")

```

Annexe 2

Exemple d'implémentation d'un onglet

```
# Tab for GEDI Data Processor
gedi_tab = QWidget()
gedi_layout = QVBoxLayout()
gedi_tab.setLayout(gedi_layout)

outdirLabel = QLabel("Enter the local directory containing CSV files to be merged (output_csv):")
outdirLayout = QHBoxLayout() # Create a new layout for the output directory
outdirLineEdit = QLineEdit()
browseoutButton = QPushButton("📂 Browse")
browseoutButton.clicked.connect(lambda: browse_directory(outdirLineEdit))
browseoutButton.setStyleSheet("""
    QPushButton {
        background: #3498db;
        color: white;
        border-radius: 8px;
        padding: 10px;
        font-size: 20px;
        border: none;
        font-weight: bold;
    }
    QPushButton:hover {
        background: #2980b9;
    }
    QPushButton:pressed {
        background: #1f618d;
    }
""")
outdirLayout.addWidget(outdirLineEdit)
outdirLayout.addWidget(browseoutButton)

mergeButton = QPushButton("🔗 Merge CSV")
mergeButton.clicked.connect(lambda: merge_csv_on_id(outdirLineEdit.text()))
mergeButton.setStyleSheet("""
    QPushButton {
        background: #08c993;
        color: white;
        border-radius: 8px;
        padding: 10px;
        font-size: 20px;
        border: none;
        font-weight: bold;
    }
    QPushButton:hover {
        background: #0aa378;
    }
    QPushButton:pressed {
        background: #0a9f6a;
    }
""")
```

Annexe 3

Implémentation du glisser/déposer et de la recherche de données manquantes

```
def search_granule(file_path):
    # Directory where .h5 files are stored (change this to the appropriate directory)
    h5_directory = file_path
    expected = []
    aliste = []
    part1 = ['GEDI01', 'GEDI02']
    part2 = ['A', 'B']
    part3 = []
    # List all .h5 files in the directory
    for file_name in os.listdir(h5_directory):
        if file_name.endswith(".h5"):
            parts = file_name.split('_')

            if parts[3] not in part3:
                part3.append(parts[3])

    part3 = sorted(part3)
    for i in range(len(part1)):
        for j in range(len(part2)):
            for k in range(len(part3)):

                l = part1[i] + '_' + part2[j] + '_' + part3[k]
                if part1[i] + '_' + part2[j] != 'GEDI01_A':
                    if l not in expected:
                        expected.append(l)

    for file_name in os.listdir(h5_directory):
        if file_name.endswith(".h5"):
            parts = file_name.split('_')
            actual = parts[1] + '_' + parts[2] + '_' + parts[3]
            aliste.append(actual)

    loose = []
    compt = 0
    for i in range(len(aliste)):
        if expected[i] != aliste[i]:
            compt += 1
            aliste.insert(i, " ")
            print("expected :", expected[i])
            loose.append(expected[i])
    if expected[-1] != aliste[-1]:
        aliste.append(' ')
        compt += 1
        print("expected :", expected[-1])
        loose.append(expected[-1])
    loose = ' / '.join(f"{element}" for element in loose)
    if compt != 0:
        QMessageBox.information(None, "Process Completed", f"The following files are missing {loose}")
    else:
        QMessageBox.information(None, "Process Completed", "0 missing files")

# Drag and drop events
def drag_enter_event(event):
    if event.mimeData().hasUrls():
        event.acceptProposedAction()

def drop_event(event, label):
    urls = event.mimeData().urls()
```

Annexe 4

Implémentation de fonctions d'entrées de fichiers

```

### PROCESSING
def browse_zip():
    dir_name = QFileDialog.getExistingDirectory(None, "Select Directory")

def browse_directory(inDirLineEdit):
    dir_name = QFileDialog.getExistingDirectory(None, "Select Directory")
    if dir_name:
        inDirLineEdit.setText(dir_name)

def browse_kml(kmlLineEdit):
    file_name, _ = QFileDialog.getOpenFileName(None, "Select KML File", "", "KML files (*.kml)")
    if file_name:
        kmlLineEdit.setText(file_name)
        extract_roi_from_kml(file_name, line_edit_refs['roi'])

def browse_csv(csvLineEdit):
    file_name, _ = QFileDialog.getOpenFileName(None, "Select CSV File", "", "CSV files (*.csv)")
    if file_name:
        csvLineEdit.setText(file_name)

def extract_roi_from_kml(kml_file, roiLineEdit):
    with open(kml_file, 'rb') as f: # Open the file in binary mode
        k = kml.KML()
        k.from_string(f.read())

        features = list(k.features())
        if not features:
            QMessageBox.critical(None, "Error", "No features found in the KML file.")
            return

        # Assuming the first feature's geometry is the ROI
        feature = features[0]
        if hasattr(feature, 'geometry'):
            geom = feature.geometry
        elif hasattr(feature, 'features'):
            geom = list(feature.features())[0].geometry
        else:
            QMessageBox.critical(None, "Error", "No geometry found in the KML feature.")
            return

        bounds = geom.bounds
        ul_lat, ul_lon = bounds[3], bounds[0]
        lr_lat, lr_lon = bounds[1], bounds[2]
        global roi
        roi = f"{ul_lat},{ul_lon},{lr_lat},{lr_lon}"
        global ROI
        ROI = Polygon([(ul_lon, ul_lat), (lr_lon, ul_lat), (lr_lon, lr_lat), (ul_lon, lr_lat)])
        roiLineEdit.setText(roi)

```

Annexe 5

Implémentation de numéro d'identification, du SNR et du VA

```
def create_ids(shot_numbers):
    indices = np.arange(len(shot_numbers))
    ids = [f"{shot}_{index}" for shot, index in zip(shot_numbers, indices)]
    return np.array(ids)

def calculate_SNR(rx_modeamps, mean, stddev):

    maxamp = rx_modeamps.max(axis=1)
    SNR = [10*math.log10((x-y)/z) if (z>0 and x>y) else 0 for x,y,z in zip(maxamp, mean, stddev)]
    return SNR

### Calculate VA
def haversine(lon1, lat1, lon2, lat2):
    R = 6371000 # radius of Earth in meters
    phi_1 = math.radians(lat1)
    phi_2 = math.radians(lat2)

    delta_phi = math.radians(lat2 - lat1)
    delta_lambda = math.radians(lon2 - lon1)

    a = math.sin(delta_phi / 2.0) ** 2 + \
        math.cos(phi_1) * math.cos(phi_2) * \
        math.sin(delta_lambda / 2.0) ** 2
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))

    return R * c # output distance in meters

def VA(lon1, lat1, lon2, lat2, altitude):
    distance = haversine(lon1, lat1, lon2, lat2)
    return np.rad2deg(math.atan(distance / altitude))
```

Annexe 6

Exemple de fichier .spec

```
block_cipher = None

a = Analysis(
    ['APP.py'],
    pathex=['C:\\Users\\joudy\\Desktop\\python_joud'], # Ajouter le chemin où se trouvent vos fichiers Python
    binaries=[
        ('C:\\Users\\joudy\\anaconda3\\envs\\joud\\Library\\bin\\hdf5.dll', '.'),
        # Ajoutez d'autres DLLs nécessaires ici si nécessaire
    ],
    datas=[
        ('C:\\Users\\joudy\\anaconda3\\envs\\joud\\Lib\\site-packages\\xyzservices\\data\\providers.json', 'xyzservices/data'), ('MAP.py', '.'),
    ],
    hiddenimports=['pandas', 'shapely.geometry', 'geopandas', 'numpy', 'PyQt5.QtWidgets', 'PyQt5.QtCore', 'PyQt5.QtGui', 'fastkml', 'subprocess', 'PyQt5.QtWebEngineWidgets', 'matplotlib', 'folium', 'PyQt5.sip', 'PyQt5.QtWebEngineWidgets', 'h5py', 'requests', 'chardet', 'charset_normalizer']
    hookspath=[],
    hooksconfig={},
    runtime_hooks=[],
    excludes=[],
    win_no_prefer_redirects=False,
    win_private_assemblies=False,
    cipher=block_cipher,
    noarchive=False,
)

pyz = PYZ(a.pure, a.zipped_data, cipher=block_cipher)

exe = EXE(
    pyz,
    a.scripts,
    [],
    exclude_binaries=True,
    name='APP',
    debug=False,
    bootloader_ignore_signals=False,
    strip=False,
    upx=False,
    console=False,
    disable_windowed_traceback=False,
    argv_emulation=False,
    target_arch=None,
    codesign_identity=None,
    icon='icon.ico',
    entitlements_file=None,
)

coll = COLLECT(
    exe,
```

VI. Bibliographie/Sitographie

- openclassrooms.com
- search.earthdata.nasa.gov/search
- gedi.umd.edu
- inrae.fr
- schott.com
- science-ouverte.inrae.fr
- inist.fr
- agriculture.gouv.fr

VII. Abréviations

ACT : Action Transitions et Territoires

ADAS : Association pour le Développement des Activités Sociales

ALIM-H : Alimentation Humaine

APP : Application

BAP : Biologie et Amélioration des Plantes

CEMAGREF : CEntre national du Machinisme Agricole, du Génie Rural, des Eaux et Forêts

CSS : Cascading Style Sheets

CSV : Comma-Separated Values

EPST : Etablissement Public à caractère Scientifique et Technologique

GA : Génétique Animale

GEDI : Global Ecosystem Dynamics Investigation

HDF5 : Hierarchical Data Format version 5

HTML : HyperText Markup Language

INRA : Institut National de la Recherche Agronomique

INRAE : Institut National de la Recherche pour l'Agriculture, l'Alimentation et l'Environnement

IPSA : Institut Polytechnique des Sciences Avancées

IRSTEA : Institut national de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture

ISS : International Space Station

LiDAR: Light Detection and Ranging

MAP : Carte (fichier de la carte dans l'application)

MASA : Ministère de l'Agriculture et de la Souveraineté Alimentaire

MESR : Ministère de l'Enseignement Supérieur et de la Recherche

NASA : National Aeronautics and Space Administration

PAC : Politique Agricole Commune

PIB : Produit Intérieur Brut

PyQt : Python binding of the cross-platform application framework Qt

RSE : Responsabilité Sociétale des Entreprises

SNR : Signal-to-Noise Ratio

SPE : Santé des Plantes et Environnement

TRANSFORM : Aliments Produits Biosourcés et Déchets

VIII. Glossaire

CSS : Langage utilisé pour styliser le contenu d'une page web

CSV : Format de fichier permettant de stocker des données tabulaires, où chaque ligne représente un enregistrement

HDF5 : Format de fichier conçu pour stocker de grandes quantités de données, souvent utilisé dans les données scientifiques

LiDAR : Technologie de télédétection utilisant des lasers pour mesurer la distance à une surface et cartographier des objets en 3D

NASA EarthData : Plateforme de la NASA qui fournit des données d'observation de la Terre

OpenClassrooms : Plateforme de cours en ligne permettant l'apprentissage de diverses compétences, y compris des langages de programmation

PyQt : Bibliothèque Python permettant de créer des interfaces graphiques

Python : Langage de programmation polyvalent, souvent utilisé pour le traitement des données

ROI (Region of Interest) : Région d'intérêt, une zone spécifique définie pour l'analyse de données

SNR (Signal-to-Noise Ratio) : Rapport entre la puissance du signal et celle du bruit, utilisé pour déterminer la qualité d'un signal

Threading : Technique de gestion des processus permettant d'exécuter plusieurs tâches simultanément.

IX. Résumé anglais et français

Ce rapport présente le travail réalisé lors d'un stage d'ingénieur assistant à l'INRAE Montpellier, portant sur la conception d'une plateforme de téléchargement et de visualisation de données **LiDAR** en **Python**. L'objectif était de permettre aux utilisateurs d'accéder aux données **NASA**, de les traiter et de les visualiser de manière ergonomique. Le projet a nécessité l'étude du LiDAR spatial **GEDI**, l'utilisation de bibliothèques Python, et le développement d'une interface graphique conviviale avec **PyQt**. Des fonctions ont été créées pour télécharger des zones d'étude, traiter les **données** et les afficher sur une carte interactive. Bien que des difficultés liées à la gestion des tâches simultanées et à l'optimisation de l'ergonomie aient été rencontrées, le logiciel a été livré avec succès, accompagné d'un manuel d'utilisation. Les améliorations futures pourraient inclure une gestion plus fluide des processus et une meilleure prise en charge des grands ensembles de données.

This report details the work carried out during an assistant engineer internship at INRAE Montpellier, focused on the development of a platform for downloading and visualizing **LiDAR** data in **Python**. The objective was to provide users with access to NASA data, enabling them to process and visualize it in a user-friendly manner. The project involved studying the spaceborne LiDAR **GEDI**, using Python libraries, and developing a graphical interface with **PyQt**. Functions were implemented to download study areas, process the **data**, and display it on an interactive map. Despite challenges related to task management and ergonomic optimization, the software was successfully delivered along with a user manual. Future improvements could include better process management and enhanced handling of large datasets.