

Léa MOING



## Notice d'utilisation d'un indice de recouvrement des herbiers de *Zostera noltei*



Encadrée par Natacha Volto (LIENSs – La Rochelle) et Pamela Lagrange (LPO – Rochefort) – mai à juin 2024

## Table des matières

<i>Table des matières</i> .....	2
<i>Table des tableaux</i> .....	3
<i>Table des figures</i> .....	3
INTRODUCTION .....	4
1. DONNEES ET MATERIELS .....	6
A. <i>Sites d'études</i> .....	6
B. <i>Données</i> .....	6
2. METHODOLOGIE .....	10
A. <i>Indice de recouvrement</i> .....	10
B. <i>Conception de l'algorithme de l'indice de recouvrement</i> .....	12
C. <i>Estimation des échantillons</i> .....	13
3. UTILISATION DE L'EXTENSION INDICESURFACIQUEZOSTERES SUR QGIS.....	14
A. <i>Manipulations préalables avant utilisation de l'extension</i> .....	14
B. <i>Présentation et explication de l'interface</i> .....	16
4. RESULTATS.....	17
A. <i>Estimation des échantillons</i> .....	17
B. <i>Résultats de l'indice de recouvrement calculé à partir de l'extension</i> .....	21
BIBLIOGRAPHIE .....	22
ANNEXE 1 : SCRIPT PYTHON PERMETTANT LE TRAITEMENT DE L'INDICE DE RECOUVREMENT DES HERBIERS DE <i>ZOSTERA</i> <i>NOLTEI</i> .....	25
ANNEXE 2 : SCRIPTS EXTENSION QGIS .....	30
ANNEXE 3 : SCRIPT R PERMETTANT DE REALISER LE CALCUL DE L'ESTIMATION DES ECHANTILLONS A FAIRE .....	40

## Table des tableaux

<a href="#">TABLEAU 1 : OBSERVATIONS REALISEES SUR LES DIX ZONES ENTRE 2015 ET 2023</a> .....	6
<a href="#">TABLEAU 2 : DATES OPTIMALES (A MAREE BASSE ET COUVERTURE NUAGEUSE BASSE) POUR TELECHARGER LES IMAGES SENTINEL-2A SUR L'ILE DE RE ENTRE 2015 ET 2024</a> .....	7
<a href="#">TABLEAU 3 : DATES OPTIMALES (A MAREE BASSE ET COUVERTURE NUAGEUSE BASSE) POUR TELECHARGER LES IMAGES SENTINEL-2A SUR L'ILE D'OLERON ENTRE 2015 ET 2024</a> .....	7
<a href="#">TABLEAU 4 : EXEMPLE D'UN TABLEUR VIERGE AVEC COLONNES TYPES</a> .....	13
<a href="#">TABLEAU 5 : ESTIMATION DE L'ECHANTILLONNAGE EN FONCTION DES SURFACES DU ZONAGE DES HERBIERS SELON DIFFERENTES METHODES</a> .....	18

## Table des figures

<a href="#">FIGURE 1 : CARTE DE LOCALISATION DE ZOSTERES NAINES</a> .....	5
<a href="#">FIGURE 2 : SCHEMA DES TYPES DE FRAGMENTATION DU PAYSAGE OBSERVE SUR LE TERRAIN</a> .....	6
<a href="#">FIGURE 3 : NDVI MENSUEL POUR LES MOIS DE MARS AU MOIS D'AOUT DE 2018 A 2023</a> .....	8
<a href="#">FIGURE 4 : GRAPHIQUE DE L'INDICE DE RECOUVREMENT SELON ZOFFOLI ET AL. 2020</a> .....	9
<a href="#">FIGURE 5 : GRAPHIQUE DE L'INDICE DE RECOUVREMENT SELON DALLOYAU ET AL. 2023</a> .....	9
<a href="#">FIGURE 6 : GRAPHIQUE DE L'INDICE DE RECOUVREMENT D'APRES DALLOYAU ET AL. 2023 ET AJUSTEE A NOS DONNEES</a> .....	10
<a href="#">FIGURE 7 : TABLEUR PERMETTANT LA MISE A JOUR DE L'EQUATION</a> .....	10
<a href="#">FIGURE 8 : INTERFACE DE L'EXTENSION INDICESURFACIQUEZOSTERES SUR QGIS</a> .....	13
<a href="#">FIGURE 9 : NDVI D'OLERON-VIADUC</a> .....	15
<a href="#">FIGURE 10 : HISTOGRAMME DU NDVI</a> .....	15
<a href="#">FIGURE 11: K-MEANS</a> .....	16
<a href="#">FIGURE 12 : PLAN DE PRELEVEMENTS</a> .....	16
<a href="#">FIGURE 13 : ECDF</a> .....	17
<a href="#">FIGURE 14 : REPARTITION DE L'INDICE DE RECOUVREMENT DES HERBIERS DE ZOSTERES EN 2018</a> .....	19



## Introduction

Les travaux de S. Dalloyau (2008) montrent que, de 1980 à 1996, les pertuis charentais constituaient le premier site d'hivernage français pour les bernaches cravants. Il existe une relation de corrélation positive entre le nombre de bernaches cravants sur un secteur et la superficie de l'herbier : les oiseaux adaptent leur comportement alimentaire, et leur temps de stationnement, selon la disponibilité des ressources (Jacobs et al. 1981; Tubbs et Tubbs 1982; Percival et EVANS 2008; Ganter 2000). Les herbiers de Zostère naine jouent un rôle crucial dans l'écosystème marin en fixant le sédiment, en fournissant un habitat pour la biodiversité marine, en servant de nurserie pour l'ichtyofaune et en étant une ressource alimentaire pour de nombreux oiseaux herbivores comme les Bernaches cravants et divers canards. La Mer des Pertuis est notamment un site d'importance internationale pour le Canard pilet et la Bernache cravant. Ces herbiers sont donc de bons indicateurs de l'état de l'écosystème.

Les zostères sont sensibles aux perturbations d'origine naturelle ainsi qu'aux perturbations humaines (Short et Wyllie-Echeverria 1996; Cunha et al. 2005; Cabaço et al. 2008; Guimarães et al. 2012). Ces dernières années, l'augmentation des niveaux de perturbations anthropiques, telles que la pollution des eaux ou la destruction d'habitat (piétinement, creusage, raclage), a augmenté la fragmentation des herbiers et leur déclin à travers le monde (Short et Wyllie-Echeverria 1996; Duarte 2000; Cunha et al. 2005; Orth et al. 2006; Guimarães et al. 2012). Il est également attendu que l'augmentation des températures ou encore de l'acidification des océans liés aux modifications climatiques impactent profondément les herbiers (Scalpone et al. 2020; Franzitta et al. 2021).

La réduction de la surface et la diminution de la densité des herbiers impliquent une perte de leurs fonctionnalités écologiques (support de biodiversité pour la faune et la flore, reproduction et nurserie, alimentation) et biogéochimiques (réduction de la houle, sédimentation, photosynthèse, stockage de carbone, azote, phosphore). Afin de préserver ces habitats, les herbiers sont aujourd'hui considérés comme des « espèces et habitats menacés et/ou en déclin » par la convention OSPAR étendue en 2008. Les zostères sont aussi protégées au niveau européen par la Directive Habitat 92/43/CEE ainsi que par la Directive Cadre sur l'Eau 2000/60/CE. En France, la loi n°86-2 du 3 Janvier 1986 relative à l'aménagement, la protection et la mise en valeur du littoral attribue aux herbiers de zostères un statut de protection en tant qu'habitats prioritaires.

Les herbiers de zostère naine des pertuis Charentais sont suivis par les gestionnaires des réserves naturelles nationales (RNN) gérées par la LPO, soit la RNN de Lileau des Niges, des Marais d'Yves et de Moëze Oléron depuis de nombreuses années (Dalloyau 2015; Dalloyau et Robin 2013; Dalloyau 2020; Zoffoli et al. 2023). Historiquement, le suivi s'appuyait sur un échantillonnage aléatoire par quadrillage (une station tous les 100 mètres) où le recouvrement et la fragmentation de l'herbier étaient relevés sur le terrain puis modélisés et reportés au périmètre connu de l'herbier, lui-même identifié de façon exhaustive par dGPS en 2012 et 2015 (Dalloyau et Robin 2013; Dalloyau 2015). Le déploiement et la disponibilité de l'imagerie

satellites ces dernières années a permis d'affiner cette approche par la démocratisation de nouveaux outils d'évaluation. Depuis 2019, la méthodologie permettant l'évaluation des surfaces d'herbiers a évolué afin de prédire les taux de recouvrements pour *Zostera noltei* grâce aux données satellites en libre accès. Les images satellitaires utilisées (de type Sentinel-2A) présentent une résolution spatiale de 10 mètres permettant de mieux évaluer la répartition des herbiers ainsi que leur évolution intra et interannuelle. L'acquisition d'informations à une telle résolution spatiale favorise la compréhension des mécanismes fonctionnels proie/prédateur (avec les communautés d'oiseaux d'eau fréquentant les espaces protégés et leurs périphéries), de l'évolution géomorphologique des secteurs d'herbiers (e.g. ensablement, travaux...), et des interactions anthropiques.

Ce projet s'inscrit dans la continuité du projet CoEHCO (Connaissance et Évaluation de l'état des Habitats benthiques Côtiers), porté par le Parc Naturel Marin de l'Estuaire de la Gironde et de la Mer des Pertuis ainsi que la Ligue pour la Protection des Oiseaux (LPO). Le volet "herbiers de zostères" de ce projet a évalué l'état des herbiers de Zostère naine (*Zostera noltei*), des habitats marins importants et sensibles aux perturbations anthropiques, considérés comme menacés par la convention OSPAR de 2008, la DHFF et la DCE. L'état des herbiers a été réalisé par deux approches : l'une utilisant le suivi stationnel DCE portant sur un suivi qualitatif des herbiers, l'autre par un suivi surfacique initié par la LPO. C'est sur ce dernier que le développement de méthode va porter. Le but de ce stage est de mettre en place une méthode de travail reproductible de l'estimation des surfaces d'herbiers de zostères naines à partir d'images satellites Sentinel-2 Level-2A afin de calculer annuellement un indice de recouvrement. Cette méthode peut également être utilisée pour des données de drone multispectrales dont la résolution est plus fine. Il permettra sur le long terme de voir l'évolution de la surface des herbiers en lien avec les pressions annuelles collectées sur les sites.

Le laboratoire LIENSs, avec son expertise en habitats marins et géomatique, contribuera à améliorer et rendre reproductible la méthode actuelle de traitement des images satellites. Dans le cadre de ma première année de Master Télédétection et Environnement réalisé à l'Université Rennes 2, j'ai réalisé un stage facultatif de deux mois au laboratoire du LIENSs (Littoral Environnement et Sociétés) à la Rochelle.



Tableau 1 : Observations réalisées sur les dix zones entre 2015 et 2023

	Fiers d'Ars	Fosse de Loix	La Flotte	Rivedoux	Baie d'Yves	Oléron-Viaduc	Ade-Saint Trojan	Bourcefranc	Marennes	Galon d'Or
2015	68	X	X	X	48	38	X	X	X	X
2016	66	X	X	X	29	70	X	X	X	X
2017	158	X	X	X	65	222	X	X	X	X
2018	X	X	X	X	X	71	X	X	X	X
2019	55	25	17	18	24	45	36	27	45	X
2020	9	25	X	X	X	86	X	X	X	X
2021	X	X	X	X	X	X	X	X	X	X
2022	X	X	X	X	X	128	X	X	X	X
2023	43	X	X	X	X	123	X	X	X	X
<b>TOTAL</b>	<b>399</b>	<b>50</b>	<b>17</b>	<b>18</b>	<b>166</b>	<b>783</b>	<b>36</b>	<b>27</b>	<b>45</b>	<b>0</b>

Il est clair que les observations n'ont pas une revisite régulière et homogène en fonction de la zone géographique et de l'année. Par exemple, en 2019, il y a des échantillons pour neuf sites du fait que le projet CoHECO avaient de données de l'année précédente. Alors qu'en 2021, il y en a aucune ou alors entre 2015 et 2017 il y en a plusieurs mais ne sont pas utilisables du fait qu'il y a pas d'images satellitaires Sentinel-2A à cette période.

Ces observations ont permis de recueillir des données sur la présence d'herbiers, leur fragmentation, ainsi que sur le taux de recouvrement.

Deux types de données ont été collectés sur le terrain :

- Le taux de recouvrement : Ce taux est mesuré à l'aide d'un quadrat de 50x50 cm, placé sur l'herbier. Bien que le quadrat soit positionné sur de l'herbier, la zone environnante peut être hétérogène. Afin de prendre en compte cette hétérogénéité, les valeurs de recouvrement sont ajustées en fonction d'un indice de fragmentation.
- L'indice de fragmentation : Cet indice est évalué visuellement sur le terrain pour des surfaces minimales de 10x10 m (correspondant à un pixel de l'image Sentinel-2). L'indice de fragmentation est ensuite utilisé pour ajuster les valeurs de l'indice de végétation par différence normalisée (NDVI), selon le schéma suivant :

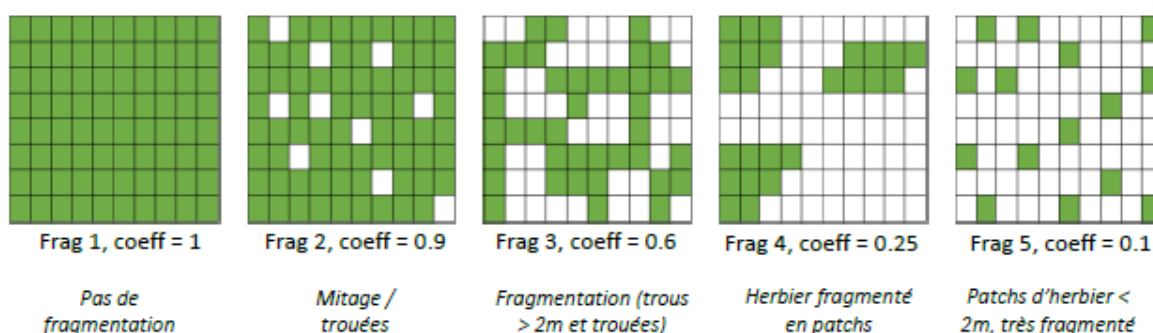


Figure 2 : Schéma des types de fragmentation du paysage observé sur le terrain

Cette méthodologie permet d'obtenir une évaluation précise du recouvrement des herbiers en tenant compte des variations locales de la végétation.

## 2) Images satellitaires Sentinel 2A

Les images satellites utilisées sont de niveau 2A (en réflectance) et ont été fournies via le programme Copernicus (« [Copernicus Browser](#) ») de l'Union européenne. Les capteurs des satellites Sentinel-2 sont des imageurs multispectraux, offrant 13 bandes spectrales en Flat REflectance (FRE) avec correction des effets atmosphériques et des effets de pente.

Ces images ont une résolution temporelle de 5 jours et une résolution spatiale de 10 mètres pour les bandes bleu, vert, rouge et proche infrarouge, de 20 à 60 mètres pour les autres bandes. Malgré leur bonne résolution temporelle, les images Sentinel-2 répondant aux critères d'analyse sont rares, c'est-à-dire avec des zones d'intérêts visibles (marée basse et sans nuages), sont rares sur la période.

Pour les années 2015, 2016, 2017, 2018, 2019, 2020, 2022 et 2023, la LPO a réalisé des observations terrain concernant le recouvrement par quadrat et la fragmentation. Les images, sélectionnées pour réaliser cet indice de recouvrement, sont présentées dans les tableaux 1 et 2.

Tableau 2 : Dates optimales (à marée basse et couverture nuageuse basse) pour télécharger les images Sentinel-2A sur l'Ile de Ré entre 2015 et 2024

	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
Mars	X	X	13	07	08	27	29	07	X	X
Avril	X	X	29	19	07	11	28	16	21	12
Mai	<b>X</b>	<b>X</b>	<b>24</b>	<b>19</b>	<b>X</b>	<b>26</b>	<b>31</b>	<b>31</b>	<b>X</b>	<b>10</b>
Juin	<b>X</b>	<b>X</b>	<b>16</b>	<b>13</b>	<b>18</b>	<b>22</b>	<b>12</b>	<b>17</b>	<b>05</b>	
Juillet	<b>X</b>	<b>23</b>	<b>16</b>	<b>13</b>	<b>03</b>	<b>22</b>	<b>X</b>	<b>02</b>	<b>07</b>	
Aout	X	12	27	02	02	06	11	11	21	

Tableau 3 : Dates optimales (à marée basse et couverture nuageuse basse) pour télécharger les images Sentinel-2A sur l'Ile d'Oléron entre 2015 et 2024

	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
Mars	X	X	10	20	20	24	29	24	04	28
Avril	X	X	29	19	09	X	X	X	18	12
Mai	<b>X</b>	<b>04</b>	<b>24</b>	<b>04</b>	<b>X</b>	<b>28</b>	<b>28</b>	<b>03</b>	<b>X</b>	<b>X</b>
Juin	<b>X</b>	<b>X</b>	<b>X</b>	<b>28</b>	<b>18</b>	<b>22</b>	<b>12</b>	<b>02</b>	<b>22</b>	
Juillet	<b>X</b>	<b>X</b>	<b>X</b>	<b>13</b>	<b>03</b>	<b>07</b>	<b>22</b>	<b>02</b>	<b>02</b>	
Aout	X	12	X	02	02	06	11	11	21	

Les années 2015, 2016 et, dans une moindre mesure, 2017 montrent qu'il y a très peu d'images disponibles, car les satellites Sentinel-2A et 2B ont été mis en orbite respectivement en 2015 et 2017. De plus, les images disponibles pour ces années présentent souvent une couverture nuageuse importante ou sont prises à marée haute.



Entre mars et août, les mois de juin et juillet présentent des valeurs de NDVI élevées, indiquant un pic de biomasse et de chlorophylle sur le site. Cela correspond généralement à un développement précoce des zostères, bien que d'autres espèces macrophytes puissent également contribuer. Utiliser des images satellitaires prises entre la mi-mai et le début août maximisera la détection des zostères, en minimisant la présence d'algues vertes et brunes qui tendent à proliférer avec l'augmentation de la température. En août et septembre, les zostères deviennent également une source d'alimentation essentielle pour certaines espèces d'oiseaux migrateurs et faisant diminuer le NDVI.

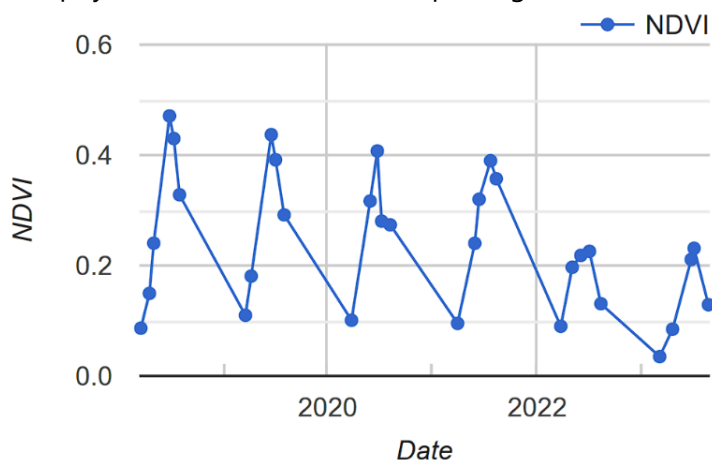


Figure 3 : NDVI mensuel pour les mois de mars au mois d'août de 2018 à 2023

## 2. Méthodologie

### A. Indice de recouvrement

Une fois les images satellitaires sélectionnées et téléchargées, la bande rouge (R : bande 4-665 nm) et la bande infrarouge (IR : bande 8-842 nm) sont utilisées pour calculer le NDVI (Tucker 1979), qui détecte la végétation à partir de son activité photosynthétique. Cet indice est très sensible à la couverture végétale et à la biomasse des herbiers (Zoffoli et al. 2020; 2023). Il est calculé comme suit :

$$\text{NDVI} = (\text{IR} - \text{R}) / (\text{IR} + \text{R})$$

Une dalle NDVI est créée et découpée selon l'emprise de distribution maximale des surfaces de zostères naines.

Zoffoli et al. 2020 a établi une relation linéaire entre le NDVI et le taux de recouvrement de l'herbier (Figure 4) :

$$\text{RECOUV\_PREDIT} = 172.06 \times \text{NDVI} - 22.18$$

Dalloyau et al. 2023 ont intégré les données d'observation terrain des herbiers, démontrant une relation non linéaire entre les taux de recouvrement et les valeurs de NDVI (Figure 5):

$$\text{IR} = 1(1 + \text{EXP}(-((-23.6137 \times (\text{NDVI}) \times (\text{NDVI}) + 24.4047 \times \text{NDVI} - 3.6108))))$$

Cette équation est appliquée aux données terrain actuelles. Pour ce faire, des données d'observation sur une année ainsi que la dalle NDVI correspondante ont été nécessaires. Les valeurs de chaque pixel du raster NDVI ont été extraites en fonction des données terrain.

Les valeurs de NDVI inférieures à 0.12 et supérieures à 0.8 n'ont pas été prises en compte. En effet, selon Zoffoli et al. 2020, les pixels NDVI en dehors de la plage 0.12-0.8 sont à masquer. La limite inférieure correspond à des sédiments nus ou à une contamination par une couche d'eau. La limite supérieure est déterminée à partir de mesures radiométriques comme la saturation du NDVI sur une couverture d'herbier marin dense. Selon Barillé et al. 2010, des valeurs supérieures à 0.8 ne correspondent pas aux herbiers marins, mais à l'accumulation de macroalgues dérivées. En examinant la figure 5, entre 0.4 et 0.6, il y a un plateau suivi d'une diminution, indiquant une saturation du NDVI.

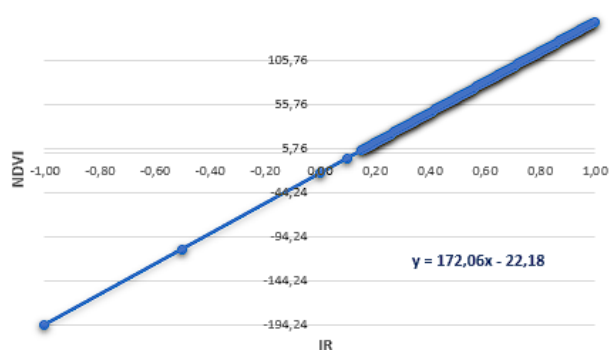


Figure 4 : Graphique de l'indice de recouvrement selon Zoffoli et al. 2020

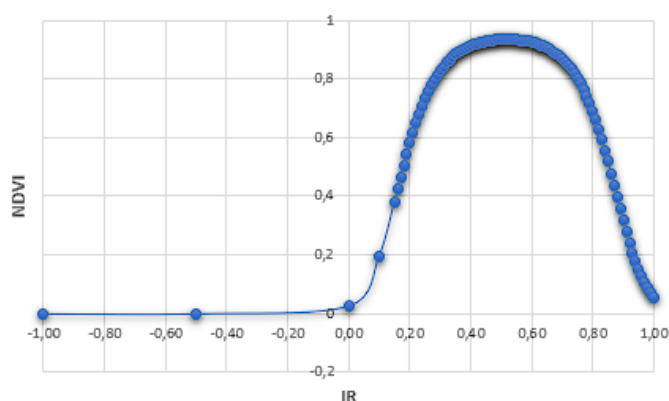


Figure 5 : Graphique de l'indice de recouvrement selon Dalloyau et al. 2023

L'équation finale (Figure 6) est obtenue en multipliant le NDVI par le pourcentage du recouvrement théorique qui correspond à l'indice de fragmentation en regardant le paysage sur 10 m (de 0.1 à 1) multiplié par le taux de recouvrement sur le quadrat (de 0 à 100). Cela donne un produit de double validation/ajustement permettant de traduire l'indice de végétation NDVI en taux de recouvrement foliaire en fonction des composantes environnementales locales.

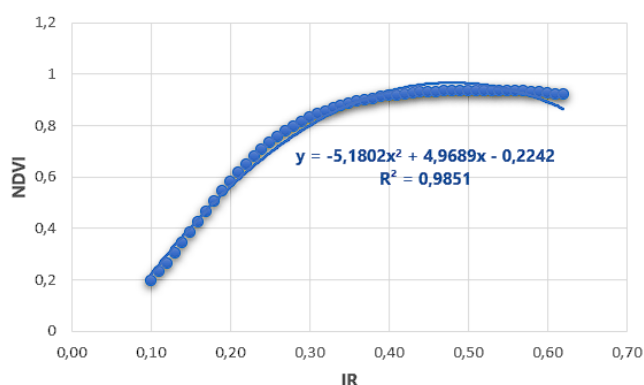


Figure 6 : Graphique de l'indice de recouvrement d'après Dalloyau et al. 2023 et ajustée à nos données

**IR en % =  $\frac{1}{1 + \exp(-((-5,1802 \times (\text{NDVI}) \times (\text{NDVI}) + 4,9689 \times \text{NDVI} - 0,2242))} \times [\text{RECOUV\_QUADRA} \times \text{FRAGMENTATION}]$**

Il faudra réaliser un ajustement annuel de l'équation étant donné que les valeurs de NDVI, de recouvrement et de fragmentation changent tous les ans en fonction de la période où les échantillons sont réalisés. Il faut, également, prendre en compte que la biomasse et la chlorophylle de la végétation évolue et est rarement la même au fil du temps.

Pour ce faire, il suffit de reprendre le tableur qui permet de calculer l'équation « EQUATION\_IR » en modifiant les valeurs. Il faut pour ce faire extraire les valeurs des pixels du NDVI où les observations se situent. Puis, les copier-coller dans le tableau ainsi que les valeurs de fragmentation et de recouvrement.

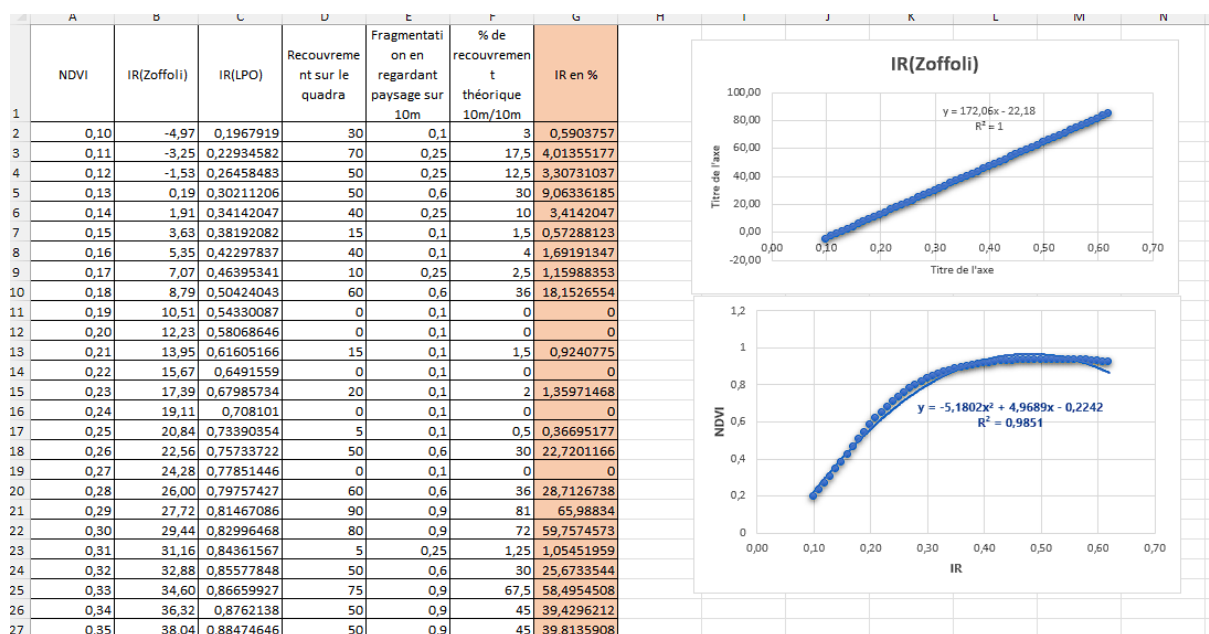


Figure 7 : Tableur permettant la mise à jour de l'équation

En modifiant les valeurs du NDVI, l'équation de l'indice de recouvrement brut sans la fragmentation et le recouvrement va changer et donc une nouvelle équation sur le deuxième graphique va apparaître.

## B. Conception de l'algorithme de l'indice de recouvrement

L'algorithme « IndiceSurfaciqueZosteresAlgorithm » est conçu pour calculer un indice de recouvrement pour les herbiers de *Zostera noltei*, en prenant en compte la couverture, le NDVI et la fragmentation de l'herbier.

Pour concevoir celui-ci plusieurs étapes sont nécessaires.

### 1) Initialisation de l'algorithme

Lors de l'initialisation, l'algorithme définit les paramètres nécessaires pour son fonctionnement. Ces paramètres incluent deux couches raster (Bandes rouge et infrarouge), une couche vectorielle représentant la zone d'intérêt, un fichier CSV contenant les observations de terrain, plusieurs coefficients pour le calcul de l'indice surfacique, et les destinations des fichiers de sortie pour les différents résultats (CSV modifié, NDVI recadré, indice surfacique recadrée, le pourcentage théorique de recouvrement et l'indice de recouvrement final).

### 2) Traitements des données rasters

L'algorithme commence par lire les données des deux couches raster fournies (Bandes rouge et infrarouge). Ces bandes sont utilisées pour calculer le NDVI. Les valeurs de NDVI sont ensuite filtrées pour ne conserver que celles comprises entre 0.12 et 0.8. Ce NDVI filtré est ensuite enregistré dans un nouveau raster temporaire.

### 3) Découpage et Calcul de l'Indice Surfacique

Le NDVI temporaire est ensuite découpé selon les limites de la couche vectorielle de la zone d'intérêt. Ensuite, l'algorithme utilise les coefficients fournis pour calculer l'indice surfacique basé selon l'équation. Cet indice est enregistré dans un raster temporaire qui est également découpé selon les limites de la zone d'intérêt.

### 4) Traitement des Données CSV et Interpolation

L'algorithme lit ensuite le fichier CSV contenant les observations de terrain. Les valeurs de fragmentation sont converties en valeurs numériques spécifiques et les coordonnées sont transformées en système de projection UTM. Le NDVI est interpolé pour chaque point d'observation. Les valeurs interpolées sont ajoutées au CSV qui est ensuite sauvegardé.

### 5) Calcul de la Couverture Théorique et Finale

L'algorithme utilise l'interpolation des valeurs de recouvrement et de fragmentation pour calculer un pourcentage de couverture théorique pour chaque pixel du NDVI filtré. Cette couverture théorique est enregistrée dans un nouveau raster.



Ensuite, l'algorithme multiplie l'indice surfacique recadré par la couverture théorique pour obtenir l'indice de recouvrement final. Le résultat est enregistré dans un raster final.

#### 6) Finalisation et Sorties

Enfin, l'algorithme renvoie les chemins des fichiers de sortie : le CSV modifié avec les valeurs de NDVI interpolées, le raster de l'indice surfacique recadré, et le raster de la couverture théorique.

Cet algorithme est ainsi conçu pour intégrer des observations de terrain avec des données raster pour produire un indicateur quantitatif utile pour l'étude des herbiers de *Zostera noltei*.

*Pour plus d'informations sur la conception d'une extension QGIS :*

*[http://www.qgistutorials.com/en/docs/3/processing\\_python\\_plugin.html](http://www.qgistutorials.com/en/docs/3/processing_python_plugin.html).*

### C. Estimation des échantillons

La LPO n'a pas suivi une méthodologie standard pour la collecte des échantillons, ce qui a entraîné une hétérogénéité des échantillons en fonction des zones géographiques et des années (*Tableau 1*). Cependant, ces observations sont cruciales pour développer l'indice de recouvrement des herbiers. Il est essentiel de déterminer si le nombre d'échantillons recueillis est suffisant pour couvrir notre zone d'étude et garantir un indice avec un bon niveau de confiance.

Pour obtenir une représentativité géographique adéquate, il est primordial que les points GPS soient répartis sur l'ensemble du site d'étude et non concentrés dans une seule zone. Cela assurera une homogénéité des pixels.

Le NDVI a été utilisé pour caractériser les pixels de la zone en les classant via une classification supervisée K-means, à partir des images satellitaires Sentinel-2A. Une grille a été créée sur cette classification, avec un échantillon par carreau. La taille de cette grille doit être ajustée selon les besoins de l'étude. Plus la grille est fine, plus elle est fiable et offre des résultats précis pour l'indice de recouvrement. Avec cette grille, un plan de prélèvement a été établi, garantissant une répartition parfaite grâce au carroyage.

Le test de Kolmogorov-Smirnov (KS) permet de mesurer la distance maximale entre deux distributions. Plus le résultat du test est faible, meilleure est la représentativité de l'échantillon. Si la distance entre les deux fonctions de distribution cumulative (ECDF) est petite, cela signifie que la distribution des échantillons est similaire à celle de l'ensemble de la zone, indiquant une bonne représentativité. En réduisant la taille des carreaux de la grille, le nombre d'échantillons et leur dispersion augmentent, ce qui peut améliorer la précision et la représentativité des échantillons. Le *tableau 5* montre que plus la grille est fine, plus les échantillons sont représentatifs. Cela va permettre d'estimer la taille d'échantillon optimal pour détecter une différence significative (*Annexe 3*).

Ensuite, le théorème de Cochran permet d'estimer la taille minimale de l'échantillon en fonction du niveau de marge d'erreur souhaité (5% soit 0.05), du niveau de confiance souhaité (95% soit un z-score de 1.96) et de la proportion de la caractéristique à laquelle on s'attend dans la population (50% soit 0.5) ainsi que du nombre d'échantillon que l'on aura trouvé grâce à notre plan de prélèvements fait au préalable. Le théorème de Cochran est :

$$n_0 = \frac{z^2 p(1 - p)}{E^2}$$

Puis, lorsque la taille de la population est petite ou finie, il faut ajuster la taille de l'échantillon. Comme suit :

$$n = \frac{n_0}{1 + \left( \frac{n_0 - 1}{N} \right)}$$

Soit :

z = valeur de z dans la table z en fonction du niveau de confiance souhaité

p = la proportion de la caractéristique présente dans la population à laquelle on s'attend

E = marge d'erreur

N = taille de la population

$n_0$  = la taille l'échantillon calculée grâce au théorème de Cochran

n = taille ajustée de l'échantillon pour une population finie

### 3. Utilisation de l'extension IndiceSurfaciqueZosteres sur Qgis

#### A. Manipulations préalables avant utilisation de l'extension

Pour utiliser le plugin développé pour QGIS, un ensemble de manipulations préalables est nécessaire. Voici les étapes à suivre :

1. **Téléchargement des images satellitaires** : Téléchargez les images satellitaires de l'année spécifique. Chaque année nécessite deux images : une pour la zone nord (incluant l'Île de Ré, la Baie d'Yves) et une pour la zone sud (incluant l'Île d'Oléron, Marennes, Galon d'Or).
2. **Disponibilité de l'emprise des herbiers de zostères naines** : Assurez-vous d'avoir téléchargé ou de disposer de l'emprise géographique des herbiers de zostères naines. Cette emprise sera utilisée pour découper les données NDVI et les indices de taux de recouvrement.
3. **Données d'observations terrain** : Disposez des données d'observations terrain sous forme de tableurs au format CSV avec comme délimitation un ';' . Assurez-vous que ces fichiers contiennent au moins les colonnes suivantes :
  - **Latitude / Longitude** : Colonnes nommées 'X\_WGS84' et 'Y\_WGS84' pour les coordonnées géographiques.
  - **Taux de recouvrement** : Colonne nommée 'RECOUV\_corr', avec des valeurs comprises entre 0 et 100.

- **Coefficient de fragmentation** : Colonne nommée 'FRAG\_corr', avec des valeurs comprises entre 0.1 et 1.

**Il est crucial que les noms de colonnes soient exactement comme spécifiés ci-dessous pour que le plugin fonctionne correctement. Si il y a d'autres colonnes cela ne dérange pas le bon fonctionnement de l'algorithme.**

*Tableau 4 : Exemple d'un tableur vierge avec colonnes types*

ID	Date	X_WGS84	Y_WGS84	RECOUV_corr	FRAG_corr
1	20/07/2023	-1.193049	45.862869	70	0.25

- 4. Choix de la sortie des fichiers** : Sélectionnez le dossier de sortie où seront enregistrés les fichiers résultants générés par le plugin.

En suivant ces étapes préliminaires, vous serez prêt à utiliser le plugin sur QGIS pour automatiser le traitement des images satellitaires, le calcul du NDVI, l'extraction des données d'observations terrain et la création des indices de taux de recouvrement, en prenant en compte les spécificités géographiques et environnementales locales des herbiers de zostères naines.

## B. Présentation et explication de l'interface

Le processus d'utilisation du plugin pour QGIS implique plusieurs étapes clés afin d'automatiser l'analyse des données satellitaires et des observations terrain des herbiers de zostères naines.

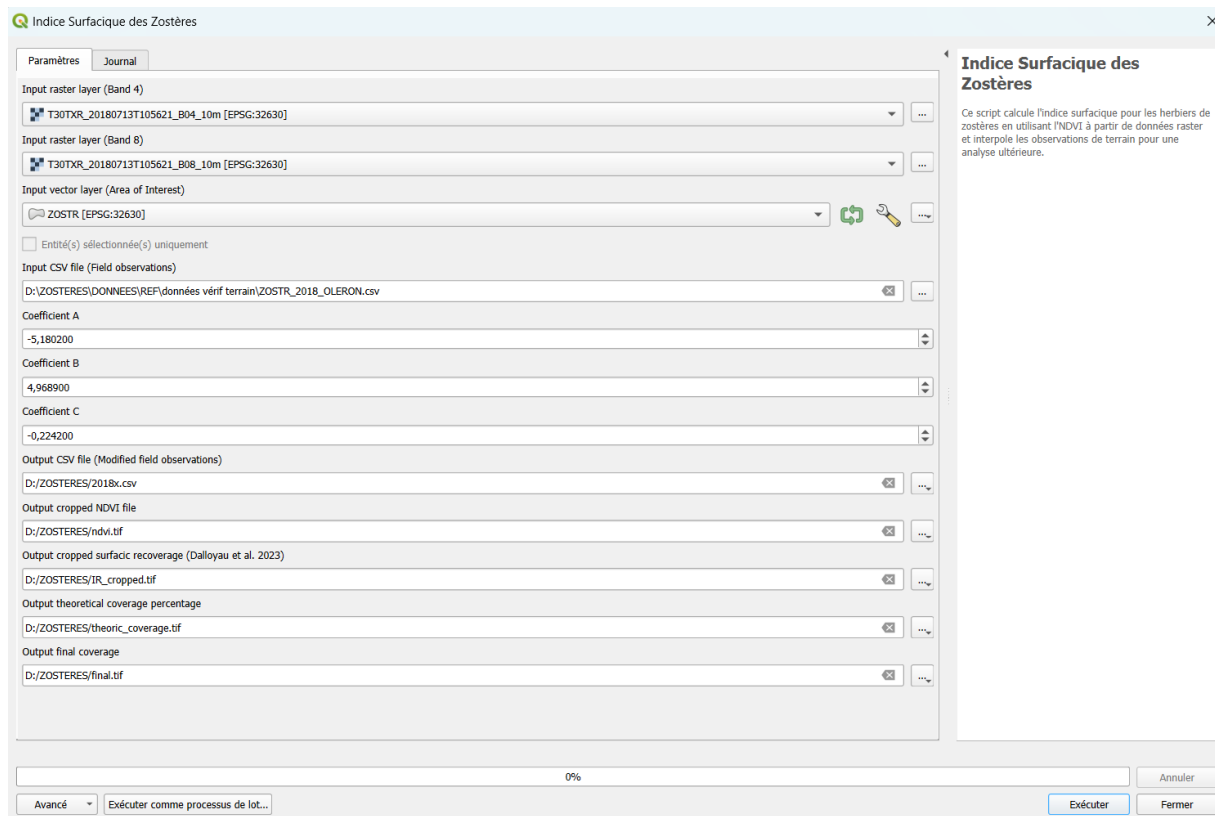


Figure 8 : Interface de l'extension Indice Surfacing Zostères sur QGIS

La *figure 8* illustre des entrées de fichiers qui sont à remplir. Tout d'abord, il faut rentrer les bandes des images satellitaires du rouge et de l'infrarouge permettant de calculer le NDVI et par la suite l'indice surfacic. Ensuite, il faut ajouter une couche shapefile représentant l'emprise de nos herbiers de zostères. Et enfin, nos observations terrain doivent être ajoutées en tant que fichier csv délimité par un « ; ».

Dans un second temps, les trois coefficients permettent de calculer l'équation. Ce sont des coefficients par défaut car ils sont calculés grâce aux données terrain et au NDVI de chaque année. Ce sont ces coefficients qui vont permettre de calculer l'évolution des herbiers.

Et enfin, il reste à renseigner le nom et l'emplacement de sauvegarde des différents fichiers. Des étapes intermédiaires ont été nécessaires, il s'agit notamment du NDVI découpé à l'indice surfacic final en passant par le tableur csv modifié ou encore le pourcentage de recouvrement théorique. Cliquer sur exécuter permet de générer les différents fichiers en sorties. Il ne reste plus qu'à récupérer les rasters ou tableurs et les importer dans QGIS ou un autre logiciel de SIG.



## 4. Résultats

### A. Estimation des échantillons

La zone d'étude Oléron-Viaduc illustre l'estimation des échantillons.

Dans un premier temps, le NDVI a été calculé (*Figure 9*). Les différentes nuances de couleurs représentent les variations des valeurs NDVI, allant des plus basses (zones moins végétalisées) aux valeurs les plus élevées (zones plus végétalisées).

L'histogramme (*Figure 10*) montre la distribution des valeurs NDVI dans la région d'étude. Il permet de visualiser la fréquence des différentes valeurs NDVI et d'identifier les tendances générales dans la végétation. Il y a beaucoup plus de pixels du NDVI autour de 0.4 qu'à 0.7, indiquant une végétation avec des taux de chlorophylle plus importants. Cela est accentué par le fait que cette région peut être imprégnée d'eau en raison des marées recouvrant certaines zones.

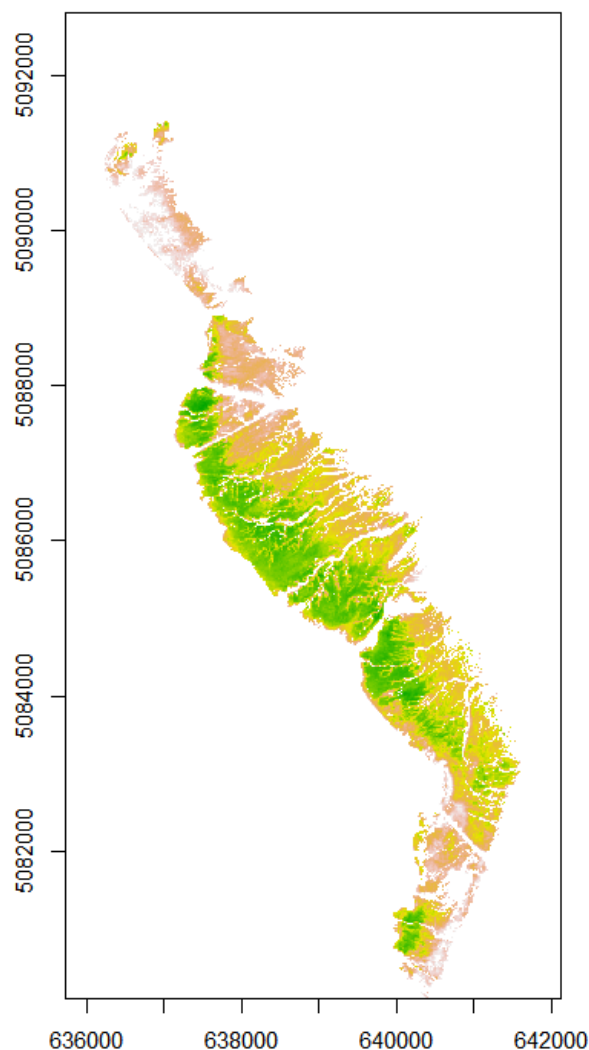


Figure 9 : NDVI d'Oléron-Viaduc

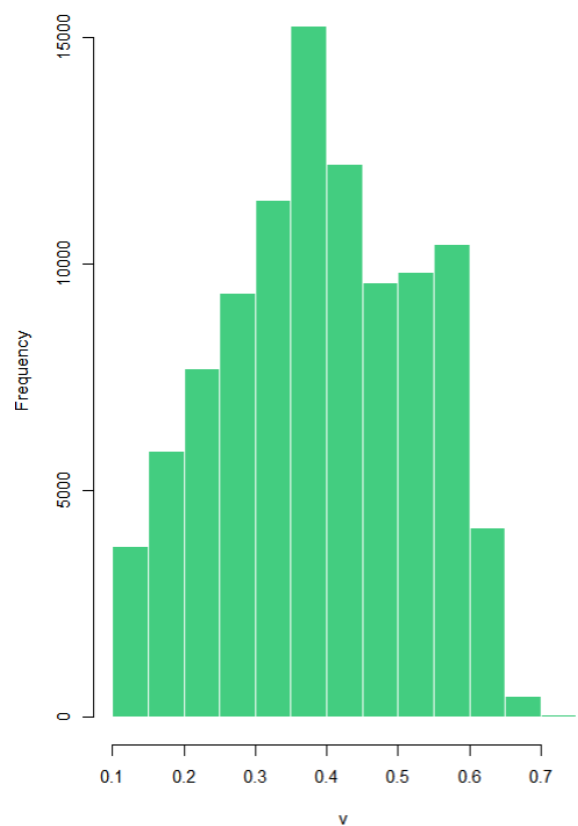


Figure 10 : Histogramme du NDVI

Le NDVI varie entre 0 et 0.7, permettant de déduire qu'il y a 7 classes distinctes de végétation. Une classification non-supervisée k-means a donc été réalisée pour ces 7 classes.

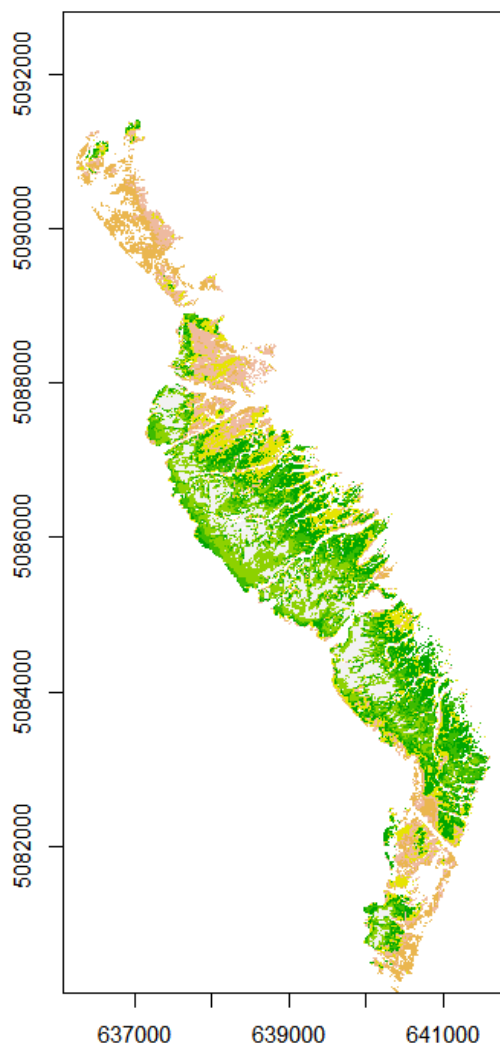


Figure 11: K-means d'Oléron-Viaduc

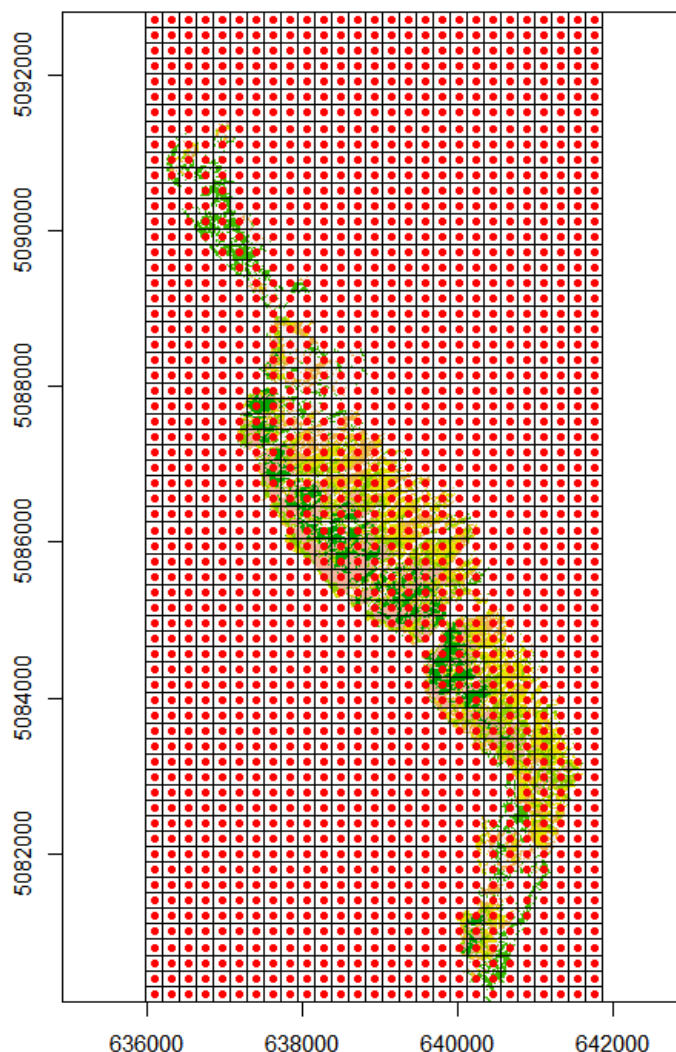


Figure 12 : Superposition des points de prélèvements sur les classes de NDVI

Elle révèle 7 zones hétérogènes en termes de pixels de végétation. Par exemple, la zone centrale (classe 6) d'Oléron-Viaduc montre des valeurs de NDVI élevées, suggérant la présence d'herbiers de *Zostera noltei*.

Une grille de 200/200 mètres a été superposée sur cette parcelle, divisant le raster en cellules de taille égale. Chaque cellule représente une unité d'échantillonnage potentielle. Les points rouges indiquent les centres géométriques de chaque cellule de la grille, utilisés pour extraire les valeurs NDVI aux points échantillonnés.

La *figure 12* combine les classes NDVI avec les points de prélèvements, permettant de visualiser où les prélèvements ont été effectués par rapport aux différentes classes de NDVI. La taille du carroyage pourra être mieux ajustée à l'avenir en fonction des besoins et des moyens.

Le *tableau 5* illustre le fait que plus le carroyage est fin, plus il y aura d'observations et donc qu'il faudra de moyens et de temps pour les réaliser.

Le test de KS a permis de comparer la distribution des valeurs NDVI dans l'ensemble de la zone d'étude avec celle des valeurs NDVI aux points de prélèvements. L'objectif est de vérifier si les échantillons prélevés sont représentatifs de l'ensemble de la région étudiée. Il compare l'ECDF des valeurs NDVI de l'ensemble de la zone avec celles des valeurs NDVI aux points de prélèvement. Puis, il calcule la distance maximale entre les deux courbes ECDF, qui est utilisée pour déterminer si les deux distributions sont significativement différentes.

La p-value obtenue du test KS indique si les deux distributions sont significativement différentes. La p-value est de 0.8959 traduisant que les échantillons sont représentatifs de l'ensemble de la zone d'étude. Si la p-value avait été inférieure à un seuil (généralement 0.05), cela aurait signifié que les distributions étaient différentes.

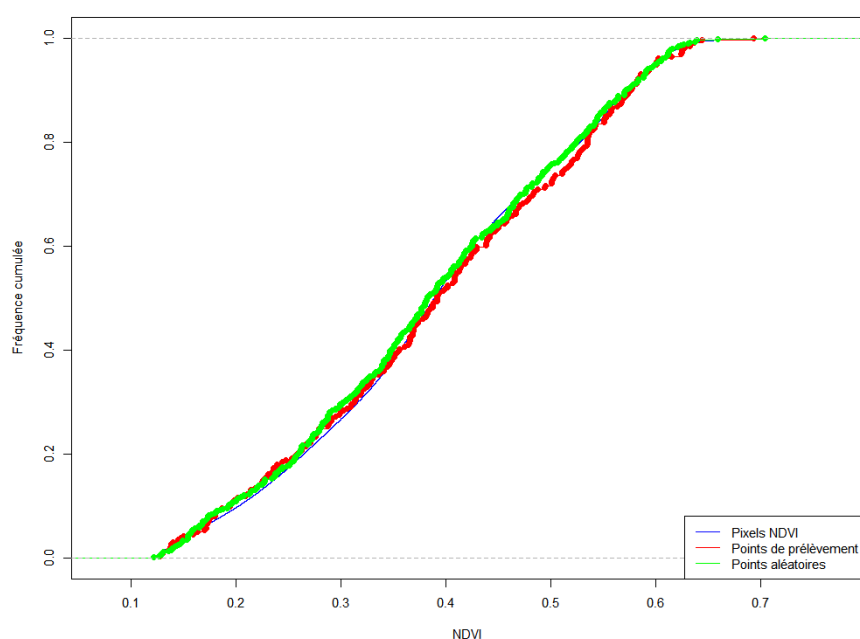


Figure 13 : Fonction de répartition cumulée des pixels de NDVI par rapport aux points de prélèvements et aux points aléatoires

La figure 13 montre l'ECDF des NDVI (courbe bleue) et compare cette distribution avec celle des valeurs NDVI aux points de prélèvements (courbe rouge) et des points aléatoires (courbe verte). Elle permet de comparer visuellement comment les différentes méthodes de prélèvements (grille vs aléatoire) capturent la distribution des

valeurs NDVI dans la région d'étude. Les courbes ECDF des valeurs NDVI globales et des points de prélèvements se chevauchent, cela indique que les points de prélèvements sont bien répartis et représentatifs de l'ensemble de la zone. La courbe verte des points aléatoires sert de référence pour évaluer la méthode de prélèvement par grille.

Le fait que les courbes ECDF suggèrent que la taille et la répartition des échantillons sont adéquates. En cas de différence significative, il serait nécessaire d'ajuster la taille de la grille ou la méthode d'échantillonnage pour obtenir une meilleure représentativité.

Le test de KS est un outil essentiel pour évaluer la représentativité des points de prélèvement dans une étude. En comparant les distributions des valeurs NDVI, il permet de s'assurer que les

échantillons prélevés reflètent fidèlement la variabilité de l'ensemble de la zone d'étude. Ce processus garantit que les conclusions tirées de l'étude sont robustes et fiables.

Dans un second temps, pour détecter une différence significative dans les valeurs NDVI, il est nécessaire de calculer la taille d'échantillon minimal en fonction de la variabilité des valeurs NDVI et du niveau de confiance souhaité. Le théorème de Cochran est utilisé pour estimer le nombre d'échantillons minimal requis pour Oléron-Viaduc afin d'obtenir un niveau de confiance élevé.

La formule du théorème de Cochran est la suivante pour Oléron-Viaduc :

$$n_0 = \frac{z^2 p(1-p)}{E^2} = \frac{1.96^2 * 0.5(1-0.5)}{0.05^2} = \frac{0.9604}{0.0025} = 384.16$$

Comme le nombre total d'observations est fini (260), la formule ajustée est appliquée :

$$n = \frac{n_0}{1 + \left( \frac{n_0 - 1}{N} \right)} = \frac{384.16}{1 + \left( \frac{384.16 - 1}{260} \right)} = \frac{384.16}{1 + (1.4737)} = 155.29 \approx 156$$

Avec :

- $z = 1.96$  (niveau de confiance de 95%)
- $p = 0.5$  (probabilité estimée)
- $E = 0.05$  (marge d'erreur de 5%)
- $N = 260$  (taille de la population)

Tableau 5 : Estimation de l'échantillonnage en fonction des surfaces du zonage des herbiers selon différentes méthodes

Zonage herbiers de Zostera noltei	Surface (km <sup>2</sup> )	Nombre d'échantillons selon ASPRS	Nbre d'échantillons selon le théorème de Cochran à 50m	Nbre d'échantillons selon le théorème de Cochran à 100m	Nbre d'échantillons selon le théorème de Cochran à 150m	Nbre d'échantillons selon le théorème de Cochran à 200m	Nbre d'échantillons selon le théorème de Cochran à 300m
Fiers d'Ars	2.75	20	284	161	95	56	29
Fosse de Loix	1.651	20	242	119	69	37	17
La Flotte	0.26	20	38	13	3	6	3
Rivedoux	0.252	20	58	18	7	8	3
Baie d'Yves	0.891	20	146	55	24	16	6
Oléron-Viaduc	10.561	20	352	281	212	156	90
Ade-Saint Trojan	2.024	20	259	131	71	44	21
Bourcefranc	0.163	20	58	20	24	16	2
Marennes	0.872	20	140	50	22	15	3
Galon d'Or	0.811	20	180	69	35	21	10
<b>TOTAL</b>	<b>20.135</b>	<b>200</b>	<b>1757</b>	<b>917</b>	<b>547</b>	<b>375</b>	<b>184</b>



Pour comprendre quelles estimations d'échantillons étaient la plus adaptées, il a fallu les comparer.

Selon l'ASPRS, les échantillons ne représentent que la part qui sert à valider les traitements. Cependant, ils sont calculés pour des grandes surfaces supérieures à 500 km<sup>2</sup> qui n'est pas le cas ici.

Le test KS et le théorème de Cochran, quant à eux, fournissent une méthode rigoureuse pour l'estimation des échantillons, essentielle pour des études et la gestion des ressources naturelles. De ce fait, cela garantit des conclusions robustes et fiables.

## B. Résultats de l'indice de recouvrement calculé à partir de l'extension

La figure 14 présente l'indice de recouvrement des zones d'Oléron-Viaduc, Ade-Saint Trojan, Bourcefranc, Marennes et Galon d'Or en 2018.

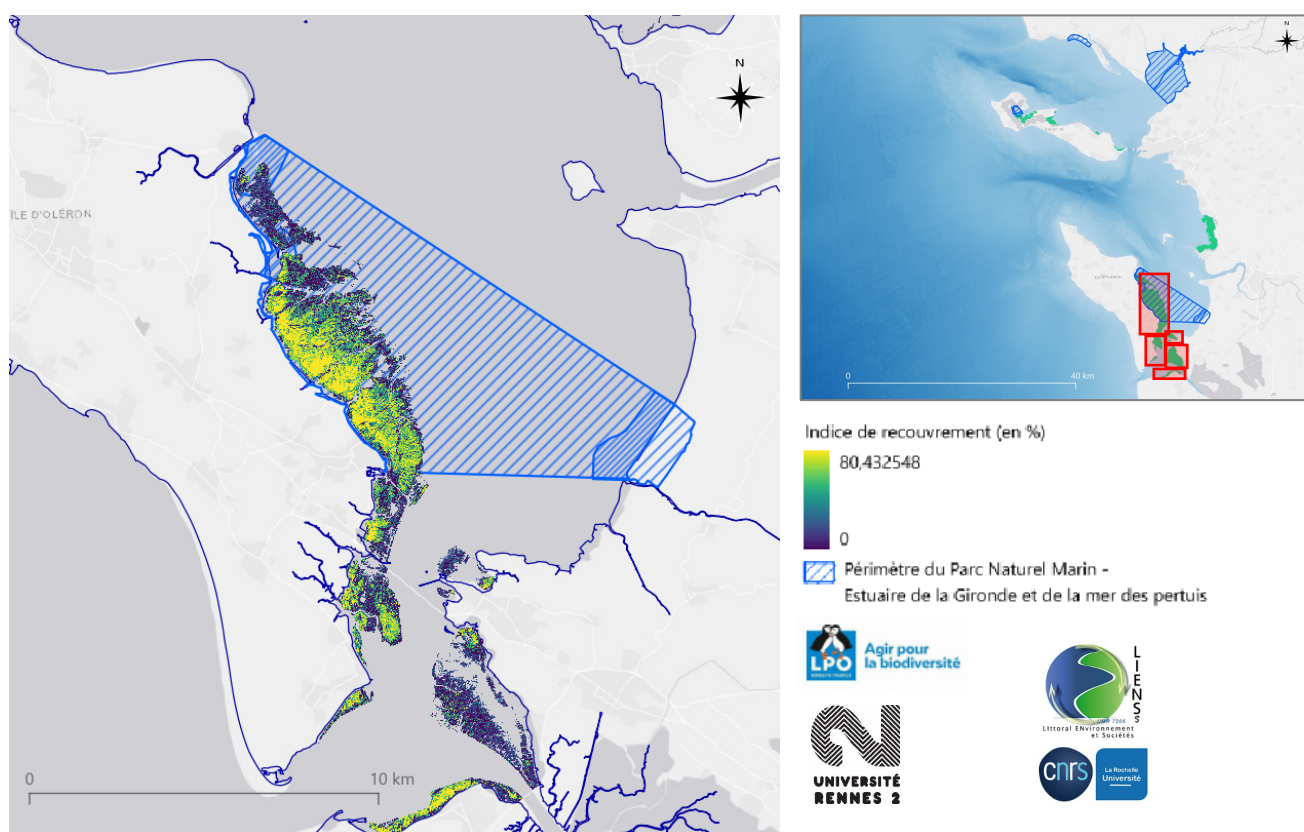


Figure 14 : Répartition de l'indice de recouvrement des herbiers de zostères en 2018

Il y a un indice de recouvrement particulièrement élevé à Oléron-Viaduc, atteignant entre 75 et 80%, indiquant des herbiers de *Zostera noltei* plus denses dans cette zone. En revanche, des régions telles que Marennes et Bourcefranc affichent un indice proche de 0%, ce qui révèle une absence d'herbiers. Cet indice est crucial pour suivre l'évolution de cette espèce et déterminer les mesures nécessaires pour sa protection ou sa restauration en cas de déclin.

## Bibliographie

Auby, I., Sébastien Dalloyau, et H. Oger-jeanneret. « Protocoles de suivi stationnel des herbiers à zostères pour la Directive Cadre sur l'Eau (DCE), *Zostera marina*, *Zostera noltii* ». Consulté le 3 mai 2024. [https://www.academia.edu/30823810/Protocoles\\_de\\_suivi\\_stationnel\\_des\\_herbiers\\_%C3%A0\\_zost%C3%A8res\\_pour\\_la\\_Directive\\_Cadre\\_sur\\_l\\_Eau\\_DCE\\_Zostera\\_marina\\_Zostera\\_noltii](https://www.academia.edu/30823810/Protocoles_de_suivi_stationnel_des_herbiers_%C3%A0_zost%C3%A8res_pour_la_Directive_Cadre_sur_l_Eau_DCE_Zostera_marina_Zostera_noltii).

Barillé, Laurent, Marc Robin, Nicolas Harin, Annaëlle Bargain, et Patrick Launeau. « Increase in seagrass distribution at Bourgneuf Bay (France) detected by spatial remote sensing ». *Aquatic Botany* 92, n° 3 (1 avril 2010): 185-94. <https://doi.org/10.1016/j.aquabot.2009.11.006>.

Benmokhtar, Salma, Marc Robin, Mohamed Maanan, et Hocein Bazairi. « Mapping and Quantification of the Dwarf Eelgrass *Zostera Noltei* Using a Random Forest Algorithm on a SPOT 7 Satellite Image ». *ISPRS International Journal of Geo-Information* 10, n° 5 (mai 2021): 313. <https://doi.org/10.3390/ijgi10050313>.

Cabaço, Susana, Raquel Machás, Vasco Vieira, et Rui Santos. « Impacts of urban wastewater discharge on seagrass meadows (*Zostera noltii*) ». *Estuarine, Coastal and Shelf Science* 78, n° 1 (1 juin 2008): 1-13. <https://doi.org/10.1016/j.ecss.2007.11.005>.

Copernicus Browser. « Copernicus Browser ». Consulté le 3 juillet 2024. <https://browser.dataspace.copernicus.eu/>.

Cunha, A. H., R. P. Santos, A. P. Gaspar, et M. F. Bairros. « Seagrass landscape-scale changes in response to disturbance created by the dynamics of barrier-islands: A case study from Ria Formosa (Southern Portugal) ». *Estuarine, Coastal and Shelf Science* 64, n° 4 (1 septembre 2005): 636-44. <https://doi.org/10.1016/j.ecss.2005.03.018>.

Dalloyau, Sébastien. *Qualité écologique des herbiers intertidaux à Zostère naine Zostera noltei dans les sites fonctionnels des Réserves Naturelles Nationales gérées par la LPO (Pertuis Charentais)*, 2020.

———. *Qualité écologique herbier sud marennnes 2015*, 2015.

Dalloyau, Sébastien, et Frédéric Robin. *Dark-bellied Brent Goose as a biodindicator of intertidal ecosystems*, 2013.

Duarte, Carlos M. « Marine biodiversity and ecosystem services: an elusive link ». *Journal of Experimental Marine Biology and Ecology* 250, n° 1 (30 juillet 2000): 117-31. [https://doi.org/10.1016/S0022-0981\(00\)00194-5](https://doi.org/10.1016/S0022-0981(00)00194-5).

Franzitta, Marco, Tiago Repolho, José Ricardo Paula, Isabel Caçador, Ana Rita Matos, Rui Rosa, et Bernardo Duarte. « Dwarf eelgrass (*Zostera noltii*) fatty acid remodelling induced by climate change ». *Estuarine, Coastal and Shelf Science* 261 (31 octobre 2021): 107546. <https://doi.org/10.1016/j.ecss.2021.107546>.

Ganter, B. « Seagrass (*Zostera* spp.) as food for brent geese (*Branta bernicla*): An overview ». *Helgoland Marine Research* 54 (7 janvier 2000): 63-70.

<https://doi.org/10.1007/s101520050003>.

Guimarães, M. Helena M. E., Alexandra H. Cunha, Rosemarie L. Nzinga, et João F. Marques. « The distribution of seagrass (*Zostera noltii*) in the Ria Formosa lagoon system and the implications of clam farming on its conservation ». *Journal for Nature Conservation* 20, n° 1 (1 janvier 2012): 30-40. <https://doi.org/10.1016/j.jnc.2011.07.005>.

« [https://www.asprs.org/a/society/divisions/pad/Accuracy/Draft\\_ASPRS\\_Accuracy\\_Standards\\_for\\_Digital\\_Geospatial\\_Data\\_PE&RS.pdf](https://www.asprs.org/a/society/divisions/pad/Accuracy/Draft_ASPRS_Accuracy_Standards_for_Digital_Geospatial_Data_PE&RS.pdf) ». Consulté le 28 juin 2024.

[https://www.asprs.org/a/society/divisions/pad/Accuracy/Draft\\_ASPRS\\_Accuracy\\_Standards\\_for\\_Digital\\_Geospatial\\_Data\\_PE&RS.pdf](https://www.asprs.org/a/society/divisions/pad/Accuracy/Draft_ASPRS_Accuracy_Standards_for_Digital_Geospatial_Data_PE&RS.pdf).

Ivajnsiĉ, Danijel, Martina Orlando-Bonaca, Daša Donša, Veno Jaša Grujić, Domen Trkov, Borut Mavriĉ, et Lovrenc Lipej. « Evaluating Seagrass Meadow Dynamics by Integrating Field-Based and Remote Sensing Techniques ». *Plants* 11, n° 9 (janvier 2022): 1196.

<https://doi.org/10.3390/plants11091196>.

Jacobs, R. P. W. M., C. Den Hartog, B. F. Braster, et F. C. Carriere. « Grazing of the seagrass *Zostera noltii* by birds at terschelling (Dutch Wadden Sea) ». *Aquatic Botany* 10 (1 janvier 1981): 241-59. [https://doi.org/10.1016/0304-3770\(81\)90026-7](https://doi.org/10.1016/0304-3770(81)90026-7).

« lea-moing/ExtensionQGIS-indice-surfacique-zosteres ». Consulté le 3 juillet 2024. <https://github.com/lea-moing/ExtensionQGIS-indice-surfacique-zosteres/tree/main>.

Orth, Robert, Tim Carruthers, William Dennison, Carlos Duarte, James Fourqurean, KENNETH JR, Anne Hughes, et al. « A Global Crisis for Seagrass Ecosystems ». *BioScience* 56 (1 décembre 2006): 987-96. [https://doi.org/10.1641/0006-3568\(2006\)56\[987:AGCFSE\]2.0.CO;2](https://doi.org/10.1641/0006-3568(2006)56[987:AGCFSE]2.0.CO;2).

Percival, Steve, et P. EVANS. « Brent Geese *Branta bernicla* and *Zostera*; Factors affecting the exploitation of a seasonally declining food resource ». *Ibis* 139 (28 juin 2008): 121-28.

<https://doi.org/10.1111/j.1474-919X.1997.tb04511.x>.

Scalpone, Cara, Jessie Jarvis, James Vasslides, Jeremy Testa, et Neil Ganju. « Simulated Estuary-Wide Response of Seagrass (*Zostera marina*) to Future Scenarios of Temperature and Sea Level ». *Frontiers in Marine Science* 7 (21 octobre 2020).

<https://doi.org/10.3389/fmars.2020.539946>.

Short, Frederick T., et Sandy Wyllie-Echeverria. « Natural and human-induced disturbance of seagrasses ». *Environmental Conservation* 23, n° 1 (1996): 17-27.

Tubbs, C. R., et J. M. Tubbs. « Brent geese *Branta bernicla bernicla* and their food in the solent, Southern England ». *Biological Conservation* 23, n° 1 (1 mai 1982): 33-54.

[https://doi.org/10.1016/0006-3207\(82\)90052-0](https://doi.org/10.1016/0006-3207(82)90052-0).

Zoffoli, Maria Laura, Pierre Gernez, Simon Oiry, Laurent Godet, Sébastien Dalloyau, Bede Ffianian Rowe Davies, et Laurent Barillé. « Remote Sensing in Seagrass Ecology: Coupled Dynamics between Migratory Herbivorous Birds and Intertidal Meadows Observed by Satellite during Four Decades ». *Remote Sensing in Ecology and Conservation* 9, n° 3 (2023): 420-33. <https://doi.org/10.1002/rse2.319>.

Zoffoli, Maria Laura, Pierre Gernez, Philippe Rosa, Anthony Le Bris, Vittorio E. Brando, Anne-Laure Barillé, Nicolas Harin, et al. « Sentinel-2 remote sensing of *Zostera noltei*-dominated intertidal seagrass meadows ». *Remote Sensing of Environment* 251 (15 décembre 2020): 112020. <https://doi.org/10.1016/j.rse.2020.112020>.



## Annexe 1 : Script Python permettant le traitement de l'indice de recouvrement des herbiers de *Zostera noltei*

```
### 1. import des librairies ###
import rasterio
from rasterio.mask import mask
from rasterio.transform import from_origin
import numpy as np
import geopandas as gpd
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from pyproj import Proj, transform
from scipy.interpolate import interp1d

### 2. Pré-traitements ###
# Chemin vers les bandes individuelles de l'image satellite
band4_path = "D:/ZOSTERES/OLERON/S2A_MSIL2A_20180713T105621_N0208_R094_T30TXR_20180713T140348.SAFE/S2A_MSIL2A_20180713T105621_N0208_R094_T30TXR_20180713T140348.SAFE/GRANULE/L2A_T30TXR_A015964_20180713T110521/IMG_DATA/R10m/T30TXR_20180713T105621_B04_10m.jp2"
band8_path = "D:/ZOSTERES/OLERON/S2A_MSIL2A_20180713T105621_N0208_R094_T30TXR_20180713T140348.SAFE/S2A_MSIL2A_20180713T105621_N0208_R094_T30TXR_20180713T140348.SAFE/GRANULE/L2A_T30TXR_A015964_20180713T110521/IMG_DATA/R10m/T30TXR_20180713T105621_B08_10m.jp2"

# Chemin vers le fichier GeoJSON
geojson_path = "D:/ZOSTERES/DONNEES/ZONES_ZOSTERES/REPRO_ZOSTR.geojson"

# Charger le fichier GeoJSON
geojson = gpd.read_file(geojson_path)

def read_and_crop_band(band_path, geojson):
    with rasterio.open(band_path) as src:
        out_image, out_transform = mask(src, geojson.geometry, crop=True)
        out_meta = src.meta.copy()
        out_meta.update({"driver": "JP2",
                        "height": out_image.shape[1],
                        "width": out_image.shape[2],
                        "transform": out_transform})
    return out_image[0], out_meta

# Lecture et découpage des bandes
band4, meta4 = read_and_crop_band(band4_path, geojson)
```

```

band8, meta8 = read_and_crop_band(band8_path, geojson)

### 3. Calcul du NDVI ###
def calculate_ndvi(nir, red):
    ndvi = (nir - red) / (nir + red)
    return ndvi

ndvi = calculate_ndvi(band8, band4)

# Filtrage des valeurs de NDVI entre 0.12 et 0.8
ndvi_filtered = np.where((ndvi >= 0) & (ndvi <= 0.8), ndvi, np.nan)

# Supprimer les valeurs NaN pour l'histogramme
ndvi_filtered_flat = ndvi_filtered[~np.isnan(ndvi_filtered)]

# Créer l'histogramme
plt.figure(figsize=(8, 6))
plt.hist(ndvi_filtered_flat, bins=20, color='purple', alpha=0.7)
plt.title('Distribution des valeurs de NDVI')
plt.xlabel('NDVI')
plt.ylabel('Fréquence')
plt.grid(True)
plt.show()

# Chemin vers le fichier CSV existant
csv_path = "D:/ZOSTERES/DONNEES/REF/données vérif terrain/ZOSTR_2018_OLERON\
.csv"

# Lire le fichier CSV existant avec un encodage spécifié
df = pd.read_csv(csv_path, encoding='ISO-8859-1', sep=';')

# Fonction pour convertir les valeurs de la colonne FRAG_corr
def convert_frag_value(value):
    if value == 1:
        return 1
    elif value == 2:
        return 0.9
    elif value == 3:
        return 0.6
    elif value == 4:
        return 0.25
    elif value == 5:
        return 0.1
    else:
        return value

```

```

# Appliquer la fonction de conversion à la colonne FRAG_corr
df['FRAG_corr'] = df['FRAG_corr'].apply(convert_frag_value)

# Afficher les premières lignes du DataFrame pour vérifier le résultat
print(df.head())

# Sauvegarder le nouveau fichier CSV
new_csv_path = "D:/ZOSTERES/DONNEES/ZOSTR_2018_NDVI.csv"
df.to_csv(new_csv_path, index=False, sep=',')

# Remplacer les virgules par des points dans les colonnes de coordonnées GPS
df['X_WGS84'] = df['X_WGS84'].str.replace(',', '.')
df['Y_WGS84'] = df['Y_WGS84'].str.replace(',', '.')

# Convertir les coordonnées GPS en EPSG:32630
df['X_UTM30N'], df['Y_UTM30N'] = transform(Proj(init='epsg:4326'),
                                           Proj(init='epsg:32630'),
                                           df['X_WGS84'].astype(float).values,
                                           df['Y_WGS84'].astype(float).values)

# Créer une colonne pour les valeurs de NDVI correspondant aux points GPS
df['NDVI_value'] = np.nan

# Remplir la colonne avec les valeurs de NDVI correspondant aux points GPS
for index, row in df.iterrows():
    x = row['X_UTM30N']
    y = row['Y_UTM30N']
    # Convertir les coordonnées UTM en indices de pixel dans l'image NDVI
    col, row = ~meta8['transform'] * (x, y)
    col, row = int(col), int(row)
    # Vérifier si les indices sont valides
    if 0 <= row < ndvi_filtered.shape[0] and 0 <= col < ndvi_filtered.shape[1]:
        # Récupérer la valeur de NDVI correspondant aux indices
        ndvi_value = ndvi_filtered[row, col]
        # Remplir la colonne NDVI_value avec la valeur de NDVI
        df.at[index, 'NDVI_value'] = ndvi_value

# Affichage du NDVI avec les points GPS superposés
plt.figure(figsize=(10, 8))
plt.imshow(ndvi_filtered, cmap='hot', extent=[meta8['transform'][2],
                                              meta8['transform'][2] \
                                              + ndvi_filtered.shape[1]*\
                                              meta8['transform'][0],
                                              meta8['transform'][5] \
                                              + ndvi_filtered.shape[0]\

```

```

        *meta8['transform'][4],
        meta8['transform'][5]))
plt.colorbar(label='NDVI')
plt.scatter(df['X_UTM30N'], df['Y_UTM30N'], c=df['NDVI_value'], cmap='hot',
            s=50, marker='o', label='Points GPS')
plt.title('NDVI avec superposition des points GPS et valeurs NDVI')
plt.xlabel('Coordonnée X (m)')
plt.ylabel('Coordonnée Y (m)')
plt.legend()
plt.show()

### 4. Calcul de l'indice surfacique ###
# Calcul de l'équation de Dalloyau et al.
indice_surf = 1.0 / (1.0 + np.exp(-1.0 * (-5.1802 * ndvi_filtered \
        * ndvi_filtered + 4.9689 \
        * ndvi_filtered - 0.2242)))

# Affichage de l'indice surfacique
plt.figure(figsize=(8, 6))
plt.imshow(indice_surf, cmap='viridis') # Choisissez la colormap appropriée
plt.colorbar(label='Indice surfacique')
plt.title('Indice surfacique calculé avec l\'équation de Dalloyau et al.')
plt.axis('off')
plt.show()

### 3. Interpolation et calcul des images de recouvrement et de fragmentation
# Créer les fonctions d'interpolation
interp_recouvrement = interp1d(df['NDVI_value'], df['RECOUV_corr'],
                                bounds_error=False, fill_value="extrapolate")
interp_fragmentation = interp1d(df['NDVI_value'], df['FRAG_corr'],
                                bounds_error=False, fill_value="extrapolate")

# Appliquer l'interpolation aux pixels de l'image NDVI
recouvrement_quadrat = interp_recouvrement(ndvi_filtered)
fragmentation_herbier = interp_fragmentation(ndvi_filtered)

# Affichage du recouvrement sur le quadrat
plt.figure(figsize=(8, 6))
plt.imshow(recouvrement_quadrat, cmap='viridis')
plt.colorbar(label='Recouvrement sur le quadrat')
plt.title('Recouvrement sur le quadrat en fonction du NDVI')
plt.axis('off')
plt.show()

# Affichage de la fragmentation de l'herbier
plt.figure(figsize=(8, 6))
plt.imshow(fragmentation_herbier, cmap='viridis')

```

```

plt.colorbar(label='Fragmentation de l\'herbier (%)')
plt.title('Fragmentation de l\'herbier en fonction du NDVI')
plt.axis('off')
plt.show()

# Flattening des valeurs de fragmentation de l'herbier
fragmentation_flat = fragmentation_herbier.flatten()

# Création de l'histogramme avec une palette de couleurs
plt.figure(figsize=(8, 6))
sns.histplot(fragmentation_flat, bins=20, palette='Blues', alpha=0.7)
plt.title('Répartition des valeurs de fragmentation de l\'herbier')
plt.xlabel('Fragmentation de l\'herbier (%)')
plt.ylabel('Fréquence')
plt.grid(True)
plt.show()

# Calcul du pourcentage de recouvrement théorique
pourcentage_recouvrement_theorique = recouvrement_quadrat * fragmentation_herbier

# Calcul de l'indice de recouvrement en pourcentage
indice_recouvrement_pourcentage = indice_surf * pourcentage_recouvrement_theorique

# Affichage de l'indice de recouvrement en pourcentage
plt.figure(figsize=(8, 6))
plt.imshow(indice_recouvrement_pourcentage, cmap='viridis')
plt.title('Indice de recouvrement en pourcentage')
plt.axis('off')
plt.show()

```

## Annexe 2 : Scripts extension Qgis

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jun 18 17:00:45 2024

@author: leamo
"""

# -*- coding: utf-8 -*-

"""
/*****
IndiceSurfaciqueZosteres
        A QGIS plugin
Indice surfacique permettant de calculer le taux de recouvrement des Zostera noltei en prenant
en compte le recouvrement, le NDVI et la fragmentation de cet herbier.

Generated by Plugin Builder: http://g-sherman.github.io/Qgis-Plugin-Builder/
-----
begin          : 2024-06-04
copyright      : (C) 2024 by Léa Moing - LIENSs (Littoral Environnement Sociétés)
email          : lea.moing@outlook.fr
*****/

/*****
*
*
* This program is free software; you can redistribute it and/or modify *
* it under the terms of the GNU General Public License as published by *
* the Free Software Foundation; either version 2 of the License, or    *
* (at your option) any later version.                                   *
*
*
*****/

"""

__author__ = 'Léa Moing - LIENSs (Littoral Environnement Sociétés)'
__date__ = '2024-06-04'
__copyright__ = '(C) 2024 by Léa Moing - LIENSs (Littoral Environnement Sociétés)'

# This will get replaced with a git SHA1 when you do a git archive

__revision__ = '$Format:%H$'
import numpy as np
import pandas as pd
from scipy.interpolate import griddata, interp1d
from osgeo import gdal, ogr
from pyproj import Proj, transform
```



```

from qgis.PyQt.QtCore import QApplication
from qgis.core import (QgsProcessing,
                        QgsFeatureSink,
                        QgsProcessingAlgorithm,
                        QgsProcessingParameterFeatureSource,
                        QgsProcessingParameterRasterLayer,
                        QgsProcessingParameterFile,
                        QgsProcessingParameterNumber,
                        QgsProcessingParameterFileDestination,
                        QgsCoordinateTransform,
                        QgsCoordinateReferenceSystem,
                        QgsProcessingException)
from qgis import processing

class IndiceSurfaciqueZosteresAlgorithm(QgsProcessingAlgorithm):
    """
    This algorithm calculates the coverage index for Zostera noltei by considering the coverage,
    NDVI, and fragmentation of this seagrass. The result is saved as a TIFF file.
    """

    OUTPUT = 'OUTPUT'
    INPUT_RASTER_BAND4 = 'INPUT_RASTER_BAND4'
    INPUT_RASTER_BAND8 = 'INPUT_RASTER_BAND8'
    INPUT_VECTOR = 'INPUT_VECTOR'
    INPUT_CSV = 'INPUT_CSV'
    COEFFICIENT_A = 'COEFFICIENT_A'
    COEFFICIENT_B = 'COEFFICIENT_B'
    COEFFICIENT_C = 'COEFFICIENT_C'
    OUTPUT_CSV = 'OUTPUT_CSV'
    OUTPUT_NDVI_CROPPED = 'OUTPUT_NDVI_CROPPED'
    OUTPUT_SURFACIC_CROPPED = 'OUTPUT_SURFACIC_CROPPED'
    OUTPUT_THEORETICAL_COVERAGE = 'OUTPUT_THEORETICAL_COVERAGE'
    OUTPUT_FINAL_COVERAGE = 'OUTPUT_FINAL_COVERAGE'

    def initAlgorithm(self, config):
        """
        Define the inputs and output of the algorithm.
        """

        self.addParameter(
            QgsProcessingParameterRasterLayer(
                self.INPUT_RASTER_BAND4,
                self.tr("Input raster layer (Band 4)")
            )
        )

        self.addParameter(

```

```

        QgsProcessingParameterRasterLayer(
            self.INPUT_RASTER_BAND8,
            self.tr('Input raster layer (Band 8)')
        )
    )

    self.addParameter(
        QgsProcessingParameterFeatureSource(
            self.INPUT_VECTOR,
            self.tr('Input vector layer (Area of Interest)'),
            [QgsProcessing.TypeVectorAnyGeometry]
        )
    )

    self.addParameter(
        QgsProcessingParameterFile(
            self.INPUT_CSV,
            self.tr('Input CSV file (Field observations)'),
            extension='csv'
        )
    )

    self.addParameter(
        QgsProcessingParameterNumber(
            self.COEFFICIENT_A,
            self.tr('Coefficient A'),
            type=QgsProcessingParameterNumber.Double,
            defaultValue = -5.1802
        )
    )

    self.addParameter(
        QgsProcessingParameterNumber(
            self.COEFFICIENT_B,
            self.tr('Coefficient B'),
            type=QgsProcessingParameterNumber.Double,
            defaultValue = 4.9689
        )
    )

    self.addParameter(
        QgsProcessingParameterNumber(
            self.COEFFICIENT_C,
            self.tr('Coefficient C'),
            type=QgsProcessingParameterNumber.Double,
            defaultValue = -0.2242
        )
    )

    self.addParameter(
        QgsProcessingParameterFileDestination(

```

```

        self.OUTPUT_CSV,
        self.tr('Output CSV file (Modified field observations)'),
        fileFilter='CSV files (*.csv)'
    )
)

self.addParameter(
    QgsProcessingParameterFileDestination(
        self.OUTPUT_NDVI_CROPPED,
        self.tr('Output cropped NDVI file'),
        fileFilter='TIFF files (*.tif)'
    )
)

self.addParameter(
    QgsProcessingParameterFileDestination(
        self.OUTPUT_SURFACIC_CROPPED,
        self.tr('Output cropped surfacic recoverage (Dalloyau et al. 2023)'),
        fileFilter='TIFF files (*.tif)'
    )
)

self.addParameter(
    QgsProcessingParameterFileDestination(
        self.OUTPUT_THEORETICAL_COVERAGE,
        self.tr('Output theoretical coverage percentage'),
        fileFilter='TIFF files (*.tif)'
    )
)

self.addParameter(
    QgsProcessingParameterFileDestination(
        self.OUTPUT_FINAL_COVERAGE,
        self.tr('Output final coverage'),
        fileFilter='TIFF files (*.tif)'
    )
)

```

```

def processAlgorithm(self, parameters, context, feedback):
    # Récupérer les paramètres d'entrée
    raster_band4 = self.parameterAsRasterLayer(parameters, self.INPUT_RASTER_BAND4, context)
    raster_band8 = self.parameterAsRasterLayer(parameters, self.INPUT_RASTER_BAND8, context)
    vector_layer = self.parameterAsVectorLayer(parameters, self.INPUT_VECTOR, context)
    csv_file = self.parameterAsFile(parameters, self.INPUT_CSV, context)
    output_csv = self.parameterAsFileOutput(parameters, self.OUTPUT_CSV, context)
    coefficient_a = self.parameterAsDouble(parameters, self.COEFFICIENT_A, context)

```

```

coefficient_b = self.parameterAsDouble(parameters, self.COEFFICIENT_B, context)
coefficient_c = self.parameterAsDouble(parameters, self.COEFFICIENT_C, context)
output_ndvi_cropped = self.parameterAsFileOutput(parameters, self.OUTPUT_NDVI_CROPPED, context)
output_surfacic_cropped = self.parameterAsFileOutput(parameters, self.OUTPUT_SURFACIC_CROPPED, context)
output_theoretical_coverage = self.parameterAsFileOutput(parameters, self.OUTPUT_THEORETICAL_COVERAGE, context)
output_final_coverage = self.parameterAsFileOutput(parameters, self.OUTPUT_FINAL_COVERAGE, context)

if raster_band4 is None or raster_band8 is None or output_ndvi_cropped is None:
    raise QgsProcessingException(self.tr("One or more input parameters are not correctly specified."))

feedback.pushInfo("Input raster Band 4: {}".format(raster_band4.source()))
feedback.pushInfo("Input raster Band 8: {}".format(raster_band8.source()))
feedback.pushInfo("Output cropped NDVI: {}".format(output_ndvi_cropped))
feedback.pushInfo("Output CSV: {}".format(output_csv))
feedback.pushInfo("Output cropped surfacic index: {}".format(output_surfacic_cropped))
feedback.pushInfo("Output theoretical coverage percentage: {}".format(output_theoretical_coverage))

# Lire les données raster
try:
    band4 = gdal.Open(raster_band4.source())
    band8 = gdal.Open(raster_band8.source())

    if band4 is None or band8 is None:
        raise QgsProcessingException(self.tr("Failed to open raster files. Please check if paths are correct and files are accessible."))

    bande_raster4 = band4.GetRasterBand(1)
    bande_raster8 = band8.GetRasterBand(1)
    donnees_raster4 = bande_raster4.ReadAsArray().astype(np.float32)
    donnees_raster8 = bande_raster8.ReadAsArray().astype(np.float32)

except Exception as e:
    raise QgsProcessingException(self.tr("Error reading raster data: {}".format(str(e))))

# Vérifier les dimensions des rasters
if donnees_raster4.shape != donnees_raster8.shape:
    raise QgsProcessingException(self.tr("Raster dimensions do not match."))

# Appliquer le calcul du NDVI
try:
    ndvi = (donnees_raster8 - donnees_raster4) / (donnees_raster8 + donnees_raster4)

```

```

except Exception as e:
    raise QgsProcessingException(self.tr("Error calculating NDVI: {}").format(str(e)))

# Filtrer les valeurs de NDVI entre 0.12 et 0.8
try:
    mask = (ndvi < 0.12) | (ndvi > 0.8)
    ndvi_filtered = np.copy(ndvi)
    ndvi_filtered[mask] = -9999
except Exception as e:
    raise QgsProcessingException(self.tr("Error applying NDVI filter: {}").format(str(e)))

# Enregistrer le NDVI filtré dans un nouveau raster temporaire
try:
    driver = gdal.GetDriverByName('GTiff')
    ndvi_temp = "/vsimem/ndvi_temp.tif" # Utilisation de la mémoire virtuelle pour le fichier temporaire
    ndvi_ds = driver.Create(ndvi_temp, raster_band4.width(), raster_band4.height(), 1, gdal.GDT_Float32)
    ndvi_ds.SetProjection(band4.GetProjection())
    ndvi_ds.SetGeoTransform(band4.GetGeoTransform())
    ndvi_ds.GetRasterBand(1).WriteArray(ndvi_filtered)
    ndvi_ds.GetRasterBand(1).SetNoDataValue(-9999)
    ndvi_ds.FlushCache()
except Exception as e:
    raise QgsProcessingException(self.tr("Error writing NDVI temporary output: {}").format(str(e)))

# Découper le raster NDVI selon l'emprise du vector_layer
try:
    warp_options = gdal.WarpOptions(
        cutlineDSName=vector_layer.source(),
        cropToCutline=True,
        dstNodata=-9999
    )
    gdal.Warp(output_ndvi_cropped, ndvi_temp, options=warp_options)
except Exception as e:
    raise QgsProcessingException(self.tr("Error cropping NDVI output: {}").format(str(e)))

# Calcul de l'indice surfacique avec des coefficients modifiables
# Calcul de l'équation de Dalloyau et al.
try:
    # Récupérer les coefficients des paramètres d'entrée
    coefficient_a = self.parameterAsDouble(parameters, self.COEFFICIENT_A, context)
    coefficient_b = self.parameterAsDouble(parameters, self.COEFFICIENT_B, context)
    coefficient_c = self.parameterAsDouble(parameters, self.COEFFICIENT_C, context)

    # Calcul de l'indice surfacique avec les coefficients spécifiés

```

```

        indice_surf = 1.0 / (1.0 + np.exp(-1.0 * (coefficient_a * ndvi_filtered * ndvi_filtered + coefficient_b * ndvi_filtered - coefficient_c)))
    except Exception as e:
        raise QgsProcessingException(self.tr("Error calculating surfacic index: {}").format(str(e)))
)

# Enregistrer l'indice surfacique dans un raster TIFF
try:
    driver = gdal.GetDriverByName('GTiff')
    surfacic_temp = "/vsimem/surfacic_temp.tif" # Utilisation de la mémoire virtuelle pour le fichier temporaire
    surfacic_ds = driver.Create(surfacic_temp, raster_band4.width(), raster_band4.height(), 1, gdal.GDT_Float32)
    surfacic_ds.SetProjection(band4.GetProjection())
    surfacic_ds.SetGeoTransform(band4.GetGeoTransform())
    surfacic_ds.GetRasterBand(1).WriteArray(indice_surf)
    surfacic_ds.GetRasterBand(1).SetNoDataValue(-9999)
    surfacic_ds.FlushCache()
except Exception as e:
    raise QgsProcessingException(self.tr("Error writing surfacic index output: {}").format(str(e)))

# Découper l'indice surfacique selon l'emprise du fichier shapefile
try:
    warp_options_surfacic = gdal.WarpOptions(
        cutlineDSName=vector_layer.source(),
        cropToCutline=True,
        dstNodata=-9999
    )
    gdal.Warp(output_surfacic_cropped, surfacic_temp, options=warp_options_surfacic)
except Exception as e:
    raise QgsProcessingException(self.tr("Error cropping surfacic index output: {}").format(str(e)))

# Lire le fichier CSV existant
try:
    df = pd.read_csv(csv_file, encoding='ISO-8859-1', sep=';')

    def convert_frag_value(value):
        if value == 1:
            return 1
        elif value == 2:
            return 0.9
        elif value == 3:
            return 0.6
        elif value == 4:
            return 0.25

```



```

        elif value == 5:
            return 0.1
        else:
            return value

df['FRAG_corr'] = df['FRAG_corr'].apply(convert_frag_value)
df['X_WGS84'] = df['X_WGS84'].str.replace(',', '.')
df['Y_WGS84'] = df['Y_WGS84'].str.replace(',', '.')
x_utm30n, y_utm30n = transform(Proj(init='epsg:4326'), Proj(init='epsg:32630'), df['X_
WGS84'].astype(float).values, df['Y_WGS84'].astype(float).values)
df['X_UTM30N'] = x_utm30n
df['Y_UTM30N'] = y_utm30n

df['NDVI_value'] = np.nan

ndvi_filtered_ds = gdal.Open(output_ndvi_cropped)
ndvi_filtered = ndvi_filtered_ds.GetRasterBand(1).ReadAsArray()
meta8 = ndvi_filtered_ds.GetGeoTransform()

for index, row in df.iterrows():
    x = row['X_UTM30N']
    y = row['Y_UTM30N']
    x_origin = meta8[0]
    y_origin = meta8[3]
    pixel_width = meta8[1]
    pixel_height = meta8[5]

    col = int((x - x_origin) / pixel_width)
    row = int((y - y_origin) / pixel_height)
    col, row = int(col), int(row)
    if 0 <= row < ndvi_filtered.shape[0] and 0 <= col < ndvi_filtered.shape[1]:
        ndvi_value = ndvi_filtered[row, col]
        df.at[index, 'NDVI_value'] = ndvi_value

df.to_csv(output_csv, index=False, encoding='ISO-8859-1', sep=';')
except Exception as e:
    raise QgsProcessingException(self.tr("Error processing CSV file: {}").format(str(e)))

try:
    interp_recouvrement = interp1d(df['NDVI_value'], df['RECOUV_corr'], bounds_error=False, fill_value="extrapolate")
    interp_fragmentation = interp1d(df['NDVI_value'], df['FRAG_corr'], bounds_error=False, fill_value="extrapolate")

    recouvrement_quadrat = interp_recouvrement(ndvi_filtered)
    fragmentation_herbier = interp_fragmentation(ndvi_filtered)

```

```

        pourcentage_recouvrement_theorique = recouvrement_quadrat * fragmentation_herbi
er
        pourcentage_recouvrement_theorique = np.clip(pourcentage_recouvrement_theoriqu
e,0,100)

        out_ds2 = driver.Create(output_theoretical_coverage, ndvi_filtered.shape[1], ndvi_filter
ed.shape[0], 1, gdal.GDT_Float32)
        out_ds2.SetGeoTransform(meta8)
        out_ds2.GetRasterBand(1).WriteArray(pourcentage_recouvrement_theorique)
        out_ds2.GetRasterBand(1).SetNoDataValue(-9999)
        out_ds2.FlushCache()
        out_ds2 = None

    except Exception as e:
        raise QgsProcessingException(self.tr("Error interpolating and calculating coverage: {}")
.format(str(e)))

    # Re-couper l'indice surfacique pour qu'il se superpose avec le % théorique
    try:
        warp_options_surfacic = gdal.WarpOptions(
            cutlineDSName=output_theoretical_coverage,
            cropToCutline=True,
            dstNodata=-9999
        )
        gdal.Warp(output_surfacic_cropped, surfacic_temp, options=warp_options_surfacic)

    except Exception as e:
        raise QgsProcessingException(self.tr("Error cropping surfacic index output to theoretic
al coverage: {}").format(str(e)))

    # Charger l'indice surfacique recoupé
    try:
        surfacic_cropped_ds = gdal.Open(output_surfacic_cropped)
        surfacic_cropped = surfacic_cropped_ds.GetRasterBand(1).ReadAsArray()
        meta = surfacic_cropped_ds.GetGeoTransform()
    except Exception as e:
        raise QgsProcessingException(self.tr("Error loading cropped surfacic index: {}").format(
str(e)))

    # Charger le pourcentage de recouvrement théorique
    try:
        theoric_coverage_ds = gdal.Open(output_theoretical_coverage)
        theoric_coverage = theoric_coverage_ds.GetRasterBand(1).ReadAsArray()
        meta = theoric_coverage_ds.GetGeoTransform()
    except Exception as e:
        raise QgsProcessingException(self.tr("Error loading theoretical coverage: {}").format(st
r(e)))

```

```

# Vérifier que les dimensions des deux rasters sont les mêmes
if surfacic_cropped.shape != theoric_coverage.shape:
    raise QgsProcessingException(self.tr("Dimensions of cropped surfacic index and theor
etical coverage do not match."))

# Calculer l'indice de recouvrement final
try:
    final_coverage = surfacic_cropped * theoric_coverage

    # Sauvegarder l'indice de recouvrement final
    out_raster = driver.Create(output_final_coverage, surfacic_cropped.shape[1], surfacic_c
ropped.shape[0], 1, gdal.GDT_Float32)
    out_raster.SetGeoTransform(meta)
    out_raster.SetProjection(surfacic_cropped_ds.GetProjection())
    out_raster.GetRasterBand(1).WriteArray(final_coverage)
    out_raster.FlushCache()

except Exception as e:
    raise QgsProcessingException(self.tr("An error occurred while saving the final coverag
e index: {}".format(str(e))))
    return {self.OUTPUT_CSV: output_csv, self.OUTPUT_SURFACIC_CROPPED: output_surfacic
_cropped, self.OUTPUT_THEORETICAL_COVERAGE: output_theoretical_coverage}

def name(self):
    return "indicesurfaciquezosteres"

def displayName(self):
    return self.tr("Indice Surfamique des Zostères")

def group(self):
    return self.tr("Mes Outils Personnalisés")

def groupId(self):
    return "mesoutils"

def shortHelpString(self):
    return self.tr("Ce script calcule l'indice surfamique pour les herbiers de zostères en utiliza
nt l'NDVI à partir de données raster et interpole les observations de terrain pour une analyse
ultérieure.")

def tr(self, string, context=""):
    return QApplication.translate(context, string)

def createInstance(self):
    return IndiceSurfaciqueZosteresAlgorithm()

```

## Annexe 3 : Script R permettant de réaliser le calcul de l'estimation des échantillons à faire

```
# Charger les bibliothèques
library(raster)

## Le chargement a nécessité le package : sp

library(sp)
library(sf)

## Linking to GEOS 3.12.1, GDAL 3.8.4, PROJ 9.3.1; sf_use_s2() is TRUE

library(dplyr)

##
## Attachement du package : 'dplyr'

## Les objets suivants sont masqués depuis 'package:raster':
##
##   intersect, select, union

## Les objets suivants sont masqués depuis 'package:stats':
##
##   filter, lag

## Les objets suivants sont masqués depuis 'package:base':
##
##   intersect, setdiff, setequal, union

library(ggplot2)
library(clustMixType) # Pour la classification K-means
library(viridis)

## Le chargement a nécessité le package : viridisLite

# 1. Charger et traiter les données NDVI
ndvi_raster <- raster("NDVI_MARENNES.tif")

# Afficher les valeurs NDVI
summary(ndvi_raster)

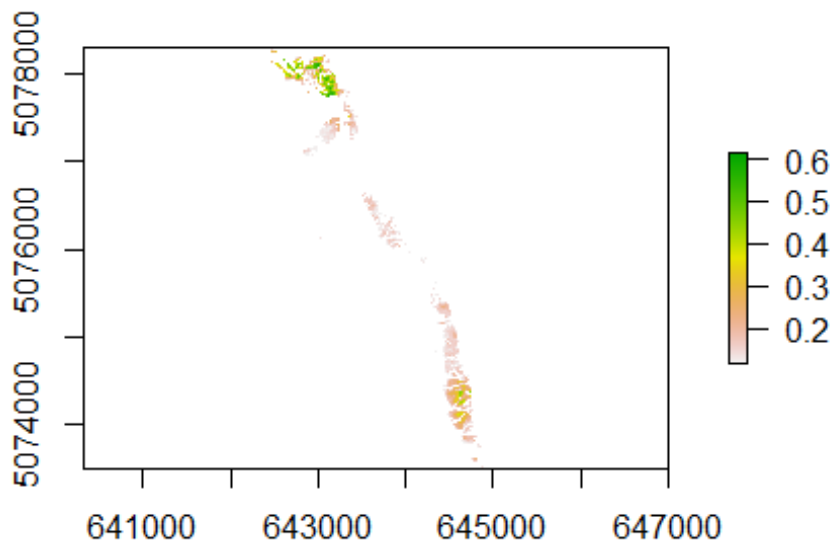
## Warning in .local(object, ...): summary is an estimate based on a sample of 1e+05 cells (85.
91% of all cells)

##      NDVI_MARENNES
## Min.   1.200000e-01
## 1st Qu. 1.421087e-01
## Median  1.731197e-01
## 3rd Qu. 2.565026e-01
```

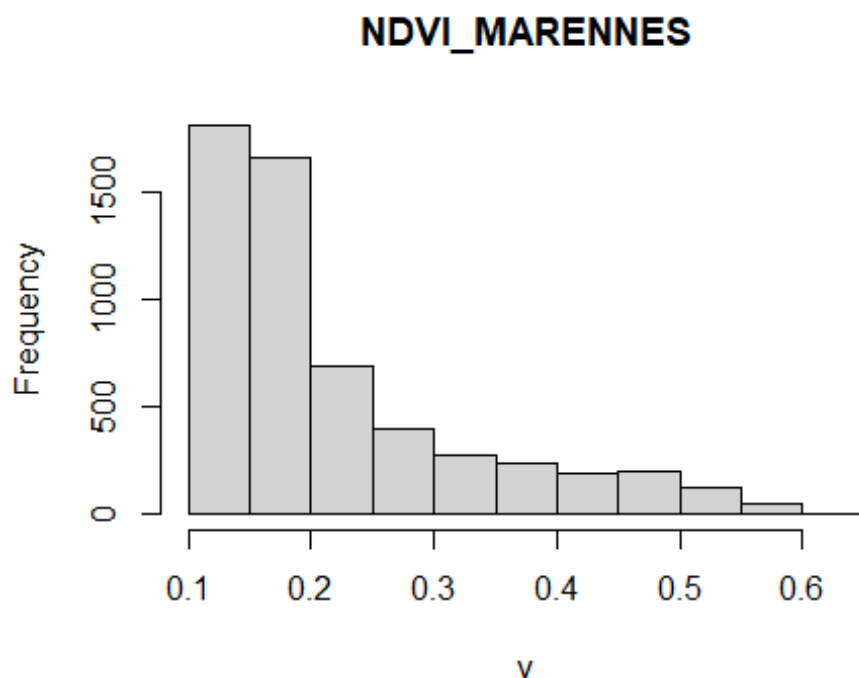
```
## Max. 6.140625e-01
## NA's 1.104070e+05
```

```
# Afficher les valeurs NDVI
```

```
plot(ndvi_raster)
```



```
hist(ndvi_raster)
```



```
# 2. Classification K-means sur les valeurs NDVI
```

```
ndvi_values <- getValues(ndvi_raster)
```

```
ndvi_values_non_na <- na.omit(ndvi_values)# rajouter commentaires
```

```
kmeans_result <- kmeans(ndvi_values_non_na, centers=7)
```

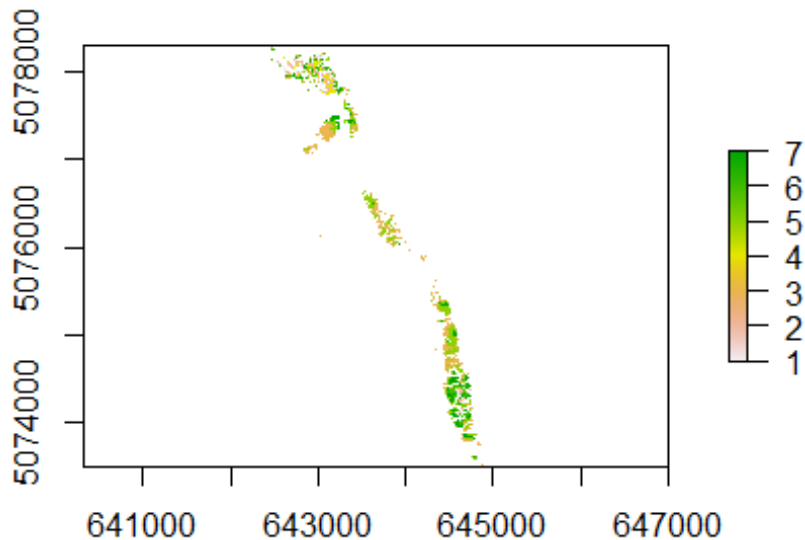
```
# Créer un vecteur pour stocker les classes NDVI
```

```
ndvi_classified <- rep(NA, length(ndvi_values))
```

```
ndvi_classified[!is.na(ndvi_values)] <- kmeans_result$cluster

# Créer un raster avec les classes NDVI
ndvi_classified_raster <- setValues(ndvi_raster, ndvi_classified)

# Afficher les classes NDVI
plot(ndvi_classified_raster)
```

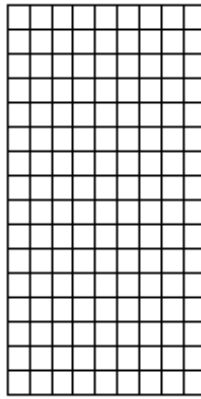


```
# 3. Importer la couche shapefile pour définir la zone d'intérêt
zone_interet <- st_read("DONNEES/ZONES_ZOSTERES/MARENNES_9.shp")

## Reading layer `MARENNES_9' from data source
## `D:\ZOSTERES\DONNEES\ZONES_ZOSTERES\MARENNES_9.shp' using driver `ESRI Shapefile'
## Simple feature collection with 7 features and 88 fields
## Geometry type: MULTIPOLYGON
## Dimension: XY
## Bounding box: xmin: 642440.2 ymin: 5073500 xmax: 644883.5 ymax: 5078303
## Projected CRS: WGS 84 / UTM zone 30N

# 4. Créer une grille sur la parcelle
extent_ndvi <- extent(ndvi_raster)
grid_size <- 300 # Taille de la cellule de la grille en mètres
grille <- raster(extent_ndvi, ncol=ceiling((extent_ndvi[2] - extent_ndvi[1]) / grid_size), nrow=ceiling((extent_ndvi[4] - extent_ndvi[3]) / grid_size))
grille <- setValues(grille, 1:ncell(grille))
grille_sp <- rasterToPolygons(grille, dissolve=TRUE)
plot(grille_sp)
```





```
# Calculer les centroids
centroids <- coordinates(grille_sp)

# 5. Comparer la distribution des NDVI avec les points de prélèvement
# Utiliser la fonction extract pour obtenir les valeurs NDVI aux points échantillonnés
sampled_ndvi <- extract(ndvi_raster, centroids)

# Test de Kolmogorov-Smirnov
ks_test <- ks.test(ndvi_values_non_na, sampled_ndvi)

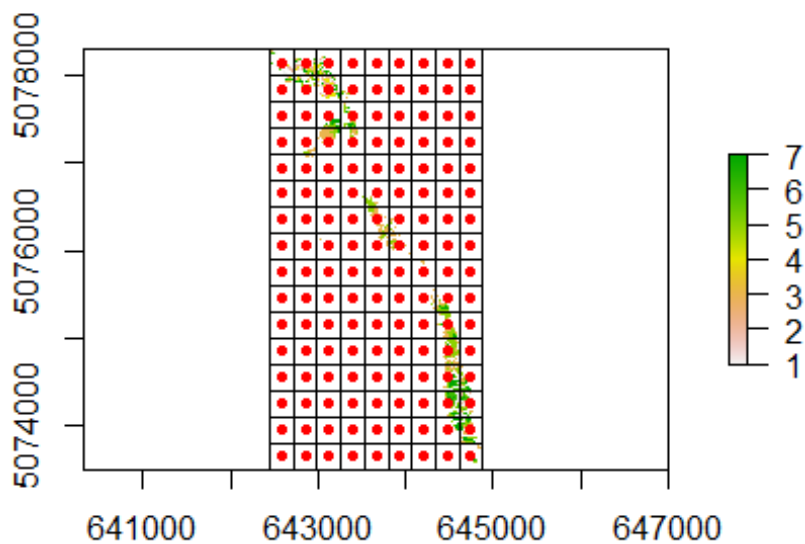
print(ks_test)

## Warning in ks.test.default(ndvi_values_non_na, sampled_ndvi): les valeurs p
## seront approximées en présence d'ex-aequos

# Estimer la taille d'échantillon optimal
sample_size <- function(ndvi_values_non_na, alpha=0.05, var_boot_ndvi) {
  # Calculer la taille d'échantillon minimal pour détecter une différence significative
  n_min <- (var_boot_ndvi * (1 + (1/alpha))) / (ndvi^2)
  return(n_min)
}

# Afficher la grille et les points de prélèvement sur la carte des classes NDVI
plot(ndvi_classified_raster, main="Classes NDVI et points de prélèvement")
plot(grille_sp, add=TRUE, border="black")
points(centroids, col="red", pch=20)
```

## Classes NDVI et points de prélèvement



*# Visualisation de la fonction de répartition cumulée*

```
ecdf_ndvi <- ecdf(ndvi_values_non_na)
```

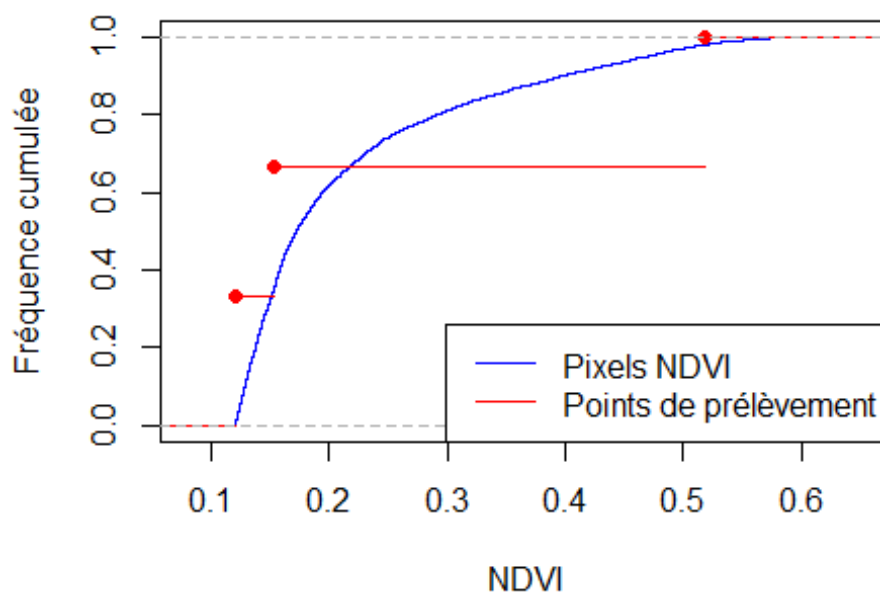
```
ecdf_sampled <- ecdf(sampled_ndvi)
```

```
plot(ecdf_ndvi, main="Fonction de répartition cumulée des NDVI", xlab="NDVI", ylab="Fréquence cumulée", col="blue")
```

```
lines(ecdf_sampled, col="red")
```

```
legend("bottomright", legend=c("Pixels NDVI", "Points de prélèvement"), col=c("blue", "red"), lty=1)
```

## Fonction de répartition cumulée des NDVI



```
summary(ecdf_sampled)
```

```
## Empirical CDF: 3 unique values with summary
##   Min. 1st Qu. Median   Mean 3rd Qu.  Max.
## 0.1205 0.1371 0.1537 0.2645 0.3365 0.5193

# Comparaison avec la méthode aléatoire
set.seed(123) # Pour reproductibilité
random_points <- spsample(as(extent_ndvi, "SpatialPolygons"), n=length(centroids), type="random")
random_ndvi <- extract(ndvi_raster, random_points)

ecdf_random <- ecdf(random_ndvi)

plot(ecdf_ndvi, main="Fonction de répartition cumulée des NDVI", xlab="NDVI", ylab="Fréquence cumulée", col="blue")
lines(ecdf_sampled, col="red")
legend("bottomright", legend=c("Pixels NDVI", "Points de prélèvement"), col=c("blue", "red"), lty=1)
lines(ecdf_random, col="green")
legend("bottomright", legend=c("Pixels NDVI", "Points de prélèvement", "Points aléatoires"), col=c("blue", "red", "green"), lty=1)
```

