



---

# **Analyse géomatique et structurale de linéaments morpho-tectoniques sur la Haute-Chaîne du Jura suisse**



*Le Locle ©Gérard Benoît à la Guillaume, Neuchâtel Tourisme*

## **Mémoire de stage de Certificat complémentaire en géomatique**

*Présenté par*

Clothilde Jost

Août 2023

Sous la direction de :

Dr. Pierre Lacroix (Université de Genève)

Composition du jury :

Dr. Jérôme Chablais  
(Hydro-Géo Environnement)

Dr. Gregory Giuliani  
(Université de Genève)



### Droits d'auteur

Les citations tirées du présent mémoire ne sont permises que dans la mesure où elles servent de commentaire, référence ou démonstration à son utilisateur. La citation doit impérativement indiquer la source et le nom de l'auteur. La loi fédérale sur le droit d'auteur est applicable.



## Remerciements

Je tiens à remercier tout d'abord Hydro-Géo Environnement qui m'a accueillie à bras ouverts. Merci à Marc Vallet pour m'avoir obtenu ce stage chez Hydro-Géo Environnement, et à Jérôme Chablais de m'avoir supervisée et encadrée avec bienveillance. Merci à Pierre Lacroix pour sa supervision, sa disponibilité sans faille et sa gentillesse. Merci David Polasek pour ta patience et ton expertise sur les notions de géologie structurale, Maud Brentini pour avoir fait apparaître miraculeusement les documents HGE dont j'avais besoin dans ma « petite boîte », pour ton aide sur les notions de stratigraphie et pour les tips QGIS.

Merci à Marius Gruber, Ludovic Savoy et Antoine Haddad pour vos idées constructives. Merci à Lydiane Thureau et Nadine Thureau pour leur gentillesse et leur aide administrative sans faille. Merci à tous ceux qui ont partagé mon quotidien avec leurs sourires, leurs salades, leur enthousiasme pour la géologie et l'environnement et leur bonne humeur : Blaise, Caroline, Elme, Fanny, Florian, Alexandra, Iris, Jérôme B., Lucas, Marc Hott., Marc O., Marie-Laure, Mickaël, Pierre-Clément, Vincent.

Merci aux membres et professeurs du Certificat complémentaire en géomatique de l'Université de Genève pour toutes les connaissances que j'ai acquises lors des cours du semestre de printemps 2023.

Enfin, un grand merci à mes proches pour leur soutien indéfectible dans cette dernière ligne droite de mes études universitaires.



## Résumé

L'étude des structures morpho-tectoniques comme les linéaments permet de caractériser la connectivité d'un réseau de fractures. Les zones de fractures bien connectées présentent un intérêt pour la prospection géothermique. La télédétection de linéaments à l'aide de modèles numériques de terrain et de systèmes d'informations géographiques (SIG) constitue une étape préliminaire importante dans le cadre de campagnes de prospections géothermiques. Elle permet notamment de pointer des zones d'intérêt à forte densité de fractures et de comprendre le régime de champs de contraintes géomécaniques en place.

Dans le cadre des projets de géothermie présents et futurs de Hydro-Géo Environnement, ce travail de stage a pour but principal d'utiliser des modèles numériques de terrain (MNT) acquis par télédétection LiDAR pour l'identification des linéaments morpho-tectoniques et d'étudier comment optimiser le traitement et la caractérisation par SIG de ces derniers.

Notamment, cette étude comprend la réalisation d'un inventaire des outils SIG existants pour la détection des linéaments morpho-tectonique, un test d'automatisation de l'identification des linéaments, la création d'un outil sur QGIS à l'aide de Python permettant de simplifier la préparation des données pour l'analyse des linéaments à l'aide de systèmes d'information géographiques et la définition de méthodes pour la digitalisation et l'analyse des linéaments.

Les résultats montrent que les plugins Network GT, SAGA et GDAL présentent les meilleurs outils pour l'analyse des linéaments. Le plugin créé lors de ce travail, Lineament Analysis, permet de réduire de moitié le temps de préparation des données raster afin de digitaliser les linéaments, même si certaines fonctionnalités du plugin pourraient bénéficier d'améliorations. Enfin, les méthodes de digitalisation et d'analyse des linéaments ont permis d'étudier la connectivité du réseau de fractures d'une zone précise, Le Locle, et de définir des zones d'intérêts géothermiques.

Mots-clés : système d'information géographique (SIG), télédétection, linéaments morpho-tectoniques, Haute-Chaîne du Jura, Python, QGIS.



## Table des matières

1.	Introduction .....	10
1.1.	Contexte général .....	10
1.2.	Objectifs du projet et enjeux de l'étude .....	10
1.3.	Présentation de l'organisation hôte .....	11
1.4.	Exposé du sujet .....	11
2.	Généralités .....	14
2.1.	Téledétection et technologie LiDAR.....	14
2.2.	Contexte géologique et structural régional de la Haute Chaîne du Jura et du Locle	14
3.	Méthodologie géomatique ad hoc.....	17
3.1.	Présentation et caractérisation des données de départ .....	17
3.1.1.	Données rasters .....	17
3.1.2.	Données vectorielles.....	17
3.1.3.	Données en ligne (WMS).....	18
3.2.	Présentation et caractérisation des plugins QGIS utilisés .....	18
3.3.	Étapes de travail – workflow.....	19
3.3.1.	Préparation des données pour l'identification des linéaments.....	19
3.3.2.	Essai d'automatisation de l'identification des linéaments morpho-tectoniques.....	21
3.3.3.	Création du plugin QGIS Lineament Analysis pour la préparation des tuiles MNT – automatisation et optimisation Python .....	25
3.3.4.	Digitalisation semi-automatique des linéaments morpho-tectoniques sur la zone du Locle à l'aide du plug-in NetworkGT .....	26
3.3.5.	Analyse géométrique et topologique des linéaments à l'aide de NetworkGT .....	28
3.3.5.1.	Analyse géométrique .....	28
3.3.5.2.	Échantillonnage par grille.....	28
3.3.5.3.	Analyse topologique.....	30
4.	Résultats.....	32
4.1.	Linéaments identifiés et digitalisés dans la zone du Locle .....	32
4.2.	Résultats de l'analyse géométrique .....	32
4.2.1.	Diagramme de Rose .....	32
4.2.2.	Recouvrements chronologiques et familles de linéaments.....	33
4.2.3.	Répartition des longueurs de linéaments .....	35
4.3.	Résultats de l'analyse topologique – Propriétés topologiques .....	37
4.3.1.	Nœuds et branches .....	37
4.3.2.	Diagrammes ternaires.....	38
4.3.3.	Nombre de connexions par branche.....	39
4.3.4.	Nombre de connexions par ligne .....	41
4.3.5.	Intensités $P_{21}$ & $P_{22}$ .....	42
4.3.6.	Intensité $P_{21}$ pour les unités stratigraphiques .....	45
5.	Discussion.....	48
5.1.	Géologie structurale.....	48
5.2.	Les SIG au service de la caractérisation des linéaments et optimisations Python sur QGIS.....	52
5.3.	Problèmes rencontrés.....	52



5.4.	Discussion personnelle – objectifs du stage .....	53
6.	Conclusion .....	54
6.1.	Synthèse générale.....	54
6.2.	Critique de la réalisation concrète.....	54
7.	Bibliographie .....	56
8.	Liste des annexes .....	60



## Liste des figures

Figure 1 : Carte géographique de la zone d'étude.....	13
Figure 2 : Carte des unités stratigraphiques affleurantes sur la zone d'étude.....	15
Figure 3 : Carte tectonique de Eichenberger et al., (2020).....	16
Figure 4 : Polygone de la zone d'étude sur relief swissALTI3D.....	19
Figure 5 : Carte des zones de digitalisation automatique.....	22
Figure 6 : Seuil pour l'ombrage 045/45, avec comme valeur de pixel de seuil 150. ....	23
Figure 7 : Carte de résultat de « Raster skeletonization » du plugin SAGA.....	24
Figure 8 : Schéma de processus de traitement du plugin Lineament Analysis.....	25
Figure 9 : Étapes de digitalisation des linéaments.....	26
Figure 10 : Étapes d'analyse des linéaments. ....	28
Figure 11 : Grille d'échantillonnage .....	29
Figure 12 : Schéma de topologie (Sanderson et Nixon, 2015).....	30
Figure 13 : Carte des linéaments digitalisés manuellement .....	32
Figure 14 : Diagramme de Rose pondéré par la longueur des linéaments.....	33
Figure 15 : Carte des linéaments digitalisés colorisés selon leur famille.....	34
Figure 16 : Histogramme de la distribution de la longueur des linéaments.....	35
Figure 17 : Carte montrant les linéaments colorisés selon leur longueur.....	36
Figure 18 : Carte montrant le résultat de « Branches and Nodes » de Network GT.....	38
Figure 19 : Diagramme ternaire montrant la proportion des types de nœuds.....	39
Figure 20 : Diagramme ternaire montrant les différents types de branches.....	39
Figure 21 : Carte montrant le nombre de connexions par branche. ....	40
Figure 22 : Carte montrant le nombre de connexions par ligne.....	41
Figure 23 : Carte montrant l'intensité $P_{21}$ .....	43
Figure 24 : Carte montrant l'intensité sans dimension $P_{22}$ .....	44
Figure 25 : Histogramme de l'intensité $P_{21}$ moyenne.....	46
Figure 26: Boxplot de l'intensité $P_{21}$ moyenne pour les unités stratigraphiques.....	47
Figure 27 : Carte tectonique avec les linéaments de la zone du Locle.....	50
Figure 28 : Carte géologique avec les linéaments de la zone du Locle.....	51
Figure 29 : Modèle de départ réalisé avec le Modeleur Graphique sur QGS 3.18.1. ....	61
Figure 30: Image du contenu du dossier créé par le plugin Plugin Builder. ....	62
Figure 31 : Image d'une partie du code Python contenu dans le fichier <code>__init__.py</code> .....	62
Figure 32 : Interface de Qt Designer et interface de Lineament Analysis. ....	63

## Liste des tables

Tableau 1 : Propriétés de taille des linéaments par familles.....	35
Tableau 2 : Récapitulation topologique.....	37



## Liste des abréviations

LiDAR : Light Detection and Ranging

MNT : Modèle Numérique de Terrain

SIG : Système d'Information Géographique

## Glossaire

Aquifère : Couche, formation géologique imprégnée d'eau.

Affleurement : Intersection d'une formation géologique, d'un gisement avec la surface du sol.

Anticlinal : Se dit d'un pli dont les éléments à l'intérieur de la courbure étaient, originellement, les plus bas et donc les plus vieux.

ArcPy : Bibliothèque de scripts Python développée par Esri pour automatiser certaines tâches sur ArcGIS.

Axe du pli : Ligne imaginaire parallèle qui indique la direction du pli.

Bassin d'avant-pays : bassin d'accumulation parallèle à une chaîne de montagnes qui reçoit les sédiments issus de l'érosion.

Bassin sédimentaire : Bassin d'accumulation de sédiments issus de l'érosion.

Branche : segment de fracture dont les extrémités sont des nœuds.

Branche I-I, I-C, C-C : types de branches dont la nomenclature dépend des nœuds d'extrémités (détaillé à la section 3.3.5.3).

Chevauchement : recouvrement de couches de roches par mouvement compressif

Chronostratigraphie : Organisation et division des couches de roches selon leurs relations d'âge.

Forage : Ensemble des techniques permettant de creuser un puits

Formation géologique : Ensemble de couches regroupées selon leurs relations spatiales et temporelles.

Fractures : accident tectonique cassant.

Énergie géothermique : énergie stockée sous forme de chaleur dans le sous-sol.

Gradient géothermique : taux de variation de la chaleur en fonction de la profondeur.

Haute-Chaîne : Partie interne surélevée de la chaîne du Jura.

L1 : Set de linéaments orientés NNO-SSE (propre à cette étude).

L2 : Set de linéaments orientés N-S (propre à cette étude).

L3 : Set de linéaments orientés NE-SO (propre à cette étude).

Linéament morpho-tectonique : élément linéaire dû à un accident tectonique.

Lithologie : formation géologique.

Molasse : formation sédimentaire détritique.

Morpho-tectonique : dont la morphologie est due à l'activité tectonique

Nœud : Extrémité ou point de rencontre de fractures dans un réseau de fractures.

Ombrage : analyse de surface d'un DEM par illumination artificielle depuis une source ponctuelle

Open source : Logiciel libre d'accès, réutilisable et modifiable.





Partie externe du Jura : par rapport à la subduction entre la plaque européenne et la Plaque africaine, plus proche du front de subduction.

Partie interne du Jura : par rapport à la subduction entre la plaque européenne et la Plaque africaine, plus éloignée du front de subduction.

Perméabilité : Propriété d'une roche de permettre la pénétration de l'eau.

Plugin : Outil ou set d'outils pouvant être ajoutés dans un logiciel open source.

Porosité : volume relatif des vides présents dans la roche.

PyQGIS : Bibliothèque de scripts Python développée pour automatiser certaines tâches sur QGIS.

Réservoir : roche perméable imprégnée de pétrole, de gaz naturel ou d'eau.

Sismique 2D et 3D : méthodes d'investigation du sous-sol à l'aide d'ondes sismiques produites artificiellement.

Stylolite : Surfaces de dissolution en dents de scie dues à la pression.

Subduction : Plaque tectonique océanique qui passe en dessous d'une plaque tectonique continentale du fait de la tectonique des plaques.

Syndinal : Se dit d'un pli dont les éléments à l'intérieur de la courbure étaient, originellement, les plus hauts, et donc les plus jeunes.

Thalweg d'érosion : différence marquée dans le paysage du passage d'une couche de roche à l'autre due à l'érosion.

Topologie : Études des relations entre les fractures d'un réseau

Valorisation directe : Valorisation de l'énergie géothermique sans utilisation de pompe à chaleur.



## 1. Introduction

### 1.1. Contexte général

L'étude des structures morpho-tectoniques telles que les linéaments est nécessaire à la bonne caractérisation des propriétés d'écoulement des fluides dans un réservoir (Polasek, 2020). En effet, la topologie et la géométrie d'un système de fractures influencent grandement sa connectivité (Sanderson & Nixon, 2018). Les zones de fractures bien connectées constituent des zones d'intérêt pour la prospection géothermique (Polasek, 2020). La caractérisation des linéaments tectoniques de surface apporte également une première compréhension des champs de contraintes passés et actuels, primordial pour les prévisions géomécaniques en cas de planification de forages profonds.

Depuis le 1<sup>er</sup> janvier 2018, la Suisse a adopté la Stratégie énergétique 2050 (OFEN, 2023c). Il s'agit d'un ensemble de mesures législatives visant la loi sur l'énergie, les stratégies Réseaux électriques, la loi sur le CO<sub>2</sub> ainsi que la loi sur l'approvisionnement en électricité (LApEI). La révision de la loi sur le CO<sub>2</sub> vise à réduire les émissions de gaz à effet de serre de -50 % d'ici à 2030 et à atteindre la neutralité carbone d'ici 2050 (OFEV, 2023).

Dans ce contexte, la Confédération suisse encourage le développement et l'exploitation des énergies renouvelables. L'énergie géothermique, non émettrice de CO<sub>2</sub>, fait partie des énergies renouvelables disponibles sur le territoire suisse et les projets visant la valorisation directe sont au bénéfice de contributions de l'Office fédéral de l'Énergie (OFEN)(OFEN, 2023a). Ainsi, les projets de prospection ou de valorisation directe de l'énergie géothermique peuvent obtenir des contributions allant jusqu'à 60% des coûts (OFEN, 2023b).

L'étude des linéaments à partir de modèles numériques de terrain (MNT) et à l'aide de systèmes d'information géographique (SIG) permet de réaliser des analyses structurales préliminaires peu coûteuses, permettant de désigner des zones d'intérêts à forte densité de fractures afin de cibler les études structurales de terrain (Palamakumbura et al., 2020). Ce type d'analyse peut être intégré dans certains programmes de prospections géothermiques accompagnés par l'OFEN.

### 1.2. Objectifs du projet et enjeux de l'étude

Dans le cadre du Certificat Complémentaire en Géomatique, cette étude vise à explorer les outils SIG permettant de faciliter la caractérisation des linéaments morpho-tectoniques dans le cadre de travaux de prospection géothermique. Elle s'inscrit dans un projet existant au sein de l'entreprise Hydro-Géo Environnement (HGE), qui vise à évaluer le potentiel géothermique de la commune du Locle. Il s'agit également de développer une méthode et des outils simples qui permettront de faciliter les futures prospections géothermiques de Hydro-Geo Environnement.

Les principaux objectifs de ce travail sont : (1) la réalisation d'un inventaire des outils SIG existants pour la détection des linéaments morpho-tectoniques, (2) l'évaluation de la faisabilité d'une automatisation de l'identification des linéaments sur une large zone d'étude,



(3) la création d'un outil sur QGIS à l'aide de Python, permettant de simplifier la préparation des données pour l'analyse des linéaments à l'aide de SIG et (4) la définition de méthodes pour la digitalisation et l'analyse des linéaments.

De façon générale, il s'agira de répondre aux questions suivantes : comment optimiser au mieux la digitalisation, le traitement et l'analyse des linéaments morpho-tectoniques à l'aide des SIG ? Quels outils existent afin de faciliter l'analyse structurale ? Comment approcher l'automatisation et l'optimisation de la collecte de données et la préparation de ces dernières à l'aide de la programmation ? Quelle méthode est la plus efficace afin d'obtenir le matériel nécessaire à l'analyse ?

### **1.3. Présentation de l'organisation hôte**

Hydro-Géo Environnement a été fondée en janvier 2016 après la fusion entre HydroGéo Conseils et Hydro-Geol. Cette entreprise possède de nombreux pôles de compétences techniques, qui touchent autant l'hydrogéologie, l'environnement, la géothermie de faible à grande profondeur, la géologie d'exploration, la géotechnique et l'énergie.

Hydro-Géo Environnement concentre notamment ses activités en Suisse romande et en France voisine. Dans le cadre de ses prestations en géothermie, elle propose son expertise en géologie et géophysique, en environnement, en ingénierie de forage, en ingénierie de production, en gestion de projets, en hydrogéologie, en ingénierie des réservoirs, en géologie d'opérations et suivi de forage et en énergie.

Parmi ses activités liées à la géothermie, le Dr. Jérôme Chablais, associé gérant responsable de la géologie et de la géothermie profonde au sein de Hydro-Géo Environnement, supervise le stage de Clothilde Jost, étudiante en géomatique, pour la réalisation d'une étude sur la caractérisation structurale des linéaments morpho-tectoniques de la Haute-Chaîne du Jura suisse à l'aide d'outils SIG et de programmation, en particulier QGIS et Python.

David Polasek, spécialiste en géologie structurale, analyse de fracturation, champs de contraintes et tendance aux cisaillements des failles et le Dr. Maud Brentini, experte en lithostratigraphie, en prospection géothermique et en cartographie SIG, apportent également leur expertise respective à ce projet.

### **1.4. Exposé du sujet**

Lors d'études structurales traditionnelles, l'analyse de terrain relève les mesures liées à différentes structures de déformation comme, les failles, les fractures et les stylolites (Peacock & Sanderson, 2018; Polasek, 2020; Radaideh & Mosar, 2021). Pour les analyses topologiques, un ou plusieurs affleurements géologiques de taille variant de plusieurs dizaines de centimètres à quelques mètres et représentatifs des familles de linéaments trouvées sur le terrain sont sélectionnés et photographiés. Les fractures sont ensuite systématiquement relevées sur la zone choisie, et les relations topologiques entre elles sont analysées, soit in situ, soit à partir des clichés récoltés sur le terrain (Palamakumbura et al., 2020). Pour autant



qu'un affleurement représentatif soit trouvé, ces méthodes de relevé sont très précises et fiables. Elles sont cependant très chronophages, et ne permettent pas d'analyses sur de grandes zones. C'est pourquoi l'identification des linéaments à l'aide de SIG est de plus en plus utilisée afin de permettre la collecte de données sur un plus grand secteur (Ahmadi & Pekkan, 2021; Jaboyedoff et al., 2010, 2018; Montoya, 2018; Radaideh & Mosar, 2021; Scheiber et al., 2015; Tanner et al., 2020; Wolfe et al., 2020). La détection des fractures à l'aide de MNT constitue une très bonne façon de réaliser des études structurales préliminaires à des études de prospection. Toutefois, il est important de préciser que les données obtenues par télédétection doivent impérativement être vérifiées par des études de terrain (Polasek, 2020).

Les réseaux de linéaments digitalisés à l'aide de MNT, comme ceux obtenus sur le terrain, peuvent être traités pour obtenir les relations topologiques entre les différentes fractures. Sanderson et Nixon ont notamment défini une méthode pour analyser un réseau de fractures quantitativement et permettre de lier les données topologiques aux propriétés de connectivité et de potentiel de percolation (Sanderson & Nixon, 2015, 2018).

Peacock and Sanderson (2018) ont également décrit les sept « piliers » de l'analyse structurale, qui sont : (1) les descriptions géologiques de base, (2) les analyses géométriques et topologiques, (3) les relations chronologiques entre les fractures, (4) la cinématique de déformation, (5) la tectonique, (6) la mécanique de déformation, et (7) l'écoulement des fluides.

Ainsi, le présent projet sur l'analyse géomatique et structurale de linéaments morpho-tectoniques sur la Haute-Chaîne du Jura suisse propose d'étudier la place des techniques SIG et pour la digitalisation des linéaments et l'analyse des géométries et de la topologie de ces derniers.

Après une recherche des outils préexistants sur QGIS, une évaluation de la faisabilité de la digitalisation automatique des linéaments à partir de MNT a été réalisée à l'aide du plugin Network GT. Cette digitalisation automatisée sur une large zone d'étude (section 3.3.2, Fig. 5)), a été abandonnée au profit de la digitalisation manuelle sur une plus petite zone, le Locle (Fig. 4). Ensuite, un plugin a été créé pour faciliter l'importation et la préparation des données nécessaires à la digitalisation manuelle des linéaments (Fig. 8, annexes 1, 2, 3, 4 & 5).

Une fois les outils et workflows maîtrisés, il a été décidé d'appliquer les procédures SIG investiguées à une zone d'étude en cours par HGE, Le Locle. La digitalisation (Fig. 9) des linéaments morpho-tectoniques autour du Locle (Fig. 1 & 13) a été réalisée à l'aide des outils de digitalisation QGIS et une méthode d'analyse des linéaments obtenus a été élaborée à partir des méthodes de Sanderson et Nixon (2015, 2018) et de Nyberg et al. (2018) afin d'évaluer la géométrie et la topologie de ceux-ci dans le cadre d'une étude préliminaire la prospection géothermique au Locle (Fig. 10).

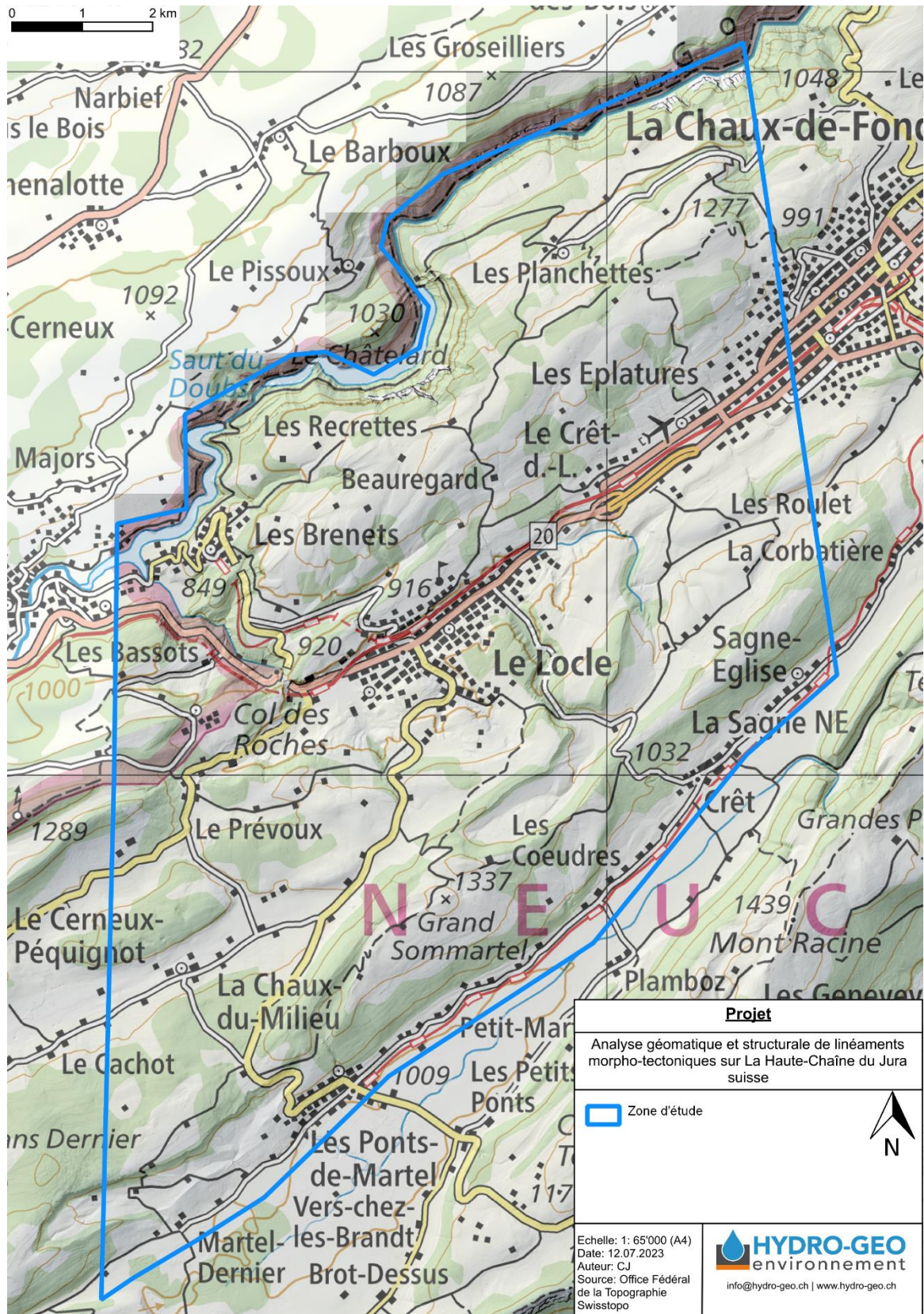


Figure 1 : Carte montrant la zone d'étude en bleu sur fond de carte de l'Office Fédéral de la Topographie Swisstopo.



## 2. Généralités

### 2.1. Télédétection et technologie LiDAR

La télédétection est un ensemble de techniques qui permettent de caractériser à distance des objets à l'aide d'images, de points et d'anomalies. Ces objets incluent notamment des structures géologiques et géomorphologiques (Scheiber et al., 2015). Il existe plusieurs façons de récolter des données, active ou passive, par avion, par drones, ou encore par satellite (Tanner et al., 2020).

La technologie de télédétection aéroportée active *light detection and ranging* ou LiDAR, permet de générer, par balayage laser, un nuage de points caractérisé par des coordonnées en 3 dimensions dans un système de coordonnées cartésien (Ahmadi & Pekkan, 2021; Jaboyedoff et al., 2010, 2018). Ces nuages de points permettent de créer des modèles de terrain avec une grande précision altimétrique (10 cm) (*Acquisition de données LiDAR*, 2023).

Depuis 2017 et jusqu'à 2024, l'Office Fédéral de topographie Swisstopo réalise une campagne d'acquisition de données LiDAR sur tout le territoire Suisse (*Acquisition de données LiDAR*, 2023). Ces données font partie de celles qui ont été utilisées pour la digitalisation des linéaments.

### 2.2. Contexte géologique et structural régional de la Haute Chaîne du Jura et du Locle

La Haute-Chaîne du Jura se trouve à l'ouest du bassin d'avant-pays alpin, au sein du massif du Jura, une structure géologique de type *Fold-and-Thrust Belt* (« chaîne de plis et chevauchements », Marro et al., 2023). La chaîne du Jura s'étend sur près de 350 km et prend une forme arquée, entourée par le bassin sédimentaire de la Molasse à l'est, le graben oligocène du Rhin au nord et le graben de la Bresse à l'ouest (Sommaruga et al., 2017). Le terme Haute-Chaîne se réfère à la partie interne, surélevée du Jura, caractérisée par des plis majeurs orientés entre E-O et N-S. Cette partie interne s'oppose à la partie externe du Jura, ou Jura tabulaire, constituée de plateaux et de zones de failles (« faisceaux ») (Sommaruga et al., 2017).

La zone d'étude, incluant le Locle, se trouve entièrement dans la partie centrale de la Haute-Chaîne (Fig. 3, Eichenberger et al., 2020). Elle est traversée par plusieurs anticlinaux orientés NE-SO et ENE-OSO (Fig.3, Eichenberger et al., 2020). À l'affleurement (Fig. 2), on retrouve une succession stratigraphique allant du Jurassique moyen ( $174,1 \pm 1 - 163,5 \pm 1$  Ma, Cohen et al., 2013) jusqu'au Miocène ( $23,03 - 5,333$  Ma, Cohen et al., 2013) avec une lacune de 100 millions d'années entre le Barrémien ( $\sim 130$  Ma, Cohen et al., 2013) et le Burdigalien (20, 44 Ma, Cohen et al., 2013; Eichenberger et al., 2020).



0 1 2 km

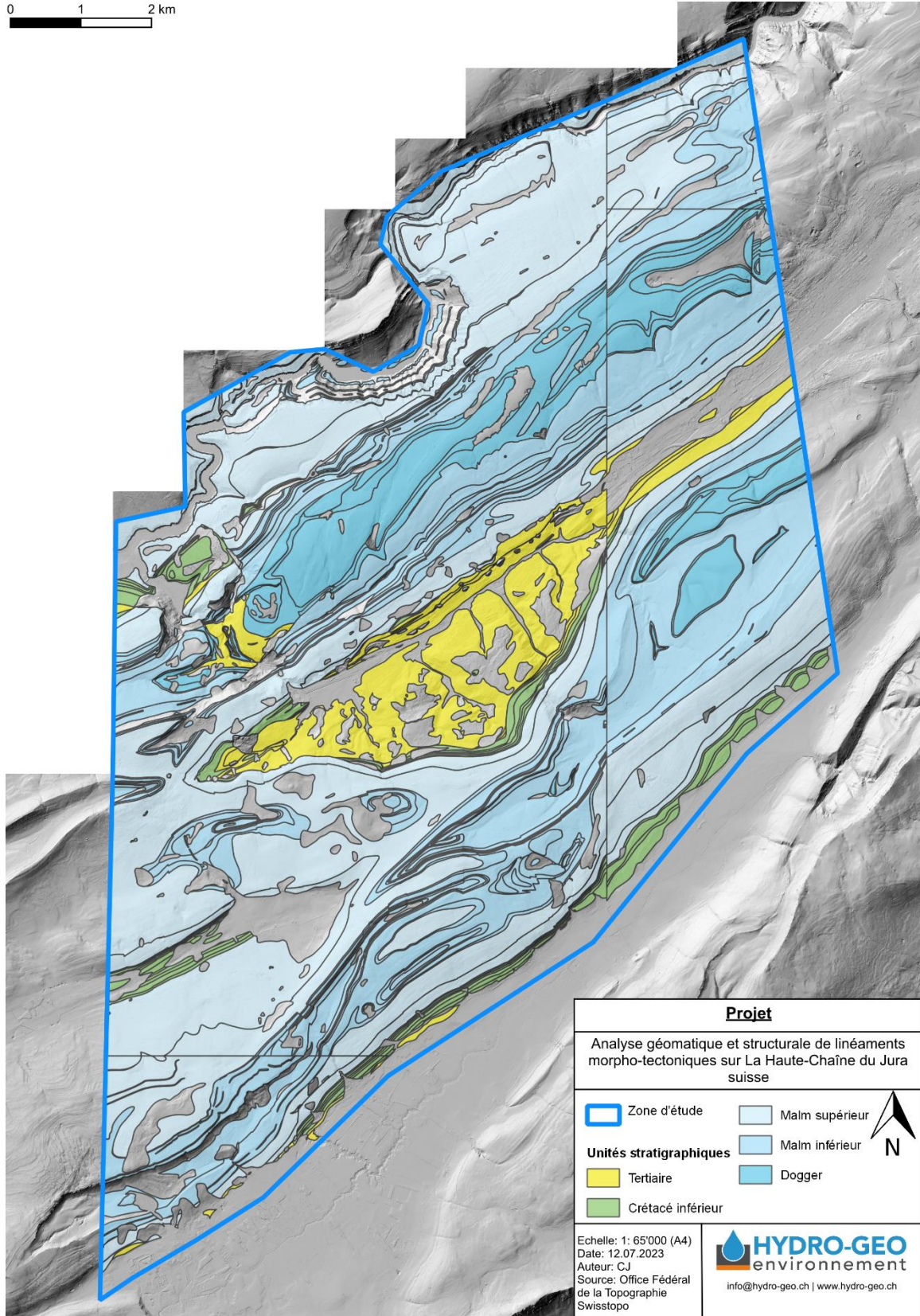


Figure 2 : Carte montrant les unités stratigraphiques affleurantes sur la zone d'étude, modifiée d'après la couche GeoCover V3 (Office Fédéral de la Topographie Swisstopo). Le fond de carte est la couche swissALTI3D relief monodirectionnel de l'Office Fédéral de la Topographie Swisstopo.







### 3. Méthodologie géomatique ad hoc

#### 3.1. Présentation et caractérisation des données de départ

Tout le projet a été réalisé sur QGIS 3.18.1 with GRASS 7.8.5, dans le système de coordonnées de référence (SCR) suisse, soit CH1903+ / LV95. Toutes les données ont été téléchargées ou créées directement dans ce système de coordonnées. Le cadre de référence de ce système de coordonnées, le cadre MN95, est celui qui a été utilisé pour ce projet.

Afin de pouvoir réaliser l'analyse, un shapefile contenant le polygone de la zone d'étude a été créé (Fig. 4), avec les outils de digitalisation basiques disponibles sur la version QGIS du projet. Ce shapefile a permis par la suite de redimensionner les données à la taille de la zone d'étude lorsque nécessaire.

##### 3.1.1. Données rasters

Les tuiles MNT swissALTI3D sont la base de cette étude. Elles proviennent de l'Office Fédéral de la topographie swisstopo :

(<https://www.swisstopo.admin.ch/en/geodata/height/alti3d.html#download>)

Ces tuiles ont été téléchargées selon l'emprise du polygone de la zone d'étude et directement dans le SCR CH1903+ / LV95. L'utilisateur peut choisir entre deux résolutions : 2 et 0.5 [m]. Pour ce projet, une résolution de 0.5 [m] a été sélectionnée. L'exportation se fait sous forme de liens de téléchargement des tuiles au format XYZ ZIP contenus dans un fichier CSV, qu'il faut ensuite extraire à l'aide de Windows PowerShell avec les commandes suivantes :

```
$ProgressPreference = 'SilentlyContinue'
```

```
gc .\file.csv | % {iwr $_ -outf $(split-path $_ -leaf)}
```

On obtient ainsi des fichiers XYZ ZIP pour chaque tuile de la zone, que l'on peut ensuite dézipper dans un dossier. Elles sont ensuite prêtes à être importées dans le projet QGIS.

##### 3.1.2. Données vectorielles

Pour l'analyse, le fichier shapefile des linéaments digitalisé devra contenir dans sa table attributaire des informations liées à la géologie, contenues dans les données GeoCover V3. Ces données proviennent également de swisstopo :

(<https://www.swisstopo.admin.ch/fr/geodata/geology/maps/geocover.html#download>)

Leur échelle est au 1 : 25 000, dans le SCR CH1903+ / LV95. Le téléchargement se fait tuile par tuile selon les divisions de cartes de l'Atlas Géologique Suisse au 1 : 50'000 ; celles correspondant à la zone d'étude sont les suivantes : 172 Le Locle, 51 Val-de-ruz, 162 travers, 15 Les bois.

Lors de la digitalisation, il est nécessaire de différencier d'éventuels linéaments d'infrastructures humaines telles que des routes, des chemins pédestres ou encore des lignes de téléphérique. C'est pourquoi, la couche shapefile swissTNE Base (réseau transport) a également été téléchargée sur le site de swisstopo dans le système de coordonnées du projet



(<https://www.swisstopo.admin.ch/fr/geodata/landscape/tne.html>) afin de la contraindre aux linéaments interprétés.

### 3.1.3. Données en ligne (WMS)

Certaines données n'avaient pas besoin d'être téléchargées hors de QGIS, puisque swisstopo met à disposition une partie de ses données via les *Web Mapping Services* (WMS) de l'Infrastructure fédérale de données géographiques (IFDG). Il suffit alors d'établir la connexion WMS directement depuis QGIS avec le lien suivant : <https://wms.geo.admin.ch/?LANG=fr>. À l'aide de la connexion WMS – IFDG, les couches suivantes ont été consultées et utilisées comme fond de carte pour plusieurs mises en page :

- SWISSIMAGE Fond de plan (résolution 10 [cm])
- swissALTI3D relief monodirectionnel (résolution 2 [m])
- Cartes nationales (couleur, 1 : 10'000 à 1 : 1 million)
- Cartes nationales (grise, 1 : 10'000 à 1 : 1 million)
- Hydrologie swissTLMRegio (Précision 20-60 [m])

## 3.2. Présentation et caractérisation des plugins QGIS utilisés

Plusieurs outils SIG ont été créés spécialement dans l'optique de digitaliser et d'analyser des linéaments morpho-tectoniques. Cependant, tous ne fonctionnent pas sur QGIS et tous ne sont pas open source. Parmi ceux-ci, on retrouve notamment le LINE module du paquet Geomatica v12, logiciel propriétaire payant de Aeromap Technology Systems, qui permet de digitaliser automatiquement les linéaments (Radaideh et al., 2016). Le set d'outils FracPaQ permet d'analyser de façon simultanée l'orientation et la position spatiale d'une fracture et de calculer la perméabilité d'un set de fractures (Healy et al., 2017). Il ne fonctionne cependant que sur Matlab.

Sur QGIS il existe trois sets d'outils open source, avec leurs spécificités propres : Network GT, Monte Carlo Slip Statistics Toolkit (MCSST) et GeoTrace. Le premier inclut des outils permettant de digitaliser automatiquement les linéaments, des outils d'échantillonnage, d'analyse géométrique et topologique et de propriétés des flux de fluides (Nyberg et al., 2018). Il ne fonctionne cependant que sur les versions de QGIS allant de 3.1 à 3.99. Le deuxième permet d'estimer le déplacement des blocs séparés par des failles le long de transects (Wolfe et al., 2020). Le dernier, GeoTrace, est un outil d'aide à la digitalisation de linéaments. Une phase de prise en main et d'essais de chacun de ces plugins a été menée en début de stage afin d'isoler les outils qui seraient appliqués sur la zone d'étude du Locle. GeoTrace et MCSST n'ont ainsi pas été utilisés dans cette étude, car le premier ajoute des erreurs de digitalisation et ne raccourcit pas le temps de digitalisation et le second propose des outils qui ne sont pas relevant pour cette étude. C'est le plugin Network GT qui a été choisi pour le traitement des linéaments digitalisés, moyennant un travail sur QGIS 3.18.1 with GRASS 7.8.5.

Pour la réalisation du plugin de ce projet (Fig. 8, annexes 1,2,3,4 & 5), nommé Lineament Analysis, d'autres plugins ont été nécessaires : Plugin Builder 3 et Plugin Reloader. Plugin



Builder 3 permet de créer les fichiers et dossiers de base pour la création d'un plugin (Fig. 30, QGIS Project, 2020) et Plugin Reloader permet de recharger un plugin pendant sa réalisation sans devoir redémarrer QGIS (QGIS Project, 2020).

Enfin, les plugins Geospatial Data Abstraction Library (GDAL) et System for Automated Geoscientific Analyses (SAGA), utiles pour les traitements de données raster et vectoriel, ont été également utilisés.

### 3.3. Étapes de travail – workflow

#### 3.3.1. Préparation des données pour l'identification des linéaments

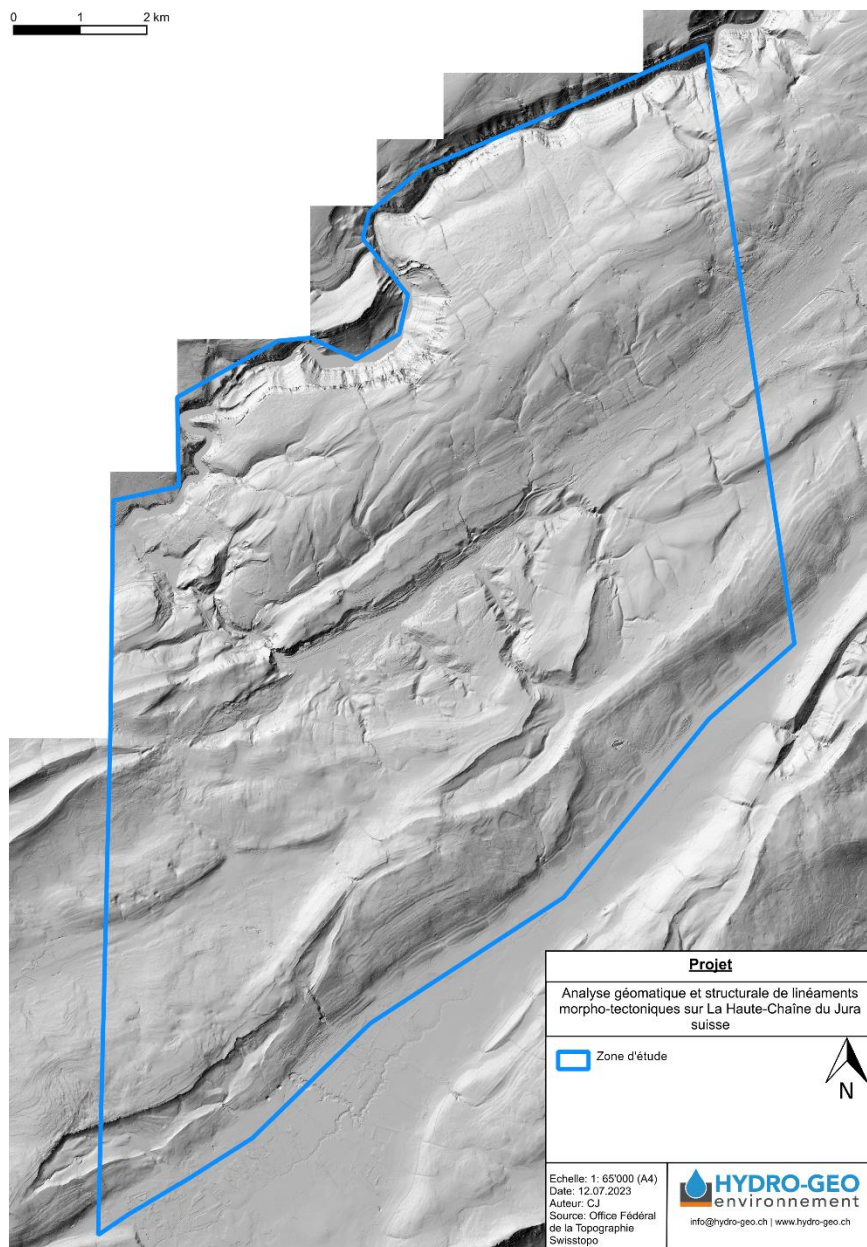


Figure 4 : Polygone de la zone d'étude sur fond de swissALT13D relief monodirectionnel de l'Office Fédéral de la Topographie Swissstopo.



La première étape du *workflow* de ce projet consiste à créer un shapefile contenant le polygone de la zone d'étude (Fig. 4). Dans ce cas précis, la zone d'étude a été dessinée pour englober le périmètre d'une étude de potentiel géothermique en cours chez Hydro-Géo Environnement, ciblée sur le Locle ainsi qu'une partie du périmètre d'une étude de potentiel géothermique future de la Chaux-de-Fonds. L'aire d'intérêt de ce travail reste cependant restreinte, afin de contraindre le temps de digitalisation des linéaments en lien avec la durée du stage. Une extension de l'aire d'intérêt sera menée ultérieurement par le bureau.

Pour créer ce shapefile, une nouvelle couche shapefile, avec le système de coordonnées CH1903+ LV95, a été créée avec QGIS puis le polygone de la zone d'étude a été dessiné avec l'outil « ajouter une entité polygonale ».

Les données nécessaires à l'analyse (détaillées en section 3.1) ont été ensuite importées dans le projet QGIS. Des données contenues dans GeoCover V3, seules les couches « bedrock » (shapefile polygon contenant les polygones de surface correspondant aux lithologies affleurantes) et « linear object » (shapefile contenant les failles) ont été importées pour chaque tuile.

L'outil « fusionner les couches vecteur » a été utilisé pour créer un shapefile unique pour le *bedrock* et les *linear objects* respectivement, qui ont ensuite été découpés aux dimensions de la zone d'étude en utilisant l'outil « Découper » et le shapefile du polygone de la zone d'étude comme couche de superposition.

La couche shapefile swissTNE Base a également été découpée à l'étendue de la zone d'étude avec l'outil « Découper », de façon similaire aux shapefiles GeoCover V3.

En amont du développement du plugin Lineament Analysis (le sujet de ce travail ; description complète en section 3.3.3), il a été nécessaire d'importer la totalité des tuiles MNT afin de pouvoir les fusionner en utilisant l'outil fusion du plugin GDAL. Le raster fusionné a ensuite été coupé avec l'outil Clip raster with polygon du plugin SAGA aux dimensions du polygone de la zone d'étude.

Une fois cette étape réalisée, le projet QGIS a été sauvegardé en tant que projet contenant les données brutes et un nouveau projet a été créé dans lequel les tuiles de MNT non fusionnées et les couches GeoCover V3 et swissTNE Base non découpées à l'étendue de la zone d'étude ont été retirées afin de réduire le temps de traitement et de chargement du projet.

Six ombrages ont été réalisés sur le raster MNT fusionné (annexes 6, 7, 8, 9, 10 & 11), afin de faire ressortir les linéaments. L'outil « ombrage » permet d'illuminer de façon artificielle un MNT en choisissant l'angle d'incidence et l'orientation de la source de lumière. Les fractures orientées perpendiculairement à la lumière ressortent mieux (Scheiber et al., 2015). Dans ce contexte, afin de faire ressortir des fractures avec des orientations différentes, des sources de lumière avec des directions azimutales (premier nombre à 3 chiffres) et des inclinaisons différentes (deuxième nombre à 2 chiffres) ont été déterminées : NE 045/45, SE 135/45, SO 225/45, NO 315/45, NE 045/25, SO 225/25.



L'angle d'incidence à 45° permet de faire ressortir la plupart des structures (Scheiber et al., 2015) et l'angle de 25°, rasant, utilisé pour les orientations perpendiculaires aux structures dominantes, permet de faire ressortir les structures manquantes (Scheiber et al., 2015).

Ces données vont ensuite être utilisées pour la digitalisation des linéaments, décrite en section 3.3.4.

### 3.3.2. Essai d'automatisation de l'identification des linéaments morpho-tectoniques

Avant de se restreindre à la zone étendue du Locle, le projet initial était de réaliser une digitalisation automatique des linéaments sur une zone plus grande, qui s'étendait de la faille de Vallorbe-Pontarlier (au NE du lac de Joux) jusqu'à Soleure, en incluant Yverdon-les-Bains et la Chaux-de-Fonds (Fig. 5). Cette grande zone était subdivisée en 8 sous-zones (Fig. 5). Le téléchargement des données pour cette zone a été réalisé de façon identique à la méthode présentée en section 3.1, à l'exception de la zone d'intérêt et de la résolution des tuiles MNT swissALTI3D (2 [m] au lieu de 0,5 [m]). Les tuiles MNT swissALTI3D ont d'abord été téléchargées pour trois des sous-zones : Vallorbe, Pontarlier et Yverdon, afin de limiter la quantité de données à télécharger dans un premier temps.

La digitalisation automatique aurait permis de couvrir une très grande zone, étant donné que la partie la plus chronophage de l'analyse de linéaments morpho-tectoniques est la digitalisation manuelle. Afin de pouvoir utiliser l'outil « Digitize Fracture Network » du plugin Network GT, qui permet de digitaliser automatiquement les linéaments à partir d'une image raster binaire, il est nécessaire de transformer en premier lieu les ombrages en rasters binaires.

Afin de transformer les rasters d'ombrage en raster binaires et faire ressortir les structures principales, un calcul de seuil a été réalisé, à l'aide de l'outil « Raster Calculator ». Le principe est de définir une valeur de seuil pour chacun des ombrages afin de faire ressortir certaines structures. Les pixels en dessous de la valeur de seuil se voient attribuer la valeur de 0 et apparaissent en noir, et ceux au-dessus de la valeur seuil se voient attribuer la valeur de 1 et apparaissent en blanc (Fig. 6).

Les rasters résultants sont les couches d'entrée de l'outil « Digitize Fracture Network » de Network GT. L'outil effectue une *skeletonization* de l'image binaire et la convertit en shapefile contenant des lignes. Cependant, après de multiples essais qui consistaient à changer la valeur de seuil pour créer le raster binaire (valeurs de seuil de pixel testées pour chaque ombrage : 100, 110, 120, 150), variant ainsi le niveau de détail des structures mises en évidence, les fichiers de sortie de Digitize Fracture Network étaient inutilisables en l'état à cause de la création d'« artefacts » de digitalisation, qui génèrent des linéaments avec de nombreuses sinuosités (similaire à la Fig. 7). En effet, lors d'une analyse de linéaments, ce sont des lignes droites qui sont préférées afin d'obtenir une meilleure mesure d'orientation de ces derniers (Peacock & Sanderson, 2018; Polasek, 2020).

Un autre outil, « Raster skeletonization », au sein du plugin SAGA, a été testé pour digitaliser les linéaments (Fig. 7). Cependant le problème restait le même qu'avec l'outil de Network GT.

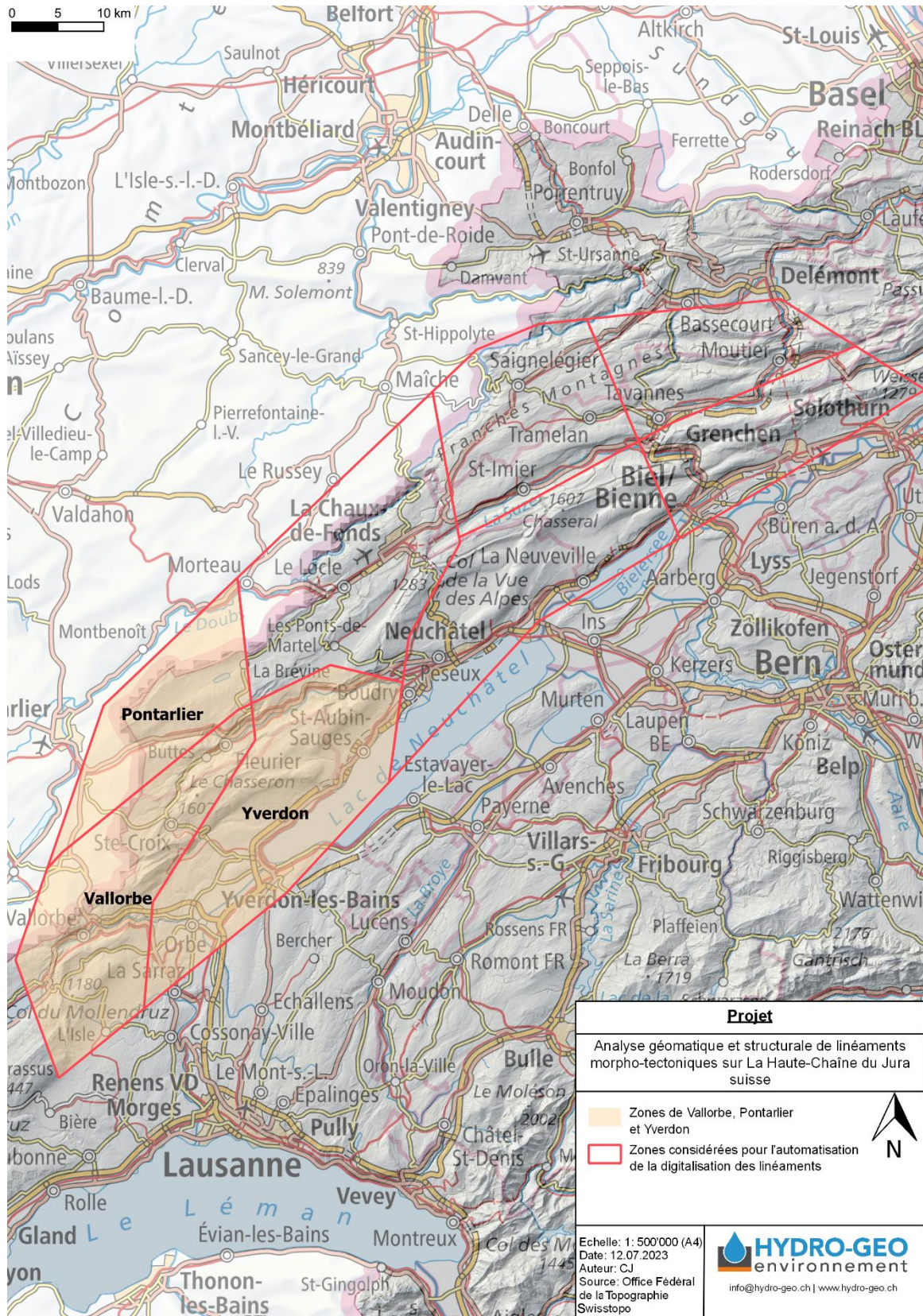
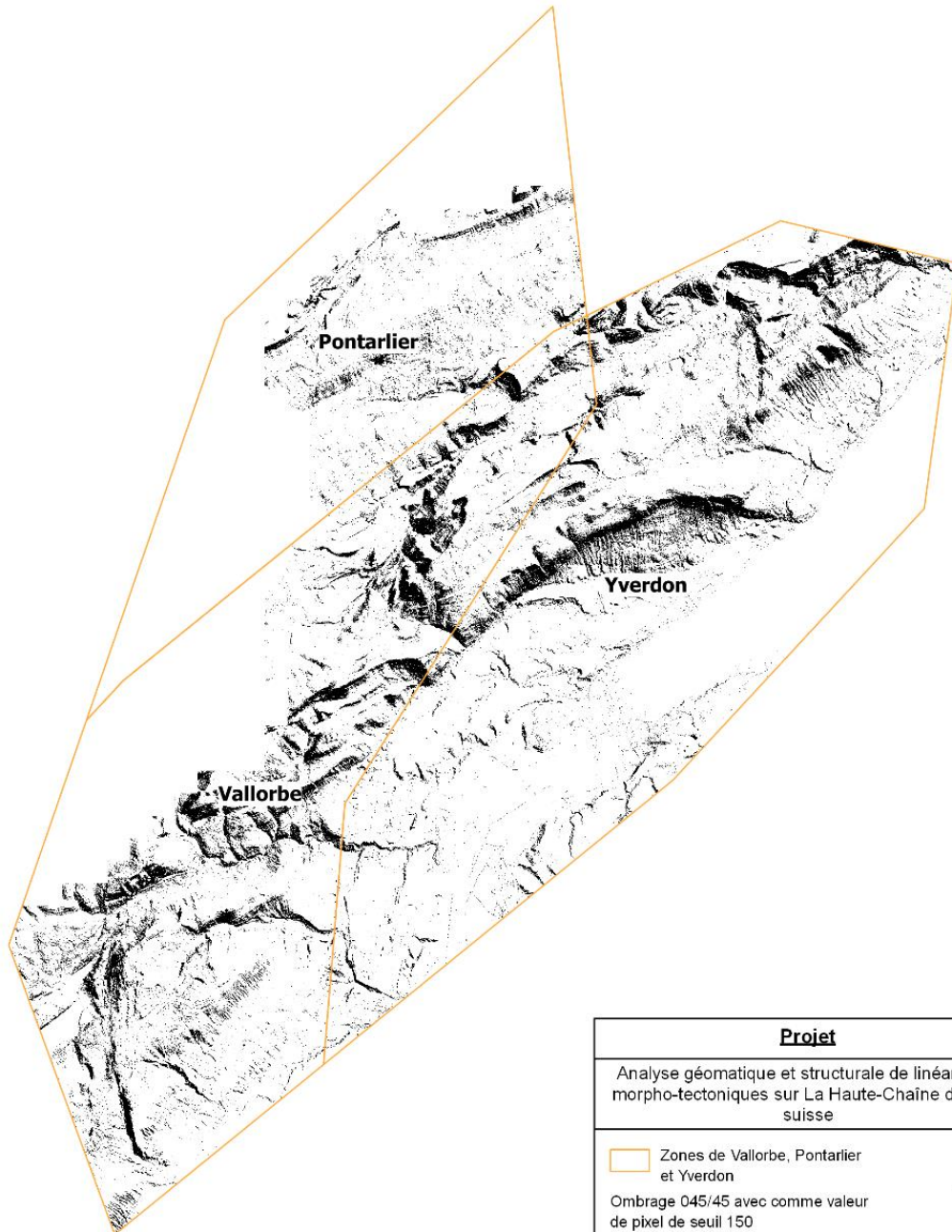


Figure 5 : Carte montrant les 8 zones considérées pour réaliser la digitalisation automatique. Les zones de Vallorbe, Pontarlier et Yverdon sont mises en évidence, car les essais ont été réalisés sur ces dernières. Les fonds de carte choisis sont le swissALTI3D relief monodirectionnel et les Cartes Nationales (couleurs) de l'Office Fédéral de la Topographie Swisstopo.







<b>Projet</b>	
Analyse géomatique et structurale de linéaments morpho-tectoniques sur La Haute-Chaine du Jura suisse	
 Zones de Vallorbe, Pontarlier et Yverdon	 N
Ombrage 045/45 avec comme valeur de pixel de seuil 150	
	
Echelle: 1: 250'000 (A4) Date: 12.07.2023 Auteur: CJ	 info@hydro-geo.ch   www.hydro-geo.ch

Figure 6 : Exemple de résultat de l'un des calculs de seuil pour l'ombrage 045/45, avec comme valeur de pixel de seuil 150.

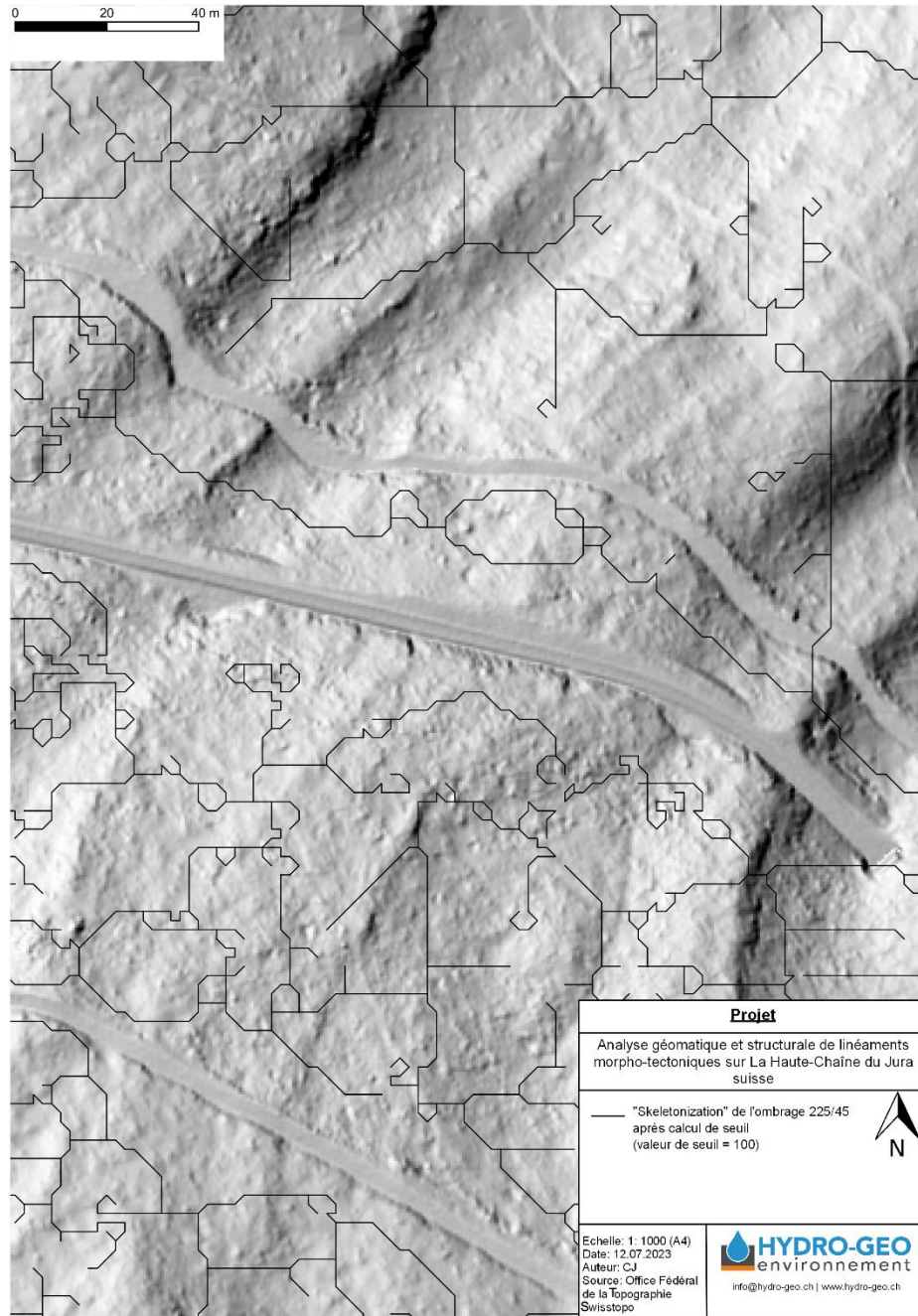


Figure 7 : Carte montrant le résultat de « Raster skeletonization » du plugin SAGA sur l'ombrage 225/45, zoomé de près afin de montrer les « artefacts » de digitalisation. Avant d'utiliser l'outil « skeletonization », l'ombrage 225/45 a été transformé en raster binaire en réalisant un calcul de seuil, ici avec une valeur de seuil de 100. Le résultat est similaire à l'outil « Digitize Fracture network » du plugin Network GT. Le fond de carte choisi est le swissALTI3D relief monodirectionnel de l'Office Fédéral de la Topographie Swisstopo.

Un essai de simplification du « squelette » obtenu afin d'atténuer les artefacts a été réalisé à l'aide de l'outil « Simplifier » de QGIS. Deux algorithmes de simplification ont été testés : Douglas-Peucker et Visvalingam-Whyatt. Le premier parcourt itérativement les sommets d'une polyligne et en réduit le nombre en identifiant les sommets les plus éloignés d'un segment de ligne selon une valeur de tolérance (*How Simplify Line and Simplify Polygon work—ArcGIS Pro | Documentation, 2023*). Le deuxième parcourt aussi itérativement les sommets pour en réduire le nombre, mais il identifie les sommets à retirer en calculant l'aire





triangulaire entre un vertex et ses voisins et enlève les sommets avec la plus petite aire selon une valeur de tolérance (*How Simplify Line and Simplify Polygon work—ArcGIS Pro | Documentation, 2023*). Même en testant plusieurs valeurs de tolérance, la simplification prend beaucoup trop de temps (plus d'une heure) et le résultat reste inutilisable pour analyser la géométrie et la topologie des linéaments.

Les études réalisées dans la littérature qui ont réussi une digitalisation automatique des linéaments ont été conduites dans des zones propices à celle-ci. Il s'agit de zones où les linéaments sont facilement identifiables sur les MNT, avec peu de végétation ou autre couverture du sol et des reliefs nets (ex : Arabie Saoudite ou Jordanie, Ahmadi & Pekkan, 2021; Radaideh et al., 2016 respectivement). Aussi, Salui (2018) a réalisé une étude comparative de digitalisation automatique des linéaments sur des MNT et des images Landsat-8 OLI de la même zone. L'étude a conclu que la digitalisation réalisée à partir des images satellites était meilleure que celle réalisée à partir des MNT, mais que dans les deux cas, le taux de détection était insatisfaisant (Salui, 2018).

Au vu des résultats obtenus et des conclusions partagées par la littérature, il a été décidé d'abandonner la digitalisation automatique et de réduire la zone d'étude à la zone d'étude du Locle et de ses alentours décrite en section 3.3.1.

### 3.3.3. Création du plugin QGIS Lineament Analysis pour la préparation des tuiles MNT – automatisation et optimisation Python

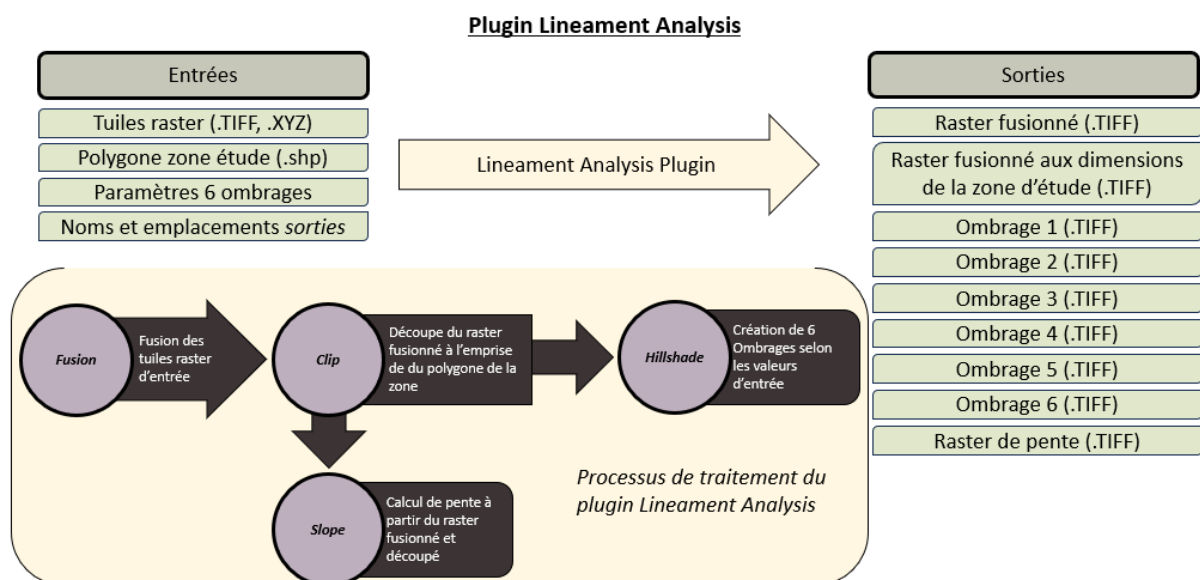


Figure 8 : Schéma de processus de traitement du plugin Lineament Analysis.

Le processus de préparation des tuiles MNT pour permettre la digitalisation des linéaments comporte un gros problème d'optimisation : l'importation d'un très grand volume de données raster dans QGIS et le travail de celles-ci afin d'obtenir un seul MNT englobant toute la zone de travail. Il est également nécessaire de créer un second projet QGIS une fois les données traitées, pour alléger le projet et optimiser le temps des phases de traitement. À titre informatif, pour la zone retenue du Locle, il a été nécessaire d'importer 148 tuiles, totalisant 16,9 Go de données.



La raison principale de la création de ce plugin est donc d'améliorer la préparation de données nécessaire à la digitalisation manuelle des linéaments, en automatisant le processus de fusion des tuiles MNT et en réalisant les ombrages (annexes 6, 7, 8, 9, 10 & 11) et un calcul de pente, sans avoir à importer les tuiles dans le projet QGIS afin de gagner du temps. Le plugin créé pour cela, sur QGIS 3.18.1 with GRASS 7.8.5 et Python 3.7, a été nommé Lineament Analysis (Fig. 8, annexes 1, 2, 3, 4 & 5) et la description complète des étapes de réalisation ainsi que les versions de tous les plugins et programmes nécessaires à son fonctionnement sont disponible dans l'annexe 1.

### 3.3.4. Digitalisation semi-automatique des linéaments morpho-tectoniques sur la zone du Locle à l'aide du plug-in NetworkGT

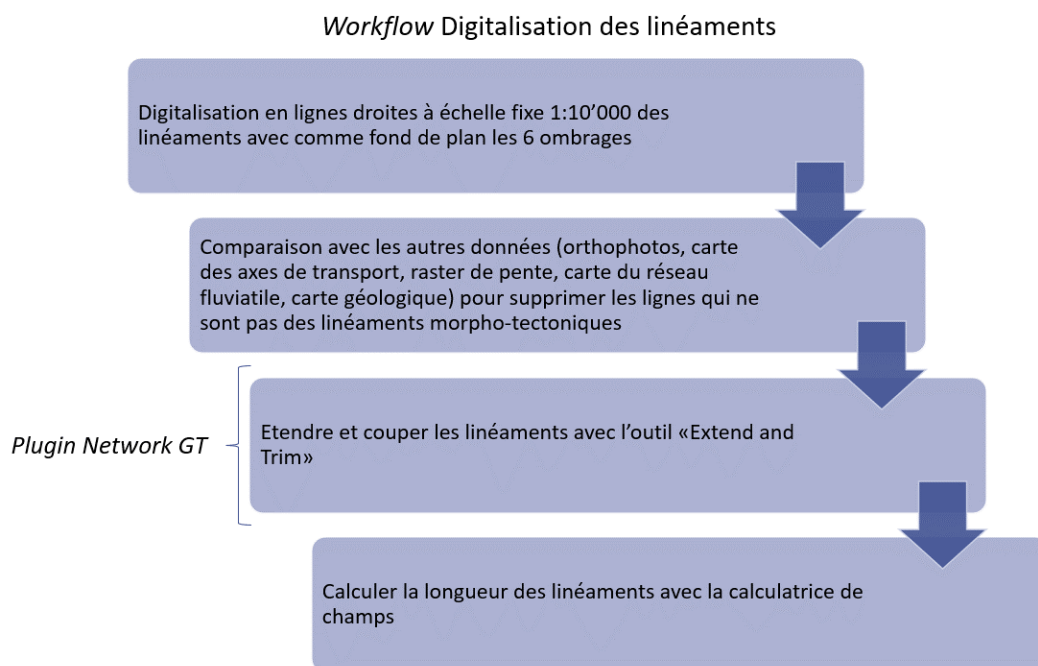


Figure 9 : Étapes de digitalisation des linéaments

Une fois les données préparées selon les étapes décrites précédemment, les linéaments morpho-tectoniques ont été digitalisés selon la méthode décrite sur la figure 9. Plusieurs paramètres peuvent biaiser l'interprétation des linéaments, notamment la variation d'échelle, l'illumination et la variabilité de l'opérateur, qu'il faut prendre en compte lors de la réalisation de cette étape (Scheiber et al., 2015).

En effet, en fonction de l'échelle, les linéaments qui ressortent ne seront peut-être pas les mêmes. Afin de contrer ce biais, les linéaments du Locle ont été digitalisés à une échelle fixe au 1 : 10'000, déterminée en observant le niveau de détail du MNT fusionné à disposition.

Le biais d'illumination est observé lorsque seulement une ou deux orientations d'ombrage sont utilisées par l'opérateur pour identifier les linéaments. Dans cette étude, pour minimiser ce biais, 6 ombrages ont été utilisés pour identifier les linéaments (annexes 6, 7, 8, 9, 10 & 11).

En outre, les linéaments obtenus varient du fait de l'expérience dans la digitalisation de linéaments et du domaine d'études de l'opérateur (Scheiber et al., 2015). Ils varient aussi en



fonction du temps alloué à l'interprétation (Scheiber et al., 2015). Plus qu'un biais, il s'agit de l'interprétation personnelle de chaque opérateur, et il est difficile d'harmoniser cet aspect. Peacock et al. (2019), proposent une méthodologie afin de minimiser les variations dues à l'opérateur : extraire des lignes droites, qui permettent une meilleure mesure de l'orientation, mais qui ne rendent pas compte de la réelle morphologie du linéament, ou travailler par cadres à échelle fixe, en commençant par une plus petite échelle pour interpréter les grandes structures d'abord et changer au minimum 3 fois d'ombrage par échelle (Peacock & Sanderson, 2018). Dans ce projet, comme la digitalisation des linéaments n'était pas le sujet principal, il a été décidé de ne digitaliser qu'à l'échelle du 1 : 10 000, mais d'utiliser 6 ombrages différents. Des lignes droites ont été extraites.

Concrètement, à échelle fixe, les linéaments ont été digitalisés sur l'ensemble de la zone d'étude en parcourant l'ensemble de la zone pour chaque ombrage. Ensuite, les linéaments ont été comparés à l'orthophotographie de la couche SWISSIMAGE afin de ne pas prendre en compte les infrastructures humaines. Dans la même optique, ils ont aussi été comparés à la couche swissTNE Base (réseau transport) pour exclure les chemins et les routes et à la couche Hydrologie swissTLMRegio pour exclure les rivières. Dans ce dernier cas, certaines rivières ont tout de même été gardées, les rivières se mettant souvent en place dans des zones de faille existantes. Les linéaments digitalisés ont été par la suite comparés à la carte géologique GeoCover V3 afin de vérifier que certaines fractures n'étaient en fait pas des thalwegs d'érosion, qui suivent les limites des couches géologiques.

À l'aide du plugin Network GT, le shapefile des linéaments digitalisés a été ensuite amélioré pour garantir un bon recouplement des linéaments et une bonne topologie. L'outil Extend and Trim a été utilisé deux fois : une première pour étendre les linéaments jusqu'aux linéaments voisins s'ils étaient à moins de 30 [m] puis une deuxième fois afin de supprimer les « excédents » correspondant à des erreurs de digitalisation, lorsqu'ils faisaient moins de 10 [m] de longueur.

Afin de pouvoir réaliser la suite des analyses, un nouveau champ a été calculé dans la table attributaire du shapefile des linéaments avec la calculatrice de champ : le champ de longueur des linéaments, calculé avec la formule suivante : `length($geometry)`.



### 3.3.5. Analyse géométrique et topologique des linéaments à l'aide de NetworkGT

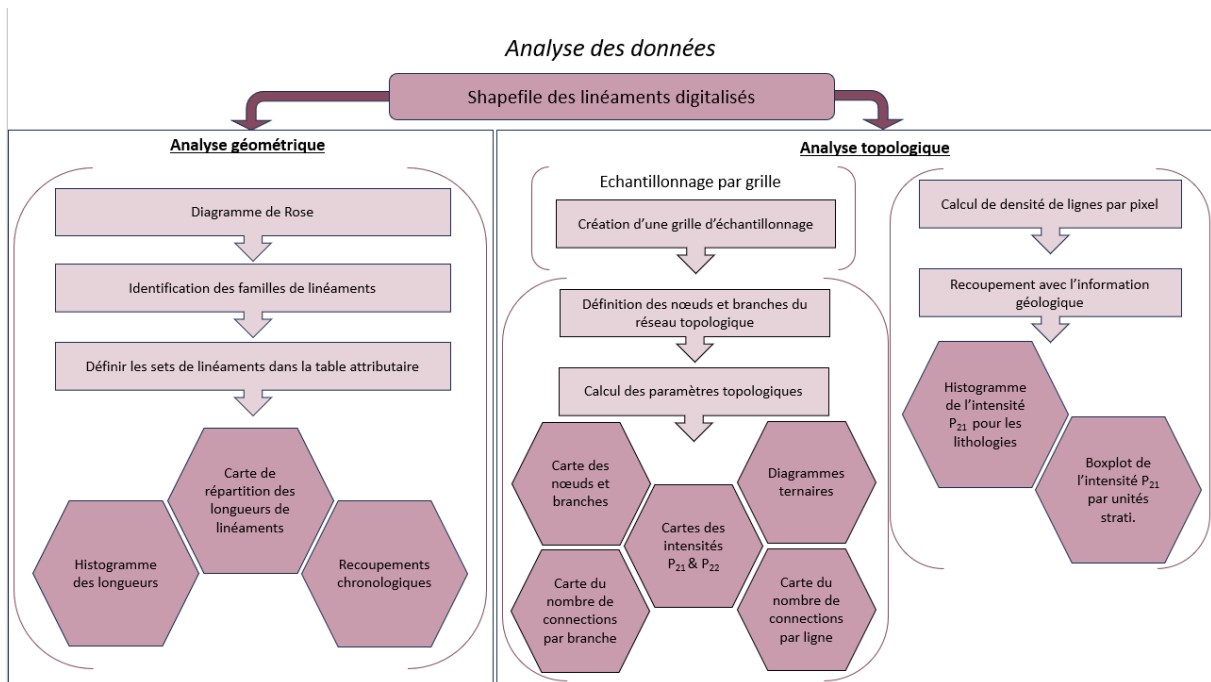


Figure 10 : Étapes d'analyse des linéaments une fois qu'ils ont été digitalisés.

#### 3.3.5.1. Analyse géométrique

L'analyse géométrique (Fig. 10) consiste à extraire les informations géométriques des linéaments, comme leur orientation, la répartition de leur longueur et la famille à laquelle ils appartiennent.

Un diagramme de Rose pondéré par la longueur des linéaments et d'aire égale, pour limiter les biais visuels lors de la détermination des familles (Sanderson & Peacock, 2020), a été créé avec l'outil « Rose Diagram » de Network GT (Fig. 14). La longueur des linéaments provient du champ de longueur créé auparavant. Parmi les paramètres de l'outil, la taille de *bins* choisie est de 5 degrés et l'angle réciproque a également été représenté.

Après avoir identifié les familles (= set) de linéaments en interprétant le diagramme de Rose, l'outil « Define sets » permet d'ajouter un champ dans la table attributaire contenant la famille auquel le linéament appartient. Les trois familles identifiées sont : 165 à 30 degrés, 30 à 100 degrés et 100 à 165. L'outil ne prend en compte que les orientations entre 0 et 180 degrés. Lorsqu'un set est à cheval sur le 0 et/ou sur le 180, il faut saisir d'abord la valeur la plus grande (Nyberg et al., 2018).

La table attributaire respective a été exportée au format CSV afin de pouvoir réaliser un histogramme de la fréquence de la longueur des linéaments sur Excel.

#### 3.3.5.2. Échantillonnage par grille

Pour réaliser les analyses de topologie, il est nécessaire de définir une grille d'échantillonnage sur la zone d'étude. Pour cela, il existe un outil dans le plugin Network GT : « Contour Grid ». La taille des mailles de la grille choisie pour la zone d'étude est de 500 [m] x 500 [m], qui semblait être un bon compromis entre niveau de détail et temps de traitement (Fig. 11).



0 1 2 km

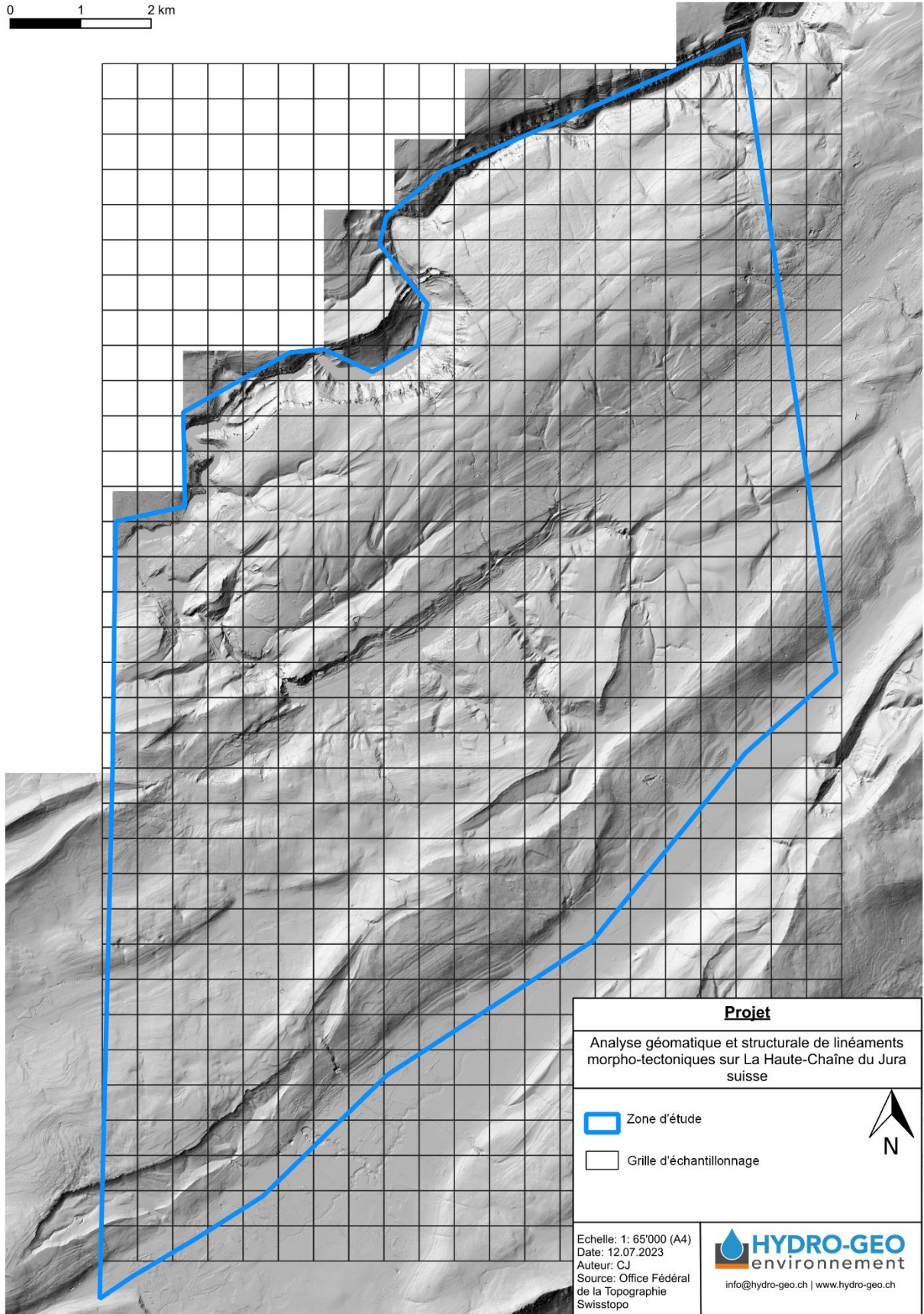


Figure 11 : Grille d'échantillonnage avec des mailles de 500 x 500 [m] et zone d'étude (en bleu), sur fond de swissALTI3D relief monodirectionnel de l'Office Fédéral de la Topographie Swisstopo.



### 3.3.5.3. Analyse topologique

L'analyse topologique est l'étude des types de connexions entre les linéaments. Elle permet de mieux comprendre la connectivité d'un système de fractures en 2D et n'est pas influencée par la déformation. Un système est généralement composé de nœuds, endroits où se rejoignent des fractures, et de branches, segments de linéament séparés par des nœuds. Il y a trois types de nœuds : les nœuds I, isolés en bout de branche, les nœuds X, d'où partent 4 branches et les nœuds Y, d'où partent trois branches (Fig. 12). Il y a également une terminologie correspondante pour les branches, qui est liée aux types de nœuds à chaque extrémité, soit les branches I-I (« isolées » avec deux nœuds I), I-C (« semi-connectées » avec un nœud I et un nœud X ou Y) et C-C (« connectées », sans nœud I) (Sanderson & Nixon, 2015).

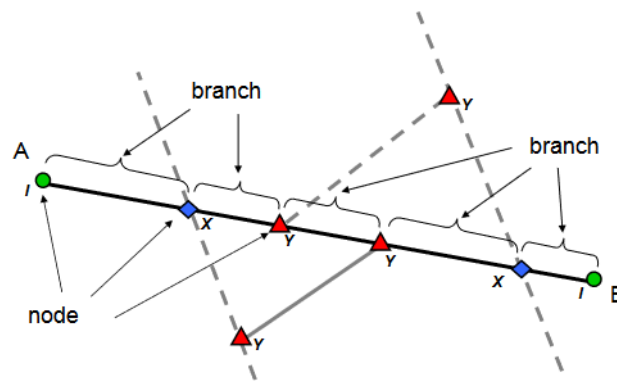


Figure 12 : Schéma montrant les différentes terminologies liées à la topologie : les nœuds « I » sont les cercles verts, les nœuds « X » sont les carrés bleus, les nœuds « Y » sont les triangles rouges et les branches sont les segments de linéaments entre les nœuds (Sanderson et Nixon, 2015).

Afin de déterminer les types de nœuds et de branches des linéaments du Locle, l'outil « Branches and Nodes » de Network GT a été utilisé (Fig. 18). Ce dernier prend en couche d'entrée le shapefile des linéaments et la grille d'échantillonnage de la section 3.3.5, ce qui permettra de calculer des paramètres pour les mailles de la grille et de représenter les relations topologiques sous forme de grille dans la section 4.

La proportion des types de nœuds et des types de branches est généralement représentée par des diagrammes ternaires (section 4.3.2, Fig. 19 & 20), qui ont été calculés et représentés à l'aide de l'outil « Topology Parameters » du plugin Network GT. Cet outil crée une nouvelle couche shapefile à partir des deux fichiers créés par l'outil « Branches and Nodes », avec une table attributaire contenant les informations liées à la topologie.

Les paramètres qui ont été choisis pour caractériser les linéaments du Locle sont l'intensité  $P_{21}$  (mesure de la densité de linéament ou intensité 2D :  $P_{21} = \frac{\sum \text{longueur des linéaments}}{\text{Aire}}$  (Sanderson & Nixon, 2015), Fig. 23), l'intensité  $P_{22}$  (intensité sans dimension :  $P_{22} = P_{21} \times \text{Longueur moyenne des branches}$  (Nyberg et al., 2018), Fig. 24), le nombre de connexions par ligne (Fig. 22) et le nombre de connexions par branche (Fig. 21). Tous ces paramètres sont calculés par l'outil « Topology Parameters ».

Afin de pouvoir comprendre l'impact du type de lithologie et de pouvoir apprécier dans quelles unités stratigraphiques les fractures sont principalement situées en surface, l'intensité  $P_{21}$  a été calculée par pixel sur la zone d'étude puis recoupée avec les lithologies présentes



dans la couche GeoCover V3 (Fig. 25 & 26). La première étape a consisté à calculer la densité de lignes  $P_{21}$  à l'aide de l'outil « Line density », avec des pixels de 100 [m] et un rayon de recherche de 250 [m]. Ainsi chaque pixel contient la densité de lignes dans un rayon de 250 [m]. Ensuite la couche de points obtenue a été recoupée avec les lithologies avec l'outil « Intersection ». Une table contenant le résumé de l'intensité  $P_{21}$  pour chaque lithologie a été exportée avec l'outil « Statistiques par catégorie » sous forme de fichier CSV, à partir duquel un histogramme de l'intensité  $P_{21}$  moyenne en fonction de la formation géologique (Fig. 25) et un boxplot de l'intensité  $P_{21}$  en fonction des unités stratigraphique (Fig. 26) ont été créés.



## 4. Résultats

### 4.1. Linéaments identifiés et digitalisés dans la zone du Locle

Au total, 661 linéaments ont été identifiés et digitalisés (Fig. 13) sous forme de polygones contenues dans un shapefile. Ils ont été digitalisés sur la zone d'étude à échelle fixe, au 1 : 10'000, à partir des 6 ombrages (annexes 6 à 11) décrits en section 3.3.1 et selon la méthode décrite sur la figure 9.

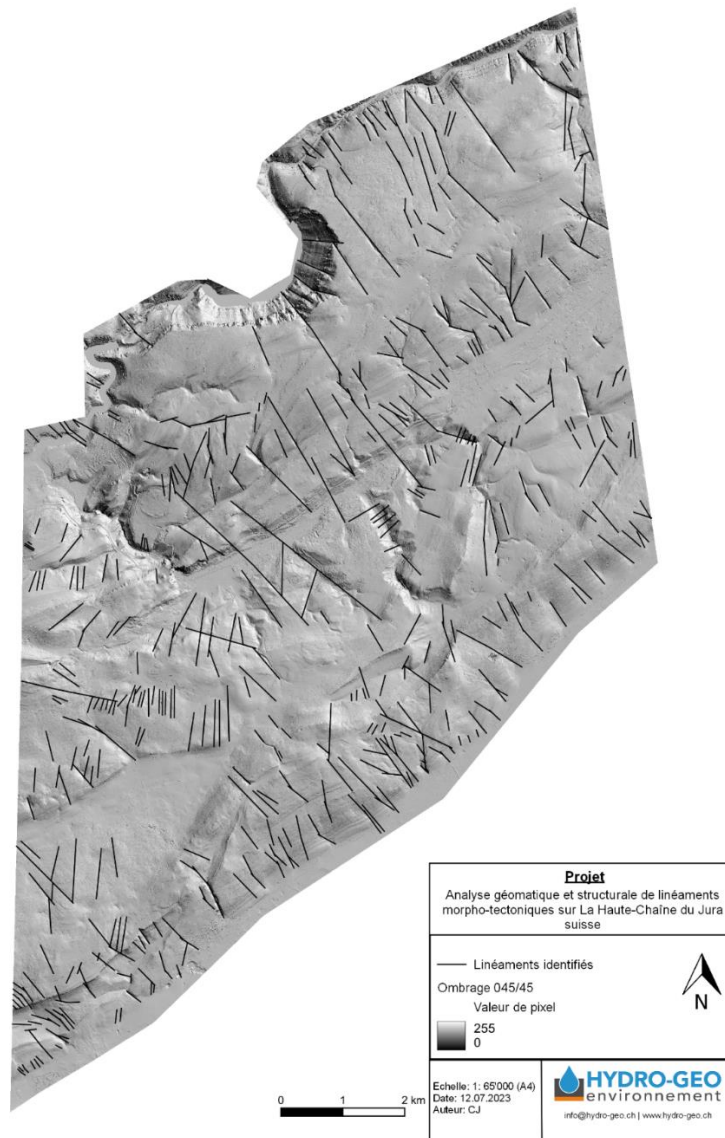


Figure 13 : Carte des linéaments digitalisés manuellement sur la zone d'étude du Locle à l'échelle 1 : 10'000, en suivant le processus présenté sur la figure 9.

### 4.2. Résultats de l'analyse géométrique




#### 4.2.1. Diagramme de Rose

Le diagramme de Rose pondéré par les longueurs des linéaments permet de répartir les linéaments en familles selon leur « strike azimut » (orientation de leur trace en surface). Le





diagramme de Rose des linéaments du Locle (Fig. 14) met en évidence trois familles : une allant de N100° à N165°, une allant de N165° à N30° et une allant de N30° à N100°.

-  L2 (N165° - N30°) : N-S
-  L3 (N30° - N100°) : NE-SO
-  L1 (N100° - N165°) : NNO-SSE

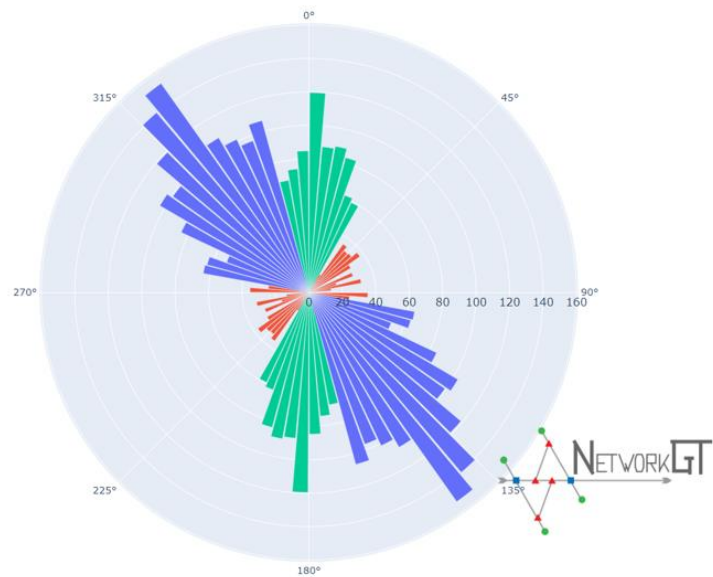


Figure 14 : Diagramme de Rose pondéré par la longueur des linéaments et d'aires égales, avec comme taille de « bins » 5 degrés. Le diagramme est généré automatiquement par la fonction « Rose Diagram » du plugin Network GT.

#### 4.2.2. Recouvrements chronologiques et familles de linéaments

Les relations de recouvrements entre les familles de linéaments permettent de comprendre dans quel ordre chronologique elles se sont mises en place (Gasparo Morticelli et al., 2017). Cependant, sans indicateurs cinématiques récoltés sur le terrain, les relations chronologiques entre les linéaments de la zone du Locle ne sont pas univoques. Ainsi, la nomenclature choisie pour les familles de linéaments n'est pas basée sur un ordre chronologique, mais sur l'angle d'orientation, avec le plus de linéaments pour la famille L1 orientée NNO-SSE, puis la famille L2, orientée N-S et enfin la famille L3, orientée NE-SO. Les linéaments colorisés selon leur appartenance à une famille sont visibles sur la figure 15.



0 1 2 km

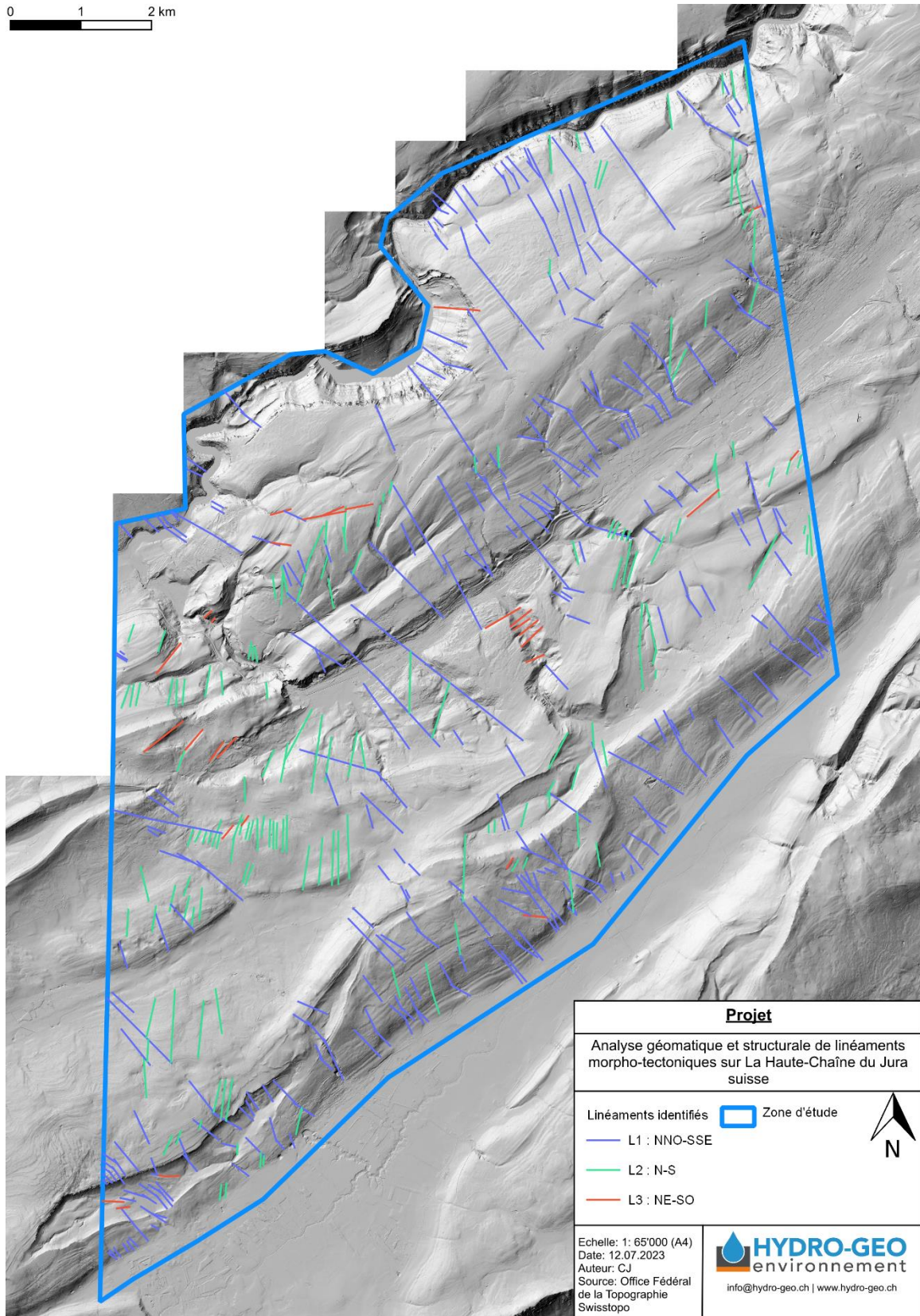


Figure 15: Carte montrant les linéaments digitalisés colorisés selon leur famille d'orientation d'appartenance L1 (bleu), L2 (vert) ou L3 (rouge). Les couleurs sont les mêmes que sur le diagramme de Rose présenté en figure 14. Le fond de carte est le swissALTI3D relief monodirectionnel provenant de l'Office Fédéral de la Topographie Swisstopo.



### 4.2.3. Répartition des longueurs de linéaments

Les longueurs de linéaments digitalisés varient entre 10,07 [m] et 2238,5 [m] (Fig. 16 & 17). La longueur moyenne des linéaments toutes familles confondues est de 318 [m] et l'écart-type est de 271 [m] (Tableau 1). En moyenne, la famille L2 a les plus longs linéaments et la famille L3 a les plus petits (Tableau 1). Les linéaments de la famille L1 montrent la plus grande variabilité et des linéaments en moyenne presque aussi longs que ceux de la famille L2 (Tableau 1). L'histogramme de la figure 16 permet de visualiser la distribution de taille des linéaments globale et la figure 17 permet de voir la répartition de la taille des linéaments sur la zone d'étude.

Tableau 1 : Propriétés de taille des linéaments par familles

Famille de linéament	Longueur moyenne [m]	Ecart-type [m]
L1 & L2 & L3	318	271
L1	316	298
L2	326	215
L3	283	162

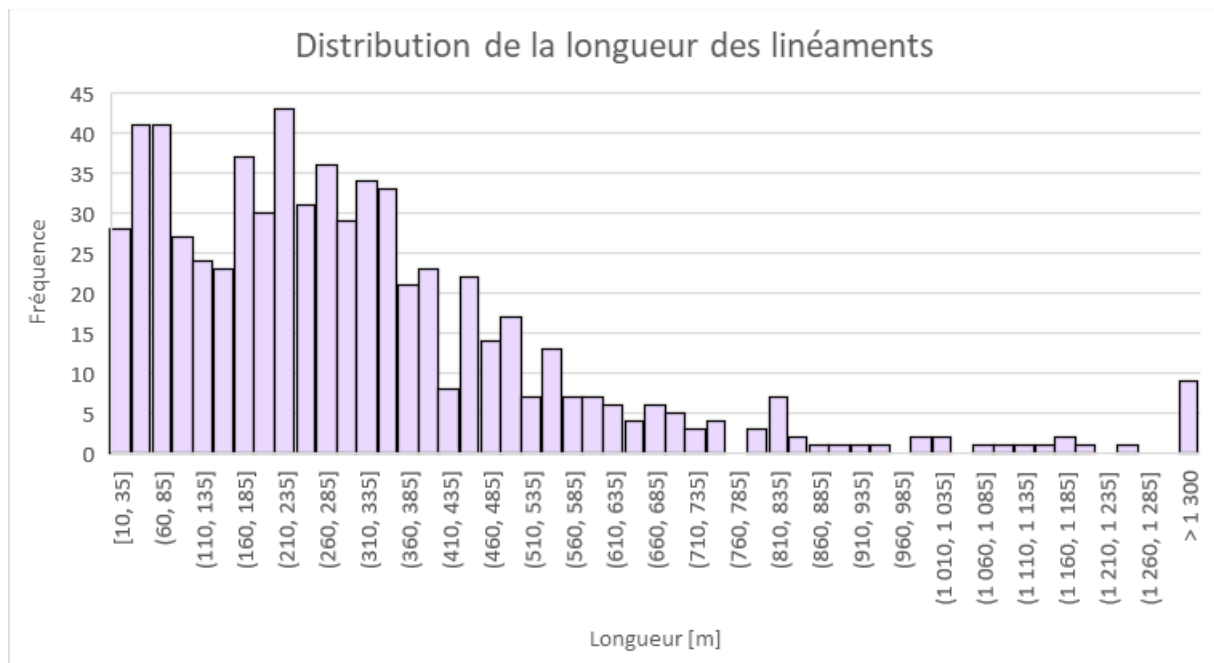


Figure 16 : Histogramme de la distribution de la longueur des linéaments sur la zone d'étude du Locle.



0 1 2 km

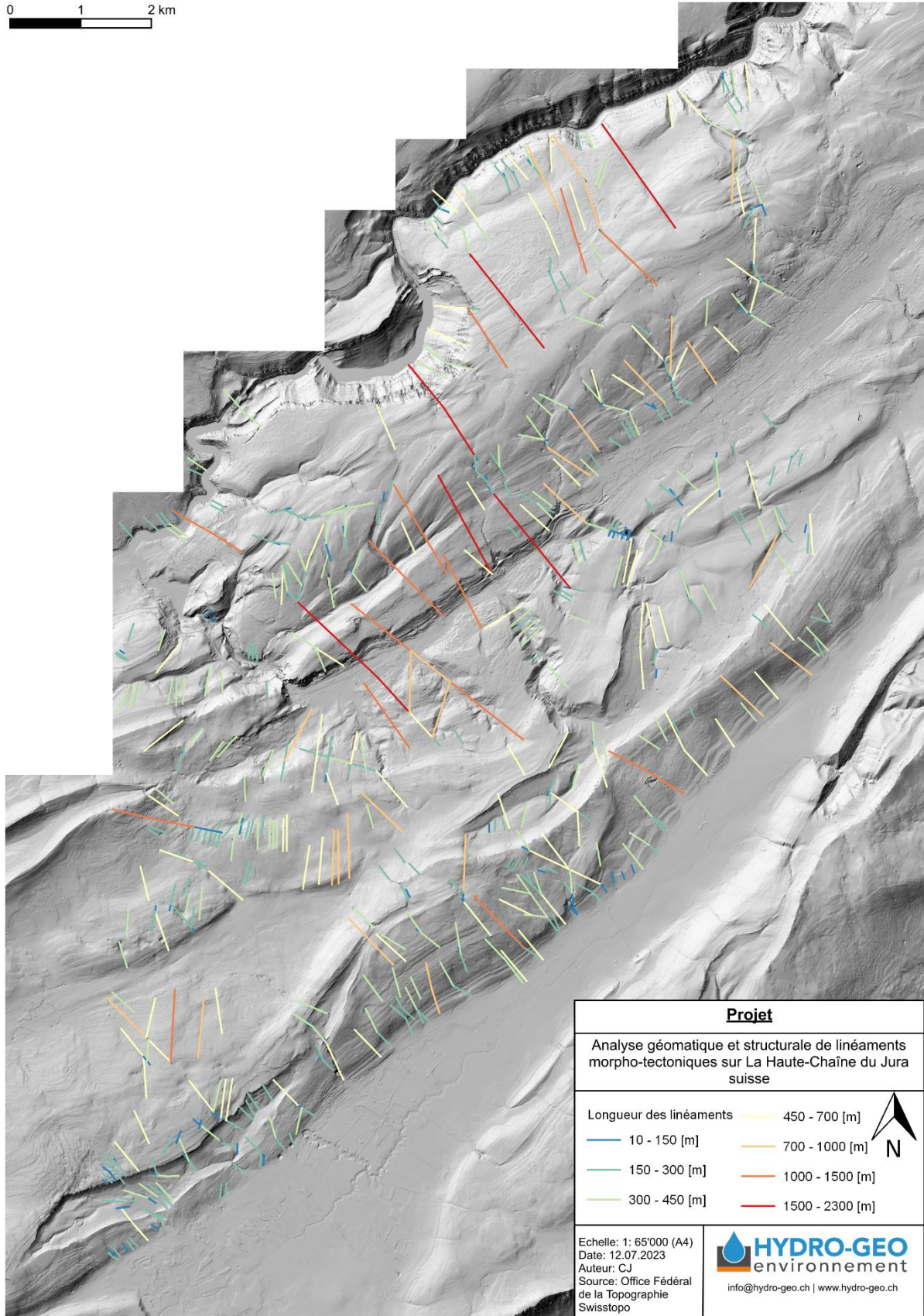


Figure 17 : Carte montrant les linéaments colorisés selon leur longueur, sur fond de swissALTI3D relief monodirectionnel de l'Office Fédéral de la Topographie Swisstopo.



### 4.3. Résultats de l'analyse topologique – Propriétés topologiques

#### 4.3.1. Nœuds et branches

L'outil « Nodes and Branches » de Network GT a permis de classifier les nœuds du réseau de linéament digitalisé pour la zone d'étude (Tableau 2). Les nœuds isolés « I » représentent la grande majorité des types de nœuds. Les branches ont été regroupées selon les appellations de Sanderson et Nixon (2015) (Tableau 2, Fig. 18). La carte de la figure 18 montre les nœuds et les branches générés par l'outil « Nodes and Branches ». On remarque que les branches C – C, qui ont une meilleure connectivité, sont localisées au centre et au S-SE de la zone d'étude. Les zones recherchées pour la valorisation géothermique de moyenne profondeur sont celles avec une bonne connectivité.

Tableau 2 : Récapitulation topologique

Type de noeuds		Nombre
X		29
Y		454
I		2533
Type de branche	Appellations Sanderson et Nixon (2015)	Nombre
X - X	C - C	13
X - Y	C - C	17
I - X	C - I	75
Y - Y	C - C	214
I - Y	C - I	1081
I - I	I - I	799

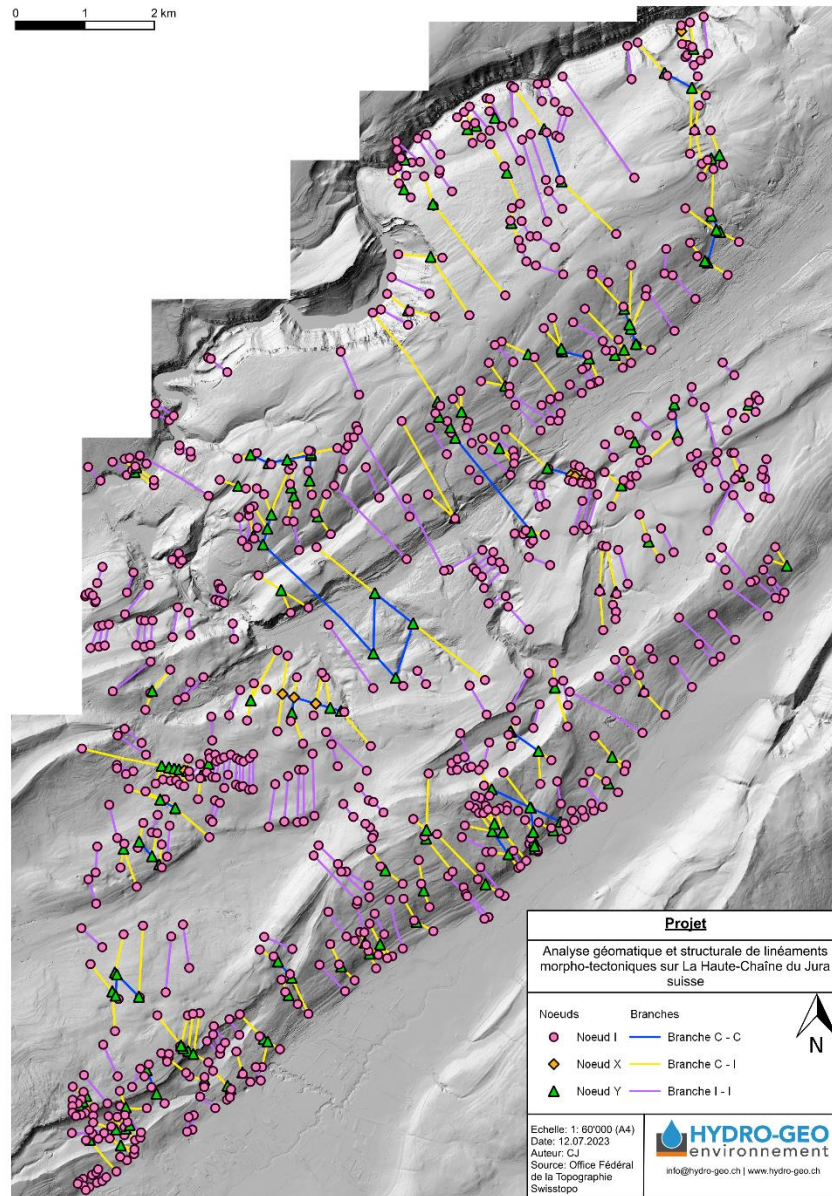


Figure 18 : Carte montrant le résultat de l'outil « Branches and Nodes » du plugin Network GT. Les trois types de nœuds I, X et Y apparaissent sous forme de cercle rose, losange orange et triangle vert respectivement. Les branches correspondantes C-C, C-I et I-I apparaissent en bleu, jaune et rose respectivement. Le fond de carte est le swissALT3D relief monodirectionnel provenant de l'Office Fédéral de la Topographie Swisstopo.

#### 4.3.2. Diagrammes ternaires

Les diagrammes ternaires permettent de représenter les proportions de types de nœuds (Fig. 19) et les proportions de types de branches (Fig. 20) dans chacune des mailles de la grille d'échantillonnage décrite en 3.3.5.2. Sur les deux diagrammes sont aussi représentées le nombre de connexions par branche théorique en fonction des proportions de nœuds ou de branches (Nyberg et al., 2018; Sanderson & Nixon, 2015). La valeur de 1.56 représente le seuil de percolation des fluides dans un réseau de fractures en 2D (Sanderson & Nixon, 2015) qui n'est atteint que pour 4 zones disjointes de la grille (Fig. 19). Sur la figure 19, il apparaît que la plupart des mailles ne possèdent pas de nœuds X. Sur la figure 20, on voit qu'il y a une bonne proportion de branches C – I dans le réseau de linéaments du Locle.

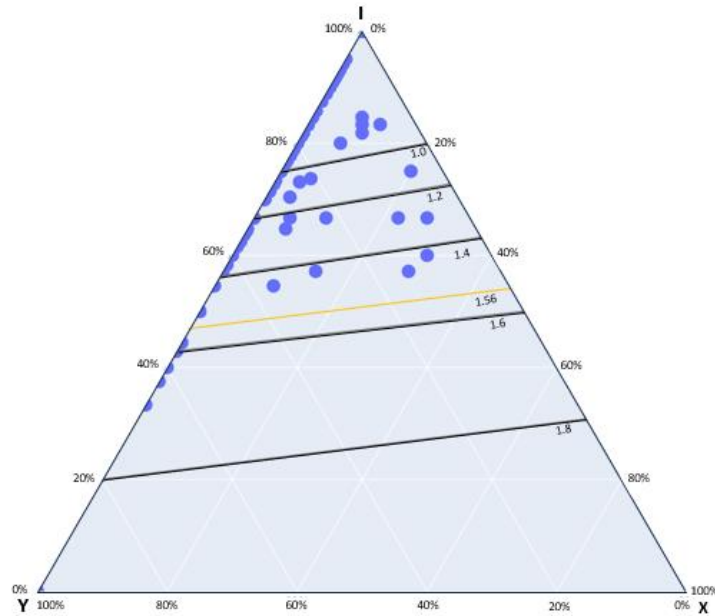


Figure 19 : Diagramme ternaire montrant la proportion des types de nœuds « X », « Y » et « I ». Les lignes noires et jaunes montrent le nombre de connexions par branche correspondant aux proportions de types de nœud.

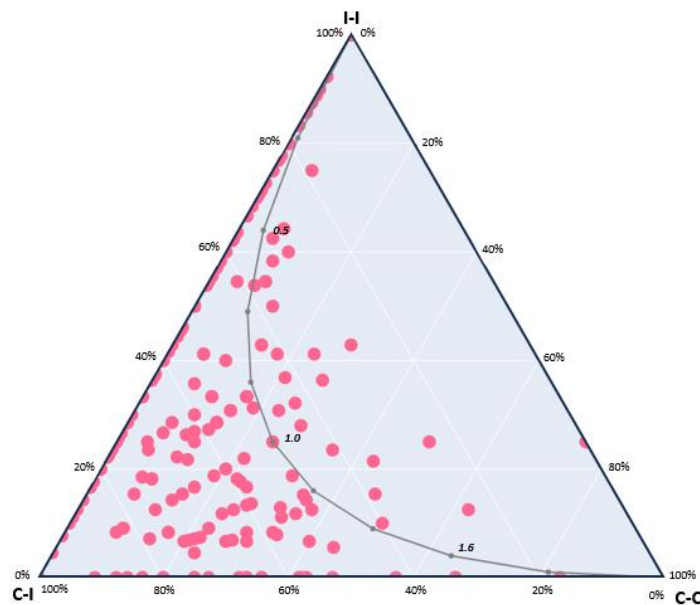


Figure 20 : Diagramme ternaire montrant les différents types de branches et la courbe du nombre de connexions par branche correspondant aux proportions de types de branches.

### 4.3.3. Nombre de connexions par branche

Le nombre de connexions par type de branche de chaque maille de la grille d'échantillonnage peut être représenté sur une carte (Fig. 21). Le nombre de connexions par type de branche est un bon indice pour comprendre la connectivité du réseau de linéaments (Nyberg et al., 2018; Sanderson & Nixon, 2015).



La valeur de seuil de 1.56, défini par Sanderson et Nixon (2015) est le seuil de connectivité des branches permettant la percolation des fluides dans un réseau de fractures en 2D. On remarque sur la figure 21 que 4 zones atteignent ce seuil.

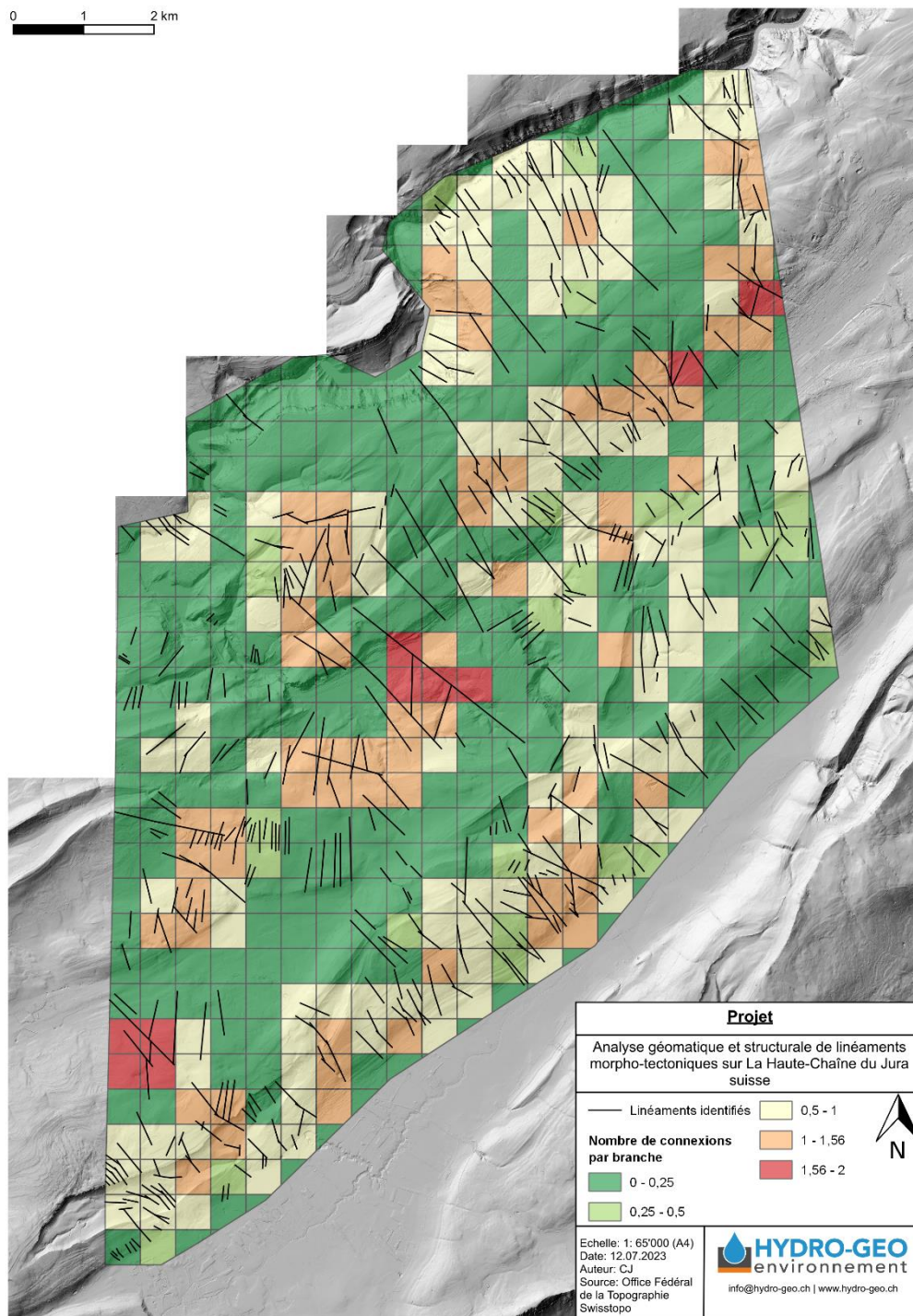


Figure 21 : Carte montrant le nombre de connexions par branches pour chaque maille de la zone d'échantillonnage, sur fond de swissALTI3D relief monodirectionnel de l'Office Fédéral de la Topographie Swisstopo.





#### 4.3.4. Nombre de connexions par ligne

Le nombre de connexions par ligne est également un indicateur important dans l'optique d'évaluer le potentiel de percolation d'un système de fractures en 2D (Fig. 22, Robinson, 1984; Sanderson & Nixon, 2015). La valeur de seuil de percolation pour le nombre de connexions par ligne est de 3.57 (Robinson, 1984). Il s'agit de la valeur théorique à partir de laquelle la percolation du système est possible. Elle n'est atteinte que dans une série de 4 mailles jointives dans la zone d'étude (en orange, Fig. 22).

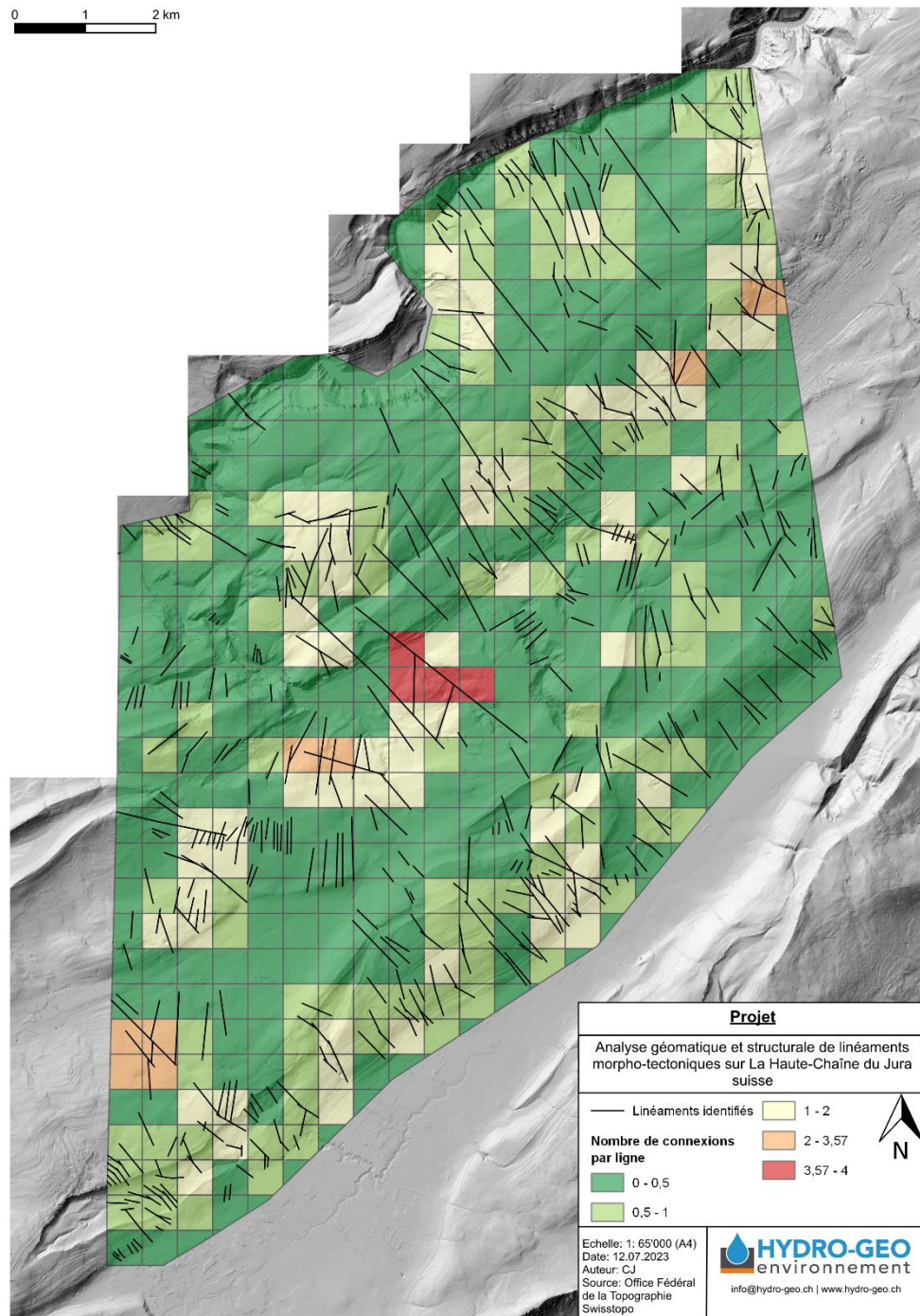


Figure 22 : Carte montrant le nombre de connexions par ligne pour chaque maille de la zone d'échantillonnage, sur fond de swissALTI3D relief monodirectionnel de l'Office Fédéral de la Topographie Swisstopo.



#### 4.3.5. Intensités $P_{21}$ & $P_{22}$

L'intensité 2D ou  $P_{21}$  permet de se rendre compte de la densité de fractures pour chaque maille de la zone d'échantillonnage. Pour rappel, chacune des mailles fait 500 x 500 [m]. La carte de la figure 23 représente cette intensité sur la grille d'échantillonnage et permet de mettre en évidence certaines zones avec une plus forte densité de linéaments par rapport aux mailles voisines. Il n'existe pas de valeur seuil liée à la connectivité du réseau pour l'intensité 2D. Elle est utilisée pour le calcul de l'intensité sans dimension ou  $P_{22}$  (Fig. 24), pour laquelle une valeur supérieure à 5.61 signifie que le système de fractures est connecté (Robinson, 1984). Seules deux mailles dépassent cette valeur de seuil, au centre de la zone et au nord (Fig. 24).

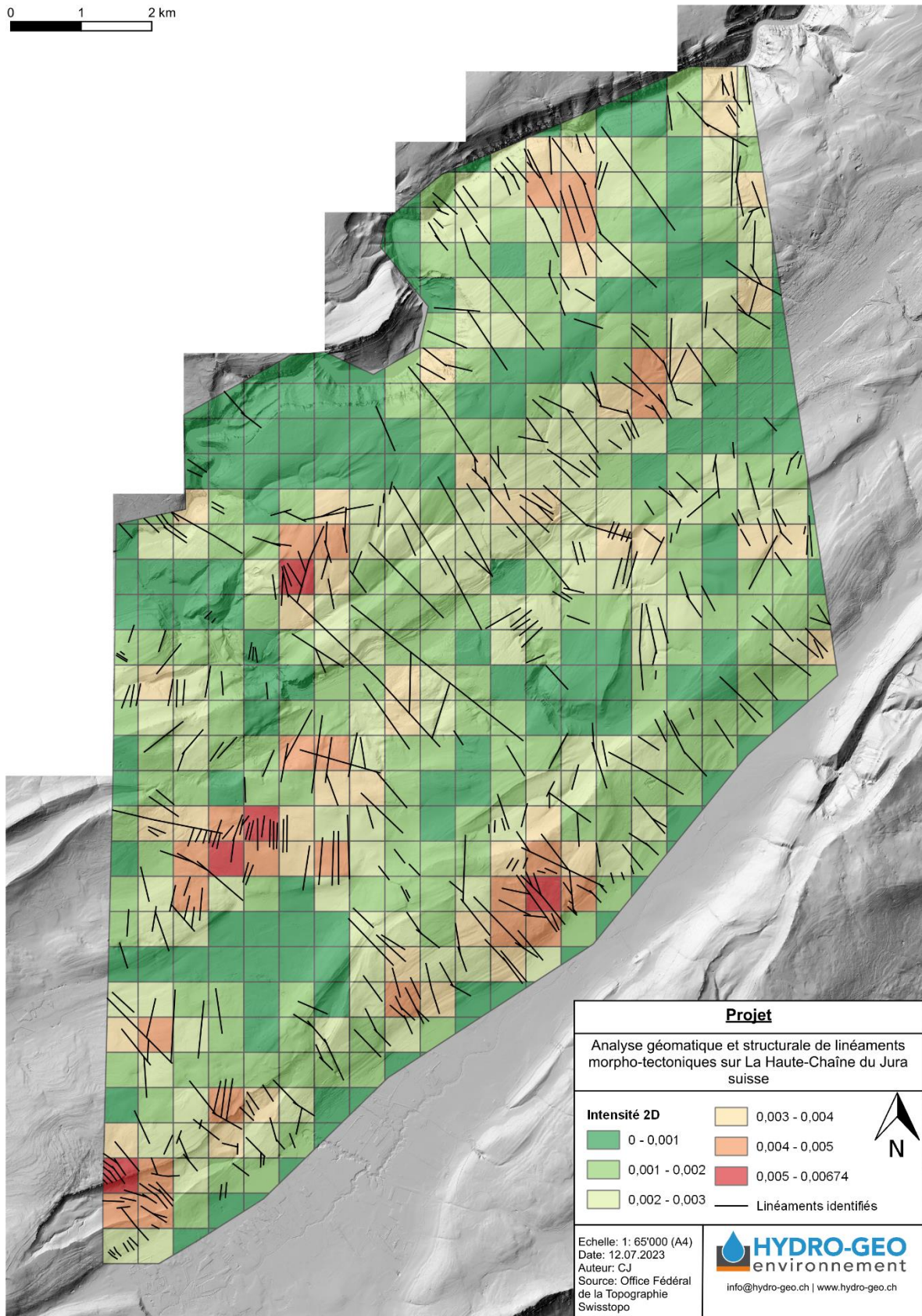


Figure 23 : Carte montrant l'intensité  $P_{21}$  pour chaque maille de la zone d'échantillonnage, sur fond de swissALTI3D relief monodirectionnel de l'Office Fédéral de la Topographie Swisstopo.



0 1 2 km

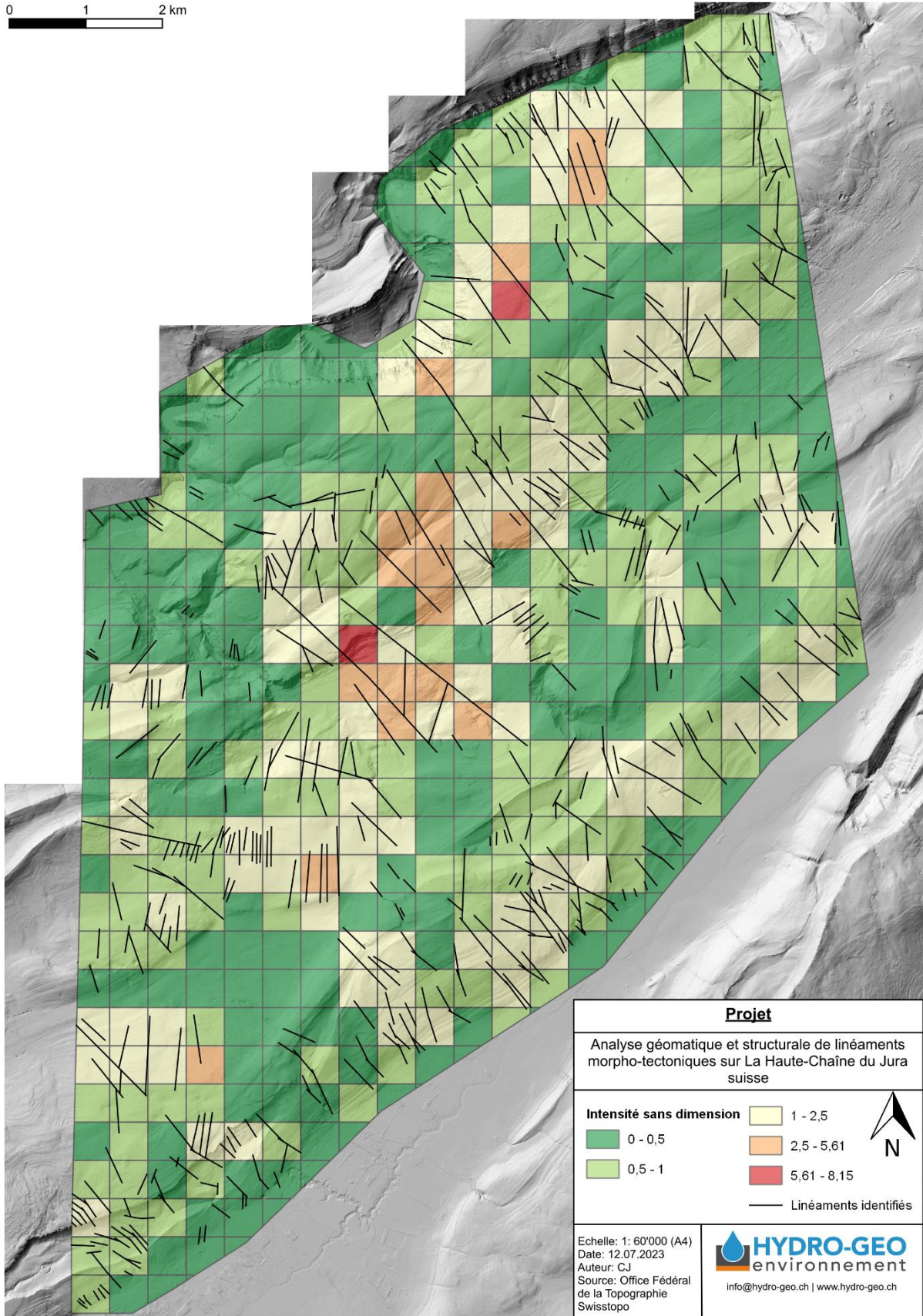


Figure 24 : Carte montrant l'intensité sans dimension  $P_{22}$  pour chaque maille de la zone d'échantillonnage, sur fond de swissALTI3D relief monodirectionnel de l'Office Fédéral de la Topographie Swisstopo



#### 4.3.6. Intensité $P_{21}$ pour les unités stratigraphiques

Afin de mieux comprendre l'influence du type de lithologie sur la présence des fractures en surface et afin de mettre en évidence certaines unités stratigraphiques qui auraient un bon potentiel de fracturation, l'intensité  $P_{21}$  a été calculée sous forme de pixels de 100 [m] de diamètre, avec un rayon de recherche des linéaments de 250 [m]. Une fois recoupée avec l'information de la carte géologique, cette intensité a été représentée sous forme d'histogramme (Fig. 25). Les formations stratigraphiques sont ordonnées selon leur âge, la plus jeune étant la ligne du haut (« Tertiaire ») et la plus vieille la ligne la plus basse (« Hauptrogenstein: Couches à *Ostrea acuminata* ») (Fig. 25). Elles ont été également regroupées selon des « unités » stratigraphiques, du plus vieux au plus jeune : Dogger, Malm inférieur, Malm supérieur, Crétacé inférieur et Tertiaire (Fig. 25). On remarque que la plus grande intensité moyenne  $P_{21}$  est présente dans l'unité du Malm inférieur.

Les intensités  $P_{21}$  moyennes par unités stratigraphiques ont été également représentées sous forme de graphique boîte à moustache (« boxplot », Fig. 26). On remarque sur ce graphique que c'est le Malm inférieur qui a la valeur médiane de  $P_{21}$  la plus élevée (Fig. 26).

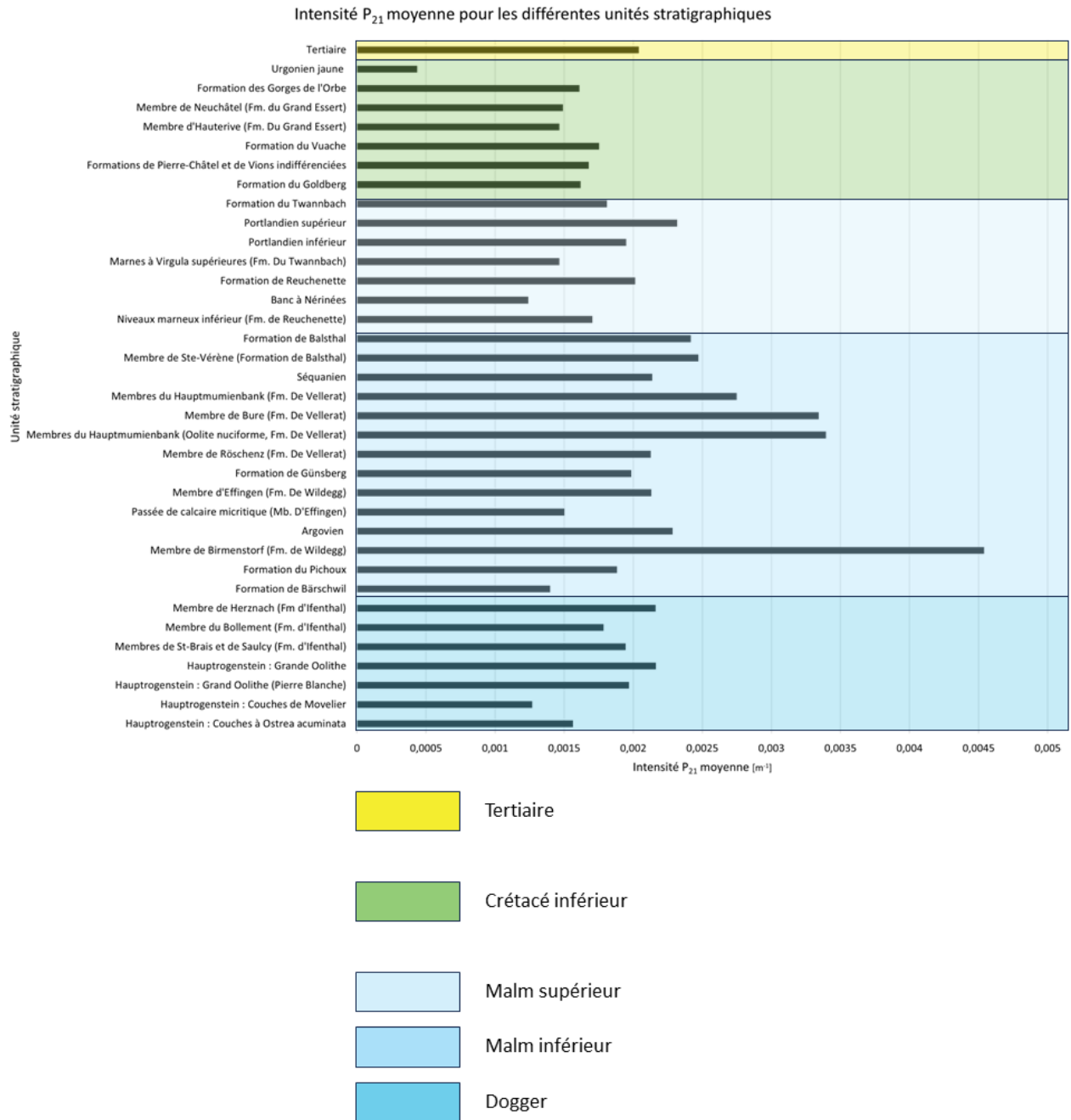


Figure 25 : Histogramme de l'intensité  $P_{21}$  moyenne pour les unités stratigraphiques affleurantes sur la zone d'étude du Locle.

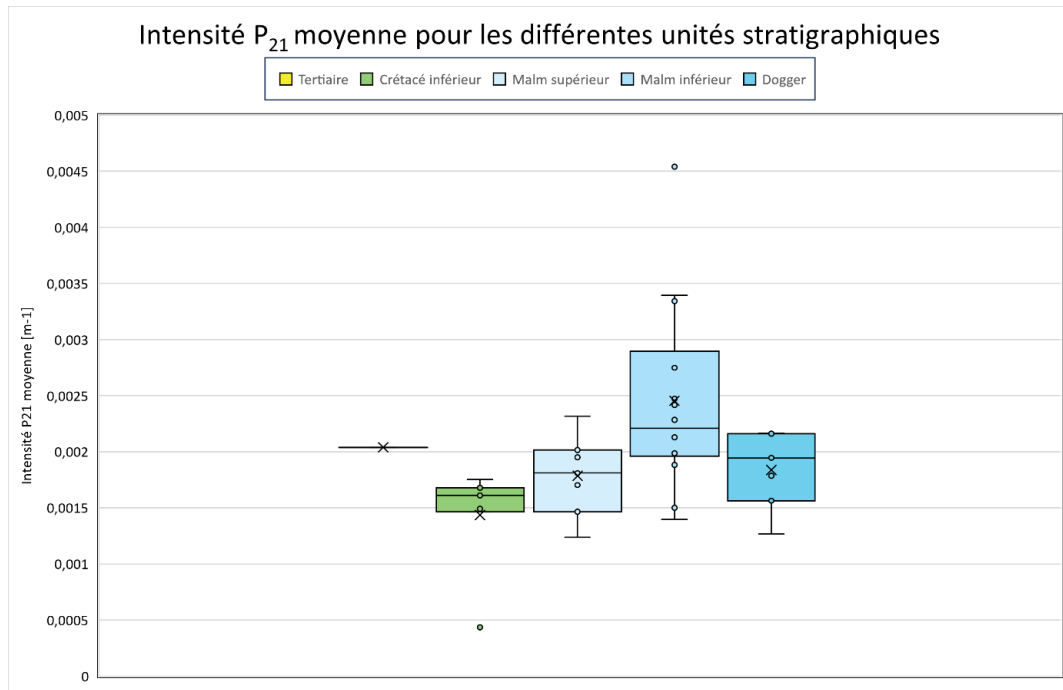


Figure 26: Graphique boîte à moustaches de l'intensité  $P_{21}$  moyenne pour les unités stratigraphiques.



## 5. Discussion

La discussion de ce mémoire de stage est construite en 4 parties : (1) une première les résultats structuraux obtenus par l'analyse géomatique et leurs implications géologiques ; (2) une discussion sur la viabilité des outils SIG utilisés ; (3) une discussion sur les problèmes rencontrés et leur amélioration possible ; et (4) une discussion personnelle sur le stage, les objectifs et le retour d'expérience obtenu dans le cadre du Certificat Complémentaire en Géomatique de l'Université de Genève.

### 5.1. Géologie structurale

Les linéaments identifiés sur la zone du Locle sont principalement localisés dans certaines structures géologiques clés, qui aident à les faire ressortir sur le MNT (Fig. 27). En effet, ils sont principalement localisés sur des axes d'anticlinaux de la région : les anticlinaux du Grand Som Martel et celui d'Entre deux Monts au sud de la zone d'étude, les anticlinaux du Bois du Creux et du Grand Calirou au centre de la zone d'étude et les anticlinaux de Pouillerel et du lac de Moron au nord. Certaines des structures apportant la plus grande connectivité sur la zone d'étude sont quant à elle situées au niveau de l'axe du synclinal du Locle, dans des lithologies tertiaires (Fig. 27 et 28). Il pourrait y avoir une influence des mouvements glaciaires sur la distribution des linéaments dans les vallées, les lithologies tertiaires et les dépôts quaternaires. En effet les glaciers érodent les lithologies affleurantes dans les vallées, lissant les linéaments tectoniques, et déposent des sédiments, qui forment des linéaments géomorphologiques, qui sont difficiles à discerner des linéaments morpho-tectoniques.

Les axes d'anticlinaux se traduisent géomorphologiquement par la présence de reliefs et de crêtes, qui font affleurer les roches et ressortir les linéaments, ce qui explique la position des linéaments vus sur le MNT. Les trois familles de linéaments identifiées grâce au diagramme de Rose (Fig. 14) ne semblent pas être localisées préférentiellement dans l'une ou l'autre des unités stratigraphiques (Fig. 28) ; cependant, la plus grande proportion de linéaments est présente dans le Malm inférieur (Fig. 26 & 28).

L'étude topologique du réseau de linéaments du Locle révèle que la connectivité du réseau en surface est dans l'ensemble limitée et que seules certaines zones atteignent les valeurs de seuil de percolation de 1.56 connexion par branche (Fig. 21, Sanderson & Nixon, 2015). Seule une zone atteint la valeur de 3.57 connexions par lignes, valeur de seuil critique pour la percolation des fluides (Fig. 22, Robinson, 1984). Cette même zone atteint également la valeur de seuil de connectivité de 5.61 pour l'intensité  $P_{22}$  (Fig. 24, Nyberg et al., 2018; Sanderson & Nixon, 2015). Ces valeurs sont de bons indicateurs de la connectivité du réseau, il faut cependant noter que la connectivité en surface ne reflète pas exactement la connectivité en subsurface, puisque les failles et les fractures sont des objets planaires et non linéaires. L'interprétation 2D de ces objets planaires uniquement avec leur interaction avec la surface ne révèle donc pas le potentiel complet, ni n'infirme la présence d'un réseau connecté en subsurface en 3 dimensions.

Les zones mises en évidence par ces indices sont donc des zones d'intérêt géothermique préliminaires, qu'il est nécessaire de compléter par des investigations approfondies au moyen





d'autres méthodes d'exploration du sous-sol, comme la sismique réflexion 2D et 3D, les méthodes électriques profondes, la gravimétrie ou encore par des forages d'exploration, afin de confirmer la présence ou non des fractures et failles. La présence d'un réseau de faille, bien connecté, est essentielle pour la valorisation géothermique d'aquifères fracturés de moyenne profondeur, pour autant qu'ils soient couplés à des propriétés matricielles suffisantes et à un bon gradient géothermique.

Enfin, bien qu'il soit possible de voir quelles unités stratigraphiques sont traversées par les linéaments en surface (Fig. 28), cela ne veut pas forcément dire que les fractures ou les failles passent uniquement par ces unités et pas par d'autres en profondeur, dans le voisinage de structures géologiques complexes qui caractérisent la zone (Eichenberger et al., 2020; Marro et al., 2023; Polasek, 2020; Radaideh & Mosar, 2021; Rusillon & Savoy, 2023; Sommaruga et al., 2017).

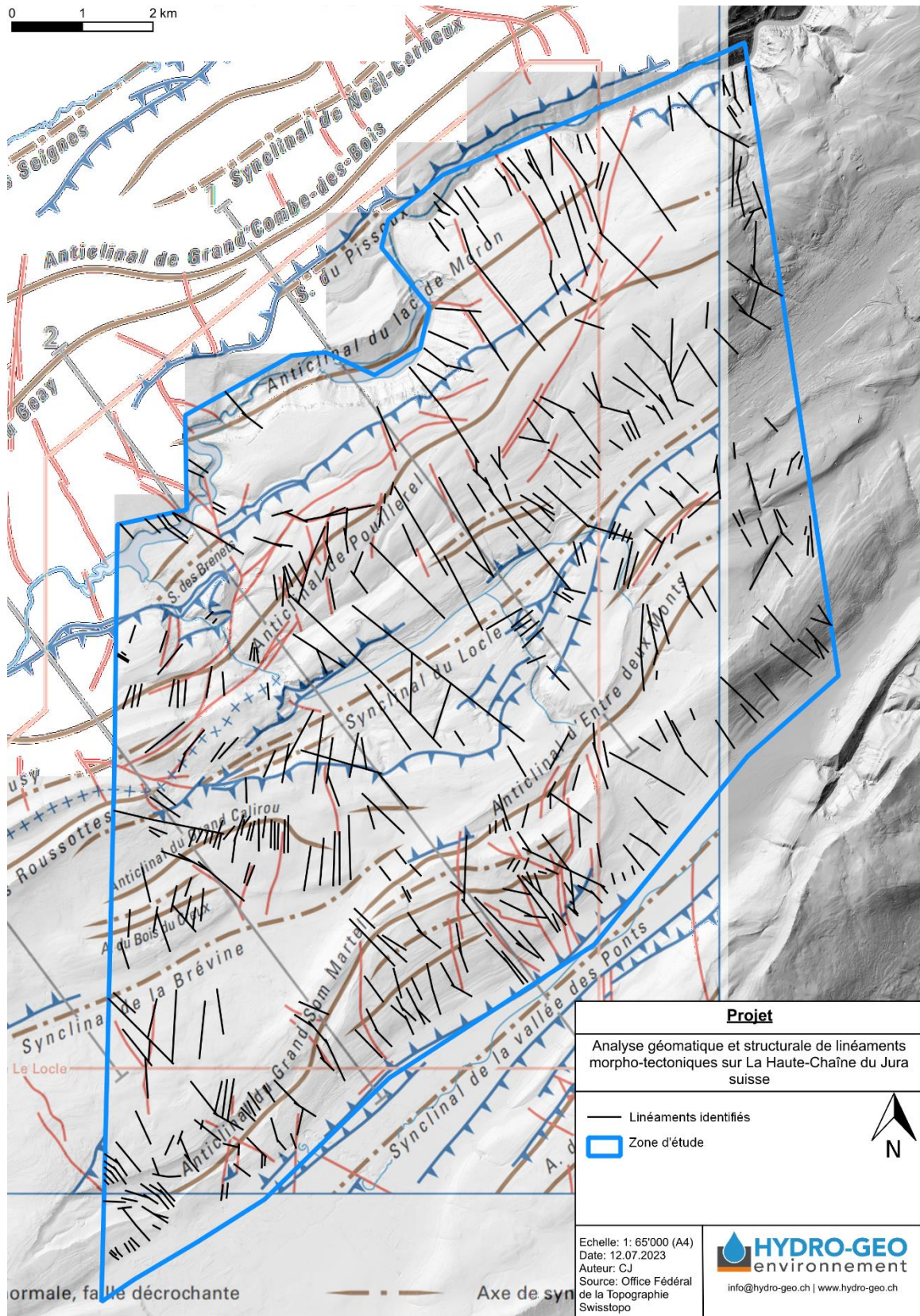


Figure 27 : Carte tectonique de Eichenberger et al. (2020) avec les linéaments de la zone du Locle en superposition. La carte de fond est la swissALT13D relief monodirectionnel de l'Office Fédéral de la Topographie Swisstopo.



0 1 2 km

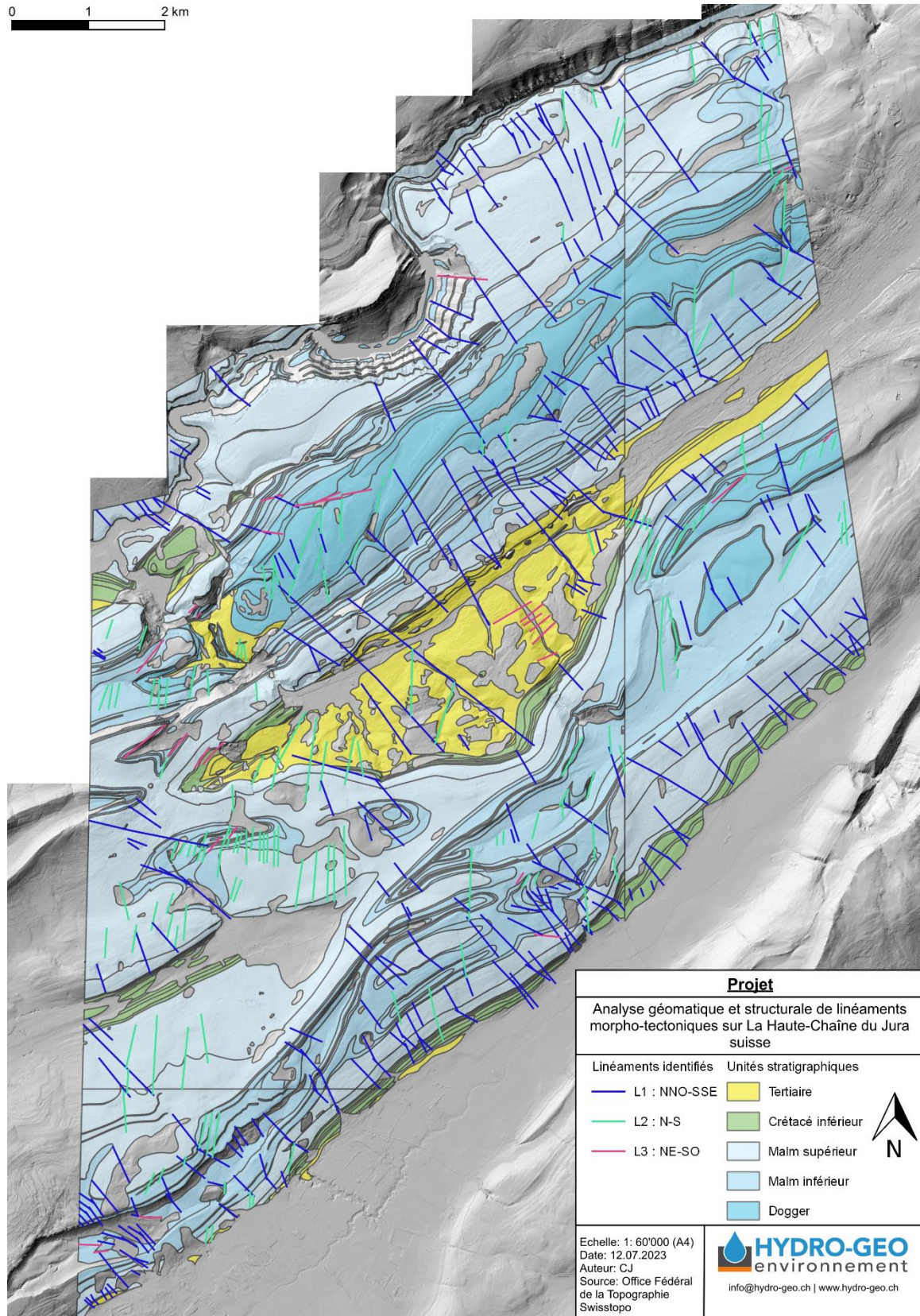


Figure 28 : Carte géologique modifiée d'après GeoCover V3 (Office Fédéral de la Topographie Swiss topo) afin de montrer les unités stratigraphiques avec les linéaments de la zone du Locle superposés. Le fond de carte est le swissALTI3D relief monodirectionnel de l'Office Fédéral de la Topographie Swiss topo.



## 5.2. Les SIG au service de la caractérisation des linéaments et optimisations Python sur QGIS

La télédétection des linéaments à partir de MNT a permis de caractériser et de digitaliser le réseau de fracture de la zone d'étude du Locle. Les outils existants sur QGIS (SAGA, GDAL, Network GT et les outils natifs de QGIS) permettent de réaliser les étapes de préparation des données et d'analyse des linéaments, même si certaines étapes n'étaient pas optimisées pour l'analyse. La digitalisation automatique à l'aide des outils « Digitize fracture network » de Network GT et « Raster skeletonization » de SAGA n'a pas été satisfaisante et ne constitue donc pas une alternative à l'identification et la digitalisation manuelle des linéaments par un opérateur. La résolution d'erreurs introduites par la digitalisation automatique et sa correction peut prendre plus de temps que la digitalisation manuelle et est moins précise, car l'interprétation géomorphologique de l'opérateur constitue un point important de la digitalisation de linéaments tectoniques.

La réalisation du plugin Lineament Analysis sur QGIS 3.18.1 à l'aide de Python 3.7 (objet de cette étude, Fig.8 et annexes 1 à 5) était justifiée pour réduire le temps d'importation et de traitement initial des tuiles MNT.

En termes de gain de temps et d'optimisation, après chronométrage, le temps d'importation des tuiles MNT dans le projet QGIS est le même que le temps que prend le plugin Lineament Analysis pour réaliser l'entièreté du traitement des tuiles MNT. Puisque les couches ne sont pas importées automatiquement, l'utilisateur peut choisir celles qu'il souhaite importer dans son projet QGIS.

La digitalisation manuelle n'aurait pas pu être possible sans l'utilisation de SIG, notamment QGIS. La possibilité d'afficher les couches géoréférencées (MNT, couches géologiques, couches de réseaux de transports, orthophotographie), de les superposer et de les comparer était vitale afin de pouvoir identifier les linéaments (Fig. 9).

Enfin, dans le cadre de l'analyse topologique et géométrique, il s'agissait au début de réaliser un deuxième plugin uniquement pour effectuer l'analyse à partir du shapefile de linéaments. Cependant, au cours de la recherche méthodologique il est devenu clair que le plugin n'aurait pas été réalisable dans le sens où chaque étape d'analyse nécessite d'être vérifiée par l'opérateur afin d'ajuster les paramètres pour obtenir le meilleur résultat possible. Il valait donc mieux générer une méthode (Fig. 10) impliquant certains outils préalablement choisis dans la boîte à outils de Network GT.

## 5.3. Problèmes rencontrés

Plusieurs défis ont été rencontrés lors de la réalisation de cette étude de linéaments morpho-tectoniques au sein de la Haute-Chaîne du Jura.

La première difficulté a été de déterminer si la détection et digitalisation automatique des linéaments était réalisable ou non. Cette étape a comporté de nombreux essais infructueux. Aussi, certains problèmes étaient dus à la compatibilité des logiciels, packages et plugins avant le passage à QGIS 3.18.1 with GRASS 7.8.5. QGIS 3.6.1 with GRASS 7.6.1 et certains packages



et plugins, en particulier une collection d'algorithmes Python, scikit-image, et Network GT, étaient incompatibles.

Ensuite, transposer les connaissances de développement de plugin sur ArcGIS Pro sur QGIS n'était pas trivial. En effet, l'environnement de développement sur QGIS est significativement différent de celui d'ArcGIS Pro. L'interface au langage Python spécifique pour les fonctions QGIS (PyQGIS) diffère en effet entre ArcPy et ArcGIS Pro. Le plus grand défi a été de déboguer le code, car la console de messages d'erreur Python sur QGIS ne donnait pas de message précis au début du développement. Les forums de discussion, les blogs, les livres et les vidéos d'apprentissage ont donc été d'une grande aide (Herbert Santos, 2019; Lawhead, 2015; Rob van Putten, 2021a, 2021b, 2022; The Qt Company Ltd., s. d.; Ujaval Gandhi, 2023).

Enfin, la digitalisation manuelle a nécessité quelques essais et des discussions avec David Polasek afin de bien comprendre comment identifier les linéaments morpho-tectoniques. Ceci met en avant l'importance d'avoir un opérateur SIG/géomaticien avec une connaissance en géologie structurale aguerrie. Les problèmes de topologies du réseau observés après la première digitalisation ont nécessité une attention toute particulière, mais ceux-ci ont pu être résolus rapidement.

#### **5.4. Discussion personnelle – objectifs du stage**

Ce stage chez Hydro-Géo Environnement, dans le cadre du Certificat Complémentaire en Géomatique m'a permis de mettre en pratique les connaissances acquises lors des cours du certificat. En particulier, j'ai pu acquérir de l'autonomie dans la conduite d'un projet de géomatique, appliqué à mon domaine d'étude, la géologie. Lors de l'élaboration du projet, j'ai pu formuler mes propres hypothèses de recherche et trouver la meilleure façon d'y répondre, à l'aide des SIG. L'environnement de travail chez Hydro-Géo Environnement était propice au développement de mes idées que j'ai pu directement inclure dans le projet. Mes compétences sur QGIS, sur les applications de la télédétection et surtout en programmation Python appliquée aux SIG se sont grandement améliorées. En effet, j'ai pu notamment allouer un peu de mon temps lors du stage pour l'apprentissage, que ce soit dans la création de plugins, de PyQT ou des techniques de télédétection appliquées à la géologie structurale. J'ai maintenant les outils nécessaires pour appliquer mes compétences géomatiques à des projets variés dans ma future vie professionnelle.



## 6. Conclusion

### 6.1. Synthèse générale

L'analyse géomatique et structurale de linéaments morpho-tectoniques sur la Haute-Chaîne du Jura suisse a été réalisée à l'aide de QGIS 3.18.1 with GRASS 7.8.5 et Python 3.7 avec succès.

Afin d'optimiser au mieux le traitement et l'analyse des données acquises par télédétection, un inventaire des outils SIG pouvant servir à cette analyse a été effectué, un test d'automatisation de détection des linéaments morpho-tectoniques et un plugin pour le traitement des tuiles MNT (« Lineament Analysis », Fig. 8, annexes 1 à 5) ont été réalisés. De plus, une méthode de digitalisation (Fig. 9) ainsi qu'une autre pour l'analyse des linéaments (Fig. 10) ont été définies et ont permis de mettre en évidence les zones d'intérêt de prospection géothermique avec de bons indices de connectivité des fractures dans la zone du Locle.

Le plugin Lineament Analysis permet principalement de combler le manque d'un outil efficace afin de fusionner de nombreuses tuiles MNT sans devoir les importer dans le projet QGIS. Il a été pensé spécialement pour l'analyse géomatique des linéaments morpho-tectoniques et permet de réaliser automatiquement les ombrages nécessaires à l'analyse.

Finalement, la récolte de données MNT et le géotraitement par SIG des linéaments morpho-tectoniques est une étape préliminaire au travail de terrain et aux investigations géophysiques (Scheiber et al., 2015). Cette étude s'inscrit donc dans le cadre de mandats existants chez Hydro-Géo Environnement sur la zone du Locle et sera utile dans la prospection géothermique pour de futurs mandats.

### 6.2. Critique de la réalisation concrète

Le plugin Lineament Analysis, fonctionnel en l'état, pourrait bénéficier d'améliorations qui n'ont pas été implémentées par manque de temps. Parmi ces améliorations, il y a la découpe du MNT fusionné à la zone d'étude avec le polygone de la zone d'étude. En effet, actuellement, le MNT est découpé à l'étendue du polygone et non pas à la forme exacte du polygone. C'est un problème qui est solvable, mais qui aurait nécessité un temps de recherche additionnel et d'essai. Permettre l'enregistrement des fichiers de sorties sous un autre format que TIFF, serait également appréciable. Pour l'instant, il n'est pas possible pour l'utilisateur de choisir le système de référence, qui est fixé à CH1903+ LV95. Permettre ce choix ainsi que d'ajouter des valeurs par défaut d'azimut et d'inclinaison des sources de lumière pour les ombrages serait également des améliorations possibles. Finalement, on pourrait ajouter une génération de noms automatique pour les fichiers de sauvegarde en fonction des ombrages et une barre de progression. Au vu du temps passé à développer le plugin et à le déboguer, il faut compter environ 2 semaines pour ajouter ces fonctionnalités au plugin.

Pour les méthodes de digitalisation et d'analyse, il s'agira d'améliorer surtout la digitalisation, fonction du temps qui peut être allouée à cette tâche. Notamment, l'utilisation de plusieurs échelles (Scheiber et al., 2015) contre une seule dans cette étude (1 : 10'000) permettrait



d'affiner la détection. Toutefois, les deux méthodes et le plugin sont applicables à d'autres zones d'étude en l'état.



## 7. Bibliographie

- Acquisition de données LiDAR. (2023). Office fédéral de topographie swisstopo.  
<https://www.swisstopo.admin.ch/fr/connaissances-faits/geoinformation/donnees-lidar.html>
- Ahmadi, H., & Pekkan, E. (2021). Fault-Based Geological Lineaments Extraction Using Remote Sensing and GIS—A Review. *Geosciences*, 11(5), 183. <https://doi.org/10.3390/geosciences11050183>
- Cohen, K. M., Finney, S. C., Gibbard, P. L., & Fan, J. X. (2013). *The ICS International Chronostratigraphic Chart. Episode 36 : 199-204.*
- Continuum Analytics. (s. d.). *Anaconda Navigator (2.4.2)* [Logiciel]. Continuum Analytics, Inc.
- Eichenberger, U., Mojon, P.-O., Gogniat, S., Pictet, A., Blant, D., Locatelli, D., Metral, V., & Morard, A. (2020). *113 Le Locle avec partie 1123 Le Russey, Notice explicative de l'Atlas géologique de la Suisse au 1 : 25 000 (49 Figures, 1 tableau et 1 planche hors texte).*
- Gasparo Morticelli, M., Avellone, G., Sulli, A., Agate, M., Basilone, L., Catalano, R., & Pierini, S. (2017). Mountain building in NW Sicily from the superimposition of subsequent thrusting and folding events during Neogene : Structural setting and tectonic evolution of the Kumeta and Pizzuta ridges. *Journal of Maps*, 13(2), 276-290. <https://doi.org/10.1080/17445647.2017.1300546>
- Healy, D., Rizzo, R. E., Cornwell, D. G., Farrell, N. J. C., Watkins, H., Timms, N. E., Gomez-Rivas, E., & Smith, M. (2017). FracPaQ : A MATLAB™ toolbox for the quantification of fracture patterns. *Journal of Structural Geology*, 95, 1-16. <https://doi.org/10.1016/j.jsg.2016.12.003>
- Herbert Santos. (2019, novembre 19). *Converting Model in Plug-in QGIS*. StackExchange.  
<https://gis.stackexchange.com/questions/337702/converting-model-in-plug-in-in-qgis>
- Hermann Schinagl. (2018). *Link Shell Extension (3.9.3.5)* [Logiciel].  
<https://schinagl.priv.at/nt/hardlinkshellext/linkshellextension.html>
- How Simplify Line and Simplify Polygon work—ArcGIS Pro | Documentation.* (2023).  
<https://pro.arcgis.com/en/pro-app/latest/tool-reference/cartography/how-simplify-line-works.htm>
- Jaboyedoff, M., Abellán, A., Carrea, D., Derron, M.-H., Matasci, B., & Michoud, C. (2018). Mapping and Monitoring of Landslides Using LIDAR. In R. P. Singh & D. Bartlett (Éds.), *Natural Hazards* (1<sup>re</sup> éd., p. 397-420). CRC Press. <https://doi.org/10.1201/9781315166841-17>
- Jaboyedoff, M., Oppikofer, T., Abellán, A., Derron, M.-H., Loye, A., Metzger, R., & Pedrazzini, A. (2010). Use of LIDAR in landslide investigations : A review. *Natural Hazards*, 61(1), 5-28.  
<https://doi.org/10.1007/s11069-010-9634-2>
- Lawhead, J. (2015). *QGIS Python Programming Cookbook*. Packt Publishing Ltd.
- Marro, A., Hauvette, L., Borderie, S., & Mosar, J. (2023). Tectonics of the Western Internal Jura fold-and-thrust belt : 2D kinematic forward modelling. *Swiss Journal of Geosciences*.  
<https://doi.org/10.1186/s00015-023-00435-2>
- Microsoft. (2023). *Visual Studio Code* [Logiciel]. Microsoft.
- Montoya, S. (2018, juillet 23). *Extracting geological faults and orientation representation in QGIS with GeoTrace—Tutorial*. Hatari Labs. <https://hatarilabs.com/ih-en/extracting-geological-faults-and-orientation-representation-in-qgis-with-geotrace-tutorial>





Nyberg, B., Nixon, C. W., & Sanderson, D. J. (2018). NetworkGT : A GIS tool for geometric and topological analysis of two-dimensional fracture networks. *Geosphere*, 14(4), 1618-1634. <https://doi.org/10.1130/GES01595.1>

OFEN, O. fédéral de l'énergie. (2023a). *Géothermie*. <https://www.bfe.admin.ch/bfe/fr/home/versorgung/erneuerbare-energien/geothermie.html>

OFEN, O. fédéral de l'énergie. (2023b). *Promotion de la géothermie*. <https://www.bfe.admin.ch/bfe/fr/home/foerderung/erneuerbare-energien/foerderung-geothermie.html>

OFEN, O. fédéral de l'énergie. (2023c). *Qu'est-ce que la Stratégie énergétique 2050?* <https://www.bfe.admin.ch/bfe/fr/home/politik/energiestrategie-2050/was-ist-die-energiestrategie-2050.html>

OFEV, O. fédéral de l'environnement. (2023). *Objectifs suisses de réduction des émissions de gaz à effet de serre*. <https://www.bafu.admin.ch/bafu/fr/home/themen/thema-klima/klimawandel-stoppen-und-folgen-meistern/massnahmen-der-schweiz-zur-verminderung-ihrer-treibhausgasemissionen/ziele-der-schweiz-zur-verminderung-ihrer-treibhausgasemissionen.html>

Palamakumbura, R., Krabbendam, M., Whitbread, K., & Arnhardt, C. (2020). Data acquisition by digitizing 2-D fracture networks and topographic lineaments in geographic information systems : Further development and applications. *Solid Earth*, 11(5), 1731-1746. <https://doi.org/10.5194/se-11-1731-2020>

Peacock, D. C. P., & Sanderson, D. J. (2018). Structural analyses and fracture network characterisation : Seven pillars of wisdom. *Earth-Science Reviews*, 184, 13-28. <https://doi.org/10.1016/j.earscirev.2018.06.006>

Polasek, D. (2020). *Kinematic and Hydrogeological Role of the Jura Mountains' Haute-Chaîne Fracture Network*, unpublished.

QGIS Project. (s. d.). Cheat sheet for PyQGIS. *QGIS Documentation 3.28*. Consulté 20 juillet 2023, à l'adresse [https://docs.qgis.org/3.28/en/docs/pyqgis\\_developer\\_cookbook/cheat\\_sheet.html](https://docs.qgis.org/3.28/en/docs/pyqgis_developer_cookbook/cheat_sheet.html)

QGIS Project. (2020). *PyQGIS developer cookbook Release 3.4*.

Radaideh, O. M. A., Grasemann, B., Melichar, R., & Mosar, J. (2016). Detection and analysis of morphotectonic features utilizing satellite remote sensing and GIS : An example in SW Jordan. *Geomorphology*, 275, 58-79. <https://doi.org/10.1016/j.geomorph.2016.09.033>

Radaideh, O. M. A., & Mosar, J. (2021). Cenozoic Tectonic Deformation Along the Pontarlier Strike-Slip Fault Zone (Swiss and French Jura Fold-and-Thrust Belt) : Insights From Paleostress and Geomorphic Analyses. *Tectonics*, 40(5). <https://doi.org/10.1029/2021TC006758>

Rob van Putten (Réalisateur). (2021a, janvier 15). *QGIS tutorial—Plugin development—Part 1* [Tutoriel]. Youtube. <https://www.youtube.com/watch?v=GccxvQ1ypbc>

Rob van Putten (Réalisateur). (2021b, janvier 25). *QGIS tutorial—Plugin development—Part 2* [Tutoriel]. Youtube. <https://www.youtube.com/watch?v=TXujpzl4Yvs>

Rob van Putten (Réalisateur). (2022, janvier 17). *QGIS tutorial—Plugin development—Part 3* [Tutoriel]. Youtube. <https://www.youtube.com/watch?v=KQ8mOpVQ62w>



Robinson, P. C. (1984). Numerical calculations of critical densities for lines and planes. *Journal of Physics A: Mathematical and General*, 17(14), 2823. <https://doi.org/10.1088/0305-4470/17/14/025>

Rusillon, E., & Savoy, L. (2023). *Caractérisation des unités réservoirs du Malm inférieur : Modèle lithostratigraphique des horizons productifs potentiels intra Malm—Rapport de Synthèse*. Hydro-Géo Environnement.

Salui, C. L. (2018). Methodological Validation for Automated Lineament Extraction by LINE Method in PCI Geomatica and MATLAB based Hough Transformation. *Journal of the Geological Society of India*, 92(3), 321-328. <https://doi.org/10.1007/s12594-018-1015-6>

Sanderson, D. J., & Nixon, C. W. (2015). The use of topology in fracture network characterization. *Journal of Structural Geology*, 72, 55-66. <https://doi.org/10.1016/j.jsg.2015.01.005>

Sanderson, D. J., & Nixon, C. W. (2018). Topology, connectivity and percolation in fracture networks. *Journal of Structural Geology*, 115, 167-177. <https://doi.org/10.1016/j.jsg.2018.07.011>

Sanderson, D. J., & Peacock, D. C. P. (2020). Making rose diagrams fit-for-purpose. *Earth-Science Reviews*, 201, 103055. <https://doi.org/10.1016/j.earscirev.2019.103055>

Scheiber, T., Fredin, O., Viola, G., Jarna, A., Gasser, D., & Łapińska-Viola, R. (2015). Manual extraction of bedrock lineaments from high-resolution LiDAR data : Methodological bias and human perception. *GFF*, 137(4), 362-372. <https://doi.org/10.1080/11035897.2015.1085434>

Sommaruga, A., Mosar, J., Schori, M., & Gruber, M. (2017). The Role of the Triassic Evaporites Underneath the North Alpine Foreland. In *Permo-Triassic Salt Provinces of Europe, North Africa and the Atlantic Margins* (p. 447-466). Elsevier. <https://doi.org/10.1016/B978-0-12-809417-4.00021-5>

Tanner, D. C., Buness, H., Igel, J., Günther, T., Gabriel, G., Skiba, P., Plenefisch, T., Gestermann, N., & Walter, T. R. (2020). Fault detection. In *Understanding Faults* (p. 81-146). Elsevier. <https://doi.org/10.1016/B978-0-12-815985-9.00003-5>

The Qt Company Ltd. (s. d.). Qt Designer Manual. *Qt Documentation*. Consulté 20 juillet 2023, à l'adresse <https://doc.qt.io/qt-5/qt designer-manual.html>

Ujaval Gandhi. (2023, juillet 8). Building a Python Plugin (QGIS3) [Html]. *QGIS Tutorials and Tips*. [https://www.qgistutorials.com/en/docs/3/building\\_a\\_python\\_plugin.html](https://www.qgistutorials.com/en/docs/3/building_a_python_plugin.html)

Wolfe, F. D., Stahl, T. A., Villamor, P., & Lukovic, B. (2020). Short communication : A semiautomated method for bulk fault slip analysis from topographic scarp profiles. *Earth Surface Dynamics*, 8(1), 211-219. <https://doi.org/10.5194/esurf-8-211-2020>



*Stage Certificat en géomatique 2023 – Jost Clothilde*

*Analyse géomatique et structurale de linéaments morpho-tectoniques sur La Haute-Chaîne du Jura suisse*

---



## 8. Liste des annexes

ANNEXE 1 : Plugin Lineament Analysis

ANNEXE 2 : script\_modele\_preprocessing.py

ANNEXE 3 : \_\_init\_\_.py

ANNEXE 4 : lineament\_analysis\_dialog.py

ANNEXE 5 : lineament\_analysis.py

ANNEXE 6 : Carte d'ombrage 045/45 avec les linéaments du Locle

ANNEXE 7 : Carte d'ombrage 135/45 avec les linéaments du Locle

ANNEXE 8 : Carte d'ombrage 225/45 avec les linéaments du Locle

ANNEXE 9 : Carte d'ombrage 315/45 avec les linéaments du Locle

ANNEXE 10 : Carte d'ombrage 045/25 avec les linéaments du Locle

ANNEXE 11 : Carte d'ombrage 225/25 avec les linéaments du Locle



## ANNEXE 1 : Plugin Lineament Analysis

Le plugin Lineament Analysis a été réalisé et fonctionne avec les programmes et plugins suivant : QGIS 3.18.1 with GRASS 7.8.5, les plugins Plugin Builder 3, Plugin Reloader, SAGA et GDAL, Python 3.7, les programmes Qt Designer with QGIS 3.6.1 custom widgets, l'extension Link Shell Extension (Hermann Schinagl, 2018) Visual studio code (Microsoft, 2023) et Anaconda Navigator (Continuum Analytics, s. d.). Le plugin fonctionne également sur la version *long-term release* de QGIS : 3.28.9 « Firenze ». Il n'a pas été testé sur d'autres versions des programmes, par manque de temps lors du présent projet, il est donc conseillé à l'utilisateur futur d'installer les mêmes versions des programmes cités afin de faire fonctionner le plugin. La méthode suivie pour développer ce plugin est celle décrite par QGIS Projet (2020), en combinaison avec plusieurs blogs (QGIS Project, s. d.; The Qt Company Ltd., s. d.; Ujaval Gandhi, 2023), vidéos YouTube (Rob van Putten, 2021a, 2021b, 2022) et forum de discussion sur le développement de plugin sous QGIS (Herbert Santos, 2019).

La première étape pour la réalisation du plugin a été de réaliser un modèle (Fig. 28) à l'aide du Modeleur Graphique disponible sur QGIS 3.18.1. Ce modèle contient toutes les étapes de traitement à partir de la fusion et jusqu'au calcul de seuil. Une option de calcul de pente a été

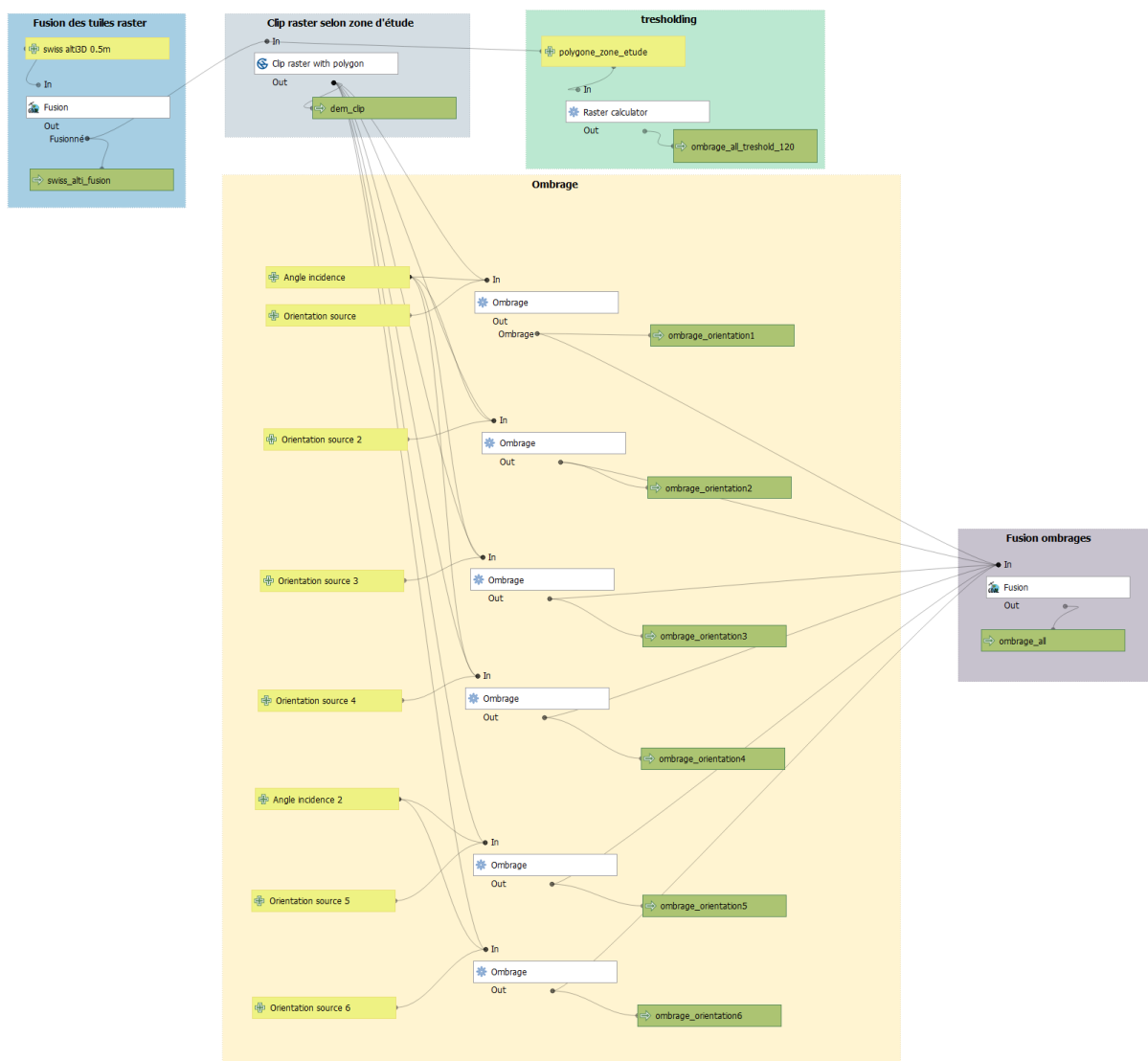


Figure 29 : Modèle de départ réalisé avec le Modeleur Graphique sur QGS 3.18.1.



ajoutée après, lors du développement du plugin. Le Modeleur Graphique a une option pour exporter le modèle en script Python, ce qui permet d'obtenir le nom des différents algorithmes en langage Python (voir annexe 2). Bien que constituant une bonne étape de départ, finalement ce sont des fonctions différentes et plus optimisées qui ont été utilisées dans le plugin final.

Ensuite, il faut créer le dossier de base du plugin Lineament Analysis, de préférence dans un dossier local qui regroupe tous les plugins personnels, à l'aide du plugin Plugin Builder 3. Une fois le dossier créé, il faut compiler le fichier ressources.qrc, qui référence les fichiers binaires utilisés dans le plugin, notamment l'icône de l'outil. Les autres fichiers ne doivent pas être modifiés. Pour compiler ressources.qrc il faut utiliser la commande « pyrcc5 » dans une fenêtre de commande comme Anaconda prompt, en tapant la ligne de code suivante : « pyrcc5 ressources.qrc -o ressources.py ». La commande « pyrcc5 » fait partie du package Qt5 faisant partie de QGIS 3.18.1. Pour la suite, d'utiliser l'extension Link Shell afin d'ajouter un lien direct vers le dossier dans les profils actifs de QGIS en suivant le cheminement suivant :

1. Aller dans le dossier qui contient le dossier du plugin puis faire clic droit et sélectionner « Pick Link Source »
2. Retourner dans le projet QGIS, aller dans les Préférences puis sélectionner « Profils utilisateurs » puis « ouvrir le dossier du profil actif »
3. Naviguer jusqu'au dossier contenant les plugins (« plugins ») dans le dossier nommé « python »
4. Dans le dossier « plugins », faire un clic droit et sélectionner « Drop As » puis « Symbolic link »
5. Redémarrer QGIS
6. En allant voir les plugins dans « Installer/gérer les extensions », le plugin apparaît maintenant, mais lorsque l'on clique dessus, une boîte vide apparaît, car le plugin n'est pas encore configuré ni développé.

L'utilisation de Link Shell Extension permet de régler la plupart de problème de chemins d'accès lors du développement du plugin dans les étapes ci-après.

Dans le dossier créé par Plugin Builder, il y a de nombreux dossiers et fichiers (Fig. 29). Les seuls qui nécessitent d'être modifiés sont :

- `__init__.py`, qui contient les instructions d'initiation de l'algorithme de traitement.
- `lineament_analysis_dialog_base.ui`, qui contient les informations et les instructions pour l'interface utilisateur (« *User Interface* », *UI*). Ce fichier est modifié par le biais de Qt Designer with QGIS 3.6.1 custom widgets, qui permet d'ajouter des entités (« widgets ») pour interagir avec l'utilisateur.
- `lineament_analysis_dialog.py`, qui contient les instructions pour lier les entités ajoutées avec Qt Designer avec les actions de l'utilisateur et l'algorithme de traitement.
- `Lineament_analysis.py`, qui contient l'algorithme de traitement des données, ou Classe principale.
- `Metadata.txt`, qui contient les metadatas concernant le plugin.

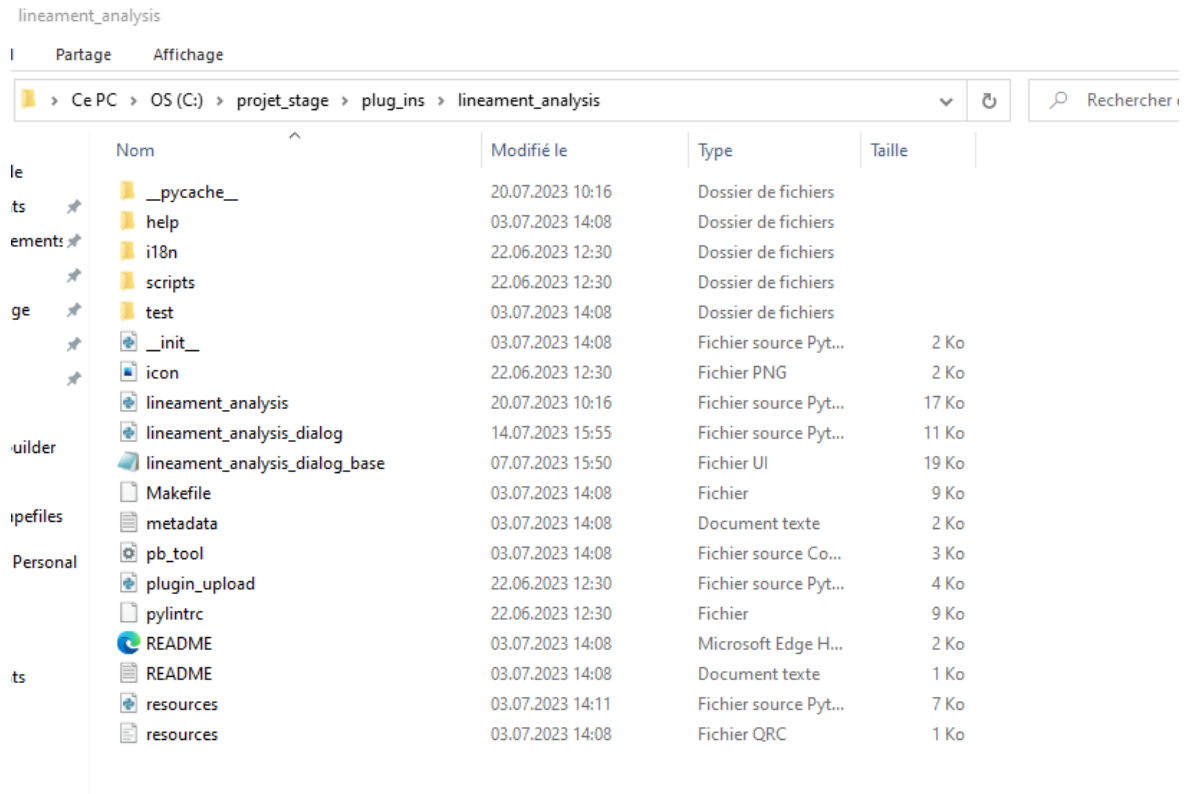


Figure 30: Image du contenu du dossier créé par le plugin Plugin Builder.

\_\_init\_\_.py (annexe 3) ne contient pas beaucoup de lignes de code, mais il est nécessaire de bien spécifier la classe principale que l'outil doit appeler lorsque l'utilisateur clique sur l'icône dans QGIS (Fig. 30).

```
# noinspection PyPep8Naming
def classFactory(iface): # pylint: disable=invalid-name
    """Load LineamentAnalysis class from file LineamentAnalysis.

    :param iface: A QGIS interface instance.
    :type iface: QgsInterface
    """
    #
    from .lineament_analysis import LineamentAnalysis
    return LineamentAnalysis(iface)
```

Figure 31 : Image d'une partie du code Python contenu dans le fichier \_\_init\_\_.py, notamment la spécification de la classe principale que l'outil doit appeler : LineamentAnalysis.



Ensuite, il faut démarrer Qt Designer et ouvrir le fichier `lineament_analysis_dialog_base.ui` et ajouter les éléments d'interface qui seront nécessaires (Fig. 31). Il est important de bien

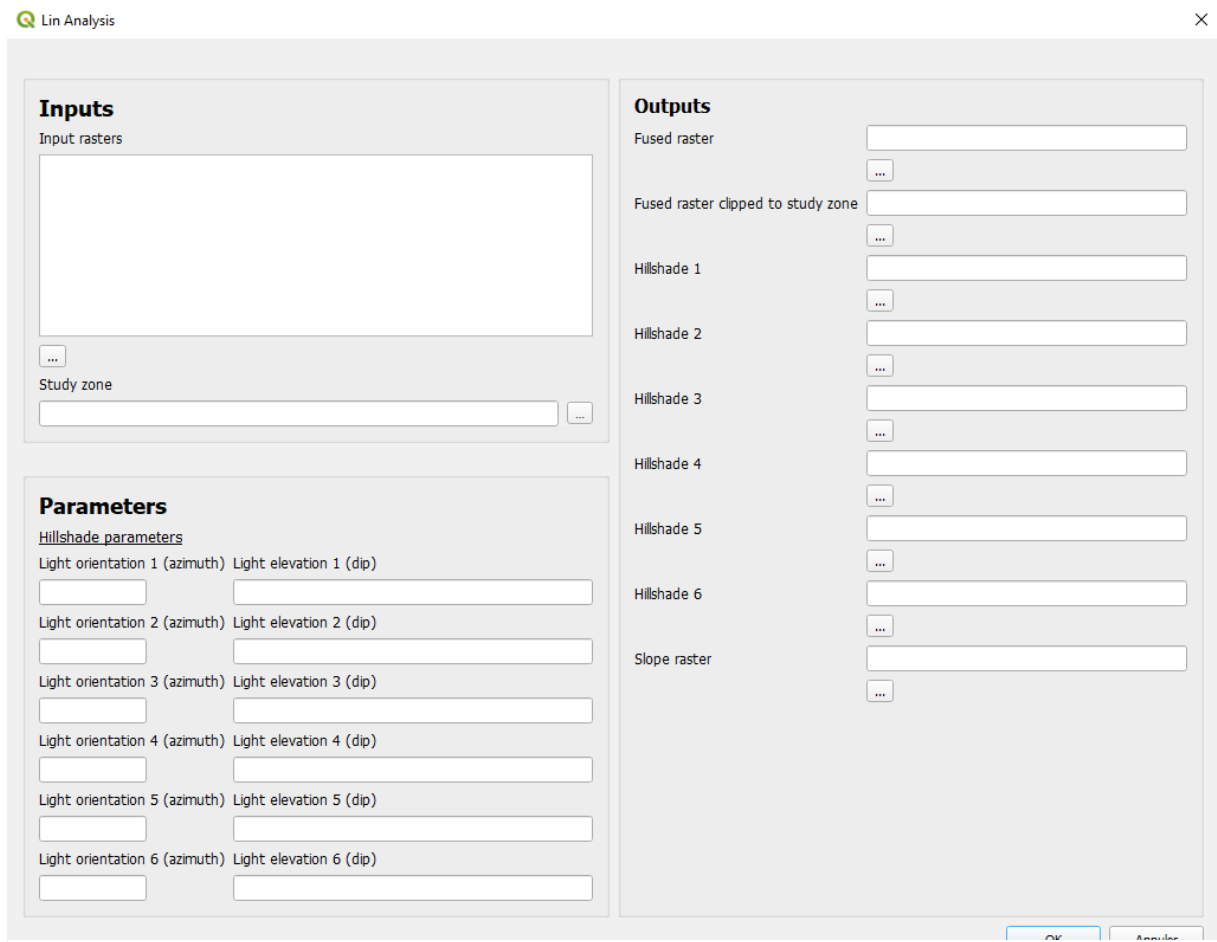
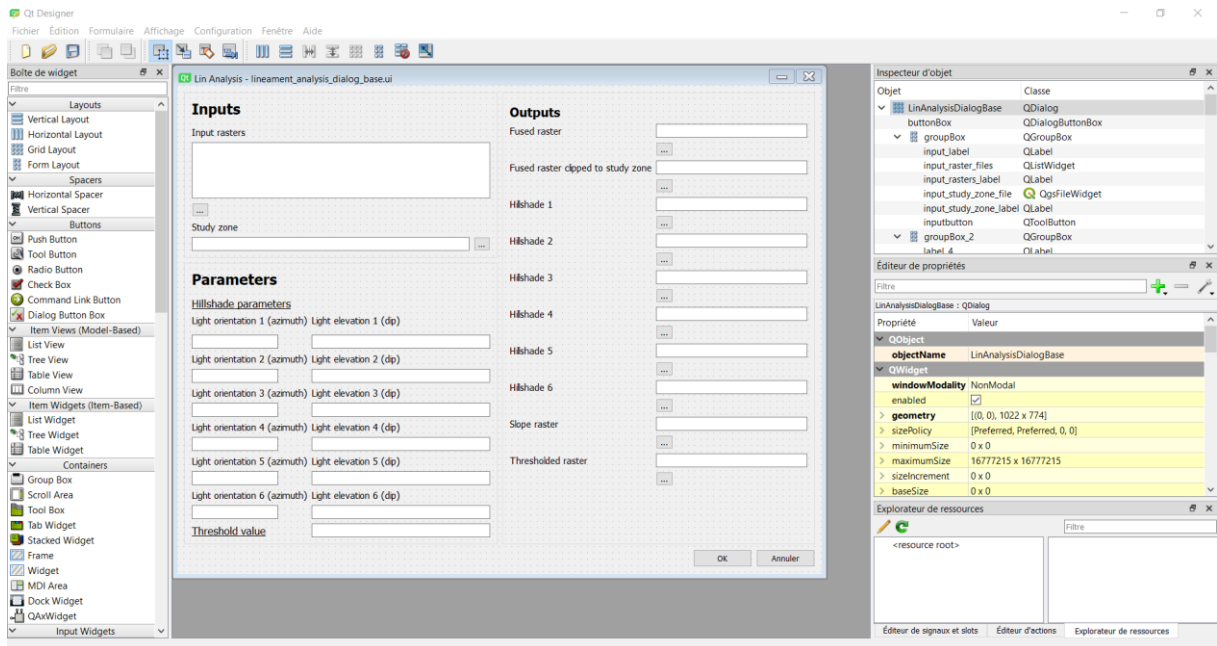


Figure 32 : L'image du haut montre l'interface de Qt Designer avec le fichier `lineament_analysis_dialog_base.ui` ouvert. Les éléments de l'interface ont été rajoutés et ne sont pas générés automatiquement par Plugin Builder. L'image du bas montre à quoi ressemble l'interface de l'outil Lineament Analysis ouvert depuis QGIS 3.18.1.





donner un nom à chaque objet, car c'est ce nom qui sera appelé dans le fichier `lineament_analysis_dialog.py`.

Puis, dans `lineament_analysis_dialog.py`, il faut dire à l'algorithme quoi faire pour chaque élément d'interface. En effet, sans lien d'action à un *widget*, il ne se passe rien lorsque l'on clique dessus dans QGIS. Ce script contient donc toutes les actions utilisateurs et indique à l'algorithme où stocker les fichiers et les valeurs entrées par l'utilisateur. Il crée également une instance de dialogue pour pouvoir lier ce script avec le script `lineament_analysis.py`. L'entièreté du code de dialogue est disponible dans l'annexe 4.

Finalement, `lineament_analysis.py` contient le code Python qui implémente la classe principale du plugin, `LineamentAnalysis`. Cette classe importe d'abord les fichiers d'entrée en fonction de ce qui a été décidé par l'utilisateur depuis le fichier `lineament_analysis_dialog` puis elle performe le traitement dessus. D'abord elle fusionne les tuiles raster MNT avec la fonction `gdal.Warp`, puis elle coupe le raster fusionné aux dimensions de la fenêtre déterminée par le polygone d'entrée choisi par l'utilisateur avec la fonction `gdal.Translate`. Ensuite elle réalise les ombrages avec les valeurs entrées par l'utilisateur avec la fonction `gdal.DEMProcessing`. Un raster de pente est créé également avec la fonction `gdal.DEMProcessing`. Cette dernière étape a été ajoutée afin de rajouter un élément de comparaison pour la digitalisation manuelle des linéaments. A contrario, l'étape de calcul de seuil a finalement été enlevée, car elle manquait de pertinence pour la digitalisation manuelle des linéaments.

Enfin, à chaque étape, les fichiers de sortie sont enregistrés en format TIF sous le nom et dans le dossier déterminé par l'utilisateur. Le code de ce script Python, avec les commentaires dans le code, est disponible en annexe 5.



## ANNEXE 2 : script\_modèle\_preprocessing.py

''''

Date : June 2023

Author : Clothilde Jost

Model exported as python from Graphical Modeler .

Name : Preprocessing

With QGIS : 31801

''''

```
from qgis.core import QgsProcessing
from qgis.core import QgsProcessingAlgorithm
from qgis.core import QgsProcessingMultiStepFeedback
from qgis.core import QgsProcessingParameterNumber
from qgis.core import QgsProcessingParameterMultipleLayers
from qgis.core import QgsProcessingParameterRasterDestination
from qgis.core import QgsProcessingParameterVectorLayer
from qgis.core import QgsCoordinateReferenceSystem
from qgis.core import QgsProcessingParameterRasterLayer, QgsProcessingParameterNumber
import processing
```

```
class Modle(QgsProcessingAlgorithm):
```

```
    THRESHOLD_PARAM = 'threshold'
```

```
    def initAlgorithm(self, config=None):
```

```
        self.addParameter(QgsProcessingParameterRasterLayer('rasters', 'Rasters',
type=QgsProcessing.TypeRasterMulti, defaultValue=None))
```

```
        for i in range(1, 7):
```

```
            self.addParameter(QgsProcessingParameterNumber(f'light{i}_orientation', f'Light {i}
Orientation', QgsProcessingParameterNumber.Double, defaultValue=0))
```

```
            self.addParameter(QgsProcessingParameterNumber(f'light{i}_angle', f'Light {i} Angle',
QgsProcessingParameterNumber.Double, defaultValue=45))
```

```
            self.addParameter(QgsProcessingParameterRasterDestination('Swiss_alti_fusion',
'swiss_alti_fusion', createByDefault=True, defaultValue=None))
```

```
            self.addParameter(QgsProcessingParameterRasterDestination('Ombrage_all_treshold_120',
'ombrage_all_treshold_120', createByDefault=True, defaultValue=None))
```

```
            self.addParameter(QgsProcessingParameterRasterDestination('Dem_clip', 'dem_clip',
createByDefault=True, defaultValue=None))
```

```
            self.addParameter(QgsProcessingParameterVectorLayer('StudyZonePolygon', 'Study Zone
Polygon', types=[QgsProcessing.TypeVectorPolygon]))
```

```
            self.addParameter(QgsProcessingParameterNumber(self.THRESHOLD_PARAM, 'Threshold
Value', QgsProcessingParameterNumber.Double, defaultValue=120))
```

```
    def processAlgorithm(self, parameters, context, model_feedback):
```

```
        # Use a multi-step feedback, so that individual child algorithm progress reports are adjusted for
        # the overall progress through the model
```

```
        feedback = QgsProcessingMultiStepFeedback(10, model_feedback)
```

```
        results = {}
```

```
        outputs = {}
```



```
# Get the list of selected rasters
rasters = self.parameterAsLayerList(parameters, 'rasters', context)
light_sources = []
for i in range(1, 7):
    light_orientation = self.parameterAsDouble(parameters, f'light{i}_orientation', context)
    light_angle = self.parameterAsDouble(parameters, f'light{i}_angle', context)
    light_sources.append({'orientation': light_orientation, 'angle': light_angle})

# Fusion input raster tiles
alg_params = {
    'DATA_TYPE': 5,
    'EXTRA': "",
    'INPUT': rasters,
    'NODATA_INPUT': None,
    'NODATA_OUTPUT': None,
    'OPTIONS': "",
    'PCT': False,
    'SEPARATE': False,
    'OUTPUT': self.parameterAsOutputLayer(parameters, 'Swiss_alti_fusion', context)
}
outputs['Fusion'] = processing.run('gdal:merge', alg_params, context=context,
feedback=feedback, is_child_algorithm=True)
results['Swiss_alti_fusion'] = outputs['Fusion']['OUTPUT']

feedback.setCurrentStep(1)
if feedback.isCanceled():
    return {}

# Clip raster with polygon
alg_params = {
    'INPUT': outputs['Fusion']['OUTPUT'],
    'POLYGONS':
QgsProcessingFeatureSourceDefinition(self.parameterAsVectorLayer(parameters,
'StudyZonePolygon', context).source()),
    'OUTPUT': self.parameterAsOutputLayer(parameters, 'Dem_clip', context)
}
outputs['ClipRasterWithPolygon'] = processing.run('saga:cliprasterwithpolygon', alg_params,
context=context, feedback=feedback, is_child_algorithm=True)
results['Dem_clip'] = outputs['ClipRasterWithPolygon']['OUTPUT']

feedback.setCurrentStep(3)
if feedback.isCanceled():
    return {}

# Ombrage pour les 6 orientations et angles d'incidence
hillshade_outputs = []
for i, light_source in enumerate(light_sources):
    light_orientation = light_source['orientation']
    light_angle = light_source['angle']
```



```
hillshade_params = {
    'INPUT': outputs['ClipRasterWithPolygon']['OUTPUT'],
    'BAND': 1,
    'Z_FACTOR': 1,
    'AZIMUTH': light_orientation,
    'ALTITUDE': light_angle,
    'COMPUTE_EDGES': False,
    'ZEVENBERGEN': False,
    'OUTPUT': self.parameterAsOutputLayer(parameters, f'ombrage_orientation_{i+1}', context),
}
processing.run('qgis:hillshade', hillshade_params, context=context, feedback=feedback)
hillshade_output_path = hillshade_params['OUTPUT']
hillshade_outputs.append(hillshade_output_path)

feedback.setCurrentStep(4)
if feedback.isCanceled():
    return {}

# Save individual hillshades as raster layers
for i, hillshade_output_path in enumerate(hillshade_outputs):
    hillshade_layer_name = f'Hillshade_{i+1}'
    QgsRasterLayer(hillshade_output_path, hillshade_layer_name, 'gdal').write()

# Fusion all hillshades
alg_params = {
    'DATA_TYPE': 5,
    'EXTRA': "",
    'INPUT': hillshade_outputs,
    'NODATA_INPUT': None,
    'NODATA_OUTPUT': None,
    'OPTIONS': "",
    'PCT': False,
    'SEPARATE': False,
    'OUTPUT': self.parameterAsOutputLayer(parameters, 'Ombrage_all', context)
}
processing.run('gdal:merge', alg_params, context=context, feedback=feedback,
is_child_algorithm=True)
fusion_output_path = alg_params['OUTPUT']

# Save fused hillshade as a raster layer
fusion_layer_name = 'Fusion_Hillshade'
QgsRasterLayer(fusion_output_path, fusion_layer_name, 'gdal').write()

# Load the hillshades and the fused hillshade into the project
project = QgsProject.instance()
project.addMapLayers([QgsRasterLayer(path, name) for path, name in zip(hillshade_outputs,
['Hillshade_1', 'Hillshade_2', 'Hillshade_3', 'Hillshade_4', 'Hillshade_5', 'Hillshade_6'])])
project.addMapLayer(QgsRasterLayer(fusion_output_path, fusion_layer_name))
```



```
# Raster calculator
threshold_value = self.parameterAsDouble(parameters, self.THRESHOLD_PARAM, context)
alg_params = {
    'CELLSIZE': 0,
    'CRS': QgsCoordinateReferenceSystem('EPSG:2056'),
    'EXPRESSION': f'"Fusion_Hillshade@1" > {threshold_value}',
    'EXTENT': outputs['ClipRasterWithPolygon']['OUTPUT'],
    'LAYERS': outputs['ClipRasterWithPolygon']['OUTPUT'],
    'OUTPUT': self.parameterAsOutputLayer(parameters, 'Ombrage_all_treshold', context)
}
processing.run('qgis:rastercalculator', alg_params, context=context, feedback=feedback,
is_child_algorithm=True)

feedback.setCurrentStep(10)
if feedback.isCanceled():
    return {}

return results

def name(self):
    return 'Modèle'

def displayName(self):
    return 'Modèle'

def group(self):
    return ""

def groupId(self):
    return ""

def shortHelpString(self):
    return """"<html><body><h2>Description de l'algorithme</h2>
<p></p>
<h2>Paramètres en entrée</h2>
<h3>Angle incidence</h3>
<p></p>
<h3>Orientation source</h3>
<p></p>
<h3>swiss alti3D 0.5m</h3>
<p></p>
<h3>Angle incidence 2</h3>
<p></p>
<h3>Orientation source 2</h3>
<p></p>
<h3>Orientation source 3</h3>
<p></p>
<h3>Orientation source 4</h3>
<p></p>
"""
```



```
<h3>Orientation source 5</h3>
<p></p>
<h3>Orientation source 6</h3>
<p></p>
<h3>swiss_alti_fusion</h3>
<p></p>
<h3>ombrage_all</h3>
<p></p>
<h3>ombrage_orientation1</h3>
<p></p>
<h3>ombrage_orientation2</h3>
<p></p>
<h3>ombrage_orientation3</h3>
<p></p>
<h3>ombrage_orientation4</h3>
<p></p>
<h3>ombrage_orientation5</h3>
<p></p>
<h3>ombrage_orientation6</h3>
<p></p>
<h3>ombrage_all_treshold_120</h3>
<p></p>
<h3>dem_clip</h3>
<p></p>
<h3>polygone_zone_etude</h3>
<p></p>
<h2>Sorties</h2>
<h3>swiss_alti_fusion</h3>
<p></p>
<h3>ombrage_all</h3>
<p></p>
<h3>ombrage_orientation1</h3>
<p></p>
<h3>ombrage_orientation2</h3>
<p></p>
<h3>ombrage_orientation3</h3>
<p></p>
<h3>ombrage_orientation4</h3>
<p></p>
<h3>ombrage_orientation5</h3>
<p></p>
<h3>ombrage_orientation6</h3>
<p></p>
<h3>ombrage_all_treshold_120</h3>
<p></p>
<h3>dem_clip</h3>
<p></p>
<br></body></html>""
def createInstance(self):
```



return Modle()



## ANNEXE 3 : \_\_init\_\_.py

```
# -*- coding: utf-8 -*-
"""
/*****

LineamentAnalysis
    A QGIS plugin
This plugin prepares and analyse raster data for lineament interpretation
Generated by Plugin Builder: http://g-sherman.github.io/Qgis-Plugin-Builder/
-----
begin      : 2023-07-03
copyright  : (C) 2023 by CLJ
email      : clothilde.jost@gmail.com
git sha    : $Format:%H$

*****/

/*****
*
* 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.
*

*****/

This script initializes the plugin, making it known to QGIS.
"""

# noinspection PyPep8Naming
def classFactory(iface): # pylint: disable=invalid-name
    """Load LineamentAnalysis class from file LineamentAnalysis.

    :param iface: À QGIS interface instance.
    :type iface: QgsInterface
    """
    #
    from .lineament_analysis import LineamentAnalysis
    return LineamentAnalysis(iface)
```





## ANNEXE 4 : lineament\_analysis\_dialog.py

```
# -*- coding: utf-8 -*-
"""
/*****

LineamentAnalysisDialog
    A QGIS plugin
This plugin prepares raster data for lineament interpretation
Generated by Plugin Builder: http://g-sherman.github.io/Qgis-Plugin-Builder/
-----
begin          : 2023-07-03
git sha        : $Format:%H$
copyright      : (C) 2023 by CLJ
email          : clothilde.jost@gmail.com

*****/

/*****
*
* 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.                                   *
*
*
*****/

"""

import os

from qgis.PyQt import QtCore
from PyQt5.QtCore import pyqtSignal
from qgis.PyQt.QtCore import Qt
from qgis.PyQt import QtWidgets
from qgis.PyQt.QtGui import QIcon
from qgis.PyQt import uic
from qgis.PyQt.QtWidgets import QDialog, QDialogBox, QListWidgetItem,
QDialogButtonBox, QProgressDialog, QTableWidgetItem
from qgis.core import QgsRasterLayer
from qgis.core import QgsProcessingMultiStepFeedback
from qgis.core import QgsProcessingContext, QgsProcessingMultiStepFeedback,
QgsProcessingFeatureSourceDefinition, QgsRasterLayer
from qgis import processing
from qgis.core import QgsProcessingAlgorithm
import lineament_analysis
```



# This loads your .ui file so that PyQt can populate your plugin with the elements from Qt Designer

```
FORM_CLASS, _ = uic.loadUiType(os.path.join(
    os.path.dirname(__file__), 'lineament_analysis_dialog_base.ui'))
```

```
class LineamentAnalysisDialog(QtWidgets.QDialog, FORM_CLASS):
```

```
    def __init__(self, parent=None):
```

```
        """Constructor."""
```

```
        super(LineamentAnalysisDialog, self).__init__(parent)
```

```
        # Set up the user interface from Designer through FORM_CLASS.
```

```
        # ...
```

```
        self.setupUi(self)
```

```
        self.file_dialog = QFileDialog() # Create a QFileDialog instance
```

```
        #INPUTS
```

```
        #Create an empty list or dictionary for the input rasters:
```

```
        #self.raster_files = []
```

```
        #Defin the file_paths attribute
```

```
        self.file_paths = []
```

```
        # Connect the button signals to the input list
```

```
        self.inputbutton.clicked.connect(self.selectRasterFiles)
```

```
        #OUTPUTS
```

```
        # Create new instances variables to hold the file paths for the outputs files:
```

```
        self.fus_rast_file_path = ""
```

```
        self.fus_rast_clip_file_path = ""
```

```
        self.hill_1_file_path = ""
```

```
        self.hill_2_file_path = ""
```

```
        self.hill_3_file_path = ""
```

```
        self.hill_4_file_path = ""
```

```
        self.hill_5_file_path = ""
```

```
        self.hill_6_file_path = ""
```

```
        self.slope_file_path = ""
```

```
        # Connect the button signals to the output list
```

```
        self.fus_rast_button.clicked.connect(self.selectFusRastFile)
```

```
        self.fus_rast_clip_button.clicked.connect(self.selectFusRastClipFile)
```

```
        self.hill_1_button.clicked.connect(self.selectHill1File)
```

```
        self.hill_2_button.clicked.connect(self.selectHill2File)
```

```
        self.hill_3_button.clicked.connect(self.selectHill3File)
```

```
        self.hill_4_button.clicked.connect(self.selectHill4File)
```

```
        self.hill_5_button.clicked.connect(self.selectHill5File)
```



```
self.hill_6_button.clicked.connect(self.selectHill6File)  
self.slope_button.clicked.connect(self.selectSlopeFile)
```

```
#INPUTS-----
```

```
def selectRasterFiles(self):  
    file_dialog = QFileDialog()  
    file_dialog.setFileMode(QFileDialog.ExistingFiles)  
    file_dialog.setOption(QFileDialog.DontUseNativeDialog)  
    file_dialog.setNameFilter("*.tif *.XYZ")  
  
    if file_dialog.exec_():  
        selected_files = file_dialog.selectedFiles()  
        if selected_files: # Check if any files were selected  
            self.file_paths = [] # Update the file_paths attribute  
            for file_path in selected_files: # Iterate over the selected files  
                self.file_paths.append(file_path)  
                # Add each file path to the QListWidget  
                item = QListWidgetItem(file_path)  
                self.input_raster_files.addItem(item)  
            self.prepareData()
```

```
#OUTPUTS-----
```

```
# Handle the selected file for fused raster  
def selectFusRastFile(self):  
    self.showFileDialog(self.fus_rast_file, "Fused Raster")  
  
# Handle the selected file for clip fused raster  
def selectFusRastClipFile(self):  
    self.showFileDialog(self.fus_rast_clip_file, "Clip Fused Raster")  
  
# Handle the selected file for hillshade 1  
def selectHill1File(self):  
    self.showFileDialog(self.hill_1_file, "Hillshade 1")  
  
# Handle the selected file for hillshade 2  
def selectHill2File(self):  
    self.showFileDialog(self.hill_2_file, "Hillshade 2")  
  
# Handle the selected file for hillshade 3  
def selectHill3File(self):  
    self.showFileDialog(self.hill_3_file, "Hillshade 3")  
  
# Handle the selected file for hillshade 4  
def selectHill4File(self):  
    self.showFileDialog(self.hill_4_file, "Hillshade 4")
```



```
# Handle the selected file for hillshade 5
def selectHill5File(self):
    self.showFileDialog(self.hill_5_file, "Hillshade 5")

# Handle the selected file for hillshade 6
def selectHill6File(self):
    self.showFileDialog(self.hill_6_file, "Hillshade 6")

# Handle the selected file for hillshade fused
def selectSlopeFile(self):
    self.showFileDialog(self.slope_file, "Slope")

#Update the empty line in the widget to show the name and path selected
def showFileDialog(self, file_widget, title):
    file_dialog = QFileDialog()
    file_dialog.setFileMode(QFileDialog.AnyFile)
    file_dialog.setAcceptMode(QFileDialog.AcceptSave)
    file_dialog.setOption(QFileDialog.DontUseNativeDialog)
    file_dialog.setWindowTitle(title)
    file_dialog.setDefaultSuffix(".tif")
    file_dialog.setNameFilter("*.tif")

    if file_dialog.exec_():
        selected_output_files = file_dialog.selectedFiles()
        file_path = selected_output_files[0]
        file_widget.setFilePath(file_path) # Update the QgsFileWidget with the file path

# Update the corresponding file path variable based on the file_widget
if file_widget == self.fus_rast_file:
    self.fus_rast_file_path = file_path
elif file_widget == self.fus_rast_clip_file:
    self.fus_rast_clip_file_path = file_path
elif file_widget == self.hill_1_file:
    self.hill_1_file_path = file_path
elif file_widget == self.hill_2_file:
    self.hill_2_file_path = file_path
elif file_widget == self.hill_3_file:
    self.hill_3_file_path = file_path
elif file_widget == self.hill_4_file:
    self.hill_4_file_path = file_path
elif file_widget == self.hill_5_file:
    self.hill_5_file_path = file_path
elif file_widget == self.hill_6_file:
    self.hill_6_file_path = file_path
```



```
elif file_widget == self.slope_file:  
    self.slope_file_path = file_path
```

```
#Prepare the data for the Run method in lineament_analysis.py
```

```
def prepareData(self):
```

```
    #Get the values from the UI components
```

```
    input_raster_files = self.file_paths
```

```
    input_study_zone_file = self.input_study_zone_file
```

```
    light_or_1_value = self.light_or_1_value.text()
```

```
    light_or_2_value = self.light_or_2_value.text()
```

```
    light_or_3_value = self.light_or_3_value.text()
```

```
    light_or_4_value = self.light_or_4_value.text()
```

```
    light_or_5_value = self.light_or_5_value.text()
```

```
    light_or_6_value = self.light_or_6_value.text()
```

```
    light_el_1_value = self.light_el_1_value.text()
```

```
    light_el_2_value = self.light_el_2_value.text()
```

```
    light_el_3_value = self.light_el_3_value.text()
```

```
    light_el_4_value = self.light_el_4_value.text()
```

```
    light_el_5_value = self.light_el_5_value.text()
```

```
    light_el_6_value = self.light_el_6_value.text()
```

```
    fus_rast_file = self.fus_rast_file_path
```

```
    fus_rast_clip_file = self.fus_rast_clip_file_path
```

```
    hill_1_file = self.hill_1_file_path
```

```
    hill_2_file = self.hill_2_file_path
```

```
    hill_3_file = self.hill_3_file_path
```

```
    hill_4_file = self.hill_4_file_path
```

```
    hill_5_file = self.hill_5_file_path
```

```
    hill_6_file = self.hill_6_file_path
```

```
    slope_file = self.slope_file_path
```

```
#Return the values directly
```

```
return (
```

```
    input_raster_files,
```

```
    input_study_zone_file,
```

```
    light_or_1_value,
```

```
    light_or_2_value,
```

```
    light_or_3_value,
```

```
    light_or_4_value,
```

```
    light_or_5_value,
```

```
    light_or_6_value,
```

```
    light_el_1_value,
```

```
    light_el_2_value,
```

```
    light_el_3_value,
```

```
    light_el_4_value,
```

```
    light_el_5_value,
```



```
light_el_6_value,  
fus_rast_file,  
fus_rast_clip_file,  
hill_1_file,  
hill_2_file,  
hill_3_file,  
hill_4_file,  
hill_5_file,  
hill_6_file,  
slope_file  
)  
  
# Validate the file paths  
if not all([input_raster_files, input_study_zone_file, fus_rast_file, fus_rast_clip_file]):  
    raise ValueError("One or more file paths are empty in the dialog.")
```



## ANNEXE 5 : lineament\_analysis.py

```
# -*- coding: utf-8 -*-
"""
/*****

LineamentAnalysis
    A QGIS plugin
This plugin prepares raster data for lineament interpretation
Generated by Plugin Builder: http://g-sherman.github.io/Qgis-Plugin-Builder/
-----
begin          : 2023-07-03
git sha        : $Format:%H$
copyright      : (C) 2023 by CLJ
email          : clothilde.jost@gmail.com

*****/

/*****
*
* 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.
*

*****/
"""
from qgis.PyQt.QtCore import QSettings, QTranslator, QCoreApplication
from qgis.PyQt.QtGui import QIcon
from qgis.PyQt.QtWidgets import QAction
from PyQt5.QtWidgets import QMessageBox
from qgis.core import (
    QgsProcessing,
    QgsProcessingAlgorithm,
    QgsProcessingParameterRasterLayer,
    QgsProcessingParameterVectorLayer,
    QgsProcessingParameterBoolean,
    QgsProcessingParameterNumber,
    QgsProcessingParameterString,
    QgsProcessingParameterFolderDestination,
    QgsProcessingParameterFileDestination,
    QgsProcessingOutputRasterLayer,
    QgsProcessingFeedback,
    QgsProcessingContext,
)
# Import the processing module
```



```
import processing
import subprocess
# Import modules for gdal
import numpy as np
import tempfile
import os
import subprocess
from osgeo import gdal, ogr, osr
import math
# Initialize Qt resources from file resources.py
from .resources import *
# Import the code for the dialog
from .lineament_analysis_dialog import LineamentAnalysisDialog
import os.path
```

```
class LineamentAnalysis:
    """QGIS Plugin Implementation."""

    def __init__(self, iface):
        """Constructor.

        :param iface: An interface instance that will be passed to this class
            which provides the hook by which you can manipulate the QGIS
            application at run time.
        :type iface: QgsInterface
        """
        # Save reference to the QGIS interface
        self.iface = iface
        # initialize plugin directory
        self.plugin_dir = os.path.dirname(__file__)
        # initialize locale
        locale = QSettings().value('locale/userLocale')[0:2]
        locale_path = os.path.join(
            self.plugin_dir,
            'i18n',
            'LineamentAnalysis_{}.qm'.format(locale))

        if os.path.exists(locale_path):
            self.translator = QTranslator()
            self.translator.load(locale_path)
            QApplication.installTranslator(self.translator)

        # Declare instance attributes
```





```
self.actions = []
self.menu = self.tr(u'&Lineament Analysis')

# Check if plugin was started the first time in current QGIS session
# Must be set in initGui() to survive plugin reloads
self.first_start = None

# noinspection PyMethodMayBeStatic
def tr(self, message):
    """Get the translation for a string using Qt translation API.

    We implement this ourselves since we do not inherit QObject.

    :param message: String for translation.
    :type message: str, QString

    :returns: Translated version of message.
    :rtype: QString
    """
    # noinspection PyTypeChecker,PyArgumentList,PyCallByClass
    return QApplication.translate('LineamentAnalysis', message)

def add_action(
    self,
    icon_path,
    text,
    callback,
    enabled_flag=True,
    add_to_menu=True,
    add_to_toolbar=True,
    status_tip=None,
    whats_this=None,
    parent=None):
    """Add a toolbar icon to the toolbar.

    :param icon_path: Path to the icon for this action. Can be a resource
        path (e.g. ':/plugins/foo/bar.png') or a normal file system path.
    :type icon_path: str

    :param text: Text that should be shown in menu items for this action.
    :type text: str

    :param callback: Function to be called when the action is triggered.
    :type callback: function
```



```
:param enabled_flag: A flag indicating if the action should be enabled
    by default. Defaults to True.
:type enabled_flag: bool

:param add_to_menu: Flag indicating whether the action should also
    be added to the menu. Defaults to True.
:type add_to_menu: bool

:param add_to_toolbar: Flag indicating whether the action should also
    be added to the toolbar. Defaults to True.
:type add_to_toolbar: bool

:param status_tip: Optional text to show in a popup when mouse pointer
    hovers over the action.
:type status_tip: str

:param parent: Parent widget for the new action. Defaults None.
:type parent: QWidget

:param whats_this: Optional text to show in the status bar when the
    mouse pointer hovers over the action.

:returns: The action that was created. Note that the action is also
    added to self.actions list.
:rtype: QAction
"""

icon = QIcon(icon_path)
action = QAction(icon, text, parent)
action.triggered.connect(callback)
action.setEnabled(enabled_flag)

if status_tip is not None:
    action.setStatusTip(status_tip)

if whats_this is not None:
    action.setWhatsThis(whats_this)

if add_to_toolbar:
    # Adds plugin icon to Plugins toolbar
    self.iface.addToolBarIcon(action)

if add_to_menu:
    self.iface.addPluginToRasterMenu(
```



```
        self.menu,
        action)

self.actions.append(action)

return action

def initGui(self):
    """Create the menu entries and toolbar icons inside the QGIS GUI."""

    icon_path = '/plugins/lineament_analysis/icon.png'
    self.add_action(
        icon_path,
        text=self.tr(u'Lineament Analysis'),
        callback=self.run,
        parent=self.iface.mainWindow())

    # will be set False in run()
    self.first_start = True

def unload(self):
    """Removes the plugin menu item and icon from QGIS GUI."""
    for action in self.actions:
        self.iface.removePluginRasterMenu(
            self.tr(u'&Lineament Analysis'),
            action)
        self.iface.removeToolBarIcon(action)

def run(self):
    # Create an instance of LineamentAnalysisDialog
    dialog = LineamentAnalysisDialog()

    """Run method that performs all the real work"""

    # Create the dialog with elements (after translation) and keep reference
    # Only create GUI ONCE in callback, so that it will only load when the plugin is started
    if self.first_start == True:
        self.first_start = False
        dialog = LineamentAnalysisDialog()

    # show the dialog
    dialog.show()
```



```
# Run the dialog and retrieve the values
result = dialog.exec_()
# Assign the values using getter methods of the dialog object
try:
    input_raster_files = [dialog.input_raster_files.item(i).text() for i in
range(dialog.input_raster_files.count())]
    input_study_zone_file = dialog.input_study_zone_file.filePath()
    light_or_values = [
        dialog.light_or_1_value.text(),
        dialog.light_or_2_value.text(),
        dialog.light_or_3_value.text(),
        dialog.light_or_4_value.text(),
        dialog.light_or_5_value.text(),
        dialog.light_or_6_value.text()
    ]
    light_el_values = [
        dialog.light_el_1_value.text(),
        dialog.light_el_2_value.text(),
        dialog.light_el_3_value.text(),
        dialog.light_el_4_value.text(),
        dialog.light_el_5_value.text(),
        dialog.light_el_6_value.text()
    ]
    fus_rast_file = dialog.fus_rast_file.filePath()
    fus_rast_clip_file = dialog.fus_rast_clip_file.filePath()
    hill_files = [
        dialog.hill_1_file.filePath(),
        dialog.hill_2_file.filePath(),
        dialog.hill_3_file.filePath(),
        dialog.hill_4_file.filePath(),
        dialog.hill_5_file.filePath(),
        dialog.hill_6_file.filePath()
    ]
    slope_file = dialog.slope_file.filePath()

# Validate the file paths
if not all([input_raster_files, input_study_zone_file, fus_rast_file, fus_rast_clip_file]):
    raise ValueError("One or more file paths are empty.")
if input_raster_files is None:
    raise ValueError("Missing 'rasters' parameter in the input.")
#if light_or_1_value is None:
#raise ValueError("Missing 'light_or_1_value' parameter in the input.")

#Fusion of the rasters-----
# Open the input raster datasets
```



```
input_layers = []
for file_path in input_raster_files:
    dataset = gdal.Open(file_path)
    if dataset is not None:
        input_layers.append(dataset)
    else:
        print(f"Error opening raster file: {file_path}")

if len(input_layers) == 0:
    raise ValueError("nothing to see...")

# Merge the raster files using gdal.Warp()
merged_dataset = gdal.Warp(fus_rast_file, input_layers, dstSRS='EPSG:2056')
#####PARAMETRE SYS. PROJ. : Changer le numéro EPSG pour changer le
système de projection.

# Close the input datasets
for dataset in input_layers:
    dataset = None

# Check if the merged file exists
if not os.path.exists(fus_rast_file):
    raise ValueError("Fusion raster file not found.")

# Clip raster with polygon-----
# Open the merged raster dataset
input_dataset = gdal.Open(fus_rast_file)
if input_dataset is None:
    raise ValueError("Error opening input raster file")

# Validate the file path
if not input_study_zone_file:
    raise ValueError("Study zone file path is empty.")

# Open the study zone polygon
study_zone_layer_data_source = ogr.Open(input_study_zone_file)
if study_zone_layer_data_source is None:
    raise ValueError("Error opening study zone raster mask")

# Get the study zone layer from the data source
study_zone_layer = study_zone_layer_data_source.GetLayer()

# Get the four summit coordinates of the polygon
polygon_feature = study_zone_layer.GetFeature(0)
polygon_geometry = polygon_feature.GetGeometryRef()
```



```

polygon_extent = polygon_geometry.GetEnvelope()
x_min, x_max, y_min, y_max = polygon_extent

# Specify the window based on the polygon extent
proj_win = [x_min, y_max, x_max, y_min]

# Use gdal.Translate to create the clipped raster
clipped_dataset = gdal.Translate(fus_rast_clip_file, input_dataset, projWin=proj_win)

# Check if the clipped file exists
if not os.path.exists(fus_rast_clip_file):
    raise ValueError("Clipped raster file not found.")

# Hillshade for 6 orientations and angles of incidence-----
for i in range(len(light_or_values)):
    light_or_value = light_or_values[i]
    light_el_value = light_el_values[i]

    # Get the output hillshade file path
    output_hillshade_file = hill_files[i]

    # Calculate hillshade using GDAL's built-in function
    hillshade = gdal.DEMProcessing(output_hillshade_file, fus_rast_clip_file, 'hillshade',
    computeEdges=False, azimuth=light_or_value, altitude=light_el_value)

    # Save the hillshade raster to a file
    gdal.Translate(output_hillshade_file, output_hillshade_file)

    # Clear the hillshade array from memory
    hillshade = None

# Slope raster-----
input_raster_file = fus_rast_clip_file
output_slope_file = slope_file

# Calculate slope using GDAL's gdaldem algorithm
slope = gdal.DEMProcessing(output_slope_file, input_raster_file, 'slope')

# Clear the slope array from memory
slope = None

except Exception as e:
    # Display an error message dialog
    error_message = "An error occurred: " + str(e)

```



```
QMessageBox.critical(None, "Error", error_message)

# Print the error to the console
print(error_message)

# Return an empty dictionary or None to indicate failure
return {}

return result

def createInstance(self):
    return MyTool()

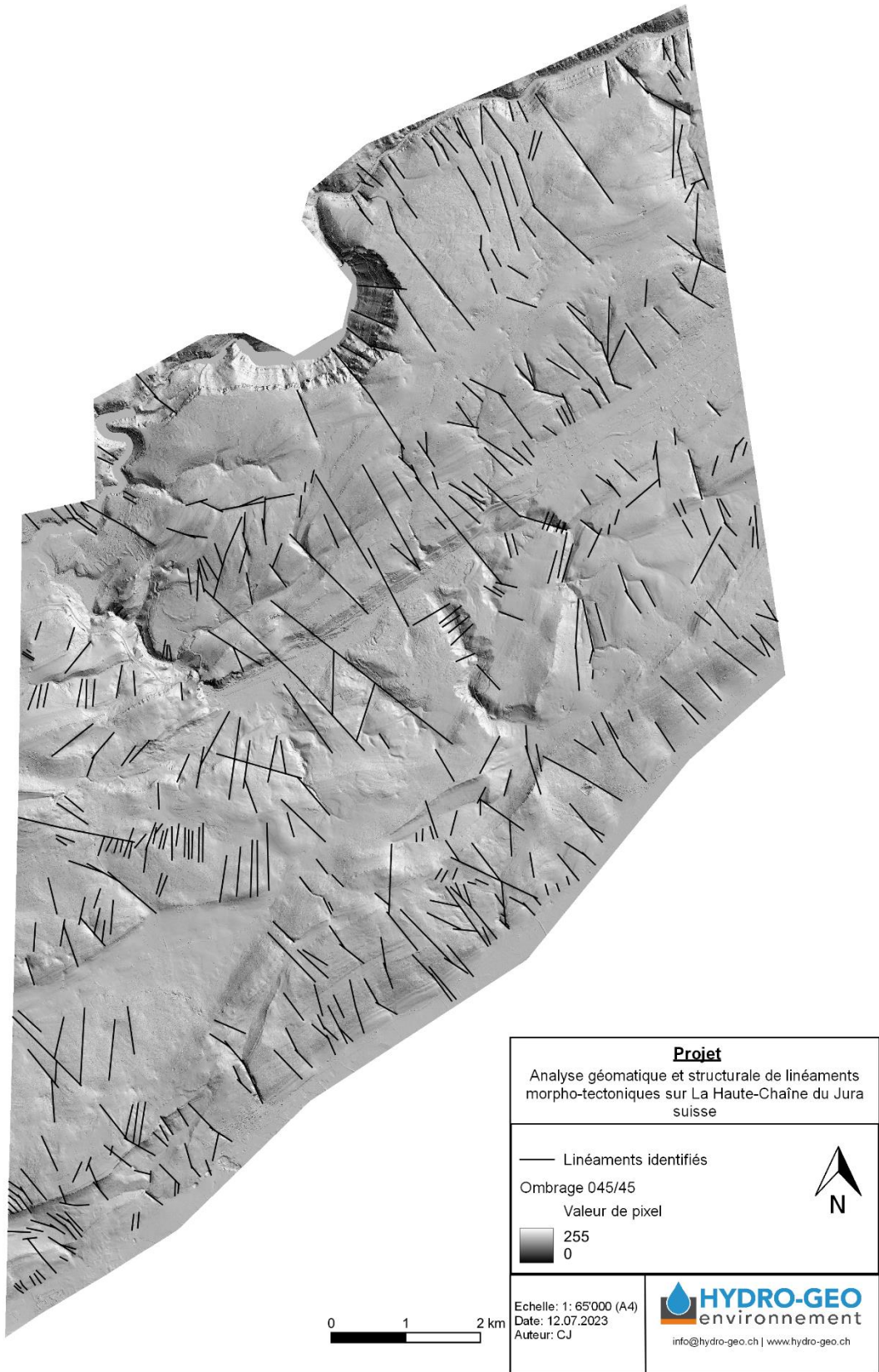
def group(self):
    return ""

def groupId(self):
    return ""

def shortHelpString(self):
    return ""
```



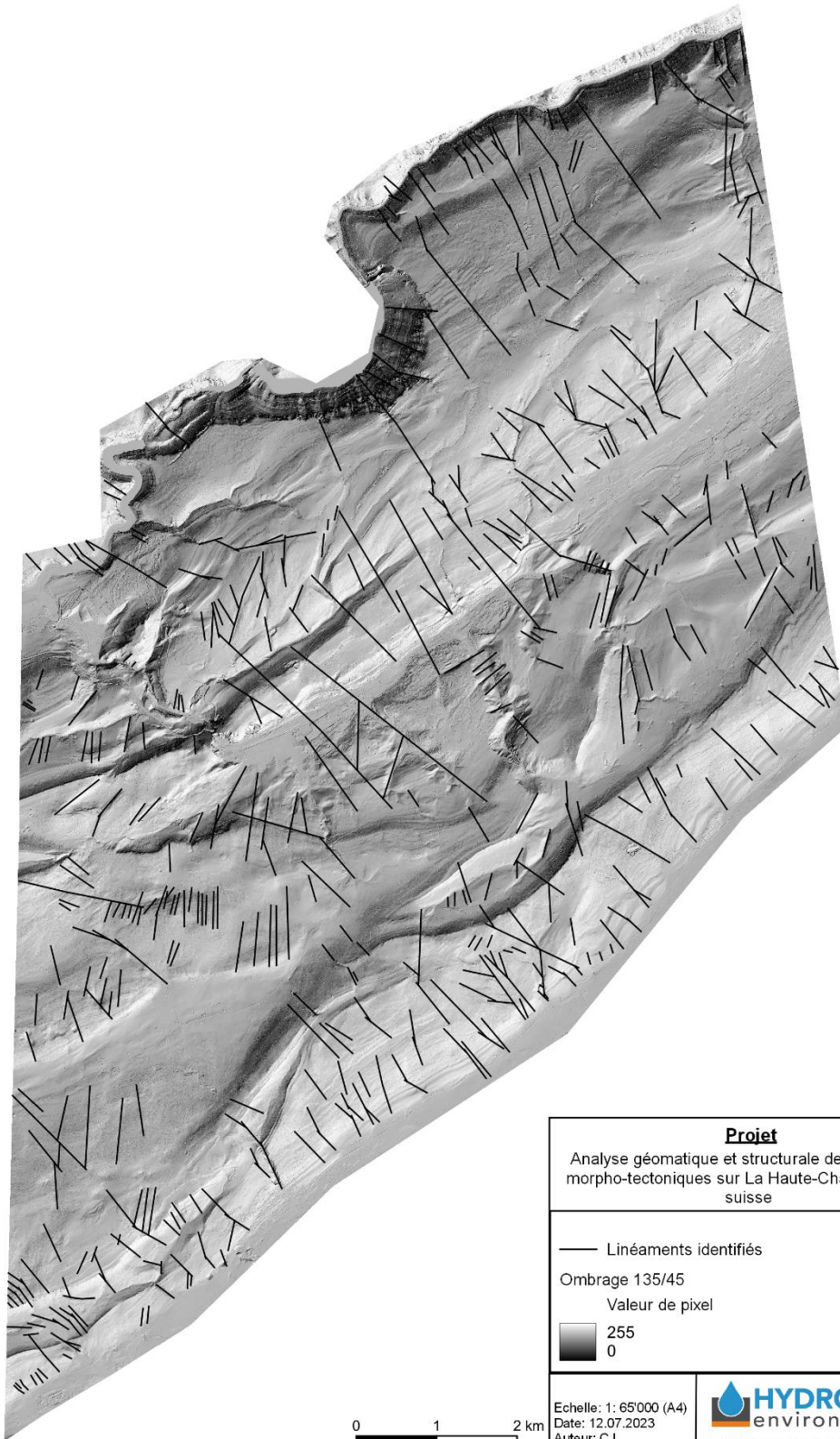
## ANNEXE 6 : Carte d'ombrage 045/45 avec les linéaments du Locle





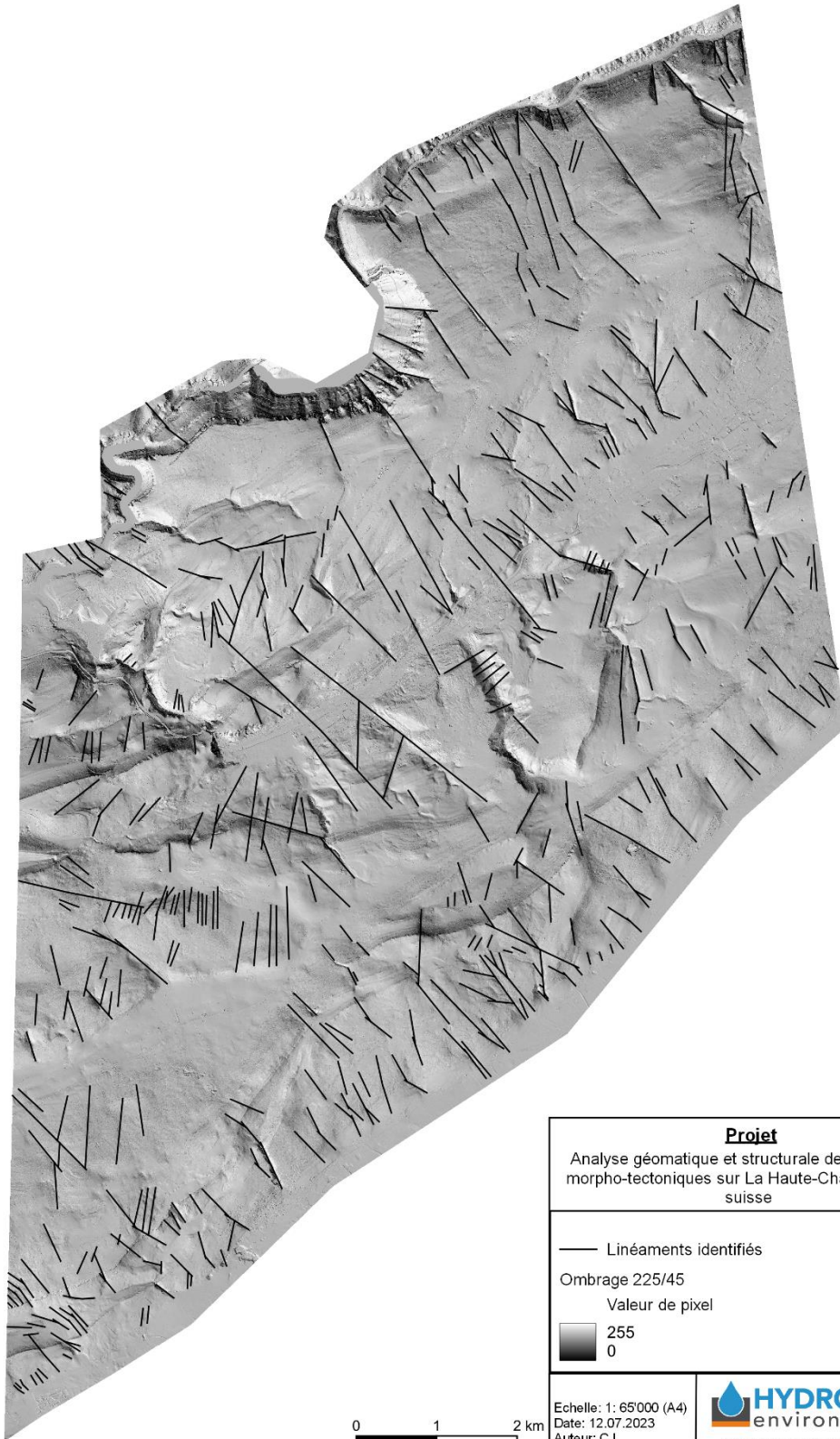


## ANNEXE 7 : Carte d'ombrage 135/45 avec les linéaments du Locle



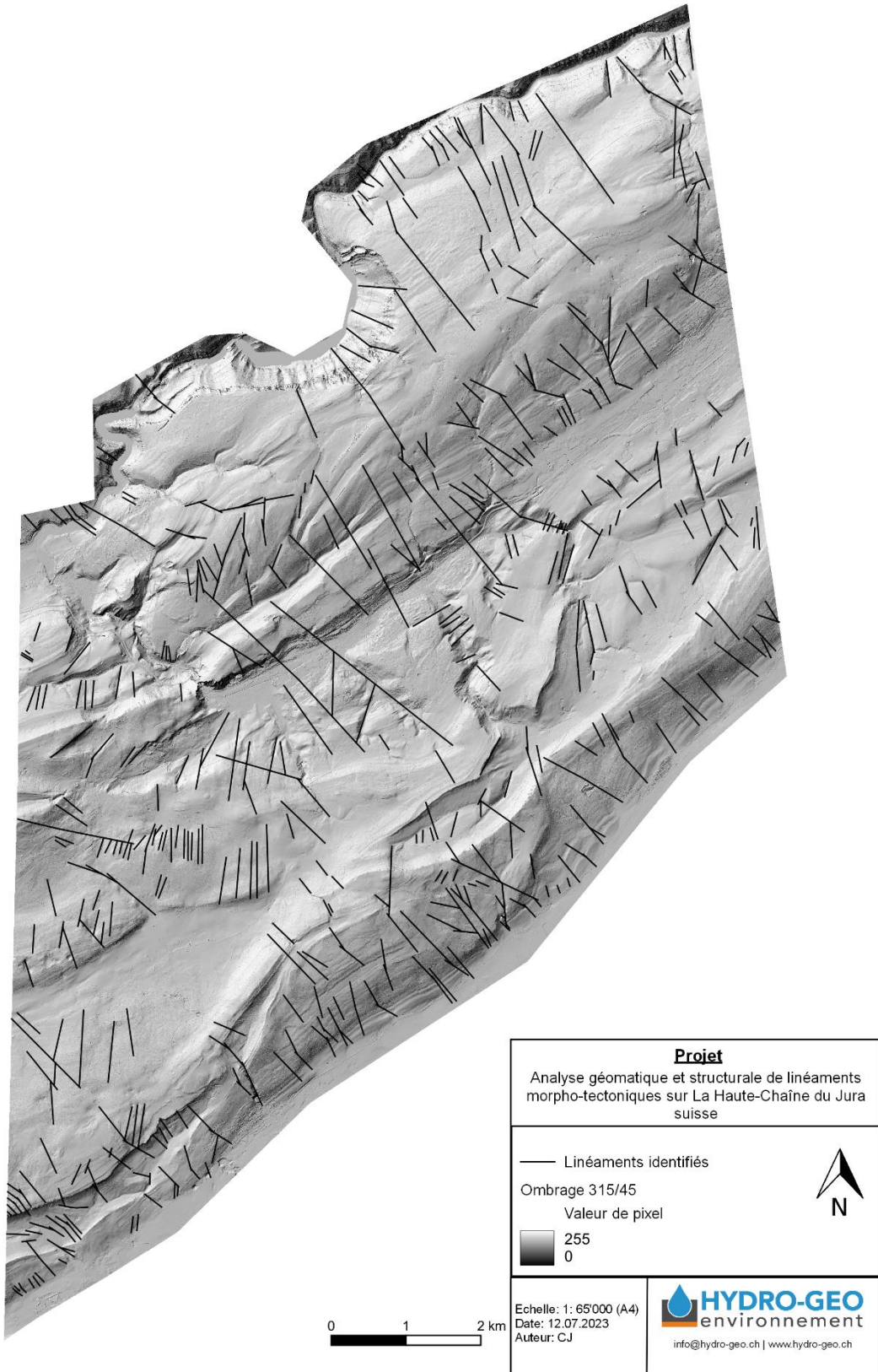


## ANNEXE 8 : Carte d'ombrage 225/45 avec les linéaments du Locle



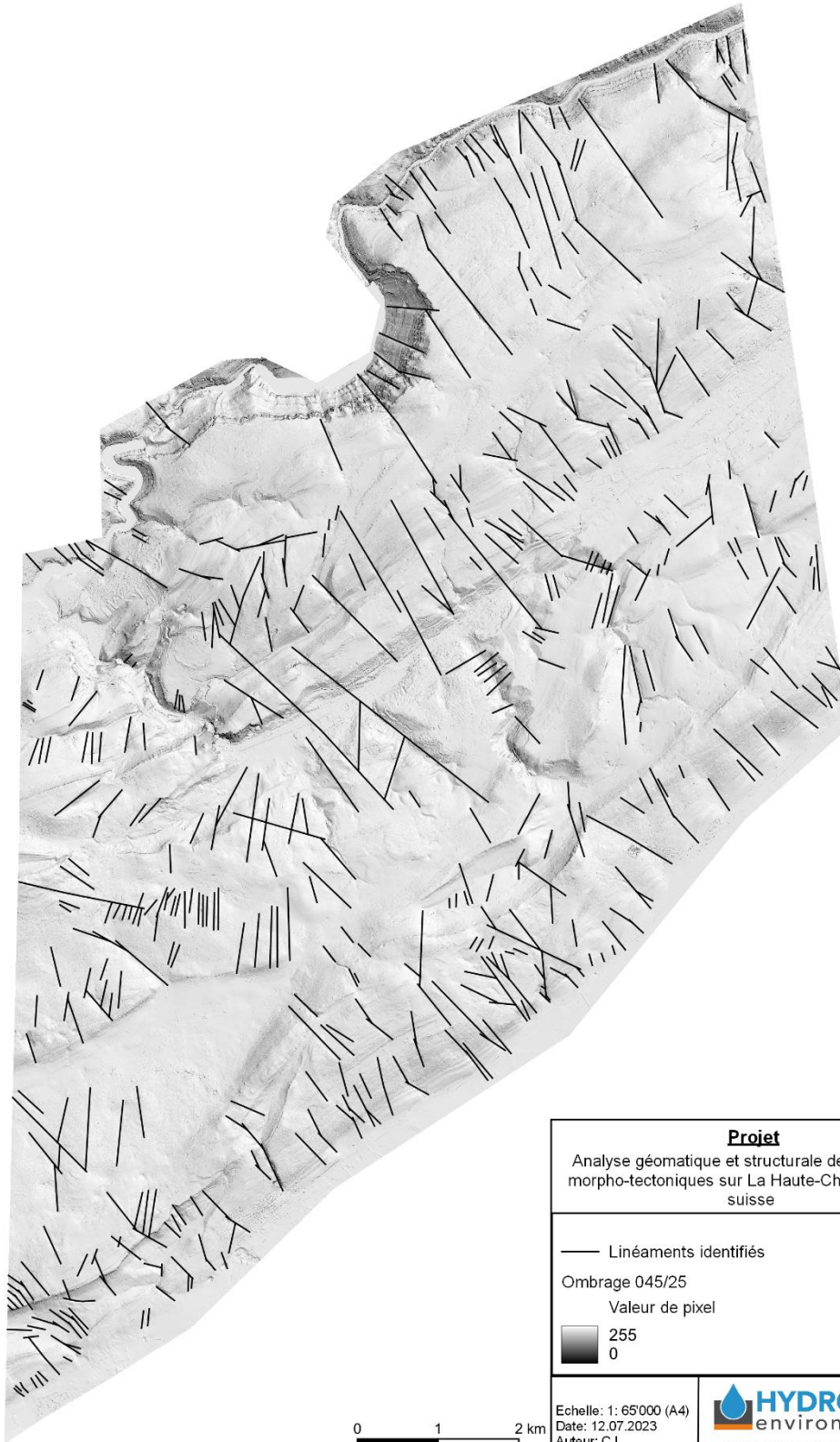


## ANNEXE 9 : Carte d'ombrage 315/45 avec les linéaments du Locle





## ANNEXE 10 : Carte d'ombrage 045/25 avec les linéaments du Locle





## ANNEXE 11 : Carte d'ombrage 225/25 avec les linéaments du Locle

