

Table des matières

| | |
|--|-----------|
| Introduction..... | 4 |
| Contexte général..... | 4 |
| Données et méthodologie..... | 7 |
| Description des données..... | 7 |
| Mise à jour de la carte..... | 10 |
| Développement d'un script python pour faire une différence de moyenne sur une période donnée..... | 18 |
| Fonctionnement général du script..... | 18 |
| Représentation schématique du fonctionnement de l'outil..... | 19 |
| Présentation d'un ModelBuilder simplifié..... | 20 |
| Comparaison du script par rapport au ModelBuilder..... | 21 |
| Description des outils utilisés dans le script python..... | 21 |
| Présentation du code..... | 24 |
| Création de l'outil et rédaction de l'aide..... | 32 |
| Présentation de l'outil et exemples d'utilisation..... | 36 |
| Conclusion et pistes de développement..... | 38 |
| Bibliographie..... | 39 |
| Logiciels utilisés..... | 39 |
| Annexes..... | 40 |
| Annexe 1..... | 41 |
| Annexe 2..... | 42 |
| Annexe 3..... | 52 |
| Annexe 4..... | 53 |
| Annexe 5..... | 54 |

Introduction

Ce travail a été effectué dans le cadre du Certificat de géomatique de l'Université de Genève, ce document est un rapport de stage de fin de formation. Le stage a eu lieu dans le groupe sols et substrats à l'hepia (haute école du paysage, d'ingénierie et d'architecture de Genève), sur le site de Lullier. Le groupe sols et substrats fait partie de l'inTNE (institut Terre Nature Environnement) de l'hepia.

Les objectifs du stage étaient de mettre à jour les données des analyses de sol cultivés à Genève ainsi que la carte des surfaces agricoles relative. Une amélioration de la base de données a aussi été proposée. Pour terminer, un outil de comparaison des données et de représentation sous forme de carte a été développé.

Les responsables du stage étaient Frédéric Lamy et Alice Johannes. Le représentant hepia et membre du jury est Alain Dubois, le représentant UNIGE et membre du jury est Anthony Lehmann. Le troisième membre du jury est Antoine Besson, chargé d'enseignement en agronomie, à l'hepia.

Contexte général

Selon l'article 13 de l'OPD (ordonnance sur les paiements directs) [1], les apports par les agriculteurs en azote et en phosphore ne doivent pas être excédentaires. Les engrais doivent donc être répartis selon les besoins des cultures et en tenant compte des taux déjà présents dans le sol. Afin de connaître ces taux, les agriculteurs sont tenus de faire des analyses de sol au moins tous les 10 ans, pour toutes leurs parcelles (pour les exceptions, voir OPD).

Suivant une méthode standardisée, les agriculteurs prélèvent à l'aide d'une tarière une quinzaine d'échantillons sur chacune de leurs parcelles, de manière à ce que l'ensemble des échantillons soit représentatif. Ces échantillons sont analysés au laboratoire de la filière d'agronomie de l'hepia Lullier, groupe sol et substrat. Ce laboratoire est reconnu par l'OFAG (Office fédéral de l'agriculture) et est agréé depuis 1993 pour effectuer ces analyses. Le laboratoire transmet les résultats à l'agriculteur. Ce dernier se base alors sur ces résultats pour procéder au calcul de ses bilans de fumure et appliquer les fertilisants rationnellement et dans le respect de la loi. Cela lui permet de percevoir les paiements directs et de satisfaire aux exigences PER (prestations écologiques requises).

Les analyses des échantillons portent sur les propriétés physico-chimiques du sol étudié. Les paramètres minimums analysés dans le cadre des PER sont : le pH, le phosphore et le potassium. Pour les grandes cultures, la teneur en matière organique est aussi requise. Grâce à la transcription des résultats de ces analyses dans une base de données, il est possible de cartographier les sols agricoles et de comprendre l'évolution des constituants des sols. Il faut remarquer que dans certaines conditions bien précises, les analyses de sol sont dispensées. Il n'y a donc pas de données concernant ces parcelles.

Il existe une carte publique basée sur ces travaux d'analyses qui est disponible sur le site SITG [2]. La dernière mise à jour des données date de 2005, les modifications effectuées par Alice Johannes en 2013 n'ont pas encore été mises en ligne. Ces données peuvent être consultées et peuvent servir notamment aux agriculteurs. En effet, ces derniers effectuent des rotations et des échanges de parcelles avec des voisins ou héritent de terrains. Ils peuvent alors retrouver les dernières analyses disponibles, les utiliser dans leur calcul de bilan de fumure et planifier de nouvelles analyses.

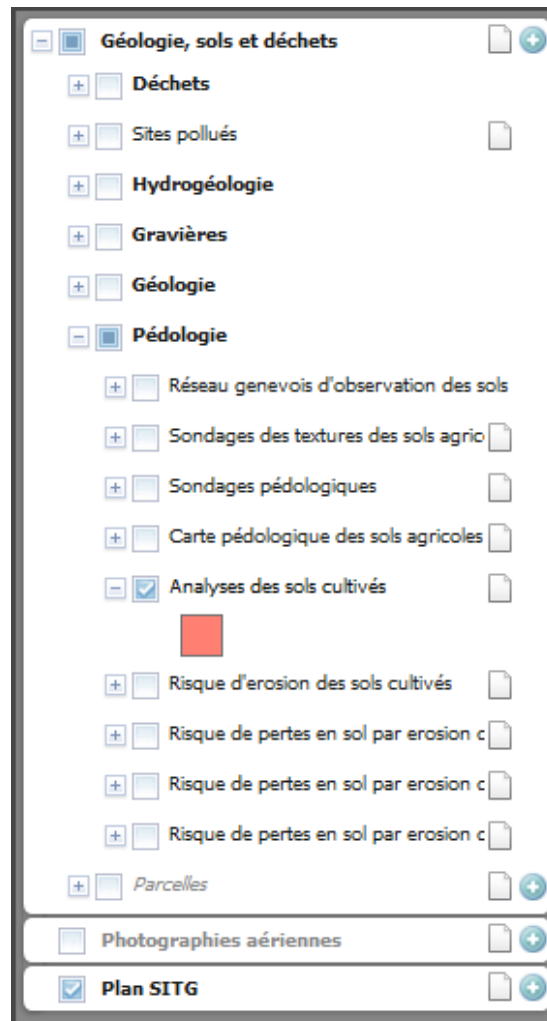


Illustration 1: Données à activer pour accéder à la couche des analyses des sols cultivés.

Ces données peuvent aussi intéresser divers services étatiques et des bureaux d'études en environnement. Il est donc très important de continuer à maintenir ces données à jour le plus régulièrement possible. C'est un des buts de ce stage qui sera décrit dans ce rapport.

La carte des analyses de sols est consultable sur le site SITG à l'adresse suivante : <http://ge.ch/geoportail/pro/>. Il faut rechercher et activer les données de « géologie, sols et déchets », puis dérouler le thème « pédologie » et activer la couche « analyses des sols cultivés » (voir illustrations 1 et 2).

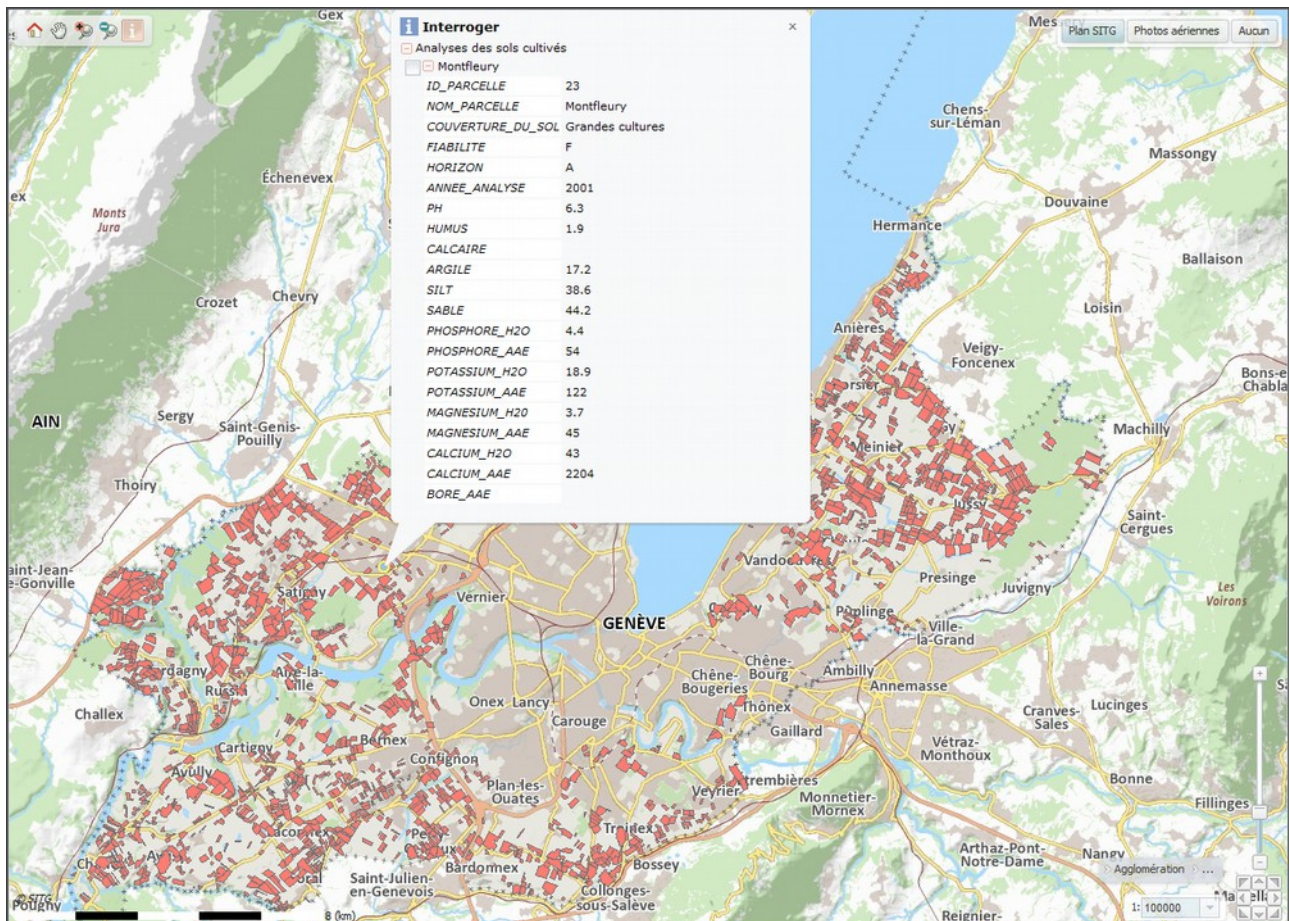


Illustration 2: Carte des analyses de sols cultivés disponible sur le site SITG, il est possible d'interroger les polygones avec l'outil adapté afin d'afficher les analyses, comme montré ci-dessus.

Les données affichées publiquement sont issues d'une table maintenue par le groupe sols et substrats de l'hepia Genève qui contient toutes les analyses effectuées et leur relation avec une surface agricole. Toutes les données ne sont pas rendues publiques pour des raisons de protection des données personnelles. Seules les données de localisation des surfaces agricoles et des analyses sont rendues publiques.

Au sein du groupe sols et substrats, ces données sont analysées et traitées de différentes manières. Par exemple, le suivi de l'évolution d'un paramètre du sol dans le temps est régulièrement effectué. La méthode de traitement des données pour obtenir cette évolution est actuellement entièrement effectuée à la main dans ArcMap. Un des objectifs du stage est de développer un outil codé en langage python qui permet d'automatiser le processus de traitement des données et d'en afficher le résultat. Cela doit être possible en permettant à l'utilisateur de choisir un certain nombre de variables.

Données et méthodologie

Description des données

Le travail a été réalisé avec ArcGIS 10.2.2, il a porté sur une couche de polygones qui représente les surfaces agricoles et sur une table reliée qui contient les résultats et les informations de toutes les analyses de sol. Ces données sont structurées comme suit :

Remarques : Les champs en vert sont les champs de jointure par numéro de surface agricole. Les champs en orange ne sont pas rendus publics sur le site du SITG soit car ces informations ne sont pas utiles au grand public, soit pour des raisons de confidentialité.

| Couche des surfaces agricoles | |
|-------------------------------|--|
| Nom | Surfaces_agricoles |
| Projection | CH1903+_LV95 |
| Champs | |
| OBJECTID | Champ automatique créé par ArcGIS qui attribue un numéro unique à chaque polygone |
| SHAPE | Champ automatique créé par ArcGIS qui définit le type d'objet dessiné sur la couche, ici : polygone |
| SHAPE_Length | Champs automatiques qui contiennent les informations géométriques des polygones. |
| SHAPE_Area | |
| ID_SURFACE_AGRICOLE | Même numéro que le champ « OBJECTID », permet de relier la couche à la table contenant les données d'analyses de sol |
| PREMIERE_DATE | Indique l'année de la première analyse géoréférencée, c'est le début de validité d'un polygone de surface agricole |
| DATE_FIN | Indique la date de fin de validité d'une parcelle agricole, soit parce qu'elle a été divisée, soit parce qu'elle a été fusionnée à une autre |

| Table des analyses de sol | |
|--|---|
| Nom | BDD_ANA_1993_2014 |
| Champs descriptifs et données complémentaires liées aux analyses | |
| OBJECTID_1 | Champ automatique créé par ArcGIS qui attribue un numéro unique à chaque analyse |
| ID_ANALYSE | Numéro identifiant de l'analyse |
| SURFACE_AGRICOLE | Numéro identifiant de la surface agricole qui permet de lier la table à la couche des polygones |
| FIABILITE | Fiabilité du géoréférencement |
| REMARQU_PLACE | Remarques sur l'emplacement, à renseigner lors de la saisie |
| COMMUNE | Numéro de commune |
| PARCELLE_CADASTRALE | Numéro de parcelle cadastrale donnée par l'agriculteur |
| NOM_PARCELLE | Nom de la parcelle agricole selon l'agriculteur |
| NOM | Nom de l'agriculteur |
| PRENOM | Prénom de l'agriculteur |
| COUVERTURE_DU_SOL | Type de culture (vignes, grandes cultures, maraîchage, ...) |
| CLASSE | Type d'exploitation, sous-abris ou en plein champ |
| ANNEE_ANALYSE | Année de l'analyse |
| ANNEE_ANALYSE_PRELEVEMENT | Champ de type « date », toujours la même valeur que le champ « ANNEE_ANALYSE » |
| PROFONDEUR | Profondeur du prélèvement de l'échantillon |
| Résultats d'analyse | |
| MATIERE_ORGANIQUE | Taux de matière organique [%] |
| PH | pH mesuré lors de l'analyse |
| CARBONATES_TOT | Taux de carbonates totaux [%] |
| SALINITE | Salinité mesurée |
| ARGILE | Taux d'argiles [%] |
| SILT_FIN | Taux de silts fin [%] |
| SILT_GROSSIER | Taux de silts grossiers [%] |
| SILTS_TOT | Taux de silts totaux [%] |
| SABLE_FIN | Taux de sables fin [%] |
| SABLE_GROSSIER | Taux de sables grossiers [%] |

| | |
|---------------------|--|
| SABLES_TOT | Taux de sables totaux [%] |
| P_PHOSPHORE_H2O | Phosphore extrait à l'eau [mg/kg] |
| P_PHOSPHORE_AAEDTA | Phosphore extrait à l'AAEDTA* [mg/kg] |
| K_POTASSIUM_H2O | Potassium extrait à l'eau [mg/kg] |
| K_POTASSIUM_AAEDTA | Potassium extrait à l'AAEDTA* [mg/kg] |
| MG_MAGNESIUM_H2O | Magnésium extrait à l'eau [mg/kg] |
| MG_MAGNESIUM_AAEDTA | Magnésium extrait à l'AAEDTA* [mg/kg] |
| CA_CALCIIUM_H2O | Calcium extrait à l'eau [mg/kg] |
| CA_CALCIIUM_AAEDTA | Calcium extrait à l'AAEDTA* [mg/kg] |
| B_BORE_AAEDTA | Bore extrait à l'AAEDTA* [mg/kg] |
| ech | Champs propres à une utilisation interne au laboratoire. |
| d | |
| s_d | |
| prat | |
| ann | |
| sem | |
| serie | |
| N | |

* AAEDTA : acétate d'ammonium + EDTA (acide éthylène diamine tétra acétique)

D'une façon générale, les données et leur structure correspondent tout à fait à l'utilisation que le groupe sols et substrats en a. A une exception, l'étude de l'évolution de paramètres dans le temps sera une activité qui prendra de plus en plus d'importance, au fur et à mesure que les données seront nombreuses. Elle est rendue difficile lorsque les surfaces agricoles sont changées. En effet, dans l'état actuel, un polygone de surface agricole qui n'est plus valide n'est pas relié avec le polygone qui l'a remplacé.

Pour pallier ce problème, il convient de constituer une table supplémentaire qui relie les surfaces agricoles périmées aux nouvelles qui ont pris leur place. Ainsi, un historique de l'évolution des surfaces agricoles est tenu. Cette nouvelle table a été mise en place par mes soins, voir illustration 3, 35 surfaces sont concernées.

| Lien_anciennes_Surfaces_Agricoles | | | |
|-----------------------------------|------------|---------------------------|---------------------------|
| | OBJECTID * | Ancienne_surface_agricole | Nouvelle_surface_agricole |
| | 33 | 534 | 2699 |
| | 34 | 2258 | 2682 |
| | 35 | 2259 | 2682 |

Illustration 3: Table de lien entre les polygones de surfaces agricoles périmés et les nouveaux les remplaçant. La table s'intitule « Lien_anciennes_Surfaces_Agricoles et les deux champs à renseigner sont de type « entier long », la valeur <null> n'est pas acceptée.

Mise à jour de la carte

La première étape de mise à jour des données a été d'ajouter les analyses des années 2013 et 2014 à la table des analyses. De façon à pouvoir traiter les données par copier-coller et d'harmoniser les termes le plus facilement possible, cela a été effectué dans excel. Il a donc fallu exporter la table des analyses en format CSV (voir illustration 4) puis l'importer dans excel. Ensuite, les nouvelles données des années 2013 et 2014 ont été ajoutées par copier-coller. La dernière étape a été de vérifier que les données sont bien dans le même format, utilisant les mêmes termes et ayant la même précision (le même nombre de chiffres après la virgule).

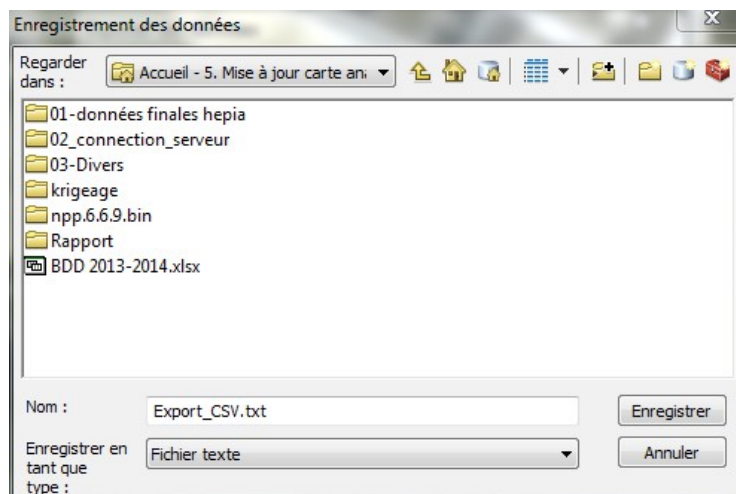


Illustration 4: Après un clic droit sur la table, il faut choisir le menu "Données" et exporter les données. Il est possible d'enregistrer la table en format texte. Les colonnes sont délimitées par des virgules, il s'agit bien d'un fichier au format CSV facilement importable dans excel.

L'avantage d'avoir utilisé excel est une manipulation plus aisée des données et le fait de laisser la table des analyses originale intacte. En cas de mauvaise manipulation qui aurait entraîné la corruption de données, il aurait été aisé de reprendre la table originale et de recommencer l'ajout des nouvelles données. Une fois les données complétées et homogénéisées, elles ont été importées en format CSV dans ArcGIS à l'aide de l'outil « Table to Table », voir illustration 5.

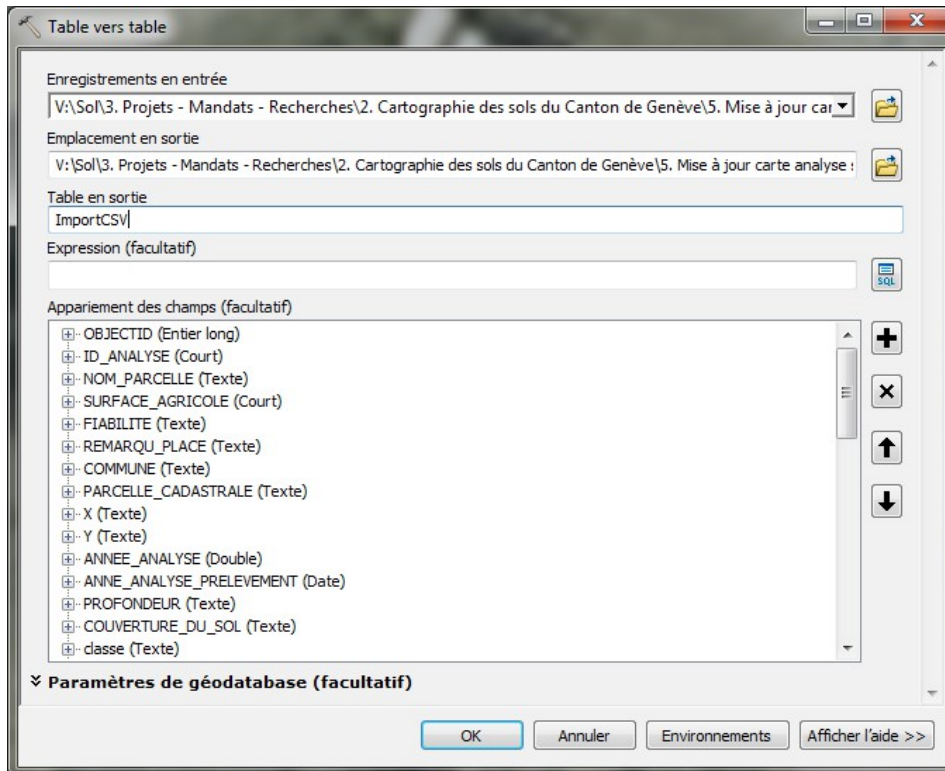


Illustration 5: Pour importer un fichier CSV dans ArcMap, il convient d'utiliser l'outil "Table to Table", de spécifier la géodatabase en sortie et le nom de la nouvelle table.

Une fois dans ArcMap, il s'agit de prendre en compte les éventuelles modifications des parcelles agricoles. Ainsi, les orthophotos contemporaines des données saisies ont été affichées. Cela implique un certain délai entre l'analyse et la saisie car les orthophotos ne sont pas tout de suite disponibles. Cependant, dans notre cas ce n'est pas gênant car ces données ne sont pas publiées, ni utilisées tout de suite après analyse. Il est par conséquent tout à fait acceptable d'attendre la disponibilité des orthophotos de la même année que l'échantillonnage.

En plus de l'orthophoto, il faut aussi afficher la numérotation la plus récente des parcelles cadastrales. Il ne faut pas supprimer les versions plus anciennes, car certains agriculteurs utilisent des numéros de parcelles qui n'existent plus sur la nouvelle numérotation.

La saisie des numéros de surfaces agricoles dans la table des analyses demande d'adapter l'espace de travail pour faciliter les opérations manuelles, ce qui diminue le risque d'erreur et accélère le processus de saisie. A noter que cette opération de mise à jour est plus facile sur un ordinateur doté de deux écrans. Les tables peuvent être affichées sur un écran entier, l'autre écran restant réservé pour ArcMap. Les points suivants énumèrent les changements à effectuer :

1. Personnalisation de l'interface d'affichage des tables :

- Ouvrir la table des analyses et la table des surfaces agricoles. Les deux tables sont ouvertes dans la même fenêtre et sont accessibles par les onglets (Voir illustration 6).

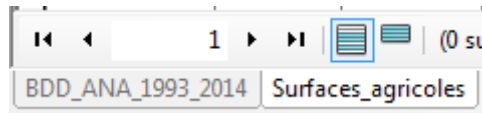


Illustration 6: Les deux tables ouvertes sont disponibles dans des onglets

- Créer un nouveau groupe d'onglets horizontal (Voir illustration 7).

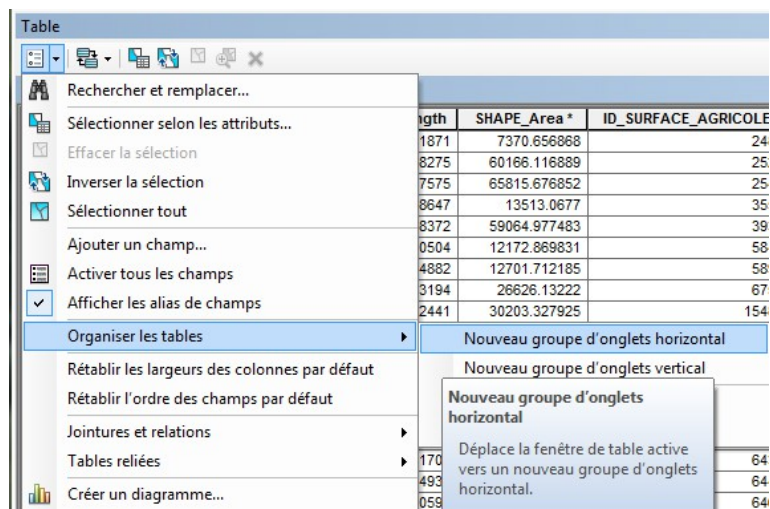


Illustration 7: Création d'un nouveau groupe d'onglets horizontal

- Réarranger la taille allouée à chaque table et choisir de n'afficher que les entités sélectionnées de la table des surfaces agricoles. La grande taille d'affichage de la table des analyses permet plus facilement de voir l'ensemble des parcelles d'un agriculteur. En générale, on ne travaille que sur une surface agricole à la fois, il n'y a donc pas besoin d'attribuer une grande place sur l'écran à cette table (voir illustration 8).

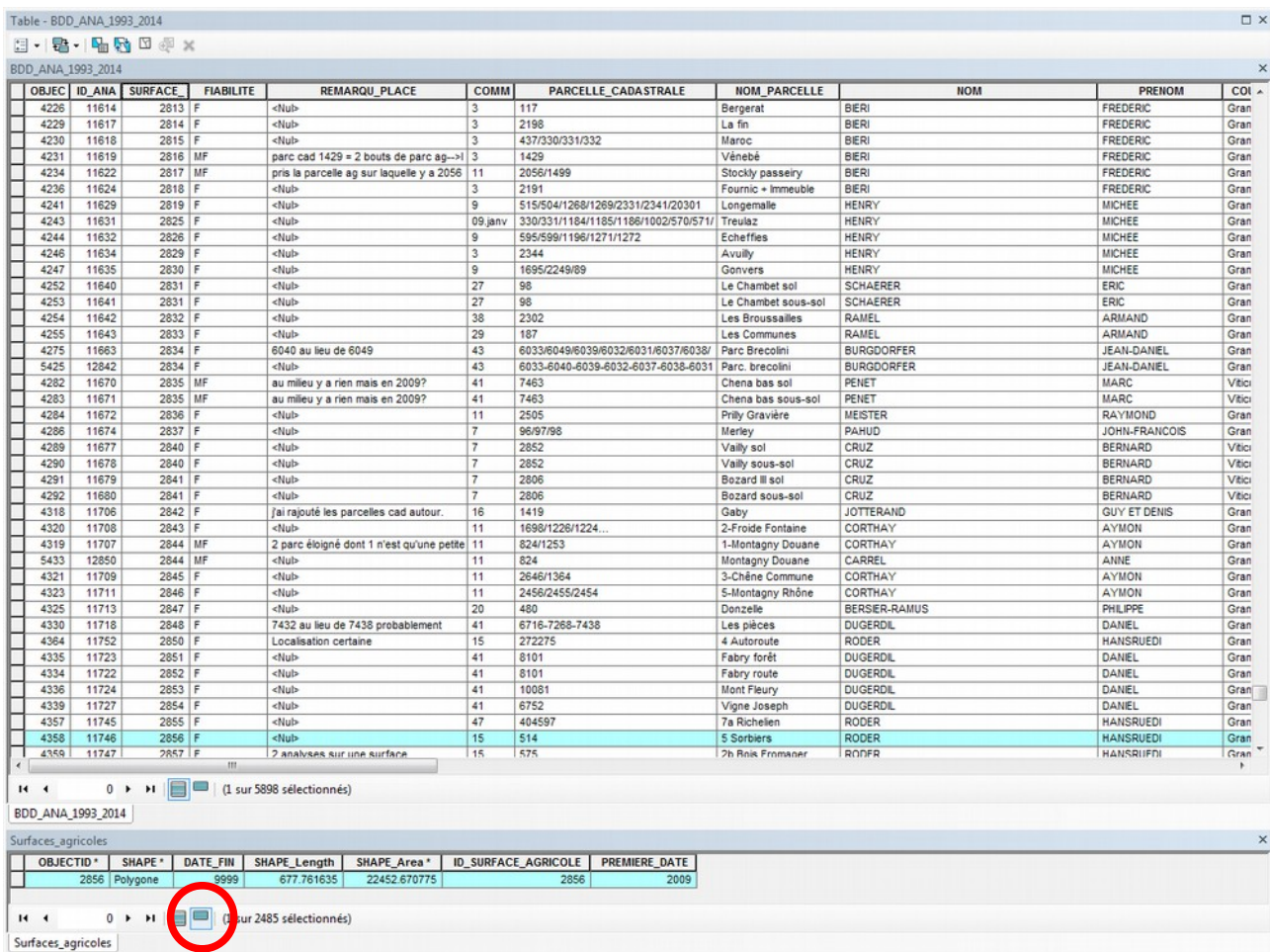


Illustration 8: Agencement de la visualisation des tables activation de la vue des entités sélectionnées pour la table des surfaces agricoles

2. Personnalisation de l'interface graphique d'ArcMap

L'agencement de l'interface graphique d'ArcMap permet aussi une meilleure visualisation des données et accélère les opérations manuelles par l'affichage et le rapprochement de certains boutons.

-Placer la barre d'outils généraux verticalement à gauche de la fenêtre d'ArcMap, faire apparaître les boutons « Sélection selon les attributs » et « Zoom sur les entités sélectionnées » grâce à la personnalisation d'interface (voir illustration 9 et annexe 1).

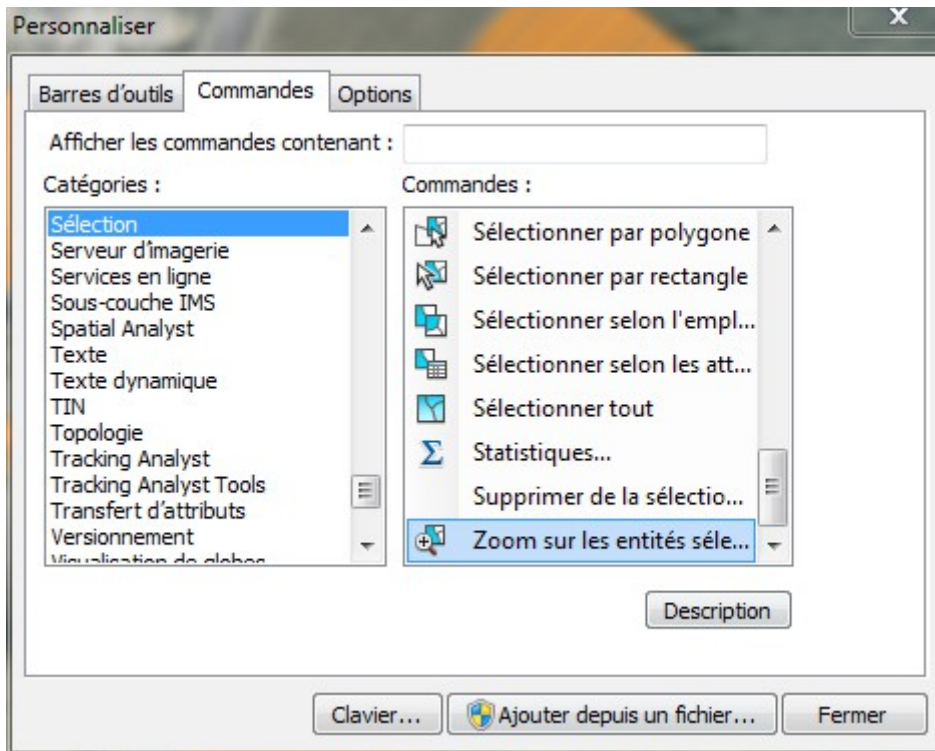


Illustration 9: Ajout des boutons "Sélection selon les attributs" et "Zoom sur les entités sélectionnées" par glisser déposer dans l'interface ArcMap depuis la fenêtre Personnaliser

La saisie des surfaces agricoles dans la table des analyses commence par la recherche de la surface agricole pour une analyse dont le numéro de commune et le numéro de parcelle cadastrale est fourni par l'agriculteur. Cela est effectué par une sélection par attribut sur la couche des parcelles cadastrales. L'expression SQL qui le permet est la suivante :

```
SELECT * FROM « Nom de la couche des parcelles cadastrales » WHERE : NO_COMM = X
AND NO_PARCELL = Y
```

Ici, X et Y correspondent respectivement au numéro de commune et au numéro de parcelle. Il peut arriver que les agriculteurs fournissent des numéros de parcelle et de commune qui ne soient plus actuels. Cela se traduit par une sélection de parcelle qui ne correspond pas à une surface cultivée ou une combinaison de numéro de commune et de parcelle qui n'existe pas. Pour tenter de remédier à ce problème, il convient d'effectuer une sélection sur une couche de parcelles cadastrales plus ancienne, si disponible. Il est aussi possible d'effectuer un tri croissant sur la colonne du nom de l'agriculteur. Ainsi, la table est réorganisée et toutes les parcelles du même agriculteur sont visualisables. Il faut alors rechercher des noms de parcelles similaires pour voir si l'agriculteur n'avait pas déjà fait une analyse sur la parcelle concernée. Si tel est le cas, il est possible de directement renseigner le numéro de surface agricole et inscrire une petite note dans le champ « REMARQU_PLACE ». Lorsque des numéros de surfaces agricoles sont attribués selon des démarches indirectes, il faut renseigner le champ « FIABILITE » avec « MF » qui signifie « moyennement fiable ». Lorsque le numéro de surface agricole est trouvé directement, ce champ sera renseigné d'un « F » qui signifie fiable.

La saisie des surfaces agricoles dans la table des analyses suit trois variantes possibles :

1. La surface agricole existe déjà et correspond à une culture cohérente sur l'ortho-photo.



Illustration 10: Ici le polygone de surface agricole existe déjà et semble représentatif de la surface réellement cultivée.

Dans ce cas, le numéro de la surface agricole est directement saisi dans la table des analyses.

2. La surface agricole existe déjà, mais semble ne plus correspondre à une surface cultivée particulière.

Dans ce cas, soit le polygone peut être légèrement modifié, voir illustration 11, et tout de même représenter la surface agricole, soit la date de fin du polygone est inscrite dans la table des surfaces agricoles et un nouveau polygone est créé. La table de correspondance entre ancienne et nouvelle surface agricole doit aussi être complétée en renseignant l'ancienne surface et la nouvelle.



Illustration 11: Ici, le polygone ne respecte manifestement pas fidèlement la surface réellement cultivée. Il convient alors d'en modifier le pourtour pour exclure les arbres.

3. La surface agricole n'existe pas.

Dans ce cas, un polygone qui représente la surface agricole doit être créé. Il peut correspondre tout à fait à la parcelle cadastrale et peut donc être copié-collé depuis cette couche (voir illustration 12).

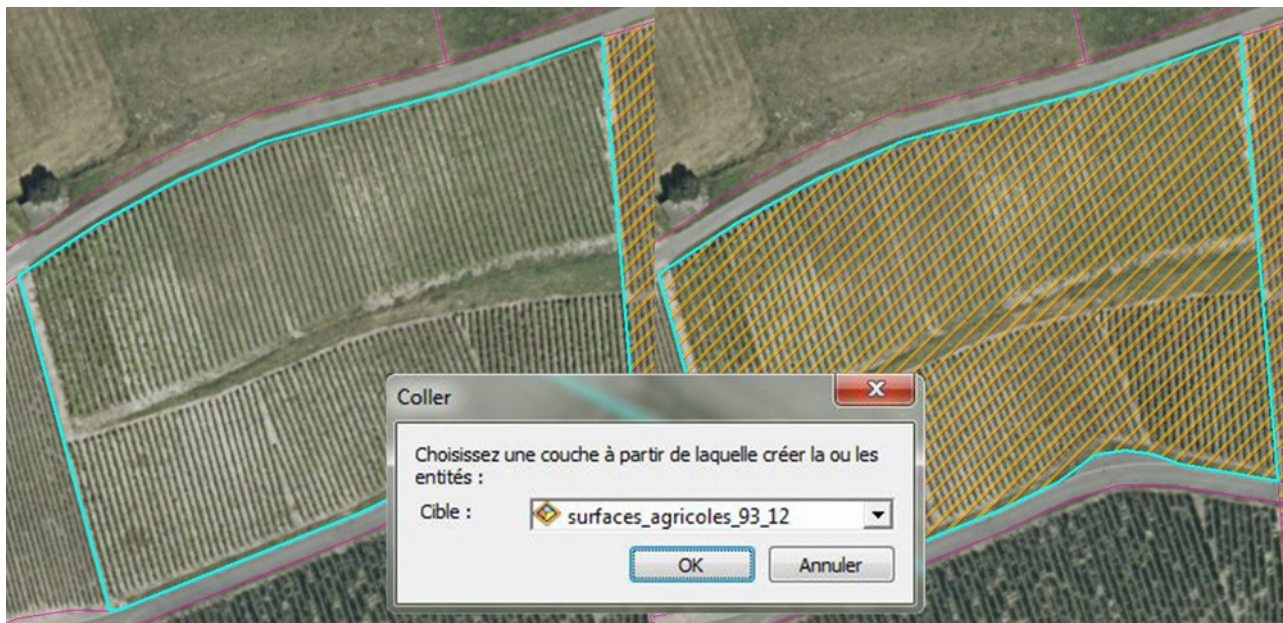


Illustration 12: Ici la surface réellement cultivée correspond tout à fait à la parcelle cadastrale. Il suffit donc de copier la surface cadastrale et de la coller dans la couche des surfaces agricoles. Le raccourci clavier « Ctrl+C » puis « Ctrl+V » est pratique et rapide.

Sinon, le polygone sera créé à la main en prenant soin de respecter le contour exacte des cultures (voir illustration 13).



Illustration 13: Ici la surface réellement cultivée ne correspond pas à la parcelle cadastrale, il s'agit donc de saisir le polygone de surface agricole à la main.

Dans tous les cas, la date de création (correspond à l'année analyse) du nouveau polygone est renseignée et le numéro de surface agricole est inscrit dans la table des analyses.

Développement d'un script python pour faire une différence de moyenne sur une période donnée

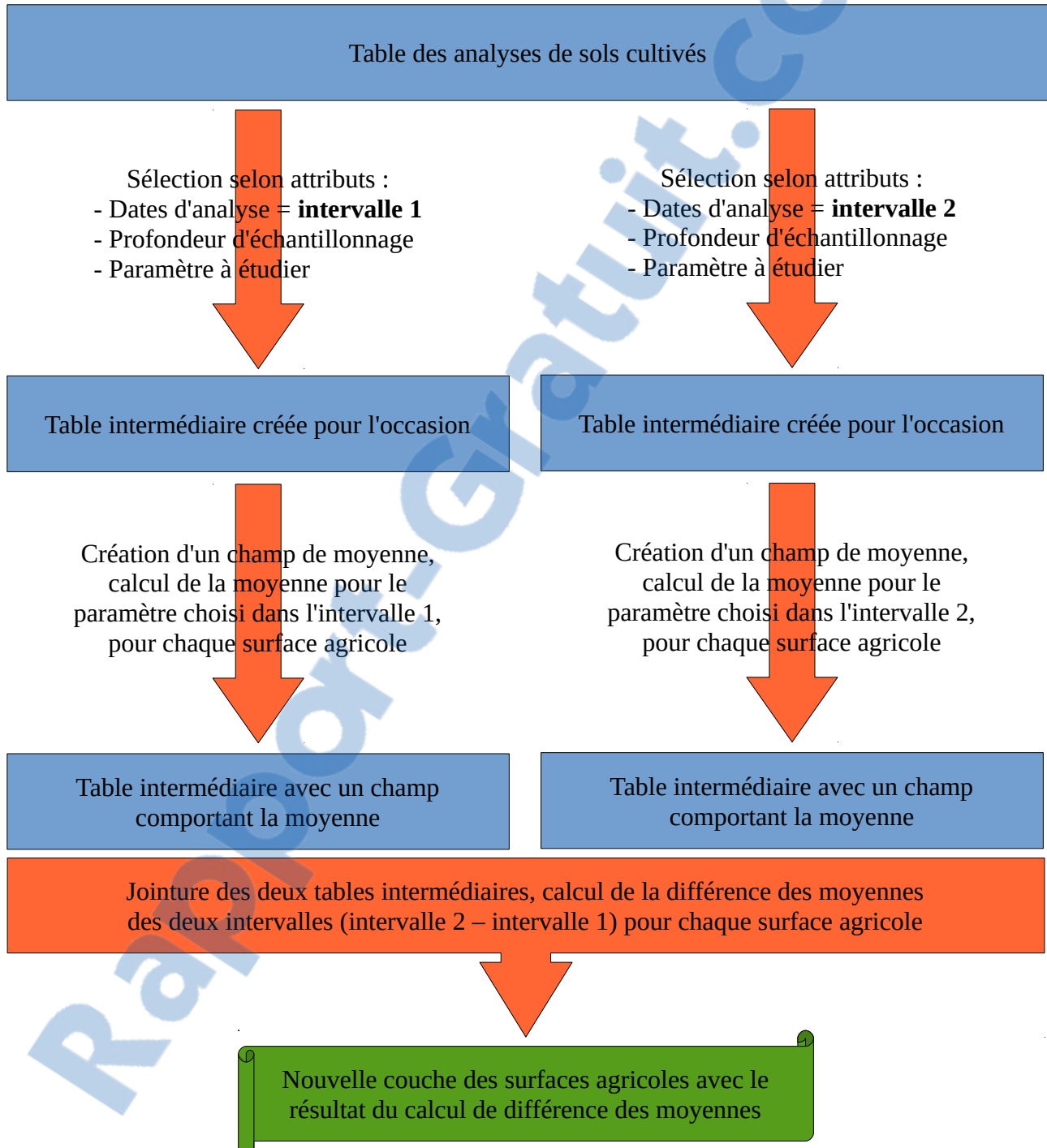
Fonctionnement général du script

La comparaison des données d'analyse pour mettre en lumière l'évolution de paramètres était jusqu'à présent effectué à la main [3]. C'est une procédure longue, fastidieuse et qui comporte un risque d'erreur non négligeable. Certaines manipulations ont directement lieu dans les tables originales, ce qui constitue un risque de corruption des données en cas d'erreur. Grâce à la programmation d'un outil spécifique, il est possible d'accélérer le processus, libérant du temps au chercheur pour se concentrer sur des tâches importantes à sa recherche. L'objectif de cet outil est de pouvoir comparer l'évolution d'un paramètre entre deux périodes de temps. L'évolution des paramètres du sol n'est pas la même en fonction de la profondeur d'échantillonnage. Ainsi, l'utilisateur doit pouvoir choisir si l'analyse concerne les échantillons prélevés dans le sol (2 - 20 cm) ou dans le sous-sol (20 - 40 cm). L'utilisateur doit pouvoir choisir deux intervalles de temps durant lesquels la moyenne des mesures d'un paramètre choisi seront calculées pour chaque surface agricole. Ensuite, la différences des moyennes pour chaque surface agricole dans les deux intervalles est calculée. Le résultat de ce calcul est mis en relation avec la couche des surfaces agricoles, dans le but de pouvoir être visualisé sur une carte.

Afin de préserver les données originales de tout risque de corruption, tout traitement de données doit être réalisé dans des tables séparées, créés pour l'occasion. De plus, l'affichage de la différence des moyennes sera réalisé dans une nouvelle couche et non directement dans la couche des surfaces agricoles. Cela permet de supprimer ou de conserver plus facilement des données calculées, sans risque de supprimer par inadvertance des données de la table d'origine. Cette dernière ne s'alourdit pas de nombreuses données à chaque fois que l'outil est utilisé.

Représentation schématique du fonctionnement de l'outil

Sur le diagramme ci-dessous, nous pouvons visualiser le fonctionnement général de l'outil.



Présentation d'un ModelBuilder simplifié

Une version simplifiée du processus de traitement des données en vue de l'étude de la comparaison d'un paramètre de qualité du sol dans le temps a été réalisée grâce au ModelBuilder, voir illustration 14. Cette version simplifiée était destinée à des tests divers afin, par exemple, de vérifier le bon enchaînement des outils et les paramètres généraux nécessaires. Ce ModelBuilder a été sauvegardé dans la boîte à outils « Cartographie-sol » qui contient aussi l'outil, comme décrit à la page 32. Le ModelBuilder peut servir de base pour le développement d'un tout autre outil de traitement des données d'analyses de sol ou permettre à une personne de modifier l'enchaînement et les outils pour répondre à une éventuelle autre question de recherche qui n'a pour l'instant pas été formulée. Il est en effet plus simple et plus rapide de modifier un ModelBuilder que de modifier un script python. Cela est vrai lorsque le ModelBuilder n'est pas souvent utilisé. Dans le cas contraire, le temps total de préparation du ModelBuilder pour le faire fonctionner de multiples fois devient vite plus long que de développer un script dédié. Le risque d'erreur par les multiples opérations manuelles est aussi diminué, par contre les possibilités sont limitées par les connaissances de la personne du langage de programmation python.

Il faut noter que ce ModelBuilder simplifié joint le résultat du traitement des données directement dans la couche des surfaces agricoles originale, ce qui peut constituer un risque de perte de données, par exemple lors d'opérations manuelles de suppression de champs accidentelles. Il convient donc de l'utiliser avec soin et de veiller particulièrement à bien supprimer les champs voulus lors de nettoyages de la table de la couche.

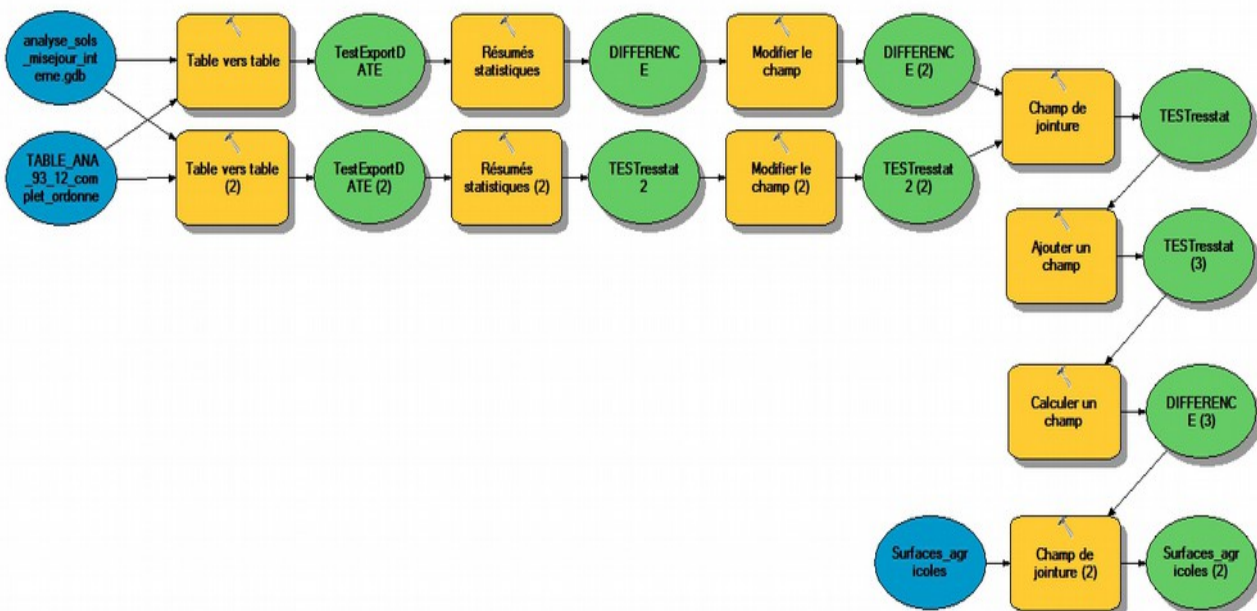


Illustration 14: Aperçu du ModelBuilder simplifié. En entrée, il faut renseigner la géodatabase, la table des analyses et la couche des surfaces agricoles. Le résultat du traitement est directement joint à la table de la couche des surfaces agricoles.

Comparaison du script par rapport au ModelBuilder

Il y a plusieurs avantages à utiliser un script plutôt qu'un ModelBuilder. Les deux principaux sont, une fois l'écriture du script terminée, la grande rapidité et la simplicité d'exécution. De plus, si le script a été correctement écrit, il va effectuer une série d'actions sans erreur, ce qui n'est pas toujours évitable lors d'un travail manuel.

Le script offre plus de souplesse et permet aisément d'effectuer certaines opérations qui sont tout simplement impossibles ou très compliquées avec le ModelBuilder. Par exemple, dans le script réalisé lors de ce travail, les noms de champ ainsi que noms de couches sont automatiquement nommés en fonction du paramètre de qualité de sol observé, de la profondeur d'échantillonnage et des intervalles de temps choisis par l'utilisateur. Cela ne serait pas facilement réalisable dans le ModelBuilder et nécessiterait des manipulations par l'utilisateur.

Un des désavantages du script est qu'il n'est réalisable que par une personne qui a déjà une certaine expérience de la programmation ainsi que du langage python. En effet, dans le contexte professionnel, un utilisateur inexpérimenté n'aura que peu, voir pas de temps prévu par son employeur pour développer ces connaissances.

Un autre désavantage est le temps qu'il faut consacrer à l'écriture du code, aux divers tests, au débogage, à la création de l'interface de saisie des paramètres et à la rédaction de l'aide. Il n'est donc pas rentable de réaliser un script pour des opérations effectuées ponctuellement.

Description des outils utilisés dans le script python

Pour arriver au résultat escompté, il est nécessaire de mettre en œuvre différents outils disponibles dans ArcGIS. Ici, les outils seront brièvement décrits en fonction de leur utilisation dans le script, pour plus d'informations sur leur fonctionnement détaillé, il convient de consulter l'aide en ligne d'ArcGIS [4]. Cette aide contient toutes les informations nécessaires au fonctionnement des outils ainsi que des exemples d'utilisation en langage de programmation python. Tous les outils disponibles dans la « ArcToolbox » sont aussi disponibles en tant que fonctions python, c'est la raison pour laquelle les scripts sont développés dans ce langage de programmation.

| Outils/fonctions utilisés dans le script | |
|---|--|
| Fonctions générales non spécifiques au traitement des données | |
| arcpy.env.workspace.overwriteOutput | Permet ou empêche d'écraser des données existantes. La variable de type booléenne prend « True » ou « False » comme paramètre. Dans le script, elle est définie sur « True » pour permettre l'écrasement de façon à pouvoir faire tourner le script plusieurs fois de suite. |
| arcpy.env.workspace | Permet de définir l'environnement de travail (géodatabase), tous les output des outils prendront place dans l'environnement défini. |

| | |
|--------------------------------|--|
| arcpy.GetParameterAsText(X) | Permet d'introduire des paramètres saisis par l'utilisateur. Le X est une valeur numérique d'index qui permet de relier la fonction à un champ dans l'interface graphique. A noter qu'en python, la numérotation commence à 0. |
| arcpy.AlterField_management() | Permet de renommer le nom des champs afin qu'il soit représentatif du paramètre de sol choisi, des dates d'intervalles et de la profondeur d'échantillonnage. Cela permet à l'utilisateur qui fera tourner plusieurs fois le script avec des paramètres différents de reconnaître immédiatement de quelles données il s'agit. Cette fonction est utilisée deux fois. |
| arcpy.JoinField_management() | Permet de joindre des champs à une autre table. Cette fonction est utilisée deux fois, une fois pour le calcul de différence de moyenne entre les deux intervalles de temps et une fois pour joindre le résultat de la différence à la nouvelle couche affichée en fin de script. |
| arcpy.AddField_management() | Permet d'ajouter un champ dans une table. Cette fonction est utilisée pour ajouter le champ de calcul de différence des moyennes. |
| arcpy.mapping.MapDocument() | Permet de définir l'espace de travail courant avec « CURRENT » comme paramètre. |
| arcpy.mapping.ListDataFrames() | Permet de lister les « Data frames » dans un .mxd. |
| arcpy.mapping.Layer() | Permet de sélectionner une couche. |
| arcpy.mapping.AddLayer() | Permet d'ajouter une couche dans un « Data frame ». |
| arcpy.RefreshActiveView() | Permet de rafraîchir l'affichage pour afficher de nouvelles données. |
| arcpy.Delete_management() | Permet de supprimer tout type de données pris en charge par ArcGIS (tables, géodatabases, couches, boîtes à outils, dossiers, ...). Dans le script, cette fonction est utilisée pour supprimer les tables intermédiaires créées lors du déroulement du traitement. Cela permet de nettoyer la géodatabase des données inutiles. |

| | |
|---|--|
| arcpy.AddMessage() | Permet de faire apparaître un message dans la console lors de l'exécution du script. Il est intéressant de jalonner le script de messages car l'utilisateur impatient comprend alors ce qui se passe et que le script fonctionne, l'ordinateur n'a pas « planté ». De plus, en cas d'erreur, cela rend la recherche de la partie du code fautive plus aisée. |
| str() | Permet de changer le type en format « string », soit un format texte. Cette fonction est utilisée pour transformer des nombres en texte. |
| Fonctions utilisées pour le traitement des données | |
| arcpy.TableToTable_conversion() | Permet d'extraire des données de la table des analyses en les sélectionnant par date et par profondeur d'échantillonnage. Créé une table avec le résultat de la sélection. Cette fonction est utilisée deux fois, une fois pour chaque intervalle de temps. La profondeur d'échantillonnage est la même pour les deux intervalles afin que la comparaison soit correcte. |
| arcpy.Statistics_analysis() | Permet de calcul des moyennes des valeurs du paramètre de qualité de sol choisi pour chaque surface agricole, dans les deux intervalles de temps. Cette fonction est utilisée deux fois, une fois pour chaque intervalle de temps. Créé une table avec le résultat du traitement. |
| arcpy.CalculateField_management() | Permet de calculer la valeur d'un champ. Cette fonction a été utilisé pour calculer la différence des moyennes des mesures du paramètre choisi dans les intervalles de temps, pour chaque surface agricole. L'expression pour effectuer ce calcul a été préalablement mise en forme et placée dans une variable (Expression) utilisée en paramètre. Le format de l'expression est en python, le VBScript n'étant pas pris en charge par la fonction (retourne une erreur et arrête le script). |

| | |
|---|---|
| arcpy.FeatureClassToFeatureClass_conversion() | Permet de convertir ou copier des classes d'entités. Cette fonction a été utilisé pour dupliquer la couche des surfaces agricoles en vue de laisser la couche originale vierge de tout changement et d'afficher le résultat du calcul des différences de moyennes pour chaque surface agricole dans une nouvelle couche. Cette dernière est correctement nommée en fonction du paramètre de qualité du sol choisi, des dates d'intervalles et de la profondeur d'échantillonnage. |
|---|---|

Pour plus d'informations sur l'utilisation de ces fonctions, voir la présentation du code, au point suivant.

Présentation du code

Dans cette section, nous verrons plus en détail le code python qui permet de réaliser une nouvelle couche qui montre l'évolution d'un paramètre choisi entre deux intervalles de temps. Bien que le script soit commenté comme il se doit, dans cette section, les commentaires et les messages ont été enlevé pour faciliter la lecture et se concentrer sur l'essentiel du code. Le code complet est visualisable en annexe 2.

La première étape est de définir les options de base pour le script. Premièrement, il s'agit de déclarer l'encodage des caractères :

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

Il faut ensuite importer la librairie arcpy, autoriser l'écrasement de fichiers :

```
import arcpy  
arcpy.env.overwriteOutput = True
```

Les premiers paramètres saisis par l'utilisateur sont introduits, à savoir l'environnement (la géodatabase) et la table en entrée (la table des analyses de sol) :

```
arcpy.env.workspace = arcpy.GetParameterAsText(0)  
InTable = arcpy.GetParameterAsText(1)
```

La table des analyses définie, nous devons y effectuer une sélection grâce à l'outil « TableToTable ». Un certain nombre de paramètres doivent donc être déclarés. Il faut déclarer les dates du premier intervalle de temps et la profondeur d'échantillonnage souhaitée. On remarque le numéro d'index de la profondeur qui n'est pas dans la continuité de numérotation, cela est explicable par le fait que la sélection de la profondeur a été implémentée dans un deuxième temps alors que le script et d'autres paramètres étaient déjà définis. Cela n'influe en rien l'exécution du code.

```
date_1 = arcpy.GetParameterAsText(2)
date_2 = arcpy.GetParameterAsText(3)
prof = arcpy.GetParameterAsText(7)
```

Il faut aussi rédiger l'expression qui permettra d'effectuer la sélection dans la table. Cette expression est fournie comme paramètre pour l'outil « TableToTable », il est obligatoire de la fournir sous forme de « string ». Ici, il s'agit de sélectionner les entrées dans la table des analyses dont l'année d'analyse (= année d'échantillonnage) est supérieure ou égale à la date de début et inférieure ou égale à la date de fin du premier intervalle saisi par l'utilisateur. Les nombres des dates sont changés en format « string ». De plus, la profondeur saisie par l'utilisateur est aussi prise en compte dans la sélection.

```
Expression = "ANNEE_ANALYSE >= "+str(date_1) +" AND ANNEE_ANALYSE  
<= "+str(date_2)+" AND PROFONDEUR = '"+str(prof)+"'"
```

Il faut spécifier l'emplacement de sortie des tables créées lors du déroulement de script. Ici, les tables seront placées dans la géodatabase saisie par l'utilisateur.

```
Out = arcpy.env.workspace
```

Tous les paramètres sont définis, il est possible d'effectuer la sélection définie dans l'expression et de l'exporter vers une nouvelle table nommée « TtoT_1 ».

```
arcpy.TableToTable_conversion(InTable, Out,"TtoT_1", Expression)
```

Certaines variables seront réutilisées plus tard, la terminologie est gardée pour éviter les confusions. Ainsi, elles doivent être libérées pour ne pas créer de conflits.

```
del Out
del Expression
```

La sélection et l'export effectués, nous pouvons procéder au calcul de la moyenne des analyses du paramètre de qualité du sol choisi par l'utilisateur dans le premier intervalle. Cela sera fait grâce à la fonction « Statistics_analysis » dont il est nécessaire de déclarer les paramètres. En entrée, nous choisissons la table créée précédemment qui contient la sélection pour le premier intervalle de temps. En sortie, nous choisissons un nom de table qui montre que le traitement par moyenne (résumé statistique) a eu lieu. Le champ saisi par l'utilisateur correspond au paramètre de qualité du sol étudié. Ce champ doit être mis en format « string » et doit se terminer par « MEAN » pour spécifier le traitement par moyenne en paramètre par la fonction. C'est ce qui est réalisé et stocké dans la variable « champ_1 ».

```
intable = "TtoT_1"  
Out = "TtoT_1_Res"  
champ = arcpy.GetParameterAsText(6)  
champ_1 = champ+" MEAN"
```

Tous les paramètres sont définis, il est possible de réaliser le traitement des données. Ici, le dernier paramètre spécifie que le traitement aura bien lieu pour chaque surface agricole.

```
arcpy.Statistics_analysis(intable, Out, champ_1,  
"SURFACE_AGRICOLE")
```

A nouveau certaines variables seront réutilisées plus tard, la terminologie est gardée pour éviter les confusions. Ainsi, elles doivent être libérées pour ne pas créer de conflits.

```
del Out  
del intable
```

La table TtoT_1_Res contient donc le résultat du traitement des données pour le premier intervalle de temps. Un champ nommé automatiquement par l'outil « Statistics_analysis » a été créé. Dans un souci de clarté, il convient de garder les intitulés des champs évocateurs, il faut donc renommer ce champ. Pour le faire, nous allons faire appel à la fonction « AlterField_management » dont il faut déclarer les paramètres. La table en entrée est la table créée précédemment, il faut aussi mettre en forme le champ avec le nom du paramètre pour que le bon champ soit pris en paramètre (« NomChamp_1 »). Il convient aussi de choisir le nouveau nom de champ qui sera stocké dans la variable « Renomme_1 ».

```
intable_1 = "TtoT_1_Res"  
NomChamp_1 = "MEAN_"+champ  
Renomme_1 = champ+"_1"
```


L'action de renommage de champ est réalisée comme suit.

```
arcpy.AlterField_management(intable_1, NomChamp_1, Renomme_1,
Renomme_1)
```

Le traitement des données pour le premier intervalle est désormais terminé. Il faut à présent répéter les mêmes opérations pour le deuxième intervalle de temps. Le code utilisé est exactement le même, sauf naturellement les dates choisies par l'utilisateur, le nom des tables créées et le nom des champs créés. Il n'est donc pas nécessaire de commenter un code identique à quelques exceptions près. Il est collé ci-dessous, seules les grandes étapes sont relevées et commentées.

Sélection et export du deuxième intervalle de temps en conservant la même profondeur que pour le premier intervalle.

```
date_3 = arcpy.GetParameterAsText(4)
date_4 = arcpy.GetParameterAsText(5)
Expression = "ANNEE_ANALYSE >= "+str(date_3) +" AND ANNEE_ANALYSE
<= "+str(date_4)+" AND PROFONDEUR = '"+str(prof)+"'"
Out = arcpy.env.workspace
arcpy.TableToTable_conversion(InTable,Out , "TtoT_2", Expression)
del Out
del Expression
```

Traitement par moyenne du deuxième intervalle de temps.

```
intable = "TtoT_2"
Out = "TtoT_2_Res"
arcpy.Statistics_analysis(intable, Out, champ_1,
"SURFACE_AGRICOLE")
del Out
del intable
```

Modification du nom du champ créé dans la nouvelle table nommée « TtoT_2_Res ».

```
intable_2 = "TtoT_2_Res"
Renomme_2 = champ+"_2"
arcpy.AlterField_management(intable_2, NomChamp_1, Renomme_2,
Renomme_2)
```

Nous disposons donc de deux tables (« TtoT_1_Res » et « TtoT_2_Res ») qui contiennent les données traitées de chaque surface agricole pour les deux intervalles de temps et la même profondeur. Afin de pouvoir faire la différence entre les deux intervalles, nous devons joindre les deux tables par le champ des surfaces agricoles. Cela est réalisé en faisant appel à la fonction « JoinField_management ». Le champ de moyenne de la table « TtoT_2_Res » sera ajouté à la table « TtoT_1_Res ». Tous les paramètres nécessaires sont déjà déclarés, nous pouvons directement procéder à la jointure.

```
arcpy.JoinField_management(intable_1, "SURFACE_AGRICOLE",
intable_2, "SURFACE_AGRICOLE", Renomme_2)
```

Afin de calculer la différence des moyennes, il faut ajouter un champ à la table « TtoT_1_Res ». Ce champ doit être nommé correctement pour représenter ce qu'il contient, il commence donc par « Difference_ » et contient le paramètre de qualité du sol étudié ainsi que la date de début du premier intervalle et la date de fin du deuxième intervalle. La mise en forme de ce nom est stocké dans une variable nommée « Champdecalcul ».

```
Champdecalcul = "Difference_"+champ+"_"+date_1+"_a_"+date_4
```

La fonction « AddField_management » est exécutée pour réaliser l'ajout du champ de calcul. Ce champ doit être défini par différentes caractéristiques, ici le plus important est que le type de données qu'il contiendra est « FLOAT », soit un nombre réel. Il est aussi défini que le champ peut être nul et ne contenir aucune donnée, soit « NULLABLE » et « NON_REQUIRED ».

```
arcpy.AddField_management("TtoT_1_Res", Champdecalcul, "FLOAT",
"", "", "", "", "NULLABLE", "NON_REQUIRED", "")
```

Nous avons maintenant un champ qui va accueillir le résultat de la soustraction des moyennes des intervalles 1 et 2. Ce calcul sera effectué grâce à la fonction « CalculateField_management ». Il est donc nécessaire de définir une expression qui permet le calcul, elle sera stockée dans la variable « Expression » et utilisée en paramètre. Il faut relever que l'expression est en langage python car le VBScript, bien qu'il soit spécifié comme supporté sur le site d'aide d'ArcGIS, entraîne une erreur et arrête le script. A nouveau, il est important de construire l'expression en format « string ». A noter que l'intervalle 1 est soustrait à l'intervalle 2 pour mettre en lumière l'évolution du paramètre de qualité du sol étudié.

```
Expression = "!" + Renomme_2 + "! - !" + Renomme_1 + "!"
```

Une fois la variable Expression définie, il faut faire appel à la fonction « CalculateField_management » pour réaliser le calcul. A noter que comme l'expression est en python, il faut le spécifier en paramètre : « PYTHON ».

```
arcpy.CalculateField_management("TtoT_1_Res", Champdecalcul,
Expression, "PYTHON", "")
```

Le calcul a été effectué, il convient maintenant de joindre le résultat à une copie de la couche des surfaces agricoles. Premièrement, il faut adapter l'intitulé de la profondeur qui sera dans le nom de notre nouvelle couche. En effet, celui-ci contient un tiret et des espaces, cela est rejeté par l'outil « FeatureClassToFeatureClass » utilisé pour cloner la couche des surfaces agricoles. Comme la profondeur peut prendre deux valeurs différentes, il faut effectuer un test pour appliquer la mise en forme correcte pour les deux cas possibles. Si la variable « prof » qui contient la profondeur correspond exactement à « 2 – 20 cm », il faut la transformer en « 2_20_cm ». Si elle correspond exactement à « 20 – 40 cm », il faut la changer en « 20_40_cm ». A noter que l'indentation doit impérativement être respectée. Bien que cela ne soit pas possible étant donné que la profondeur est saisie par l'utilisateur dans une liste pré-établie, si la profondeur a une autre valeur que les deux attendues, le script tente de continuer si cela ne génère pas d'erreur. Cela peut être utile si quelqu'un modifie le script en amont en modifiant la valeur de la profondeur.

```
if prof == '2 - 20 cm':
    prof = "2_20_cm"
elif prof == '20 - 40 cm':
    prof = "20_40_cm"
else :
    pass
```

A présent la profondeur d'échantillonnage est en forme pour pouvoir être intégrée dans le nom de la couche. Il peut donc être défini et placé dans une variable « NomCouche ». Le nom de la couche contient donc le sujet qui est les surfaces agricoles, le paramètre de qualité du sol étudié qui est stocké dans la variable « champ », les années de début du premier intervalle et de fin du deuxième intervalle et se termine par la profondeur d'échantillonnage

```
NomCouche =
"Surfaces_Agricoles_"+champ+"_"+date_1+"_a_"+date_4+"_"+prof
```

Nous disposons de tous les paramètres nécessaires au fonctionnement de la fonction « FeatureClassToFeatureClass ». La couche « Surfaces_agricoles » sera copiée dans la géodatabase définie en début de script et prendra le nom défini dans la variable « NomCouche » définie ci-dessus.

```
arcpy.FeatureClassToFeatureClass_conversion("Surfaces_agricoles",  
arcpy.env.workspace, NomCouche)
```

La nouvelle couche créée, nous pouvons y joindre le résultat du calcul des différences de moyennes des mesures du paramètre de qualité du sol étudié entre deux intervalles. A nouveau, la fonction « JoinField_management » sera utilisée. La couche cible est celle créée précédemment et dont le nom est stocké dans la variable « NomCouche », le champ de correspondance s'intitule « ID_SURFACE_AGRICOLE ». La table qui contient le champ de calcul est « TtoT_1_Res », le champ de correspondance est « SURFACE_AGRICOLE » et le champ à joindre est « Champdecacul ».

```
arcpy.JoinField_management(NomCouche, "ID_SURFACE_AGRICOLE",  
"TtoT_1_Res", "SURFACE_AGRICOLE", Champdecacul)
```

Le traitement des données est complété et la nouvelle couche a été créée, le script pourrait s'arrêter ici. Dans un souci de bonne tenue de la géodatabase, les tables intermédiaires créées lors du processus qui n'ont plus aucune utilité seront supprimées. Pour le réaliser, la fonction « Delete_management » est utilisée. Cette fonction est très simple, il suffit de renseigner en paramètre le nom de ce que l'on souhaite effacer. Dans notre cas, quatre tables sont concernées : TtoT_1, TtoT_2, TtoT_1_Res, TtoT_2_Res.

```
arcpy.Delete_management("TtoT_1")  
arcpy.Delete_management("TtoT_2")  
arcpy.Delete_management("TtoT_1_Res")  
arcpy.Delete_management("TtoT_2_Res")
```

Dans un souci d'élégance, la couche créée est affichée à l'espace de travail, ce qui évite à l'utilisateur de devoir ouvrir la fenêtre d'ArcCatalog, de rechercher la couche et de l'ajouter dans ArcMap. Pour ce faire, il convient de stocker dans une variable « mxd », l'espace de travail courant grâce à la fonction « mapping.MapDocument ».

```
mxd = arcpy.mapping.MapDocument("CURRENT")
```

Il faut ensuite lister les « Data Frames » disponibles dans l'espace de travail et de stocker cette liste dans la variable « df ».

```
df = arcpy.mapping.ListDataFrames(mxd, "*")[0]
```

Le nom de la couche à ajouter est stocké dans une variable « couche » grâce à la fonction « mapping.Layer » qui prend en paramètre le nom de la couche, ici la variable « NomCouche ».

```
couche = arcpy.mapping.Layer(NomCouche)
```

Nous pouvons ajouter la couche au « Data Frame » courant en faisant appel à la fonction « mapping.AddLayer ». La nouvelle couche sera placée dans le « Data Frame » courant, au dessus des éventuelles autres couches grâce au paramètre « TOP ».

```
arcpy.mapping.AddLayer(df, couche, "TOP")
```

Pour terminer, il faut rafraichir la vue afin d'afficher les données de la couche fraîchement ajoutée. Cela est réalisé grâce à la fonction « RefreshActiveView », sans paramètres.

```
arcpy.RefreshActiveView()
```

Le script est à présent terminé, les données ont été traitées, le résultat a été affiché et les éléments créés lors du processus ont été nettoyés. Le script peut être utilisé dans la console python de ArcMap, mais il est préférable de proposer à l'utilisateur un outil comportant une interface graphique qui lui permet de saisir facilement les paramètres nécessaires au fonctionnement du script. Une aide qui renseigne sur chaque paramètre à saisir peut aussi être affichée. La création de l'interface graphique et la rédaction de l'aide sont décrits dans le point suivant.

Création de l'outil et rédaction de l'aide

Dans ArcMap, il existe un assistant de création d'outil. Cet assistant permet de créer simplement une interface graphique à un script en attribuant des champs de saisie pour les paramètres. Le type de données que chaque champ va accueillir doit être en adéquation avec le type de paramètre attendu par le script. Ainsi, pour chaque paramètre, il faut choisir un type de données parmi une liste déroulante. Il est aussi possible de définir si le champ est obligatoire ou facultatif ou s'il comporte une valeur par défaut. Le processus d'importation de script dans une boîte à outil spécialement créée pour l'occasion est visualisable sur les illustrations 15 à 17.

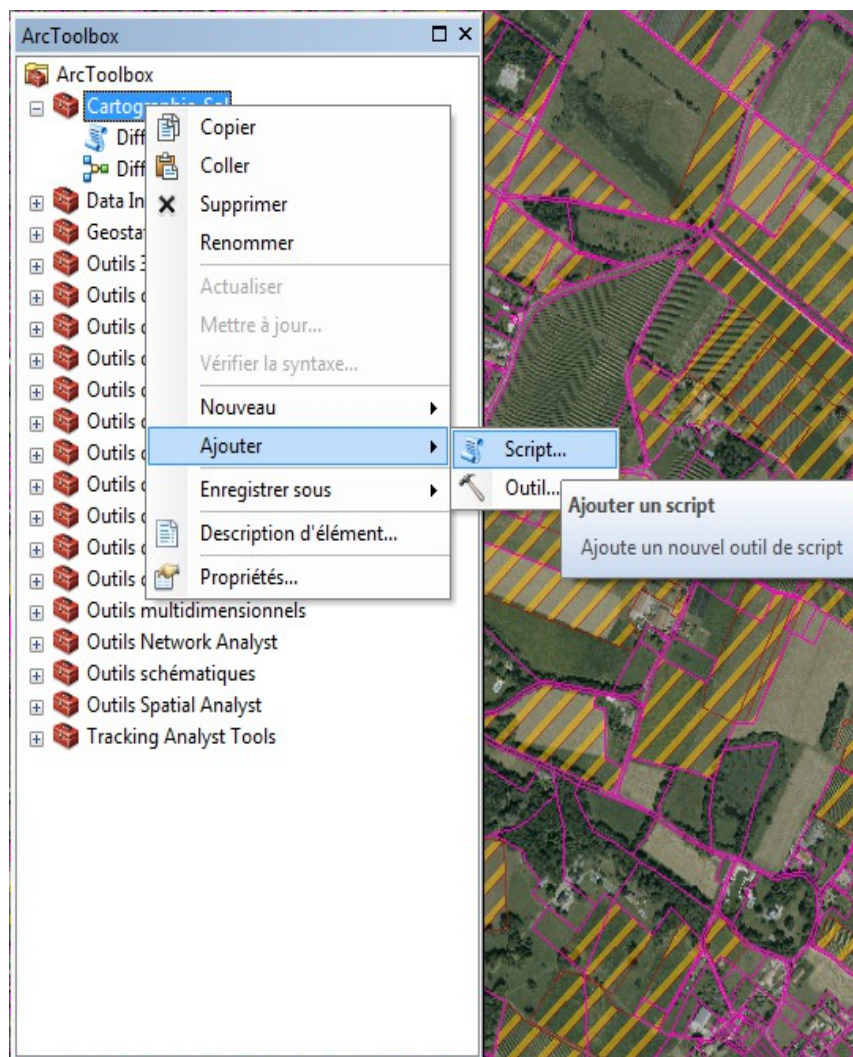


Illustration 15: Pour importer un nouveau script dans une boîte à outils, ici « Cartographie-sol », créée pour l'occasion, il faut faire un clic droit sur la boîte et sélectionner "ajouter" puis "script".

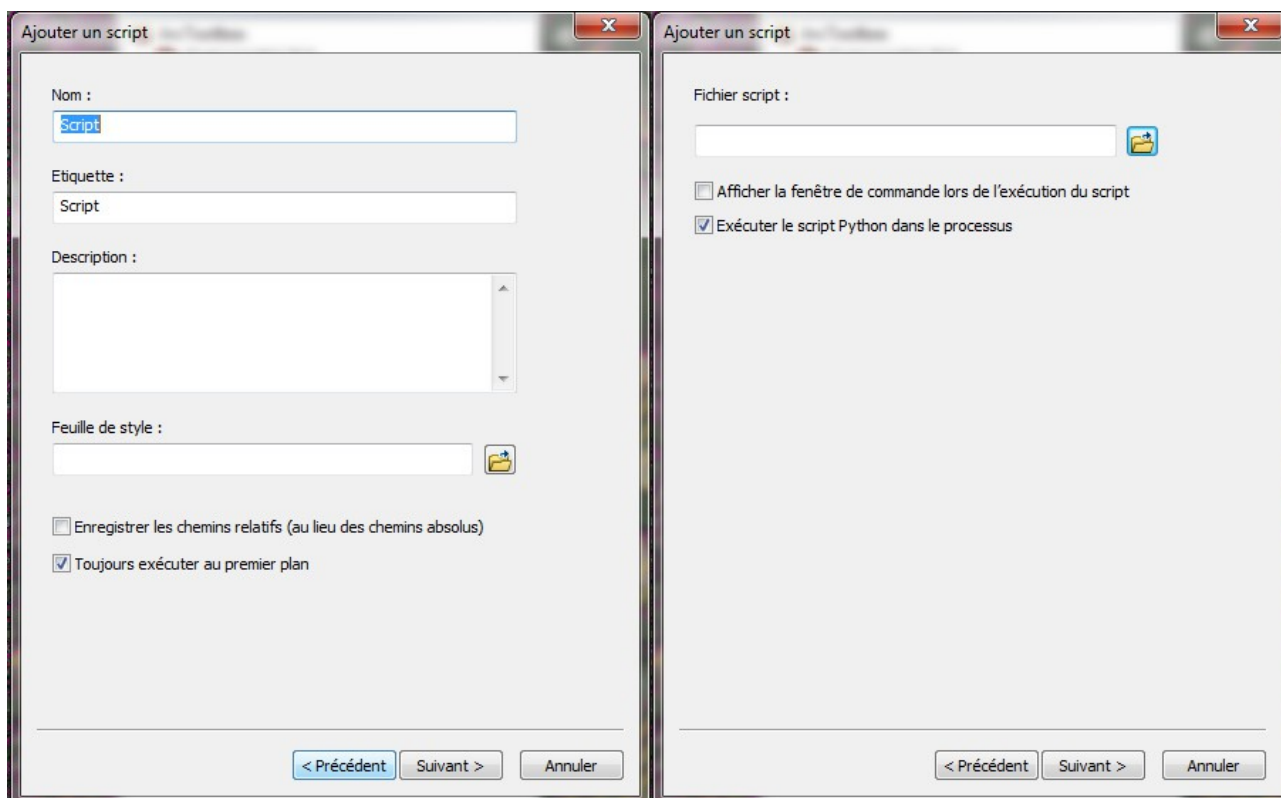


Illustration 16: Une fenêtre d'importation s'ouvre alors. Dans le premier panneau (gauche), il faut définir le nom et l'étiquette du script. Une feuille de style peut être ajoutée au format XML pour modifier l'apparence de l'outil à sa convenance. Dans le deuxième panneau (droite), il faut renseigner le fichier contenant le code. Cela est coché par défaut, mais il est nécessaire de vérifier que le script python s'exécute bien dans le processus.

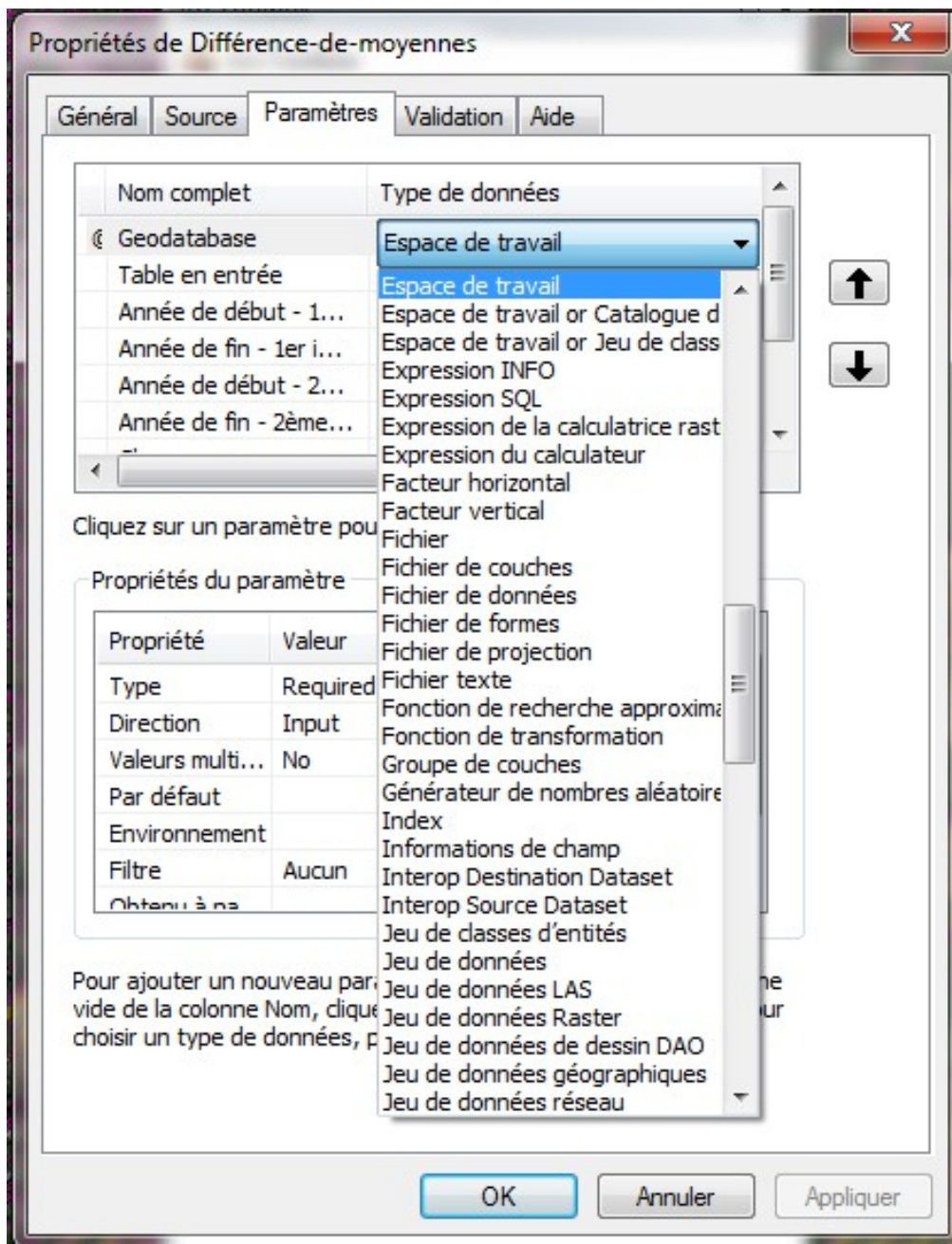


Illustration 17: Sur le troisième panneau, il faut renseigner les paramètres du script et le type de données qui sont attendus. Ici, le premier paramètre du script "arcpy.GetParameterAsText(0)" est relié à une variable d'environnement "arcpy.env.workspace" dans le code, le type de données attendu est donc un espace de travail. Les paramètres doivent être saisis dans l'ordre exact de leur numéro d'index dans le code (voir le tableau des outils et fonctions utilisés dans le script page 21). En dessous du cadre de saisie des paramètres se situe un cadre de propriétés des paramètres. Ici, de nombreuses propriétés sont définies, par exemple si le champ est obligatoire ou les éventuelles valeurs par défaut affichées.

Une fois l'importation du script terminée, il faut rédiger l'aide de l'outil. L'interface de saisie de l'aide est disponible avec un clic droit sur le script fraîchement importé dans notre boîte à outils et en sélectionnant « Description d'élément... », puis en cliquant sur « Mise à jour » dans la fenêtre qui s'ouvre, voir illustration 18.

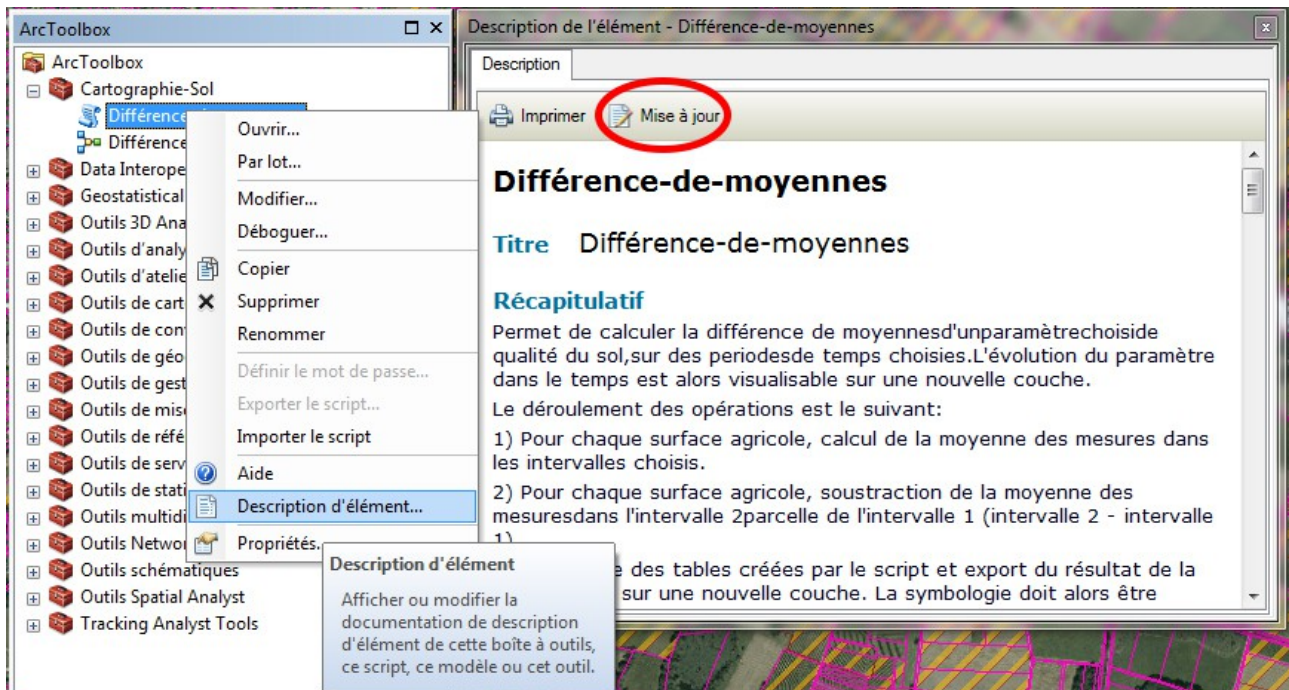


Illustration 18: Après avoir cliqué sur "Mise à jour", un formulaire de saisie de l'aide apparaît avec tous les paramètres du script. Il s'agit alors de rédiger des indications qui aideront l'utilisateur à remplir les champs de saisie de paramètres correctement.

Après avoir rédigé l'aide, l'outil est prêt à être utilisé. C'est ce qui sera développé dans le point suivant.

Présentation de l'outil et exemples d'utilisation

Dans cette section, l'interface graphique de l'outil sera présentée. Deux exemples d'utilisation seront détaillés.

Comme nous pouvons le voir sur l'illustration 19, l'outil demande 8 paramètres à l'utilisateur. L'espace de travail est saisi dans le champ de la géodatabase. La table en entrée doit être la table qui contient les analyses de sol. Les années de début et de fin du premier intervalle sont définies par l'utilisateur, tout comme les années de début et de fin du deuxième intervalle. Le champ est sélectionné dans une liste constituée à partir des noms de champs de la table en entrée. Et pour terminer, la profondeur d'échantillonnage est demandée, l'utilisateur choisit entre deux possibilités pré-établies : « 2 - 20 cm » ou « 20 - 40 cm ». Pour ce dernier champ, une valeur par défaut de « 2 - 20 cm » est définie car elle représente une très grande majorité : 5529 échantillons sur 5898 au moment de la rédaction de ce rapport.

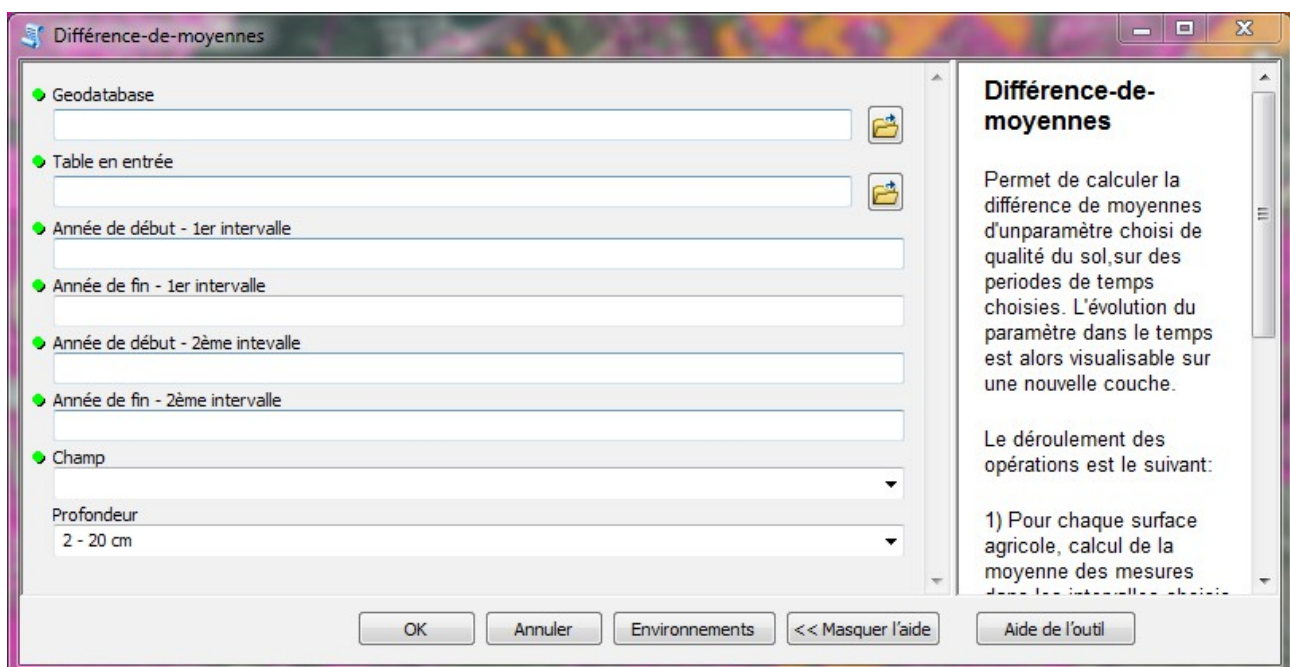


Illustration 19: Vue de l'outil avec les différents champs de saisie des paramètres. A droite, une aide est affichée (si elle est masquée, cliquer sur "Afficher l'aide >>") et renseigne l'utilisateur sur chaque paramètre à saisir.

L'outil présenté, nous pouvons réaliser un test en choisissant le premier intervalle de 1993 à 2003 et le deuxième intervalle 2004 à 2014. Le paramètre de qualité du sol étudié sera le pH et la profondeur d'échantillonnage 2 - 20 cm, voir illustration 20.

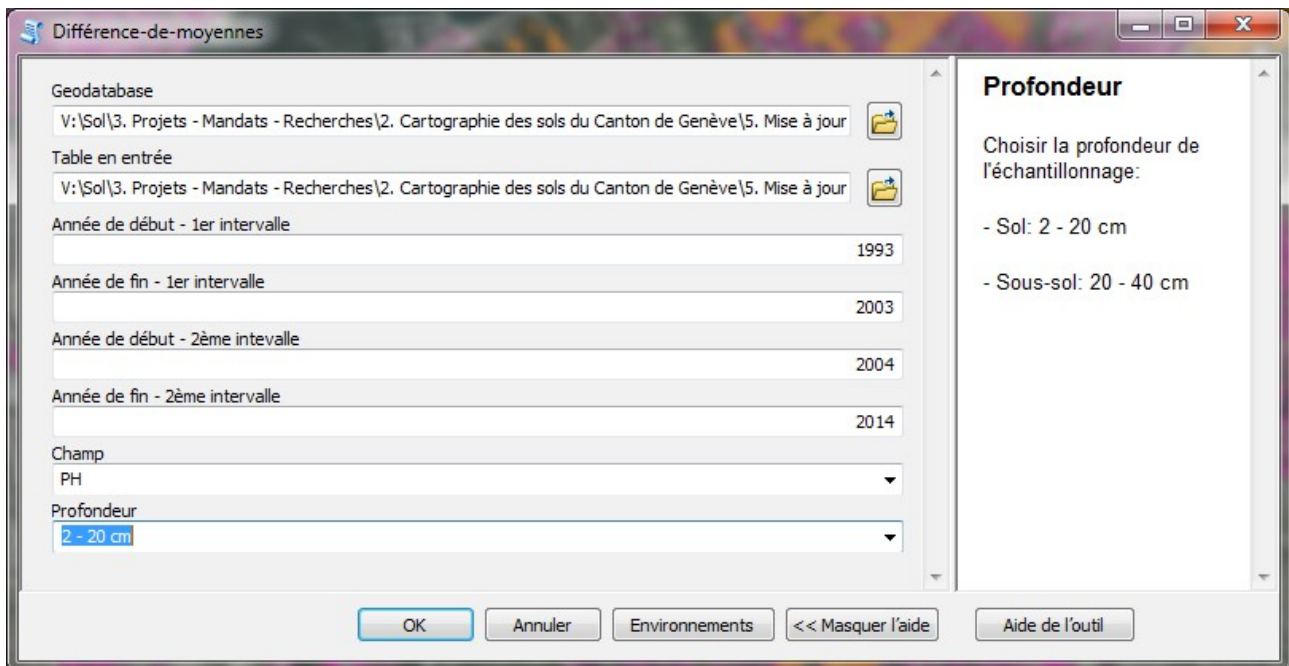


Illustration 20: Exemple d'utilisation de l'outil.

Une fois le script terminé avec succès, voir illustration 21, la nouvelle couche est ajoutée à l'espace de travail et affichée automatiquement. Il faut alors travailler la symbologie afin de représenter l'évolution du paramètre. Après une brève mise en page, le résultat de ce traitement sous forme d'une carte représentant l'évolution du pH est visible en annexe 3. Pour comparaison, une autre carte réalisée avec les mêmes paramètres mais avec une profondeur d'échantillonnage de « 20 - 40 cm » est visible en annexe 4.

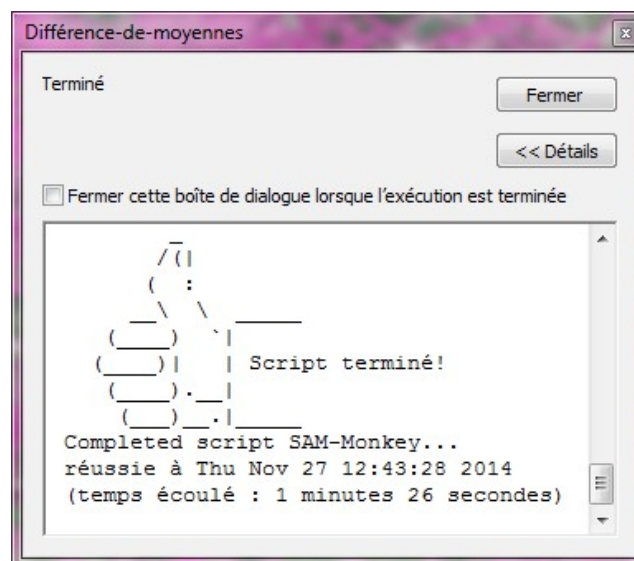


Illustration 21: Le script est terminé, il faut travailler la symbologie de la nouvelle couche et faire une mise en page avant d'exporter la carte.

Conclusion et pistes de développement

La table des analyses des sols cultivés à été mise à jour tout comme la couche des surfaces agricoles. Les données sont prêtes pour les mises à jour des prochaines années.

L'exploitation de ces données a jusqu'à présent toujours été faite à la main. Suite à ce travail, un outil programmé en python permet de réaliser des cartes d'évolution d'un paramètre de qualité du sol. Cet outil est à disposition du groupe sols et substrats et permet de réaliser en moins d'une minute trente des manipulations qui pouvaient prendre plusieurs dizaines de minutes.

La nécessité d'exploitation des données sera d'autant plus grande que la masse de données augmente. Ainsi, dans le but de réaliser des analyses toujours plus poussées, il serait intéressant de développer un outil semblable à celui codé durant ce stage, à l'exception qu'il permettrait de ramener l'évolution d'un paramètre par année et non par moyenne sur des durées empiriquement décidées par l'utilisateur, voir formule d'exemple ci-dessous.

$$\begin{array}{l} 1995 = 6 [pH] \\ 2003 = 7 [pH] \end{array} \rightarrow +1 [pH] / 8 \text{ ans} = +0,125 / \text{an}$$

L'outil développé durant ce stage restant important puisqu'il permet justement à l'utilisateur de définir lui même les paramètres de temps. Les moyennes étant facilement comprises par le grand public, même après codage d'un outil plus poussé, les cartes créées avec le script présenté dans ce rapport présentent toujours un intérêt en matière de communication.

Un aspect important à ne pas négliger est de publier les mises à jour de la couche des surfaces agricoles sur le site du SITG. En effet, les données présentes semblent dater de 2005, les modifications apportées en 2013 par Alice Johannes semblent ne pas avoir été mises en ligne. La mise à disposition de ces données est importante, comme nous avons pu le voir à la page 6. Il serait donc souhaitable que ces données soient partagées et de renseigner clairement la date de la dernière mise à jour dans les métadonnées de la couche, ce qui n'est pas le cas pour l'instant.

En annexe 5, nous pouvons visualiser un résumé statistique pour chacune des deux cartes créées précédemment (voir page 37 et annexes 3 et 4). Il est intéressant de constater qu'il y a des valeurs si extrêmes qu'elles en sont aberrantes. Il s'agit d'erreurs qu'il conviendra de répertorier et de corriger si possible.

Ce stage au groupe sols et substrats m'a permis de me retrouver en contexte professionnel dans le domaine de la géomatique. Cela a été très enrichissant au niveau de l'expérience acquise, notamment en programmation python. En effet, durant les cours théoriques lors de la formation à l'université de Genève, beaucoup de notions sont abordées. Pendant les exercices pratiques effectués, nous restons cependant assistés. Se retrouver seul face à un problème encore jamais abordé, avec pour limites ses propres connaissances, pousse à s'améliorer rapidement.

Bibliographie

- 1: Confédération Suisse, *Ordonnance sur les paiements directs versés dans l'agriculture*, 2013
- 2: Service de géomatique et de l'organisation de l'information (SGOI), <http://ge.ch/sitg>, Le territoire genevois à la carte
- 3: Johannes Alice, *Mise à jour de la carte d'analyses des sols agricoles du canton de Genève*, 2013
- 4: ESRI, <http://resources.arcgis.com/en/help/main/10.1/index.html>, ArcGIS resources - Help Library

Logiciels utilisés

- **LibreOffice 4.3.4.1 Writer** pour la rédaction et mise en page de ce rapport.
- **Microsoft Excel 2013** pour l'ajout de données dans la table des analyses de sol.
- **ArcGIS ArcCatalog 10.2.2** pour la gestion de données ArcGIS et la visualisation de métadonnées.
- **ArcGIS ArcMap 10.2.2** pour la saisie des surfaces agricoles, la création de l'interface de l'outil, l'affichage et la réalisation de cartes.
- **Notepad++ 6.6.9** pour la rédaction du code python.

Annexes

Annexe 1 : Agencement des outils et des tables pour une mise à jour efficace des données.

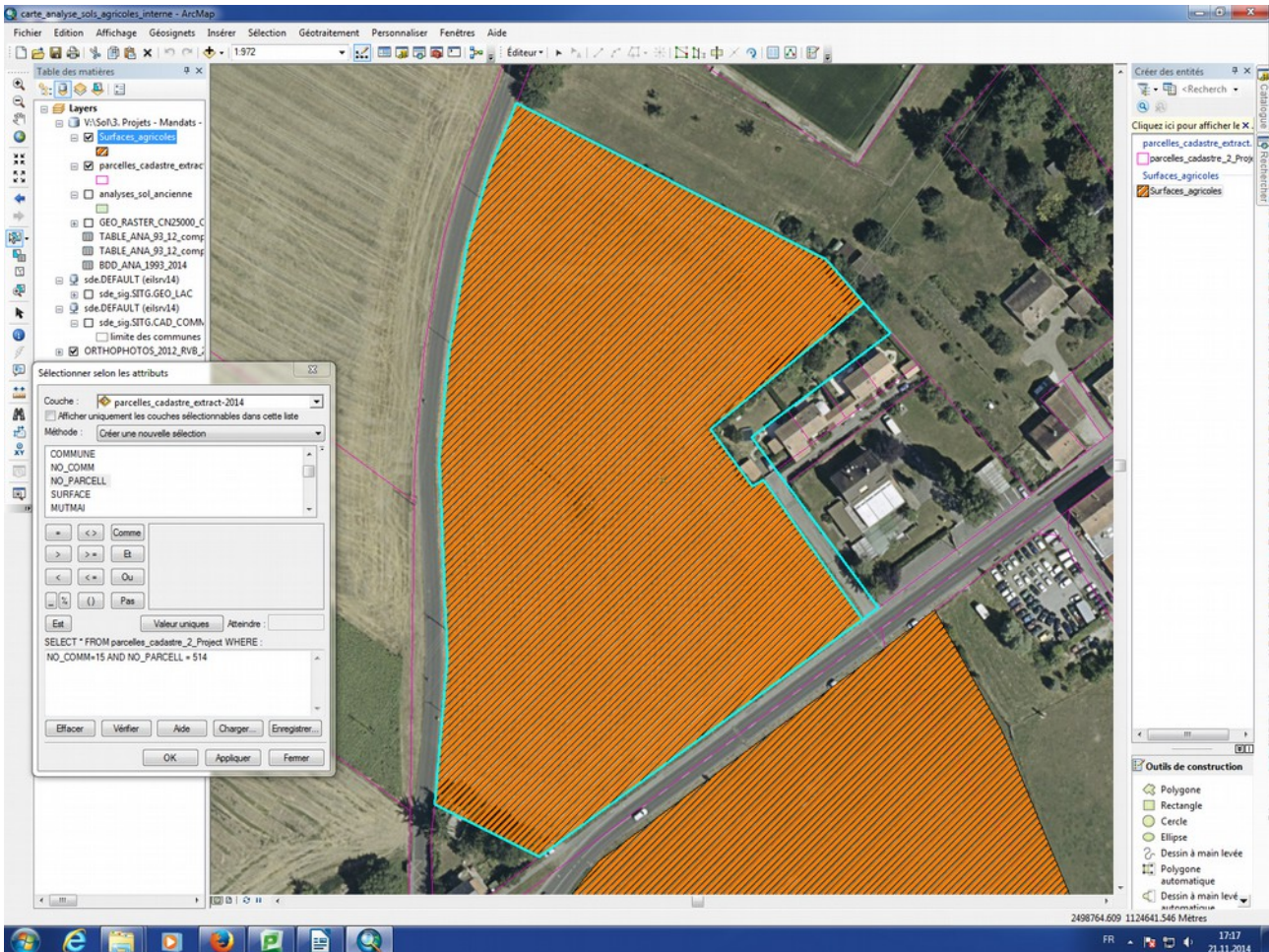


Table - BDD_ANA_1993_2014

| OBJET | ID_ANA | SURFACE | FIABILITE | REMARQU_PLACE | COMM | PARCELLE_CADASTRALE | NOM_PARCELLE | NOM | PRENOM | COUVRE |
|-------|--------|---------|------------|--------------------------------------|------|---------------------|-------------------------|---------------|---------------|-------------|
| 4321 | 11709 | 2845 | F | <Nul> | 11 | 2846/1364 | 3-Chêne Commune | CORTHAY | AYMON | Grandes c |
| 4323 | 11711 | 2846 | F | <Nul> | 11 | 2456/2455/2454 | 5-Montagny Rhône | CORTHAY | AYMON | Grandes c |
| 4325 | 11713 | 2847 | F | <Nul> | 20 | 480 | | BERSIER-RAMUS | PHILPPE | Grandes c |
| 4330 | 11718 | 2848 | F | 7432 au lieu de 7430 probablement | 41 | 6716-7260-7430 | Les pièces | DUGERDIL | DANIEL | Grandes c |
| 4334 | 11752 | 2850 | F | Localisation certaine | 15 | 272275 | 4 Autroville | RODER | HANSRUEDI | Grandes c |
| 4335 | 11723 | 2851 | F | <Nul> | 41 | 8101 | Fabry forêt | DUGERDIL | DANIEL | Grandes c |
| 4334 | 11722 | 2852 | F | <Nul> | 41 | 8101 | Fabry route | DUGERDIL | DANIEL | Grandes c |
| 4338 | 11724 | 2853 | F | <Nul> | 41 | 10081 | Mont Fleury | DUGERDIL | DANIEL | Grandes c |
| 4339 | 11727 | 2854 | F | <Nul> | 41 | 8752 | Vigne Joseph | DUGERDIL | DANIEL | Grandes c |
| 4357 | 11745 | 2855 | F | <Nul> | 47 | 404597 | 7a Richesem | RODER | HANSRUEDI | Grandes c |
| 4358 | 11746 | 2856 | F | <Nul> | 15 | 514 | 5 Sorbiers | RODER | HANSRUEDI | Grandes c |
| 4359 | 11747 | 2857 | F | 2 analyses sur une surface | 15 | 575 | 2b Bois Fromager | RODER | HANSRUEDI | Grandes c |
| 4360 | 11748 | 2857 | F | 2 analyses sur une surface | 15 | 575 | 2a Bois Fromager | RODER | HANSRUEDI | Grandes c |
| 4361 | 11749 | 2858 | F | <Nul> | 15 | 574 | 1a Mûcher | RODER | HANSRUEDI | Grandes c |
| 4363 | 11751 | 2858 | F | 2 analyses sur une surface | 15 | 574 | 1b Mûcher | RODER | HANSRUEDI | Grandes c |
| 4362 | 11750 | 2859 | F | <Nul> | 15 | 472 | 3a Baraques | RODER | HANSRUEDI | Grandes c |
| 4377 | 11785 | 2860 | comm2, 192 | <Nul> | 28 | 1466 | Hermance II | CHOLLET | PIERRE | Grandes c |
| 4388 | 11776 | 2861 | F | <Nul> | 48 | 3688 | Vessey 10 | VECCHO | AGRICULTEURS | Grandes c |
| 4309 | 11777 | 2862 | F | <Nul> | 48 | 6179 | Vessey 12 | VECCHO | AGRICULTEURS | Grandes c |
| 5337 | 12725 | 2863 | F | <Nul> | 47 | 93 | Chappu 100 | FALCONNET | YVES | Grandes c |
| 5338 | 12726 | 2864 | F | <Nul> | 47 | 82 | Chappu 300 | FALCONNET | YVES | Grandes c |
| 4407 | 11795 | 2865 | F | <Nul> | 7 | 2.29023E+15 | Moller | FOEX | JEAN-FRANCOIS | Grandes c |
| 4409 | 11797 | 2866 | F | <Nul> | 18 | 10189 | Grard | FOEX | JEAN-FRANCOIS | Grandes c |
| 4413 | 11801 | 2867 | F | <Nul> | 7 | 7272 | Ans-la-Ville | FOEX | JEAN-FRANCOIS | Grandes c |
| 4416 | 11804 | 2868 | F | <Nul> | 35 | 9 | Blonin | CHEVALLEY | FRERES | Grandes c |
| 4427 | 11815 | 2869 | F | <Nul> | 15 | 93 | Marteman | MERCER | CLAUDE | Viticulture |
| 4428 | 11816 | 2870 | F | <Nul> | 38 | 2142 | Juas-bas | CARRAT | ANDRE | Grandes c |
| 4430 | 11818 | 2871 | F | <Nul> | 38 | 2240 | Village 2 | CARRAT | ANDRE | Grandes c |
| 4529 | 11917 | 2871 | F | <Nul> | 38 | 2239 | Grand Pré | THORENS | <Nul> | Grandes c |
| 4531 | 11919 | 2871 | F | <Nul> | 38 | 2240 | Grand pré Carrat | THORENS | <Nul> | Grandes c |
| 4429 | 11817 | 2872 | F | <Nul> | 38 | 2236 | Village 1 | CARRAT | ANDRE | Grandes c |
| 4440 | 11828 | 2873 | F | 2 analyses pour 1 surface cf DA11829 | 14 | 2496/2480 | Haut de la vigne Butte | FAVRE | BERTRAND | Grandes c |
| 4441 | 11829 | 2873 | F | 2 analyses pour 1 surface cf DA11828 | 14 | 2496/2480 | Haut de la vigne Extrém | FAVRE | BERTRAND | Grandes c |
| 4442 | 11830 | 2874 | F | <Nul> | 14 | 3064 | Petit loton 3 | FAVRE | BERTRAND | Grandes c |
| 4476 | 11841 | 2875 | F | <Nul> | 7 | 2811 | La barraque | CAOUDX | NICOLAS | Viticulture |
| 4485 | 11850 | 2876 | F | Parcelle 6696 faux 5908 | 41 | 6696/2072/2073 | Petite Montagne Bas | CRETEGNY | WILLY | Viticulture |
| 4486 | 11851 | 2877 | F | <Nul> | 41 | 5908 | Petite Montagne Haut | CRETEGNY | WILLY | Viticulture |
| 4487 | 11852 | 2880 | F | <Nul> | 41 | 6698 | Tour d'eau | CRETEGNY | WILLY | Viticulture |
| 4488 | 11853 | 2881 | F | <Nul> | 41 | 6939 | Les Adobes | CRETEGNY | WILLY | Viticulture |
| 4489 | 11854 | 2882 | F | <Nul> | 41 | 5389 | Cimetière | CRETEGNY | WILLY | Viticulture |
| 4493 | 11858 | 2883 | F | <Nul> | 41 | 6150/6369 | Chardonnay village Pes | CRETEGNY | WILLY | Viticulture |
| 4494 | 11859 | 2884 | F | <Nul> | 41 | 2045 | Tuilère Vignier | CRETEGNY | WILLY | Viticulture |
| 4495 | 11860 | 2885 | F | <Nul> | 41 | 5865/6244 | Marnes 1 | CRETEGNY | WILLY | Viticulture |
| 4496 | 11861 | 2886 | F | <Nul> | 41 | 4312/8705 | Galardes | CRETEGNY | WILLY | Viticulture |
| 4497 | 11862 | 2887 | F | Entoure le hangar | 41 | 6801 | Vigne Hangar | CRETEGNY | WILLY | Viticulture |

Surfaces_agricoles

| OBJECTID | SHAPE * | DATE_FIN | SHAPE_Length | SHAPE_Area * | ID_SURFACE_AGRICOLE | PREMIERE_DATE |
|----------|----------|----------|--------------|--------------|---------------------|---------------|
| 2856 | Polygone | 9999 | 677.761635 | 22452.670775 | 2856 | 2009 |

Surfaces_agricoles

Annexe 2 : Code python commenté tel qu'il est utilisé en script. Remarquons que comme l'encodage des caractères est en utf-8, il faut remplacer les caractères spéciaux par leur équivalent utf-8 dans les messages affichés dans la console python. Ainsi, par exemple un « é » doit être remplacé par « \xc3\xa9 » ou un « è » doit être remplacé par « \xc3\xa8 ».

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

#=====#
# Dernière mise à jour: 15.12.2014 #
# Description: script python pour réaliser une cartographie de #
# l'évolution de différents paramètres de qualité du sol dans #
# ArcGIS. #
# Codé en 2014 par Luc Rebetez pour hepia Lullier, section #
# agronomie, groupe étude des sols et substrats #
# Travail réalisé dans le cadre du CAS en géomatique de #
# l'université de Genève. #
# Travail effectué sous direction de Pascal Boivin, sous #
# supervision de Frédéric Lamy et Alice Johannes #
# Répondants au certificat: Anthony Lehmann(UNIGE) et Alain #
# Dubois(hepia) #
# Jury du certificat: Anthony Lehmann (UNIGE), Alain Dubois #
# (hepia) et Antoine Besson (hepia) #
# Contact: luc.rebetez@gmail.com #
#=====#

# Import du module arcpy

import arcpy

# Les fichiers avec un nom existant seront écrasés

arcpy.env.overwriteOutput = True

# Definition des variables d'environnement et de la table en
entrée

##### Paramètre 0

arcpy.env.workspace = arcpy.GetParameterAsText(0) #
L'environnement géodatabase est défini par l'utilisateur

##### Paramètre 1
```



```
InTable = arcpy.GetParameterAsText(1) # La table des analyses en
entrée est définie par l'utilisateur

arcpy.AddMessage("Environnement et table en entr\xc3\xa9e
d\xc3\xa9finis avec succ\xc3\xa8s!")

# 1 ### Sélection et traitement par moyenne du 1er intervalle de
temps

arcpy.AddMessage("S\xc3\xa9lection de la 1\xc3\xa8re
p\xc3\xa9riode demand\xc3\xa9e...")

# Selection des dates voulues pour la première moyenne puis
export vers table "TtoT_1"

##### Paramètre 2

date_1 = arcpy.GetParameterAsText(2) # Saisie par l'utilisateur
de la date de début du 1er intervalle

##### Paramètre 3

date_2 = arcpy.GetParameterAsText(3) # Saisie par l'utilisateur
de la date de fin du 1er intervalle

##### Paramètre 7

prof = arcpy.GetParameterAsText(7) # Saisie par l'utilisateur de
la profondeur prise en compte: sol (2-20cm) ou sous-sol (20-40cm)

Expression = "ANNEE_ANALYSE >= "+str(date_1) +" AND ANNEE_ANALYSE
<= "+str(date_2)+" AND PROFONDEUR = '"+str(prof)+"'" # Expression
qui permet de sélectionner l'intervalle saisi par l'utilisateur,
les chiffres doivent être changées en format "string", la
profondeur est aussi prise en compte

Out = arcpy.env.workspace # La table contenant les données du 1er
intervalle sera créée vers la géodatabase courante
```

```
arcpy.TableToTable_conversion(InTable,Out , "TtoT_1", Expression)
# On fait appel à l'outil "TableToTable_conversion" pour créer la
table "TtoT_1" et sélectionner au passage les dates et la
profondeur qui nous intéressent grâce au paramètre "Expression"

# On supprime les variables "Out" et "Expression" qui seront
utilisées plus tard

del Out

del Expression

arcpy.AddMessage("Sélection effectuée avec
succès!")

arcpy.AddMessage("Traitement de la 1ère sélection
par moyenne...")

# Traitement par moyenne des années sélectionnées

intable = "TtoT_1" # On spécifie la table en entrée pour l'outil
"Statistics_analysis"

Out = "TtoT_1_Res" # La table après traitement sera sommée
"TtoT_1_Res"

##### Paramètre 6

champ = arcpy.GetParameterAsText(6) # Constituant du sol à
cartographeur saisi par l'utilisateur

champ_1 = champ+" MEAN" # Mise en forme du champ pour qu'il soit
accepté par l'outil "Statistics_analysis": on ajoute " MEAN" à la
fin pour indiquer quel traitement effectuer (moyenne)

arcpy.Statistics_analysis(intable, Out, champ_1,
"SURFACE_AGRICOLE") # On fait appel à l'outil
"Statistics_analysis" pour calculer la moyenne du constituant du
sol sélectionné dans le 1er intervalle

# On supprime les variables "Out" et "intable" qui seront
utilisées plus tard
```

```
del Out

del intable

arcpy.AddMessage("Traitement de la 1\rc3\ra8re s\rc3\ra9lection
effectu\rc3\ra9 avec succ\rc3\ra8s!")

arcpy.AddMessage("Renommage du champ...")

# Modification du nom du champ pour qu'il soit évocateur du
constituant du sol cartographié

intable_1 = "TtoT_1_Res" # On spécifie la table en entrée, ne pas
del car on utilise ce nom pour une jointure plus tard

NomChamp_1 = "MEAN_"+champ # On met en forme le nom de champ pour
indiquer que la moyenne a été appliquée et qu'il soit utilisable
dans l'outil "AlerField_management"

Renomme_1 = champ+"_1" # On ajoute un 1 à la fin du nom car c'est
la première intervalle de temps

arcpy.AlterField_management(intable_1, NomChamp_1, Renomme_1,
Renomme_1) # On applique l'outil "AlterField_management" pour
changer le nom du champ afin qu'il soit cohérent et évocateur

arcpy.AddMessage("Champ renomm\rc3\ra9 avec succ\rc3\ra8s!")

# 2 ### Sélection et traitement par moyenne du 2ème intervalle de
temps

arcpy.AddMessage("S\rc3\ra9lection de la 2\rc3\ra8me
p\rc3\ra9riode demand\rc3\ra9e...")

# Selection des dates voulues pour la première moyenne puis
export vers table "TtoT_1"

##### Paramètre 4
```

```
date_3 = arcpy.GetParameterAsText(4) # Saisie par l'utilisateur
de la date de début du 2eme intervalle

##### Paramètre 5

date_4 = arcpy.GetParameterAsText(5) # Saisie par l'utilisateur
de la date de fin du 2eme intervalle

Expression = "ANNEE_ANALYSE >= "+str(date_3) +" AND ANNEE_ANALYSE
<= "+str(date_4)+" AND PROFONDEUR = '"+str(prof)+"'" # Expression
qui permet de sélectionner l'intervalle saisi par l'utilisateur,
les chiffres des années doivent être changées en format "string",
la profondeur est aussi prise en compte

Out = arcpy.env.workspace # La table contenant les données du
2eme intervalle sera créée vers la géodatabase courante

arcpy.TableToTable_conversion(InTable,Out , "TtoT_2", Expression)
# On fait appel à l'outil "TableToTable_conversion" pour créer la
table "TtoT_2" et sélectionner au passage les dates et la
profondeur qui nous intéressent grâce au paramètre "Expression"

# On supprime les variables "Out" et "Expression" qui peuvent
être utilisées plus tard

del Out

del Expression

arcpy.AddMessage("Sélection effectuée avec
succès!")

arcpy.AddMessage("Traitement de la 2ème sélection
par moyenne...")

# Traitement par moyenne des années sélectionnées

intable = "TtoT_2" # On spécifie la table en entrée pour l'outil
"Statistics_analysis"

Out = "TtoT_2_Res" # La table après traitement sera sommée
"TtoT_2_Res"
```

```
arcpy.Statistics_analysis(intable, Out, champ_1,
"SURFACE_AGRICOLE") # On fait appel à l'outil
"Statistics_analysis" pour calculer la moyenne du constituant du
sol sélectionné dans le 2eme intervalle

# On supprime les variables "Out" et "intable" qui peuvent être
utilisées plus tard

del Out

del intable

arcpy.AddMessage("Traitement de la 2\xc3\xa8me s\xc3\xa9lection
effectu\xc3\xa9 avec succ\xc3\xa8s!")

arcpy.AddMessage("Renommage du champ...")

# Modification du nom du champ

intable_2 = "TtoT_2_Res" # On spécifie la table en entrée, ne pas
del car on utilise ce nom pour une jointure plus tard

Renomme_2 = champ+"_2" # On ajoute un 2 à la fin du nom car c'est
la deuxième intervalle de temps

arcpy.AlterField_management(intable_2, NomChamp_1, Renomme_2,
Renomme_2) # On applique l'outil "AlterField_management" pour
changer le nom du champ afin qu'il soit cohérent et évocateur

arcpy.AddMessage("Champ renomm\xc3\xa9 avec succ\xc3\xa8s!")

# 3 ### Calcul de la différence des moyennes

# Jointure des tables de moyennes

arcpy.AddMessage("Jointure des deux tables...")
```

```
arcpy.JoinField_management(intable_1, "SURFACE_AGRICOLE",
intable_2, "SURFACE_AGRICOLE", Renomme_2) # On applique l'outil
"JoinField_management" pour joindre les deux intervalles selon la
surface agricole, "TtoT_1_Res" devient alors la table qui
permettra de faire la différence des moyennes entre les deux
intervalles choisis

arcpy.AddMessage("Jointure effectu\xc3\xa9e avec succ\xc3\xa8s!")

# Ajout d'un champ pour calculer la différence des moyennes

arcpy.AddMessage("Ajout d'un champ \"Difference\" pour calculer
la diff\xc3\xa9rence des deux moyennes...")

Champcalcul = "Difference_"+champ+"_"+date_1+"_a_"+date_4 #
Mise en forme du nom du champ de calcul pour qu'il soit cohérent
et évocateur, on ajoute "Difference_" au début, on ajoute le nom
du paramètre choisi par l'utilisateur et les dates de
l'intervalle total

arcpy.AddField_management("TtoT_1_Res", Champcalcul, "FLOAT",
"", "", "", "", "NULLABLE", "NON_REQUIRED", "") # On fait appel à
l'outil "AddField_management" pour ajouter le champ de calcul
correctement nommé

arcpy.AddMessage("Champ ajout\xc3\xa9 avec succ\xc3\xa8s!")

# Calcul de la différence des moyennes dans le nouveau champ

arcpy.AddMessage("Calcul de la diff\xc3\xa9rence des
moyennes...")

Expression = "!" + Renomme_2 + "!" - "!" + Renomme_1 + "!" # Mise en forme
de l'expression en python pour l'outil
"CalculateField_management", la différence des moyennes est
écrite ici, on soustrait le 1er intervalle au 2ème intervalle
pour avoir l'évolution du paramètre observé
```

```
arcpy.CalculateField_management("TtoT_1_Res", Champdec calcul,
Expression, "PYTHON", "") # On fait appel à l'outil
"CalculateField_management" qui applique l'expression définie ci-
dessus et effectue la soustraction dans le champ créé à cet effet
(Champdec calcul)

arcpy.AddMessage("Calcul effectu\xc3\xa9 avec succ\xc3\xa8s!")

# 4 ### Elimination des objets créés inutiles et affichage du
résultat

# Export des données de la couche de surfaces agricoles

# Il faut changer le tiret pour que "prof" soit accepté comme nom
de couche, le tiret est un caractère non valide

arcpy.AddMessage("Copie de la couche des surfaces agricoles...")

if prof == '2 - 20 cm': # On teste la valeur de "prof" et si elle
correspond à la valeur du sol, on modifie sa valeur en remplaçant
les espaces et le tiret par des tirets bas
    prof = "2_20_cm"
elif prof == '20 - 40 cm': # On teste la valeur de "prof" et si
elle correspond à la valeur du sous-sol, on modifie sa valeur en
remplaçant les espaces et le tiret par des tirets bas, "elif" est
la contraction de "else if"
    prof = "20_40_cm"
else :
    pass # Bien que peu probable, si une autre valeur de
profondeur est entrée que celles attendues, le script continue si
cela ne génère pas d'erreur; bien que pas obligatoire, il est
toujours préférable de clore une instruction "if" par un "else"
qui traite toutes les autres possibilités

NomCouche =
"Surfaces_Agricoles_"+champ+"_"+date_1+"_a_"+date_4+"_"+prof #
Mise en forme du nom de la couche pour qu'il soit cohérent et
évocateur, ajout de la profondeur dans le nom
```

```
arcpy.FeatureClassToFeatureClass_conversion("Surfaces_agricoles",
arcpy.env.workspace, NomCouche) # Copie de la couche originale
des surfaces agricoles pour éviter de l'altérer

arcpy.AddMessage("Copie de la couche des surfaces agricoles
effectu\xc3\xa9e avec succ\xc3\xa8s!")

# Jointure à la couche des surfaces agricoles

arcpy.AddMessage("Jointure du r\xc3\xa9sultat de la
diff\xc3\xa9rence des moyennes \xc3\xa0 la couche des surfaces
agricoles...")

arcpy.JoinField_management(NomCouche, "ID_SURFACE_AGRICOLE",
"TtoT_1_Res", "SURFACE_AGRICOLE", Champdecalcul) # On fait appel
à l'outil "JoinField_management" pour joindre le champ du calcul
de la différence des moyennes à notre nouvelle couche créée pour
l'occasion dont il faudra travailler la symbologie

arcpy.AddMessage("Jointure effectu\xc3\xa9e avec succ\xc3\xa8s!")

# Nettoyage des tables inutiles

arcpy.AddMessage("Nettoyage des tables inutiles...")

# On fait appel à l'outil "Delete_management" pour effacer les
tables inutiles créées durant le script

arcpy.Delete_management("TtoT_1")
arcpy.Delete_management("TtoT_2")
arcpy.Delete_management("TtoT_1_Res")
arcpy.Delete_management("TtoT_2_Res")

arcpy.AddMessage("Nettoyage effectu\xc3\xa9 avec succ\xc3\xa8s!")

# Ajout de la couche à l'espace de travail

mxd = arcpy.mapping.MapDocument("CURRENT") # On définit l'espace
de travail courant
```



```

df = arcpy.mapping.ListDataFrames(mxd, "*")[0] # On liste les
"Data frames" disponibles

couche = arcpy.mapping.Layer(NomCouche) # On selectionne la
couche créée en fin de script

arcpy.mapping.AddLayer(df,couche,"TOP") # On applique la fonction
"mapping.AddLayer" pour ajouter la couche au data frame

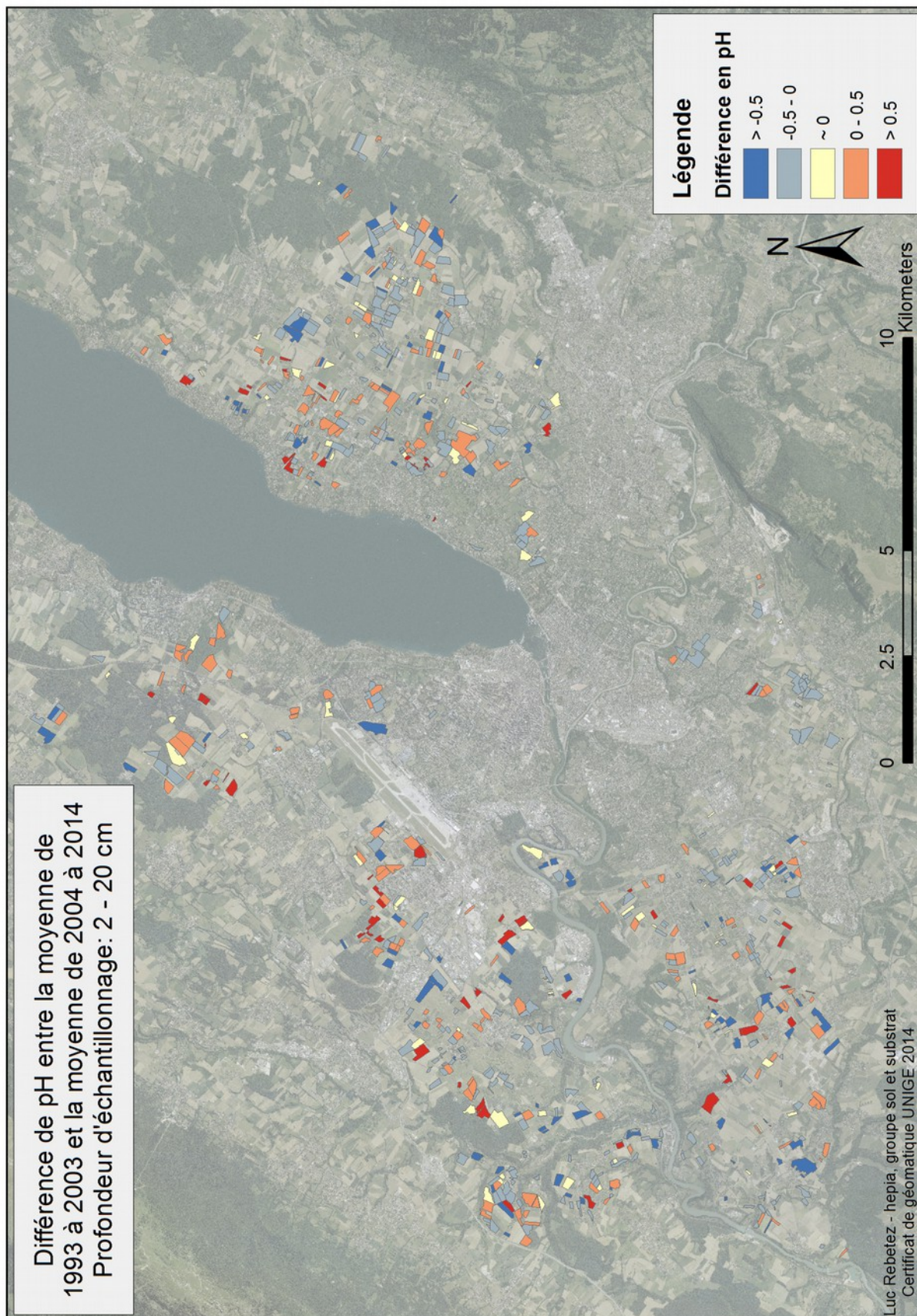
arcpy.RefreshActiveView() # On rafraichis la vue pour afficher
les données

arcpy.AddMessage("      _")
arcpy.AddMessage("      /(|")
arcpy.AddMessage("      (:")
arcpy.AddMessage(" _\ \ _____")
arcpy.AddMessage(" (____) `|")
arcpy.AddMessage(" (____)|   | Script termin\xc3\xa9!")
arcpy.AddMessage(" (____)._|")
arcpy.AddMessage(" (____)_.|_____")

#arcpy.AddMessage("Vous pouvez travailler la symbologie de la
couche des surfaces agricoles, créée pour l'occasion, pour voir
l'évolution du paramètre choisi.")

```

Annexe 3 :

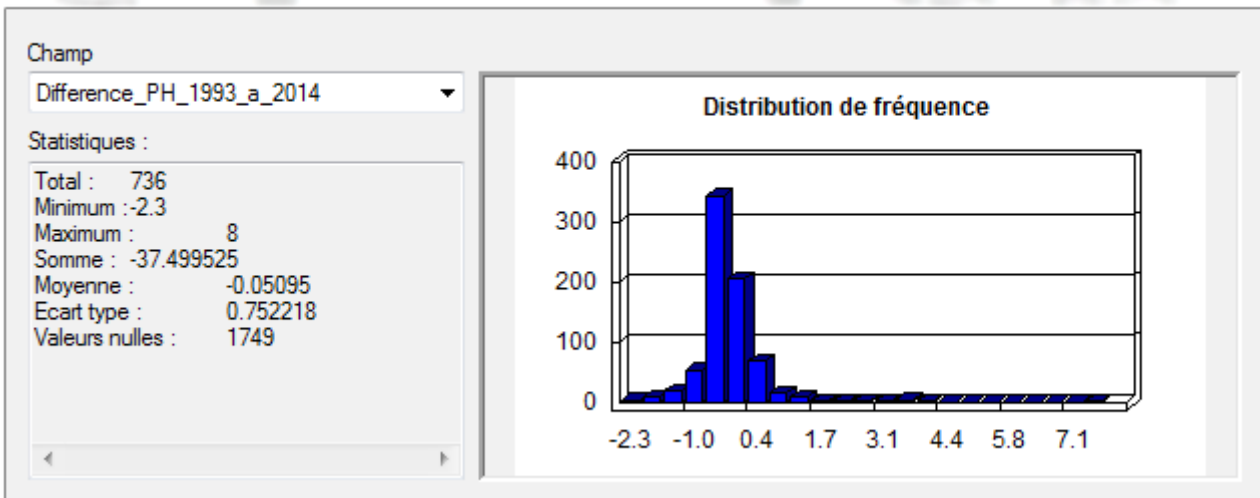


Annexe 4 :



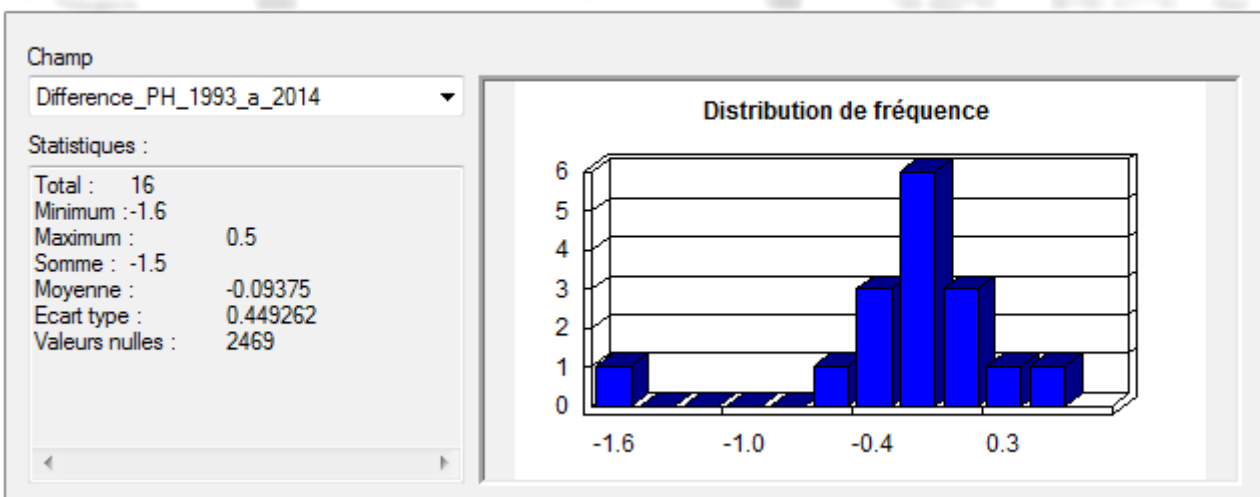
Annexe 5 :

Statistiques de Surfaces_Agricoles_PH_1993_a_2014_2_20_cm



Dans ce résumé statistique issu des données de la carte réalisée avec une profondeur d'échantillonnage de 2 - 20 cm, nous constatons des valeurs uniques extrêmes. Il s'agit probablement d'erreurs qui se sont glissées lors de manipulations de données. Pour améliorer la précision de l'ensemble, il faudrait vérifier toutes les valeurs dont la différence amène des résultats supérieurs à 1.7.

Statistiques de Surfaces_Agricoles_PH_1993_a_2014_20_40_cm



Dans ce résumé statistique issu des données de la carte réalisée avec une profondeur d'échantillonnage de 20 - 40 cm, les valeurs extrêmes sont moins prononcées, mais on constate aussi que les valeurs négatives inférieures à -0.5 mériteraient d'être vérifiées.