

# Chaîne « Risque » – EvolutionOverTime

## Table des matières

I. Présentation du script.....	2
I.1. Description.....	2
I.2. Traitements.....	3
I.3. Dépendances.....	4
II. Utilisation du script.....	5
II.1. Paramètres.....	5
II.2. Lancement.....	6
III. Exemples de résultats.....	7

## Index des illustrations

Illustration 1: Flux de données du script EvolutionOverTime.....	3
Illustration 2: Dépendances du script EvolutionOverTime avec la chaîne interne.....	4

# I. Présentation du script

## I.1. Description

**Note :** Ce script peut être utilisé avec des OCS d'origine différente (issues de données vecteurs type BD Topo, de données Pléiades stéréo, de donnée SPOT mono...), pour peu qu'elles utilisent la même nomenclature, et que la zone d'étude soit totalement incluse dans chacune (contrainte du croisement raster-vecteur). Bien entendu, les résultats sont dépendants de la qualité des OCS !

Le but est de **quantifier l'évolution de l'occupation du sol d'une parcelle**.

On considère une « OCS de référence », dite  $t_0$ , et une liste d'OCS postérieure, dites  $[t_{0+x}]$ .

L'évolution est seulement quantifiée pour chaque couple  $t / t+1$ .

Elle peut l'être suivant un taux minimal et/ou une surface minimale, pour une classe OCS donnée.

Il y a **quatre possibilités de paramétrage** :

- si le taux seul franchit son seuil (on ne renseigne pas de seuil pour la surface)
- si la surface seule franchit son seuil (on ne renseigne pas de seuil pour le taux)
- si au moins un des deux franchit son seuil, peu importe si l'autre n'indique pas d'évolution
- si les deux franchissent leur seuil respectif

Même si seulement un des deux paramètres est utilisé, les valeurs brutes sont produites pour les deux.

En sortie, trois cas de figure sont possibles :

- **aucune évolution** : le taux et/ou la surface de changement reste en-deça du seuil fixé
- **évolution positive** : le taux et/ou la surface de changement passe au-delà du seuil fixé, en valeur positive (= la classe a augmentée entre  $t$  et  $t+1$ )
- **évolution négative** : le taux et/ou la surface de changement passe au-delà du seuil fixé, en valeur négative (= la classe a diminuée entre  $t$  et  $t+1$ )

La limitation en nombre de caractères du nom des champs dans un vecteur empêche de savoir exactement quelle évolution est quantifiée dans chaque champ. Pour parer cela, un fichier texte est généré, rendant compte de façon explicite de la correspondance entre le nom du champ et l'évolution quantifiée.

Extrait de fichier texte produit :

```
def_evo_1 --> évolution entre t0 et t1, pour la classe 'bati' (label 11000) :
  evo_1 --> taux d'évolution brut (%)
  evo_s_1 --> surface d'évolution brute (m²)
Evolution quantifiée : taux à 10 % ET surface à 50 m²

def_evo_2 --> évolution entre t0 et t1, pour la classe 'vegetation' (label 23000) :
  evo_2 --> taux d'évolution brut (%)
  evo_s_2 --> surface d'évolution brute (m²)
Evolution quantifiée : taux à 10 % OU surface à 50 m²
```

## I.2. Traitements

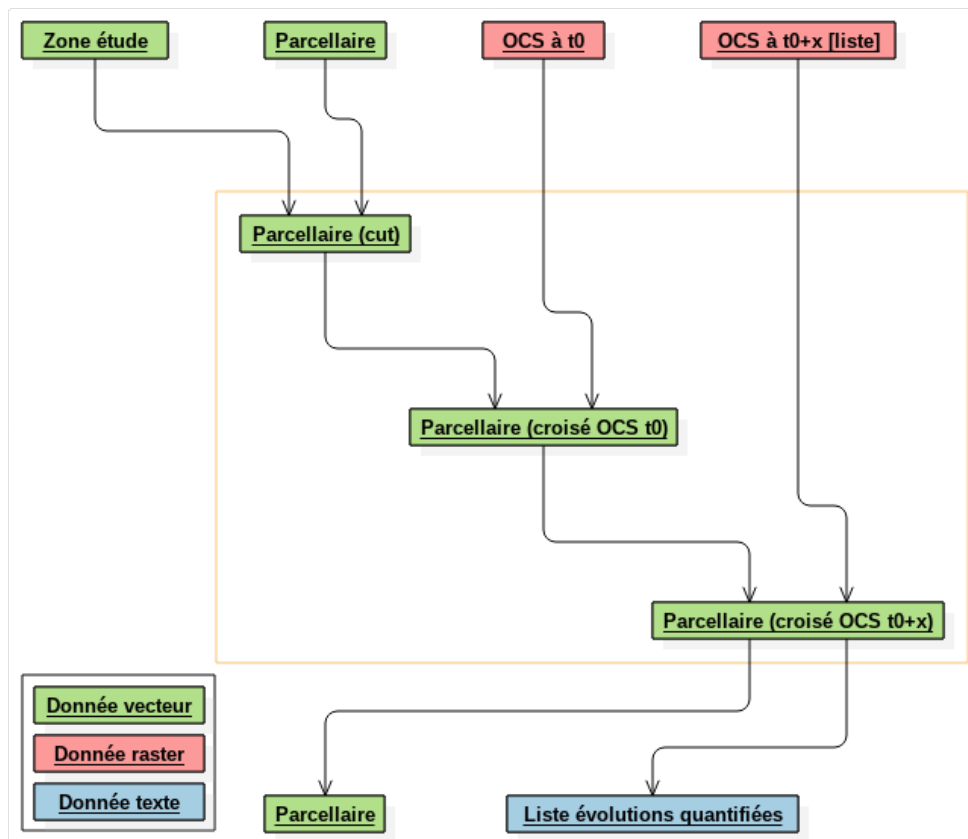


Illustration 1: Flux de données du script EvolutionOverTime.  
Les fichiers temporaires sont dans le cadre orange.

La 1<sup>re</sup> étape consiste à découper le parcellaire à l’emprise de la zone d’étude.

Il est ensuite croisé avec l’OCS à  $t_0$ , puis avec toutes les autres OCS à  $t_{0+x}$ .

Un requêtage SQL est alors réalisé, afin de quantifier, pour chaque parcelle, les évolutions sur les classes OCS renseignées par l’utilisateur (en termes de taux et/ou de surface).

Le fichier texte est également produit, afin de mieux comprendre les champs du fichier vecteur de sortie.

### I.3. Dépendances

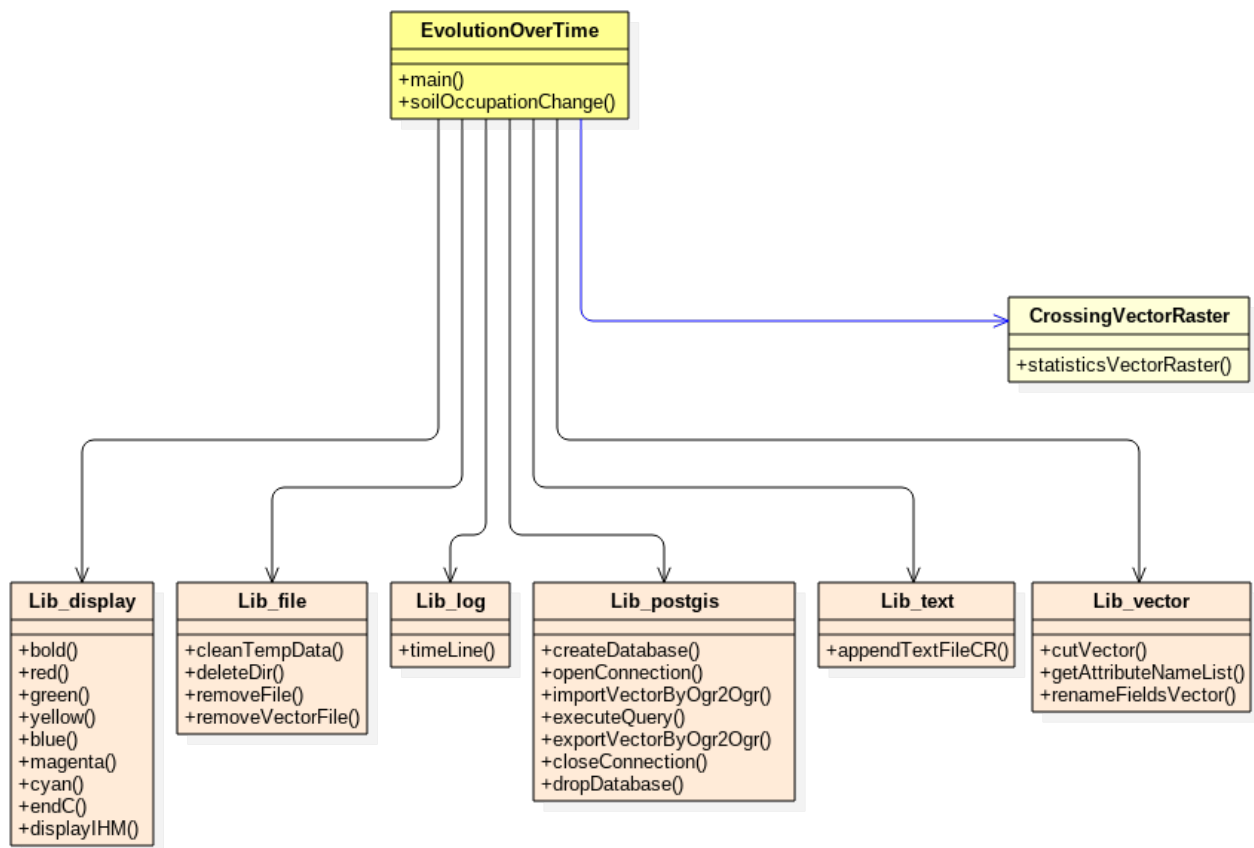


Illustration 2: Dépendances du script EvolutionOverTime avec la chaîne interne.

En plus de ces dépendances au reste de la chaîne interne, il fait appel aux bibliothèques ‘os’, ‘argparse’.

## II. Utilisation du script

### II.1. Paramètres

#### Paramètres spécifiques du script :

- **input\_plot\_vector (-in)** : fichier parcellaire (vecteur entrée)
- **output\_plot\_vector (-out)** : fichier parcellaire avec l'indicateur (vecteur sortie)
- **footprint\_vector (-emp)** : fichier d'emprise de la zone d'étude (vecteur entrée)
- **input\_t0\_file (-it0)** : classification OCS à  $t_0$  (= référence) (raster entrée)
- **input\_tx\_files\_list (-itxl)** : liste des classifications OCS à  $t_{0+x}$  (liste raster entrée)
- **evolutions\_list (-evol)** : liste des évolutions à quantifier, par classe (en taux et/ou surface). Par défaut :  
'11000:10:50:and 12000:10:50:and 21000:10:50:and 22000:10:50:and 23000:10:50:and'
- **class\_label\_dico (-cld)** : liste des correspondances de classes des fichiers OCS en entrée. Par défaut :  
'11000:Bati 12000:Route 21000:SolNu 22000:Eau 23000:Vegetation'

Seuls les paramètres en gras sont obligatoires.

En plus de ces paramètres spécifiques, le script fait appel à des paramètres liés à PostGIS (IP hôte, numéro de port, nom d'utilisateur, mot de passe, nom de la base de données, nom du schéma, encodage des caractères), liés aux fichiers (projection, valeur NoData, format/extension des rasters/vecteurs) et aux paramètres généraux (fichier log, sauvegarde des fichiers temp, ré-écriture sur fichier existant, niveau de debug).

#### Structure du paramètre 'evolutions\_list' :

##### **1:2:3:4**

- 1 = numéro de classe OCS (label)
- 2 = taux d'évolution (si 0, pas de prise en compte du taux dans la quantification)
- 3 = surface d'évolution (si 0, pas de prise en compte de la surface dans la quantification)
- 4 ['and', 'or'] = si 2 et 3 renseignées, on considère une parcelle comme ayant évolué, soit si le taux ET la surface évoluent ('and'), soit si le taux OU la surface évoluent ('or')

Ex. **11000:10:50:and 12000:10:50:and 21000:10:50:and 22000:10:50:and 23000:10:50:and** (valeur par défaut) =

- détection de l'évolution de la classe 11000 à 10 % et 50 m<sup>2</sup>
- détection de l'évolution de la classe 12000 à 10 % et 50 m<sup>2</sup>
- détection de l'évolution de la classe 21000 à 10 % et 50 m<sup>2</sup>
- détection de l'évolution de la classe 22000 à 10 % et 50 m<sup>2</sup>
- détection de l'évolution de la classe 23000 à 10 % et 50 m<sup>2</sup>

Ex. **11000:0:50:and** = détection de l'évolution de la classe 11000 à 50 m<sup>2</sup> (uniquement la surface)

Ex. **23000:10:0:and** = détection de l'évolution de la classe 23000 à 10 % (uniquement le taux)

Ex. **21000:10:50:or** = détection de l'évolution de la classe 21000 à 10 % ou 50 m<sup>2</sup>

## II.2. Lancement

Exemples d'utilisation pour un suivi du bâti (11000) et de la végétation (23000), à plus de 10 % et 50 m<sup>2</sup> d'évolution.

Via le séquenceur :

Il s'agit de la tâche 30\_RSQ dans le séquenceur.

C'est un script autonome, qui n'a normalement besoin d'aucune dépendance avec d'autres applications.

Attention cependant aux autres scripts utilisant PostGIS, avec risque de conflit d'accès, si lancés simultanément.

```
<Task30_RSQ_EvolutionOverTime_List>
  <Task30_RSQ_EvolutionOverTime>
    <InputPlotVector>/mnt/RAM_disk/EvolutionOverTime/parcellaire.shp</InputPlotVector>
    <OutputPlotVector>/mnt/RAM_disk/EvolutionOverTime/EvolutionOverTime.shp</OutputPlotVector>
    <FootprintVector>/mnt/RAM_disk/EvolutionOverTime/zone_etude.shp</FootprintVector>
    <InputT0File>/mnt/RAM_disk/EvolutionOverTime/OCS_sat_t0.tif</InputT0File>
    <InputTxFilesList>
      <InputTxFile>/mnt/RAM_disk/EvolutionOverTime/OCS_sat_t1.tif</InputTxFile>
      <InputTxFile>/mnt/RAM_disk/EvolutionOverTime/OCS_sat_t2.tif</InputTxFile>
    </InputTxFilesList>
    <EvolutionList>
      <Evolution>11000:10:50:and</Evolution>
      <Evolution>23000:10:50:and</Evolution>
    </EvolutionList>
  </Task30_RSQ_EvolutionOverTime>
</Task30_RSQ_EvolutionOverTime_List>
```

Pensez aussi à renseigner le paragraphe 'classification' dans les paramètres généraux (lien nom/numéro de classe).

En ligne de commande :

Il se lance comme tout autre script Python :

```
python3 -m EvolutionOverTime
-in /mnt/RAM_disk/EvolutionOverTime/parcellaire.shp
-out /mnt/RAM_disk/EvolutionOverTime/EvolutionOverTime.shp
-emp /mnt/RAM_disk/EvolutionOverTime/zone_etude.shp
-it0 /mnt/RAM_disk/EvolutionOverTime/OCS_sat_t0.tif
-itx1 /mnt/RAM_disk/EvolutionOverTime/OCS_sat_t1.tif /mnt/RAM_disk/EvolutionOverTime/OCS_sat_t2.tif
-evol 11000:10:50:and 23000:10:50:and
-cld 11000:Bati 12000:Route 21000:SolNu 22000:Eau 23000:Vegetation
```

Dans un souci de compréhension, seuls les paramètres propres au script sont renseignés.

Appel en tant que fonction :

S'agissant d'un indicateur indépendant, il n'a pas vocation à se faire appeler comme fonction dans un autre script Python. L'appel peut néanmoins se faire comme toute autre fonction :

```
from EvolutionOverTime import soilOccupationChange

soilOccupationChange('/mnt/RAM_disk/EvolutionOverTime/parcellaire.shp',
  '/mnt/RAM_disk/EvolutionOverTime/EvolutionOverTime.shp',
  '/mnt/RAM_disk/EvolutionOverTime/zone_etude.shp',
  '/mnt/RAM_disk/EvolutionOverTime/OCS_sat_t0.tif',
  ['/mnt/RAM_disk/EvolutionOverTime/OCS_sat_t1.tif', '/mnt/RAM_disk/EvolutionOverTime/OCS_sat_t2.tif'],
  evolutions_list=['11000:10:50:and', '23000:10:50:and'],
  class_label_dico={11000:'Bati', 12000:'Route', 21000:'SolNu', 22000:'Eau', 23000:'Vegetation'})
```

Dans un souci de compréhension, seuls les paramètres propres à la fonction sont renseignés.

### III. Exemples de résultats