

Chaîne « Risque » – AreasUnderUrbanization

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 AreasUnderConstruction.....	3
Illustration 2: Dépendances du script AreasUnderUrbanization avec la chaîne interne.....	4

I. Présentation du script

I.1. Description

Note : Ce script peut être utilisé pour produire la donnée Oc0 du Référentiel National de Vulnérabilité aux Inondations (= « Zones en voies d'urbanisation »)¹.

Le but premier est de **discerner les parcelles construites de celles disponibles**. Pour cela, un seuil de détectabilité du bâti est paramétré, qui peut dépendre de la surface de la parcelle elle-même.

L'information de bâti peut provenir soit d'une liste de vecteurs (exemple BD Topo avec bâti indifférencié + bâti industriel + bâti remarquable), soit d'un raster sous forme de masque binaire (0 = pas de bâti ; 1 = bâti).

Si un PLU et/ou un PPRi est renseigné dans les fichiers d'entrée, une seconde étape est alors lancée, uniquement si la parcelle a été détectée disponible à l'étape précédente, pour **déterminer la constructibilité de la parcelle**. Le résultat donnera ici trois possibilités :

- **parcelle construite** : présence de bâti
- **parcelle disponible** : absence de bâti
 - **parcelle disponible, mais non-constructible** : en zonage A/N du PLU et/ou en zonage PPRi
 - **parcelle disponible, et constructible** : en zonage U/AU du PLU et hors zonage PPRi

Le champ contenant l'information de zonage, et les valeurs associées à ce zonage ('U' et 'AU' pour le PLU, 'rouge' et 'bleu' pour le PPRi) sont paramétrables.

Dans le cas où la constructibilité entre en jeu, et pour mieux exploiter le fichier de sortie, il sera précisé, dans le champ final qui contient le résultat de l'algorithme, la raison de la (non-)constructibilité de la parcelle.

Par exemple : constructible car en zone U du PLU et hors-zonage PPRi, non-constructible car en zone N du PLU.

¹ https://www.ecologique-solidaire.gouv.fr/sites/default/files/20160923_Guide_GT_Referentiel_vulnerabilite.pdf

I.2. Traitements

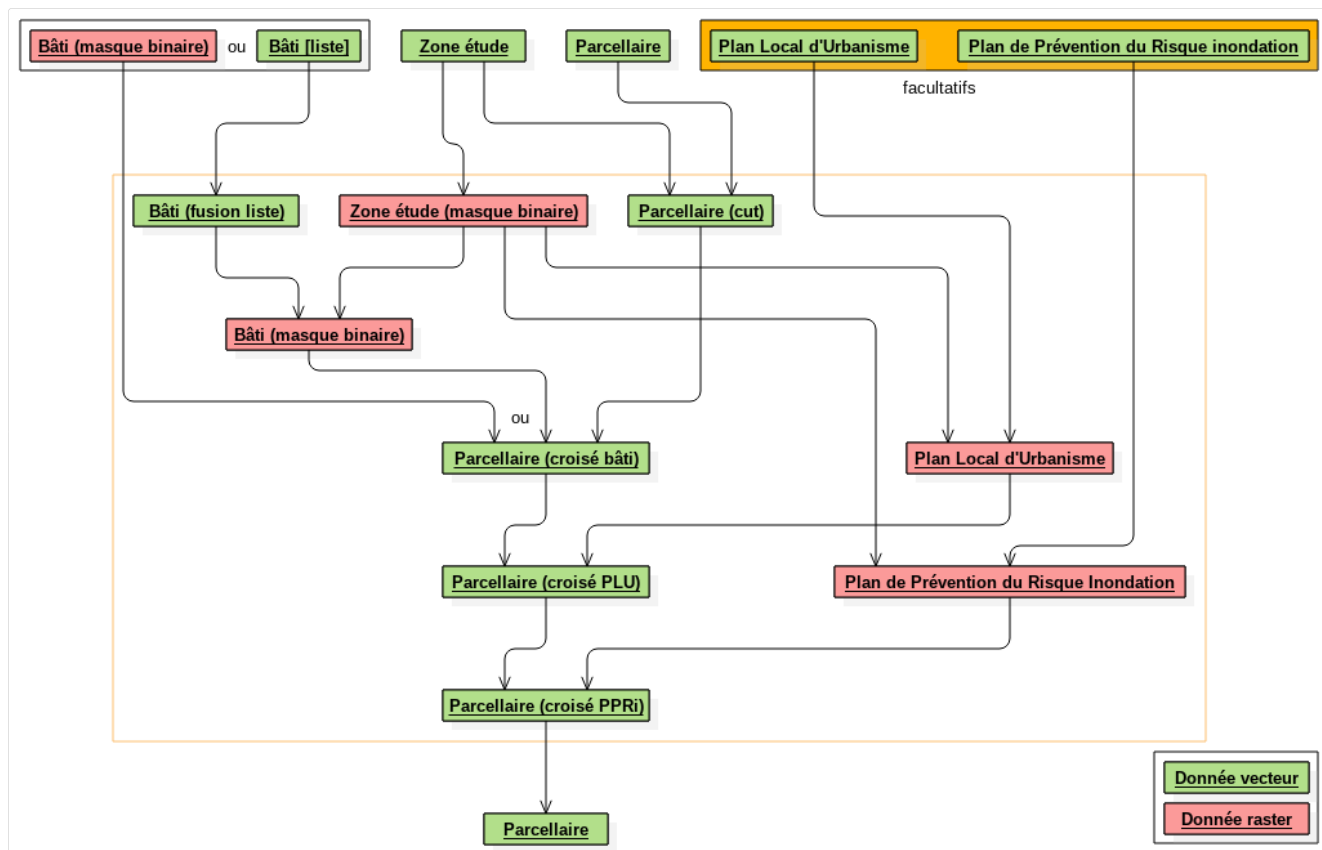


Illustration 1: Flux de données du script AreasUnderConstruction.
Les fichiers temporaires sont dans le cadre orange.

Les traitements pour ce script sont différents suivant les données d'entrée. La seule étape commune à tous les cas de figure est le découpage du parcellaire à l'emprise de la zone d'étude.

- Si le bâti en entrée est une liste de fichiers vecteurs, alors un masque binaire est généré pour le croiser avec le parcellaire découpé. Avant cela, les fichiers vecteurs sont fusionnés pour n'en obtenir qu'un seul, qui sera rastérisé sur la base d'un autre masque binaire, obtenu via une rastérisation du fichier d'emprise.
- Si le bâti est directement rentré en masque binaire, il sera immédiatement croisé avec le parcellaire découpé.
- Si un fichier PLU est donné en entrée, il sera rastérisé suivant le masque binaire du bâti, puis également croisé avec le parcellaire découpé.
- De même, si un fichier PPRI est donné en entrée, il sera rastérisé et croisé avec le parcellaire.

Enfin, des requêtes SQL sont exécutées pour déterminer la disponibilité des parcelles (présence/absence de bâti), et le cas échéant, la constructibilité (suivant les zonages PLU et PPRI).

I.3. Dépendances

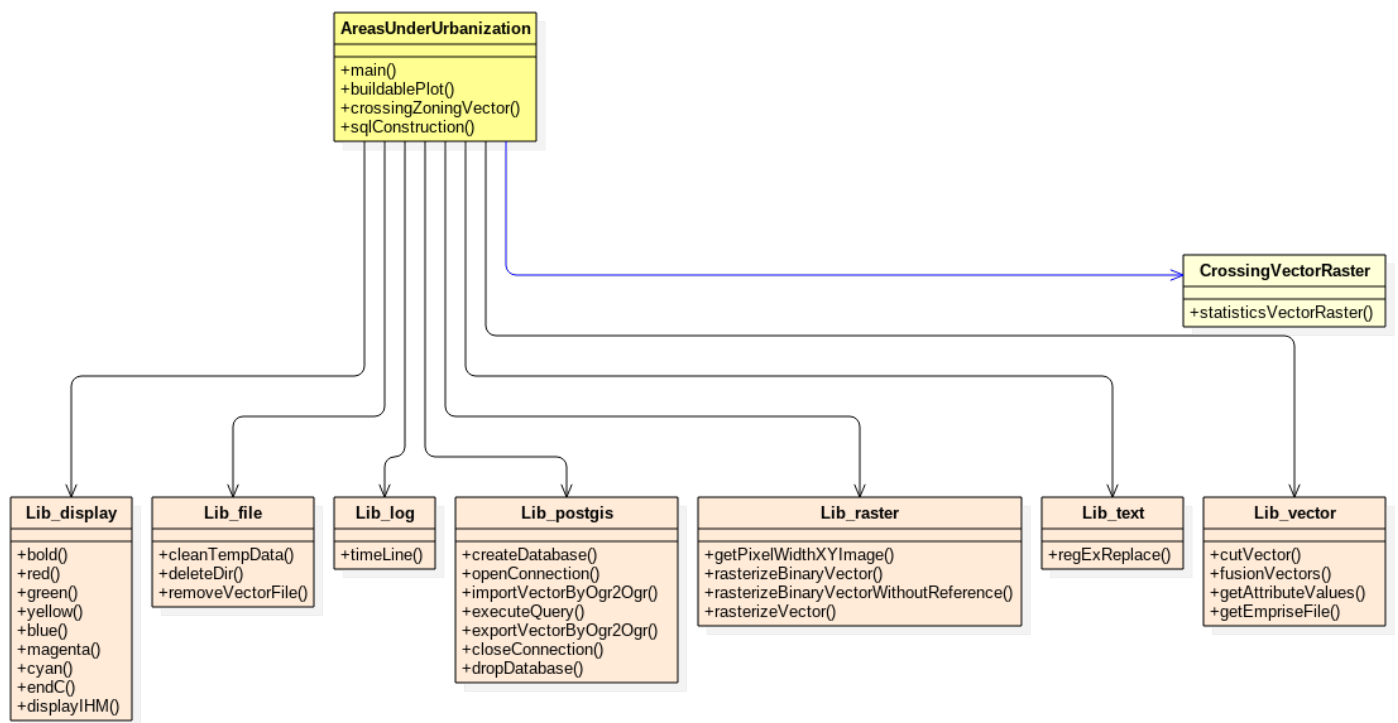


Illustration 2: Dépendances du script AreasUnderUrbanization avec la chaîne interne.

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

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_built_file (-ibr)** : masque binaire du bâti (raster entrée)
- **input_built_vector_list (-ibvl)** : fichiers du bâti (liste vecteurs entrée)
- **input_plu_vector (-iplu)** : fichier du Plan Local d'Urbanisme (vecteur entrée)
- **input_ppr_vector (-ippr)** : fichier du Plan de Prévention du Risque inondation (vecteur entrée)
- **min_built_size_list (-mbsl)** : liste des paramètres de surface minimale de bâti, fonction de la surface de la parcelle, pour la considérer construite. Par défaut : 'None:100:20 100:None:40'
- **plu_field (-pluf)** : champ du PLU contenant le zonage. Par défaut : 'TYPEZONE'
- **plu_u_values (-pluu)** : liste des valeurs pour le zonage PLU « U ». Par défaut : 'U'
- **plu_au_values (-plua)** : liste des valeurs pour le zonage PLU « AU ». Par défaut : 'AU AUc AUc'
- **ppr_field (-pprf)** : champ du PPRi contenant le zonage. Par défaut : 'CODEZONE'
- **ppr_r_values (-pprr)** : liste des valeurs pour le zonage PPRi « rouge ». Par défaut : 'R1 R2 R3'
- **ppr_b_values (-pprb)** : liste des valeurs pour le zonage PPRi « bleu ». Par défaut : 'B1 B2 B2-1 B2-2 B3'

Seuls les paramètres en gras sont obligatoires.

Au moins un des deux paramètres du bâti d'entrée doit être renseigné.

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 'min_built_size_list' :

1:2:3

- 1 = taille minimale de la parcelle (si None, pas de surface minimale)
- 2 = taille maximale de la parcelle (si None, pas de surface maximale)
- 3 = seuil de détectabilité du bâti

Ex. **None:100:20 100:None:40** (valeur par défaut) =

- détection du bâti à partir de 20 m² pour les parcelles inférieures à 100 m²
- détection du bâti à partir de 40 m² pour les parcelles supérieures à 100 m²

Ex. **None:None:50** = détection du bâti à partir de 50 m², peut importe la surface de la parcelle

Ex. **100:1000:30** = détection du bâti à partir de 30 m², uniquement pour les parcelles entre 100 et 1 000 m²

II.2. Lancement

Exemples d'utilisation avec seulement le masque binaire du bâti en entrée, où l'on cherche le bâti de plus de 20 m² pour les parcelles de moins de 100 m², et le bâti de plus de 40 m² pour celles de plus de 100 m².

Via le séquenceur :

Il s'agit de la tâche 20_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.

```
<Task20_RSQ_AreasUnderUrbanization_List>
  <Task20_RSQ_AreasUnderUrbanization>
    <InputPlotVector>/mnt/RAM_disk/AreasUnderUrbanization/parcellaire.shp</InputPlotVector>
    <OutputPlotVector>/mnt/RAM_disk/AreasUnderUrbanization/AreasUnderUrbanization.shp</OutputPlotVector>
    <FootprintVector>/mnt/RAM_disk/AreasUnderUrbanization/zone_etude.shp</FootprintVector>
    <InputBuiltFile>/mnt/RAM_disk/AreasUnderUrbanization/OCS_sat_masque_bati.tif</InputBuiltFile>
    <InputBuiltVectorsList>
    </InputBuiltVectorsList>
    <InputPluVector></InputPluVector>
    <InputPprVector></InputPprVector>
    <MinBuiltSizesList>
      <MinBuiltSize>None:100:20</MinBuiltSize>
      <MinBuiltSize>100:None:40</MinBuiltSize>
    </MinBuiltSizesList>
    [...]
  </Task20_RSQ_AreasUnderUrbanization>
</Task20_RSQ_AreasUnderUrbanization_List>
```

[...] : les paramètres de zonage PLU/PPRi ne sont pas illustrés ici

En ligne de commande :

Il se lance comme tout autre script Python :

```
python3 -m AreasUnderUrbanization
-in /mnt/RAM_disk/AreasUnderUrbanization/parcellaire.shp
-out /mnt/RAM_disk/AreasUnderUrbanization/AreasUnderUrbanization.shp
-emp /mnt/RAM_disk/AreasUnderUrbanization/zone_etude.shp
-ibr /mnt/RAM_disk/AreasUnderUrbanization/OCS_sat_masque_bati.tif
-mbsl None:100:20 100:None:40
```

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 AreasUnderUrbanization import buildablePlot

buildablePlot('/mnt/RAM_disk/AreasUnderUrbanization/parcellaire.shp',
              '/mnt/RAM_disk/AreasUnderUrbanization/AreasUnderUrbanization.shp',
              '/mnt/RAM_disk/AreasUnderUrbanization/zone_etude.shp',
              '/mnt/RAM_disk/AreasUnderUrbanization/OCS_sat_masque_bati.tif',
              [], ',', ' ', min_built_size_list=['None:100:20', '100:None:40'])
```

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

III. Exemples de résultats