

# Annexe D : Utilisation de l’outil SARSAR sur Terrascope

## 1. Besoin logiciel

Pour accéder et utiliser l’outil “SARSAR”, l’utilisateur a besoin :

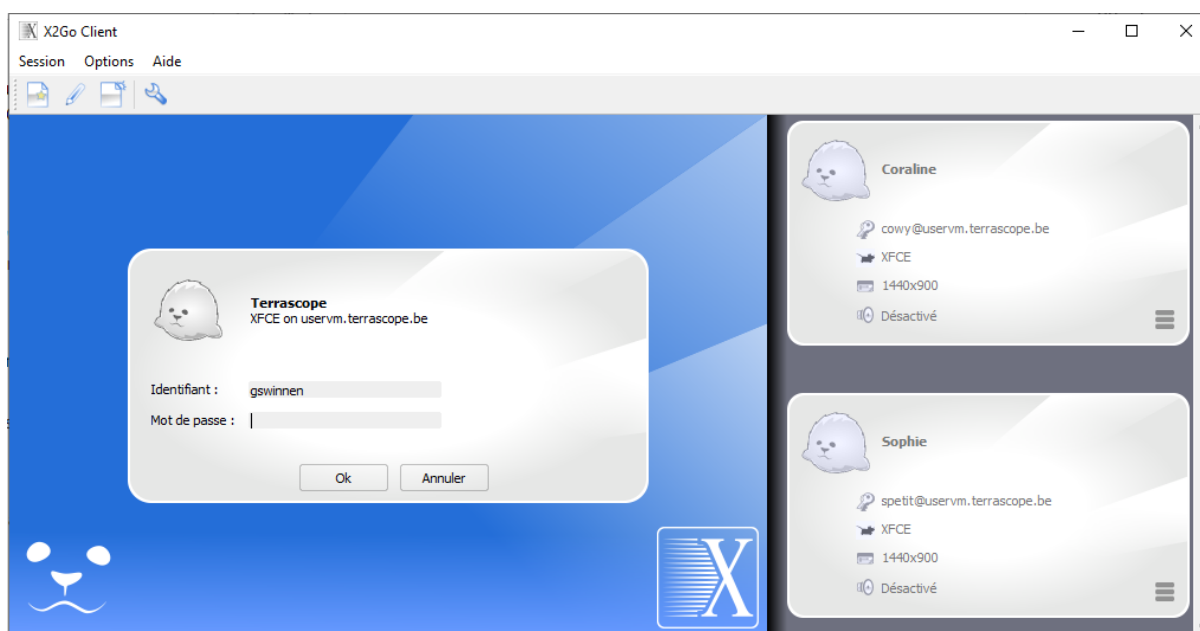
- D’un compte Terrascope : <https://terrascope.be/fr> ;
- D’une machine virtuelle Terrascope : <https://terrascope.be/nl/form/vm> ;
- Du logiciel X2Go Client pour accéder à sa machine virtuelle : <https://terrascope.be/fr/how-access-your-terrascope-vm> ;
- D’installer sur la machine virtuelle le certificat d’authentification ssh permettant la connexion à la base de données : contacter [g.swinnen@issep.be](mailto:g.swinnen@issep.be) ;
- D’installer le package SARSAR sur sa machine virtuelle : contacter [g.swinnen@issep.be](mailto:g.swinnen@issep.be).

## 2. Accéder à l’outil

Une fois ces éléments installés et configurés, l’outil SARSAR est exploitable. Ci-dessous, une description d’une procédure de prise en main de l’outil.

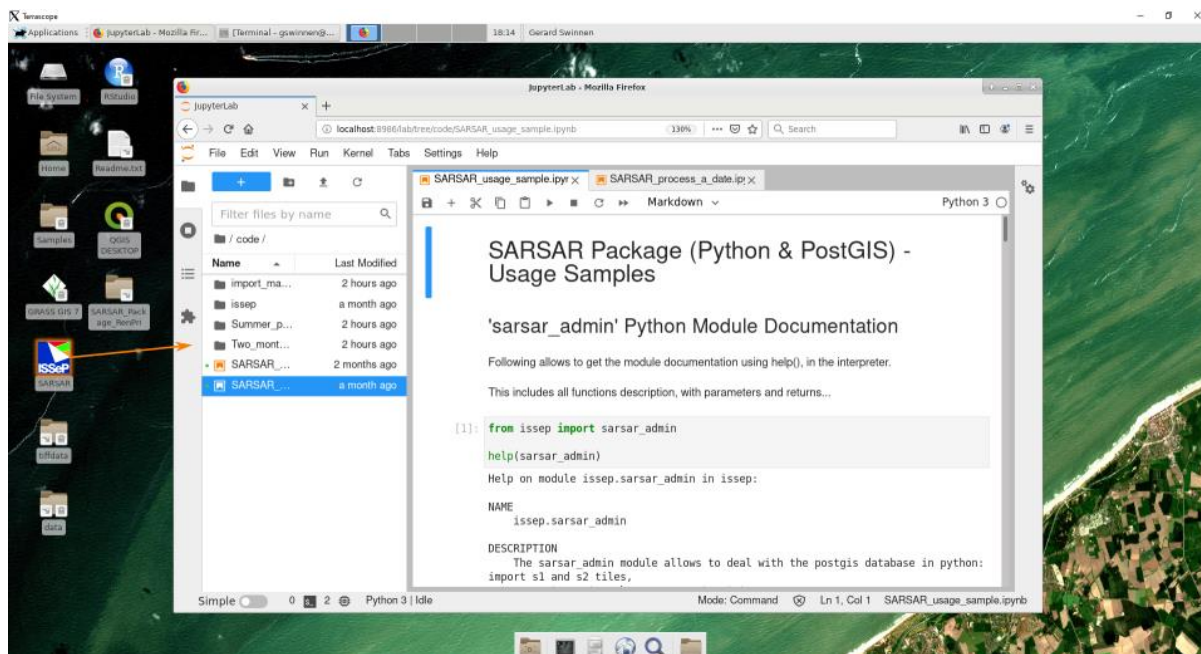
### a. Connexion à sa machine virtuelle avec X2GO

Ouvrir X2GO et se connecter à sa VM.



## b. Accès aux scripts SARSAR

Une fois connecté à sa VM, l'icône "SARSAR" présente sur le bureau permet d'activer l'environnement virtuel et d'accéder à Jupyter Lab. Là, les différents codes développés pour le projet SARSAR sont disponibles et peuvent être utilisés. Ces codes se présentent comme des Jupyter Notebooks et sont stockés dans des sous-répertoires (volet latéral gauche de Jupyter Lab) en fonction de leur utilisation.



Le sous-répertoire "import\_massif\_des\_données" contient des scripts pour importer les données Sentinel-1 et Sentinel-2 par mois et par an.

Le sous-répertoire "Summer\_process" contient des scripts pour réaliser la détection et qualification de changement pour le pas de temps annuel, à la fin de chaque été.

Le sous-répertoire "Two\_months\_process" contient des scripts pour réaliser la détection et qualification de changement pour le pas de temps bimestriel.

Le sous-répertoire "issep" contient les fonctions de base utilisées par tous les scripts SARSAR. Il ne faut pas y toucher. Une description de ces fonctions est disponible au **point 3**.

Sont également disponibles, deux Jupyter Notebooks contenant des exemples d'utilisation des fonctions de base SARSAR décrites au **point 3** : "SARSAR\_usage\_sample" et "SARSAR\_process\_a\_date". Il est vivement conseillé de parcourir ces deux Jupyter Notebooks et de tester les différentes parties pour se familiariser avec les fonctions SARSAR.

### 3. Fonctions de base SARSAR

#### a. Fonction « `crop_sar_raw_rasters` », pour clipper/cropper les images importées en DB sur la Wallonie

Cette fonction permet de supprimer dans une table temporaire donnée (du type « `S2_YYYYMMDD_bXX` » ou « `S1_YYYYMMDD_vXX` ») toutes les tuiles d'un km<sup>2</sup> n'étant pas en intersection avec le territoire wallon, sur lequel un buffer positif de 500m a été appliqué pour des raisons de prudence.

Les appels à cette fonction sont très simples et intuitifs :

```
-- Params :  
-- =====  
--   import_date (format yyyyymmdd)  
--   import_type ('s1' or 's2')  
--   import_band ('bXX' format, or 'vvX'/'vhX')  
  
select crop_sar_raw_rasters('20200418', 's2', 'b02');  
  
select crop_sar_raw_rasters('20210217', 's1', 'vh37');
```

#### b. Fonction « `compute_sar_raw_rasters` », pour générer les mini-rasters par SAR sur les images brutes (à partir d'une table temporaire)

Cette fonction permet de générer les mini-rasters pour chaque SAR couvert par les tuiles d'une table temporaire donnée, qui pour rappel représente une bande brute à une date d'acquisition précise.

```
-- Params :  
  
-- =====  
  
-- import_date (format yyyyymmdd)  
-- import_type ('s1' or 's2')  
-- import_band ('bXX' format, or 'vvX'/'vhX')  
  
select compute_sar_raw_rasters('20200418', 'S2', 'b08');
```

La fonction retourne le nombre de tuples insérés dans la table de calcul « **sar\_index\_rasters** », c'est-à-dire le nombre de mini-rasters sur les SAR qui ont pu être générés sur base de la table temporaire. Les précédents mini-rasters portant sur la même date d'acquisition et sur la même bande sont automatiquement purgés. La fonction renvoie une exception explicite si l'image brute nécessaire n'a pas encore été importée dans la DB.

### c. Fonction « **compute\_sar\_index\_stats** », pour générer les statistiques par SAR sur un indice et une date d'acquisition spécifiques

Cette fonction va dans un premier temps (et grâce à l'appel d'une fonction intermédiaire « **compute\_sar\_index\_rasters\_only** ») générer des mini-rasters pour l'indice et la date d'acquisition demandés, sur base des mini-rasters de bandes brutes précédemment calculés (voir fonction ci-avant)

Exemple : le système pourra générer les mini-rasters pour l'indice NDVI si les mini-rasters pour les bandes b04 et b08 ont été précédemment générés pour la date d'acquisition souhaitée.

La fonction va dans un second temps (et grâce à l'appel d'une fonction intermédiaire « **compute\_sar\_index\_stats\_only** ») calculer les statistiques, SAR par SAR, sur base de l'ensemble des pixels constituant chaque mini-raster.

```
-- Params :  
  
-- =====  
  
--   import_date (format yyyyymmdd)  
  
--   index_name ('NDVI','BI', etc)  
  
select compute_sar_index_stats('20200418', 'NDVI');
```

La fonction retourne le nombre de tuples insérés dans la table de calcul « **sar\_index\_stats** ». Les précédentes statistiques portant sur la même date d'acquisition et sur le même indice sont automatiquement purgés. La fonction renvoie une exception explicite si l'indice spécifié n'est pas défini dans la table « **indexes\_def** ».

## 4. Accès au serveur postgresSQL et à la base de données

L'accès au serveur et la navigation sur celui-ci requiert la connaissance de ligne de commande linux. Dans sa VM, ouvrir un terminal et taper la commande suivante permettant de se connecter en ssh au serveur postgresSQL :

```
ssh issep@postgres-issep.vgt.vito.be
```

Taper le mot de passe

Vous arrivez sur le serveur. Sur le serveur tout se fait en ligne de commande linux.

Pour accéder à la base de données stockées sur ce serveur, tapez :

```
psql -U issep -d sarsar
```

Pour sortir de la db, tapez :

```
exit
```

Pour quitter le serveur, tapez :

```
exit
```

Exemple :

```
[cowy@cowy ~]$ ssh issep@postgres-issep.vgt.vito.be  
issep@postgres-issep.vgt.vito.be's password:  
Last login: Wed Sep 29 11:07:23 2021 from 192.168.116.244  
  
[issep@postgres-issep ~]$ psql -U issep -d sarsar  
psql (11.13)
```

```
Type "help" for help.
```

```
sarsar=>
```

```
sarsar=> exit
```

```
[issep@postgres-issep ~]$ exit
```

```
logout
```

```
Connection to postgres-issep.vgt.vito.be closed.
```

## 5. Description des tables de la base de données

Le package possède quelques tables réparties en groupes :

- Des tables de définitions : pour définir les SAR, les indices
- Des tables de références : représentation du territoire wallon
- Des tables de calcul : pour stocker les étapes/résultats intermédiaires ou finaux lors des différents traitements
- Des tables temporaires : pour le stockage temporaire des rasters bruts importés en DB

Chaque table est décrite dans les sections suivantes.

### a. La table de définition « sar\_def » dédiée à la définition des SAR

La table « sar\_def » contient la définition des 2300 SAR à étudier.

id	nom	shape_leng	codecarto	id_sar	id_segment	info	descriptio	shape_area	geom
	character varying (254)	numeric	character varying (32)	character varying (30)	[PK] character varying (254)	character varying (254)	character varying (254)	numeric	geometry
1	Hanger La Comète	317.756448450	25005-ISA-0001-02	25005_X_001	25005-ISA-0001-02	[null]	[null]	4851.14853187	0106000020777...
2	Entreprise de voine Van ...	166.131982138	25005-ISA-0003-01	25005_L_004	25005-ISA-0003-01	[null]	[null]	1295.02510569	0106000020777...
3	Entrepôts militaires à Lil...	1012.51887147	25014-ISA-0001-01	25014_L_001	25014-ISA-0001-01	[null]	[null]	53418.4713983	0106000020777...
4	Établissement Sciens Rog...	242.592454881	25014-ISA-0002-01	25014_L_003	25014-ISA-0002-01	[null]	[null]	1427.32132780	0106000020777...
5	Établissement Defalque ...	332.801627649	25014-ISA-0003-01	25014_L_004	25014-ISA-0003-01	[null]	[null]	2407.06169657	0106000020777...
6	Filature de coton Allard	2178.20942345	25014-ISA-0005-01	25014_L_006	25014-ISA-0005-01	[null]	[null]	36698.7887860	0106000020777...
7	Établissements Denolin n...	578.505140159	25014-ISA-0007-01	25014_L_008	25014-ISA-0007-01	[null]	[null]	2117.42796422	0106000020777...
8	Tannerie Dobeck	92.6052102647	25014-ISA-0008-01	25014_L_010	25014-ISA-0008-01	[null]	[null]	266.706991690	0106000020777...

Le champ « id\_segment » est défini comme clé primaire, et permet donc d'identifier univoquement chacun des SAR. Cette clé est utilisée comme clé étrangère dans les autres tables pour référencer un SAR.

Le champ « geom » contient l'emprise spatiale du SAR, exprimée sous la forme d'un MULTIPOLYGON dans le SRID EPSG:32631. Ce champ est indexé spatialement.

Cette table est initialement constituée à partir d'un fichier Shapefile importé dans PostgreSQL/PostGIS via les commandes « shp2pgsql » et « psql », durant la phase de déploiement du package. La commande « shp2pgsql » produit le script SQL permettant de créer la table et d'ajouter

les données provenant du shapefile, sans toutefois l'exécuter. L'exécution du script est alors réalisée via la commande « psql ».

La taille de cette table avec son index spatial ne représente en DB que quelques Mo

## b. La table de définition « indexes\_def » dédiée à la définition des indices calculables

La table « indexes\_def » contient la définition des indices calculables sur les rasters importés dans le Package.

index_name	rast_a_name	rast_b_name	index_formula	output_type
[PK] character varying (12)	character varying (12)	character varying (12)	character varying (100)	character varying (12)
1	BVI	b02	$\frac{(\text{rast1}) - (\text{rast2})}{(\text{rast1}) + (\text{rast2})}$ :float	32BF
2	BI	b04	$\frac{\sqrt{((\text{rast1})^2 + (\text{rast3})^2) + (\text{rast2})^2}}{2}$ :float	32BF
3	BI2	BI2_part1	$\frac{\sqrt{((\text{rast1}) - (\text{rast2}))^2 + (\text{rast2})^2}}{2}$ :float	32BF
4	BI2_part1	b04	$\frac{(\text{rast1})^2 + (\text{rast2})^2 + (\text{rast2})^2}{2}$ :float	32BF
5	CI	b04	$\frac{(\text{rast1}) + (\text{rast2})}{(\text{rast1}) + (\text{rast2})}$ :float	32BF
6	NDVI	b11	$\frac{(\text{rast1}) - (\text{rast2})}{(\text{rast1}) + (\text{rast2})}$ :float	32BF
7	NDVI	b08	$\frac{(\text{rast1}) + (\text{rast2})}{(\text{rast1}) + (\text{rast2})}$ :float	32BF
8	NDWI	b08	$\frac{(\text{rast1}) - (\text{rast2})}{(\text{rast1}) + (\text{rast2})}$ :float	32BF
9	NDWI2	b03	$\frac{(\text{rast1}) - (\text{rast2})}{(\text{rast1}) + (\text{rast2})}$ :float	32BF
10	SAVI	b04	$1.5 * \frac{(\text{rast1}) - (\text{rast2})}{(\text{rast1}) + (\text{rast2}) + 0.5}$ :float	32BF
11	SBI	b04	$\frac{\sqrt{((\text{rast1})^2 + (\text{rast3})^2) + (\text{rast2})^2}}{2}$ :float	32BF
12	VH	vh	$\frac{(\text{rast1})}{(\text{rast1})}$ :float	32BF

Dans le package, un indice est calculé sur base d'une formule basée sur 2 inputs au maximum, qui peuvent être des « bandes brutes » (telles que « b02 », « b03 », etc pour du Sentinel-2 ou « vv », « vh » pour du Sentinel-1) ou des indices eux-mêmes.

Il est donc possible de construire des indices complexes, nécessitant plus de 2 inputs (par exemple 4 bandes brutes) en combinant successivement ces bandes dans des indices intermédiaires. On définit en quelque sorte un indice sur base d'autres indices. Exemple : l'indice « BI2 » est défini par une formule appliquée sur les bandes brutes « b03 », « b04 » et « b08 ». Dans notre package, il convient dès lors de calculer d'abord un indice intermédiaire « BI2\_part1 » sur base des bandes brutes « b03 » et « b04 », et ensuite de calculer l'indice final « BI2 » sur base de l'indice intermédiaire « BI2\_part1 » et de la bande brute « b08 ».

Chaque tuple de cette table est donc composé :

- D'un nom d'indice : « index\_name »
- Du nom du premier input : « rast\_a\_name »
- Du nom du second input : « rast\_b\_name »
- De la formule pour calculer l'indice sur base des 2 inputs : « index\_formula »
- Du format de sortie pour le résultat calculé (par défaut en 32bits virgule flottante) : « output\_type »

Les indices peuvent être adaptés ou ajoutés dans le Package à tout moment au gré de l'opérateur.

### c. La table de référence « wallonia\_buff\_500m », pour le clipping des images sur la Wallonie

La table « wallonia\_buff\_500m » ne contient qu'un seul tuple, qui est l'emprise spatiale de la Wallonie avec un buffer positif de 500m, exprimée en SRID EPSG:32631.



Cette table permet de clipper/cropper les images brutes importées dans le Package dès le début des traitements (tuiles MGRS pour Sentinel-2, tuiles encore plus grandes pour Sentinel-1), afin de minimiser au plus tôt l'espace de stockage nécessaire au système et d'accélérer les traitements ultérieurs.

### d. La table de calcul « sar\_index\_rasters », pour le stockage des clip/crop de rasters sur chaque SAR, par date d'acquisition

La table « sar\_index\_rasters » est une table initialement vide destinée à stocker les clip/crop de rasters sur chaque SAR par date, également appelés « mini-rasters ».

	sar_id_segment character varying (254)	acq_date date	comp_time timestamp without time zone	index_name character vary	rast raster
1	57093-ISA-0002-01	2020-04-18	2021-03-10 11:08:26.839583	b04	01000001000000000000024400000...
2	57093-ISA-0012-01	2020-04-18	2021-03-10 11:08:26.839583	b04	01000001000000000000024400000...
3	57093-ISA-0005-01	2020-04-18	2021-03-10 11:08:26.839583	b04	01000001000000000000024400000...
4	57072-ISA-0001-01	2020-04-18	2021-03-10 11:08:26.839583	b04	01000001000000000000024400000...
5	57093-ISA-0012-01	2020-04-18	2021-03-10 11:08:32.672829	b08	01000001000000000000024400000...
6	57093-ISA-0005-01	2020-04-18	2021-03-10 11:08:32.672829	b08	01000001000000000000024400000...
7	57072-ISA-0001-01	2020-04-18	2021-03-10 11:08:32.672829	b08	01000001000000000000024400000...
8	57093-ISA-0002-01	2020-04-18	2021-03-10 11:08:32.672829	b08	01000001000000000000024400000...

Les clip/crop de raster proviennent soit des bandes brutes (« b02 », « b03 », « vh37 », etc) soit d'indices calculés (« NDVI », « BI », etc) et sont générés lors des appels aux différentes fonctions de Package que nous verrons plus loin dans cette documentation.

Chaque tuple de cette table représente donc un « mini-raster » d'une seule bande découpé exactement selon l'emprise du SAR référencé, pour une date indiquée. Ce mini-raster peut représenter un bande brute spécifique ou un indice calculé.

Dans le premier exemple ci-dessus, nous voyons un ensemble de tuples représentant les clip/crop sur 8 SAR des rasters « b04 » et « b08 », représentant des données brutes Sentinel-2 acquise à la date du 18/04/2020 (champ « acq\_date »), et calculés-processés le 10/03/2021 à 11h08.



	sar_id_segment character varying (254)	acq_date date	comp_time timestamp without time zone	index_name character vary	rast raster
1	57093-ISA-0012-01	2020-04-18	2021-03-10 11:09:40.818956	NDVI	010000010000000000000024400000...
2	57093-ISA-0005-01	2020-04-18	2021-03-10 11:09:40.818956	NDVI	010000010000000000000024400000...
3	57072-ISA-0001-01	2020-04-18	2021-03-10 11:09:40.818956	NDVI	010000010000000000000024400000...
4	57093-ISA-0002-01	2020-04-18	2021-03-10 11:09:40.818956	NDVI	010000010000000000000024400000...

Dans le second exemple ci-dessus, nous voyons un ensemble de tuples représentant les clip/crop sur 4 SAR du raster « NDVI », représentant des données sur l'indice calculé NDVI pour la date d'acquisition du 18/04/2020 (champ « acq\_date »), et calculés-processés le 10/03/2021 à 11h09.

A noter que les rasters de cette table (champ « rast ») sont également indexés spatialement et exprimés dans le SRID EPSG:32631.

### e. La table de calcul « sar\_index\_stats », pour le stockage des statistiques par SAR sur les indices calculés, par date d'acquisition

La table « sar\_index\_stats » est destinée à stocker les statistiques par SAR et par date d'acquisition sur les indices calculés (non sur les bandes brutes), et ce à partir d'une analyse statistique des pixels composants les mini-rasters de la table « sar\_index\_rasters ».

	sar_id_segment character varying (254)	acq_date date	comp_time timestamp without time zone	index_name character vary	pixel_count bigint	index_mean numeric	index_stddev numeric	index_min numeric	index_max numeric
1	57072-ISA-0001-01	2020-04-18	2021-03-10 11:17:49.250632	NDVI	7	0.197	0.026	0.171	0.254
2	57093-ISA-0002-01	2020-04-18	2021-03-10 11:17:49.250632	NDVI	14	0.171	0.014	0.143	0.187
3	52074-ISA-0018-01	2021-02-17	2021-02-19 17:19:27.019391	VH	4	0.040	0.028	0.018	0.086
4	57081-ISA-0053-01	2021-02-17	2021-02-19 17:19:27.019391	VH	6	0.260	0.109	0.078	0.383
5	63079-ISA-0050-01	2021-02-17	2021-02-19 17:19:27.019391	VH	5	0.172	0.036	0.124	0.231
6	62121-ISA-0003-01	2021-02-17	2021-02-19 17:19:27.019391	VH	8	0.057	0.028	0.026	0.113

Nous retrouvons de nouveau la référence vers le SAR (via une clé étrangère), la date d'acquisition de la donnée et la date du calcul des statistiques d'indices.

En plus de cela nous avons, pour chacun des indices étudiés (NDVI et VH ci-dessus), 5 valeurs :

- Le nombre de pixels 10mx10m constituant le mini-raster sur le SAR
- La moyenne arithmétique calculée pour l'indice sur ces pixels
- L'écart-type calculé pour l'indice sur ces pixels
- La valeur minimale relevée pour l'indice sur ces pixels
- La valeur maximale relevée pour l'indice sur ces pixels

f. Les tables temporaires « S2\_YYYYMMDD\_bXX » et « S1\_YYYYMMDD\_vX » pour l'import des données brutes Sentinel 2 et Sentinel 1

Ces tables temporaires sont créées lors de l'import des fichiers GeoTIFF Sentinel 1 et Sentinel 2 dans la DB PostgreSQL/PostGIS. Cet import est réalisé via l'outil « raster2pgsql » fourni avec PostGIS.

Les tables temporaires n'ont plus d'utilité une fois qu'elles ont été traitées pour alimenter en « mini-rasters par SAR » la table de calcul « sar\_index\_rasters » décrite ci-dessus. Du moins si l'ensemble des SAR à analyser ne doit pas être complété à l'avenir.