

Licence Professionnelle Universitaire
Systèmes d'Informations Géographique

Localisation, caractérisation et hiérarchisation des têtes de bassin versant sur le territoire de la Sèvre Nantaise

*L'open-source au service
des
chaînes et automatisations de traitements SIG*

Anthony Coudart

Rapport de stage du 16 avril au 14 septembre 2018

EPTB Sèvre nantaise

Moulin de Nid d'Oie
10 bis route de Nid d'Oie - CS 49405
44194 CLISSON Cedex
www.sevre-nantaise.com

Enseignant référent :
Frédéric Pouget



Maître de stage :
Sébastien Renou

1 Remerciements

En premier lieu je tiens à remercier mon maître de stage Sébastien Renou et mon second maître de stage (officieux) Antoine Rivière pour leur professionnalisme, leurs nombreux conseils, leur disponibilité, leur implication, pour la place qu'ils ont su me faire dans leur antre technologique et pour la confiance qu'ils m'ont accordée en me prenant dans l'équipe. Si les choix technologiques contraignants ont été une source de souffrance (relative), ils ont aussi été les vecteurs de montée en compétences et d'une prise de conscience générale sur les immenses possibilités offertes par ces outils.

Je remercie chaleureusement toute l'équipe de l'EPTB pour l'accueil qu'ils m'ont réservé que ce soit dans et hors les murs de cette institution, pour les petits gâteaux du matin et confiseries toujours bienvenues. Merci également pour tous les échanges, professionnels ou non, pour leur intérêt et leur bienveillance à l'égard du projet. Je leur souhaite à tous de la réussite dans leurs projets divers et variés et dans l'utilisation et l'appropriation de l'outil élaboré ici !

Je remercie également tous les collègues extérieurs à l'EPTB avec qui j'ai pu avoir la chance d'échanger durant ces quelques mois, Mickaël Le Bihan, Stéphane Renou, Lise Guégniard, Jérôme Fernandez, Benjamin Magand, Alix Augier, Arnaud Mevel, Nathalie Pécheux et tous ceux que j'oublie sans le vouloir.

Je remercie tout particulièrement deux professeurs Frédéric Pouget et Alain Layec pour le cœur qu'ils mettent à l'ouvrage, pour leur confiance et avoir su trouver les mots quand le doute m'assaillait. Longue vie à la LUPSIG !

Comment ne pas remercier cette promotion Cassini ? Sans qui cette année n'aurait pas été la même, sans qui les choses auraient certainement été encore plus dures. Merci pour leur aide, pour les échanges, les partages, les folles journées et nuits de labeurs masochistes, bon vent à tous.

Enfin mention spéciale pour celle qui m'a soutenu et accompagné dans cette idée un peu folle de retourner à l'école et qui malgré les difficultés m'a toujours réservé le plus beau des sourires. Merci pour son soutien et pour son aide de relectrice patiente et avisée, maintenant place à la suite...

Table des matières

1 Remerciements	2
Glossaire des abréviations	5
2 Introduction	6
3 Présentation de la structure d'accueil et du territoire	8
4 Définition de la tête de bassin versant	10
5 Fonctionnalités des têtes de bassins versant	12
5.1 Fonctions hydrauliques et hydromorphologiques :.....	12
5.2 Fonctions biologiques et trophiques :.....	13
5.3 Fonctions Physico-chimiques :.....	14
6 Contexte réglementaire	15
6.1 Ce que dit le SDAGE.....	15
6.2 Ce que dit le SAGE.....	15
6.3 Les leviers réglementaires.....	16
6.4 Solutions et incitations réglementaires expérimentées sur le bassin Loire-Bretagne.....	16
7 Contexte de l'étude	18
7.1 Échanges avec les bassins voisins.....	18
7.2 Choix des technologies.....	19
7.2.1 GRASS GIS, tout en puissance.....	19
7.2.1.1 Démarrage de GRASS.....	20
7.2.1.2 Les extensions GRASS.....	21
7.2.1.3 Installation d'une extension.....	21
7.2.1.4 Les extensions "Stream".....	22
7.2.2 Postgres / Postgis, la force du SQL.....	22
7.2.3 L'interface Web, l'outil universel.....	23
7.3 Les fonctions unitaires dans Processus.....	23
7.4 Les fonctions en chaîne dans Processing chains.....	25
7.4.1 R et Rstudio, l'outil statistique à la portée de tous.....	27
7.4.2 Le Wiki, la documentation interne.....	27
8 Localisation des têtes de bassin versant	28
8.1 Référentiels hydrographiques disponibles.....	28
8.1.1 Tableau comparatif du linéaire hydrographique en fonction des référentiels.....	28
8.2 Création d'un référentiel hydrographique composite.....	30
8.2.1 Assemblage des couches DDTM et IGN.....	30
8.2.2 Correction de la table composite.....	31
8.2.2.1 Préparation du MNT.....	32
8.2.2.2 Comblement des dépressions.....	32
8.2.2.3 Construction d'un réseau théorique d'écoulement.....	33
8.2.2.3.1 Construction du flux d'accumulation.....	33
8.2.2.3.2 Construction du réseau d'écoulement théorique.....	33
8.2.2.4 Correction de la table composite – raccordements tronçons isolés et gestion des anastomoses.....	34
8.3 Construction des micros bassins versants.....	37
8.4 Construction des têtes bassin versant.....	39
9 Caractérisation des têtes de bassin versant	42
9.1 Caractérisation physique et morphologique – Sensibilité des TBV.....	43
9.1.1 Grille d'indicateurs à l'échelle des caractéristiques physiques de la TBV.....	44

9.1.2 Préparation des données rasters nécessaires aux calculs.....	44
9.1.3 Surface de la TBV.....	45
9.1.4 Pente moyenne du bassin versant.....	46
9.1.5 Pente du chemin le plus long.....	47
9.1.6 Indice de Gravélius (ou indice de compacité).....	48
9.1.7 Temps de concentration des pluies (Passini).....	49
9.1.8 Indice de densité de chevelu.....	50
9.1.9 Densité du réseau de points bas.....	51
9.1.10 Position de la TBV dans le réseau hydrographique.....	52
9.2 Caractérisation des pressions.....	53
9.2.1 Grilles d'indicateurs à l'échelle du lit mineur et de la bande riveraine.....	53
9.2.2 Grille d'indicateurs à l'échelle des têtes de bassin versant.....	53
9.2.3 Indice de pression (ou pression liée à l'occupation du sol).....	54
9.2.4 Couche composite cohérente d'occupation du sol.....	55
9.2.4.1 Zones urbaines.....	55
9.2.4.2 Zones urbaines perméables.....	56
9.2.4.3 Réseaux routiers et voies ferrées.....	56
9.2.4.4 Boisements.....	56
9.2.4.5 Surfaces agricoles.....	56
9.2.4.6 L'OCS Théia.....	57
9.2.4.7 Assemblage des différents types d'occupation du sol.....	57
9.2.4.8 Assemblage final de la table OCS bvsn.....	58
9.2.4.9 Principes de requêtes sur la table d'occupation du sol composite.....	58
9.2.5 Les prélèvements en eau.....	60
9.2.6 Les stations d'épurations.....	60
9.2.7 Les haies.....	60
9.2.8 Les plans d'eau et les mares.....	61
9.2.9 Les cours d'eau enterrés.....	61
9.2.10 Le stockage des résultats.....	61
10 Hiérarchisation des têtes de bassin versant.....	65
10.1 Homogénéiser les données – centrer / réduire / borner.....	65
10.2 Assembler les indicateurs – les scénarios de pondérations.....	66
10.3 Classer les résultats – la discrétisation.....	67
10.4 La vulnérabilité.....	69
11 Résultats, critiques et perspectives.....	70
11.1 Résultats cartographiques.....	70
11.2 Critiques et perspectives d'amélioration.....	70
11.2.1 Les indicateurs.....	70
11.2.2 Les données.....	71
11.2.3 L'affinement de la priorisation.....	71
11.2.4 La visualisation.....	71
12 Conclusion.....	78
13 Bibliographie.....	79
13.1 Sites internet :.....	81
14 Annexes.....	83

Glossaire des abréviations

AELB : Agence de l'Eau Loire Bretagne
AERM : Agence de l'Eau Rhin Meuse
AFB : Agence Française pour la Biodiversité
BNPE : Banque Nationale des Prélèvements en Eau
BV : Bassin versant
CLC : Corine Land Cover
CLE : Commission Locale de l'Eau
DCE : Directive Cadre sur l'Eau
DEC : Diagnostics environnementaux communaux
DDT : Direction Départementale des Territoires
DDTM : Direction Départementale des Territoires et de la Mer
EPCI : Établissement Public de Coopération Intercommunale à fiscalité propre
EPTB : Établissement Public Territorial de Bassin
ERC : principe « Éviter, Compenser, Réduire », en matière d'aménagement du territoire.
FMA : Forum des Marais Atlantiques
GEMAPI : compétence GEstion des Milieux Aquatiques et Prévention des Inondations
ID / IDs : Identifiant(s) dans une table Postgres
LEMA : Loi sur l'Eau et les Milieux Aquatiques
MEDDE : Ministère du Développement Durable et de l'Environnement
MNT : Modèle Numérique de Terrain
PAGD : Plan d'Aménagement et de Gestion Durable
PDPG : Plan Départemental de Protection du milieu aquatique et de Gestion des ressources piscicoles
PG : Postgres
REH : Réseau d'Évaluation des Habitats
SAGE : Schéma d'Aménagement et de Gestion des Eaux (échelle micro)
SDAGE : Schéma Directeur d'Aménagement et de Gestion des Eaux (échelle macro)
SRCE : Schéma Régional de Cohérence Écologique
SYsMA : SYstèmes de Suivi des Milieux Aquatiques
TBV : Tête de Bassin Versant
ZH : Zone humide
ZNT : Zone de Non Traitement

2 Introduction

Les petits ruisseaux font les grandes rivières, cet adage que l'on retrouve dans tant de pays comme la Turquie, l'Italie, la Roumanie ou les Pays-Bas et qui nous rappelle qu'un ensemble de petites choses accumulées finissent par former un tout important, ne trouve pas de meilleure illustration que dans le dessin d'un réseau de cours d'eau.

Car ces grands fleuves, qui serpentent langoureusement au travers de nos paysages avant de rejoindre la mer, ne sont-ils pas les enfants de ces sources nombreuses et peu visibles dans notre environnement ? Ce réseau de petits cours d'eau dense ressemble beaucoup dans sa forme fractale aux nervures d'une feuille, aux veines qui irriguent le corps humain... Et cette ressemblance dépasse la forme, ces cours d'eau, à l'instar de nos artères, alimentent le territoire en fluides vitaux et participent à l'équilibre écologique.

Pour l'homme aussi l'eau est un bien indispensable, la majorité de la population humaine est installée près des points d'eau, côtiers ou fluviaux. Les sources sont, encore aujourd'hui, parfois marquées par la présence d'un puits, d'un lavoir, d'un calvaire, d'une petite chapelle, d'un toponyme ou d'une légende qui reflète l'importance et les cultes anciens que les sociétés humaines leur ont voués. Selon Élisée Reclus au milieu du 19^e siècle, les sociétés du sud et d'Orient y étaient d'autant plus sensibles que l'eau y était rare, les sources y étaient alors vénérées et l'eau économisée. Pourtant l'industrialisation de la fin du 19^e et la révolution mécanique intervenue, entre autres, dans le monde agricole à partir de la fin de seconde guerre mondiale, ont fait muter la vision populaire d'un cours d'eau source de vie vers un cours d'eau utilitaire. Dès lors, l'aménagement du territoire n'a eu de cesse de vouloir maîtriser cette ressource, que ce soit pour s'en protéger, pour en profiter ou pour s'en débarrasser.

Irrigation, drainage, chenalisation, artificialisation, nécessité pour certains, plaies pour d'autres, ces actions, quelques soit l'intérêt qu'elles peuvent servir, ont eu pour conséquence indirecte une dégradation nette de la qualité des eaux continentales.

Depuis plusieurs années, les sociétés occidentales semblent avoir pris la mesure des problématiques engendrées. En France, la première loi sur l'eau voit le jour en 1964 avec pour principal aboutissement la constitution des agences de l'eau et d'une gestion par bassin hydrographique. En 1992, la seconde loi sur l'eau ira plus loin en posant le principe de l'eau comme patrimoine commun de la nation, puis la loi sur l'eau et les milieux aquatiques (LEMA) en 2006 transposera la DCE (Directive Cadre sur l'Eau) européenne dans le droit français.

Malgré tout, aujourd'hui de nombreux rapports font le constat d'un mauvais état des cours d'eau, en France ou à l'étranger. Le réchauffement climatique pose de nombreuses questions sur l'avenir de cette ressource, que ce soit lors d'épisodes de sécheresse comme au Brésil en 2017 et au Moyen Orient (où de nombreux observateurs dont l'Académie Nationale des Sciences américaines voient dans la sécheresse de 2007/2010 en Syrie un catalyseur ayant contribué à l'instabilité politique que l'on connaît aujourd'hui) ou lors d'épisodes de crues, comme les a connus le bassin parisien ces deux dernières années.

Alors, si les sources sont les parents des fleuves, les problématiques liées à l'eau trouvent peut-être leur origine dans les parties apicales des bassins versants¹. C'est ce que semble indiquer les résultats des recherches faites sur les têtes de bassin versant. Bien que peu étudiées avant les années 80 elles ont connu un accroissement significatif du nombre de parutions depuis le début des années 2000 (source AFB). Tout d'abord, les cours d'eau en tête de bassin versant représentent de **60 à 80 % du linéaire total de cours d'eau** (Schumm, 1956 - Shreve, 1969 - Meyer & Wallace, 2001 - Benda *et al.*, 2005 - Le Bihan, 2009 - Malavoi, 2009). Malgré tout, leur cartographie s'avère très incomplète, de

1 Un bassin versant désigne une portion de territoire délimitée par une ligne de partage des eaux et irriguée par un même réseau hydrographique, dans lequel l'ensemble des eaux de surface rejoint le même point exutoire.

nombreux cours d'eau en tête de bassin demeurent méconnus (Meyer *et al.*, 2001), des travaux d'études (Lhéritier 2012, Van Nguyen 2012) montrent que jusqu'à 50 % du réseau peuvent être oubliés sur les cartes, Sur le bassin versant de la Vilaine en Bretagne, la cartographie des cours d'eau réalisée en collaboration avec l'IGN représente un apport moyen de 30 % de linéaire supplémentaire par rapport au référentiel IGN actuel.

Ces manquements se concentrent essentiellement sur ces têtes de bassin versant où les écoulements sont faibles et parfois temporaires, sur des secteurs historiquement très aménagés et anthropisés. Or, ce sont sur ces zones que les litiges liés à la loi sur l'eau sont les plus nombreux.

Afin de clarifier la situation, l'État a commandé à ses services décentralisés le 3 juin 2015 (DDT, DDTM) de réaliser des inventaires départementaux. Cependant, les délais et les moyens mis en œuvre, ainsi que l'implication des acteurs du territoire sont très variables d'un département à l'autre aboutissant à des résultats de qualité hétérogène. On notera dans les inventaires livrés des erreurs topologiques et une absence d'interopérabilité entre départements. La vérification sur site n'étant pas systématique dans tous les départements, la plupart des inventaires sont considérés par les services de l'État comme évolutifs au gré des incohérences pouvant être signalées par les acteurs locaux.

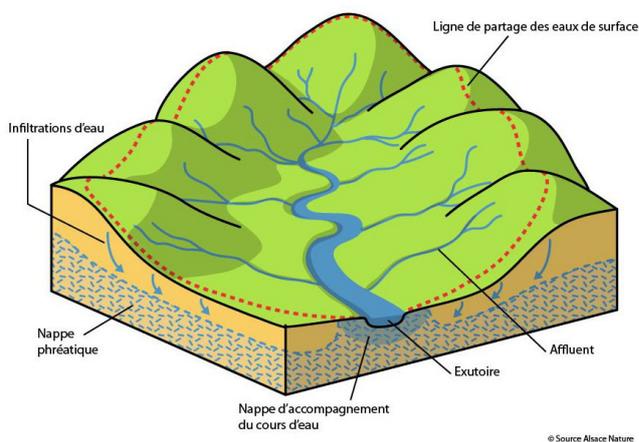


Illustration 1: Définition graphique de la notion de bassin versant

Toutefois, cet intérêt pour les têtes de bassin versant de la part des planificateurs et organismes publics d'aménagement de l'espace - et en particulier dans le domaine de la gestion des milieux aquatiques - est positif et ne peut qu'améliorer leur prise en compte. Car si les têtes de bassin versant représentent la majorité du linéaire des cours d'eau, il en va de même dans les différents compartiments de services écosystémiques² auxquels elles contribuent. Par exemple en matière d'auto épuration de l'eau, elles constituent des lieux privilégiés (Oraison *et al.*, 2012). Ainsi, depuis quelques années, les comités de bassin et commissions locales de l'eau se sont emparés du sujet, en commençant par la volonté d'améliorer la connaissance des têtes de bassin grâce à leur cartographie et à leur caractérisation. L'intérêt premier de ces délimitations s'inscrit dans le cadre de la gestion des milieux aquatiques. Au-delà, cette démarche vise à décloisonner l'aménagement de l'espace par une meilleure concertation des différents acteurs, sylviculteurs, services d'urbanismes, gestionnaires des eaux pluviales, entrepreneurs agricoles, services des grands travaux...

La commande de la présente étude trouve son origine dans la disposition 46-1 du SAGE de la Sèvre Nantaise qui prévoit pour l'année 2018 une localisation précise ainsi qu'une amélioration des connaissances sur ces zones. Dans les pages qui suivent il s'agira de présenter une chaîne de traitement informatique automatisée consistant en la localisation, la caractérisation et la hiérarchisation des têtes de bassin versant. Des études similaires ont déjà vu le jour depuis le début des années 2010 sur le bassin Loire Bretagne, mais, sur certaines étapes, les méthodologies utilisées sont peu reproductives (en l'occurrence, majoritairement sur la correction et la préparation des référentiels hydrologiques). L'idée générale de l'étude consiste donc à fournir « clé en main » un outil de localisation et de caractérisation, en passant par les étapes indispensables de préparation et de correction des données. L'intérêt est pour la structure accueillante de pouvoir rejouer à l'infini le processus avec de nouveaux jeux de données, référentiels MNT ou hydrographique ou encore avec de nouveaux scénarios de priorisation et, pour les autres structures, de pouvoir s'appuyer sur une méthodologie décrite pas à pas, voire de s'équiper de l'interface web dédiée.

2 Les services écologiques rendus par les écosystèmes généralement classés comme bien commun et/ou public car vitaux ou utiles pour l'humanité, les autres espèces et les activités économiques.

3 Présentation de la structure d'accueil et du territoire

L'EPTB de la Sèvre Nantaise est un syndicat mixte ayant pour mission la gestion et l'aménagement du bassin versant de la Sèvre Nantaise. Le territoire dont il a la gestion représente 2 350 km², 123 communes, 16 intercommunalités pour 320 000 habitants et plus de 2000 kilomètres de rivières cumulés. Ce territoire correspond au bassin versant hydrologique de la Sèvre Nantaise et permet à l'EPTB d'assurer ces missions avec cohérence dans le respect du principe de solidarité amont – aval.

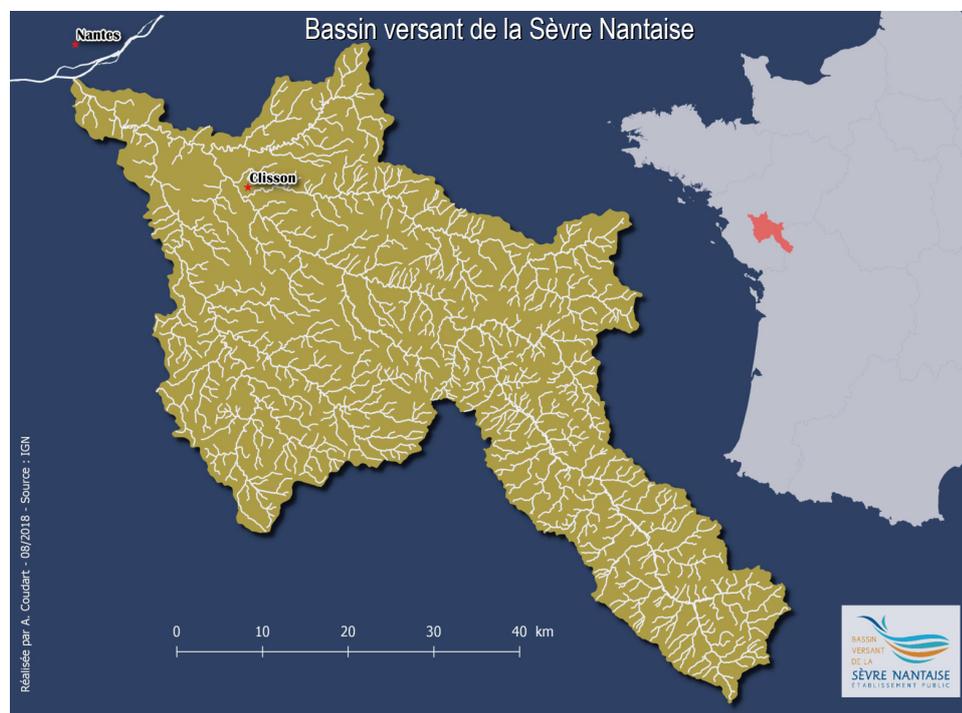


Illustration 2: Territoire du bassin versant de la Sèvre Nantaise

Comme la plupart des structures de bassins, l'EPTB a pour particularité de s'affranchir des limites administratives habituelles. Ici, le territoire s'étale sur 4 départements (Deux-Sèvres, Vendée, Maine-et-Loire et Loire-Atlantique) et 2 régions (Pays de la Loire et Nouvelle Aquitaine). Ces adhérents sont composés des 4 départements ainsi que des EPCI présents sur le territoire (communautés de communes, communautés d'agglomérations, métropole, syndicat d'eau potable).

Sa forme juridique n'a eu de cesse d'évoluer depuis ces origines dans les années 80, la dernière réforme de la GEMAPI applicable au 1^{er} janvier 2018 en est la dernière mouture. Le rôle de l'EPTB est donc d'assurer l'animation en matière de gestion des milieux aquatiques, une mission qui se décline en plusieurs typologies d'actions :

- Fédérer les acteurs du bassin versant autour des problématiques liées à l'eau
- Améliorer la qualité de l'eau
- Lutter contre les pollutions
- Entretien et restaurer les milieux aquatiques
- Prévenir les inondations
- Valoriser le patrimoine (naturel et culturel)
- Informer, sensibiliser et communiquer

Le financement de ces actions est assuré par l'Agence de l'Eau Loire Bretagne, les régions, les départements et les membres de l'EPTB. L'ensemble concourt à la mise en conformité avec la directive cadre européenne sur l'eau (directive DCE 2000/60) qui prévoyait initialement d'atteindre un bon état écologique des différents milieux pour 2015, la dernière échéance étant aujourd'hui programmée pour 2027. Le dernier rapport 2018 de l'Agence de l'Eau estime à 4 % la part des cours d'eau en bon état sur le territoire Loire aval et côtiers vendéens (dont fait partie le bassin de la Sèvre Nantaise) ce qui en fait aujourd'hui le sous-bassin Loire Bretagne le plus dégradé.

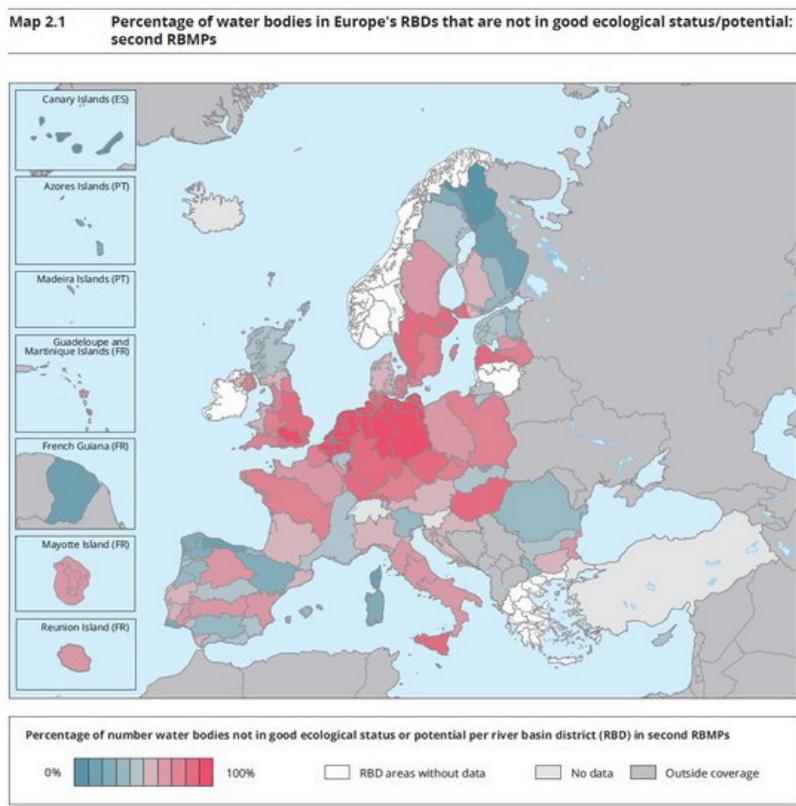


Illustration 3: Pourcentage de cours d'eau en bon état écologique au regard des objectifs DCE. Freshwaterblog, 2018, source union européenne

Les objectifs fixés sont encadrés par le SDAGE Loire Bretagne, décliné localement par le SAGE de la Sèvre Nantaise. Ce document de planification est une feuille de route listant en détail les actions à mener. Il est approuvé par un arrêté préfectoral, ces dispositions sont donc opposables dans l'ensemble des documents d'urbanisme ou d'aménagement du territoire.

4 Définition de la tête de bassin versant

Les têtes de bassin versant définissent l'extrémité amont du réseau hydrographique, mais la définition de l'amont reste très discutée et les interprétations sont multiples que ce soit entre les différentes parties du monde ou même en France entre les différentes agences de l'eau. Les critères de distinctions retenus ne sont pas toujours les mêmes :

- Superficie du BV (Adams & Spotila, 2005 - Brummer & Montgomery, 2003 - Meyer *et al.*, 2003... USA)
- Gabarit du lit mineur (Wipfli *et al.*, 2007 USA - AERM, 2009 France)
- Hydromorphologie (Uchida *et al.*, 2005 - Japon)
- Hydrologie (Touchart, 2006 - France)
- Hydrographie (Strahler, 1952 - Vannote *et al.*, 1980 - France)

Si les critères semblent plus ou moins pertinents en fonction des échelles d'analyses, certains s'avèrent compliqués à mettre en œuvre sur de grands territoires comme le critère de définition hydromorphologique qui retient le secteur à l'amont de « la zone de sédimentation dominante ». **Le critère retenu par le comité de bassin Loire Bretagne pour le SDAGE 2016 – 2021 est un critère hydrographique décrit par Vannote en 1980 et qui retient les cours d'eau de rangs de Strahler 1 et 2 et présentant une pente supérieure à 1 %** (quand Strahler incluait les rangs 3 dans sa description des zones amont). Ce critère de pente peut être adapté localement pour les cours d'eau à faible puissance spécifique et présentant un risque de non-atteinte des objectifs environnementaux. Les rangs de Strahler (1957) sont une classification des réseaux hydrologiques, l'une des premières avec celle d'Horton, à considérer la source comme unité d'origine et non pas les fleuves. D'un point de vue sociologique la différence est notable, dans ce cas, ce n'est plus la taille qui compte mais l'origine. Cette hiérarchisation du chevelu attribue à chaque tronçon une valeur entière qui caractérise son importance et sa place dans l'ensemble. Elle consiste à accorder aux cours d'eau un rang qui augmente de l'amont vers l'aval, celui-ci étant incrémenté de 1 dès lors que deux rangs de même ordre entrent en confluence.

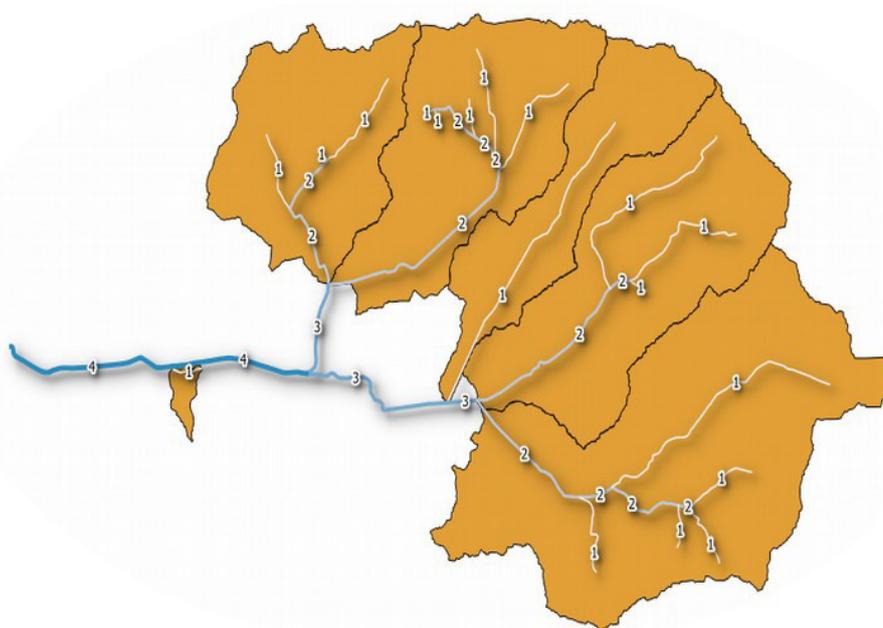


Illustration 4: Schéma classification selon l'ordre de Strahler

Au sein des différentes agences de l'eau françaises, toutes ne portent pas la même importance aux têtes de bassin versant et seule l'AELB apporte une définition officielle dans son SDAGE.

- **L'agence Seine - Normandie :**

« Partie amont des bassins versants et par extension tronçon amont des cours d'eau qui, en zone de relief notamment, sont le plus souvent moins exposés aux pressions anthropiques que les parties avalées (mais restent très fragiles) et qui de ce point de vue constituent des secteurs de référence à préserver. (rangs 1 et 2). »

- **L'agence Rhin - Meuse :**

Simplement évoquées dans la **partie 'bonne pratique sur les plans d'eau'**, « Prévoir dans les Plans d'aménagement et de gestion durable (PAGD) ou dans les règlements de chaque SAGE, en fonction de la sensibilité du milieu, de son état actuel et de son fonctionnement, des critères conditionnant la délivrance des autorisations ou l'acceptation des déclarations de création de nouveaux plans d'eau, voire leur interdiction sur les zones les plus fragiles (têtes de bassin, notamment en première catégorie piscicole, zones de faibles débits, etc.) ».

Définition par les rangs 1 et 2, trouvée dans des documents mais pas de définition dans le SDGE 2016 – 2021.

- **L'agence Rhône - Méditerranée :**

Évoquées dans les parties ERC, polluant en milieu sensible et dans la partie sur la non-dégradation des milieux aquatiques, « ...les têtes de bassin versant compte tenu de **leur importance au plan de la qualité de l'eau** et des équilibres hydro-sédimentaires... ».

Pas de définition dans le SDAGE 2016 - 2021

- **L'agence Adour - Garonne :**

Largement présentes dans le SDAGE 2016 – 2021, à différents chapitres, définies comme jouant un **rôle bénéfique pour l'atteinte des objectifs de bon état**. Disposant des **orientations D21 et D22** sur l'amélioration de la connaissance et sur le renforcement et la restauration des chevelus hydrographiques. Dans la partie sur le **réchauffement climatique**, ce paramètre doit être pris en considération pour la gestion et la préservation des têtes de bassin, et enfin dans la **partie sur les plans d'eau**. Pour autant, aucune définition claire n'est donnée dans le SDAGE.

- **L'agence Artois - Picardie :**

Seulement citées dans le **chapitre concernant les plans d'eau** : « limiter la création et l'extension de plans d'eau en tête de bassin des cours d'eau, connus pour leur vulnérabilité. »

Nous parlerons de la définition et des citations des têtes de bassin versant dans le SDAGE **Loire Bretagne** dans la partie sur le contexte réglementaire. L'EPTB de la Sèvre Nantaise faisant partie du bassin Loire Bretagne et étant soumis à son SDAGE, la définition choisie par le comité de bassin Loire-Bretagne sera donc notre porte d'entrée. Le choix est également fait de ne pas inclure le critère de pente dans la définition, en effet, ce dernier ne trouve pas d'appui scientifique concret dans la bibliographie et la précision apportée par le SDAGE classe le territoire du SAGE Sèvre Nantaise dans la catégorie des cours d'eau à faible puissance spécifique et présentant un risque de non-atteinte des objectifs environnementaux. Cette définition, bien que discutable d'un point de vue hydrologique, car étant soumis au référentiel cartographique utilisé et donc aux biais d'absence évoqués plus haut, a le mérite d'être claire, reproductible et facilement modélisable avec des outils géomatiques.

5 Fonctionnalités des têtes de bassins versant

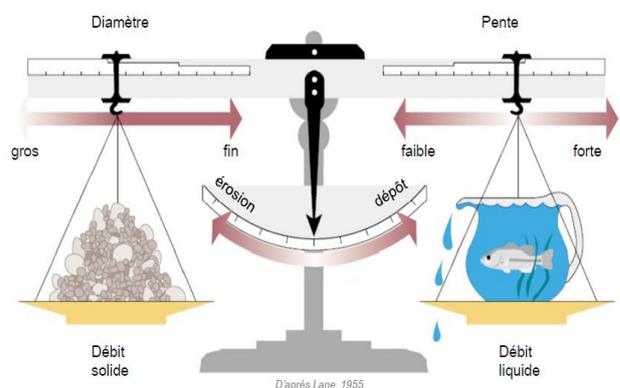
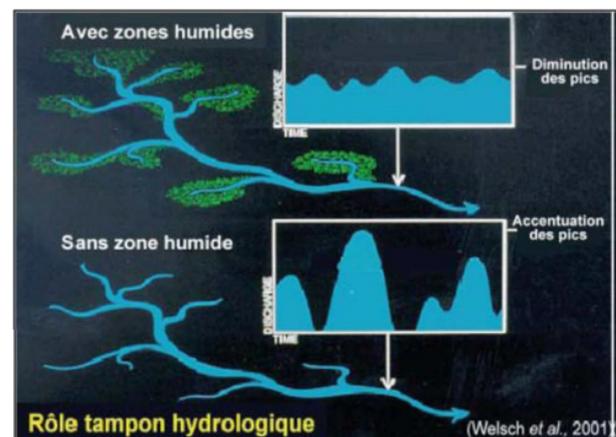
Les têtes de bassin versant sont décrites comme notre « **capital écologique** » par Lucien Maman en 2007 alors en charge de la question à l'AELB. Par ce terme, il introduisait la notion des services écosystémiques rendus par ces complexes hydrauliques. La bibliographie scientifique internationale nous montre que ces derniers sont nombreux et de différentes natures, avec un point commun, l'état de dégradation conditionnant fortement l'état des cours d'eau de rang supérieur en connexion aval. Comme cité précédemment, le réseau des rangs 1 et 2 représente en moyenne 75 % du réseau total sachant que jusqu'à 90 % des cours d'eau de TBV connus ont été recalibrés dans certains départements (Colin, 2015). Les enjeux et objectifs d'atteintes de bonne qualité des eaux justifient pleinement l'intérêt qui est porté à leur préservation et à leur restauration. La présente étude n'ayant pas l'objectif de travailler spécifiquement sur ces aspects fonctionnels, ce qui suit est issu d'un travail de recherche bibliographique au travers des différents travaux d'études sur les TBV de ces dernières années. Pour plus de détails, un ensemble de guides édités par l'ONEMA (actuelle AFB) permettra de développer largement les informations. **L'intérêt de détailler ici les différentes fonctions permet d'introduire les logiques qui ont prévalu aux choix des indicateurs de caractérisation qui seront exposés dans le cœur technique de l'étude.**

5.1 Fonctions hydrauliques et hydromorphologiques :

Le SDAGE Loire Bretagne dans son chapitre 11 définit les TBV comme notre « **capital hydrologique** » par déclinaison du capital écologique. Dans ce cas, cette dénomination se réfère au fait que le chevelu participe pour 50 à 70 % du volume d'eau transitant dans les cours d'eau aval (Alexander, 2007). Cette participation majeure conditionne de fait le double rôle des TBV dans **l'écrêtage des crues** en tant qu'éponge en période de pluie **et le soutien d'étiage** en tant que château d'eau alimentant l'aval en période sèche. Les zones humides qui sont une composante importante du complexe TBV sont autant de zones tampon, majoritairement de faible surface individuelle mais représentant une surface totale importante (Lhéritier, 2012), contribuant à diminuer les pics hydrauliques (Welsch et al, 2001). De plus, un ruisseau présentant une morphologie naturelle et non recalibrée favorise la désynchronisation du transit de l'amont vers l'aval (Gomi et al, 2002) participant de fait à l'écrêtage des crues³.

Les têtes de bassin versant sont parfois définies par le critère hydromorphologique comme situées en amont des premières zones de sédimentations dominantes. Cela signifie qu'elles sont des zones de production sédimentaire. Elles sont le premier maillon de la chaîne d'**équilibre dynamique** proposée par Borland et Lane en 1955.

Ce schéma représente le conditionnement du transport solide par la pente et le débit. Plus la granulométrie des éléments solides constatés est grosse plus la pente et le débit liquide sont importants, l'érosion en sera, par voie de



3 Abaissement du débit de pointe d'une crue par stockage, rétention artificielle ou en favorisant l'extension dans les zones d'expansion des crues (prairies alluviales, zones humides, marais alluviaux...).

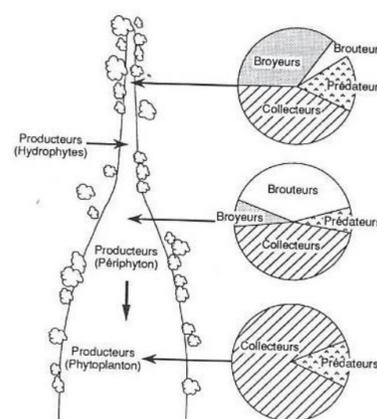
conséquence, augmentée. À l'inverse plus les débits et les pentes sont faibles et plus les matériaux du lit de la rivière seront fins ayant pour conséquence le dépôt de ces derniers plutôt qu'un transport érosif. Il est communément accepté que ce que l'on trouve dans le lit d'une rivière à un endroit précis est mobilisable par le cours d'eau à cet endroit précis.

Les ruisseaux ont donc un rôle primordial dans l'hydromorphologie des réseaux hydrographiques, puisqu'ils constituent la **zone de production sédimentaire** (du rang 1 jusqu'au rang 4). Ainsi, les alluvions et colluvions fournis par les ruisseaux et les versants constituent une part importante des sédiments des rivières apparentées à la zone de transfert. Les fleuves quant à eux sont des zones de stockage. Toutefois, s'ils ont un tel pouvoir de production sédimentaire, celui-ci n'est pas égal et sera **conditionné par la puissance spécifique du cours d'eau, sa pente, son profil hydrogéologique et sa taille.**

5.2 Fonctions biologiques et trophiques⁴ :

En termes de biodiversité, les têtes de bassin versant composent une **mosaïque d'habitats** particulièrement riches (AFB). Boisements humides, tourbières, plaines alluviales, marais alluviaux, zones humides, profil courant et rapide, profil lent et serpentiforme, mares... autant d'habitats permettant d'abriter une diversité en termes de micro organisme, bactéries, champignons, invertébrés. Les têtes de bassin versant sont souvent décrites comme les **nurseries** des rivières, nombres d'études démontrent que le cycle de vie de certaines populations piscicoles dépendent de ces zones par exemple le Chabot (*Cottus gobio*), l'Écrevisse à pattes blanches (*Austropotamobius pallipes*), la Lamproie de planer (*Lampetra planeri*), la Moule perlière (*Margaritifera margaritifera*) (Mathieu, 2010). Plusieurs études (Lhéritier, 2012 - ONEMA, 2010) ont montré que les populations piscicoles en aval sont plus importantes lorsque les têtes de bassin sont en bon état et connectées au reste du réseau. Présentant une amplitude thermique plus réduite qu'à l'aval, ce sont également des **zones refuges** pour les espèces sténothermes⁵ d'eau froide migrant vers les zones les plus amonts des cours d'eau en période estivale (Meyer *et al.*, 2007).

Une tête de bassin versant préservée, dont la ripisylve⁶ est en bon état présentera des **températures plus fraîches** propices aux espèces inféodées. En termes d'habitats, la ripisylve permet la formation des sous berges particulièrement riches en biodiversité. Une étude similaire sur le bassin versant du Couesnon montrait qu'entre 86 à 89 % des TBV en présentaient (Jan, 2012). De plus, **la ripisylve et l'écosystème environnant permettent l'introduction de matière organique** qui va ensuite constituer les premiers maillons de la chaîne alimentaire (Henner, 2013). Les invertébrés broyeurs occupent une bonne part dans le peuplement d'invertébrés benthiques des ruisseaux, leur rôle dans l'assimilation de la matière organique par le cours d'eau en la transformant mécaniquement est très important.



Comme rappel à l'interaction systématique des différents compartiments, le bois en rivière issu de la ripisylve environnante présente très souvent (69 % des cas) (Le Bihan, 2017) un rôle hydromorphologique (rétenteur, déflecteurs, augmentation de la diversité des faciès d'écoulement...).

4 Relatif à l'alimentation.

5 Adjectif qualifiant les organismes ne tolérant que des variations de température de faible amplitude autour des valeurs moyennes.

6 Forêt alluviale.

5.3 Fonctions Physico-chimiques :

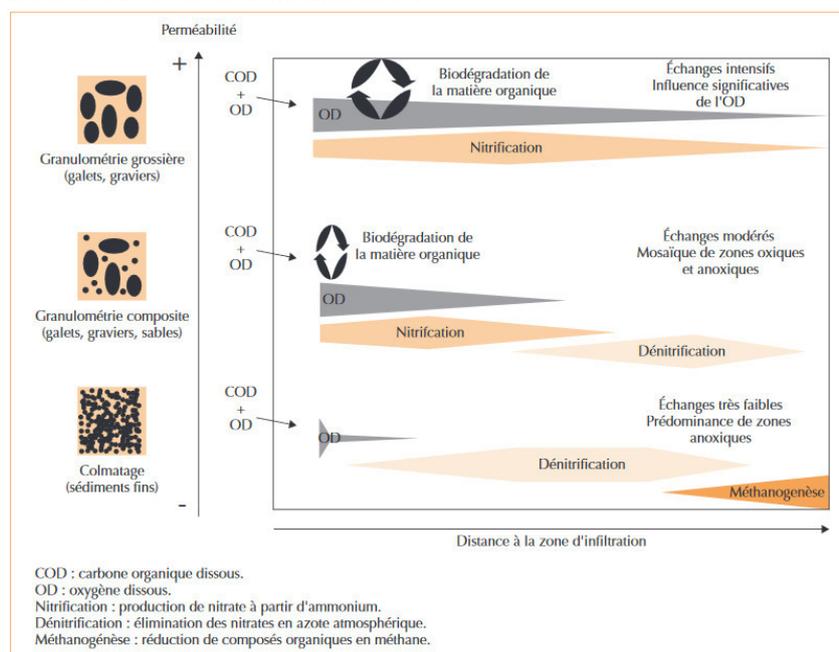
Les cours d'eau et leur écosystème associé disposent d'un panel d'agents contribuant à l'auto-épuration. L'auto-épuration comprend l'ensemble des processus naturels qui permettent le recyclage des éléments minéraux et organiques et l'élimination des charges excessives (Ostroumov, 1998). Ici, comme dans les autres compartiments, la qualité de l'eau des têtes de bassin versant conditionne la qualité à l'aval (Alexander et al, 2007).

Les **zones humides** en tête de bassin versant sont des **zones de forte dénitrification**⁷ (Pinay, 2015) pouvant participer à hauteur de 120 à 140 kg/ha de dénitrification en moyenne pour le cas de ZH saturées en eau toute l'année (Agro-Transfert, université de Rennes, 2009) Elles sont également source de polluant. On estime que 60 % de la charge en nitrates trouvée dans les cours d'eau d'ordre supérieur à 3 proviendrait des cours d'eau de rang 1 (Alexander et al., 2007).

La **ripisylve**, qu'elle soit aulnaie, saulaie, frênaie participe activement à la bonne qualité de l'eau en tant que **filtres naturels** et permet également de maintenir une température plus faible (cahier technique du bassin de la Loire, CEN, 2017).

Les **tourbières** de ces zones sont, quant-à elles, des **puits de carbone** contribuant à fixer ce dernier (Laggoun-Défarge et al, 2008 - Joosten et Clarke, 2002).

Les **surfaces hyporhéique**⁸ importantes de part l'importance du réseau de rangs 1 et 2, sont propices aux procédés de dénitrification (Thomas et al., 2001 - Bohlke et al., 2004 - Mulholland et al., 2004) et de **nitrification**⁹ en fonction du substrat. Toutefois, ces dernières fonctions sont à mesurer en fonction d'autres paramètres physiques tel que les pentes, la granulométrie¹⁰ ou le substrat géologique. Sur une source imperméable où la couche d'alluvions est faible, la surface hyporhéique sera réduite et donc sa capacité d'auto-épuration par voie de conséquence (AFB via eaufrance.com, 2016). En fonction de la granulométrie, les micros organismes à l'œuvre ne seront pas les mêmes et les fonctions remplies par ces derniers seront donc différentes.



7 4e et dernière phase du cycle de l'azote avant l'assimilation par les plantes et l'atmosphère.

8 Zones d'interface entre la nappe et le cours d'eau.

9 3e phase du cycle de l'azote.

10 Étude de la répartition des éléments d'une roche, d'un gravier, d'un sable selon leur taille.

6 Contexte réglementaire

La source de cette étude trouve son origine dans les textes réglementaires qui encadrent les actions territoriales en matière de gestion des milieux aquatiques. En premier lieu, rappelons que la France, comme tous les états membres de l'union européenne, a pour devoir de se conformer aux objectifs de la DCE (**Directive Cadre sur l'Eau**). Ces objectifs visent une atteinte du "bon état écologique" des cours d'eau à différents pas de temps. Ainsi pour accompagner les acteurs de l'eau dans cette démarche des documents officiels font office de feuille de route. À l'échelle macro (Loire/Bretagne, territoire de l'Agence de l'Eau Loire Bretagne), nous trouverons le SDAGE (Schéma Directeur d'Aménagement et de Gestion des Eaux) et à l'échelle micro nous trouverons les SAGEs (Schéma d'Aménagement et de Gestion des Eaux) qui transcrivent le SDAGE dans un cadre local en l'occurrence la Sèvre Nantaise dans cette étude.

6.1 Ce que dit le SDAGE

Le SDAGE Loire Bretagne 2016-2021, consacre un chapitre à part entière à la question des têtes de bassins versants. Mais cette thématique est présente dans d'autres chapitres. Ainsi dès la présentation générale du SDAGE il y est fait référence dans les SRCE (Schémas Régionaux de Cohérence Écologique), où elles sont décrites comme jouant un rôle de réservoirs de biodiversité.

Le chapitre 1, qui introduit la nécessaire refonte des aménagements de cours d'eau, présente les têtes de bassin versant comme des milieux sensibles, dont le bon fonctionnement est important pour l'ensemble du bassin et sur lesquels il est nécessaire de porter une attention particulière.

Dans le chapitre 8 concernant la préservation des zones humides et plus précisément dans la partie 8A-2, les têtes de bassins versant font parties des zones ciblées comme présentant un intérêt particulier en matière d'acquisition foncière.

Dans le chapitre 9 concernant la préservation de la biodiversité aquatique, il apparaît que les têtes de bassins versants peuvent faire l'objet de mesures particulières dans les plans départementaux pour la protection du milieu aquatique et la gestion des ressources piscicoles (PDPG).

Le chapitre 11 lui est spécialement consacré, en 2 volets : 11A – Restaurer et préserver les têtes de bassin versant et 11B – Favoriser la prise de conscience et la valorisation des têtes de bassin versant. Elles y sont décrites comme "notre capital hydrologique". La disposition 11A-1 impose aux SAGEs un inventaire systématique des zones de têtes de bassin versant ainsi qu'une analyse des caractéristiques notamment écologiques et hydrologiques. C'est dans ce chapitre qu'une définition est donnée, **elles sont décrites comme les bassins versants des cours d'eau dont le rang de Strahler est inférieur ou égal à 2 et dont la pente est supérieur à 1 %**. Ce critère de pente peut-être adapté localement pour les cours d'eau à faible puissance spécifique et présentant un risque de non atteinte des objectifs environnementaux. Ce dernier critère ne trouve pas d'appui dans la bibliographie scientifique. La disposition 11A-2 préconise que les SAGEs hiérarchisent les têtes de bassin versant en fonction des pressions et de l'état des masses d'eau pour définir des objectifs et principes de gestion adaptés.

6.2 Ce que dit le SAGE

Dans le SAGE du bassin de la Sèvre Nantaise validée par arrêté préfectoral en 2015, les préconisations du SDAGE sont reprises et font l'objet de la disposition 46-1 : **Améliorer la connaissance des têtes de bassin versant**. Cette disposition reprend le travail de pré localisation validée par la CLE (Commission Locale de l'Eau) le 11 juillet 2013 et prévoit **une étude précisant la localisation, la caractérisation des fonctionnalités** aux regards des enjeux associés (qualité d'eau, quantité, biodiversité, morphologie), ainsi que le niveau de vulnérabilité et les modes de gestion. Il est également précisé que la méthode de travail doit-être établie en concertation avec les bassins versants limitrophes dans un souci d'homogénéité et de cohérence inter-bassins.

6.3 Les leviers réglementaires

La délimitation des têtes de bassin versant et leur caractérisation n'a pas pour finalité de rajouter un nouveau cadre réglementaire (MAMAN L., 2017 - AUGIER A., 2017 - Entretiens avec la DDTM 35 et l'Agence de l'Eau Loire Bretagne, délégations Centre et Bretagne - Retours d'expérience têtes de bassin versant v1.8 de l'AFB). Comme énoncé plus tôt, l'objectif est de proposer un découpage fin du territoire pour aider les différents acteurs de l'aménagement du territoire à rationaliser leur actions visant la reconquête de la qualité de l'eau. Les têtes de bassins versant, que ce soient les cours d'eau ou leurs zones humides associées sont déjà concernées par la législation en vigueur (LEMA, 2006). Leur protection est associée à leur cartographie, hors, ce sont précisément sur ces cours d'eau de faibles dimensions et intermittents que la cartographie fait défaut (Meyer *et al.*, 2003 - Lhéritier, 2012 - Nguyen Van R., 2012). L'État mène actuellement par le biais de ses services décentralisés (DDT, DDTM), une campagne de cartographie des cours d'eau. Cette dernière à une portée juridique forte puisque c'est sur cette base que la protection liée à la LEMA s'appliquera ou non. Dans le cas ou un rang 1 ne serait pas cartographié, il perdrait ainsi sa protection juridique. De plus les rangs 0 ou zones de sources (Benda *et al.*, 2005), hors périmètre de zone humide inscrit au PLU(i) ou autre protection communautaire (Natura 2000...), ne sont aujourd'hui pas concernés par les mesures de protection alors qu'ils sont des éléments indissociables des cours d'eau en tête de bassin (Le Bihan, 2009).

En conséquence et au regard des enjeux concernant la protection de ces milieux et des pressions dont ils font l'objet, il apparaît intéressant de pouvoir proposer aux pouvoirs publics des pistes de mesures applicables sur leurs territoires. En matière d'urbanisme, les communes et intercommunalités possèdent un levier majeur avec les PLU(i), **la délimitation et la caractérisation des têtes de bassins versant peuvent apporter des éléments de connaissance pouvant faciliter la prise de décisions**, comme par exemple des sites sensibles à protéger de l'imperméabilisation, ou des sites de réservoirs de biodiversité. Ainsi sans rentrer dans une politique de protection globale de tête de bassin versant, il est tout à fait possible de jouer sur des leviers conditionnant l'aménagement du territoire à la prise en compte des services écosystémiques rendus par ces espaces.

Quelques exemples :

- Urbanisation possible avec traitement des eaux pluviales en infiltration à la parcelle.
- Compensation environnementale si destruction des services rendus par une tête de bassin versant.
- Identification de zone de source
- Protection d'éléments bocagers en secteur à fort risque érosif
- ...

6.4 Solutions et incitations réglementaires expérimentées sur le bassin Loire-Bretagne

- *La charte de l'agriculture et de l'urbanisme du Morbihan* -- par le département du Morbihan (depuis 2008). Cette charte incite les communes et intercommunalités à traduire dans leurs documents d'urbanisme un règlement "interdisant toutes constructions à 35 m des cours d'eau et tout comblement, affouillement, exhaussement de terrains (sauf création de retenues d'irrigation autorisées dans le respect de la loi sur l'eau et mise aux normes environnementales autorisée)."
- *Le SCOT 2011* -- Communauté d'agglomération de la presqu'île de Guérande (CAP Atlantique). L'objectif 1.2.2 Protéger les milieux humides et les abords des cours d'eau, contient une prescription concernant les zones de sources (objet d'une étude interne consistant à élaborer une méthodologie d'identification de l'origine des rangs 1) : "Les collectivités préciseront les zones de sources effectives afin de :

- Les intégrer dans la trame verte et bleue des documents d'urbanisme locaux
- S'assurer que les travaux d'aménagements ne suppriment pas ces zones ni ne les déconnectent des cours d'eau ou, en cas de reconfiguration des zones de sources (notamment en espace urbain existant ou programmé), que ces travaux intègrent les moyens de préserver l'alimentation des cours d'eau ou d'améliorer leur fonctionnement hydraulique. "

- *Le SAGE Couesnon* -- (depuis 2013). La délimitation des têtes de bassin versant a été inscrite dans le règlement et le PAGD du SAGE Couesnon. Il y est indiqué que les travaux sur les cours de tête de bassin versant sont interdits sur le bassin versant du Couesnon (rangs 1 et 2 avec 1 % de pente). Plus précisément :
 - La mise en place d'ouvrage faisant obstacle à l'écoulement de l'eau et à la migration des poissons (chute supérieure à 20 cm).
 - Le reprofilage du cours d'eau.
 - Le busage d'un cours d'eau.
 - L'artificialisation des berges (à partir de 20 m).

(C'est à ce jour, le seul exemple connu d'intégration effective des têtes de bassin versant dans un règlement de SAGE.)

- *Les zones humides prioritaires* -- Étude menée par le Forum des Marais (FMA). Étude et éléments de compréhension sur la portée réglementaire des dispositifs ZHIEP (Zone Humide d'Intérêt Environnemental Particulier) et ZSGE (Zones humides Stratégiques pour la Gestion de l'Eau).

7 Contexte de l'étude

Au-delà des objectifs de délimitation et de caractérisation des têtes de bassin du bassin versant de la Sèvre Nantaise, l'étude s'articulait autour d'un certains nombres de contraintes. La méthodologie à mettre en œuvre se devait d'être reproductible et automatisée. Les détails de la contrainte technologique seront abordés dans la partie dédiée. D'autre part, une contrainte de cohérence territoriale est précisée dans le SAGE de manière explicite « la méthode de travail devra-être établie en concertation avec les bassins versants limitrophes dans un souci d'homogénéité ».

7.1 Échanges avec les bassins voisins

Le début de l'étude a donc été voué en partie à la recherche bibliographique et surtout à la concertation. Ainsi de nombreux contacts ont été pris parmi les acteurs locaux :

- Structures voisines.
- Anciens stagiaires ou responsables de stages similaires.
- Partenaires techniques.
- Partenaires financiers.
- Partenaires réglementaires.
- Élus locaux.

Ces nombreux contacts et échanges ont été très importants. Mickaël Le Bihan de l'AFB est le référent incontournable de l'ouest en matière de têtes de bassin versant. L'importance croissante de la thématique lui permet d'inciter les structures à s'organiser et à se concerter afin que chacun s'enrichisse et enrichisse le sujet en évitant les doublons et travaux « solitaires ». Ainsi certains ont pu travailler spécifiquement sur les aspects sociétaux, bibliographiques et structurels (Augier, 2017), sur les aspects localisations et caractérisations avec l'outil ARCGIS (Cirou, 2017), ou sur les aspects localisations et caractérisation avec l'outil QGIS (Salpin, 2016) sur l'aspect hiérarchisation (SYLOA, 2017/2018). L'EPTB de la Sèvre Nantaise a dirigé ses travaux vers l'automatisation des traitements de correction et de préparation des données ainsi que de délimitation, caractérisation et hiérarchisation des TBV. L'EPTB Vilaine (Guégniard, 2018) s'occupant de réaliser des mesures de terrains afin d'évaluer les corrélations entre les indices issus de traitement SIG et les mêmes indices issus d'un protocole de relevé terrain. Une rencontre réunissant l'EPTB Sèvre Nantaise, le SYLOA (syndicat mixte en charge de l'estuaire de la Loire), l'EPTB Vilaine et le Forum des marais nous a permis d'échanger sur les aspects purement techniques. La participation à une réunion de présentation organisée par l'EPTB Vilaine sur les travaux en cours sur leur bassin a été l'occasion d'échanger de manière plus générale avec l'ensemble des partenaires techniques, financiers, réglementaires sur le sujet mais également de présenter les premiers résultats des travaux de recherches de corrélation entre indicateurs SIG et terrain. Ces échanges se sont avérés instructifs pour la suite de notre étude.

Plusieurs structures voisines se sont fait connaître suite à l'information diffusée en début de stage et se disent en attente des résultats et techniques utilisées pour les reproduire sur leur territoire.

Ce travail d'animation et de concertation est une constante générale dans le domaine de l'eau. Cette ressource touchant de fait à de multiples problématiques. Il est à noter que c'est d'ailleurs l'un des sens donné par les agences de l'eau et l'AFB (cf. entretien avec M. Le Bihan). L'étude des têtes de bassin versant a pour fonction de participer à une approche transversale et de réunir autour d'un même périmètre de travail, les sylviculteurs, agriculteurs, urbaniste, aménageurs, animateurs de bassin, techniciens rivières, etc.

En effet la qualité de l'eau dépend de son utilisation et de l'occupation du sol qu'elle parcourt, les objectifs d'atteinte de bon état doivent donc être partagé par tous les acteurs.

Fort de ces échanges, les étapes suivantes ont été identifiées :

- Localisation des têtes de bassin versant (*délimiter des ensembles de ruisseaux constituant des TBV*)
 - Préparation du MNT
 - Préparation du référentiel hydrographique
 - Ordination des cours d'eau selon le rang de Strahler
 - Création des bassins versants
- Caractérisation des têtes de bassin versant (*dresser la carte d'identité des TBV*)
- Hiérarchisation des têtes de bassin versant (*ordonner les TBV en fonction de règles de pondération*)

7.2 Choix des technologies

Comme le montre la bibliographie existante sur le sujet, de nombreux outils permettent de réaliser les traitements nécessaires à la correction ou à la création de référentiels hydrographique ainsi qu'à l'étude sur les TBV dans leur ensemble.

Pour cette étude le choix c'est porté sur des outils open source réputés pour leur performance et leur fiabilité.

- **GRASS Gis** pour les séries de traitements de localisation des TBV
- **Postgres / Postgis** pour gérer, stocker les données mais également pour toute une série de traitements complexes visant à la fois la correction et la génération de référentiels tout autant que pour les analyses spatiales que constituent la caractérisation et la hiérarchisation des TBV
- Le **Web** et en particulier le langage **PHP** pour l'interface de traitement.
- **R et Rstudio** pour l'analyse statistique.
- Libre Office pour le rapport.

GRASS comme R ont été installés sur le serveur distant de l'EPTB afin d'interagir avec la base de donnée Postgres (elle aussi présente sur le serveur distant). L'alliance GRASS / Postgres est interfacée sur des pages web en **PHP** ayant pour résultat une interaction ergonomique et surtout une capacité de traitement basé sur celle du serveur (64 go RAM) et non sur les machines locales de la structure. L'utilisateur peu ainsi réaliser l'ensemble des traitements depuis une machine performante et un environnement unique.

Cette démarche d'utilisation d'outils open-source s'inscrit dans une démarche globale portée par le pôle analyse territoriale et systèmes d'informations de l'EPTB. L'ouverture des codes sources des outils développés en interne fait partie des objectifs à court terme de l'équipe. La philosophie sous-jacente porte les valeurs du partage et de la mise en commun des outils publics afin que chacun s'enrichisse de l'autre tout en l'enrichissant.

7.2.1 GRASS GIS, tout en puissance

Le choix a été fait de recourir à la technologie **GRASS**. GRASS GIS est un logiciel de SIG relativement atypique, reconnu pour sa robustesse et sa fiabilité. L'origine du projet remonte à 1982 et a été institué par différentes agences gouvernementales et universités américaines, le cœur de l'application étant développé par l'armée américaine

elle-même. L'objectif initial était d'aider à l'aménagement du territoire. Sans rentrer dans les détails techniques, on peut préciser que le développement est continu depuis 1982, la dernière version stable en date étant la 7.4 (datant du 1^{er} février 2018). Le logiciel est open source (licence GPL), libre et construit sur une base modulaire, ce qui signifie que chaque fonction est un module à part entière. Si GRASS offre la possibilité de travailler avec des vecteurs, sa spécialité principale reste les rasters.

Ses fonctionnalités, sont proches de celles de Spatial Analyst ou 3D Analyst d'Arcgis, son gros point positif étant qu'il est ouvert, gratuit et tout aussi efficace. En revanche sa prise en main peu demander une phase d'adaptation, car si une partie de sa boîte à outils est disponible via Qgis, pour profiter pleinement de toutes les fonctions il est nécessaire de passer directement par sa stand-alone ou sa console.

GRASS diffère de la plupart des autres SIG dans le fait qu'il est nécessaire de constituer une base de données propre pour travailler différents fichiers. Il n'est pas possible de glisser déposer un .tif ou un .shp à la volée par exemple. Toutefois une fois ce pré-requis intégré, il devient d'une grande efficacité. Pour aider à la compréhension, un tutoriel en français est disponible sur le site [portail SIG](http://portailsig.org)¹¹ (bien que datant de 2012, l'article reste pertinent pour appréhender l'outil). Les modules complémentaires ajoutés au fur et à mesure sont parfaitement décrits dans la documentation de GRASS et permettent de rester à jour.

Les deux éléments à retenir pour démarrer avec GRASS sont la 'LOCATION' (secteur) et le 'MAPSET' (jeu de cartes). La LOCATION correspond à la zone géographique de travail et à un système de projection (SRID) particulier, on pourra donc changer de LOCATION au choix en changeant de projet ou en changeant de zone géographique d'étude. Le MAPSET correspond lui au jeu de données. Par défaut un MAPSET PERMANENT est disponible à la création d'une LOCATION.

Une fois la LOCATION créée et le MAPSET choisi, il est possible de travailler via l'interface graphique ou par la console. Dans les pages suivantes, les lignes de commandes permettant d'appeler les fonctions via la console de GRASS figurent sous **cette forme** (ces lignes de commandes commençant par le nom du module il sera facile, le cas échéant, de les retrouver dans l'interface graphique).

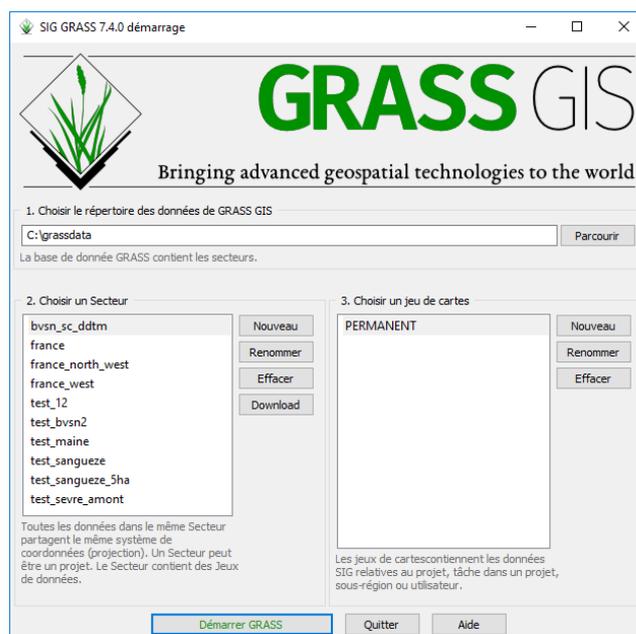
Attention ! Les exemples devront être adaptés à votre LOCATION et vos MAPSET.

GRASS GIS 7.4 (ou antérieures) peut être téléchargé en stand-alone sur [cette page](#)¹².

7.2.1.1 Démarrage de GRASS

Commencer par lancer l'application, une page de démarrage découpée en 3 parties s'affiche :

1. Choix du répertoire de données
 - C'est la localisation du répertoire d'écriture dans l'arborescence système.
2. Choix du secteur
 - Le point important qui distingue le secteur est son système de projection unique. Attention à bien le choisir. On peut estimer qu'un secteur est un projet.
3. Choix du jeu de cartes



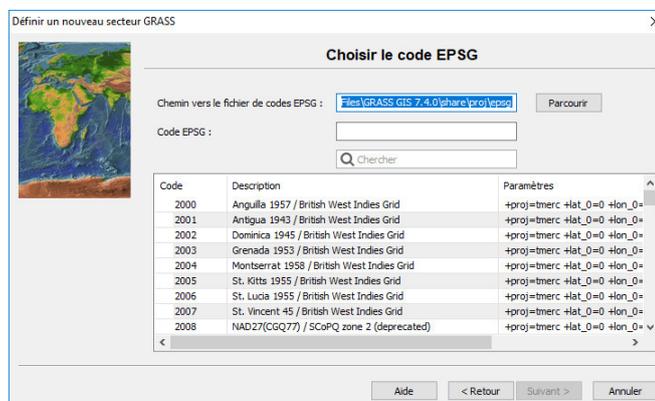
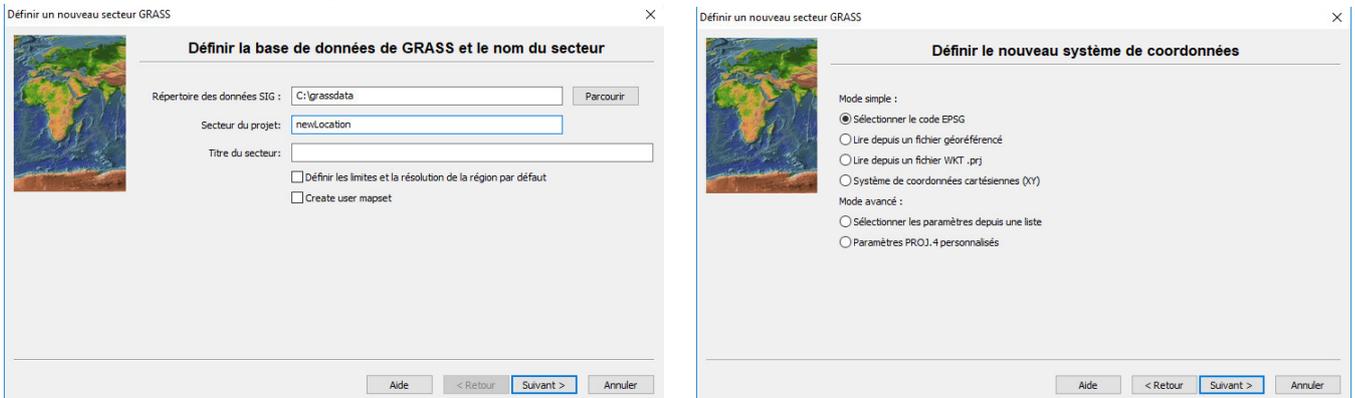
11 <http://www.portailsig.org/content/grass-gis-pas-pas-pour-les-debutants-1-demarrage-de-l-application-secteurs-locations-jeux-de>

12 <https://grass.osgeo.org/download/>

- Le jeu de cartes contient les données, on peut se servir du PERMANENT par défaut ou en créer au besoin. Une convention attribut à PERMANENT le jeu de données d'origine à conserver intact. L'utilisation reste libre.

Les premières étapes d'un travail avec GRASS consistent donc à attribuer un répertoire d'écriture, puis à choisir un système de projection (donc une zone géographique de travail), et enfin à définir des répertoires de stockage de données.

Pour la définition du secteur, après l'avoir nommé, l'idéal est de définir soit même le code EPSG comme suit :



7.2.1.2 Les extensions GRASS

Les extensions sont des modules complémentaires qui sont téléchargeables très simplement et permettent d'ajouter des fonctions au logiciel de base. [r.stream.order](#) et [r.stream.basins](#) en font parties. L'ensemble des modules complémentaires sont disponibles [ici](#)¹³.

7.2.1.3 Installation d'une extension

La fonction [g.extension](#) permet d'installer simplement une extension. Il est possible d'installer une extension dans l'interface graphique de GRASS ou via la console. La fonction télécharge et installe l'extension appelée en paramètre, à la manière d'un 'APT GET' sur Linux.

13 <https://grass.osgeo.org/grass74/manuals/addons/>

Exemple d'appel de la fonction g.extension dans la console GRASS

```
g.extension extension=r.stream.order
```

7.2.1.4 Les extensions "Stream"

Il existe une série d'extensions "[stream](#)" qui regroupe plusieurs fonctions d'ordre hydrographique.

7.2.2 Postgres / Postgis, la force du SQL

PostgreSQL est un logiciel de gestion de base de données relationnelle et objets. Son origine trouve sa source à l'université de Berkeley aux États-Unis en 1986. C'est aujourd'hui La base donnée libre de référence. Pour cette étude nous avons utilisé PostgreSQL 9.6 avec pgAdmin 3 comme outil d'administration graphique.



Postgres utilise le langage SQL et permet une programmation procédurale via le PL/pgSQL. Outre ces capacités de stockage, traitement (requêtes complexes) et d'administration d'une grande quantité de données, il donne accès à :

- De nouveaux types de donnés
- De nouvelles fonctions et fonctions d'agrégats
- De nouveaux opérateurs
- De nouvelles méthodes d'indexation
- ...

Ainsi, de nombreuses extensions sont disponibles pour ce SGBD, à l'image de Postgis, sa cartouche spatiale, permettant l'interaction avec des objets géographique.

Ainsi nous l'utiliserons en matière de correction topologique, de création de référentiel, pour l'ensemble des traitements lié à l'analyse spatiale et bien sûr pour stocker nos données.

La puissance du couple Pstgres / Postgis en fait aujourd'hui un maillon indispensable pour un travail traitant des données spatiales. La fiabilité, la grande flexibilité et les possibilités d'optimisation permises par ces outils sont sans aucune mesure avec les logiciels de SIG classiques comme Qgis ou Arcgis pour la réalisation de traitements géographiques. Enfin, il s'agit de logiciels open-source accessibles pour tous, gratuitement et bénéficiant de constantes contributions de la part de leurs communautés de développeurs.

Pour les besoins de cette étude, Postgis 2.4 sera nécessaire et de nombreuses fonctions seront développées en interne pour aboutir aux résultats souhaités.

Il est possible de télécharger le package Postgres de BigSQL [ici](#)¹⁴ et Postgis [ici](#)¹⁵. L'aide en ligne en français est disponible [ici](#)¹⁶. Pour plus d'informations sur l'installation référez-vous au chapitre sur cette dernière.

14 <https://www.openscg.com/bigsql/pgadmin3/>

15 <https://postgis.net/install/>

16 <https://docs.postgresql.fr/9.6/>

7.2.3 L'interface Web, l'outil universel

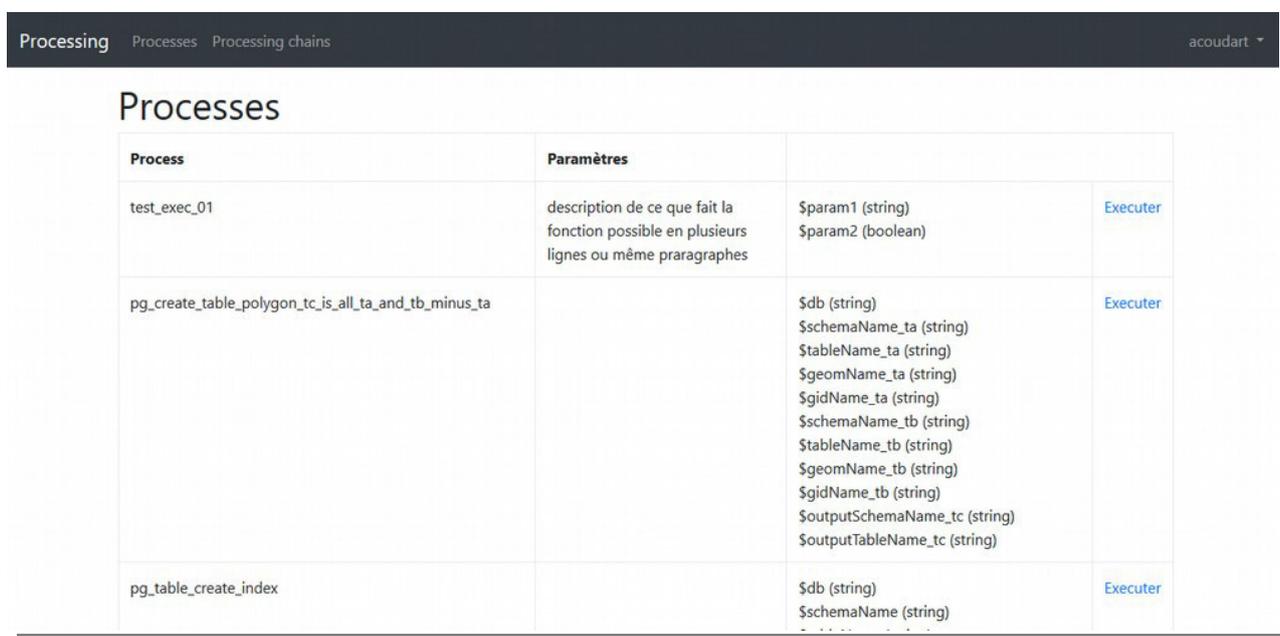
La commande initiale de l'EPTB précisait également l'élaboration d'un site Web pour lier l'ensemble de ces outils en une même interface et utilisable simplement via un navigateur. L'idée générale est de proposer à l'utilisateur un environnement ergonomique unique, évitant ainsi les allers-retours entre différents logiciels. Cette interface doit ainsi permettre de lister l'ensemble des commandes informatiques permettant de mener à bien les étapes de délimitation et caractérisation des têtes de bassin, de pouvoir les exécuter, voire de les chaîner pour automatiser ces traitements.

Le langage de programmation **PHP** a été choisi car il est aujourd'hui le principal langage utilisé pour les outils web développés ou utilisé par l'EPTB (outil de suivi des niveaux d'eau, SYSMA [le Websig de gestion des milieux aquatiques], la photothèque, la base contact, etc). PHP sert à l'animation dynamique des pages HTML, au couplage avec Postgres (lancement des requêtes, récupération des données de la base, utilisation immédiate ou ultérieure) et peut également exécuter des lignes de commandes (GRASS ou GDAL dans notre cas). Il est le langage de programmation le plus utilisé dans le monde avec 82 % des parts de marché des sites Web en 2016 (source Wikipédia).

L'application développée par l'équipe de l'EPTB sera, à terme, publiée en open source. Toutes les commandes GRASS, GDAL, SQL nécessaires à la délimitation et caractérisation des têtes de bassin ont été implantées dans le service. Une page « Processes » les liste dans leur intégralité. Si l'interface est aujourd'hui peu ergonomique, cela sera amené à évoluer avec le temps. Il est important de préciser que l'outil n'a pas pour seul objectif de remplir cette fonction autour des TBV, il sera amené à devenir l'interface privilégiée pour tout un ensemble de traitements informatiques nécessaires au fonctionnement de l'observatoire du bassin de la Sèvre Nantaise. Par exemple, le téléchargement des flux IGN et leur intégration dans une table en fonction de paramètres comme l'enveloppe du territoire, est aujourd'hui une action directement réalisée à partir de la page Web sans devoir passer par pgAdmin. Son identité visuelle sera alors revue plus tard, l'objectif principal étant les fonctionnalités techniques de l'outil.

7.3 Les fonctions unitaires dans Processes

La page « Processes » de l'outil liste les différents traitements exécutables depuis l'interface. Chaque « process » correspond à un fichier contenant sous forme de fonction PHP le code source et la déclaration des variables nécessaires à la réalisation de ce traitement.



Process	Paramètres	
test_exec_01	description de ce que fait la fonction possible en plusieurs lignes ou même paragraphes	\$param1 (string) \$param2 (boolean) Executer
pg_create_table_polygon_tc_is_all_ta_and_tb_minus_ta		\$db (string) \$schemaName_ta (string) \$tableName_ta (string) \$geomName_ta (string) \$gidName_ta (string) \$schemaName_tb (string) \$tableName_tb (string) \$geomName_tb (string) \$gidName_tb (string) \$outputSchemaName_tc (string) \$outputTableName_tc (string) Executer
pg_table_create_index		\$db (string) \$schemaName (string) Executer

Lorsque l'utilisateur clic sur « Exécuter » il bascule sur la page de paramétrage de la fonction choisie (sur l'exemple ci-dessous, l'utilisateur peu choisir différents paramètres comme la LOCATION, le MAPSET, son référentiel raster en entrée, son référentiel cours d'eau en entrée et le nom du raster en sortie) :

Processing Processes Processing chains

Process grass_r_stream_basins by unknown

/var/www/observ/processing/src/App/Processes/Generics/grass/grass_r_stream_basins.php

grassUser	<input type="text" value="observ"/>
grassMemoryMB	<input type="text" value="6000"/>
grassDataDir	<input type="text" value="/var/www/observ/out_processing/grass_data"/>
grassLocation	<input type="text" value="test_13"/>
grassMapset	<input type="text" value="PERMANENT"/>
grassInputRasterDirectionName	<input type="text" value="MNT_sinkless_direction"/>
grassInputRasterStreamCategoriseName	<input type="text" value="sel_strahler_cat_2"/>
grassOutputRasterBasinsName	<input type="text" value="MNT_sinkless_basins"/>

Côté programmation, chaque élément de la page est déclaré dans une fonction PHP organisée de manière à générer automatiquement les pages de traitement.

- Exemple d'une fonction GRASS passée en PHP.
 - (cette fonction, « *grass_r_stream_basins* », correspond au module GRASS *r.stream.basins*).
1. Dans un premier temps on déclare le nom de la fonction
 2. Dans un second temps on déclare toutes les variables qui seront appelées et qui sont paramétrables par l'utilisateur. Les paramètres sont eux-mêmes composés de plusieurs variables comme la valeur par défaut qui s'affiche automatiquement à l'affiche de la page, ou une description (non visible sur cette fonction) qui permet de savoir à quoi correspond le paramètre dans un langage plus verbeux.
 3. Le fichier contient ensuite la fonction PHP qui réalise la préparation de la commande à exécuter. Dans ce cas, il s'agit d'une simple ligne de commande GRASS et de ses différents paramétrages (cette commande peut-être une commande SQL, un appel de commande GDAL, etc)
 4. Ensuite la commande préparée est renvoyée puis exécutée par l'application qui enregistre automatiquement les conditions et résultats d'exécution (durée du traitement, variables utilisées, échec ou réussite du traitement) et affiche le résultat renvoyé à l'issue de l'exécution.

```

1 <?php
2
3 /**
4  * grass_r_stream_basins
5  *
6  * @param string $grassUser TextType DEFAULT=GRASS_USER
7  * @param string $grassMemoryMB TextType DEFAULT=GRASS_MEMORY
8  * @param string $grassDataDir TextType DEFAULT='/var/www/observ/out_processing/grass_data'
9  * @param string $grassLocation TextType DEFAULT='test_13'
10 * @param string $grassMapset TextType DEFAULT='PERMANENT'
11 * @param string $grassInputRasterDirectionName TextType DEFAULT='MNT_sinkless_direction'
12 * @param string $grassInputRasterStreamCategoriseName TextType DEFAULT='sel_strahler_cat_2'
13 * @param string $grassOutputRasterBasinsName TextType DEFAULT='MNT_sinkless_basins'
14 *
15 */
16 function grass_r_stream_basins($grassUser,$grassMemoryMB, $grassDataDir, $grassLocation, $grassMapset, $grassI
17 // $process_uuid=printf("uniqid(): %s\r\n", uniqid());
18
19
20 $command = GRASS_EXEC . ' ' . $grassDataDir . '/' . $grassLocation . '/' . $grassMapset . ' --exec' .
21 ' r.stream.basins dir=' . $grassInputRasterDirectionName . '@' . $grassMapset . ' stream=' $grass
22 ' basins=' . $grassOutputRasterBasinsName .
23 ' memory=' . $grassMemoryMB .
24 ' --overwrite';
25
26 $command = '' . $command . '';
27 $command = 'sudo /bin/su - ' . $grassUser . ' -c ' . $command;
28
29 // echo '<BR>' . $command . '<BR>';
30
31 return array('command' => $command, 'type' => 'exec');
32 }
33

```

Illustration 5: Exemple d'une fonction Grass passé en PHP

7.4 Les fonctions en chaîne dans Processing chains

Les fonctions unitaires sont également mobilisables comme des ensembles de traitement. L'avantage est qu'il est ainsi possible d'automatiser une tâche comme la caractérisation des têtes de bassin versant qui comme nous le verrons calcul un ensemble d'une trentaine d'indicateurs. Ceux-ci sont donc jouable en un clic grâce à la fonction ci-dessous **indicateurs_caractérisation_tbv**. Pour cela on stocke notre donnée dans une table pg listant des chaînes reliées à une table pg listant des process. La table « chaîne » possède un identifiant que l'on retrouve comme clé étrangère dans la table de « process ». Cette logique sera utilisée plus tard pour le modèle de donnée de caractérisation des têtes de bassin versant.

1. On déclare une chaîne dans un table Postgres dédiée :

id [PK] serial	pc_name character varying	create_at timestamp without time zone	modified_at timestamp without time zone	created_by character varying	modified_by character varying
1	test	2018-05-29 10:42:57.90873		srenou	
2	hydro_strahler_rank_from_rast_mnt	2018-05-29 13:06:58.799916		srenou	
3	hydro_mnt_surcoureusement_from_pg_hydro_vect	2018-06-11 17:31:10.831109		srenou	
4	recup_flux_par_lot	2018-06-15 11:48:31.678439		ariviere	
5	indice_pression_tbv	2018-08-08 17:24:56.997907		acoudart	
8	indicateurs_caracterisation_tbv	2018-08-08 17:29:31.678848		acoudart	

2. On déclare les fonctions unitaires (process) dans une table dédiée avec un champ ordre [...de traitement] (order) et un champ chaîne [clé étrangère de la table « chaîne »](id_pc), et un champs pointant le chemin du process :

	id [PK] integer	id_pc integer	order integer	data character varying	id_process character varying
23	23	8	1		/var/www/observ/processing/src/App/Processes/Tr_TBV/Caracterisation/tbv_surface.php
24	24	8	2		/var/www/observ/processing/src/App/Processes/Tr_TBV/Caracterisation/tbv_pente_moy.php
25	25	8	3		/var/www/observ/processing/src/App/Processes/Tr_TBV/Caracterisation/tbv_pente_moy_longest_path.php
26	26	8	4		/var/www/observ/processing/src/App/Processes/Tr_TBV/Caracterisation/tbv_gravelius.php
27	27	8	5		/var/www/observ/processing/src/App/Processes/Tr_TBV/Caracterisation/tbv_passini.php
28	28	8	6		/var/www/observ/processing/src/App/Processes/Tr_TBV/Caracterisation/tbv_densite_ce.php
29	29	8	7		/var/www/observ/processing/src/App/Processes/Tr_TBV/Caracterisation/tbv_densite_point_bas.php
30	30	8	8		/var/www/observ/processing/src/App/Processes/Tr_TBV/Caracterisation/tbv_position_in_reseau_hydro.php
31	31	8	9		/var/www/observ/processing/src/App/Processes/Tr_TBV/Caracterisation/tbv_taux_plan_eau_impact_ce.php
32	32	8	10		/var/www/observ/processing/src/App/Processes/Tr_TBV/Caracterisation/tbv_taux_zh_connect_ce.php

3. Les chaînes de process sont affichées dynamiquement sur la page HTML « Processing chain ».

4. L'utilisateur choisit la chaîne de son choix ici **indicateurs_caracterisation_tbv** :

- L'ordre d'apparition des process suit l'ordre attribué dans la table des process unitaires.
- Il est possible de commencer le traitement à partir d'une étape donnée (dans le cas de l'échec de l'une des fonctions par exemple).
- Une option permet de modifier automatiquement tous les paramètres portant le même nom (par exemple le nom du schéma de stockage des résultats).

Les fonctions PHP ne sont pas fournies en annexes. Toutefois, l'EPTB de la Sèvre Nantaise étant dans une démarche d'ouverture du code de l'ensemble de ces outils, des informations complémentaires peuvent être demandées à la structure (URL).

7.4.1 R et Rstudio, l'outil statistique à la portée de tous

Dans notre étude, nous générerons une série d'indicateurs de caractérisation associés à chaque tête de bassin. Les valeurs de ces indicateurs seront pondérées selon des scénarios de calcul, puis associées afin d'aboutir à une note globale de sensibilité, d'enjeu ou de vulnérabilité à l'échelle de chaque tête de bassin. Ainsi, l'enjeu qualité d'eau additionnera entre autre, la densité de haie dans la TBV, la densité de zones humides dans la TBV, le volume de prélèvement dans la TBV.

Chacun de ces indicateurs est d'une nature différente (linéaire, superficie, volume, etc) Il n'est donc pas possible d'assembler les valeurs telles-queelles. Il nous sera donc nécessaire de passer par une étape de traitement statistique afin de pouvoir agglomérer les données entre elles.

Pour nous aider à analyser nos données et ainsi faire les choix qui s'imposent en matière de traitement de l'information nous avons décidé d'utiliser **le logiciel R** et son environnement de développement **Rstudio**. Encore une fois, le choix inclut une dimension libre et open-source, mais également la recherche de performance. R est un logiciel qui a fait ses preuves, qui possède l'avantage d'une librairies de plus 15 000 extensions et est (comme les extensions) accessible gratuitement.

Le choix a également été guidé par sa capacité à se connecter à une base de donnée Postgres, par sa simplicité d'utilisation, par la qualité de visualisation des données et par la possibilité d'automatiser certaines tâches grâce à un langage de programmation spécifique nommé également R. Enfin il pourrait intégrer la suite logiciel présente sur le serveur distant.

7.4.2 Le Wiki, la documentation interne

Enfin, l'ensemble de l'étude est documentée dans un wiki interne, à l'EPTB basé sur l'outil open source Wikimedia (moteur de Wikipédia). L'usage de cet outil est généralisé au sein de l'EPTB, une grande partie des procédures, méthodologies, tutoriels sont ainsi disponibles pour tous les agents.

Dans une démarche d'ouverture, les pages consacrées à la présente étude sont rendues accessibles au public. Là encore, la démarche s'opère dans l'esprit de l'open source et du partage d'expérience. Les pages ont été alimentées régulièrement au fur et à mesure de l'avancée des opérations. Ainsi tout au long du stage, il nous a été possible de communiquer avec nos partenaires sur nos solutions techniques via ce procédé.

Le wiki public est accessible via ce [lien](#)¹⁷.



17 <http://observatoire.sevre-nantaise.com/pages-techniques/?page=Localisation%20et%20caract%C3%A9risation%20des%20t%C3%AAtes%20de%20bassin%20versant>

8 Localisation des têtes de bassin versant

Localiser les TBV signifie définir leur périmètre, leur(s) point(s) de départ et leur exutoire. Pour rappel, selon la directive cadre sur l'eau, un bassin versant est l'espace drainé par un cours d'eau et ses affluents. L'ensemble des eaux qui tombent dans cet espace convergent vers un même point de sortie appelé exutoire, cours d'eau, lac, mer, océan, etc.

La méthode qui suit utilise les outils GRASS et Postgres/Postgis.

8.1 Référentiels hydrographiques disponibles

Le choix du référentiel hydrographique est une donnée majeure dans le travail d'étude des têtes de bassin versant car de fait, il conditionne leur localisation et le travail de recoupement de données qui en découle. Son choix est donc, à juste titre, la première pierre à poser pour édifier une cartographie cohérente des têtes de bassin versant. Si l'ensemble de l'étude demande à s'interroger sur les données, celle-ci est particulièrement majeure.

Les référentiels disponibles :

- Bd Carthage
- Bd Topo V2.1
- Bd Topo 151
- Inventaire DDTM
- Tracé(s) théorique(s)

8.1.1 Tableau comparatif du linéaire hydrographique en fonction des référentiels

Chaque référentiel possède ses points forts et points faibles. L'échelle de numérisation et les objectifs n'étant pas les mêmes, le linéaire visible est très disparate. Le comparatif s'appuie sur les données du bassin versant de la Sèvre Nantaise avec un zoom sur le département 49, département où l'inventaire des cours d'eau a été réalisé, livré et validé :

Référentiel	Zone	Linéaire de cours d'eau en km
Carthage	BVSN	2 313
Flux IGN troncon cours deau bd_topo	BVSN	3 377
IGN v2.1	BVSN	2 662
Tracé théorique CE (Moreau, 2016)	BVSN	19 953
Tracé théorique CE (Coudart, 2018)	BVSN	4 339
Carthage	BVSN 49	470
Flux IGN troncon cours deau bd_topo	BVSN 49	725
IGN v2.1	BVSN 49	773
DDT 49	BVSN 49	606
Tracé théorique CE (Moreau, 2016)	BVSN 49	2 869
Tracé théorique CE (Coudart, 2018)	BVSN 49	855

Illustration 6: Tableau de comparatif des différents référentiels hydrographiques disponibles

Décryptage du tableau :

La **Bd Carthage** est une donnée réalisée à petite échelle sur tout le territoire français (Validité au 1:100 000), à l'échelle locale sa fiabilité est donc toute relative, c'est pourquoi c'est le référentiel qui nous donne le plus petit niveau de linéaire de CE (Cours d'Eau).

La différence entre les 2 **tracés théoriques** s'explique par une différence notable entre les paramètres utilisés. Pour le travail de Moreau, 2016, aucune valeur d'accumulation dans les cellules n'avait été retenue. Pour le travail de Coudart, 2018, une valeur minimale de cellule d'accumulation de 4 000 a été retenue. Cette valeur équivaut à 10 hectares sur le terrain. Au-dessous, il est considéré que la puissance spécifique n'est pas à même de structurer un cours d'eau (Cirou, 2017).

Le flux de la **Bd topo**, tronçon cours d'eau, est la donnée la plus récente du référentiel IGN à grande échelle. Le référentiel qui a été fourni aux services de l'État dans le cadre de la démarche d'inventaire des cours d'eau (la version **Bd topo 151**) se rapprochait plus de la donnée actuelle (flux wms) que de la **version 2.1** qui étonnamment montre une différence notable de presque un tiers de linéaire cumulé à l'échelle du bassin quand cette donnée est proche des autres référentiels à l'échelle du département. À titre de comparaison le calcul sur la version 151 nous donnait 3387 km de cours d'eau.

Ce comparatif nous indique que le choix de la donnée de départ aura une incidence majeure sur les résultats obtenus. Entre un référentiel maximal comptant presque 20 000 km de cours d'eau et un minimal en compte lui à peine 2000 on comprend aisément que la cartographie des têtes de bassin versant ne dira pas la même chose. Le choix par le nombre n'est pour autant pas forcément le meilleur choix.

Nous avons décidé rapidement d'écarter Carthage de l'étude, ce référentiel ne nous permettant pas de travail à l'échelle micro nécessaire. Le référentiel de l'IGN à l'avantage d'être une donnée topologiquement juste dans une grande majorité des cas, mais les études montre qu'elle tend à « oublier » une part importante du réseau de rang 1 (Meyer et al., 2003 – Lhéritier, 2012 - Van Nguyen, 2012).

Les tracés théoriques ont comme avantage de simuler parfaitement les écoulements de surface en fonction d'un modèle numérique de terrain. Toutefois plusieurs biais le discrédite, d'une part il s'agit d'un modèle qui ne prend pas en compte les aménagements anthropiques parfois vieux de plusieurs siècles ayant pu conduire à une modification de la localisation des cours d'eau. D'autre part il s'appuie sur une donnée de terrain ne distinguant pas une digue, d'une route ou d'une élévation naturelle. Ainsi, des traversées d'aménagement humain (route, voie ferré, etc) peuvent se voir contrarier et renvoyer un écoulement le long de la route quand dans la réalité il la traverse ou l'inverse.

Sur l'image ci-contre, le réseau théorique généré ne tient pas compte de la route, en la traversant, alors que sur le terrain son tracé est contrarié et tend à bifurquer au sud. À l'ouest de la route un tracé est généré là où les écoulements n'ont pas encore eu la puissance nécessaire à la création d'un lit ou bien là où un lit a pu être enterré. Donc si les tracés théoriques nous apportent incontestablement des informations elles sont néanmoins à prendre avec précaution.

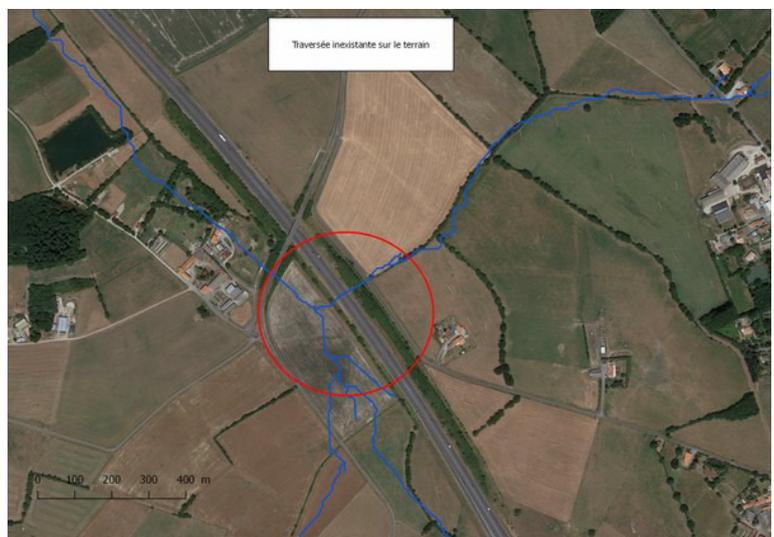


Illustration 7: Incohérence de tracé théorique

Les données fournies par les services de l'État ont vocation à devenir référentiels juridiques sur lesquels s'applique la loi. Dans la plupart des cas, ils sont également validés par les commissions locales de l'eau. Ces données sont en revanche très hétérogènes quant à leur qualité de forme ou de fond. En effet les moyens mis en œuvre peuvent être différents d'un département à l'autre. À l'usage, le constat a été fait que les zones limitrophes interdépartementales montrent des incohérences (chevauchement, données non jointives), les règles topologiques ne sont pas toujours respectées.

Malgré tout, dans un souci de cohérence institutionnelle et par élimination, **le choix du référentiel s'est porté sur les inventaires réalisés par les services de l'État lorsque celui-ci est disponible et sur le flux IGN dans le cas contraire.**

En août 2018, pour le bassin de la Sèvre Nantaise, seuls les inventaires de la Loire-Atlantique et du Maine-et-Loire sont validés. La démarche est en cours en Vendée et non réalisées en Deux-Sèvres. De plus, quel que soit le référentiel, IGN comme DDT, il arrive que certains tronçons soit tracés mais non raccordés à l'ensemble hydrographique rendant leur ordination impossible. Pour toutes ces raisons il s'avère nécessaire de créer une donnée composite.

8.2 *Création d'un référentiel hydrographique composite*

8.2.1 Assemblage des couches DDTM et IGN

Dans un premier temps on cherche à regrouper l'ensemble des données disponibles, on récupère donc les données des différentes DDTM et de l'IGN. Sans se soucier des tronçons orphelins.

L'assemblage s'effectue en deux étapes :

- Constitution d'une couche DDTM (r020_territoire_hydrographie.t_cours_eau_police_de_l_eau_ddtm_v02) par insertion sélective (et découpage) par département des données DDTM. Des règles de priorité ont été convenues vis-à-vis de la qualité de la donnée, la couche du département 44 ayant été jugée de meilleure qualité elle est prioritaire sur celle du 49, les tampons servent à la sélection.
1. Insertion des données DDTM dans une table DDTM :
 - Buffer – 10 m = coverage49 (sur le département) pour l'import des données DDTM 49 (linéaire de cours d'eau retenus) avec découpage des géométries pour ne conserver que celles qui sont dans le buffer.
 - Buffer + 10 m =coverage44 (sur le département) pour l'import des données DDTM 44 (linéaire de cours d'eau retenus) avec découpage des géométries pour ne conserver que celles qui sont dans le buffer.
 - 85 et 79 pas d'import, données respectivement trop partielles et inexistantes.
 2. Insertion des données BDTOPO IGN (cours d'eau intermittents et permanents) dans une table DDTM + IGN :
 - Import des données BDTOPO IGN hors du coverageDDTM (= coverage49+coverage44) avec découpage des géométries pour ne conserver que celles qui sont en dehors du coverageDDTM.

Le script de création et peuplement des tables n'est pas disponible, il est trop spécifique au territoire de la Sèvre Nantaise et au modèle de donnée de la base interne.

En revanche veiller à ce que le référentiel utilisé respecte cette structure afin d'être utilisable dans le reste des opérations (il est bien sur possible aussi d'utiliser d'autre structures de table avec les fonctions de corrections) :

- gid TYPE : serial primary key
- geom TYPE : geometry(linestring,2154)
- data_source_producteur TYPE : character varying(255)
- ref_id_liste_table TYPE : bigint
- ref_uniq_id TYPE : character varying(255)
- ref_data_quality TYPE : NUMERIC

8.2.2 Correction de la table composite

Cette table regroupant les données disponibles étant maintenant disponible il nous faut la rendre exploitable par les systèmes d'informations. Les problèmes qui restent à régler sont multiples :

- Connexion des tronçons orphelins au réseau hydrologique
- Orientation amont / aval des tronçons
- Ordination des tronçons selon la méthode de Strahler

Pour connecter les tronçons isolés nous avons choisi de combler les vides entre ceux-ci et le réseau général avec le tracé théorique des écoulements. Pour se faire, nous avons forcé le réseau théorique à emprunter le chemin réel de l'eau en surcreusant le MNT avec le référentiel DDT / IGN non corrigé.



Illustration 9: Réseau hydro réel VS réseau théorique brut



Illustration 8: Réseau hydro réel VS réseau théorique surcreusé

Un réseau de drainage théorique basé uniquement sur le MNT emprunte le chemin du point le plus bas. Dans la réalité les aménagements successifs ont profondément modifié le chemin de l'eau. Sur l'image de gauche on constate de nombreuses variations entre le réel (en bleu) et le théorique (en orange), alors que sur l'image de droite, le tracé théorique surcreusé ou forcé (en pointillés jaunes) emprunte le tracé réel (en bleu) lorsque celui-ci est disponible.

Le module [r.mapcalc](#) sera utilisé pour la partie calcul de creusement et de surélévation, [v.to.rast](#) sera utilisé pour transformer le vecteur de cours d'eau en raster, [r.null](#) sera utilisé pour transformer les valeurs NULL du raster en 0 afin de ne pas interférer dans le creusement ultérieur.

Attention ! Si l'étape de "lissage" des dépressions est indispensable, l'étape de creusement est tout à fait optionnel. Dans la présente étude, son utilité réside dans son utilisation ultérieure. Ce réseau théorique forcé avec le réseau DDT et IGN nous permettra, comme nous le verrons plus loin, de combler les zones de déconnexion du référentiel IGN. De même l'étape de surélévation avant l'étape de creusement permet dans des zones proches du niveau de la mer de ne pas générer des valeurs négatives lors du surcreusement. Sur le territoire de la Sèvre Nantaise, cette étape est indispensable au surcreusement.

8.2.2.1 Préparation du MNT

- Import du vecteur cours d'eau de Postgres vers GRASS :

```
v.in.ogr input='PG:dbname=ma_database_pg host=mon_hote_pg port=mon_port
user=mon_user_pg password=mon_password_pg'
layer=mon_schema.vecteur_cours_deau_pg output=vecteur_cours_deau type=line
geometry=geom --overwrite
```

- Transformation du vecteur cours d'eau en raster (inscription de la valeur 50 mètres en dur dans les données du raster grâce à l'option "value=50") :

```
v.to.rast input="vecteur_cours_deau@PERMANENT" layer= vecteur_cours_deau
type="line" output="raster_cours_deau" use="val" value=50
```

- Transformation des valeurs NULL du raster cours d'eau :

```
r.null map="raster_cours_deau@PERMANENT" null=0
```

- Surélévation du MNT brut (pour éviter les valeurs négatives) :

```
r.mapcalc expression="mnt_sureleve = mnt_brut@PERMANENT + 50" --overwrite
```

- Creusement du MNT surélevé par rapport au référentiel hydrographique (-50m) :

```
r.mapcalc expression="mnt_creuse = mnt_sureleve@PERMANENT -
raster_cours_deau@PERMANENT" --overwrite
```

8.2.2.2 Comblement des dépressions

Pour rappel le MNT est un raster construit par interpolation des points composant le nuage de points originel, ce sont donc des algorithmes qui lient les points entre eux, créant ainsi de potentielles zones de surélévation ou de dépressions issu de l'interprétation. Afin de corriger cet état de fait et d'obtenir un MNT topologiquement le plus correct possible, ou du moins exploitable en vue de calcul des écoulements, il est nécessaire de combler les dépressions. Le module [r.fill.dir](#) sera utilisé. Dans l'exemple ci-dessous nous nous servirons du MNT creusé précédemment. Le module boucle sur le MNT jusqu'à épuisement des dépressions.

Attention ! Il est à noter que le module ne peut faire la différence entre zone de dépression réelle ou issu de l'interpolation. Il est possible dans les options de rajouter une couche de dépressions réelles (lac, étang, zones humides...)

- Comblement des zones de dépressions :

```
r.fill.dir input=mnt_creuse@PERMANENT output=mnt_creuse_sinkless
direction=mnt_creuse_direction --overwrite
```

8.2.2.3 Construction d'un réseau théorique d'écoulement

Un réseau théorique d'écoulement est une donnée générée à partir des informations d'élévation d'un MNT (Modèle Numérique de Terrain). Il simule le chemin de l'eau en fonction des valeurs d'accumulation de chaque cellule. GRASS Gis offre de nombreux modules hydrographiques performants permettant de réaliser les opérations qui suivent. [r.stream.extract](#) permet de générer, entre autre, un raster de direction de flux, un raster de cours d'eau orienté de l'amont vers l'aval et de segmenter entre chaque confluence et un vecteur de cours d'eau. Pour générer le vecteur de cours d'eau, en plus d'une donnée d'élévation en entrée, une donnée d'accumulation est nécessaire. Cette dernière peut-être obtenue à partir de [r.watershed](#).

MNT

- RGE Alti, IGN, 5m x 5m
- RGE Alti sinkless (MNT corrigé en supprimant les dépressions avec l'outil GRASS r.sink)

Fonctions GRASS

- [r.stream.extract](#)
- [r.watershed](#)

8.2.2.3.1 Construction du flux d'accumulation

- Exemple d'appel de la fonction [r.watershed](#) dans la console GRASS

```
r.watershed -b elevation=mnt_creuse_sinkless@PERMANENT  
accumulation=_watershed_accu
```

8.2.2.3.2 Construction du réseau d'écoulement théorique

- Exemple d'appel de la fonction [r.stream.extract](#) dans la console GRASS

```
r.stream.extract elevation= mnt_creuse_sinkless accumulation=_watershed_accu  
threshold=500 stream_raster=rast_stream stream_vector=vect_stream  
direction=rast_dir
```

- Le paramètre "threshold" dont la valeur a été initialisé à 500 dans cette opération renvoie à la valeur d'accumulation minimale pour qu'un cours d'eau soit créé (en l'occurrence, 500 cellules doivent s'écouler vers un point avant de former un cours d'eau, étant donné qu'une cellule mesure sur le terrain 25 m², cela équivaut à un bassin versant minimal de 12 500 m², cette valeur a été choisie afin d'obtenir un réseau théorique dense dans l'optique de raccorder les tronçons orphelins. Plus tard nous constitueront un réseau théorique de référence sur lequel nous appliquerons une valeur d'accumulation minimale de 4000, correspondant à un bassin versant d'environ 10 hectares, valeur identifiée dans le travail de recherche bibliographique de Julien Cirou et validée par l'AFB en 2017. C'est alors seulement que nous utiliserons les rasters de direction et d'accumulation).

Une fois le tracé théorique forcé réalisé, il devient possible de raccorder les tronçons isolés entre eux en utilisant le réseau théorique comme traits d'union. Il ne s'agit pas de reconstituer un réseau réel, mais bien de rendre le réseau hydrologique, qui servira de référentiel, exploitable par les SIG et ordonnable afin qu'il possède une place dans le réseau d'ensemble.

8.2.2.4 Correction de la table composite – raccords tronçons isolés et gestion des anastomoses

La suite des opérations de traitements sera réalisée via les fonctionnalités de Postgres / Postgis et une fonction développée en PLPGSQL dont le code source est disponible en annexe. Cette fonction permet de raccorder les segments faiblement isolés (- 15 mètres par défaut), les segments isolés éloignés (+ 15 mètres), elle permet de supprimer les anastomoses¹⁸ pour le calcul des rangs de strahler et calcule enfin les rangs de strahler. En effet un cours d'eau, n'a pas toujours un lit unique, en particulier dans ses sections avals ou peu pentues, mais le cours d'eau peut aussi se diviser pour d'autres raisons. Par exemple une chaussée de moulin a parfois un canal de fuite ou une rivière de contournement. Ces multiplications de d'écoulement, dans un processus automatisé de calcul d'ordination auront pour conséquence une incrémentation potentielle artificielle de la hiérarchie du cours aval. Ces fâcheuses conséquences peuvent être résolues en ne conservant qu'un écoulement principal pour le calcul de strahler. La fonction **f_reseau_correction_anastomoses_v04_01** permet d'effectuer cette correction rapidement et de manière automatisée. **Attention** : il est à noter que les zones de marais et leur réseau hydrologique particulier doivent être traitées à part. De plus certains tronçons de cours d'eau peuvent générer des erreurs pour des raisons diverses et variées. Sur le territoire de la Sèvre Nantaise un tronçon a dû être corrigée manuellement. Cette étape manuelle est la seule du processus mais sera incontournable en cas d'erreur visible.

!! Prérequis Postgres !!

Pour la suite des traitements PG, 2 schémas seront utiles, 1 pour les fonctions et 1 pour les tables.

(optionnel mais fortement conseillé) :

```
CREATE SCHEMA if not exists xx_99_utils;
```

```
CREATE SCHEMA if not exists m060_milieux_continuite_tetes_de_bv;
```

Enfin lancer les requêtes de création des fonctions (Annexes 2, 3, 4 et 12 à 21)

Étapes du traitement :

1. Import des données de GRASS vers Postgres
 - Export de GRASS vers un fichier Shape

```
v.out.ogr input=vect_stream@PERMANENT
output=../grass_data/mon_location/t_vect_stream.shp format=ESRI_Shapefile
--overwrite
```

- Import du shape dans Postgres

```
ogr2ogr -f "PostgreSQL" "PG:dbname=ma_database_pg host=mon_hote_pg
port=mon_port_pg user=mon_user_pg password=mon_password_pg" .\monshape.shp
-nln ma_table_pg -nlt POLYGON -a_srs EPSG:2154 -lco SCHEMA=mon_schema_pg
```

2. Correction de la topologie et des segments isolés
 - Fonction pg : **f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04**
 - Cette fonction a besoin de connaître l'extrémité aval du réseau.
 - Dans un premier temps le réseau va être remonté à partir du point aval et tous les segments vont être connectés à ce réseau s'ils sont à une distance inférieure ou égale à la valeur *_buffer_snap* (défaut 15 mètres).
 - [optionnel] : Les segments isolés éloignés vont être rattachés au réseau principal grâce aux données contenues dans la couche d'écoulements théoriques basée sur un MNT surcreusé (cf. partie précédente)
 - Résultats :

18 Connexion entre deux bras d'un même cours d'eau, boucle, moulin et rivière de contournement, etc).

- **Une table cours d'eau corrigés**, ne conserve pas les données de la table corrigée (perte des id d'objets...). La colonne *correction_result* permet d'identifier les tracés qui restent isolés après le traitement (*correction_result* like 'ISOLE')
- **Une table information sur les opérations réalisées**



Illustration 10: raccordement des tronçons orphelins

3. Création d'un réseau de graph avec suppression des anastomoses :
 - Fonction pg : **f_reseau_correction_anastomoses_v04_01**
 - Résultats :
 - **Une table** identique à la table en entrée avec **ajout colonne graph**
 - **Une table avec géométries d'anastomoses détectées** (utile pour visualiser les anastomoses détectées et pour le traitement ultérieur)

Principe de fonctionnement de la fonction de correction des anastomoses

Attention, cette correction du référentiel n'a pas vocation à s'afficher, elle suit une logique de graph et poursuit le seul objectif de hiérarchiser l'ensemble des tronçons au sein d'un ensemble. Si la géométrie est conservée dans la table de cours d'eau, elle ne représente en aucun cas le cours d'eau réel.

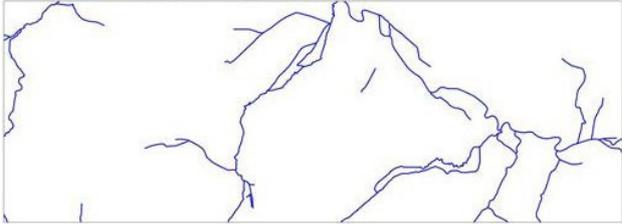
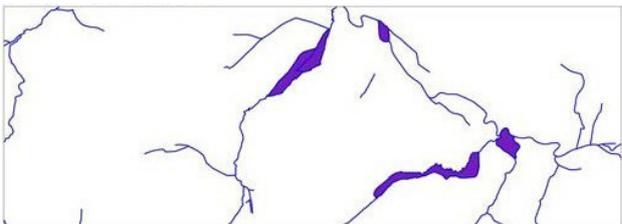
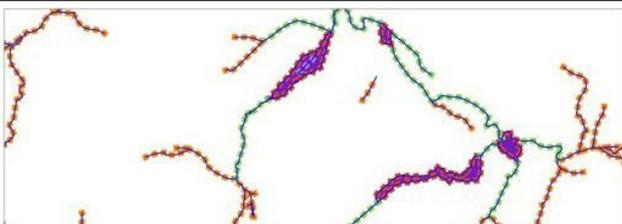
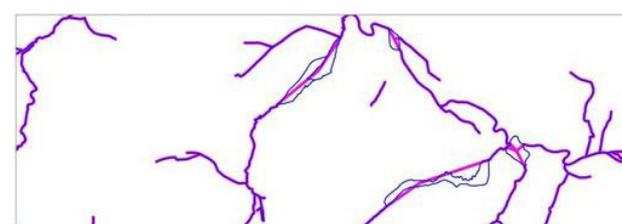
Le réseau originel et ses boucles, moulins, anastomoses :	
Détection des boucles et « polygonisation » des anastomoses :	
Détection des relations entre les cours et les anastomoses :	
Corrections du graph de cours d'eau (en rose) en raccordant chaque tronçon connecté à l'anastomose au centroïde du polygone.	

Illustration 11: Principes de fonctionnement de la fonction de correction des anastomoses

4. Lancer la fonction **f_reseau_hydro_strahler_rank_v01_04**, elle réalisera automatiquement la suite de traitement ci-dessous :
5. Calcul des rangs de strahler [premier passage] :
 - Fonction pg : **f_reseau_hydro_strahler_rank_v01_04**
 - Nécessite de connaître le point bas du réseau. Cette fonction va orienter tous les segments et calculer leur rang de strahler.
 - Résultats :
 - **Une table avec ajout de colonnes** : *geom_graph_ori* (orienté amont / aval) et *strahler_rank*

6. Ajout des rangs de strahler du tronçon précédent et suivant et id du tronçon suivant [premier passage] :
 - Fonction pg : **f_reseau_hydro_strahler_rank_v01_04**
 - Résultats :
 - **Ajout de colonnes** : *strahler_before_outlet*, *strahler_after_outlet* et *graph_gid_next*.

7. Suppression des minis tronçons :
 - Fonction pg : **f_reseau_hydro_strahler_rank_v01_04**
 - Rang de strahler = 1 qui génère un strahler ≥ 2 et dont la longueur est inférieure à une distance légèrement supérieure à la diagonale de 2 pixels de MNT (14.5 mètres). Diagonale de 2 pixels de MNT 5 m x 5 m = 14.14 m
 - Résultats :
 - **Une table graph nettoyée des petites erreurs**

8. Calcul des rangs de strahler [second passage] (suite à la suppression des minis tronçons)
 - Fonction pg : **f_reseau_hydro_strahler_rank_v01_04**
 - Résultats :
 - **Une table graph ordonnée selon la méthode strahler**

9. Ajout des rangs de strahler du tronçon précédent et suivant et id du tronçon suivant [second passage] :
 - Fonction pg : **f_reseau_hydro_strahler_rank_v01_04**
 - Résultats :
 - **Ajout de colonnes** : *strahler_before_outlet*, *strahler_after_outlet* et *graph_gid_next*.

10. Orientation de la géométrie d'origine (*geom*) en *geom_ori* sur la base de l'orientation de la *geom_graph_ori*.
 - Fonction pg : **f_reseau_hydro_strahler_rank_v01_04**
 - Cette donnée servira lors de l'extraction des points haut/bas du chemin le plus long dans chaque bassin versant.
 - Résultats :
 - **Ajout de colonne** : *geom_ori*

L'ensemble des fonctions Postgres commentées (avec exemple d'appel) sont disponibles en annexes 3 et 4.

8.3 Construction des micros bassins versants

Avant de délimiter les têtes de bassin versant, il est nécessaire de délimiter l'ensemble des micros bassins versant de chaque tronçon de cours d'eau. À ce stade il est également indispensable de repasser sous GRASS pour retravailler le MNT à partir du référentiel maintenant corrigé, ainsi que pour recalculer les rasters de direction et d'accumulation avec de nouveaux paramètres.

Étapes du traitement :

1. Copie de la table de graph dans une table qui ne contiendra qu'une seule géométrie : *geom* (pour le traitement GRASS à venir) et qui ne contient pas les segments encore isolés ('ISOLE'), car trop éloignés ou en dehors du bassin versant de référence. **Attention ! Ne pas utiliser la géométrie « graphée ».**

Un exemple de récupération des seules géométries intéressante pour la suite des traitements en annexe 5.

2. Surcreusement du MNT : Suivre la précédente procédure (cf. préparation du MNT) avec la colonne *geom* de la table créée à l'étape 1.
3. Calcul du raster de direction : Suivre la précédente procédure ([cf. Comblement des dépressions **OU** réutilisation du *mnt_sinkless* précédent] **EI** Construction d'un réseau théorique d'écoulement étape 1 et 2).
 - On utilisera donc ici, pour les raisons expliquées précédemment, la valeur de 4000 pour le « Threshold » (valeur d'accumulation).

4. Import du vecteur référentiel hydro dans GRASS

```
v.in.ogr input='PG:dbname=ma_database_pg host=mon_hote_pg port=mon_port
user=mon_user_pg password=mon_password_pg' layer=mon_schema.
stream_ddtm_ign_vect output=stream_ddtm_ign_vect type=line geometry=geom
--overwrite
```

5. Transformation du référentiel hydro en raster

- V.to.rast
- Exemple :

```
v.to.rast --overwrite input=stream_ddtm_ign_vect@PERMANENT type=line
output=stream_ddtm_ign_rast use=cat --overwrite
```

6. Délimitation des micros bassins versants

- Les fonctions du module [r.stream.basins](#) de GRASS permettent de passer à la commande de multiples paramètres qui nous permettent d'atteindre différents objectifs. Par exemple si on utilise tous les paramètres par défaut, le module nous renverra une cartographie des bassins-versants de chaque segment de cours d'eau. Si on souhaite préciser son analyse il sera nécessaire de lui passer de nouvelles informations qui seront obtenus à partir d'autres modules. La [page de documentation](#) GRASS dédiée est très détaillée et permet de visualiser rapidement les finalités d'une partie des fonctions. Dans notre cas, c'est la fonction basique qui nous intéresse.
- Exemple :

```
r.stream.basins dir=mnt_creuse_sinkless_direction@PERMANENT
stream=stream_ddtm_ign@PERMANENT basins=stream_ddtm_ign_bv_rast memory=10000
--overwrite
```

7. Transformation du raster de bassins versants en vecteur

- Exemple :

```
r.to.vect input=stream_ddtm_ign_bv_rast@PERMANENT output=stream_ddtm_ign_bv_vect
type=area -overwrite
```

8. Import des données de GRASS vers Postgres

- Export de GRASS vers un fichier Shape

```
v.out.ogr input=stream_ddtm_ign_bv_vect@PERMANENT
output=../grass_data/mon_location/stream_ddtm_ign_tbv.shp format=ESRI_Shapefile
--overwrite
```

- Import du shape dans Postgres

```
ogr2ogr -f "PostgreSQL" "PG:dbname=ma_database_pg host=mon_hote_pg
port=mon_port_pg user=mon_user_pg password=mon_password_pg"
```

```
..\stream_ddtm_ign_tbv .shp -nln ma_table_tbv_pg -nlt POLYGON -a_srs EPSG:2154  
-lco SCHEMA=mon_schema_pg
```

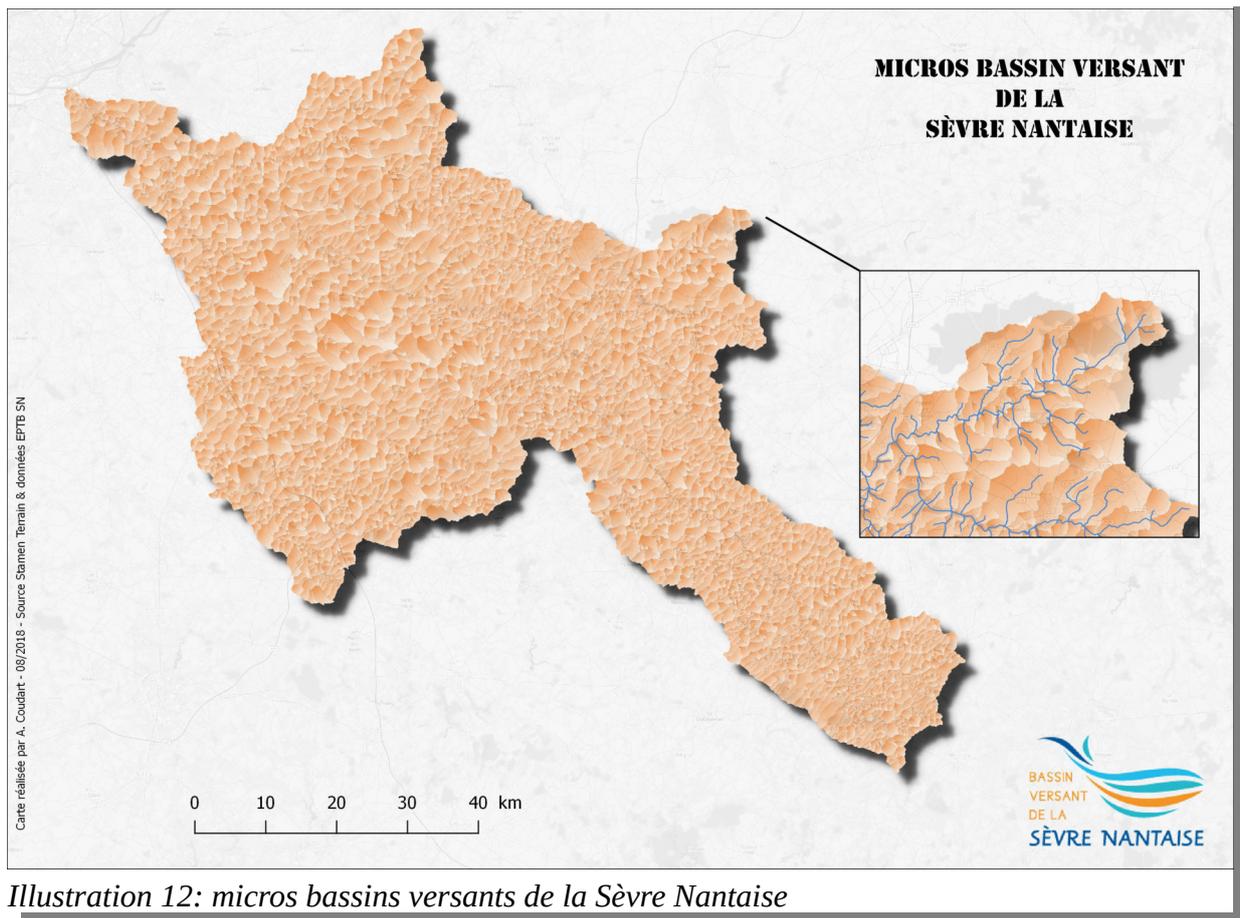


Illustration 12: micros bassins versants de la Sèvre Nantaise

Au total, le bassin de la Sèvre Nantaise, compte 7696 micros bassins versants, qui représente autant de tronçon unique de cours d'eau.

8.4 Construction des têtes bassin versant

Maintenant que nous avons délimité l'ensemble des micros bassins pour chaque tronçon du référentiel hydrologique, il nous reste à les unir pour former les têtes de bassin versant. Cette partie des opérations sera réalisée grâce à une requête SQL directement dans Postgres. En voici la logique générale :

- Les BV issus de tronçons supérieur à 2 ne sont pas pris en compte.
- Les BV issus de tronçons de rang 1 et raccordés à des tronçons supérieurs à 2 sont considérés comme des TBV à part entières.
- Les BV de rangs inférieurs ou égal à 2 sont fusionnés quand le point exutoire final du graph forme un rang 3.

Outre la fusion des micros BV en TBV la requête réalise les opérations, dont les résultats seront utiles lors des requêtes de caractérisation :

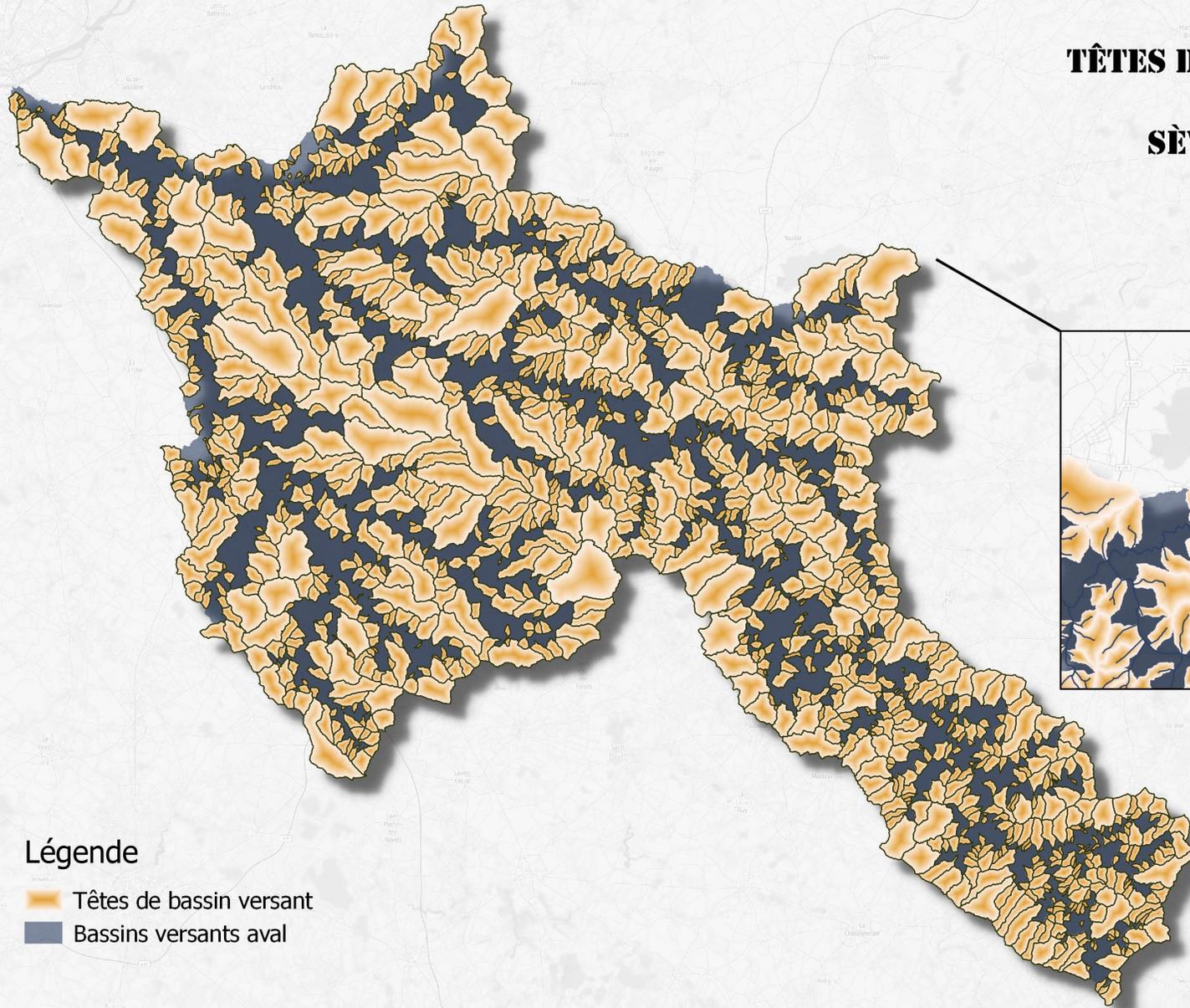
- Liste les IDs des cours d'eau composant les BV.
- Association des IDs aval du référentiel hydrologique au futur IDs de la TBV.

- Calcul du chemin le plus long dans le graph composant chaque TBV et enregistre sa géométrie.
- Récupération du point haut et du point bas du chemin le plus longtemps.
- Récupération du rang de strahler maxi de la TBV.
- Récupération du rang de strahler maxi de la TBV suivante et précédente s'il y en a.
- Suppression des TBV d'une aire inférieure à 3,1 hectares (décision interne relatif à la pertinence d'un travail d'animation ultérieur à une échelle cohérente) (cette valeur peut-être ôtée ou modifiée en fonction des choix de chaque structure).

Le script SQL commenté est disponible en annexe 6.

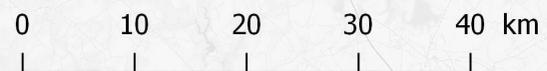
Le nombre de têtes de bassins versant sur le territoire de la Sèvre Nantaise, s'élève à 1 277 pour un total de 1 759 km² soit 74,83 % du territoire. À titre de comparaison, la délimitation des têtes de bassins versant réalisée avec comme référentiel hydrologique le tracé théorique des cours d'eau (avec pour valeur d'accumulation 4000) nous montre une superficie totale cumulée de 1610 km², soit environ 68 % du territoire. Ce résultat est tout à fait cohérent. Puisque le tracé théorique augmente le linéaire de chevelu, il a tendance à remonter les exutoires formant des rangs 3 vers l'amont. Ces résultats nous montrent également que le choix du référentiel n'est pas sans conséquence.

TÊTES DE BASSIN VERSANT DE LA SÈVRE NANTAISE



Légende

-  Têtes de bassin versant
-  Bassins versants aval



9 Caractérisation des têtes de bassin versant

La caractérisation des têtes de bassin versant consiste à évaluer l'état et les pressions qui leurs sont associées, cela revient à dresser la carte d'identité des territoires. Cette étape comprend **trois échelles** qui seront étudiées **en quatre temps** selon **trois enjeux majeurs**.

Les 3 échelles :

- *La caractérisation physique et morphologique du bassin versant* : les critères définissent la structure physique des têtes de bassin et via des formules mathématiques éprouvées permettent d'évaluer les capacités de réaction et de résilience de la TBV.
- *La caractérisation du lit mineur et de la bande riveraine* : les critères définissent l'état et les pressions qui s'exercent sur le cours d'eau ou à proximité immédiate. Les résultats permettront de cibler un certain type de travaux de restauration ou de mesure de protection spécifique au lit mineur ou aux berges...
- *La caractérisation du bassin versant* : les critères sont peu ou prou les mêmes que pour le lit mineur et la bande riveraine mais appliquées pour l'ensemble de la surface de la tête de bassin. Des éléments supplémentaires comme la densité de prélèvement et la dynamique d'urbanisation sont ajoutés. Les résultats permettront également de cibler directement des typologies d'actions à l'échelle du bassin versant, inscription dans les PLU(i), sensibilisation des élus, accompagnement agricole...

Cette **approche scalaire**, telle des poupées russes permet d'évaluer l'état et les pressions qui s'exercent sur chaque tête de bassin et sont autant d'axes de travail. En effet ces choix découlent de réalité physique, chaque compartiment subit des pressions qui lui sont propres et les réponses apportées sont donc différentes et ciblées.

Le lit mineur désigne l'espace du cours d'eau où s'effectue l'écoulement la majeure partie du temps (par opposition au lit majeur qui constitue l'espace de mobilité du cours d'eau lors d'épisode exceptionnels). Les problématiques rencontrées sont d'ordre hydromorphologique¹⁹ tels les recalibrages de la section d'écoulement (creusement et élargissement), les rectifications du linéaire (suppression des méandres...), les abreuvements directs (divagation du bétail constituant des apports de matières urées et fécales, colmatage du substrat par mobilisation massive de sédiments...). Les réponses apportées s'orientent donc vers des travaux de restauration qui visent un retour à un état cohérent. **Attention** : Cette échelle a été jumelée avec la bande riveraine. En effet, trop peu d'indicateurs homogènes sur l'ensemble du territoire pouvaient être mobilisés pour caractériser ce compartiment. Un travail de prospection terrain ciblé sur les critères essentiels devrait être envisagé pour caractériser précisément le lit mineur de chaque TBV.

La bande riveraine est définie par une largeur de 10 mètres de part et d'autre du linéaire de cours d'eau. Cette définition est la définition la plus courante dans la bibliographie (AFB, 2017 - Cirou, 2017 – Salpin, 2016 – Le COPREN, 2008). En moyenne elle varie entre un minimum de 5 mètres et un maximum de 15 mètres dans les travaux scientifiques et indique les largeurs minimales de bon fonctionnement en matière d'auto-épuration. Les problématiques rencontrées sont de l'ordre de l'aménagement immédiat du territoire, comme l'absence de ripisylve (qui favorise, entre autre l'échauffement des températures, la prolifération algale et diminue le potentiel auto-épurateur du cours d'eau) l'absence de bande enherbée ou l'artificialisation de cet espace privant ainsi le milieu naturel de ce potentiel épurateur. Les réponses apportées ici s'axeront par exemple sur une gestion raisonnée ou une restauration de la ripisylve, sur l'information au riverain, la surveillance ou les rappels à la loi en matière de respect des règles de ZNT (Zone de Non Traitement)...

19 Relatif à la forme du cours d'eau

Le bassin versant quant-à lui fait référence à l'aménagement du territoire dans son ensemble. Les actions ici sont de l'ordre de l'animation territoriale, il s'agit de travailler avec les différents acteurs du territoire afin de faire prendre conscience des enjeux que constitue les milieux aquatiques et la façon dont l'aménagement de notre territoire influe sur son équilibre et donc sur sa capacité à rendre des services écosystémiques.

La sensibilité ou caractérisation physique et morphologique désigne les caractères physiques intrinsèques du territoire, superficie, pente, réseau de cours d'eau, réel ou théorique, réactivité hydraulique... Ce compartiment n'est pas une échelle, il n'est pas un territoire d'action pour les collectivités mais il donne des informations importantes sur le caractère de la TBV et permet de mettre en perspective des enjeux. À la manière de l'approche aléa / vulnérabilité en matière de gestion des risques il permet de relativiser les indicateurs d'état et de pression.

Une première analyse en quatre temps :

- Les résultats bruts : Ils sont stockés et conservés dans la table des résultats. Ils seront transformés pour être comparable les uns aux autres malgré leur hétérogénéité d'unité.
- Les résultats centrés, réduits, borné (entre 0 et 1) : Ils sont stockés, conservés dans la table de résultats. Ces valeurs seront assemblées suivant des scénarios de pondérations.
- Les résultats sont assemblés suivant des scénarios de pondération (enjeu qualité d'eau, enjeu qualité des milieux, indice de pression, etc) avant d'être discrétisés
- Les résultats sont discrétisés, c'est-à-dire ordonnés suivant une méthode statistique ou une autre.

Chaque critère peut ou non intervenir dans la grille de lecture des **enjeux** concourant à l'objectif final de reconquête de la qualité de l'eau portée par la directive cadre sur l'eau (DCE) du 23 octobre 2000 (directive 2000/60) :

- **Enjeu qualité de l'eau (QE)**
- **Enjeu quantité de l'eau (QTE)**
- **Enjeu qualité des milieux (QM)**

Certains critères sont utilisés dans d'autres types de scénarios de pondération comme l'indice de pression qui regroupe les différents résultats d'occupation du sol, chacun pondéré avec un indice propre.

L'ensemble de ces indicateurs est le fruit d'un travail de compilation de la bibliographie existante et d'une concertation active avec les structures voisines portant des études similaires ainsi qu'avec les services de l'État et en particulier l'AFB. Lors de l'étude certains critères étudiés ont été écartés, souvent à cause de l'hétérogénéité de leur représentation sur le territoire.

9.1 Caractérisation physique et morphologique – Sensibilité des TBV

La caractérisation physique et morphologique de la tête de bassin versant consiste à éprouver sa capacité de réaction et donc de résilience. La dynamique hydraulique d'un cours d'eau pentu et réactif étant plus importante, il aura plus de facilité à s'auto-restaurer qu'un cours d'eau de plaine à faible débit. Ces données physiques nous informe sur la sensibilité des territoires. La colonne « Positif » est un paramètre pris en compte dans le calcul des scénarios de pondération, **False** inverse la valeur. En effet certains critères ont une influence positive et d'autres négatives.

Attention ! Un ordre de calcul doit être suivi : Indice de pression individuel → tous les autres Indicateurs → Indice de pression global.

9.1.1 Grille d'indicateurs à l'échelle des caractéristiques physiques de la TBV

Indicateur physique et morphologique	Formule	Pondération	Positif	Utilisation
Surface de la TBV	Surface TBV en ha = surf total	1	TRUE	Finale
Pente moyenne de la TBV	avg(valeur de pixel) par TBV	1	FALSE	Finale
Pente moyenne du chemin le plus long de la TBV	différence elevation amont-aval / longueur chemin le plus long	1	TRUE	Finale
Indice de compacité de Gravéius	$kg = \text{périmètre BV} / (2\sqrt{\pi} \cdot \text{surf})$	1	FALSE	Finale
Temps de concentration des pluies (Formule de Passini)	$= 0,108 * 3\sqrt{(\text{surf BV}(\text{km}^2) * \text{Long du plus long chemin eau (km)} / \sqrt{\text{pente du plus long chemin de l'eau}})$	1	TRUE	Finale
Densité du réseau de CE	Dens réseau CE = linéaire CE TBV / superficie TBV	1	FALSE	Finale
Densité du réseau de point bas	Dens réseau theorique = linéaire theorique TBV / superficie TBV	1	FALSE	Finale
Position de la TBV dans le réseau hydrographique	positionnement apicale (connexion sur un rang de strahler 3) ou tributaire (connexion sur des rangs de strahler 4 et 5 ou 6 et 7)	2	TRUE	Finale

Illustration 13: Grille d'indicateur à l'échelle des caractéristiques physiques de la TBV

9.1.2 Préparation des données rasters nécessaires aux calculs

Une partie du travail à suivre nécessite des données raster, celles-ci seront intégrées à la base de donnée pour en faciliter les calculs. Par souci d'optimisation de l'espace disque nous cumulerons sur un seul raster multibande les différentes informations nécessaires. Pour ce faire on utilisera 2 nouveaux modules de GRASS, [r.slope.aspect](#) et [i.group](#), ainsi qu'une nouvelle fonction de Postgis pour l'import d'un raster dans Postgres, [raster2pgsql](#).

Nous avons besoin de 3 informations :

- Raster MNT brut
- Raster de pente (en pourcentage) (slope)
- Raster aspect

Le raster de pente utilisera le raster MNT brut et surtout pas le MNT corrigé pour le tracé des écoulements théoriques.

- Création des rasters de pentes et d'aspects :

```
r.slope.aspect elevation=MNT_brut@Mon_Mapset slope=MNT_brut_slope@Mon_Mapset aspect=MNT_brut_aspect@Mon_Mapset format=percent precision=FCCELL -overwrite
```

- Création du raster 3 bandes :

```
i.group group=topo input=MNT_brut,MNT_brut_aspect, MNT_brut_slope
```

- Export au format Tif :

```
r.out.gdal input=topo output=../mon_chemin/MNT_aspect_slope_percent_grass.tif
```

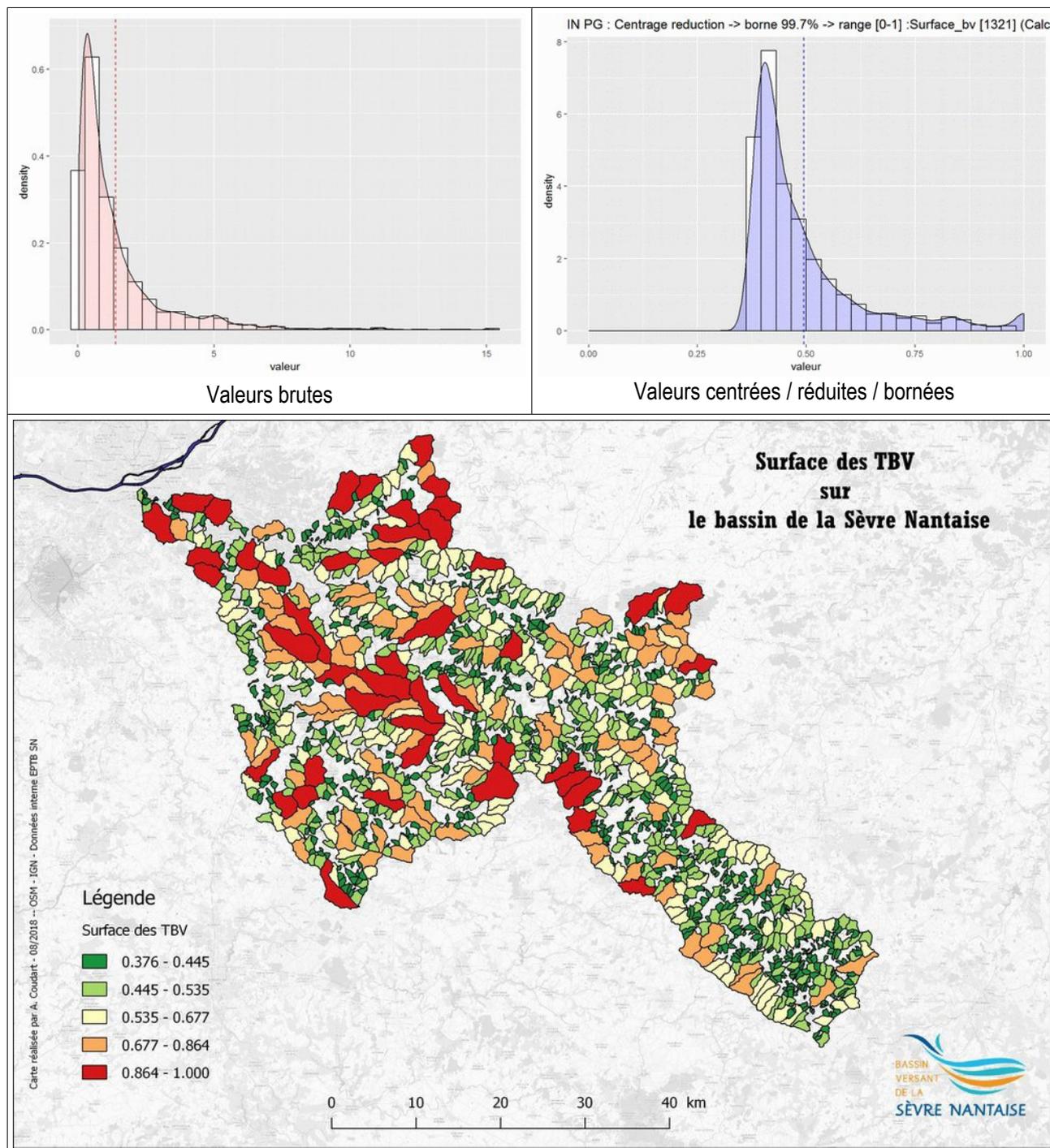
- Import dans une table Postgres :

```
raster2pgsql -d -C -Y -k -F -I -q -M -s 2154 -t 50 x50 -l 50 ../mon_chemin/MNT_aspect_slope_percent_grass.tif mon_schema_destination ma_table_destination | /psql -U mon_user -d ma_base -h mon_hote -p mon_port
```

Dans les exemples qui suivent, nous pourrons voir la distribution des données brutes de l'indicateur, la distribution après centrage, réduction, bornage ainsi que la carte associée et classifiée suivant la méthode des seuils naturels de Jenks (hormis pour le dernier indicateur relatif à la position de la TBV dans le réseau hydrographique. Ce dernier est classifié selon sa valeur).

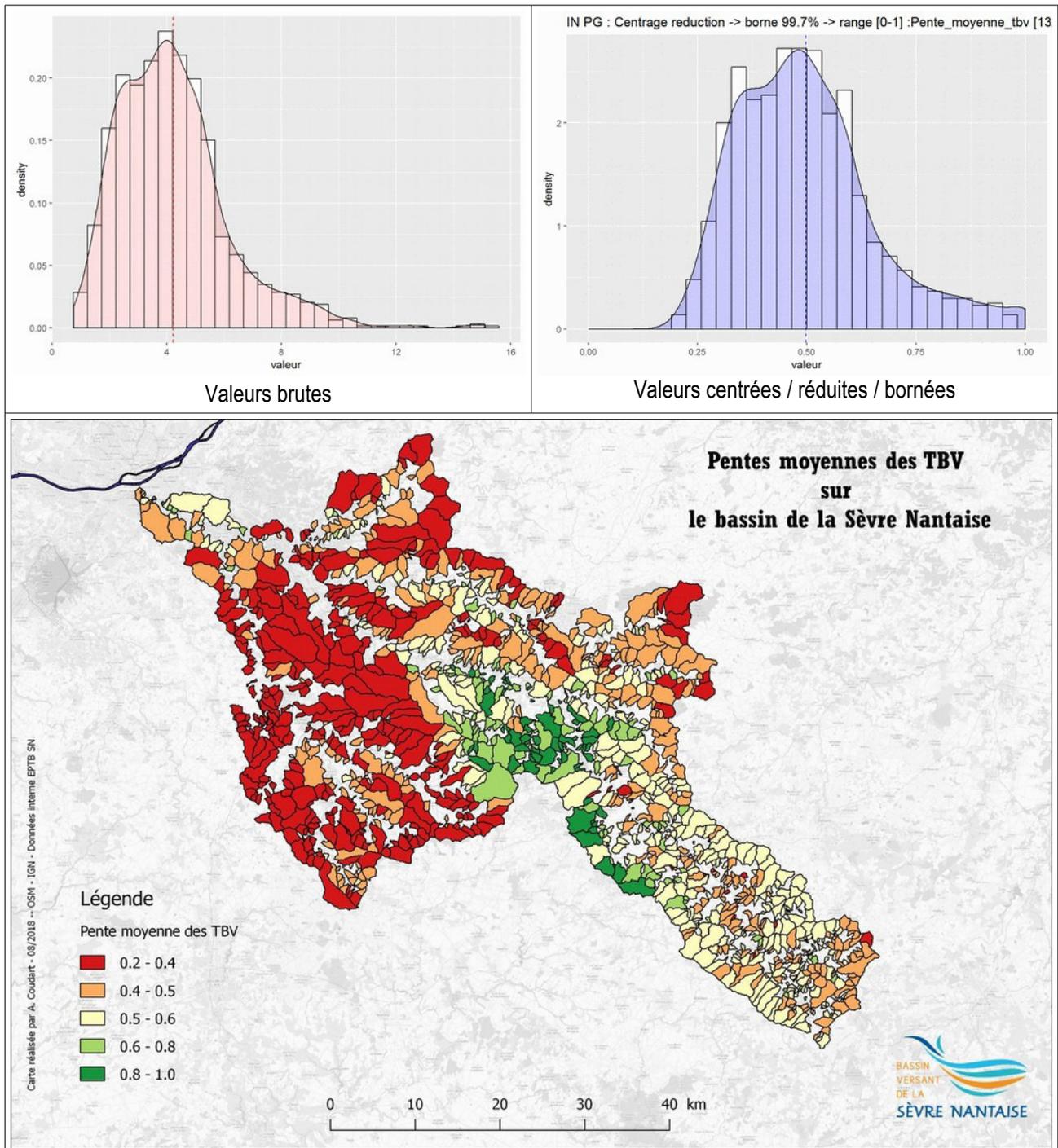
9.1.3 Surface de la TBV

Calcul visant à connaître la taille de la tête de bassin versant.



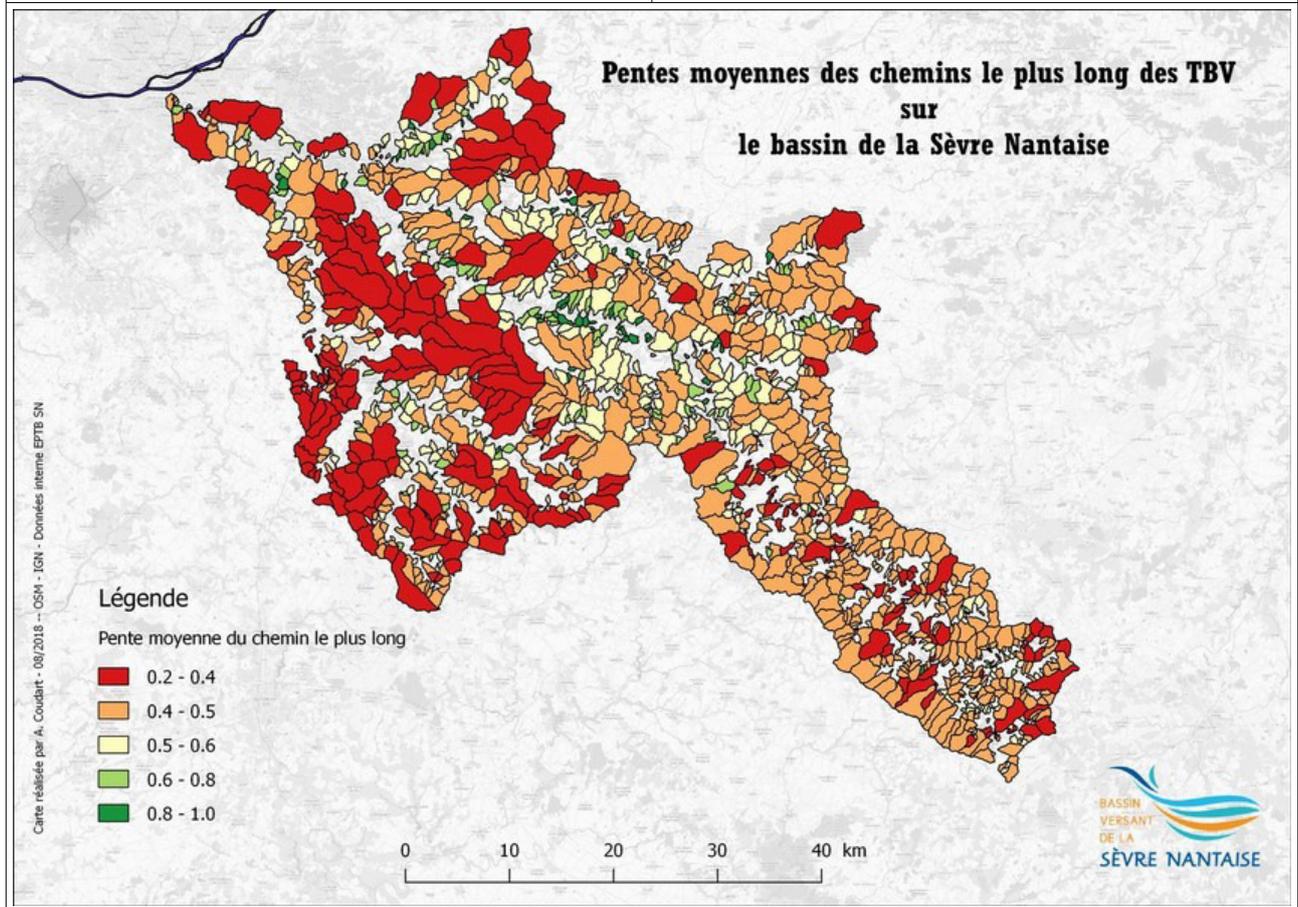
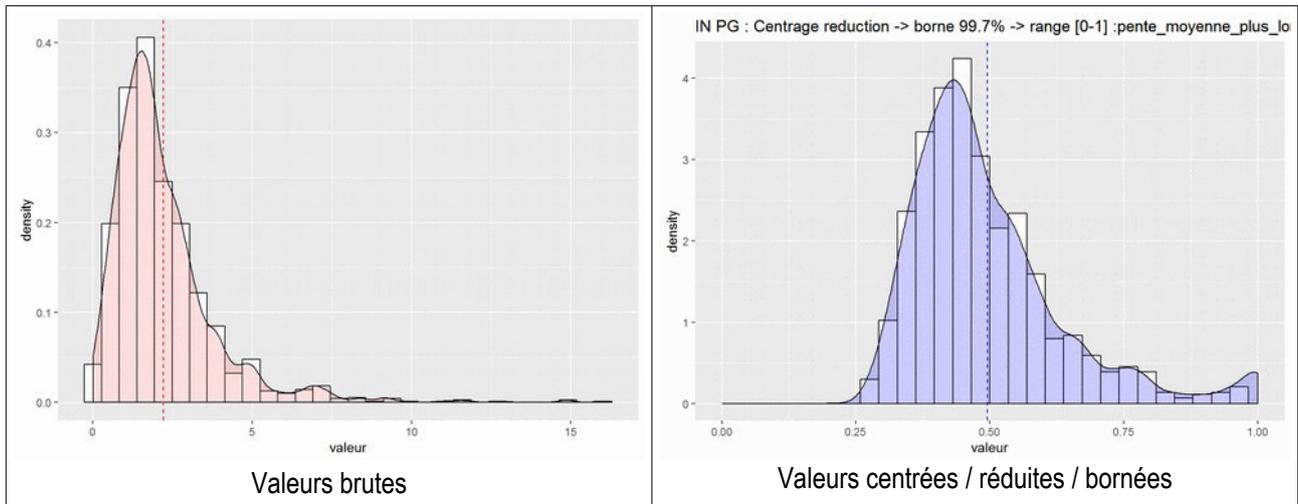
9.1.4 Pente moyenne du bassin versant

La pente moyenne du bassin versant nous donne un indice en termes de relief, plus la valeur sera élevée et plus le bassin versant sera accidenté, augmentant ainsi sa réactivité.



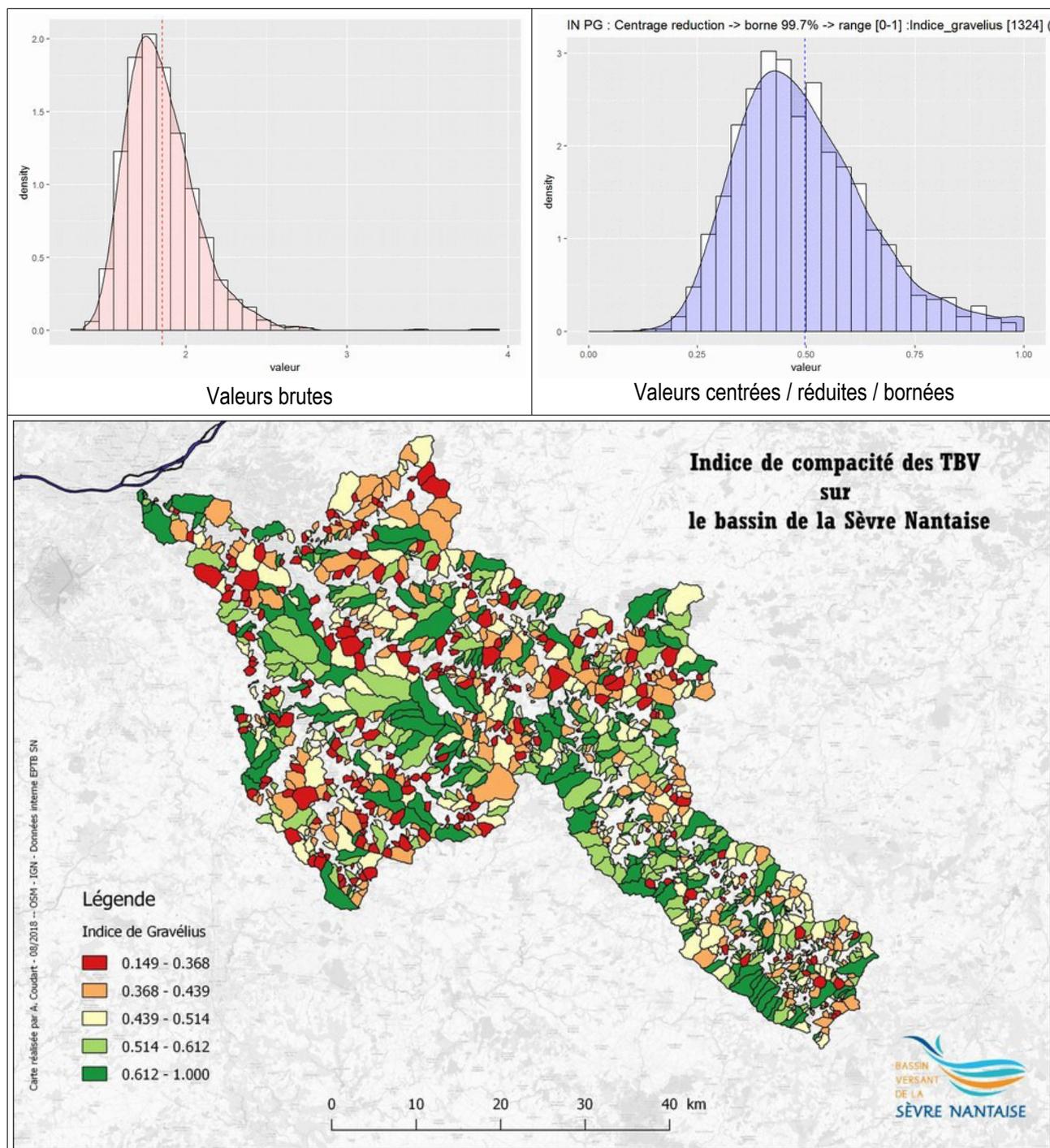
9.1.5 Pente du chemin le plus long

La pente du chemin le plus long nous informe sur la puissance spécifique du cours d'eau. Plus la valeur sera élevée plus le cours d'eau bénéficiera d'une capacité d'auto-restauration importante.



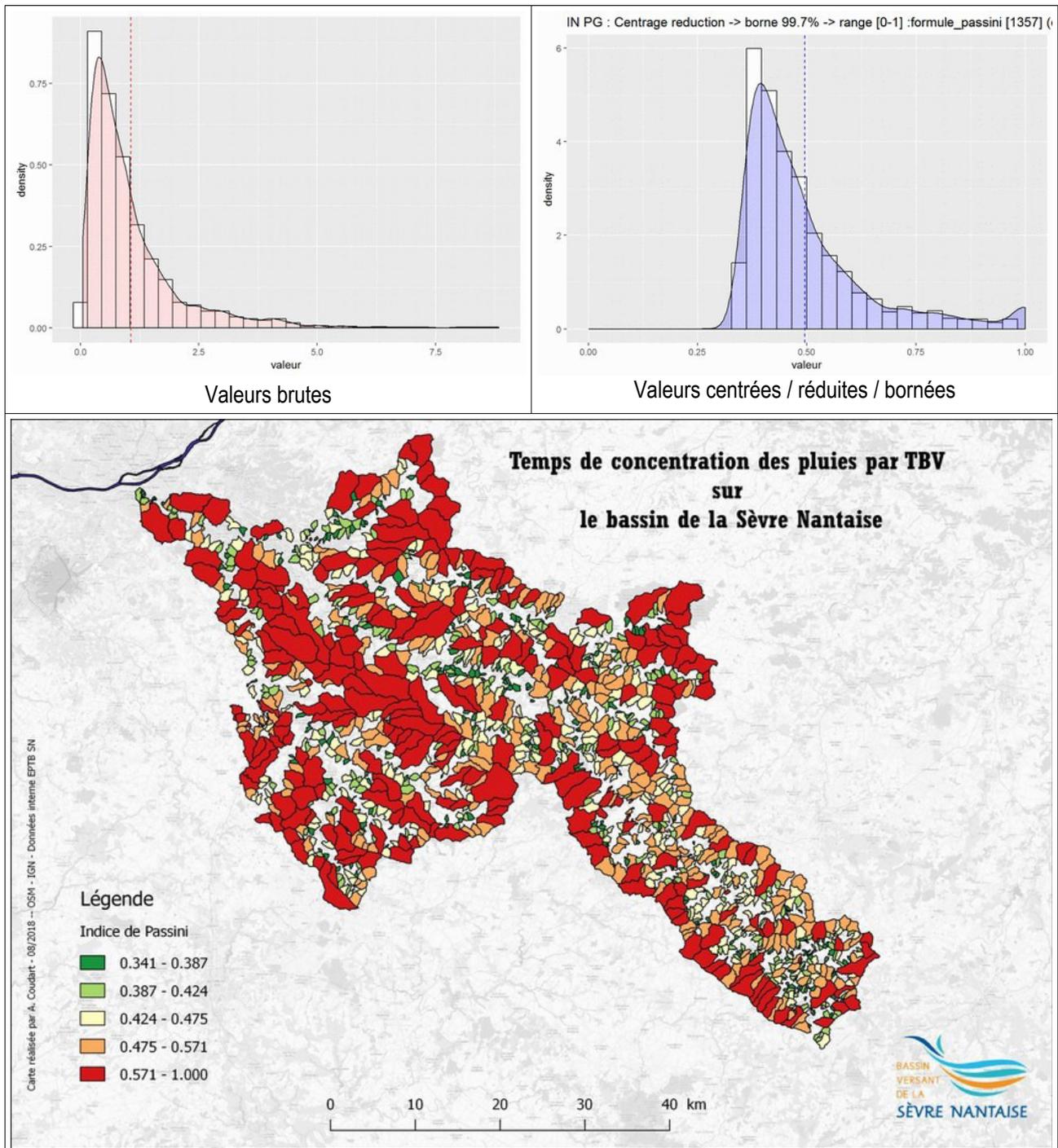
9.1.6 Indice de Gravélius (ou indice de compacité)

L'indice de Gravélius est un indice de compacité utilisé dans les modèles hydrauliques, il permet de connaître la forme générale d'un bassin versant. Un indice de 1 correspond à un bassin versant rond peu réactif aux précipitations car très probablement composé d'un réseau de cours d'eau dense, à l'inverse plus l'indice est élevé et plus la forme du bassin versant s'allonge, traduisant une plus grande réactivité et assimilé à un réseau hydraulique principal majoritaire alimenté par un réseau peu dense de cours d'eau perpendiculaire. Plus le bassin versant est allongé plus le débit de pointe des crues sera important.



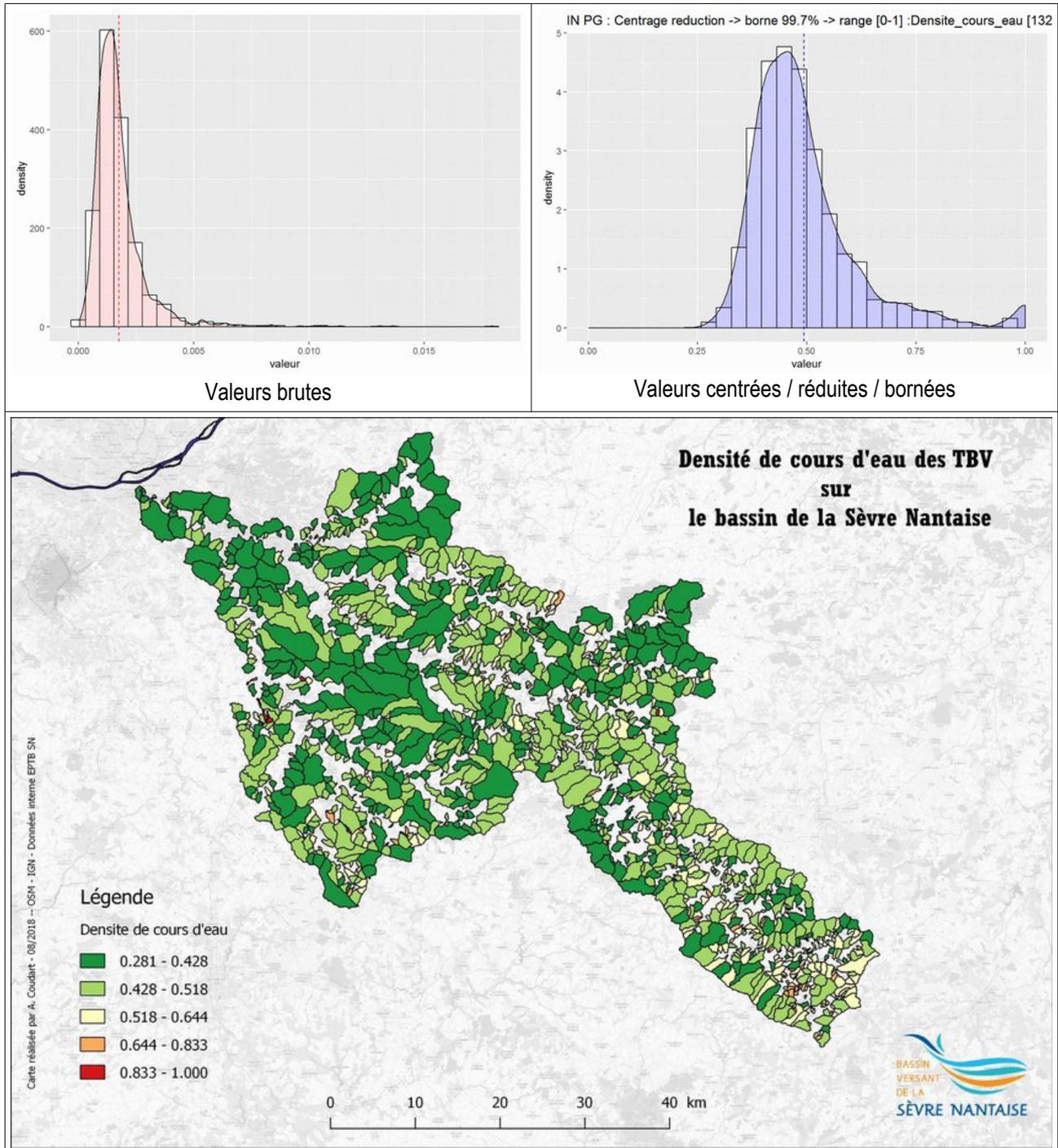
9.1.7 Temps de concentration des pluies (Passini)

Le temps de concentration des pluies est un indicateur utilisé dans les modélisations hydrauliques, il permet d'évaluer la vitesse de transfert des pluies de l'amont vers l'aval.



9.1.8 Indice de densité de chevelu

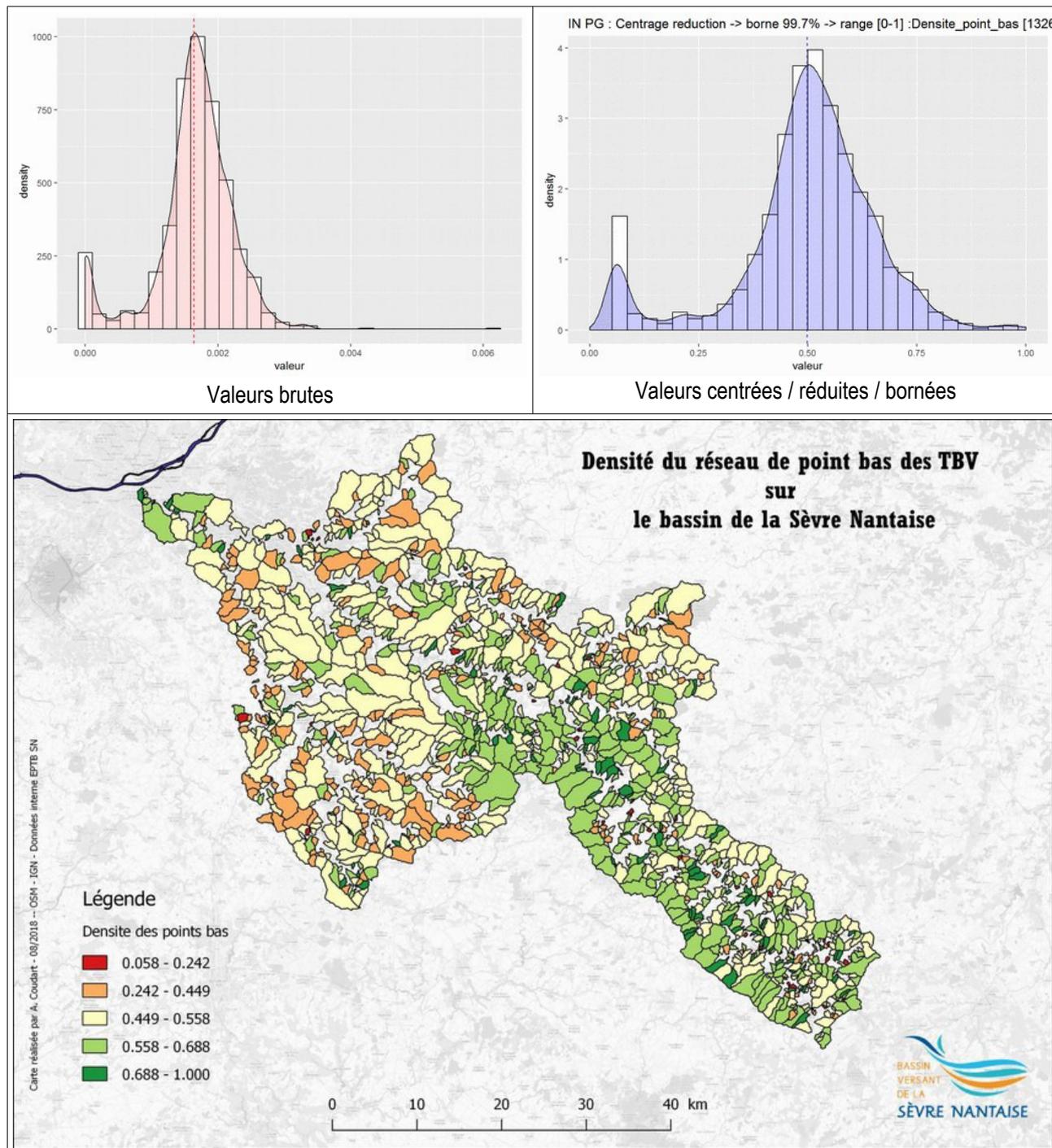
La densité de chevelu nous informe sur les capacités d'auto épuration du bassin versant. En effet plus le réseau est dense et plus les zones de connexion entre l'eau et la terre (zones hyporéïques) seront nombreuses, augmentant ainsi les zones de dénitrification et les capacités d'auto épuration du cours d'eau. Un cours d'eau de forme sinusoïdale (plaine) sera donc a priori plus auto-épurateur qu'un cours d'eau rectiligne.



9.1.9 Densité du réseau de points bas

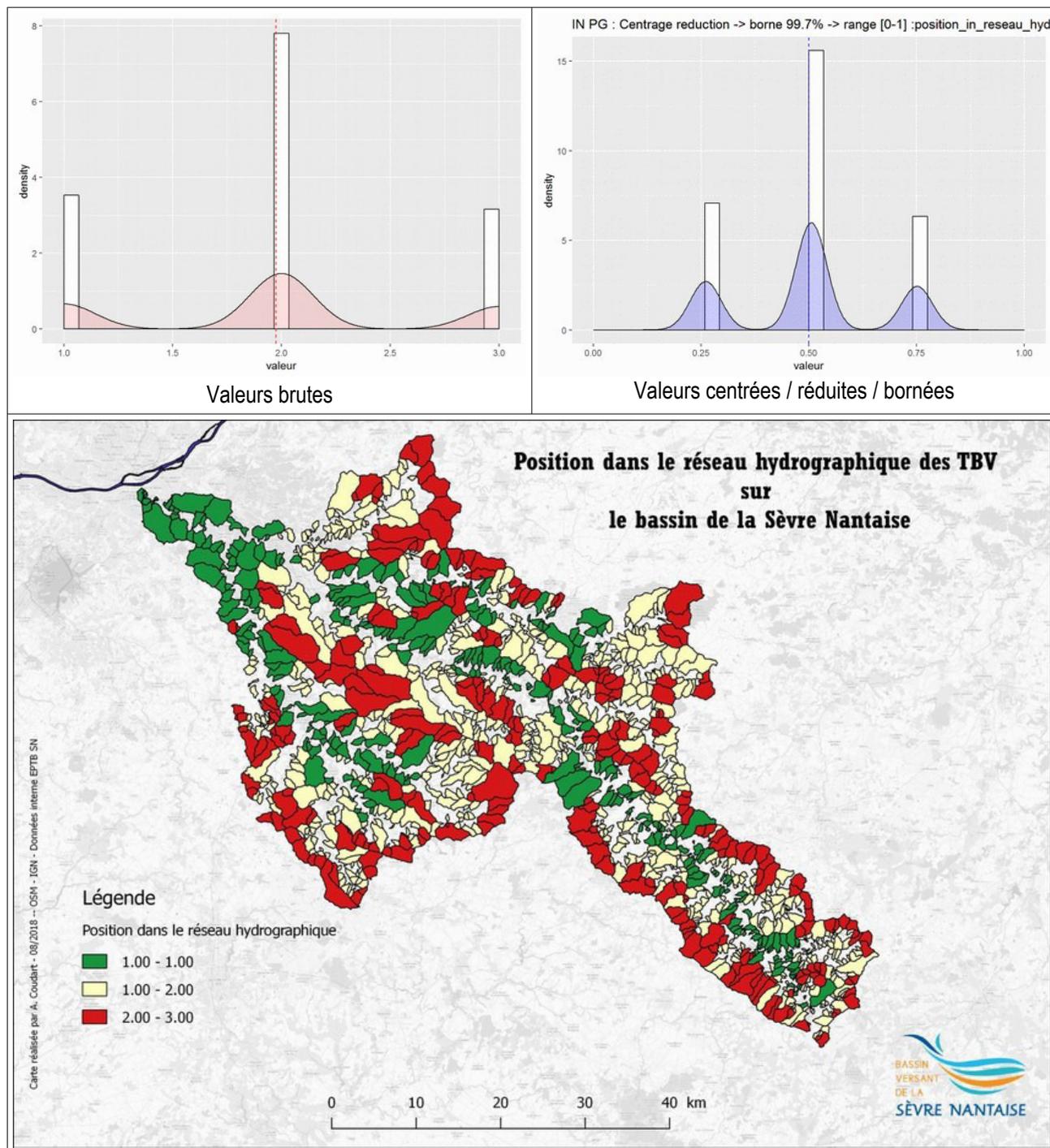
Au contraire de la densité du réseau de chevelu, cet indice porte sur le réseau d'écoulement théorique créé grâce au MNT. Cet élément apporte une information en termes de capacité de ruissellement du bassin versant.

Même script que pour la densité de chevelu en substituant le référentiel hydrologique réel par le référentiel hydrologique théorique.



9.1.10 Position de la TBV dans le réseau hydrographique

Cet indice nous apporte une information en termes de localisation de la TBV, si sa position est apicale (connectée à une extrémité amont d'un rang 3) alors la bibliographie montre que son incidence sur le milieu sera plus importante qu'une position tributaire latéral (connectée à des rangs 3 et 4) ou tributaire aval (connectée à des rangs >5). Cela s'explique par les capacités de dilution plus importante de rangs supérieurs, ainsi une tête de bassin versant apicale dégradée aura un impacté négatif fort sur les milieux à l'aval contrairement à une tête de bassin versant tributaire aval qui verra son rôle minimisé.



9.2 Caractérisation des pressions

Ces ensembles d'indicateurs ont pour rôle de dresser l'état des lieux des différentes pressions qui s'exercent sur les têtes de bassin versant. Ils sont liés à une valeur de pondération dépendant d'un enjeu (ou scénario de pondération).

9.2.1 Grilles d'indicateurs à l'échelle du lit mineur et de la bande riveraine

Indicateur d'état et pression sur le lit mineur et la bande riveraine	Formule	Pondération QM	Pondération QE	Pondération QTE	Positif	Utilisation
Taux de CE impactés par les plans d'eau en barrage	linéaire CE impacté par plan d'eau / longueur CE*100	1	1	1	TRUE	Finale
Taux de zones humides connectées au cours d'eau	superficie zh dans buff10 CE / superficie bande riveraine*100	1	1	1	FALSE	A Venir
Taux de CE avec ripisylve bande boisée ou boisement	Superficie ripisylve / superficie bande riveraine *100	1	1	0.5	FALSE	Finale
Densité de mares dans la bande riveraine	superficie mares / superficie bande riveraine *100	1	0.5	/	FALSE	Finale
Indice de pression dans la bande riveraine	Voir tableau indice de pression	2	2	0.5	TRUE	Finale
Taux d'imperméabilisation de la bande riveraine	surface imperméable / superficie bande riveraine *100	1	1	1	TRUE	Finale
Pression liées aux rejets de STEP	A définir	1	1	0.5	?	A venir

Illustration 14: Grille d'indicateurs à l'échelle du lit mineur et de la BR

9.2.2 Grille d'indicateurs à l'échelle des têtes de bassin versant

Indicateur d'état et pression sur le bassin versant	Formule	Pondération QM	Pondération QE	Pondération QTE	Positif	Utilisation
Densité surfacique de plans d'eau dans la TBV	superficie plan d'eau / superficie bv	1	1	1	TRUE	Finale
Taux de ZH dans la TBV	superficie zh / superficie TBV	1	1	1	FALSE	A venir
Densité de haies dans la TBV	linéaire de haies / superficie TBV	1	1	0.5	FALSE	Finale
Densité de haies efficaces dans la TBV	linéaire haies efficaces / superficie TBV	1	1	1	FALSE	A venir
Densité de mares dans la TBV	superficie mares / superficie TBV	1	0.5	/	FALSE	Finale
Taux d'artificialisation de la TBV	superficie transport + urb / superficie TBV	1	1	1	TRUE	Finale
Indice de pression sur la TBV	Voir tableau indice de pression	1	1	0.5	TRUE	Finale
Densité de prélèvement dans la TBV	Densité point prelev / superficie TBV & Somme volume prelev / superficie TBV	1	1	2	TRUE	Finale
Pression liées aux STEP	A définir	?	?	?	?	A venir

Illustration 15: Grille d'indicateurs à l'échelle TBV avec valeurs de pondération / enjeu

Attention : pour l'étude actuelle, les zones humides et haies efficaces qui sont toutes deux issues des données DEC²⁰ n'ont pas été utilisées car ces derniers ne sont pas terminés sur l'ensemble du bassin. Ces données seront réintégrées lors de l'utilisation de cette méthodologie TBV lors de la future révision du SAGE, ces inventaires devant être, d'ici là, terminés, validés par les communes et intégrés par l'EPTB.

9.2.3 Indice de pression (ou pression liée à l'occupation du sol)

L'indice de pression caractérise l'occupation du sol liée à des valeurs de pondération qui sont fonction de leur effets sur les milieux aquatiques. Les indices de pressions ci-dessous ont été validés par des comités d'experts sur des bases bibliographique et scientifique. Cette expertise comporte inévitablement des biais (pas d'information sur des parcelles en agriculture biologique, surfaces imperméabilisées comprenant les parcs et jardins des villes, rotation des cultures [Une année en fourrage, l'autre en culture], pas de différence entre les typologies de cultures...). Toutefois, les informations apportées sont jugées exploitables en l'état et permettent de désigner convenablement l'occupation du sol du territoire. Dans le chapitre suivant nous verrons comment constituer un référentiel d'occupation du sol.

Indice	Pondération
Forêts de feuillus et mixtes et landes ligneuses	0.1
Surfaces toujours en herbe (Prairies permanentes, landes et broussailles)	0.2
Prairies temporaires, fourrages, forêts de conifères et peupleraies	0.3
Surfaces urbaines perméables (espaces verts, parcs et jardins)	0.4
Cultures	0.5
Maraîchage, arboriculture et vignes	0.7
Surfaces imperméabilisées	1

Illustration 16: Tableau détail des indices de pression et pondérations associées

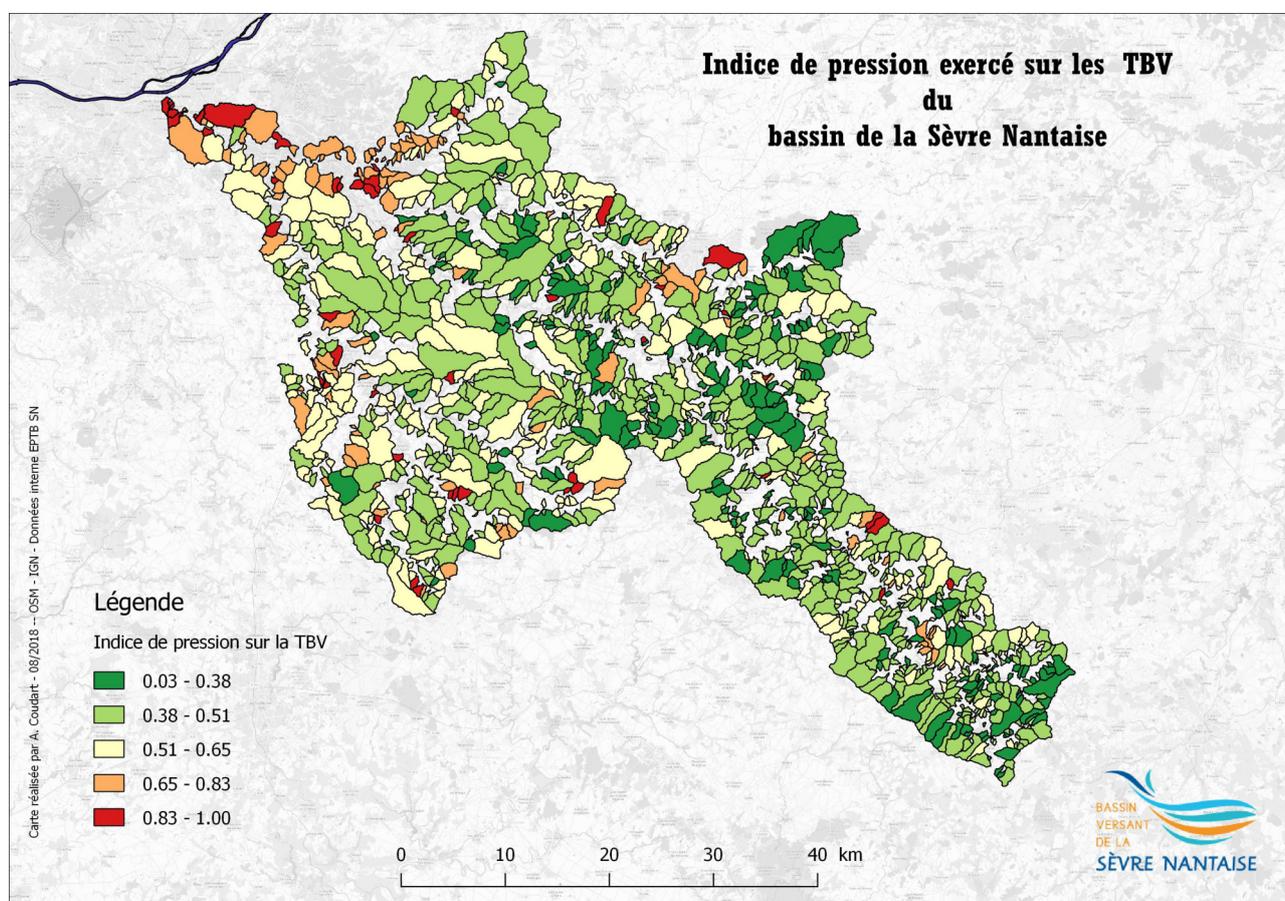


Illustration 17: Carte de l'indice de pression à l'échelle des TBV pour le bassin de la Sèvre Nantaise

L'ensemble des scripts SQL de caractérisation sont disponibles en annexes 7.

Un tableau listant les indicateurs, les sources de donnée, les formules, les id_param et d'autres variables est également disponible en annexe 1.

9.2.4 Couche composite cohérente d'occupation du sol

L'occupation du sol est une donnée majeure en matière d'état écologique. Une surface artificialisée ne permettra plus aux milieux naturels d'assurer leurs fonctionnalités originelles. Ainsi pour caractériser l'état des têtes de bassin versant il convient d'en apprécier leur couverture. Pour cela Il nous a fallu croiser les données d'occupation urbaine, des tronçons routiers, des tronçons de voies ferrées, les zones de cultures, les zones de prairies permanentes, les surfaces boisées, les surfaces en eau...

Il existe plusieurs sources de données d'occupation du sol, la plus connue à l'échelon européen est [Corine Land Cover](#). Cette donnée possède l'avantage d'être homogène et cohérente à cette échelle européenne mais son échelle d'utilisation est à 1/100 000e, ce qui lui confère peu d'intérêt en matière d'analyse locale.

La carte d'occupation du sol du [pôle Theia](#) propose une couverture à l'échelle nationale avec une résolution de 10 mètres pour les années 2016 et 2017. Ce pôle est une infrastructure de données satellitaire sous la gouvernance d'un comité directeur et d'un comité scientifique composés par les grandes institutions scientifiques nationales, comme le CNES, l'IGN, l'INRA, le CNRS, l'IRSTEA, Météo-France, le CEREMA. Les données sont produites à partir d'images satellite du programme Copernicus Sentinel 2 (résolution à 10 mètres) pour les dernières années (2016 / 2017) et Landsat 5 (résolution à 30 mètres) pour les millésimes de 2009 - 2011 et 2014.

Les données seront plus précises qu'avec CLC, mais toujours relativement peu précise à grande échelle, par exemple pour l'étude d'impact sur une bande riveraine d'un cours d'eau sur 20 mètres de large, la précision n'est pas suffisante.

[L'OCS GE](#), ou le référentiel à grande échelle concernant l'occupation du sol, est éditée par l'IGN et fournit une donnée précise, homogène sur le territoire national, en cohérence avec le RGE et millésimée. Toutefois, elle n'est pas encore disponible sur l'ensemble du territoire, en ce qui concerne le bassin versant de la Sèvre Nantaise (le département des Deux-Sèvres manque à l'appel). C'est la source de donnée la plus précise, mais partielle dans notre cas.

Enfin il est également possible de construire son propre référentiel d'occupation du sol, c'est ce que nous avons entrepris. Pour cela nous avons procédé par étapes successives.

9.2.4.1 Zones urbaines

Pour construire la couche zone urbanisée, nous avons utilisées différentes données issues de la Bd Topo de l'IGN :

- Bâti industriel
- Bâti remarquable
- Bâti indifférencié
- Gare
- Surface route (parkings, places, les routes étaient décrites dans une autre source de donnée qui sera exploitée ultérieurement...)

La méthode utilisée est celle éditée par [l'AGRESTE de Franche Comté](#) et que l'on retrouve dans de nombreuses études sur les têtes de bassins versant (cf. SYLOA, Vilaine...). La logique suivie est celle de la dilatation des éléments (buffer 50 mètres), de leur agrégation puis de l'érosion de l'élément composite (buffer -50m).

Le script de construction de la table zone urbaine est disponible en annexe 8.

9.2.4.2 Zones urbaines perméables

Les zones urbaines perméables ont été constituées avec la même logique que pour la table précédente en utilisant les données piste aéroport et terrain de sport de la bd topo de l'IGN. L'une des couches d'occupation du sol idéal aurait été l'OCS GE de l'IGN qui permet de distinguer le bâti des jardins publics ou privatif.

9.2.4.3 Réseaux routiers et voies ferrées

Le réseau routier et les voies ferrées proviennent toutes les deux de la BD topo de l'IGN. Initialement ces données polygones ont été converties en polygones. Les largeurs appliquées sont celles contenues dans la table attributaire, si la largeur des routes n'est pas indiquée, une valeur par défaut de 2 mètres a été appliquée pour les chemins et routes empierrées et de 5 m pour toutes les autres routes. Les routes de nature "bac" ont été retirées car elles n'ont pas de représentation physique sur le terrain.

Les **voies ferrées** sur le territoire ont toutes une largeur dite normale, c'est-à-dire environ 1,5 mètres qui correspondent à l'écartement des rails. Sachant que les voies sont dans la majorité des cas au moins au nombre de deux et en comptant les ballastes et poteaux supportant les caténaires et après mesure à différents endroits via photo satellite il apparaît qu'une **largeur moyenne de 15 mètres** semble pertinente.

9.2.4.4 Boisements

Les boisements sont issus de la [bd forêt v.2](#), ils sont intégrés tels quels dans la table d'occupation du sol locale.

9.2.4.5 Surfaces agricoles

Les éléments agricoles sont issus du [RPG 2016](#), en particulier le registre parcellaire (et non par îlots) de ce dernier. Ces données sont issues des déclarations des exploitants agricoles à la PAC (Politique Agricole Commune). N'y seront donc présentes que les parcelles qui font l'objet d'une déclaration, par exemple les usages agricoles "de loisir" seront absents (prairie équine de particulier, micro exploitation, exploitation agricole ne demandant pas d'aides européennes...). La donnée d'origine comprend 246 typologies différentes regroupées en 23 groupes, le tableau ci-dessous propose un regroupement en grandes catégories pour en faciliter l'exploitation ultérieure.

Libellé	Code groupe
Culture	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 14 - 15 - 24
Fourrage et rotation	11 - 16 - 19
Maraîchage / fleurs	25
Vignes	21
Vergers	20
Prairies permanentes et landes	17 - 18
Divers	28

Illustration 18: Proposition de regroupement des codifications RPG en entités homogènes

9.2.4.6 L'OCS Théia

Pour compléter les données issues de l'IGN, nous avons fait le choix d'utiliser le travail du pôle Théia. La codification de l'OCS Théia est consultable [ici](#). Ci-dessous un tableau regroupant les typologies utilisées dans notre table d'occupation du sol composite.

Code	Nature
11	culture été
12	culture hiver
31	forêt feuillus
32	forêt conifères
34	pelouses
36	lande ligneuse
41	bâti dense
42	bâti diffus
43	zones ind et com
44	surface route
45	surfaces minérales
46	plages et dunes
51	eau
53	glaciers ou neige
211	prairie
221	verger
222	vigne

Illustration 19: Codification de l'OCS Théia

9.2.4.7 Assemblage des différents types d'occupation du sol

Ces données d'occupation du sol pouvant être utiles pour de nombreux besoins, il est convenu de les réunir en une seule et même table. La table a été construite avec les références citées ci-dessus et en comblant les manques avec l'OCS du pôle Theia :

Règle de priorité : chaque couche ajoutée est découpée par les précédentes suivant l'ordre suivant.

1. Réseau routier et ferré
2. Surface en eau
3. Boisement
4. RPG
5. Urbain perméable
6. Urbain imperméable
7. OCS Théia

La table est composée des champs suivants :

- gid : identifiant unique, clé primaire
- data_theme : Thème principal (ex : Urbain)
- data_subtheme : Sous thème (ex : Perméable)
- data_utility : Utilité de la donnée (ex : Etude TBV)
- source_schema : schéma source donnée origine
- source_table : table source donnée origine

- `source_id` : identifiant source donnée origine
- `source_data` : Json regroupant l'intégralité des données de la source d'origine (champs, valeur)
- `data_create_query` : Requête SQL de création de donnée
- `data_insert_date` : Date d'insertion originel de la donnée dans l'OCS général
- `data_update_date` : Date de mise à jour de la donnée
- `geom` : Géométrie de la couche

Cette organisation nous permet de tracer la donnée d'origine et les requêtes en sont facilitées.

9.2.4.8 Assemblage final de la table OCS bvsn

Le script d'assemblage est disponible en annexe 9.

9.2.4.9 Principes de requêtes sur la table d'occupation du sol composite

- Les « **data_theme** » commençant par une Majuscule sont issu des données IGN
- Les « **data_theme** » commençant par une minuscule sont issu des données de l'OCS Théia
- Les données issues de l'IGN conservent l'ensemble de leurs champs et attributs. Ils sont stockés dans le champ « **source_data** » sous forme de Json et sont requêtables en SQL, c'est-à-dire qu'il est possible, par exemple, à partir du « **data_theme** » *Urbain* de ne sélectionner que les zones industrielles ou dans le « **data_theme** » *Boisement* de sélectionner les zones où le chêne est présent.

Occupation du sol sur le bassin de la Sèvre Nantaise

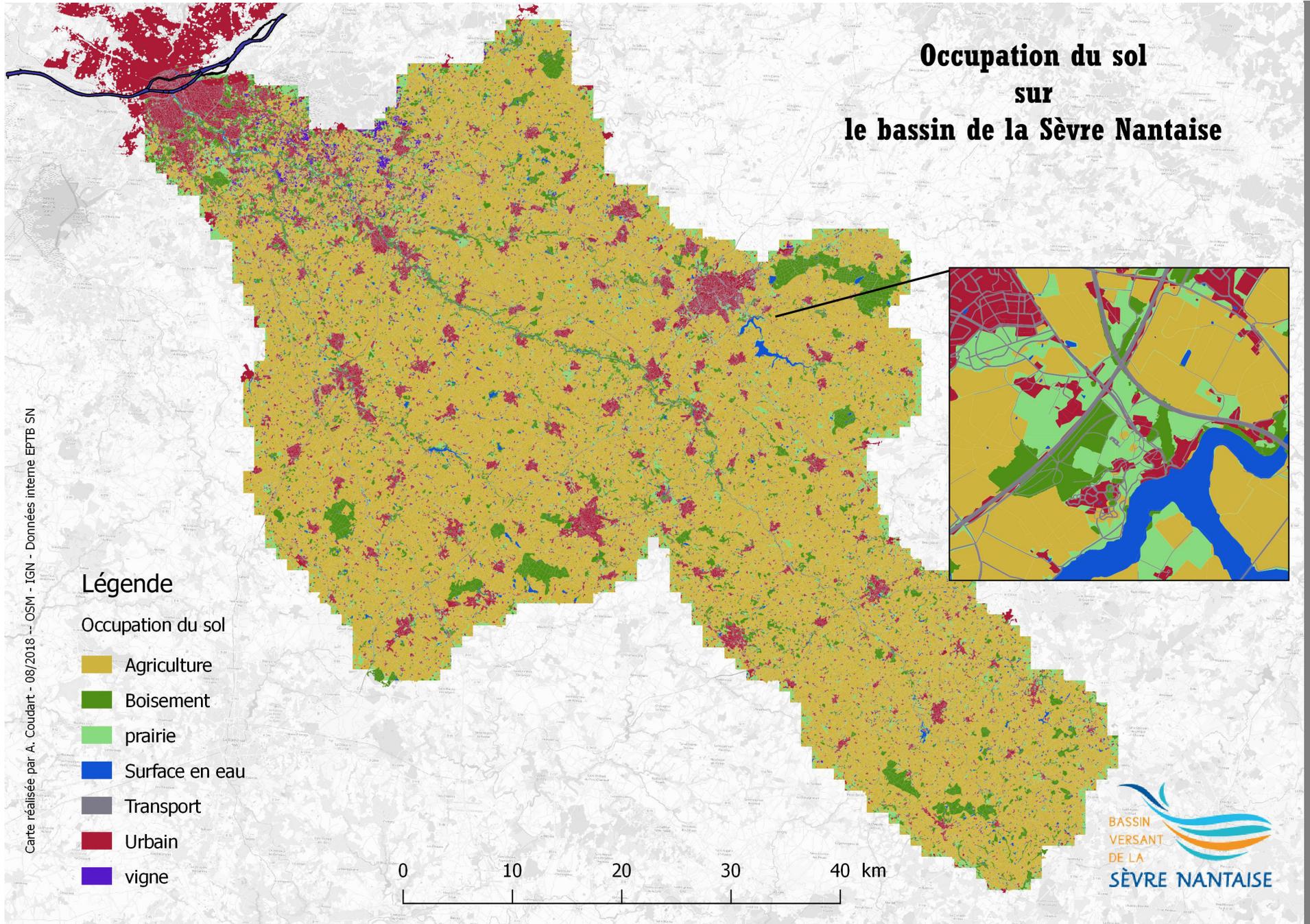


Illustration 20: Carte de l'occupation du sol sur le territoire de la Sèvre nantaise, étendu à 2 km

9.2.5 Les prélèvements en eau

Les prélèvements en eau constituent un indicateur intéressant en matière de pression sur les milieux aquatiques. Une donnée est disponible à l'échelle départementale, la Banque Nationale des Prélèvements en Eau, elle est gérée par le BRGM sur la base des informations communiquées par les agences de l'eau. Afin d'évaluer cette pression, deux méthodes étaient possibles, choisir un critère de densité de points de prélèvements par TBV, information qui nous semblait peu pertinente, ou un critère de volume prélevé par TBV. Cette information étant contenue dans la donnée de la BNPE, nous avons choisi de l'exploiter. Pour cela nous avons constitué une table regroupant les 4 départements. La requête quant à elle ne se fait que sur les prélèvements industriels et agricoles. Les captages d'eau potable ont été exclus car dans le cas du bassin de la Sèvre Nantaise, ces captages en eaux superficielles sont essentiellement présents sur les cours d'eau principaux et donc sans intérêt pour la caractérisation des têtes de bassin. À noter que la donnée est fournie avec plusieurs échelles temporelles, il est ainsi possible de calculer l'évolution du prélèvement sur les dernières années. Toutefois nous n'avons pas utilisé cette fonctionnalité.

9.2.6 Les stations d'épurations

L'impact des stations d'épuration sur les milieux aquatiques est notable, mais très hétérogène. En fonction de la capacité de l'installation, de son rendement pour chaque type de rejets (dépendant du type de filière et de sa vétusté éventuelle), mais également du débit du milieu récepteur, l'impact pourra être majeur ou faible. Si l'étude des têtes de bassin versant montre que des travaux hydromorphologiques doivent être conduits sur un secteur mais que ce dernier est également dégradé par une station d'épuration dysfonctionnant alors peut-être que le secteur devra faire l'objet de mise en conformité avant toute autre intervention.

L'EPTB mène actuellement une étude interne quant à l'impact de ces STEP sur son territoire (modélisation basée sur le logiciel NORRMAN). L'étude n'étant pas encore arrivée à terme, **nous avons fait le choix de ne pas inclure la donnée station d'épuration** dans l'étude des têtes de bassin versant, toutefois, à terme, lors de la future révision du SAGE, cette donnée devrait être disponible et sera alors intégrée aux calculs de caractérisation.

Pour tout renseignements concernant l'étude sur l'impact des stations d'épuration, vous pouvez contacter l'EPTB.

9.2.7 Les haies

Les haies participent du bon état écologique du territoire et a fortiori des milieux aquatiques. Leur utilité est multiple : zone d'habitat, filtre érosif, filtre d'éléments physico-chimiques, ombrage... Leur prise en compte s'avère indispensable pour évaluer l'état d'une tête de bassin versant. Cette donnée est disponible chez plusieurs fournisseurs. L'IGN et sa Bd topo zone de végétation propose une typologie « haie » sur l'ensemble du territoire. Les DEC nous proposent une information détaillée sur les haies, comme leur état, leur continuité, leur orientation par rapport à la pente... Malheureusement cette donnée n'est pas homogène sur l'ensemble du territoire. Enfin les fédérations de chasse ont menées des campagnes d'inventaire des haies et les proposent à leurs partenaires techniques. Malheureusement ces données ne sont pas homogènes sur le territoire tant en termes de couverture qu'en termes de qualité, de problèmes topologiques voire de projection.

Nous avons donc choisi d'utiliser en priorité la donnée fournie par l'IGN dans sa Bd topo en ne conservant que les éléments de typologie « haie », puis nous y avons ajouté par découpage (priviliégiant l'IGN) les données DEC lorsqu'elles étaient disponibles. Les éléments de l'IGN étant sous forme de polygones, pour y ajouter les données DEC en linéaires nous leur avons appliqué un tampon de 4 mètres correspondant ainsi à un houppier de 8 mètres. C'est cette donnée qui servira à calculer les taux de ripisylve sur bande riveraine.

Pour des raisons évidentes, les calculs de densité de haies sur les têtes de bassin versant sont réalisés uniquement sur les zones hors boisement.

Pour la **densité ripisylve**, qui désigne les boisements de bord de cours d'eau et dont l'analyse constitue un indicateur important, les opérations sont réalisées à partir de la même donnée mais sont inclus dans les calculs les zones boisées de feuillus et landes ligneuses.

9.2.8 Les plans d'eau et les mares

Les plans d'eau et en particulier ceux placés sur cours d'eau ont un impact négatif sur l'état des milieux aquatiques. Ils sont facteurs de réchauffement de l'eau, de stagnation défavorisant l'oxygénation, de colmatage du substrat et ainsi de banalisation des milieux, de rupture de continuité écologique sédimentaire et piscicole... Les plans d'eau hors cours d'eau peuvent également avoir un effet négatif particulièrement sur la bande riveraine car, ils captent une partie de la ressource en eau et la rendent plus facilement sensible à l'évapotranspiration et peuvent ainsi participer à l'accroissement des étiages. Pour cette étude nous avons choisi d'utiliser les données internes issues des différents inventaires, DEC, REH²¹, etc et jugées homogène sur le territoire.

Les mares à l'inverse sont considérées comme des indicateurs d'habitats positifs mais leur distinction des plans d'eau n'est pas si aisée. Nous avons considéré que des plans d'eau non connectés aux cours d'eau et d'une superficie de moins de 2 000 mètres carrés étaient des mares. La donnée utilisée est donc la même que pour les plans d'eau mais avec une sélection différentes des attributs.

9.2.9 Les cours d'eau enterrés

N'ayant pas de données précise sur le sujet, les cours d'eau enterrés ont été calculés grâce au champ « correction_context » de la table « t_cours_ref_ddtm_et_ign_corr_detail » utilisée pour la conception et correction du référentiel hydrologique. Cette table stocke dans ce champ l'origine du tracé, s'il provient du référentiel (DDT – IGN), du tracé théorique, ou d'un raccordement par buffer entre un tronçon orphelin et un tracé théorique.

Les cours d'eau considérés comme enterrés sont donc les cours d'eau issus soit des raccordements des tronçons orphelins au réseau, soit des parties de tronçon théorique. Il serait donc plus juste de parler de **cours d'eau enterrés théorique ou potentiels**, et c'est pourquoi ces derniers n'ont pas été pris en compte dans l'analyse actuelle.

9.2.10 Le stockage des résultats

L'approche utilisé pour cette étude se sert d'un modèle relationnel de donnée simple pour stocker les informations collectées. Nous avons vu précédemment la création de la table des têtes de bassin versant contenant les différentes géométries :

Table des géométries : t_tbv_01_without_caract

- Fonctions : stockage des géométries utiles des TBV et autres informations diverses
- Champs :

21 Réseau d'Évaluation des Habitats

gid = identifiant unique de la TBV et clé primaire
gid_graph_end = identifiant du dernier tronçon de cours d'eau formant la TBV
max_lenght_ce = longueur en mètres du chemin le plus longtemps
strahler_after_outlet = Rang de strahler du tronçon suivant l'exutoire de la TBV
strahler_before_outlet =
geom = géométrie de la TBV
geom_max_path = géométrie du chemin le plus long
geom_max_path_start_point = géométrie du point de départ du chemin le plus long
geom_max_path_end_point = géométrie du point d'arrivée du chemin le plus long
geom_ce = géométrie de l'ensemble des cours d'eau
geom_ce_buffer_10 = géométrie de l'ensemble des bandes riveraines (10 m) autour des cours d'eau

	gid [PK] bigint	gid_graph_end bigint	max_length_ce double precision	max_strahler_rank integer	strahler_after_outlet integer	strahler_before_outlet integer	geom geometry(MultiP	geom_max_path geometry(MultiLineStri	geom_max_path_start_pt geometry(Point,2154)	geom_max_path_end_pt geometry(Point,2154)
1	1	2	198.029044132731	1	7	7		01050000206A08000000	01010000206A0800009A999999C9BF	01010000206A080000CDCCCC
2	2	3	238.913765835672	1	7	7		01050000206A08000000	01010000206A0800000000000000C9BF	01010000206A08000000000000
3	3	4	1714.58802813623	1	7	7		01050000206A08000000	01010000206A08000001B8E8145BF	01010000206A08000033333333
4	4	6	226.654789571014	1	7	7		01050000206A08000000	01010000206A08000066666666664C	01010000206A0800009A999999
5	5	9	547.408160564671	1	7	7		01050000206A08000000	01010000206A0800008E7A0632C	01010000206A08000033333333
6	6	10	174.977846448916	1	7	7		01050000206A08000000	01010000206A08000033333333D7C	01010000206A0800009A999999
7	7	17	210.436750212752	1	7	7		01050000206A08000000	01010000206A08000000000000E4D	01010000206A080000CDCCCC
8	8	20	191.866529599744	1	7	7		01050000206A08000000	01010000206A0800006666666666D	01010000206A08000000000000
9	9	23	183.46820273199	1	7	7		01050000206A08000000	01010000206A0800000000000048D	01010000206A08000033333333
10	10	27	4373.59965255548	2	7	7			01010000206A08000057DFA3D5B4D	01010000206A080000CDCCCC
11	11	33	239.113286107054	1	7	7		01050000206A08000000	01010000206A080000CDCCCCC5CE	01010000206A08000000000000
12	12	35	688.624560723681	2	7	7		01050000206A08000000	01010000206A080000CDCCCCC70EF	01010000206A0800009A999999
13	13	39	354.116621485546	1	7	7		01050000206A08000000	01010000206A080000B53ACB54C3E1	01010000206A080000CDCCCC

Illustration 21: Table des géométries des TBV

C'est la table de base, celle autour de laquelle le modèle s'articule, c'est également la table qui sert de référence pour l'affichage et la visualisation cartographique. Elle ne contiendra aucun résultat issu des opérations de caractérisation ou de scénarisation quelconque.

Elle est reliée à une table de paramètre de caractérisation →

Table des paramètres de caractérisations : t_tbv_01_param_carac

- Fonctions : stockage des différents paramètres de caractérisation, soit les différents indicateurs de pression, état des TBV. Reliée directement à la table de résultats.
- Champs :

id_param = identifiant unique et clé primaire
nom_param = nom du paramètre
description_param = description du paramètre
positif = si positif la valeur est conservée tel quel sinon elle est inversée (0,8 devient 0,2)
path_process = chemin vers la fonction php
unité = unité du paramètre si il y a
source_data = données utilisées pour calculer le paramètres
commentaires = commentaires divers

	id_param [PK] serial	nom_param character varying(50)	description_param character varying(255)	positif boolean	path_process text
1	1321	Surface_bv	Calcul de la surface de la TBV	TRUE	/var/www/observ/processing/src/App/Process
2	1322	Perimetre_bv	Calcul du périmètre de la TBV	FALSE	/var/www/observ/processing/src/App/Process
3	1323	Pente_moyenne_tbv	Calcul la pente moyenne par TBV	FALSE	/var/www/observ/processing/src/App/Process
4	1324	Indice_gravelius	Calcul l indice de compacité des TBV selon la méthode de Gravelius -- l = forme ronde (peu r	FALSE	/var/www/observ/processing/src/App/Process
5	1325	Densite_cours_eau	Calcul la densité de cours d eau pour chaque TBV	FALSE	/var/www/observ/processing/src/App/Process
6	1326	Densite_point_bas	Calcul la densité de point bas pour chaque TBV, réalisé avec un mnt surecreusé et sinkless po	FALSE	/var/www/observ/processing/src/App/Process
7	1327	Taux_plan_eau_impact_ce	Calcul le linéaire de cours d eau impacté par des plan d eau par TBV	TRUE	/var/www/observ/processing/src/App/Process
8	1328	Taux_zh_connect_ce	Calcul le pourcentage de zh connecté à la bande riveraine des cours d eau par TBV	FALSE	/var/www/observ/processing/src/App/Process
9	1329	Taux_ripi_connect_ce	Calcul le pourcentage de ripisylve ou bande boisé en bord de cours d eau par TBV	FALSE	/var/www/observ/processing/src/App/Process
10	1330	Taux_mares_connect_ce	Calcul le pourcentage de mares connectées à la bande riveraine par TBV	FALSE	/var/www/observ/processing/src/App/Process
11	1332	Taux_impermeabilisation_tbv	Calcul le pourcentage de zones artificialisées sur chaque TBV	TRUE	/var/www/observ/processing/src/App/Process
12	1333	Taux_surface_plan_eau_tbv	Calcul le pourcentage surfacique de plan d eau sur chaque TBV	TRUE	/var/www/observ/processing/src/App/Process
13	1334	Taux_surface_zh_tbv	Calcul le pourcentage surfacique de zones humides sur chaque TBV	FALSE	/var/www/observ/processing/src/App/Process
14	1335	taux_impermeabilisation_ce	Calcul le pourcentage de surface imperméabilisée dans la bande enherbée sur chaque TBV	TRUE	/var/www/observ/processing/src/App/Process

Illustration 22: Table des paramètres de caractérisations

C'est l'ensemble des indicateurs de caractérisation et indice (elle attribue également aux scénarios un identifiant propre). C'est à partir de ces « id_param » que nous interrogerons la table des géométries pour connaître les valeurs de tel paramètre sur telle tête de bassin versant.

Elle est reliée à la table des résultats →

Table des résultats : t_tbv_01_with_result

- Fonctions : Stockage des résultats de caractérisation et pondération. Reliée directement à la table des geom des TBV et à la table paramètres de caractérisation.
- Champs :

id_tbv_param = identifiant unique et clé primaire

gid_tbv = clé étrangère renvoyant vers le gid de la table des geom

id_param = clé étrangère renvoyant vers id_param de la table des paramètres de caractérisations

valeur_type = type de la valeur (brut, centrées / réduite, résultat de pondération)

valeur = valeur résultat

date_calcul = date du calcul

stat_moyenne = moyenne des résultats des calculs pour une date et un paramètre sur l'ensemble des TBV

stat_ecart_type = écart type des résultats des calculs pour une date et un paramètre sur l'ensemble des TBV

valeur_decompose = décomposition du résultat s'il est issu d'un croisement (par exemple l'indice de pression)

stat_nb_valeur = nombre de résultats pour un calcul sur un paramètre (le nombre de TBV concerné par le calcul)

stat_valeur_origine_position_borne = pour les valeurs centrer / réduite, informe si la valeur était dans les borne ou inférieure ou supérieure (cf. partie statistique)

stat_valeur_origine = dans le cas d'une valeur issue d'un calcul, reporte la valeur d'origine.

id_tbv_param [PK] serial	gid_tbv integer	id_param integer	valeur_type character varying(510)	valeur numeric	date_calcul date	stat_moyenne numeric	stat_ecart_type numeric	valeur_decompose text	stat_nb_valeur numeric	stat_valeur_origine_position_borne character varying(255)	stat_valeur_origine numeric
148125	808	1324	Brute_centrage_reduction_bornage	0.681356	2018-08-14	1.8525814408	0.231697531132		1277	ok	2.1047
148126	978	1324	Brute_centrage_reduction_bornage	0.401177	2018-08-14	1.8525814408	0.231697531132		1277	ok	1.7152
148127	195	1324	Brute_centrage_reduction_bornage	0.255585	2018-08-14	1.8525814408	0.231697531132		1277	ok	1.5128
148128	233	1324	Brute_centrage_reduction_bornage	1.000000	2018-08-14	1.8525814408	0.231697531132		1277	sup	3.8098
148129	1161	1324	Brute_centrage_reduction_bornage	0.743649	2018-08-14	1.8525814408	0.231697531132		1277	ok	2.1913
148130	235	1324	Brute_centrage_reduction_bornage	0.700993	2018-08-14	1.8525814408	0.231697531132		1277	ok	2.1320
148131	1074	1324	Brute_centrage_reduction_bornage	0.513177	2018-08-14	1.8525814408	0.231697531132		1277	ok	1.8709
148132	681	1324	Brute_centrage_reduction_bornage	0.464406	2018-08-14	1.8525814408	0.231697531132		1277	ok	1.8031
171114	531	1355	[scenar_pond : indice_pression_sy	15.27468	2018-08-14			Indice_pression			
171115	910	1355	[scenar_pond : indice_pression_sy	8.439062	2018-08-14			Indice_pression			
171116	1295	1355	[scenar_pond : indice_pression_sy	11.96468	2018-08-14			Indice_pression			
171117	175	1359	[scenar_pond : indice_pression_sy	8.468125	2018-08-14			Indice_pression			
171118	368	1359	[scenar_pond : indice_pression_sy	6.094062	2018-08-14			Indice_pression			
171119	891	1355	[scenar_pond : indice_pression_sy	12.19187	2018-08-14			Indice_pression			
171120	70	1359	[scenar_pond : indice_pression_sy	6.529375	2018-08-14			Indice_pression			
171121	685	1359	[scenar_pond : indice_pression_sy	7.701875	2018-08-14			Indice_pression			

Illustration 23: Table des résultats

C'est la table des résultats, ici sont stockées l'ensemble des valeurs résultant d'opérations, que ce soit les opérations de caractérisation, de scénarisation pondérée, de centrage réduction... Elle possède une clé étrangère faisant référence à la table des géométrie et une autre à la table faisant référence au paramètre de caractérisation permettant ainsi la jonction entre tous les éléments. Chaque clé primaire est indexée pour optimiser les temps de traitements.

L'ensemble du script de génération du modèle de donnée est disponible en annexe 10 ainsi que le schéma le résumant en annexe 11.

10 Hiérarchisation des têtes de bassin versant

La hiérarchisation des têtes de bassin versant correspond à la volonté de classer les territoires pour prioriser les territoires les plus vulnérables.

Cette partie de l'étude s'appuie sur une pondération des indicateurs d'état, de pression et de sensibilité. Les traitements ont été réalisés suivant une méthode élaborée par le SYLOA avec quelques ajustements en fonction de la disponibilité des données et de choix interne (par exemple, nous avons pris la décision suite aux retours de l'étude menée sur la Vilaine de ne pas prendre en compte l'indicateur d'intersection des voies de transport avec les cours d'eau. En effet ce dernier n'est pas corrélant dans moins de 10 % des cas avec les données terrain et ce, suivant plusieurs méthodologies).

À l'avenir, l'EPTB pourra développer ses propres scénarios en ajoutant ou retirant certains critères ou en modifiant les valeurs de pondération. Ce travail pourra être mené en concertation avec les élus, agents et partenaires de l'EPTB.

En effet, c'est à ce stade du projet que les décisionnaires du bassin versant pourront établir des éléments de priorisation selon les enjeux, les territoires, les moyens humains et financiers mobilisables... Cette démarche d'animation et de concertation dépendra ensuite du niveau d'ambition des élus du syndicat.

Car si les têtes de bassin versant représentent 75 % du territoire, il n'est pas envisageable d'intervenir sur la totalité. C'est d'ailleurs l'intérêt de la démarche de hiérarchisation : mettre en lumière certains dysfonctionnements locaux pour y apporter des réponses ciblées et ainsi mieux comprendre l'état des masses d'eau dont les structures ont la charge.

Il s'agira donc de choisir une part de ce territoire à favoriser pour des interventions. Faudra-t-il intervenir sur les zones urbanisées ? Faudra-t-il intervenir sur les zones les plus gravement atteintes ou au contraire sur des zones intermédiaires où de petites interventions permettraient de faire basculer un état moyen en bon état ? Ce sont autant de questions auxquelles devront répondre les élus de la Sèvre Nantaise.

10.1 Homogénéiser les données – centrer / réduire / borner

Si chaque résultat nous apporte des informations pertinentes et intéressantes, que faire pour les assembler et en déduire une notation selon une multitude d'indicateurs ? En effet, certains indicateurs représentent des surfaces, d'autres des volumes, d'autres une densité, d'autres encore sont déjà indices. C'est à ce moment que les outils statistiques prennent le relais.

Pour pouvoir être comparées, les données doivent être homogénéisées. Nous avons choisi de traiter les valeurs brutes individuelles suivant la méthode centrage, réduction, bornage (et ramener entre 0 et 1).

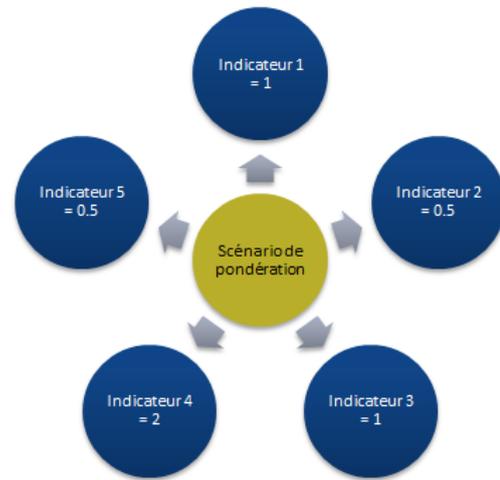
- Le centrage : On retranche à chaque valeur la moyenne de la distribution, ce qui revient à modifier l'origine, plaçant ainsi la moyenne sur le point 0 de l'axe des abscisses.
- La réduction : On divise la valeur par l'écart-type de la distribution.
- Le bornage : On ramène les valeurs extrêmes (au-delà de 3 écarts-types) à la frontière du 3e écart-type. Cette information est stockée dans le champ « stat_valeur_origine_position_borne » de la table de résultats, Si la valeur était extrêmement faible, il sera indiqué **inf**, dans le cas inverse **sup**, sinon **ok**.

Les données centrées / réduites / bornées sont reconnaissables dans le champ valeur type par un attribut se terminant par **[FOR COMP]** au contraire des valeurs brutes indiquées comme **Brute**.

Le script SQL réalisant l'opération est disponible en annexe 23.

10.2 Assembler les indicateurs – les scénarios de pondérations

Une fois homogénéisées, les données deviennent comparables. C'est donc ici qu'interviennent les scénarios de pondération. Ces derniers se comportent de manière à lier un indicateur avec un coefficient qui servira à multiplier la valeur centrée / réduite afin de pondérer cette dernière suivant un scénario quelconque. Ces derniers peuvent être aussi bien des enjeux tels que l'enjeu qualité d'eau ou un indice de pression global regroupant l'ensemble des indices de pression.



Les scénarios sont stockés dans une table du modèle de données à part :

Table des scénarios de pondération : t_tbv_01_scenar_ponde

- **Fonctions** : stockage des scénarios de pondération, c'est-à-dire l'ensemble des liaisons paramètres > valeurs. Reliées directement à la table de résultats, à la table de paramètres de pondération et à la table de paramètres de caractérisation. Le résultat des scénarios est enregistré comme étant un paramètre de caractérisation.
- **Champs** :

nom_scenar = identifiant unique et clé primaire

description_scenario = description du scénario

ordre = champ qui permettra d'organiser l'ordre dans lequel les calculs pourront être faits si besoin.

table_dest = table où est enregistrée le résultat

res_id_param = clé étrangère renvoyant vers le paramètre de caractérisation

filter_end_valeur_type = type de valeur (brut ou résultat de calcul)

nom_scenario [PK] character varying(255)	description_scenario character varying(255)	ordre numeric(5,2)	table_dest text	res_id_param integer	filter_end_valeur_type text
indice_pression_syloa_br	bande riveraine	1.00	m060_milieux_cor	1359	Brute
indice_pression_syloa_tbv	têtes de bv	1.00	m060_milieux_cor	1355	Brute
qe_syloa	ponderation syloa concernant la qualite des eaux	2.00	m060_milieux_cor	1361	[FOR_COMP]
qm_syloa	ponderation syloa concernant la qualite des milieux	2.00	m060_milieux_cor	1360	[FOR_COMP]
qte_syloa	ponderation syloa concernant la gestion quantitative	2.00	m060_milieux_cor	1362	[FOR_COMP]
sensibilite_syloa	ponderation de la sensibilite physique et morphologique des TBV	3.00	m060_milieux_cor	1363	[FOR_COMP]
qe_vulnerabilite	calcul de la vulnerabilite des TBV sur l enjeux qualite d eau	2	m060_milieux_cor	1366	[FOR_COMP]
qm_vulnerabilite	calcul de la vulnerabilite des TBV sur l enjeux qualite des mil:	2	m060_milieux_cor	1364	[FOR_COMP]
qte_vulnerabilite	calcul de la vulnerabilite des TBV sur l enjeux quantite d eau	2	m060_milieux_cor	1367	[FOR_COMP]

Illustration 24: Table, scénario de pondération

Cette table fait référence à des associations indicateurs / coefficients de pondération que l'on retrouve dans une table spécifique →

Table des paramètres de pondérations : t_tbv_01_param_ponde

- Fonctions : stockage des différentes liaisons entre les paramètres de caractérisation et des valeurs de pondération. Reliée à la table de paramètres de caractérisation et à la table de scénario de pondération.
- Champs :

id_ponde = identifiant unique et clé primaire

id_param = clé étrangère en référence à l'id_param de la table des paramètres de caractérisation

valeur_param_ponde = valeur de pondération associée

data_create_query = date de création de l'association param > valeur

scenar = clé étrangère renvoyant vers l'appartenance de la liaison à un scénario de pondérations

	id_ponde [PK] integer	id_param integer	valeur_param_ponde numeric	data_create_query text	scenar character varying
1	1	1345	0.1		indice_pression_syloa_tbv
2	2	1346	0.1		indice_pression_syloa_br
3	3	1347	0.2		indice_pression_syloa_tbv
4	4	1348	0.2		indice_pression_syloa_br
5	5	1349	0.3		indice_pression_syloa_tbv
6	6	1350	0.3		indice_pression_syloa_br
7	7	1342	0.4		indice_pression_syloa_tbv
8	8	1343	0.4		indice_pression_syloa_br
9	9	1351	0.5		indice_pression_syloa_tbv
10	10	1352	0.5		indice_pression_syloa_br
11	11	1353	0.7		indice_pression_syloa_tbv
12	12	1354	0.7		indice_pression_syloa_br
13	13	1341	1		indice_pression_syloa_tbv
14	14	1344	1		indice_pression_syloa_br
15	22	1327	1		qm_syloa
16	24	1329	1		qm_syloa
17	25	1330	1		qm_syloa
18	26	1359	2		qm_syloa
19	27	1335	1		qm_syloa
20	28	1333	1		qm_syloa
21	30	1336	1		qm_syloa

Illustration 25: Table, paramètres de pondérations

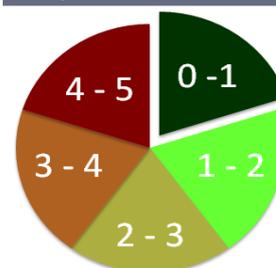
C'est donc cette table qui fait le lien entre un indicateur de caractérisation et un coefficient de pondération tirés d'un scénario. Lorsque que l'on souhaite visualiser dans un outil cartographique les résultats d'enjeu sur une tête de bassin versant, il s'agit donc des différents indicateurs pondérés (et de nouveau centrés / réduits / bornés) qui sont appelés par l'identifiant unique du scénario (dans la table de scénario).

Le script est disponible en annexe 24

10.3 Classer les résultats – la discrétisation

Enfin, pour visualiser les résultats nous avons choisi de les classer. Si l'on s'affranchissait de cette étape il serait compliqué de les interpréter. Cette action de classification raisonnée des données est appelée en statistique la **discrétisation**. Il serait tout à fait possible de s'en affranchir pour afficher les résultats selon un graduant de couleur, mais la classification possède des avantages dont nous comptons tirer profit : la simplification de la lecture, mais également la possibilité ultérieure d'attribuer à chaque classe une valeur qui pourra être utilisée pour croiser des tableaux de données (à voir dans le chapitre suivant).

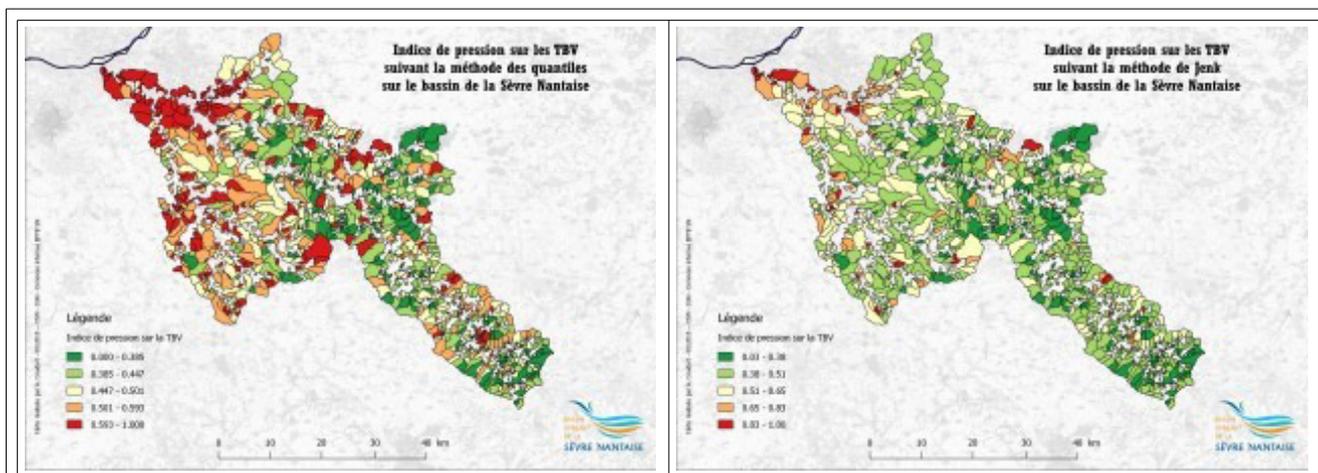
Enjeu Qualité d'Eau



L'action de classification n'est pas anodine, elle transforme profondément la donnée et la simplifie. Chaque méthode possède des avantages et des inconvénients. Les méthodes sont, en général, choisies en fonction de la distribution des populations. Les graphiques de distribution réalisés avec R pour chaque indicateur et présentés dans le chapitre concernant la caractérisation, nous ont permis de mettre en évidence leurs diverses formes. Si certains indicateurs sont fortement dissymétriques, ce n'est pas le cas des résultats sur les enjeux

Par exemple, les méthodes des **quantiles** ou des **quartiles** s'adaptent particulièrement bien à des distributions homogènes qui voient leur population répartie équitablement de part et d'autre de la moyenne. Ainsi, il sera possible de la classer suivant des parts égales.

En revanche, la méthode des seuils naturels (**Jenk** par exemple) convient parfaitement à des distributions dissymétriques.



Texte 1: Comparaison des résultats de discrétisation suivant les quantiles ou Jenk

L'exemple ci-dessous nous présente l'indice de pression subi par les têtes de bassin versant au regard de l'occupation du sol. À gauche, nous avons utilisé la méthode des quantiles qui répartit la population en classes de mêmes nombres d'individus. La méthode de droite emploie la méthode des seuils naturels de Jenk, c'est-à-dire que les bornes sont placées en fonction des ruptures visibles de répartition de la population.

Cet exemple illustre parfaitement le fait que les méthodes de classification influent sur la visualisation et l'appréhension des données. En fonction de la distribution des données, il convient d'employer des méthodes adaptées (comme nous l'avons vu plus haut dans l'étude). Si nous avons utilisé les seules valeurs brutes, il aurait été indispensable de choisir la méthode de Jenk. Le fait d'avoir centré, réduit et borné les valeurs, a eu pour conséquences de répartir les distributions selon un ordre plus symétrique. Ainsi il devient possible d'appliquer des méthodes statistiques de types quantiles ou quartiles.

10.4 La vulnérabilité

La classification des résultats des enjeux nous permet également de poursuivre les traitements vers ce que nous appellerons la **vulnérabilité**. La vulnérabilité est le résultat du croisement enjeu / sensibilité sur la TBV. Pour se faire nous avons opté pour un tableau de croisement comme les travaux sur la gestion des risques les utilisent régulièrement :

Enjeux	0 / 0,2	1	1	1	1	2
	0,20 / 0,40	1	2	2	2	3
	0,40 / 0,60	2	3	3	3	4
	0,60 / 0,80	3	4	4	5	5
	0,80 / 1	4	4	5	5	5

Un fort enjeu allié à une forte sensibilité sont synonymes d'une forte vulnérabilité

Pour définir la vulnérabilité, nous avons choisi de prioriser légèrement les valeurs d'enjeux sur les valeurs de sensibilité. Ainsi, une sensibilité forte mais se croisant avec un enjeu faible ne sera noté que d'un 2 (vulnérabilité faible) quand un enjeu fort sur une sensibilité faible sera noté d'un 4 (vulnérabilité forte).

Ces choix ont été pris en interne sans concertation. Il est évident qu'il serait préférable pour les travaux à venir de réunir un comité technique afin de faire des choix validés en groupe d'expert officiels.

Ce tableau est retranscrit dans une table PG qui sera requêtée pour obtenir le résultat final :

id_vulnerabilite [PK] serial	min_enjeu numeric	max_enjeu numeric	min_sensi numeric	max_sensi numeric	vulne_val integer
1	0	0.2	0	0.8	1
2	0	0.2	0.8	1	2
3	0.2	0.4	0	0.2	1
4	0.2	0.4	0.2	0.8	2
5	0.2	0.4	0.8	1	3
6	0.4	0.6	0	0.2	2
7	0.4	0.6	0.2	0.8	3
8	0.4	0.6	0.8	1	4
9	0.6	0.8	0	0.2	3
10	0.6	0.8	0.2	0.6	4
11	0.6	0.8	0.6	1	5
12	0.8	1	0	0.4	4
13	0.8	1	0.4	1	5

Illustration 26: Tableau des vulnérabilités

La dernière opération technique de l'étude consiste donc à requêter la table résultat sur la valeur d'enjeu et la valeur de sensibilité et à attribuer à l'identifiant de la tête de bassin versant la valeur de vulnérabilité correspondante.

Le script de la requête est disponible en annexe 22.

11 Résultats, critiques et perspectives

11.1 Résultats cartographiques

Les résultats sont mobilisés par Qgis via la base de donnée Postgres. Pour cela une vue matérialisée a été créée ([le script en disponible en annexe 25](#)). À la différence d'une vue classique, cette dernière duplique les données, ce qui améliore les performances et facilite les requêtes complexes. En revanche, la vue matérialisée ne met à jour les données que sur instruction et ne supporte pas la modification de la structure des tables références.

Les résultats obtenus ne tiennent pas compte d'un dernier niveau d'interprétation. En effet, il serait maintenant intéressant de croiser les territoires définis comme vulnérables et de les mettre en perspective avec les objectifs qui sont fixés à l'EPTB. Parmi lesquels nous pouvons citer :

- Les territoires à enjeux
- Les territoires vitrines
- Les ruisseaux à potentiels salmonicoles
- Les réservoirs biodiversités
- Les zones de protections spéciales (Natura 2000, réserve biologique intégrale, etc)
- ...

Enfin et en dernier lieu, les élus de l'EPTB devront se prononcer sur le niveau d'ambition qu'ils souhaitent apporter à chaque thématique. Au regard du positionnement de ce curseur politique, il sera alors possible de sortir une carte définitive des territoires prioritaires sur lesquels intervenir.

11.2 Critiques et perspectives d'amélioration

Le travail présenté ci-dessus, bien qu'arrivé à un certain niveau d'accomplissement possède de nombreuses perspectives d'améliorations.

11.2.1 Les indicateurs

Les indicateurs choisis pour l'étude, bien que relevant de discussions et de travaux d'experts sur les dernières années, sont critiquables. Tout d'abord, l'utilisation de données cartographiques n'implique pas forcément une corrélation terrain / bureau. À ce titre, les résultats de l'étude portée par l'EPTB Vilaine et mademoiselle Guégnard seront à suivre de près. La combinaison des indicateurs peut également être source d'interrogation. Nous constatons, par exemple, une prévalence de l'occupation du sol dans nos résultats. En effet, dans la méthode actuelle, ces données sont mobilisées pour le calcul de l'indice de pression (à la fois sur la bande riveraine et les têtes de bassin versant), ainsi que pour l'indicateur du taux d'artificialisation (à ces deux échelles également), le tout pondéré avec une valeur de deux (c'est-à-dire le coefficient le plus élevé après les cours d'eau enterrés). Ceci ne manque pas de sur-représenter les zones urbaines comme des territoires vulnérables.

Le calcul de la vulnérabilité peut également être sujet à caution. En effet, la grille de croisement enjeu/sensibilité élaborée dans le cadre du présent travail nécessitera d'être confrontée aux retours des acteurs du territoire ce qui n'a pu être réalisé pendant la durée du stage.

Il sera important et nécessaire de réunir un comité technique territorial sur l'ensemble de ces sujets afin de statuer sur ces choix et d'entériner une méthodologie validée officiellement.

11.2.2 Les données

Les données peuvent également être critiquées. D'une part, **le référentiel du réseau hydrographique** utilisé a dû être constitué par agrégat et certains départements comme la Vendée (en cours en 2018) et les Deux-Sèvres (non débuté) ne possèdent pas de référentiel complet validé par les élus de la CLE. Si certains cours d'eau devaient être déclassés ou au contraire ajoutés, la carte des têtes de bassins versant s'en verrait transformée ainsi que l'ensemble des résultats de caractérisations.

Les **données indicatrices de biodiversités** sont également très peu représentées et seraient pourtant des vectrices d'informations très intéressantes. Les Diagnostics Environnementaux Communaux, ne sont également pas uniformes sur le territoire.

Les **données sur la puissance spécifique** du cours d'eau font référence à des fonctions de calculs hydrauliques et ne prennent malheureusement pas en compte des débits précis par tronçons, ni les largeurs plein bord et profondeur de cours d'eau, qui seraient des informations d'une grande utilité pour connaître l'identité hydromorphologique d'un cours d'eau. À l'avenir, les travaux d'inventaires pendant les études pourraient contenir un certain nombre d'indicateurs à relever sur le terrain qui pourraient alimenter petit à petit la base de connaissance.

Les **données sur les stations d'épurations** n'ont pu être utilisées pour cette étude mais devraient être disponibles avant la révision du SAGE.

Les données sur l'**occupation du sol** sont également partielles. Lorsque l'ensemble du territoire sera couvert par l'OCS GE de l'IGN, il sera possible de distinguer les zones urbaines perméables des zones urbaines imperméables. Les données du RPG restent quant-à elles la meilleure référence car très complètes. Pour autant, une information sur les parcelles cultivées en agriculture biologique serait une information des plus pertinentes.

Les **données REH** de SYSMA sont assez peu utilisées, un effort d'homogénéisation sur le territoire pourrait permettre de mieux les utiliser.

Évidemment, pour un ensemble d'indicateurs on se heurte à une difficulté d'ordre opérationnelle, il est en effet très coûteux d'inventorier finement un territoire de plus de 2000 km².

11.2.3 L'affinement de la priorisation

Comme précisé à la fin du chapitre sur la vulnérabilité. Il sera nécessaire de poursuivre le travail au-delà de cette dernière en abordant les objectifs intrinsèques au territoire.

11.2.4 La visualisation

Enfin, en matière de visualisation des données et des résultats, d'importantes perspectives s'offrent à l'avenir. Voici une liste de points intéressants :

- Possibilité de visualiser sur une page Web l'ensemble des données d'une tête de bassin versant (les données brutes, travaillées, les résultats de calculs d'enjeux...).
- Visualisation sous forme de graphiques (tant de pourcentages de forêt pour tant de pourcentages d'artificialisation, etc.)
- Possibilité de comparaison avec l'ensemble du territoire ou un espace particulier.
- Fléchage Territoire → Actions prioritaires (en fonction des enjeux et vulnérabilité constatés).

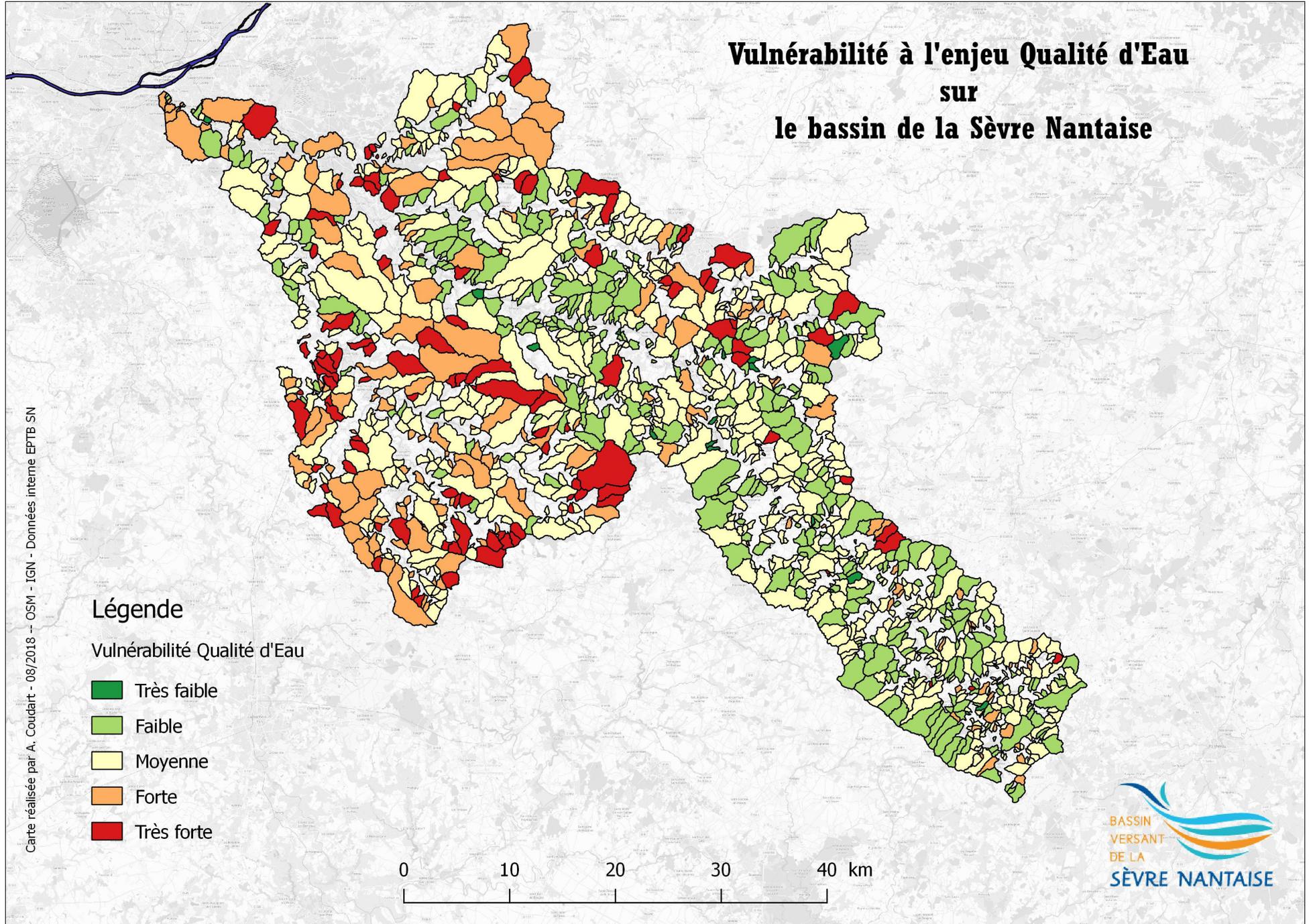
Enjeu Qualité d'Eau sur le bassin de la Sèvre Nantaise

Légende

Enjeu Qualité d'Eau



Vulnérabilité à l'enjeu Qualité d'Eau sur le bassin de la Sèvre Nantaise



Carte réalisée par A. Coudart - 08/2018 -- OSM - IGN - Données interne EPTB SN

Légende

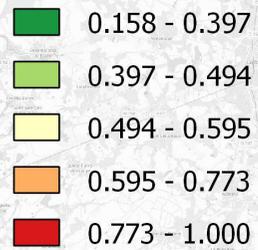
Vulnérabilité Qualité d'Eau

- Très faible
- Faible
- Moyenne
- Forte
- Très forte

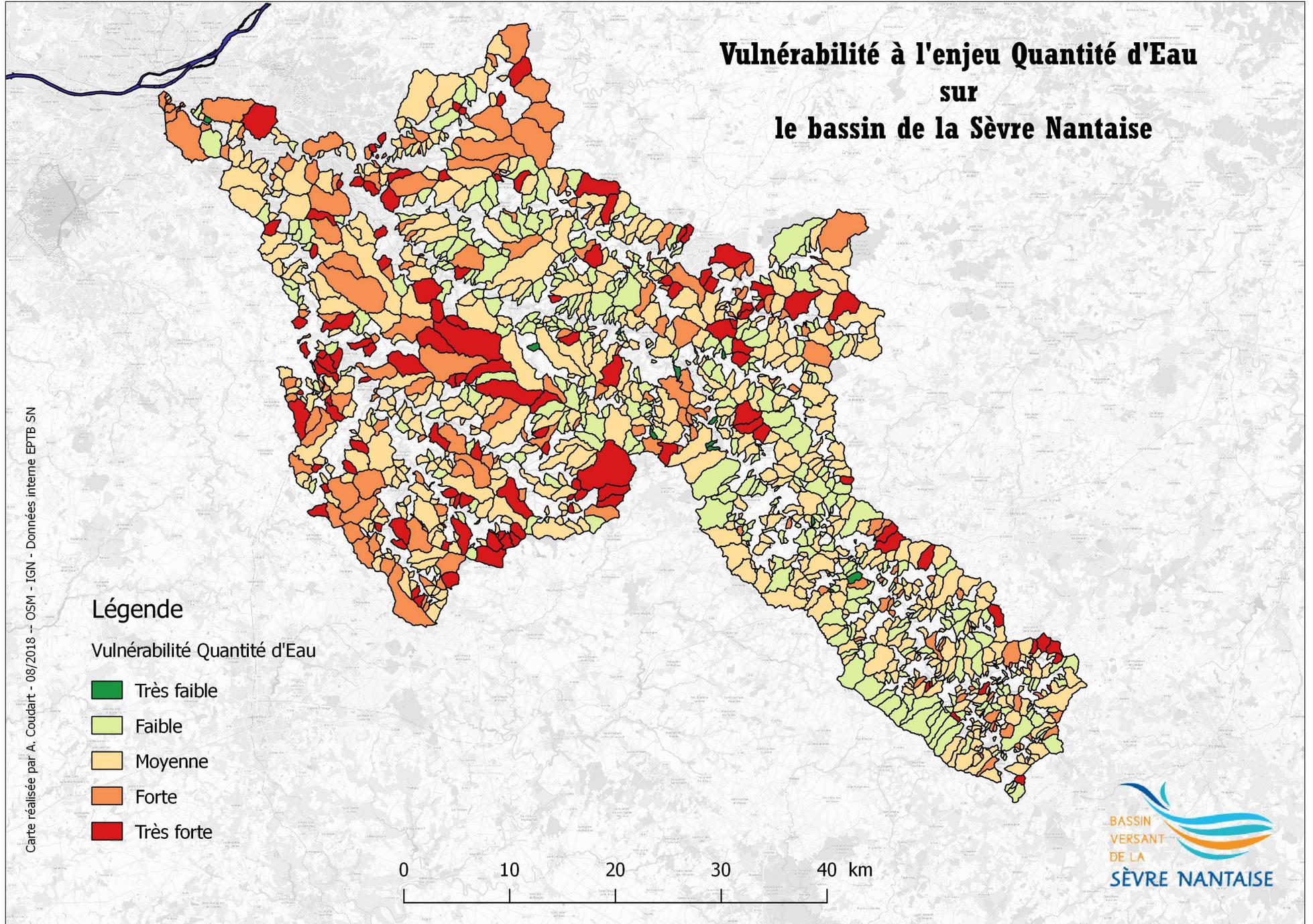
Enjeu Quantité d'Eau sur le bassin de la Sèvre Nantaise

Légende

Enjeu Quantité d'Eau



Vulnérabilité à l'enjeu Quantité d'Eau sur le bassin de la Sèvre Nantaise



Légende

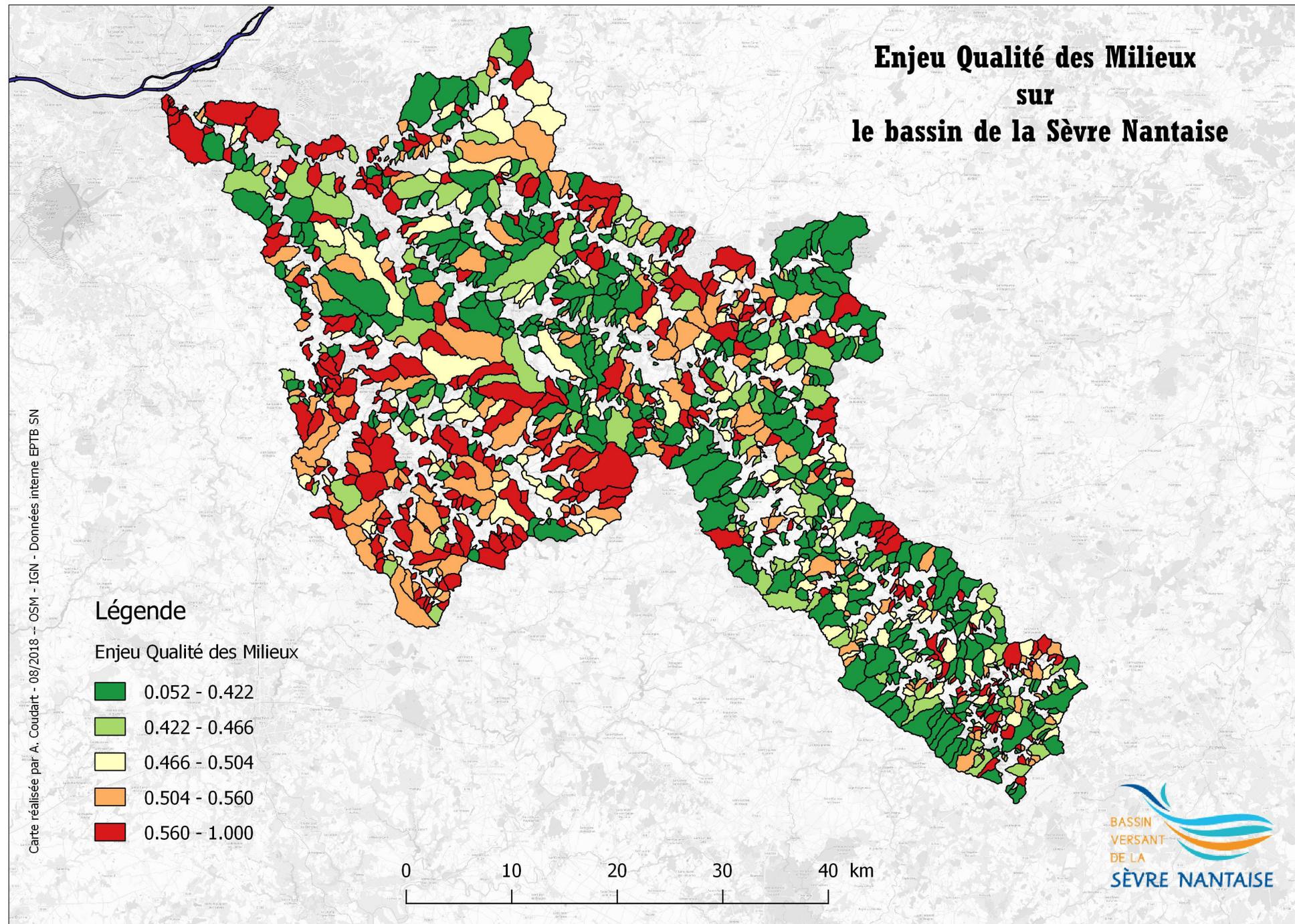
Vulnérabilité Quantité d'Eau

- Très faible
- Faible
- Moyenne
- Forte
- Très forte

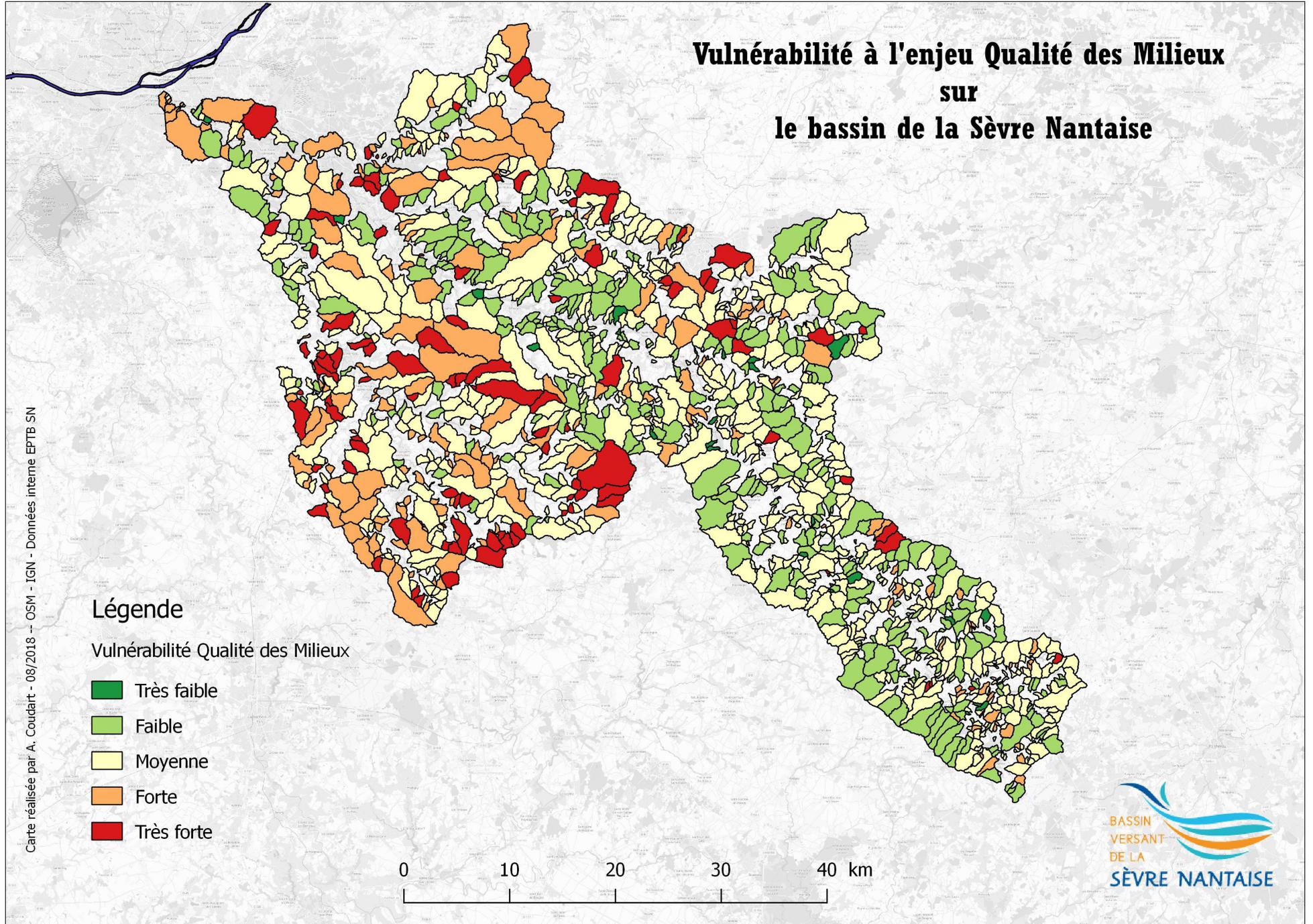
Carte réalisée par A. Coudart - 08/2018 -- OSM - IGN - Données interne EPTB SN



Enjeu Qualité des Milieux sur le bassin de la Sèvre Nantaise



Vulnérabilité à l'enjeu Qualité des Milieux sur le bassin de la Sèvre Nantaise



Légende

Vulnérabilité Qualité des Milieux

- Très faible
- Faible
- Moyenne
- Forte
- Très forte

Carte réalisée par A. Coudart - 08/2018 -- OSM - IGN - Données interne EPTB SN

0 10 20 30 40 km



12 Conclusion

La thématique des têtes de bassin versant est l'une des thématiques d'avenir dans le monde de la gestion des milieux aquatiques. Sur le bassin Loire Bretagne, les études se multiplient, par le biais de stages, mais aussi en interne et en lien avec des bureaux d'études privés.

Pour autant, aucune méthodologie claire n'est apportée par le comité de bassin lorsqu'il demande aux porteurs de SAGE de localiser, caractériser et hiérarchiser les têtes de bassin sur leur territoire.

Comme le montre la démarche conjointe du Forum des Marais et du département du Finistère, il existe une volonté de mutualiser les efforts et d'harmoniser la méthode. L'AFB, par l'entremise, entre autres, de Mickaël Le Bihan, favorise également le dialogue entre les structures et accompagne ces dernières pour qu'elles s'articulent les unes aux autres. Dans le chapitre du SAGE de la Sèvre nantaise qui lui est consacré, il est également précisé que la démarche doit se faire en concertation avec les structures voisines. En effet, une méthodologie solide et appliquée sur l'ensemble du bassin Loire Bretagne ne peut que renforcer le discours autour des enjeux des têtes de bassin versant.

Les avantages à tirer d'une harmonisation des études et méthodologies sont multiples. En effet, tout le monde est confronté aux mêmes problématiques qu'elles soient techniques ou organisationnelles.

- Constituer un référentiel cours d'eau topologiquement juste et exploitable par les systèmes d'informations géographiques est un écueil récurrent voire systématique et chacun recherche les solutions de son côté.
- Juger de la valeur d'un indicateur, de son coefficient de corrélation entre une donnée terrain et une donnée cartographique sont autant de questionnements communs.
- Choisir une méthode pour définir les têtes de bassin versant prioritaires est également un choix que chaque structure se doit d'effectuer.

Pour toutes ces raisons il est judicieux d'approfondir un domaine pour le partager avec la communauté. Ainsi, on évite les doublons et approximations. C'est dans cet esprit que nous nous sommes inspirés des travaux de nos voisins et prédécesseurs. C'est également dans cet esprit que nous avons travaillé à la conception de chaînes de traitements puis à leur automatisation. L'alimentation régulière du Wiki public poursuivait cet objectif de partage et a pu être testé dès cet été par des collègues.

C'est pour la même raison que le travail a été réalisé avec des outils open-source. Des outils libres, gratuits, disponibles partout et pour tous. Même si la gratuité ne comprend pas le temps de formation et de travail, cela reste une liberté non négligeable. L'utilisateur n'est pas captif d'une licence parfois onéreuse, une structure avec peu de moyens est en mesure avec la méthodologie proposée ici de mener une étude pointue pour un coût très faible.

La reconquête de la qualité de l'eau est une bataille de longue haleine, les enjeux économiques, écologiques et de santé publique sont des enjeux majeurs pour les générations à venir, espérons que cette pierre puisse participer à l'édifice et enrichir la connaissance.

13 Bibliographie

ALEXANDER, R.B. & AI, 2007. The role of headwater stream in downstream water quality. Journal of the American Water Resources Association.

AUGIER, 2017, Étude de la prise en compte du SDAGE Loire-Bretagne sur les têtes de bassin versant en Finistère, Forum des Marais Atlantiques en partenariat avec Conseil départemental du Finistère. Rapport de stage.

BENDA, L., & AI 2005. Geomorphology of steepland headwaters : the transition from hillslopes to channels. Journal of the American Water Resources Association.

BISHOP, K., & AI 2008. Aqua Incognita : the unknow headwaters. Hydrological Processes.

CEN , Les boisements en zones humides, cahier technique du bassin de la Loire, 2017

CHOUCARD P., 2011. Etude d'une méthodologie d'inventaire cartographique et de hiérarchisation des têtes de bassin versant dans le contexte armoricain, application au bassin versant du Couesnon. Rapport de stage. SAGE Couesnon.

CIROU, 2017, Élaboration d'une méthode de délimitation et de caractérisation des têtes de bassin versant de la Vilaine par approche cartographique, Application aux bassins versants de l'Aff et du Don, EPTB Vilaine. Rapport de stage.

COLIN M., 2015. Étude de l'hydromorphologie à l'échelle stationnelle des cours d'eau en tête de bassin versant, évaluation de l'impact des travaux de chenalisation, Rapport de stage Master 2, ONEMA.

CORPEN, 2008. Les zones tampons, un moyen de préserver les milieux aquatiques.

DARTY T., & AI, 2011. La zone hyporhéique, une composante à ne pas négliger dans l'état des lieux et la restauration des cours d'eau

GOMI T., & AI 2002. Understanding processes and downstream linkages of headwater systems

HENNER R., 2013. Les têtes de Bassin Versant, des espaces à considérer pour une gestion durable et intégrée de la ressource en eau. Rapport de stage. Université de Caen Basse-Normandie. Institution Interdépartementale du Bassin de la Sarthe.

JAN A., 2012. Proposition d'une méthodologie de hiérarchisation des cours d'eau de tête de bassin versant selon l'enjeu « morphologie ». Rapport de stage. SAGE Couesnon. Université de Rennes 1.

JOLY A., 2014. Identification cartographique et hiérarchisation des têtes de bassin versant sur les périmètres des SAGE Evre-Thau-Saint Denis et Layon-Aubance. Rapport de stage.

LAGGOUN-DEFARGE F., & AI, 2008. Les tourbières et leur rôle de stockage de carbone face aux changements climatiques.

- LE BIHAN M., 2017, Retours d'expériences techniques Chapitre 11, Têtes de bassin versant, Note technique. AFB.
- LE BIHAN M., 2017, Qu'est-ce qu'une tête de bassin versant ? Support de formation, AFB et ATBVB (Association des Techniciens Bassin Versant Breton).
- LHÉRITIER N., 2012. Les têtes de bassin : de la cartographie aux échelles mondiales et françaises à la caractérisation des ruisseaux limousins. Thèse dirigée. Université de Limoges.
- MALAVOI J.R. & BRAVARD J.P., 2010. Eléments d'hydromorphologie fluviale, ONEMA.
- MEYER J.L. & WALLACE J.B., 2001. Lost Linkages and Lotic Ecology: Rediscovering Small Streams.
- MOREAU, 2016, Analyse SIG de l'évolution des tracés des cours d'eau du Bassin Versant de la Sèvre Nantaise et automatisation de leur création, EPTB Sèvre Nantaise. Rapport de stage.
- NGUYEN VAN R., 2012. Les altérations physiques en têtes de bassin versant sur les régions Bretagne-Pays de la Loire. ONEMA.
- ORAISON, F., & Al, 2011. Restaurer l'hydromorphologie des cours d'eau et mieux maîtriser les nutriments : une voie commune ? Rapport, Office national des eaux et des milieux aquatiques et Centre national du machinisme agricole du génie rural, des eaux et des forêts, 42 p.
- PELLA H. & Al, 2011. Réseau hydrographique naturel étendu (RHE) construit à partir de la BD Carthage.
- RECLUS E., 1869 Histoire d'un ruisseau, Hetzel, rééd. 1995, Arles, Actes Sud.
- SDAGE bassin Loire Bretagne, 2016 – 2021.
- SAGE bassin Sèvre Nantaise, 2015.
- SALPIN, B., 2016, Elaboration d'une démarche concertée d'inventaire cartographique et de caractérisation des têtes de bassin versant du périmètre du SAGE estuaire de la Loire, SYLOA. Rapport de stage.
- SPITONI M., 2012. Caractérisation géospatiale des pressions qui s'exercent sur les cours d'eau de tête de bassin versant. Rapport de stage. ONEMA.
- STRAHLER, A.N., 1952. Hypsometric (are-altitude) analysis of erosional topography. Bulletin of the Geological Society of America.
- THERIN, E., VAROQUIER, S. & JACOB, F. 2015, Évaluation des atteintes aux têtes de bassin versant Caractérisation & hiérarchisation. Rapport du Syndicat Mixte Environnemental Goëlo Argoat.
- VANNOTE R.L., & Al, 1980. The River Continuum Concept. Canadian Journal of Fisheries and Aquatic Sciences.

13.1 Sites internet :

[Introduction à Grass \(Espace interministériel de l'information géographique\)](#)

[Grass tutorial : Watershed analysis](#)

[Analyse de bassins avec Grass – Qgis \(MEDDE\)](#)

[PGHydro – HYDROGRAPHIC OBJECTS IN SPATIAL DATABASE MANAGEMENT SYSTEM](#)

[Étude hydrologique du bassin versant \(concentration des pluies\)](#)

[Cartographie numérique, précis de discrétisation](#)

[Statistique et traitement des données](#)

[Aide mémoire R](#)

[Occupation de sols en Franche-Comté](#)

[Introduction à Postgis](#)

[Postgis.fr](#)

[Cours et tutoriels SQL](#)

[Forum zone humide](#)

Index des illustrations

Illustration 1: Définition graphique de la notion de bassin versant.....	7
Illustration 2: Territoire du bassin versant de la Sèvre Nantaise.....	8
Illustration 3: Pourcentage de cours d'eau en bon état écologique au regard des objectifs DCE. Freshwaterblog, 2018, source union européenne.....	9
Illustration 4: Schéma classification selon l'ordre de Strahler.....	10
Illustration 5: Exemple d'une fonction Grass passé en PHP.....	26
Illustration 6: Tableau de comparatif des différents référentiels hydrographiques disponibles.....	30
Illustration 7: Incohérence de tracé théorique.....	31
Illustration 8: Réseau hydro réel VS réseau théorique surcreusé.....	33
Illustration 9: Réseau hydro réel VS réseau théorique brut.....	33
Illustration 10: raccordement des tronçons orphelins.....	37
Illustration 11: Principes de fonctionnement de la fonction de correction des anastomoses.....	39
Illustration 12: micros bassins versants de la Sèvre Nantaise.....	42
Illustration 13: Grille d'indicateur à l'échelle des caractéristiques physiques de la TBV.....	47
Illustration 14: Grille d'indicateurs à l'échelle du lit mineur et de la BR.....	56
Illustration 15: Grille d'indicateurs à l'échelle TBV avec valeurs de pondération / enjeu.....	56
Illustration 16: Tableau détail des indices de pression et pondérations associées.....	57
Illustration 17: Carte de l'indice de pression à l'échelle des TBV pour le bassin de la Sèvre Nantaise.....	57
Illustration 18: Proposition de regroupement des codifications RPG en entités homogènes.....	59
Illustration 19: Codification de l'OCS Théia.....	60
Illustration 20: Carte de l'occupation du sol sur le territoire de la Sèvre nantaise, étendu à 2 km.....	62
Illustration 21: Table des géométries des TBV.....	65
Illustration 22: Table des paramètres de caractérisations.....	66
Illustration 23: Table des résultats.....	67
Illustration 24: Table, scénario de pondération.....	69
Illustration 25: Table, paramètres de pondérations.....	70
Illustration 26: Tableau des vulnérabilités.....	72

14 Annexes

```
/* Annexe_2 f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04.sql */
```

```
/*  
-- Function: xx_99_utils.f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04(geometry,  
name, name, name, name, name, numeric, text, text, name, name, name, name, text, name)  
  
-- DROP FUNCTION  
xx_99_utils.f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04(geometry, name, name,  
name, name, name, numeric, text, text, name, name, name, name, text, name);  
*/
```

```
CREATE OR REPLACE FUNCTION
```

```
xx_99_utils.f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04 (  
_geom_loc_root_seg geometry,  
_source_schemaname name,  
_source_tablename name,  
_source_geomname name,  
_dest_schemaname name,  
_dest_tablename name,  
_buffer_snap numeric DEFAULT 15,  
_colsnames_suffix text DEFAULT ''::text,  
_source_where_clause text DEFAULT NULL::text,  
_ecoulements_theo_schemaname name DEFAULT NULL::name,  
_ecoulements_theo_tablename name DEFAULT NULL::name,  
_ecoulements_theo_geomname name DEFAULT NULL::name,  
_ecoulements_theo_pkname name DEFAULT NULL::name,  
_ecoulements_theo_where_clause text DEFAULT NULL::text,  
_ecoulements_theo_ordername name DEFAULT NULL::name)
```

```
RETURNS text AS
```

```
$BODY$
```

```
DECLARE
```

```
/*
```

```
Chaine de traitement SQL qui corrige une couche composite qui a perdu sa cohérence  
topologique de réseau.
```

```
Est aussi capable de raccorder un tronçon isolé grace au donnée d'écoulement théorique  
(issus d'un MNT surcreusé par le tracé de la couche en entrée à corriger)
```

```
Crée deux nouvelle tables : une nouvelle table et ajoute les colonnes suivantes :
```

- 1) Table de résultats _dest_tablename
geom (linestring) = trançon hydro découpés à chaque intersection. La géométrie
est orientée
correction_result[**] : OK : cours d'eau dans réseau topologique // ISOLE :
Segment non rattaché
- 2) Table détaillant les opération réalisées :
geom geometry(LineString,2154),
geom_ori geometry(LineString,2154),
correction_context :
referentiel
Seg Isole
link_iso_2_theo
Seg Isole => link_with_theo_stream
link_to_pt_start_ori
link_to_linestring_ori
link_theo_part_or_not
ref_max_theo_order

```
Exemple d'appel :
```

```
set work_mem = '150000';
```

```
SELECT xx_99_utils.f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04 (  
_geom_loc_root_seg := ST_BUFFER(ST_SETSRID (ST_geomfromtext('POINT(355884  
6687390)'),2154),100)::geometry -- geometrie contenant le point terminal  
(aval) du segment root. Attention cette geom ne doit pas couvrir le point de départ  
du segment root  
, _source_schemaname :=  
'r020_territoire_hydrographie'::NAME  
-- nom du schema de la table contenant le réseau hydro à corriger  
, _source_tablename:='t_cours_ref_ddtm_et_ign'::  
NAME -- nom de la  
table contenant le réseau hydro avec graph corrigé  
, _source_geomname := 'geom' ::  
NAME  
-- nom de la colonne geom  
, _dest_schemaname := 'm060_milieux_continuite_tetes_de_bv' ::  
NAME -- Schéma de destination
```

```
/* Annexe_2 f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04.sql */
```

```
(création des nouvelles tables)
, _dest_tablename := 't_cours_ref_ddtm_et_ign_corr'::
NAME -- table de
destination (création des nouvelles tables)
, _buffer_snap
:=15::numeric --distance de raccordement des cours d'eau
, _colsnames_suffix :=
''::name
-- [optionnel] : suffixe qui sera ajouté aux noms des colonnes ajoutées
, _source_where_clause := ' st_intersects(geom , (SELECT geom FROM
r020_territoire_hydrographie.t_bvsn_actu_carthage) ) AND geom is not null ':: TEXT --
[optionnel] : Clause Where sur la donnée source
, _ecoulements_theo_schemaname
:= 'm060_milieus_continuite_tetes_de_bv'::name
-- [optionnel] : Schéma contenant la couche des réseau théoriques surcreusé par
la données en entrée
, _ecoulements_theo_tablename :=
't_mnt_sc_ddtm_sk_stream_500'::name
-- [optionnel] : table contenant la couche des réseau théoriques surcreusé par la
données en entrée
, _ecoulements_theo_geomname :=
'wkb_geometry'::name
-- [optionnel] : géométrie de la table contenant la couche des réseau théoriques
surcreusé par la données en entrée
, _ecoulements_theo_pkname := 'ogc_fid'
::name --
[optionnel] : pk de la table contenant la couche des réseau théoriques surcreusé par
la données en entrée
, _ecoulements_theo_where_clause := ' st_intersects(theo.wkb_geometry , (SELECT geom
FROM r020_territoire_hydrographie.t_bvsn_actu_carthage) ) ':: TEXT -- [optionnel] :
Clause Where sur la réseau théoriques
, _ecoulements_theo_ordername :=
'shreve'::name
-- [optionnel] : Colonne contenant la meilleure données de hierarchisation des cours
d'eau( amont / aval)
)
)
```

notes de version :

- V01_04 : = si création link nearest point (and not to start point) => ajoute un point au niveau de la confluence sur le segment exutoire (nécessite itération individuelle sur la recherche de seg isolés (buffer snap)) pour mettre à jour chaque géométrie si elle est impactée plusieurs fois
- V01_03 : = rattachement fin theo en utilisant link (snap_buffer) le theo est coupé à une distance légèrement < snap buffer
- V01_02 : = recherche reseau theo dans un buffer autour de l'union des seg_isoles
- V01 : first version

```
*/
mode_debug BOOLEAN := TRUE ; ---true; -- TRUE renvoi des notices
```

```
mode_test_part_function BOOLEAN :=FALSE; --TRUE -- FALSE
```

```
rcd_add_link record;
rcd_add_link_theo RECORD;
i int;
j int;
str_sql text;
str_sql_f_orientation TEXT ;
str_sql_add_link_isoles TEXT ;
str_sql_find_better_theo TEXT;
```

```
srid int:= srid from geometry_columns WHERE f_table_schema = _source_schemaname::text
AND f_table_name = _source_tablename::text AND f_geometry_column = _source_geomname::text;
_colsnames_suffix name := case WHEN _colsnames_suffix like '' THEN NULL ELSE
_colsnames_suffix END ;
```

```
col_geom_name NAME := 'geom' || coalesce ('_' || _colsnames_suffix , '') ;
```

/* Annexe_2 f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04.sql */

```

col_geom_ori_name NAME := 'geom_ori' || coalesce ('_' || _colsnames_suffix , '' ) ;
col_corr_context_name NAME := 'correction_context' || coalesce ('_' || _colsnames_suffix , '' ) ;
col_corr_result_name NAME:= 'correction_result' || coalesce ('_' || _colsnames_suffix , '' ) ;

col_corr_context_name_debug NAME := col_corr_context_name || '_debug';

source_where_clause text := CASE WHEN _source_where_clause IS NOT NULL THEN ' WHERE '
|| _source_where_clause ELSE ' ' END;
ecoulement_theo_where_clause_with_and TEXT := CASE WHEN _ecoulements_theo_where_clause
IS NOT NULL THEN ' AND ' || _ecoulements_theo_where_clause ELSE ' ' END;

dest_tablename_detail NAME:= _dest_tablename || '_detail';
BEGIN

SET client_min_messages = ERROR;
if mode_debug THEN
SET client_min_messages = NOTICE; -- pour le debug
END IF;

--
-- 1 . Copie des geom de la table source et ajout de colonnes:
-- . geom_ori (pour le stockage de la géométrie orientée)
-- . corr_context
--

str_sql := format (
  $SQL$
  DROP TABLE IF EXISTS %I.%I;
  CREATE TABLE %I.%I (
    gid SERIAL primary key
    , %I geometry(linestring,%s)
    , %I geometry(linestring,%s)
    , %I text
    , %I TEXT
    , ref_max_theo_order int DEFAULT 0
  );

  INSERT INTO %I.%I (
    %I
    , %I
  )
  SELECT ST_CollectionExtract ((ST_Dump(%I)).geom,2)
  , 'Referentiel'
  FROM
  %I.%I
  %s
  ;

  CREATE INDEX sx_%I%I ON %I.%I USING GIST (%I);
  CREATE INDEX sx_%I%I ON %I.%I USING GIST (%I);

  $SQL$
  , _dest_schemaname , dest_tablename_detail
  , _dest_schemaname , dest_tablename_detail
  , col_geom_name , srid
  , col_geom_ori_name , srid
  , col_corr_context_name
  , col_corr_context_name_debug

  , _dest_schemaname , dest_tablename_detail
  , col_geom_name
  , col_corr_context_name
  , _source_geomname
  , _source_schemaname , _source_tablename
  , source_where_clause

  , col_geom_name , dest_tablename_detail , _dest_schemaname , dest_tablename_detail ,

```

```
/* Annexe_2 f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04.sql */
```

```
col_geom_name
, col_geom_ori_name, dest_tablename_detail ,_dest_schemaname , dest_tablename_detail
, col_geom_ori_name
);

IF mode_test_part_function IS TRUE THEN
ELSE RAISE NOTICE 'str_sql: %', str_sql;
Execute str_sql;
END IF;

--
-- 2 . Orientation des geom_graph_ori :
--
--

-- 2.1 orientation du segment root

str_sql := format (
$SQL$
UPDATE %I.%I t_up SET %I = case WHEN st_intersects(st_startpoint(t_up.%I),
%L::geometry) THEN ST_reverse(t_up.%I) ELSE t_up.%I END
WHERE ST_intersects (t_up.%I , %L::geometry) ;

$SQL$
,_dest_schemaname , dest_tablename_detail , col_geom_ori_name , col_geom_name,
_geom_loc_root_seg , col_geom_name , col_geom_name
, col_geom_name , _geom_loc_root_seg
);

IF mode_test_part_function IS TRUE THEN
ELSE RAISE NOTICE 'str_sql: %', str_sql;
Execute str_sql;
END IF;

-- 2.2 orientation des autres segments

str_sql_f_orientation := format (
$SQL$
SELECT xx_99_utils.f_reseau_hydro_orient_seg_from_root_v01_01 (
_geom_loc_root_seg := %L
, _schemaname := %L
, _tablename:=%L
, _tablepkname := 'gid'
, _geomname :=
%L
, _geomoriname := %L
)
$SQL$
, _geom_loc_root_seg
, _dest_schemaname
, dest_tablename_detail
, col_geom_name
, col_geom_ori_name
);

IF mode_test_part_function IS TRUE THEN
ELSE RAISE NOTICE 'str_sql_f_orientation: %', str_sql_f_orientation;
Execute str_sql_f_orientation;
END IF;

-- 2.3 Commentaire sur les ségments correctes sur le plan topologique
str_sql := format (
$SQL$

update %I.%I t_up
SET %I = '00_Referentiel_topologie_OK'
```

```
/* Annexe_2 f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04.sql */
```

```
WHERE t_up.%I IS NOT NULL
```

```
$$$SQL$
```

```
, _dest_schemaname, dest_tablename_detail  
, col_corr_context_name  
, col_geom_ori_name  
);
```

```
IF mode_test_part_function IS TRUE THEN
```

```
ELSE
```

```
RAISE NOTICE 'SQL tag Référentiel topologie_OK: %', str_sql;
```

```
Execute str_sql;
```

```
END IF;
```

```
-- recherche des segments isolés et rattachement si distance <= buffer snap)
```

```
str_sql_add_link_isoles := format (
```

```
$$$SQL$
```

```
with w_seg_isoles as (  
SELECT t1.gid  
, t1.%I as geom  
, st_startpoint(t1.%I) as geom_start  
, st_endpoint(t1.%I) AS geom_end
```

```
from %I.%I t1  
WHERE t1.%I IS null
```

```
)
```

```
$$$SQL$
```

```
, col_geom_name  
, col_geom_name  
, col_geom_name  
, _dest_schemaname, dest_tablename_detail  
, col_geom_ori_name  
);
```

```
str_sql_add_link_isoles := str_sql_add_link_isoles ||format (
```

```
$$$SQL$
```

```
, new_link as (
```

```
select
```

```
case when t_ori_start_point.geom_l is not null  
THEN t_ori_start_point.geom_l  
ELSE t_ori.geom_l  
END ::geometry(linestring,%s) as geom
```

```
,
```

```
case when t_ori_start_point.geom_l is not null  
THEN t_ori_start_point.geom_l  
ELSE t_ori.geom_l  
END ::geometry(linestring,%s) as geom_oriented
```

```
,
```

```
case when t_ori_start_point.geom_l is not null  
THEN 'Link (buffer) : Seg Isole => Oriented Linestring (point start)'  
ELSE 'Link (buffer) : Seg Isole => Oriented Linestring (nearest point)'  
END ::text as commentaire
```

```
, case when t_ori_start_point.geom_l is not null  
THEN NULL  
ELSE gid_corr_to_up  
END ::int as gid_corr_to_up
```

```
FROM
```

```
w_seg_isoles seg_iso
```

```
$$$SQL$
```

```
, srid  
, srid  
);
```

```
str_sql_add_link_isoles := str_sql_add_link_isoles ||format (
```

```
$$$SQL$
```

```
LEFT JOIN LATERAL (  
with
```

```
/* Annexe_2 f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04.sql */
```

```
w_analyse_seg_start_with_ori_start as (  
  SELECT  
    seg_iso.geom_start as geom_pt_start  
    , st_startpoint( t_corr.%I) as geom_pt_end  
    , ST_distance(seg_iso.geom_start,st_startpoint(t_corr.%I)) dist  
  FROM %I.%I t_corr  
  WHERE ( st_intersects(st_buffer(seg_iso.geom_start,%s),t_corr.%I) )  
  AND  
  (st_intersects(st_buffer(seg_iso.geom_start,%s),st_startpoint(t_corr.%  
I) ) ) )  
  AND t_corr.gid <> seg_iso.gid  
  ORDER by ST_distance(seg_iso.geom_start,st_startpoint(t_corr.%I))  
  ASC LIMIT 1  
)  
  
, w_analyse_seg_end_with_ori_start as (  
  SELECT seg_iso.geom_end geom_pt_start  
  ,st_startpoint( t_corr.%I) geom_pt_end  
  , ST_distance(seg_iso.geom_end,st_startpoint(t_corr.%I)) dist  
  FROM %I.%I t_corr  
  WHERE ( st_intersects(st_buffer(seg_iso.geom_end,%s),t_corr.%I) )  
  AND  
  (st_intersects(st_buffer(seg_iso.geom_end,%s),st_startpoint(t_corr.  
%I) ) ) )  
  AND t_corr.gid <> seg_iso.gid  
  ORDER by ST_distance(seg_iso.geom_end,st_startpoint(t_corr.%I))  
  ASC LIMIT 1  
)  
  
, w_union as (  
  SELECT ST_makeline(geom_pt_start, geom_pt_end) as geom_l, dist FROM  
  w_analyse_seg_start_with_ori_start  
  Union ALL  
  SELECT ST_makeline(geom_pt_start, geom_pt_end) as geom_l, dist FROM  
  w_analyse_seg_end_with_ori_start  
)  
  SELECT geom_l geom_l from w_union WHERE dist IS NOT NULL order by dist asc  
  LIMIT 1  
)  
  as t_ori_start_point ON TRUE  
$SQL$  
  
, col_geom_ori_name  
, col_geom_ori_name  
, _dest_schemaname, dest_tablename_detail  
, _buffer_snap, col_geom_ori_name  
, _buffer_snap, col_geom_ori_name  
, col_geom_ori_name  
  
, col_geom_ori_name  
, col_geom_ori_name  
, _dest_schemaname, dest_tablename_detail  
, _buffer_snap, col_geom_ori_name  
, _buffer_snap, col_geom_ori_name  
, col_geom_ori_name  
);
```

```
str_sql_add_link_isoles := str_sql_add_link_isoles ||format (  
$SQL$
```

```
LEFT JOIN LATERAL (
```

```
with w_analyse_seg_start_with_ori as (
```

```
  SELECT
```

```
    t_corr.gid gid_corr_to_up
```

```
    , seg_iso.geom_start as geom_pt_start
```

```
    ,ST_ClosestPoint(t_corr.%I, seg_iso.geom_start) as geom_pt_end
```

```
    , ST_distance(seg_iso.geom_start,ST_ClosestPoint(t_corr.%I,
```

```
seg_iso.geom_start) ) dist
```

```
  FROM %I.%I t_corr
```

```
  WHERE (
```

```
    st_intersects(st_buffer(seg_iso.geom_start,%s),t_corr.%I) )
```

```
  AND t_corr.gid <> seg_iso.gid
```

```
/* Annexe_2 f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04.sql */
```

```
ORDER by  
ST_distance(seg_iso.geom_start,ST_ClosestPoint(t_corr.%I,  
seg_iso.geom_start) ) ASC LIMIT 1  
)
```

```
, w_analyse_seg_end_with_ori as (  
SELECT  
t_corr.gid gid_corr_to_up  
, seg_iso.geom_end geom_pt_start  
,ST_ClosestPoint( t_corr.%I, seg_iso.geom_end) geom_pt_end  
, ST_distance(seg_iso.geom_end,ST_ClosestPoint(t_corr.%I,  
seg_iso.geom_end)) dist  
FROM %I.%I t_corr  
WHERE ( st_intersects(st_buffer(seg_iso.geom_end,%s),t_corr.%I) )  
AND t_corr.gid <> seg_iso.gid  
ORDER by ST_distance(seg_iso.geom_end,ST_ClosestPoint(t_corr.%I,  
seg_iso.geom_end)) ASC LIMIT 1  
  
)  
  
, w_union as (  
SELECT gid_corr_to_up, geom_pt_end, ST_makeline(geom_pt_start,  
geom_pt_end) as geom_l, dist FROM w_analyse_seg_start_with_ori  
Union ALL  
SELECT gid_corr_to_up, geom_pt_end, ST_makeline(geom_pt_start,  
geom_pt_end) as geom_l, dist FROM w_analyse_seg_end_with_ori  
)  
SELECT gid_corr_to_up, geom_pt_end, geom_l geom_l from w_union WHERE  
geom_l IS NOT NULL AND dist IS NOT NULL order by dist asc LIMIT 1  
) t_ori ON TRUE
```

```
$$SQL$  
, col_geom_ori_name  
, col_geom_ori_name  
, _dest_schemaname, dest_tablename_detail  
, _buffer_snap, col_geom_ori_name  
, col_geom_ori_name  
  
, col_geom_ori_name  
, col_geom_ori_name  
, _dest_schemaname, dest_tablename_detail  
, _buffer_snap, col_geom_ori_name  
, col_geom_ori_name  
);
```

```
-- ajout de LIMIT 1 pour itérer sur chaque cas
```

```
str_sql_add_link_isoles := str_sql_add_link_isoles ||format (  
$$SQL$
```

```
WHERE (t_ori_start_point.geom_l is not null OR t_ori.geom_l is not null  
)  
LIMIT 1  
)
```

```
$$SQL$  
);
```

```
str_sql_add_link_isoles := str_sql_add_link_isoles ||format (  
$$SQL$
```

```
, w_add_point_on_oriented_seg_linked_at_nearest_point as (  
update %I.%I t_up  
set %I = (select  
xx_99_utils.ST_Addpoint_at_nearest_seg_on_linestring_V01_01 (  
_geomLinestring := t_up.%I  
, _geomPointToAdd :=  
st_endpoint(new_link.geom_oriented)  
)::geometry(linestring,2154) from new_link WHERE  
gid_corr_to_up = t_up.gid LIMIT 1  
)  
, %I = %I || ' ==> add new point'  
, %I = coalesce(%I, '') || ' ==> add new point ' || (select  
st_astext(st_endpoint(new_link.geom_oriented)) from new_link WHERE  
gid_corr_to_up = t_up.gid LIMIT 1)  
WHERE gid = (select gid_corr_to_up from new_link WHERE gid_corr_to_up is  
not null LIMIT 1)
```

```

)
$SQL$
, _dest_schemaname, dest_tablename_detail
, col_geom_ori_name
, col_geom_ori_name
, col_corr_context_name , col_corr_context_name
, col_corr_context_name_debug , col_corr_context_name_debug
);

```

str_sql_add_link_isoles := str_sql_add_link_isoles ||format (

```

$SQL$
insert into %I.%I
(%I, %I, %I)
SELECT geom, geom_oriented, commentaire from new_link
RETURNING gid as rtn;
$SQL$
, _dest_schemaname, dest_tablename_detail
, col_geom_name, col_geom_ori_name , col_corr_context_name
);

```

RAISE NOTICE 'str_sql_add_link_isoles: %', str_sql_add_link_isoles;

i=0;

IF mode_test_part_function IS TRUE THEN

ELSE

LOOP

Execute str_sql_add_link_isoles into rcd_add_link;

EXIT WHEN rcd_add_link.rtn is null;

Execute str_sql_f_orientation;

i=i+1;

RAISE NOTICE 'nombre de boucles str_sql_add_link_isoles (premier passage): %', i;

END LOOP;

END IF;

/*

-- 3 Rattachement des derniers segments isolés avec les données -- d'écoulement théoriques
surcreusés par la donnée d'origine (corrigé ou non)

*/

IF _ecoulements_theo_schemaname IS NOT NULL THEN

str_sql_find_better_theo := format (

```

$SQL$

WITH w_union_seg_isoles as (
SELECT (ST_Dump(ST_union(st_linemerge(%I)))) .geom as geom
FROM %I.%I t1
WHERE t1.%I IS NULL
)
, w_union_seg_isoles_with_max_order as (
SELECT tu.geom
, st_buffer(tu.geom,%s) geom_buff
, max(t2.ref_max_theo_order) as actual_max_order
FROM w_union_seg_isoles tu
LEFT JOIN %I.%I t2 ON (st_intersects(st_buffer(tu.geom,%s), t2.%I) AND t2.%I is null)
GROUP BY tu.geom
)
)
, w_union_seg_isoles_better_candidate as (
SELECT
union_seg_isoles.geom as seg_isole_geom
, union_seg_isoles.geom_buff as seg_isole_geom_buff
, theo.%I theo_geom
, theo.%I theo_order
FROM
w_union_seg_isoles_with_max_order union_seg_isoles
JOIN %I.%I theo ON (ST_intersects
(st_buffer(union_seg_isoles.geom_buff,%s/100.),theo.%I))
WHERE union_seg_isoles.actual_max_order < theo.%I
%s
ORDER BY theo.%I DESC,
ST_Distance(ST_intersection(union_seg_isoles.geom_buff,theo.%I),

```

```
/* Annexe_2 f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04.sql */
```

```
        ST_endpoint(theo.%I)) asc
LIMIT 1
)

, w_lst_seg_in_union_seg_to_update as (
select
    seg_iso.gid
    , seg_iso.%I geom_isol_uniq
    , CASE WHEN st_distance(st_startpoint(seg_iso.%I),ST_endpoint(c.theo_geom)) <
st_distance(st_endpoint(seg_iso.%I),ST_endpoint(c.theo_geom)) THEN
    st_startpoint(seg_iso.%I)
    ELSE
    st_endpoint(seg_iso.%I)
    END as estim_end_pt_geom_isol_uniq
    , CASE WHEN st_distance(st_startpoint(seg_iso.%I),ST_endpoint(c.theo_geom)) <
st_distance(st_endpoint(seg_iso.%I),ST_endpoint(c.theo_geom)) THEN
    st_distance(st_startpoint(seg_iso.%I),ST_endpoint(c.theo_geom))
    ELSE
    st_distance(st_endpoint(seg_iso.%I),ST_endpoint(c.theo_geom))
    END as estim_distance_end_pt_geom_isol_uniq
FROM w_union_seg_isoles_better_candidate c
LEFT JOIN %I.%I seg_iso on (ST_intersects(c.seg_isole_geom, seg_iso.%I) AND
seg_iso.%I IS NULL)

)

, w_creation_link_00 as (
SELECT
    c.theo_geom
    , c.theo_order
    , ST_endpoint(theo_geom) end_pt_theo
    , (SELECT estim_end_pt_geom_isol_uniq FROM w_lst_seg_in_union_seg_to_update ORDER
BY estim_distance_end_pt_geom_isol_uniq ASC LIMIT 1) estim_end_pt_geom_isol
    , ST_ClosestPoint(c.theo_geom,(SELECT estim_end_pt_geom_isol_uniq FROM
w_lst_seg_in_union_seg_to_update ORDER BY estim_distance_end_pt_geom_isol_uniq ASC
LIMIT 1)) as estim_start_pt_theo
from w_union_seg_isoles_better_candidate c
)

, w_creation_link_01 as (
SELECT ST_Makeline(l.estim_end_pt_geom_isol, l.estim_start_pt_theo) as geom_link
from w_creation_link_00 l
)

, w_creation_link_theo_cut_01_part_amont as (
SELECT

    l.theo_order
    , l.end_pt_theo
    , l.estim_end_pt_geom_isol
    , l.estim_start_pt_theo
    ,
    ST_collectionextract((ST_DUMP(st_split(l.theo_geom,ST_buffer(l.estim_start_pt_theo,%s
/1000.))))).geom,2) as geom_link
from w_creation_link_00 l
)

, w_creation_link_theo_02_cut_part_aval as (
select
    CASE WHEN st_union(seg_ori.%I) is null then
    pl.geom_link
    ELSE
    ST_collectionextract((ST_DUMP(st_difference(pl.geom_link,st_buffer(st_union(seg_o
ri.%I),%s*0.8))))).geom,2)
    END as geom_link
    , pl.estim_start_pt_theo
    ,pl.theo_order
FROM w_creation_link_theo_cut_01_part_amont pl
LEFT JOIN %I.%I seg_ori on (ST_intersects(pl.geom_link, seg_ori.%I ))
```

```
/* Annexe_2 f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04.sql */
```

```
WHERE ST_intersects (p1.geom_link, st_buffer(p1.end_pt_theo,%s/100. ) )
group by p1.geom_link ,p1.estim_start_pt_theo, p1.theo_order

)

, w_links as (
SELECT
geom_link
,0 as theo_order
, 'NEW Link : Isole => Theoric Stream' as context
FROM w_creation_link_01 WHERE ST_length(geom_link)>0.
union all
select
ST_RemoveRepeatedPoints(st_addpoint(geom_link,estim_start_pt_theo, 0))
, theo_order
, 'NEW Seg : Theoric Stream (part or not)' as context
FROM w_creation_link_theo_02_cut_part_aval
WHERE ST_intersects (geom_link, st_buffer(estim_start_pt_theo,%s/100. ) )
AND ST_length(geom_link)>0.

)
, w_up as (
update %I.%I t_up
SET ref_max_theo_order = (select p.theo_order FROM
w_creation_link_theo_02_cut_part_aval p limit 1)
, %I = %I || ' ==> ' || 'Seg Isole => link with theoric stream'
WHERE t_up.gid in (select lst_seg.gid FROM w_lst_seg_in_union_seg_to_update lst_seg)
)
insert into %I.%I
(%I,%I ,ref_max_theo_order)
SELECT geom link, context, theo_order FROM w_links
RETURNING gid as rtn;
;

$SQL$
, col_geom_name
, _dest_schemaname, dest_tablename_detail
, col_geom_ori_name

, _buffer_snap
, _dest_schemaname, dest_tablename_detail , _buffer_snap, col_geom_name ,
col_geom_ori_name

, _ecoulements_theo_geomname
, _ecoulements_theo_ordername
, _ecoulements_theo_schemaname, _ecoulements_theo_tablename, _buffer_snap,
_ecoulements_theo_geomname
, _ecoulements_theo_ordername
, ecoulement_theo_where_clause_with_and
, _ecoulements_theo_ordername, _ecoulements_theo_geomname, _ecoulements_theo_geomname

, col_geom_name
, col_geom_name, col_geom_name
, col_geom_name
, col_geom_name
, col_geom_name, col_geom_name
, col_geom_name
, col_geom_name
, _dest_schemaname, dest_tablename_detail, col_geom_name, col_geom_ori_name

, _buffer_snap

, col_geom_ori_name
, col_geom_ori_name , _buffer_snap
```

```
/* Annexe_2 f_reseau_hydro_correction_topologie_et_seg_isoles_v01_04.sql */
```

```
, _dest_schemaname, dest_tablename_detail, col_geom_ori_name
, _buffer_snap

, _buffer_snap

, _dest_schemaname, dest_tablename_detail
, col_corr_context_name, col_corr_context_name

, _dest_schemaname, dest_tablename_detail
, col_geom_name , col_corr_context_name

);
RAISE NOTICE 'str_sql_find_better_theo: %', str_sql_find_better_theo;

i=0;
j=0;

LOOP

Execute str_sql_find_better_theo into rcd_add_link_theo;
RAISE NOTICE 'rcd_add_link_theo.rtn: %', rcd_add_link_theo.rtn;
EXIT WHEN rcd_add_link_theo.rtn IS NULL;
RAISE NOTICE 'str_sql_find_better_theo: %', str_sql_find_better_theo;

j=0;

LOOP
RAISE NOTICE 'nombre de boucles add link: %', j;
Execute str_sql_add_link_isoles into rcd_add_link;
EXIT WHEN rcd_add_link.rtn is null;
Execute str_sql_f_orientation;
j=j+1;
--EXIT WHEN j >=2;

END LOOP;

i=i+1;
RAISE NOTICE 'nombre de boucles find_better theo: %', i;
--EXIT WHEN i =10;
END LOOP;

END IF;

-- tag segment isole

str_sql := format (
$SQL$

update %I.%I t_up
SET %I = 'Seg Isole'
WHERE t_up.%I IS NULL

$SQL$
, _dest_schemaname, dest_tablename_detail
, col_corr_context_name
, col_geom_ori_name
);
RAISE NOTICE 'SQL tag ISOLE: %', str_sql;
Execute str_sql;
```

```

--
-- CReation de la table finale line Merge
--

str_sql := format (
    $SQL$
    DROP TABLE IF EXISTS %I.%I;
    CREATE TABLE %I.%I AS
    WITH dump_merge AS (
        SELECT 'OK' as %I, (ST_Dump(St_Linemerge(ST_Union(%I)))) .geom
        ::geometry(linestring,%s) as geom
        FROM %I.%I WHERE %I IS NOT NULL
        UNION ALL
        SELECT 'ISOLE' as %I, (ST_Dump(St_Linemerge(st_union(%I)))) .geom
        ::geometry(linestring,%s) as geom
        FROM %I.%I WHERE %I IS NULL
    )
    SELECT row_number() over () ::bigint as gid
    , * from dump_merge
    ;

    CREATE INDEX sx_%I%i ON %I.%I USING GIST (%I);

    $SQL$
    ,_dest_schemaname , _dest_tablename
    ,_dest_schemaname , _dest_tablename
    , col_corr_result_name , col_geom_ori_name , srid
    ,_dest_schemaname , dest_tablename_detail, col_geom_ori_name
    , col_corr_result_name , col_geom_name , srid
    ,_dest_schemaname , dest_tablename_detail, col_geom_ori_name

    , col_geom_name, _dest_tablename ,_dest_schemaname , _dest_tablename , col_geom_name

    )
;
    RAISE NOTICE 'SQL CREATE DEST TABLE : %', str_sql;
Execute str_sql;

RETURN 'Traitement terminé . Voir résultat de correction dans la table :' ||
_dest_schemaname || '.' || _dest_tablename || ' Il faut filter avec le champ ' ||
col_corr_context_name || ' = OK.Le detail des operation est dans la table :' ||
_dest_schemaname || '.' || dest_tablename_detail ;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;

```

```
/* Annexe_3 f_reseau_correction_anastomoses_v04_01.sql */
```

```
/*
-- Function: xx_99_utils.f_reseau_correction_anastomoses_v04_01(name, name, name, name,
name, name, name, numeric, numeric)

-- DROP FUNCTION xx_99_utils.f_reseau_correction_anastomoses_v04_01(name, name, name, name,
name, name, name, numeric, numeric);
*/
CREATE OR REPLACE FUNCTION xx_99_utils.f_reseau_correction_anastomoses_v04_01 (
    _source_schemaname name,
    _source_tablename name,
    _source_geomname name,
    _source_pkname name,
    _dest_schemaname name,
    _dest_tablename_suffix name DEFAULT NULL::name,
    _dest_tpkname name DEFAULT 'gid_graph'::name,
    _snaptogrid_dist numeric DEFAULT 0.01,
    _buffer_factor numeric DEFAULT 0.1)
    RETURNS text AS
$BODY$
DECLARE
/*
Chaine de traitement SQL qui créé un réseau cohérent pour le calcul strahler en calculant
les liens au niveau des anastomoses
Va créer 2 tables :
    . xx_graph (copie de la table source avec les colonnes anastomose_context et geom_graph )
    . xx_anast (zones d'anastomose)

Exemple d'appel :

SELECT xx_99_utils.f_reseau_correction_anastomoses_v04_01 (
    _source_schemaname := 'r020_territoire_hydrographie'           -- nom du schema de la
table du réseau hydro à corriger
    , _source_tablename:='t_troncon_cours_eau_ign_bdtopo_151_bvsn' -- nom de la table du
réseau hydro à corriger
    , _source_geomname := 'geom'                                 -- nom de la colonne
géométrique du réseau hydro à corriger
    , _source_pkname := 'gid'                                   -- nom de la colonne
primary key du réseau hydro à corriger
    , _dest_schemaname := '_agent_ariviere'                     -- nom du schema de
destination
    , _dest_tablename_suffix := ''                               -- [optionnel] : suffixe
qui sera ajouté au nom des tables de résultat
    , _dest_tpkname := 'gid_graph'                               -- [optionnel] : nom de la
colonne pk de la table_graph de résultat
    , _snaptogrid_dist := 0.1                                   -- [optionnel] : maille de
la grille (en m) pour le snaptogrid (correction des micro problèmes de topologie)
    , _buffer_factor := 0.1                                     -- [optionnel] : valeur
(en m) permettant de rechercher les segments qui touchent les anastomoses
)

notes de version :
- V04 : Nouvelle logique deection des ce liées (= touchent ce bordure anastomose)
- V03 : fonction qui fait tout

*/
_dest_tablename_suffix name := case WHEN _dest_tablename_suffix like '' THEN NULL ELSE
_dest_tablename_suffix END ;

srid int:= srid from geometry_columns WHERE f_table_schema = $1::text AND f_table_name =
$2::text AND f_geometry_column = $3::text;
str_sql text;
mode_debug BOOLEAN := TRUE ; ---true; -- TRUE renvoi des notices

prefix_obj name := left ( _source_tablename || coalesce ('_' || _dest_tablename_suffix , '' )
,45); -- limite la longueur du nom de objets table ,pk , seq, idx
dest_tablename_graph NAME := prefix_obj || '_graph';
dest_tablename_anastomose NAME := prefix_obj || '_anast';

buffer_reseau_polygon numeric := _buffer_factor ;
buffer_reseau_polygon_makevalid numeric := _buffer_factor/10. ;
```

```
/* Annexe_3 f_reseau_correction_anastomoses_v04_01.sql */
```

```
buffer_reseau_polygon_isole numeric := _buffer_factor*3. ;  
buffer_anast_find_connected_for_add_link_to_point_on_surface numeric := _buffer_factor*10.;
```

```
max_gid_graph bigint;
```

```
BEGIN
```

```
SET client_min_messages = ERROR;  
if mode_debug THEN  
SET client_min_messages = NOTICE; -- pour le debug  
END IF;
```

```
/*  
-- 1 . copie et correction de la couche d'entrée avec ajout:  
-- . column anastomose_context  
-- . column geom_graph = st_snaptogrid  
*/
```

```
str_sql := format (  
$SQL$  
DROP table if exists %I.%I ;  
CREATE TABLE %I.%I AS  
WITH w_prepare AS (  
SELECT  
t.*  
, NULL::TEXT as anastomose_context  
, st_snaptogrid(ST_force2d(t.%I),%s)::geometry(linestring,%s) as geom_graph  
FROM %I.%I t  
)  
SELECT row_number() over() as %I, * from w_prepare;  
  
CREATE INDEX sx_%I_geom_graph  
ON %I.%I  
USING gist  
(geom_graph);  
CREATE INDEX sx_%I_%I  
ON %I.%I  
USING gist  
(%I);  
ALTER TABLE %I.%I ADD CONSTRAINT pkey%s PRIMARY KEY(%I);  
$SQL$  
,_dest_schename , dest_tablename_graph  
  
,_dest_schename , dest_tablename_graph  
  
,_source_geomname , _snaptogrid_dist , srid  
  
,_source_schename , _source_tablename  
  
,_dest_tpkname  
  
, dest_tablename_graph  
,_dest_schename , dest_tablename_graph  
  
, dest_tablename_graph , _source_geomname  
,_dest_schename, dest_tablename_graph  
,_source_geomname  
  
,_dest_schename , dest_tablename_graph, prefix_obj,_dest_tpkname  
);
```

```
RAISE NOTICE 'str_sql: %', str_sql;
```

```
Execute str_sql;
```

```
/*  
-- 2 . Création de la table des anastomoses
```

```
/* Annexe_3 f_reseau_correction_anastomoses_v04_01.sql */
```

```
-- STEP 01 : (buffer sur ce)
-- STEP 02 : (union des buffers)
-- STEP 03 : (exterior ring)
-- STEP 04 : (buffer négatif pour isoler les anastomoses)
-- STEP 05 : (dump + buffer de recherche : geom_buff + point on surface)
*/

/*
, 'join=mitre mitre_limit=20.0'
*/
str_sql := format (
    $SQL$
    DROP table if exists %I.%I;
    CREATE TABLE %I.%I as
    with
        step_01_buffer as (
            select
                st_buffer (geom_graph,%s) ::geometry(polygon,%s, 'endcap=square') as geom
            from %I.%I
        )
        , step_02_dump_union as (
            SELECT
                (st_dump(st_union(geom))).geom::geometry(polygon,%s) as geom
            from step_01_buffer
        )
        , step_03_exterior_ring as (
            select
                st_buffer(st_makepolygon(st_exteriorring(geom)),%s)::geometry(polygon,%s) as
                geom
            from step_02_dump_union
        )
        , step_04_buffer_neg as (
            select

                st_collectionextract(st_makevalid(st_multi(st_buffer(geom,-%s,'endcap=square'))
                ),3)::geometry(multipolygon,%s) as geom
            from step_03_exterior_ring
        )
        , step_05_01_dump as (
            select

                st_collectionextract(st_makevalid((st_dump(geom)).geom),3)::geometry(polygon,%s
                ) as geom
            from step_04_buffer_neg
        )
        , step_05_02_add_geom_buff as (
            SELECT
                row_number() over()::bigint as gid
                , geom
                , st_collectionextract(st_multi(st_buffer(geom,%s , 'join=mitre
                mitre_limit=20.0') ),3)::geometry(multipolygon,%s) as geom_buff
            FROM step_05_01_dump
        )

    SELECT
        *
        , st_pointonsurface(geom_buff)::geometry (point,%s) as geom_pt_on_surface
    FROM step_05_02_add_geom_buff;
    $SQL$
, dest_schename , dest_tablename_anastomose
, dest_schename , dest_tablename_anastomose
, buffer_reseau_polygon , srid
, dest_schename , dest_tablename_graph

, srid

, buffer_reseau_polygon_makevalid , srid

, buffer_reseau_polygon_isole , srid

, srid
```

```
/* Annexe_3 f_reseau_correction_anastomoses_v04_01.sql */
```

```
, buffer_anast_find_connected_for_add_link_to_point_on_surface , srid
, srid
);
-- index
str_sql := str_sql || format (
$SQL$
CREATE INDEX geom_%s
ON %I.%I
USING gist
(geom);

CREATE INDEX geom_buff_%s
ON %I.%I
USING gist
(geom_buff);
$SQL$

,dest_tablename_anastomose,_dest_schename , dest_tablename_anastomose
,dest_tablename_anastomose,_dest_schename , dest_tablename_anastomose
)
;

RAISE NOTICE 'str_sql: %', str_sql;

Execute str_sql;

/*
-- update de la table de graph
--
--
-- STEP 01 Détection des cours d'eau inclus dans les anastomoses :
*/
str_sql := format (
$SQL$
With w_include as (
select t_up.%I as gid
from %I.%I t_up
LEFT join %I.%I anast on (st_intersects(anast.geom, t_up.geom_graph))
--, LEFT join exists _agent_acoudart.t_tr_ce_ign_corr_anastomose_05_loc_anast_02
a_border_or_touches on (st_intersects(st_buffer(a_in.geom,0.2), t_up.geom))
WHERE t_up.anastomose_context is null AND (st_length(ST_intersection(anast.geom,
t_up.geom_graph)) *1.1) >= st_length(t_up.geom_graph)
)

update %I.%I t_up
set anastomose_context = 'include'
WHERE t_up.%I in (select gid from w_include)
;
$SQL$
,_dest_tpkname
,_dest_schename , dest_tablename_graph
,_dest_schename , dest_tablename_anastomose
,_dest_schename , dest_tablename_graph
,_dest_tpkname
);

/*
-- STEP 02 : Détection des cours d'eau en bordure d'anastomoses :
*/
str_sql := str_sql || format (
$SQL$
With w_border as (
select t_up.%I as gid
from %I.%I anast
LEFT join %I.%I t_up on (st_intersects(anast.geom_buff ,t_up.geom_graph))
WHERE t_up.anastomose_context is null
AND (st_length(ST_intersection(anast.geom_buff, t_up.geom_graph))*1.1) >=
```

```
/* Annexe_3 f_reseau_correction_anastomoses_v04_01.sql */
```

```
st_length(t_up.geom_graph)
```

```
)
```

```
update %I.%I t_up  
set anastomose_context = 'border'  
WHERE t_up.%I in (select gid from w_border)
```

```
;
```

```
$$SQL$  
,_dest_tpkname  
,_dest_schemaname , dest_tablename_anastomose  
,_dest_schemaname , dest_tablename_graph  
,_dest_schemaname , dest_tablename_graph  
,_dest_tpkname  
);
```

```
/*
```

```
-- STEP 03 :Détection des cours d'eau qui arrivent ou qui partent des anastomoses :  
-- TODO AR : recréer une geom d'anastomose grace aux geom de 'border' ce qui permet  
d'éviter de ratter des connexions (à cause du seuil de geom_buffer : cas des anastomoses  
trés fines)
```

```
*/
```

```
str_sql := str_sql || format (
```

```
$$SQL$
```

```
With
```

```
w_connected as (
```

```
select distinct t_up.%I as gid  
from %I.%I anast  
LEFT join %I.%I t_up on ( st_intersects(anast.geom_buff, t_up.geom_graph) )  
WHERE t_up.anastomose_context is null
```

```
union
```

```
SELECT DISTINCT t_up.%I as gid  
from %I.%I t_up  
join %I.%I ce on ( st_intersects(ce.geom_graph, t_up.geom_graph) and  
ce.anastomose_context in ( 'include','border' ) )  
WHERE t_up.anastomose_context is null
```

```
)
```

```
update %I.%I t_up  
set anastomose_context = 'connected'  
WHERE t_up.%I in (select distinct gid from w_connected)
```

```
;
```

```
$$SQL$  
,_dest_tpkname  
,_dest_schemaname , dest_tablename_anastomose  
,_dest_schemaname , dest_tablename_graph
```

```
,_dest_tpkname  
,_dest_schemaname , dest_tablename_graph  
,_dest_schemaname , dest_tablename_graph
```

```
,_dest_schemaname , dest_tablename_graph  
,_dest_tpkname  
);
```

```
/*
```

```
-- update Start : mise à jour de la géométrie des liens 'connected' : ajout du  
pointonsurface de l'anastomose correspondante pour in lien dont la geom part de l'anastomose
```

```
--
```

```
*/
```

```
str_sql := str_sql || format (
```

```
$$SQL$
```

```
With w_connected as (
```

```
select  
anast.gid  
,anast.geom  
, t_up.%I as gid_to_up
```

```
/* Annexe_3 f_reseau_correction_anastomoses_v04_01.sql */
```

```
    , st_reverse(st_addpoint(st_reverse(t_up.geom_graph), anast.geom_pt_on_surface))::
    geometry(linestring,%s) as geom_graph_up
from %I.%I anast
LEFT join %I.%I t_up on ( st_intersects(st_buffer(anast.geom_buff, %s),
t_up.geom_graph) AND t_up.anastomose_context='connected')
WHERE st_intersects(st_startpoint(t_up.geom_graph), st_buffer(anast.geom_buff,%s))
order by gid_to_up
)
update %I.%I t_up
set geom_graph = (select w.geom_graph_up from w_connected w where w.gid_to_up =
t_up.%I order by ST_distance(geom_graph,w.geom) ASC limit 1)
WHERE t_up.%I in (select distinct gid_to_up from w_connected )
;
$SQL$
, _dest_tpckname
, srid
, _dest_schemaname , dest_tablename_anastomose
, _dest_schemaname , dest_tablename_graph,
buffer_anast_find_connected_for_add_link_to_point_on_surface
, buffer_anast_find_connected_for_add_link_to_point_on_surface
, _dest_schemaname , dest_tablename_graph
, _dest_tpckname
, _dest_tpckname
);
```

```
/*
```

```
-- update End : mise à jour de la géométrie des liens 'connected' : ajout du pointonsurface
de l'anastomose correspondante pour in lien dont la géom arrive sur l'anastomose
*/
```

```
str_sql := str_sql || format (
$SQL$
    With w_connected as (
    select
    anast.gid
    , anast.geom
    , t_up.%I as gid_to_up
    , st_addpoint(t_up.geom_graph, anast.geom_pt_on_surface):: geometry(linestring,%s)
    as geom_graph_up
from %I.%I anast
LEFT join %I.%I t_up on ( st_intersects(st_buffer(anast.geom_buff, %s),
t_up.geom_graph) AND t_up.anastomose_context='connected')
WHERE st_intersects(st_endpoint(t_up.geom_graph), st_buffer(anast.geom_buff,%s))
order by gid_to_up
)
update %I.%I t_up
set geom_graph = (select w.geom_graph_up from w_connected w where w.gid_to_up =
t_up.%I order by ST_distance(geom_graph,w.geom) ASC limit 1)
WHERE t_up.%I in (select distinct gid_to_up from w_connected )
;
$SQL$
, _dest_tpckname
, srid
, _dest_schemaname , dest_tablename_anastomose
, _dest_schemaname , dest_tablename_graph,
buffer_anast_find_connected_for_add_link_to_point_on_surface
, buffer_anast_find_connected_for_add_link_to_point_on_surface
, _dest_schemaname , dest_tablename_graph
, _dest_tpckname
, _dest_tpckname
);
```

```
RAISE NOTICE 'str_sql: %', str_sql;
```

```
Execute str_sql;
```

```
--
-- Ajout de liens entre anastomoses qui se touchent
--
```

```
execute format ( $SQL$
```

```
/* Annexe_3 f_reseau_correction_anastomoses_v04_01.sql */
```

```
SELECT max(%I) from %I.%I ;
$SQL$
, _dest_tpkname , _dest_schemaname , dest_tablename_graph
) into max_gid_graph;
```

```
str_sql := format (
$SQL$
with
w_anastomoses_links as (
select
'anastomoses_links'::text as anastomose_context
, st_setsrid(st_makeline(anast1.geom_pt_on_surface, anast2.geom_pt_on_surface),
%s) :: geometry (linestring, %s) as geom_graph
FROM %I.%I anast1
JOIN %I.%I anast2 ON (ST_Intersects(anast1.geom_buff, anast2.geom_buff) AND
anast1.gid > anast2.gid)
)
, w_anastomoses_links_02 as (
SELECT row_number() over() as count
, *
FROM w_anastomoses_links
)

insert into %I.%I (gid_graph,anastomose_context, geom_graph) select
count+%,anastomose_context, geom_graph from w_anastomoses_links_02
;
$SQL$

,srid, srid
, _dest_schemaname , dest_tablename_anastomose
, _dest_schemaname , dest_tablename_anastomose
, _dest_schemaname , dest_tablename_graph , max_gid_graph
);
```

```
RAISE NOTICE 'str_sql: %', str_sql;
```

```
Execute str_sql;
```

```
RETURN 'Traitement ok. Utiliser la couche : ' || _dest_schemaname || '.' ||
dest_tablename_graph || ' (geom_graph) pour le calcul du Strahler.' ;
```

```
END;
```

```
$BODY$
```

```
LANGUAGE plpgsql VOLATILE
```

```
COST 100;
```

```
/* Annexe_4 f_reseau_hydro_strahler_rank_v01_04.sql */
```

```
/*
-- Function: xx_99_utils.f_reseau_hydro_strahler_rank_v01_04(geometry, name, name, name,
name, text, text)

-- DROP FUNCTION xx_99_utils.f_reseau_hydro_strahler_rank_v01_04(geometry, name, name, name,
name, text, text);
*/
CREATE OR REPLACE FUNCTION xx_99_utils.f_reseau_hydro_strahler_rank_v01_04 (
    _geom_loc_root_seg geometry,
    _source_schemaname name,
    _source_tablename name,
    _source_geomname name,
    _source_pkname name,
    _colnames_suffix text DEFAULT ''::text,
    _where_clause text DEFAULT NULL::text)
RETURNS text AS
$BODY$
DECLARE
/*
Chaine de traitement SQL qui oriente les geom d'un réseau cohérent (sans anastomoses -
cf fonction f_reseau_correction_anastomoses_vXX) et calcule le strahler rank.
Ajoute les colonnes suivantes :
. geom_graph_ori (géométrie orientée)
. strahler_rank

Exemple d'appel :

SELECT xx_99_utils.f_reseau_hydro_strahler_rank_v01_03 (
    _geom_loc_root_seg := ST_BUFFER(ST_SETSRID (ST_geomfromtext ('POINT(355884
6687390)'),2154),100)::geometry -- géométrie contenant le point terminal (aval) du
segment root. Attention cette geom ne doit pas couvrir le point de départ du segment
root
, _source_schemaname :=
'agent_ariviere' --
nom du schema de la table contenant le réseau hydro avec graph corrigé
,
_source_tablename:='t_troncon_cours_eau_ign_bdtopo_151_bvsn_graph'
-- nom de la table contenant le réseau hydro avec graph corrigé
, _source_geomname :=
'geom_graph' --
nom de la colonne geom de graph corrigé
, _source_pkname :=
'gid_graph'
-- nom de la colonne primary key
, _colnames_suffix :=
'' --
[optionnel] : suffixe qui sera ajouté aux noms des colonnes ajoutées
, _where_clause :=
NULL
-- [optionnel] : Clause Where (TODO AR : non implémenté)

)

notes de version :
- V01_04 : index GIST nom unique (pour start et end)
- V01_03 : REmontée du graph ajout LIMIT 1 pour eviter erreurs sur réseau erroné (TODO AR
: Voir a trouver une autre solution)
- V01_02 : Ajout select Distinct en -- 2.2 remontée du réseau et orientation des geom
(update geom_gaph_ori)
- V01 : first version

*/
mode_debug BOOLEAN := TRUE ; ---true; -- TRUE renvoi des notices

return_up_ids record;
i int;
str_sql text;
srid int:= srid from geometry_columns WHERE f_table_schema = _source_schemaname::text
AND f_table_name = _source_tablename::text AND f_geometry_column = _source_geomname::text;
_colnames_suffix name := case WHEN _colnames_suffix like '' THEN NULL ELSE
_colnames_suffix END ;
```

```
/* Annexe_4 f_reseau_hydro_strahler_rank_v01_04.sql */
```

```
col_geom_graph_ori_name NAME := 'geom_graph_ori' || coalesce ('_' || _colsnames_suffix , '' ) ;  
col_strahler_rank_name NAME := 'strahler_rank' || coalesce ('_' || _colsnames_suffix , '' ) ;
```

```
BEGIN
```

```
SET client_min_messages = ERROR;  
if mode_debug THEN  
    SET client_min_messages = NOTICE; -- pour le debug  
END IF;
```

```
/*  
-- 1 . Ajout des colonnes:  
-- . geom_graph_ori (pour le stockage de la géométrie orientée)  
-- . strahler_rank  
*/
```

```
str_sql := format (  
    $SQL$  
    ALTER table %I.%I DROP Column IF EXISTS %I ;  
    ALTER table %I.%I ADD Column %I geometry(linestring,%s) ;  
    ALTER table %I.%I DROP Column IF EXISTS %I ;  
    ALTER table %I.%I ADD Column %I INT ;  
  
    CREATE INDEX sx_%I%I ON %I.%I USING GIST (%I)  
    $SQL$  
    , _source_schemaname , _source_tablename , col_geom_graph_ori_name  
    , _source_schemaname , _source_tablename , col_geom_graph_ori_name , srid  
    , _source_schemaname , _source_tablename , col_strahler_rank_name  
    , _source_schemaname , _source_tablename , col_strahler_rank_name  
    , _source_tablename , col_geom_graph_ori_name , _source_schemaname , _source_tablename ,  
    col_geom_graph_ori_name  
    );
```

```
RAISE NOTICE 'str_sql: %', str_sql;
```

```
Execute str_sql;
```

```
--  
-- 2 . Orientation des geom_graph_ori :  
--  
--
```

```
-- 2.1 orientation du segment root
```

```
str_sql := format (  
    $SQL$  
    UPDATE %I.%I t_up SET %I = case WHEN st_intersects(st_startpoint(t_up.%I),  
    %L::geometry) THEN ST_reverse(t_up.%I) ELSE t_up.%I END  
    WHERE ST_intersects (t_up.%I , %L::geometry) ;  
  
    $SQL$  
    , _source_schemaname , _source_tablename , col_geom_graph_ori_name , _source_geomname ,  
    _geom_loc_root_seg , _source_geomname , _source_geomname  
    , _source_geomname , _geom_loc_root_seg  
    );
```

```
RAISE NOTICE 'str_sql: %', str_sql;
```

```
Execute str_sql;
```

```
/*  
-- 2.2 remontée du réseau et orientation des geom (update geom_gaph_ori)  
*/
```

```
i=0;  
LOOP  
    str_sql := format (  
        $SQL$  
        UPDATE %I.%I t_up SET %I = case WHEN st_intersects(st_startpoint(t_up.%I),  
        %L::geometry) THEN ST_reverse(t_up.%I) ELSE t_up.%I END  
        WHERE ST_intersects (t_up.%I , %L::geometry) ;  
  
        $SQL$  
        , _source_schemaname , _source_tablename , col_geom_graph_ori_name , _source_geomname ,  
        _geom_loc_root_seg , _source_geomname , _source_geomname  
        , _source_geomname , _geom_loc_root_seg  
        );
```

```
/* Annexe_4 f_reseau_hydro_strahler_rank_v01_04.sql */
```

```
$SQL$  
WITH w_up as (  
SELECT DISTINCT  
  t_not_ori.%I  
  , CASE WHEN ST_intersects (t_ori.%I, st_endpoint(t_not_ori.%I)) THEN  
  t_not_ori.%I ELSE ST_reverse(t_not_ori.%I) END as geom_graph_ori_to_up  
FROM %I.%I t_not_ori  
LEFT JOIN %I.%I t_ori ON (t_ori.%I is not null )  
  
WHERE ST_intersects (t_ori.%I, t_not_ori.%I) AND ST_intersects  
(st_startpoint(t_ori.%I) , t_not_ori.%I) AND (t_not_ori.%I is NULL ) AND  
(t_not_ori.%I != t_ori.%I )  
  
)
```

```
UPDATE %I.%I t_up SET %I = (SELECT w_up.geom_graph_ori_to_up FROM w_up WHERE  
w_up.%I = t_up.%I LIMIT 1)  
WHERE t_up.%I in (SELECT w_up.%I FROM w_up) RETURNING t_up.%I as gids;  
;
```

```
$SQL$  
,_source_pkname , col_geom_graph_ori_name , _source_geomname , _source_geomname ,  
_source_geomname  
,_source_schemaname , _source_tablename  
,_source_schemaname , _source_tablename , col_geom_graph_ori_name  
, col_geom_graph_ori_name , _source_geomname , col_geom_graph_ori_name ,  
_source_geomname , col_geom_graph_ori_name , _source_pkname , _source_pkname  
  
,_source_schemaname , _source_tablename , col_geom_graph_ori_name ,  
_source_pkname, _source_pkname  
,_source_pkname , _source_pkname , _source_pkname  
);
```

```
Execute str_sql into return_up_ids;  
--RAISE NOTICE 'return_up_ids: %', return_up_ids.gids;  
EXIT WHEN return_up_ids.gids is null;  
i=i+1;
```

```
END LOOP;  
RAISE NOTICE 'str_sql: %', str_sql;  
RAISE NOTICE 'nombre de boucles : %', i;
```

```
--  
-- 2.3 add index  
--  
str_sql := format (  
$SQL$
```

```
CREATE INDEX sx_start_%I%I  
ON %I.%I  
USING gist  
(St_startpoint(%I))  
;
```

```
CREATE INDEX sx_end_%I%I  
ON %I.%I  
USING gist  
(St_endpoint(%I))  
;
```

```
$SQL$  
, col_geom_graph_ori_name, _source_tablename  
,_source_schemaname , _source_tablename  
, col_geom_graph_ori_name  
  
, col_geom_graph_ori_name, _source_tablename  
,_source_schemaname , _source_tablename  
, col_geom_graph_ori_name  
)
```

```
/* Annexe_4 f_reseau_hydro_strahler_rank_v01_04.sql */
```

```
;  
  
RAISE NOTICE 'str_sql: %', str_sql;  
Execute str_sql;  
  
--  
-- 3. Compute Strahler rank on oriented graph geom  
--  
  
--  
-- 3.1 find stream heads => Strahler rank = 1  
--  
  
str_sql := format (  
  $SQL$  
  with enfant_without_parent as (  
    SELECT  
      enfant.%I as gid_enfant  
      ,parent.%I AS gid_parent  
    FROM  
      %I.%I enfant  
      LEFT JOIN %I.%I parent on  
        (st_intersects(st_endpoint(parent.%I),st_startpoint(enfant.%I)) and parent.%I !=  
        enfant.%I )  
    WHERE enfant.%I is not null AND parent.%I is null AND parent.%I IS NULL  
  )  
  UPDATE %I.%I t_up SET %I = 1 WHERE t_up.%I in (SELECT gid_enfant FROM  
  enfant_without_parent)  
  ;  
  
  $SQL$  
  , _source_pkname  
  , _source_pkname  
  , _source_schemaname , _source_tablename  
  , _source_schemaname , _source_tablename  
  , col_geom_graph_ori_name  
  , col_geom_graph_ori_name  
  , _source_pkname  
  , _source_pkname  
  , col_geom_graph_ori_name, col_strahler_rank_name, _source_pkname  
  
  , _source_schemaname , _source_tablename , col_strahler_rank_name, _source_pkname  
  )  
  
;  
RAISE NOTICE 'str_sql: %', str_sql;  
Execute str_sql;  
  
--  
-- 3.2 walk on graph (head => root end)  
--  
i=0;  
LOOP  
str_sql := format (  
  $SQL$  
  with enfant_not_ranking_find_parents_list as (  
    SELECT  
      enfant.%I as gid_enfant  
      ,array_agg(parent.%I) agg_strahler_rank  
    FROM  
      %I.%I enfant  
      LEFT JOIN %I.%I parent  
        on (st_intersects(st_endpoint(parent.%I),st_startpoint(enfant.%I)) and parent.%I  
        != enfant.%I )  
    WHERE enfant.%I is not null AND enfant.%I is null
```

```
/* Annexe_4 f_reseau_hydro_strahler_rank_v01_04.sql */
```

```
group by enfant.%I
)
, enfant_not_ranking_find_parents_list_with_not_null_rank as (
select *
, xx_99_utils.f_reseau_hydro_strahler_rank_compute_for_fild_v01_01 (agg_strahler_rank)
from enfant_not_ranking_find_parents_list where agg_strahler_rank::text NOT ilike
'%NULL%'
)
UPDATE %I.%I t_up SET %I = (
SELECT xx_99_utils.f_reseau_hydro_strahler_rank_compute_for_fild_v01_01 (
agg_strahler_rank)
FROM enfant_not_ranking_find_parents_list_with_not_null_rank t_rank
WHERE t_rank.gid_enfant = t_up.%I)
WHERE t_up.%I in (SELECT gid_enfant FROM
enfant_not_ranking_find_parents_list_with_not_null_rank)
RETURNING t_up.%I as gids
;

$SQL$
, _source_pkname
, col_strahler_rank_name
, _source_schename , _source_tablename
, _source_schename , _source_tablename
, col_geom_graph_ori_name , col_geom_graph_ori_name , _source_pkname , _source_pkname
, col_geom_graph_ori_name , col_strahler_rank_name
, _source_pkname

, _source_schename , _source_tablename , col_strahler_rank_name
, _source_pkname
, _source_pkname
, _source_pkname
)

;

Execute str_sql into return_up_ids;
EXIT WHEN return_up_ids.gids is null;
i=i+1;

END LOOP;
RAISE NOTICE 'str_sql: %', str_sql;
RAISE NOTICE 'nombre de boucles : %', i;
RAISE NOTICE 'str_sql: %', str_sql;

RETURN 'Traitement ok. Le rang de Strahler a été calculé dans la table : ' ||
_source_schename || '.' || _source_tablename || ' (strahler_rank) et (geom_graph_ori)
contient les géométries orientées.' ;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
```

```
/* Annexe_5 conservation_geom_utile_referentiel_ce_exemple.sql */
```

```
/*----- Copie des enregistrements filtrés -----*/
```

```
DROP TABLE IF EXISTS      mon_schema.t_cours_ref_ddtm_et_ign_corr_graph_grass_geom ;
CREATE TABLE
    mon_schema.t_cours_ref_ddtm_et_ign_corr_graph_grass_geom AS
SELECT
    *
FROM
    mon_schema.t_cours_ref_ddtm_et_ign_corr_graph
WHERE
    correction_result not like 'ISOLE' ;
```

```
/*
----- Suppression des colonnes géométriques inutiles (notre version de GRASS ne peut pas
charger une table avec plusieurs colonnes géométriques) -----
*/
```

```
ALTER TABLE m
    on_schema.t_cours_ref_ddtm_et_ign_corr_graph_grass_geom
DROP COLUMN
    geom_graph;
ALTER TABLE
    mon_schema.t_cours_ref_ddtm_et_ign_corr_graph_grass_geom
DROP COLUMN
    geom_graph_ori;
ALTER TABLE
    mon_schema.t_cours_ref_ddtm_et_ign_corr_graph_grass_geom
DROP COLUMN
    geom_ori;
```

```
/* Annexe_6 create_TBV_Final_without_carac_v03.sql */
```

```
/*
```

```
--V03 :  
  -- confirme utilisation de geom pour longueur du chemin le plus long  
  --  
  
-- Etape 1 : préparation : Ajout des rangs de strahler et gid du tronçon suivant à chaque  
tronçon de Cours d'eau dans la Table des Cours d'eau  
  
-- opération réalisée dans le stcipt de génération de _corr_graph
```

```
*/  
/*
```

```
Recursive pour la création de la table des bv tête de bv  
*/
```

```
drop table if exists m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract ;
```

```
CREATE TABLE m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract as
```

```
/*
```

```
-- Récursive sur la table de graph qui remonte tous les segments correspondant aux critères de  
selection des [ segments aval de tête de BV] : cad de rang de strahler <= 2 et qui vont  
former à leur extrémité un cours d'eau avec un rang de strahler >= 3  
-- la récursive liste pour tous [ segments aval de tête de BV] les chemins amonts (partiels  
justqu'aux cghemins complets)
```

```
*/
```

```
with graph_up_ce_rank_01_and_02_exit_in_rank_3_or_more as (  
  WITH RECURSIVE parcourt_up_graph(gid_graph_end, gid_graph, graph_gid_next,  
  strahler_after_outlet, strahler_before_outlet, strahler_rank, chemin, profondeur, longueur  
  --, geom_path  
  ,boucle) AS (  
    SELECT  
      g.gid_graph as gid_graph_end  
      , g.gid_graph  
      , g.graph_gid_next  
      , g.strahler_after_outlet  
      , g.strahler_before_outlet  
      , g.strahler_rank  
      , ARRAY[g.gid_graph] as chemin  
      , 1 as profondeur  
      , st_length(geom) as longueur  
      /*--, st_multi(geom_graph_ori)::geometry(multilinestring,2154) as geom_path*/  
      , false  
    FROM m060_milieux_continuite_tetes_de_bv.t_cours_ref_ddtm_et_ign_corr_graph g  
    WHERE g.strahler_after_outlet >=3 AND g.strahler_rank <=2 -- AND gid_graph in  
    (3660,3655)  
    UNION ALL  
    SELECT  
      sg.gid_graph_end  
      , g.gid_graph  
      , g.graph_gid_next  
      , g.strahler_after_outlet  
      , g.strahler_before_outlet  
      , g.strahler_rank  
      , chemin || g.gid_graph as chemin  
      , sg.profondeur +1 as profondeur  
      , st_length(g.geom)+sg.longueur as longueur  
      /*-- ,  
      st_multi(st_linemerge(st_union(sg.geom_path,g.geom_graph_ori))::geometry(multilin  
estring,2154) as geom_path*/  
      , g.gid_graph = ANY(chemin) as boucle  
    FROM m060_milieux_continuite_tetes_de_bv.t_cours_ref_ddtm_et_ign_corr_graph g  
      , parcourt_up_graph sg  
    WHERE  
g.graph_gid_next = sg.gid_graph AND NOT boucle
```

```
)
```

```
SELECT distinct gid_graph_end, gid_graph as gid_graph_start, strahler_rank,
```

```
/* Annexe_6 create_TBV_Final_without_carac_v03.sql */
```

```
    strahler_after_outlet, strahler_before_outlet, chemin, longueur
    --, geom_path
    , profondeur
    FROM parcourt_up_graph order by gid_graph_end asc, profondeur asc
)
/*
* liste des gid_graph (ce) composant les bv (id du future bv = gid_graph_end) et les
gid_graph sont les cours d'eau (geom) qui composent une tête de bv.
* Il suffira d'associer les surface de bv aux cours d'eau qui traversent les surfaces de BV.
*/
, list_graph_ce_rank_01_and_02_exit_in_3_or_up as (
    with w_col as (
        select
            distinct gid_graph_end
            , unnest(chemin) gid_graph
        from graph_up_ce_rank_01_and_02_exit_in_rank_3_or_more
    )

    select * from w_col order by gid_graph_end
)

/*
* Calcul d'indicateurs pour chaque gid_graph_end (= aval de tête de bv)
*/
, max_lengthgraph_and_nextrank_for_rank_01_and_02_exit_in_3_or_up_union as (

    select gid_graph_end
    , max(longueur) max_length_ce
    , max(strahler_rank) as max_strahler_rank
    , max(strahler_after_outlet) as strahler_after_outlet
    , max(strahler_before_outlet) as strahler_before_outlet

    from graph_up_ce_rank_01_and_02_exit_in_rank_3_or_more
    group by gid_graph_end
)

/*
* Récupération du chemin (liste des gid_graph) le plus long pour chaque gid_graph_end (=
aval de tête de bv)
*/
, geom_pathgraph_for_rank_01_and_02_exit_in_3_or_up_union as (

    select max_path.gid_graph_end
    , (select chemin.chemin from graph_up_ce_rank_01_and_02_exit_in_rank_3_or_more
chemin where max_path.gid_graph_end=chemin.gid_graph_end and max_path.max_length_ce
= chemin.longueur limit 1) max_path

    from max_lengthgraph_and_nextrank_for_rank_01_and_02_exit_in_3_or_up_union max_path
)

/*
* Décomposition ordonnée des chemins (liste des gid_graph) les plus long pour chaque
gid_graph_end (= aval de tête de bv)
*/
, geom_pathmaxlength as (
    select gid_graph_end
    , unnest(max_path) gid_graph
    , generate_subscripts(max_path, 1) AS idx_invert_order -- order inverse d'écoulement car
la récursive remontait le réseau de graph

from geom_pathgraph_for_rank_01_and_02_exit_in_3_or_up_union
)
```

```
/* Annexe_6 create_TBV_Final_without_carac_v03.sql */
```

```
/*
 * Recherche du Décomposition ordonnée des chemins (liste des gid_graph) les plus long pour
 * chaque gid_graph_end (= aval de tête de bv)
 */

, geom_pathmaxlength_gid_and_idx_start as ( -- recherche du début du path = max de
invert_order)
  select gid_graph_end
  , gid_graph
  , idx_invert_order
  , max (idx_invert_order) over(partition by gid_graph_end) as
  idx_invert_order_start_stream
  from geom_pathmaxlength
)
, geom_pathmaxlength_gid_and_idx_start_and_gid_start as ( -- attribue le gid_graph start
pour chaque gid_graph_end)
  select
  g.*
  ,st.gid_graph as gid_graph_start
  from
  geom_pathmaxlength_gid_and_idx_start g
  LEFT JOIN geom_pathmaxlength_gid_and_idx_start st on (st.gid_graph_end = g.gid_graph_end
  AND st.idx_invert_order= st.idx_invert_order_start_stream)
)

--select * from geom_pathmaxlength_gid_and_gid_start
-- geometry du chemin le plus long (ce) et start point / end point
, geom_union_pathmaxlength as (
select l.gid_graph_end
, st_multi(st_union(g_path.geom))::geometry(multilinestring,2154) as geom_max_path
, st_startpoint(g_start.geom_ori)::geometry(point,2154) as geom_max_path_start_pt
, st_endpoint(g_end.geom_ori)::geometry(point,2154) as geom_max_path_end_pt
from geom_pathmaxlength_gid_and_idx_start_and_gid_start l
LEFT JOIN m060_milieux_continuite_tetes_de_bv.t_cours_ref_ddtm_et_ign_corr_graph g_path on
(g_path.gid_graph=l.gid_graph)
LEFT JOIN m060_milieux_continuite_tetes_de_bv.t_cours_ref_ddtm_et_ign_corr_graph g_start on
(l.gid_graph_start= g_start.gid_graph)
LEFT JOIN m060_milieux_continuite_tetes_de_bv.t_cours_ref_ddtm_et_ign_corr_graph g_end on (
l.gid_graph_end = g_end.gid_graph)
group by l.gid_graph_end , g_start.geom_ori, g_end.geom_ori, gid_graph_end
)

--select * from geom_union_pathmaxlength --where gid_graph_end = 5001

, geom_ce_bv as (
  with lst_geom_bv_graph as (
    SELECT
      l.gid_graph_end
      , l.gid_graph
      , graph.strahler_rank
      , graph.geom_graph_ori
      , sbv.geom
      , sbv.gid_mbv
    FROM list_graph_ce_rank_01_and_02_exit_in_3_or_up l
    LEFT JOIN m060_milieux_continuite_tetes_de_bv.t_cours_ref_ddtm_et_ign_corr_graph graph
    on (l.gid_graph=graph.gid_graph)
    JOIN LATERAL (select mbv.geom as geom , mbv.gid as gid_mbv from
    m060_milieux_continuite_tetes_de_bv.t_micro_bv_ddtm_ign mbv where st_intersects
    (graph.geom,mbv.geom) order by st_length(st_intersection(graph.geom,mbv.geom)) desc
    limit 1) as sbv on true
  )
  -- liste des anastomoses de rang <=2
```

```
/* Annexe_6 create_TBV_Final_without_carac_v03.sql */
```

```
, anast_with_rank_sup_or_equal_3 as (  
SELECT distinct a.gid  
  
FROM m060_milieux_continuite_tetes_de_bv.t_cours_ref_ddtm_et_ign_corr_anast a  
JOIN m060_milieux_continuite_tetes_de_bv.t_cours_ref_ddtm_et_ign_corr_graph graph on  
(st_intersects(st_buffer(a.geom_pt_on_surface,0.01),graph.geom_graph_ori) and  
graph.strahler_rank>=3)  
  
)  
/*  
-- ajout des bv formés par les limites d'anastomoses : garde le chemin avec le strahler max  
*/  
attrib_anast_graph_end as (  
select  
a.geom  
, (select gid_graph_end from lst_geom_bv_graph lst_graph where st_intersects(a.geom,  
lst_graph.geom_graph_ori) order by lst_graph.strahler_rank desc limit 1) as gid_graph_end  
FROM m060_milieux_continuite_tetes_de_bv.t_cours_ref_ddtm_et_ign_corr_anast a  
WHERE a.gid not in (select alst.gid from anast_with_rank_sup_or_equal_3 alst)  
  
)  
/*  
-- add bv qui touchent anast qui possèdent un graph_end et qui n'ont pas déjà un  
graph_end d'attribué  
*/  
bv_anast as (  
select  
(select a.gid_graph_end from attrib_anast_graph_end a where st_intersects  
(a.geom,mbv.geom) order by st_length(st_intersection(a.geom,mbv.geom)) desc limit  
1) as gid_graph_end  
, mbv.geom  
from m060_milieux_continuite_tetes_de_bv.t_micro_bv_ddtm_ign mbv  
WHERE mbv.gid not in (select gid_mbv from lst_geom_bv_graph)  
)  
/*  
--select * from anast_with_rank_sup_or_equal_3  
--)  
--select * from geom_ce_bv  
  
a.gid_graph_end  
, (select mbv.geom as geom from m060_milieux_continuite_tetes_de_bv.t_micro_bv_ddtm_ign  
mbv where st_intersects (a.geom,mbv.geom) AND mbv.gid not in (select gid_mbv from  
lst_geom_bv_graph) order by st_length(st_intersection(a.geom,mbv.geom)) desc limit 1)  
as geom  
FROM attrib_anast_graph_end a  
WHERE gid_graph_end is not null  
*/  
  
-- union des bv de graph et d'anast  
, w_union as (  
select gid_graph_end , geom FROM lst_geom_bv_graph  
union all  
SELECT gid_graph_end , geom FROM bv_anast where gid_graph_end is not null  
  
)  
  
select gid_graph_end, st_union(geom) as geom from w_union  
GROUP BY gid_graph_end  
)  
  
, w_prepare as (  
select  
bv.gid_graph_end
```

```
/* Annexe_6 create_TBV_Final_without_carac_v03.sql */
```

```
, attrib.max_length_ce
, attrib.max_strahler_rank
, attrib.strahler_after_outlet
, attrib.strahler_before_outlet
, max_path.geom_max_path
, max_path.geom_max_path_start_pt
, max_path.geom_max_path_end_pt

, st_multi(ST_MakePolygon(ST_ExteriorRing(
St_collectionextract((st_dump(bv.geom)).geom,3)))) ::geometry(multipolygon,2154) as geom

from geom_ce_bv bv
LEFT JOIN max_lengthgraph_and_nextrank_for_rank_01_and_02_exit_in_3_or_up_union attrib on
(attrib.gid_graph_end=bv.gid_graph_end)
LEFT JOIN geom_union_pathmaxlength max_path on (max_path.gid_graph_end=bv.gid_graph_end)
)
, w_prepare_02 as (
select row_number() over() as gid
, gid_graph_end
, max_length_ce
, max_strahler_rank
, strahler_after_outlet
, strahler_before_outlet
,
st_collectionextract(st_makevalid(st_multi(st_collectionextract(st_makevalid(st_union(geom)),
3))),3)::geometry(multipolygon,2154) as geom
, geom_max_path
, geom_max_path_start_pt
, geom_max_path_end_pt
from w_prepare
group by
gid_graph_end
, max_length_ce
, max_strahler_rank
, strahler_after_outlet
, strahler_before_outlet
, geom_max_path
, geom_max_path_start_pt
, geom_max_path_end_pt
)
select p.*
,
st_multi(st_union(st_collectionextract(st_intersection(p.geom,ce.geom),2)))::geometry(multilinestring,2154) as geom_ce
, st_multi(st_buffer(st_union(st_intersection(p.geom,ce.geom)),10))
::geometry(multipolygon,2154)as geom_ce_buffer_10
from w_prepare_02 p
LEFT JOIN m060_milieux_continuite_tetes_de_bv.t_cours_ref_ddtm_et_ign_corr_graph ce on
(st_intersects(p.geom, ce.geom) )
group by p.gid
, p.gid_graph_end
, p.geom
, p.max_length_ce
, p.max_strahler_rank
, p.strahler_after_outlet
, p.strahler_before_outlet
, p.geom_max_path
, p.geom_max_path_start_pt
, p.geom_max_path_end_pt
;

DROP INDEX if exists m060_milieux_continuite_tetes_de_bv.sidx_t_tbv_01_without_caract_geom;
CREATE INDEX sidx_t_tbv_01_without_caract_geom
ON m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract
USING gist
(geom);
```

```
/* Annexe_6 create_TBV_Final_without_carac_v03.sql */
```

```
DROP INDEX if exists m060_milieux_continuite_tetes_de_bv.sidx_t_tbv_01_without_caract_geom_ce ;  
CREATE INDEX sidx_t_tbv_01_without_caract_geom_ce  
  ON m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract  
  USING gist  
  (geom_ce);
```

```
DROP INDEX if exists  
m060_milieux_continuite_tetes_de_bv.sidx_t_tbv_01_without_caract_geom_ce_buffer_10 ;  
CREATE INDEX sidx_t_tbv_01_without_caract_geom_ce_buffer_10  
  ON m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract  
  USING gist  
  (geom_ce_buffer_10);
```

```
/*
```

```
supression des trops petits bv
```

```
*/
```

```
DELETE from m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract WHERE  
St_area(geom)<=31000 ; -- (cad : 3.1 ha)
```

```
/*Processing Chain indicateurs_caracterisation_tbv by acoudart*/
```

```
/*Process 0 / Fonction tbv_surface.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) SELECT gid, 1321, st_area(geom)/1000000 FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract
```

```
/*Process 1 / Fonction tbv_pente_moy.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) SELECT bv.gid, 1323 , (avg(r.slope_percent))::numeric(5,2) FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
r020_territoire_physique.t_mnt_rge_slope_grass_percent r WHERE st_intersects(bv.geom,
r.geom) GROUP BY bv.gid
```

```
/*Process 2 / Fonction tbv_pente_moy_longest_path.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) WITH w1 as ( SELECT bv.gid , r.alti::numeric(5,2) as point_haut FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv LEFT JOIN
r020_territoire_physique.t_mnt_alti_aspect_slope_percent_grass r ON
st_intersects(bv.geom_max_path_start_pt, r.geom) GROUP BY bv.gid, r.alti ), w2 as( SELECT
bv.gid , r.alti::numeric(5,2) as point_bas , bv.max_length_ce as longest_path FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv LEFT JOIN
r020_territoire_physique.t_mnt_alti_aspect_slope_percent_grass r ON
st_intersects(bv.geom_max_path_end_pt, r.geom) GROUP BY bv.gid, r.alti, bv.max_length_ce ),
w3 as( SELECT w1.gid , ((point_haut - point_bas)/longest_path)*100::numeric(5,2) pente FROM
w1, w2 WHERE w1.gid = w2.gid ), -- Selection de la plus petite pente w4 as ( SELECT
min(pente) min FROM w3 WHERE pente >0 ) -- Afin d éliminer les valeurs null et négative on
leur attribut la valeur minimal de la distribution diviser par 3 SELECT w3.gid, 1356 , CASE
WHEN w3.pente < w4.min THEN w4.min/3 ELSE w3.pente END FROM w3, w4
```

```
/*Process 3 / Fonction tbv_gravelius.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) WITH w_prepare AS( SELECT gid ,st_area(geom) as surface_bv ,st_perimeter(geom) as
perimetre_bv FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract ) SELECT gid ,
1324 , (perimetre_bv/(2*(1/pi()*surface_bv)))::numeric(5,4) as gravelius FROM w_prepare
GROUP BY gid, gravelius
```

```
/*Process 4 / Fonction tbv_passini.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) WITH w1 as ( SELECT bv.gid , r.alti::numeric(5,2) as point_haut FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv LEFT JOIN
r020_territoire_physique.t_mnt_alti_aspect_slope_percent_grass r ON
st_intersects(bv.geom_max_path_start_pt, r.geom) GROUP BY bv.gid, r.alti ), w2 as( SELECT
bv.gid , r.alti::numeric(5,2) as point_bas , bv.max_length_ce as longest_path ,
st_area(bv.geom) as area FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv
LEFT JOIN r020_territoire_physique.t_mnt_alti_aspect_slope_percent_grass r ON
st_intersects(bv.geom_max_path_end_pt, r.geom) GROUP BY bv.gid, r.alti, bv.max_length_ce,
bv.geom ), w3 as( SELECT w1.gid , ((point_haut - point_bas)/longest_path)*100::numeric(5,2)
pente FROM w1, w2 WHERE w1.gid = w2.gid ), -- Selection de la plus petite pente w4 as (
SELECT min(pente) min FROM w3 WHERE pente >0 ), -- Afin d éliminer les valeurs null et
négative on leur attribut la valeur minimal de la distribution diviser par 3. w5 as( SELECT
w3.gid , CASE WHEN w3.pente < w4.min THEN w4.min/3 ELSE w3.pente END AS pente FROM w3, w4 )
SELECT w5.gid, 1357 , 0.108*(1/((w2.area/1000000)*(w2.longest_path)))/(1/w5.pente) FROM w5,
w2 WHERE w5.gid = w2.gid
```

```
/*Process 5 / Fonction tbv_densite_ce.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) WITH w_prepare as ( SELECT b.gid , st_area(b.geom) as area ,
st_multi(ST_union(st_collectionextract(xx_99_utils.st_intersection_btr(b.geom,
a.geom),2)))::geometry(multilinestring,2154) as geom_ce FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract b LEFT JOIN
m060_milieux_continuite_tetes_de_bv.t_cours_ref_ddtm_et_ign_corr_graph a on (
st_intersects(b.geom,a.geom)) GROUP BY b.gid , b.geom ) SELECT gid , 1325 , st_length
(geom_ce)/area FROM w_prepare
```

```
/*Process 6 / Fonction tbv_densite_point_bas.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) WITH w_prepare as ( SELECT b.gid , st_area(b.geom) as area ,
st_multi(ST_union(st_collectionextract(xx_99_utils.st_intersection_btr(b.geom,
a.wkb_geometry),2)))::geometry(multilinestring,2154) as geom_ce FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract b LEFT JOIN
_agent_acoudart.t_mnt_sc_sk_stream_4000 a on ( st_intersects(b.geom,a.wkb_geometry)) GROUP
```

```
/* Annexe_7 indicateurs_caracterisation_tbv.sql */
```

```
BY b.gid , b.geom ) SELECT gid , 1326 , coalesce(st_length(geom_ce)/area,0) FROM w_prepare
```

```
/*Process 7 / Fonction tbv_position_in_reseau_hydro.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param, valeur) SELECT gid, 1358 , CASE WHEN strahler_after_outlet = 3 AND strahler_before_outlet < 3 THEN 3 WHEN strahler_after_outlet in (3, 4) AND strahler_before_outlet > 2 THEN 2 WHEN strahler_after_outlet > 4 THEN 1 END FROM
```

```
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract
```

```
/*Process 8 / Fonction tbv_taux_plan_eau_impact_ce.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param, valeur) select gid, 1327, (coalesce( _agent_acoudart.f_polygon_densite_of_geom_table_v02_00 ( _input_geom_recherche := geom_ce_buffer_10 /*-- geom dans laquelle l'intersection sera réalisée */ , _input_geom_limite := geom /*-- geom limitant le résultat final (aucun résultat en dehors de cette limite ne sera pris en compte) */ , _input_geom_base_ratio := geom_ce_buffer_10 /*-- geom utilisée pour le calcul du ratio final */ , _input_geom_unite := 1 /*-- Unité métrique de la géométrie _input_geom_ref_ratio (10000 pour ha...) */ , _tcompare_schemaname := 'sysma_couches' /*-- Schema de la table contenant les infos à intersecter avec la geom en input */ , _tcompare_tablename:= 'plan_d_eau' /*-- Table contenant les infos à intersecter avec la geom en inp */ , _tcompare_geom := 'geom' /*-- Geom des infos à intersecter avec la geom en input */ , _tcompare_geomtype := 'POLYGON' /*-- Type geom de base (POLYGON, LINESTRING, POINT) */ , _tcompare_unite := 1 /*-- Unité métrique des infos à intersecter(10000 pour ha...) */ , _tcompare_where_clause := 'case when conn_sig = true and conn_terra not in (''en derivation'', ''isole'') then 1 when conn_sig = true and conn_terra is null then 1 else 0 end = 1' /*-- [optionnel] : Clause Where */ , _debug_mode := FALSE /*-- [optionnel] : TRUE pour afficher la requete dans la console */),0.00)*100)::numeric(5,2) FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract
```

```
/*Process 9 / Fonction tbv_taux_zh_connect_ce.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param, valeur) select gid, 1328, (coalesce( _agent_acoudart.f_polygon_densite_of_geom_table_v02_00 ( _input_geom_recherche := geom_ce_buffer_10 /*-- geom dans laquelle l'intersection sera réalisée */ , _input_geom_limite := geom /*-- geom limitant le résultat final (aucun résultat en dehors de cette limite ne sera pris en compte) */ , _input_geom_base_ratio := geom_ce_buffer_10 /*-- geom utilisée pour le calcul du ratio final */ , _input_geom_unite := 1 /*-- Unité métrique de la géométrie _input_geom_ref_ratio (10000 pour ha...)*/ , _tcompare_schemaname := 'm060_milieux_continuite_dec_agreg' /*-- Schema de la table contenant les infos à intersecter avec la geom en input */ , _tcompare_tablename:= 't_zh_agreg' /*-- Table contenant les infos à intersecter avec la geom en input */ , _tcompare_geom := 'geom' /*-- Geom des infos à intersecter avec la geom en input */ , _tcompare_geomtype := 'POLYGON' /*-- Type geom de base (POLYGON, LINESTRING, POINT) */ , _tcompare_unite := 1 /*-- Unité métrique des infos à intersecter(10000 pour ha...) */ , _tcompare_where_clause := '(zh_sage != 6 or zh_sage != 7)' /*-- [optionnel] : Clause Where */ , _debug_mode := FALSE /*-- [optionnel] : TRUE pour afficher la requete dans la console */),0.00)*100)::numeric(5,2) FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract
```

```
/*Process 10 / Fonction tbv_taux_ripi_connect_ce.php*/
```

```
with bv_area_bois as ( select bv.gid , xx_99_utils.st_intersection_safe(bv.geom,st_collectionextract(xx_99_utils.st_union_safe(st_collect(ocs.geom),0.000001),3)) as geom FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs on ( st_intersects(bv.geom, ocs.geom) AND ( (ocs.data_theme ilike 'Boisement' AND ocs.source_data#>> '{tfv_g11}'::TEXT[] in ( 'Forêt fermée feuillus', 'Forêt ouverte feuillus', 'Forêt fermée mixte', 'Forêt ouverte mixte', 'Lande' ) ) OR ocs.data_theme in ( 'forêt feuillus', 'lande ligneuse' ) ) ) group by bv.gid, bv.geom ) , area_haies as ( SELECT bv.gid , xx_99_utils.st_intersection_safe(bv.geom,st_collectionextract(xx_99_utils.st_union_safe(st_collect(h.geom),0.000001),3)) as geom FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv LEFT JOIN m060_milieux_continuite_tetes_de_bv.bd_haie_vege h on (st_intersects(bv.geom, h.geom) AND (h.nature in ('Haie') or h.nature is null)) group by bv.gid, bv.geom ) , concat_areas as ( select gid, geom FROM bv_area_bois union all select gid, geom FROM area_haies ) , grouping_area as ( select gid , st_collectionextract(xx_99_utils.st_union_safe(st_collect(geom),0.000001),3) as geom FROM concat_areas group by gid ) insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param, valeur) SELECT bv.gid , 1329 , coalesce((sum(st_area(xx_99_utils.st_intersection_safe(bv.geom_ce_buffer_10, ripi.geom)))) / st_area(bv.geom_ce_buffer_10)*100),0)::numeric(5,2) FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv LEFT JOIN grouping_area ripi ON (bv.gid=ripi.gid) GROUP BY bv.gid, bv.geom_ce_buffer_10
```

```
/*Process 11 / Fonction tbv_taux_mares_connect_ce.php*/
```

```
/* Annexe_7 indicateurs_caracterisation_tbv.sql */
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) select gid, 1330, (coalesce( _agent_acoudart.f_polygon_densite_of_geom_table_v02_00
( _input_geom_recherche := geom_ce_buffer_10 /*-- geom dans laquelle l'intersection sera
réalisée */ , _input_geom_limite := geom /*-- geom limitant le résultat final (aucun résultat
en dehors de cette limite ne sera pris en compte)*/ , _input_geom_base_ratio :=
geom_ce_buffer_10 /*-- geom utilisée pour le calcul du ratio final */ , _input_geom_unite :=
1 /*-- Unité métrique de la géométrie _input_geom_ref_ratio (10000 pour ha...) */ ,
_tcompare_schemaname := 'sysma_couches' /*-- Schema de la table contenant les infos à
intersecter avec la geom en input */ , _tcompare_tablename:= 'plan_d_eau' /*-- Table
contenant les infos à intersecter avec la geom en input */ , _tcompare_geom := 'geom' /*--
Geom des infos à intersecter avec la geom en input */ , _tcompare_geomtype := 'POLYGON' /*--
Type geom de base (POLYGON, LINESTRING, POINT) */ , _tcompare_unite := 1 /*-- Unité métrique
des infos à intersecter(10000 pour ha...) */ , _tcompare_where_clause := '(st_area(geom) <=
2000 and (conn_terra in (''isole'', ''en derivation'') or conn_terra is null)and conn_sig is
false)' /*-- [optionnel] : Clause Where */ , _debug_mode := FALSE /*-- [optionnel] : TRUE
pour afficher la requete dans la console */),0.00)*100)::numeric(5,2) as dens_mares_ce FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract
```

```
/*Process 12 / Fonction tbv_taux_impermeable_connect_ce.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) select bv.gid, 1335, coalesce(res.res ,0)::numeric from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
sum((st_area(st_intersection(bv.geom_ce_buffer_10, ocs.geom)) /
st_area(bv.geom_ce_buffer_10)*100)::numeric(5,2)) as res FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE st_intersects(bv.geom,
ocs.geom) AND (ocs.data_theme ilike 'Transport' AND ocs.source_data#>>'{nature}':::TEXT[ ]not
in('Chemin' , 'Route empierrée') OR ocs.data_theme in ('bâti dense', 'bâti diffus', 'zones
ind et com', 'surface route') OR ocs.data_theme ilike 'Urbain') GROUP BY bv.gid, bv.geom )
res on (res.gid=bv.gid)
```

```
/*Process 13 / Fonction tbv_taux_plan_eau_tbv.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) SELECT gid, 1333 , (coalesce( _agent_acoudart.f_polygon_densite_of_geom_table_v02_00
( _input_geom_recherche := geom /*-- geom dans laquelle l'intersection sera réalisée */ ,
_input_geom_limite := geom /*-- geom limitant le résultat final (aucun résultat en dehors de
cette limite ne sera pris en compte) */ , _input_geom_base_ratio := geom /*-- geom utilisée
pour le calcul du ratio final */ , _input_geom_unite := 1 /*-- Unité métrique de la géométrie
_input_geom_ref_ratio (10000 pour ha...) */ , _tcompare_schemaname := 'sysma_couches' /*--
Schema de la table contenant les infos à intersecter avec la geom en input */ ,
_tcompare_tablename:= 'plan_d_eau' /*-- Table contenant les infos à intersecter avec la geom
en input */ , _tcompare_geom := 'geom' /*-- Geom des infos à intersecter avec la geom en
input */ , _tcompare_geomtype := 'POLYGON' /*-- Type geom de base (POLYGON, LINESTRING,
POINT) */ , _tcompare_unite := 1 /*-- Unité métrique des infos à intersecter(10000 pour
ha...) */ , _tcompare_where_clause := '' /*-- [optionnel] : Clause Where */ , _debug_mode :=
FALSE /*-- [optionnel] : TRUE pour afficher la requete dans la console
*/),0.00)*100)::numeric(5,2) FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract ORDER BY gid
```

```
/*Process 14 / Fonction tbv_taux_zh_tbv.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) SELECT bv.gid, 1334, coalesce(res.res, 0)::numeric(5,2) FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv LEFT JOIN ( SELECT bv.gid ,
sum((st_area(st_intersection(bv.geom, zh.geom)) / st_area(bv.geom)*100)::numeric(5,2)) as
res FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_dec_agreg.t_zh_agreg zh WHERE st_intersects(bv.geom , zh.geom) and
(zh.zh_sage != 6 or zh.zh_sage != 7) GROUP BY bv.gid, bv.geom ) res ON (res.gid=bv.gid)
```

```
/*Process 15 / Fonction tbv_taux_haie_tbv.php*/
```

```
with bv_area_not_bois as ( select bv.gid , case when st_area(st_union(ocs.geom)) IS NOT NULL
THEN st_difference(bv.geom,st_collectionextract(st_union(ocs.geom),3)) ELSE bv.geom END as
geom FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs on (st_intersects(bv.geom, ocs.geom)
AND ocs.data_theme ilike 'Boisement' ) group by bv.gid, bv.geom )insert into
m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param, valeur) SELECT
bv.gid, 1336, coalesce(res.res, 0)::numeric(5,2) FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv LEFT JOIN ( SELECT bv.gid ,
sum((st_area(xx_99_utils.st_intersection_safe(bv.geom, h.geom)) /
st_area(bv.geom)*100)::numeric(5,2)) as res FROM bv_area_not_bois bv ,
m060_milieux_continuite_tetes_de_bv.bd_haie_vege h WHERE st_intersects(bv.geom , h.geom)
AND (nature in ('Haie') or nature is null) GROUP BY bv.gid, bv.geom ) res ON (res.gid=bv.gid)
```

```
/* Annexe_7 indicateurs_caracterisation_tbv.sql */
```

```
/*Process 16 / Fonction tbv_taux_haie_efficace_tbv.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) SELECT gid, 1337, (coalesce( _agent_acoudart.f_polygon_densite_of_geom_table_v02_00
( _input_geom_recherche := geom /*-- geom dans laquelle l'intersection sera réalisée */ ,
_input_geom_limite := geom /*-- geom limitant le résultat final (aucun résultat en dehors de
cette limite ne sera pris en compte) */ , _input_geom_base_ratio := geom /*-- geom utilisée
pour le calcul du ratio final*/ , _input_geom_unite := 1 /*-- Unité métrique de la géométrie
_input_geom_ref_ratio (10000 pour ha...) */ , _tcompare_schemaname :=
'm060_milieux_continuite_dec_agreg' /*-- Schema de la table contenant les infos à
intersecter avec la geom en input */ , _tcompare_tablename:= 't_haies_agreg' /*-- Table
contenant les infos à intersecter avec la geom en input */ , _tcompare_geom := 'geom' /*--
Geom des infos à intersecter avec la geom en input */ , _tcompare_geomtype := 'LINESTRING'
/*-- Type geom de base (POLYGON, LINESTRING, POINT) */ , _tcompare_unite := 1 /*-- Unité
métrique des infos à intersecter(10000 pour ha...) */ , _tcompare_where_clause := '(h_ori_sig
= ''0'' AND (h_typ between 4 and 6) AND h_cont = ''T'')' /*-- [optionnel] : Clause Where */ ,
_debug_mode := FALSE /*-- [optionnel] : TRUE pour afficher la requete dans la console
*/),0.00)*100)::numeric(5,2) FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract
```

```
/*Process 17 / Fonction tbv_taux_mares_tbv.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) SELECT gid, 1338, (coalesce( _agent_acoudart.f_polygon_densite_of_geom_table_v02_00
( _input_geom_recherche := geom /*-- geom dans laquelle l'intersection sera réalisée */ ,
_input_geom_limite := geom /*-- geom limitant le résultat final (aucun résultat en dehors de
cette limite ne sera pris en compte) */ , _input_geom_base_ratio := geom /*-- geom utilisée
pour le calcul du ratio final */ , _input_geom_unite := 1 /*-- Unité métrique de la géométrie
_input_geom_ref_ratio (10000 pour ha...) */ , _tcompare_schemaname := 'sysma_couches' /*--
Schema de la table contenant les infos à intersecter avec la geom en input */ ,
_tcompare_tablename:= 'plan_d_eau' /*-- Table contenant les infos à intersecter avec la geom
en input */ , _tcompare_geom := 'geom' /*-- Geom des infos à intersecter avec la geom en
input */ , _tcompare_geomtype := 'POLYGON' /*-- Type geom de base (POLYGON, LINESTRING,
POINT) */ , _tcompare_unite := 1 /*-- Unité métrique des infos à intersecter(10000 pour
ha...) */ , _tcompare_where_clause := '(st_area(geom) <= 2000 and (conn_terra in (''isole'',
''en derivation'')) or conn_terra is null)and conn_sig is false)' /*-- [optionnel] : Clause
Where */ , _debug_mode := FALSE /*-- [optionnel] : TRUE pour afficher la requete dans la
console */),0.00)*100)::numeric(5,3) FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract
```

```
/*Process 18 / Fonction tbv_taux_impermeable_tbv.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) select bv.gid, 1332, coalesce(res.res, 0)::numeric from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
sum((st_area(st_intersection( bv.geom, ocs.geom)) / st_area(bv.geom)*100)::numeric(5,2)) as
res FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01_ocs WHERE st_intersects(bv.geom,
ocs.geom) AND (ocs.data_theme ilike 'Transport' AND ocs.source_data#>>'{nature}':::TEXT[ ]not
in('Chemin', 'Route empierrée ') OR ocs.data_theme in ('bâti dense', 'bâti diffus', 'zones
ind et com', 'surface route') OR ocs.data_theme ilike 'Urbain') GROUP BY bv.gid, bv.geom )
res on (res.gid=bv.gid)
```

```
/*Process 19 / Fonction tbv_volume_prelev_tbv.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) SELECT bv.gid, 1339,
(coalesce((sum(a.volume))/(st_area(bv.geom)/10000),0,00))::numeric(7,2) FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv LEFT JOIN
m060_milieux_continuite_tetes_de_bv.t_prelevement_bnpe_2016_all_dep a ON
st_intersects(bv.geom, a.geom) AND a.precision_localisation not in ('Coordonnées du
centroïde de la commune') GROUP BY bv.gid, bv.geom
```

```
/*Process 20 / Fonction tbv_scenario_calcul.php*/
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur_type, valeur, valeur_decompose) With lst_scnar as (SELECT * FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_scnar_ponde scnar WHERE CASE WHEN
'indice_pression_syloa' ='' THEN 1=1 ELSE scnar.nom_scenario ilike 'indice_pression_syloa%'
END ORDER BY scnar.ordre ASC) , scnar_and_pond_pc as ( SELECT pond.* ,
pond.valeur_param_ponde/sum(pond.valeur_param_ponde) over (partition by pond.scnar) as
pc_pond , lst_s.* FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_ponde pond JOIN
lst_scnar lst_s on (lst_s.nom_scenario = pond.scnar)) , w_prepare as ( SELECT bv.gid ,
scnar.nom_scenario , param.nom_param , val.valeur , scnar.valeur_param_ponde , sum( CASE
WHEN param.positif is TRUE THEN val.valeur ELSE 1.-val.valeur END * scnar.pc_pond) over
(partition by bv.gid , scnar.nom_scenario,scnar.res_id_param ) as valeur_indice ,
coalesce( CASE WHEN param.positif is TRUE THEN val.valeur ELSE 1.-val.valeur END *
scnar.pc_pond,0) valeur_pond_by_gid , scnar.res_id_param FROM
```

/* Annexe_7 indicateurs_caracterisation_tbv.sql */

```
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract AS bv LEFT JOIN
scenar_and_pond_pc AS scenar on (1=1) LEFT JOIN
m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result AS val on (bv.gid = val.gid_tbv AND
val.id_param = scenar.id_param AND val.valeur_type ilike '%' || scenar.filter_end_valeur_type)
LEFT JOIN m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac AS param on
(param.id_param = scenar.id_param) ORDER BY bv.gid , val.gid_tbv , scenar.nom_scenario ,
valeur_pond_by_gid desc ) SELECT bv.gid , bv.res_id_param , '[scenar_pond : ' ||
bv.nom_scenario || ']' as valeur_type , valeur_indice , string_agg( case when
bv.valeur_indice = 0. or bv.valeur_indice is null THEN NULL::TEXT else bv.nom_param ||
[val_pond : ' || (100.* bv.valeur_pond_by_gid / bv.valeur_indice)::numeric(5,2) || '% ,
val_brute : ' || bv.valeur || '%]' END, ' ** ' ) as decompose_indice_pression FROM w_prepare
bv WHERE valeur_pond_by_gid is not null GROUP BY bv.gid , bv.nom_scenario ,bv.valeur_indice
,bv.res_id_param;
```

```
/* Annexe_7bis indice_pression_tbv.sql */
```

```
/*Processing Chain indice_pression_tbv by acoudart*/
```

```
Process 0 / Fonction tbv_indice_pression_foret.php
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) select bv.gid, 1345, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom, ocs.geom)))) / st_area(bv.geom))*100)::numeric(5,2)
as res FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE st_intersects(bv.geom,
ocs.geom) AND ((ocs.data_theme ilike 'Boisement' AND ocs.source_data#>>'{tfv_g11}'::TEXT[]
in('Forêt fermée feuillus', 'Forêt ouverte feuillus', 'Forêt fermée mixte', 'Forêt ouverte
mixte', 'Lande')) OR ocs.data_theme in ('forêt feuillus', 'lande ligneuse')) GROUP BY
bv.gid, bv.geom ) res on (res.gid=bv.gid);insert into
m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param, valeur) select
bv.gid, 1346, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom_ce_buffer_10, ocs.geom)))) /
st_area(bv.geom_ce_buffer_10))*100)::numeric(5,2) as res FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE
st_intersects(bv.geom_ce_buffer_10, ocs.geom) AND ((ocs.data_theme ilike 'Boisement' AND
ocs.source_data#>>'{tfv_g11}'::TEXT[] in('Forêt fermée feuillus', 'Forêt ouverte feuillus',
'Forêt fermée mixte', 'Forêt ouverte mixte', 'Lande')) OR ocs.data_theme in ('forêt
feuillus', 'lande ligneuse')) GROUP BY bv.gid, bv.geom_ce_buffer_10 ) res on (res.gid=bv.gid);
```

```
Process 1 / Fonction tbv_indice_pression_prairie_perm.php
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) select bv.gid, 1347, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom, ocs.geom)))) / st_area(bv.geom))*100)::numeric(5,2)
as res FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE st_intersects(bv.geom,
ocs.geom) AND ((ocs.data_theme ilike 'Agriculture' AND
ocs.source_data#>>'{code_group}'::TEXT[]in('17', '18')) OR ocs.data_theme in ('prairie'))
GROUP BY bv.gid, bv.geom ) res on (res.gid=bv.gid);insert into
m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param, valeur) select
bv.gid, 1348, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom_ce_buffer_10, ocs.geom)))) /
st_area(bv.geom_ce_buffer_10))*100)::numeric(5,2) as res FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE
st_intersects(bv.geom_ce_buffer_10, ocs.geom) AND ((ocs.data_theme ilike 'Agriculture' AND
ocs.source_data#>>'{code_group}'::TEXT[]in('17', '18')) OR ocs.data_theme in ('prairie'))
GROUP BY bv.gid, bv.geom_ce_buffer_10 ) res on (res.gid=bv.gid);
```

```
Process 2 / Fonction tbv_indice_pression_prairie_temp.php
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) select bv.gid, 1349, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom, ocs.geom)))) / st_area(bv.geom))*100)::numeric(5,2)
as res FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE st_intersects(bv.geom,
ocs.geom) AND ((ocs.data_theme ilike 'Boisement' AND ocs.source_data#>>'{tfv_g11}'::TEXT[]
in ('Forêt fermée conifères', 'Forêt ouverte conifères', 'Peupleraie')) OR (ocs.data_theme
ilike 'Agriculture' AND ocs.source_data#>>'{code_group}'::TEXT[] in ('11', '16', '19')) OR
(ocs.data_theme ilike 'forêt conifères')) GROUP BY bv.gid, bv.geom ) res on
(res.gid=bv.gid);insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result
(gid_tbv, id_param, valeur) select bv.gid, 1350, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom_ce_buffer_10, ocs.geom)))) /
st_area(bv.geom_ce_buffer_10))*100)::numeric(5,2) as res FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE
st_intersects(bv.geom_ce_buffer_10, ocs.geom) AND ((ocs.data_theme ilike 'Boisement' AND
ocs.source_data#>>'{tfv_g11}'::TEXT[] in ('Forêt fermée conifères', 'Forêt ouverte
conifères', 'Peupleraie')) OR (ocs.data_theme ilike 'Agriculture' AND
ocs.source_data#>>'{code_group}'::TEXT[] in ('11', '16', '19')) OR (ocs.data_theme ilike
'forêt conifères')) GROUP BY bv.gid, bv.geom_ce_buffer_10 ) res on (res.gid=bv.gid);
```

```
Process 3 / Fonction tbv_indice_pression_urb_perm.php
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
```

```
/* Annexe_7bis indice_pression_tbv.sql */
```

```
valeur) SELECT bv.gid, 1342, coalesce((res.res),0) FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv LEFT JOIN ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom, ocs.geom)))) / st_area(bv.geom))*100)::numeric(5,2)
as res FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE st_intersects(bv.geom,
ocs.geom) AND (data_theme ilike 'Urbain' AND data_subtheme ilike 'Permeable') GROUP BY
bv.gid, bv.geom ) res ON (res.gid=bv.gid);insert into
m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param, valeur) SELECT
bv.gid, 1343, coalesce((res.res),0) FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv LEFT JOIN ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom_ce_buffer_10, ocs.geom)))) /
st_area(bv.geom_ce_buffer_10))*100)::numeric(5,2) as res FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE
st_intersects(bv.geom_ce_buffer_10, ocs.geom) AND (data_theme ilike 'Urbain' AND
data_subtheme ilike 'Permeable') GROUP BY bv.gid, bv.geom_ce_buffer_10 ) res ON
(res.gid=bv.gid);
```

```
Process 4 / Fonction tbv_indice_pression_culture.php
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) select bv.gid, 1351, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom, ocs.geom)))) / st_area(bv.geom))*100)::numeric(5,2)
as res FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE st_intersects(bv.geom,
ocs.geom) AND (ocs.data_theme in ('culture été', 'culture hiver') OR (ocs.data_theme ilike
'Agriculture' AND ocs.source_data#>>'{code_group}':TEXT[] in ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '14', '15', '24')))) GROUP BY bv.gid, bv.geom ) res on
(res.gid=bv.gid);insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result
(gid_tbv, id_param, valeur) select bv.gid, 1352, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom_ce_buffer_10, ocs.geom)))) /
st_area(bv.geom_ce_buffer_10))*100)::numeric(5,2) as res FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE
st_intersects(bv.geom_ce_buffer_10, ocs.geom) AND (ocs.data_theme in ('culture été',
'culture hiver') OR (ocs.data_theme ilike 'Agriculture' AND
ocs.source_data#>>'{code_group}':TEXT[] in ('1', '2', '3', '4', '5', '6', '7', '8',
'9', '14', '15', '24')))) GROUP BY bv.gid, bv.geom_ce_buffer_10 ) res on (res.gid=bv.gid);
```

```
Process 5 / Fonction tbv_indice_pression_vigne.php
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) select bv.gid, 1353, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom, ocs.geom)))) / st_area(bv.geom))*100)::numeric(5,2)
as res FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE st_intersects(bv.geom,
ocs.geom) AND ((ocs.data_theme ilike 'verger') OR (ocs.data_theme ilike 'vigne') OR
(ocs.data_theme ilike 'Agriculture' AND ocs.source_data#>>'{code_group}':TEXT[] in
('20', '25', '21')))) GROUP BY bv.gid, bv.geom ) res on (res.gid=bv.gid);insert into
m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param, valeur) select
bv.gid, 1354, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom_ce_buffer_10, ocs.geom)))) /
st_area(bv.geom_ce_buffer_10))*100)::numeric(5,2) as res FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE
st_intersects(bv.geom_ce_buffer_10, ocs.geom) AND ((ocs.data_theme ilike 'verger') OR
(ocs.data_theme ilike 'vigne') OR (ocs.data_theme ilike 'Agriculture' AND
ocs.source_data#>>'{code_group}':TEXT[] in ('20', '25', '21')))) GROUP BY bv.gid,
bv.geom_ce_buffer_10 ) res on (res.gid=bv.gid);
```

```
Process 6 / Fonction tbv_indice_pression_artificiel.php
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur) select bv.gid, 1341, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom, ocs.geom)))) / st_area(bv.geom))*100)::numeric(5,2)
as res FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE st_intersects(bv.geom,
ocs.geom) AND ((ocs.data_theme ilike 'Transport' AND ocs.source_data#>>'{nature}':TEXT[]not
in('Chemin', 'Route empierrée')) OR ocs.data_theme in ('bâti dense', 'bâti diffus', 'zones
ind et com', 'surface route') OR ocs.data_theme ilike 'Urbain') GROUP BY bv.gid, bv.geom )
```

/* Annexe_7bis indice_pression_tbv.sql */

```
res on (res.gid=bv.gid);insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result
(gid_tbv, id_param, valeur) select bv.gid, 1344, coalesce(res.res,0) from
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv left join ( SELECT bv.gid ,
(((sum(st_area(st_intersection(bv.geom_ce_buffer_10, ocs.geom)))) /
st_area(bv.geom_ce_buffer_10))*100)::numeric(5,2) as res FROM
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv ,
m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs WHERE
st_intersects(bv.geom_ce_buffer_10, ocs.geom) AND ((ocs.data_theme ilike 'Transport' AND
ocs.source_data#>>'{nature}'::TEXT[])not in('Chemin' , 'Route empierrée')) OR ocs.data_theme
in ('bâti dense', 'bâti diffus', 'zones ind et com', 'surface route') OR ocs.data_theme
ilike 'Urbain') GROUP BY bv.gid, bv.geom_ce_buffer_10 ) res on (res.gid=bv.gid);
```

```
/* Annexe_8 construction_zones_urbaines.sql */
```

```
/*
-- Etape 1 : buffers de 50m sur les différentes couches de bati et union dans une table unique
*/
drop table if exists _agent_acoudart.__bati_buff_bvsn_step01 ;
create table _agent_acoudart.__bati_buff_bvsn_step01 as

with
w_prepare as (
select bi.geom, st_multi(st_buffer(bi.geom,50,'join=mitre
mitre_limit=2'))::geometry(multipolygon, 2154) as geom_b50
from xx_99_50_10_import_wfs.t_ign_bdtopo_bati_indifferencie bi

union all
select bindus.geom, st_multi(st_buffer(bindus.geom,50,'join=mitre
mitre_limit=2'))::geometry(multipolygon, 2154) as geom_b50
from xx_99_50_10_import_wfs.t_ign_bdtopo_bati_industriel bindus

union all
select br.geom, st_multi(st_buffer(br.geom,50,'join=mitre
mitre_limit=2'))::geometry(multipolygon, 2154) as geom_b50
from xx_99_50_10_import_wfs.t_ign_bdtopo_bati_remarquable br

union all
select g.geom, st_multi(st_buffer(g.geom,50,'join=mitre
mitre_limit=2'))::geometry(multipolygon, 2154) as geom_b50
from xx_99_50_10_import_wfs.t_ign_bdtopo_gare g

union all
select sur_rte.geom, st_multi(st_buffer(sur_rte.geom,50,'join=mitre
mitre_limit=2'))::geometry(multipolygon, 2154) as geom_b50
from xx_99_50_10_import_wfs.t_ign_bdtopo_surface_route sur_rte

)
select row_number()over()::bigint as gid,*
from w_prepare
;

DROP INDEX if exists _agent_acoudart.sidx__bati_buff_bvsn_geom_b50;

CREATE INDEX sidx__bati_buff_bvsn_geom_b50
ON _agent_acoudart.__bati_buff_bvsn_step01
USING gist
(geom_b50);

-- Etape 2 : fusion des buffers qui se touchent

drop table if exists _agent_acoudart.__bati_buff_bvsn_step02 ;
create table _agent_acoudart.__bati_buff_bvsn_step02 as
with
w_prepare as (
select (st_dump(st_collectionextract(st_union(geom_b50),3))).geom::geometry(polygon,2154) as
geom
from _agent_acoudart.__bati_buff_bvsn_step01 bb
)
select row_number()over()::bigint as gid , * from w_prepare
;

-- Etape 3 : erosion du buffer de -50m

drop table if exists _agent_acoudart.__bati_buff_bvsn ;
create table _agent_acoudart.__bati_buff_bvsn as
with
w_prepare as (
select st_multi(st_collectionextract(st_buffer(bb.geom,-50,'join=mitre
mitre_limit=1'),3))::geometry(multipolygon,2154) as geom
from _agent_acoudart.__bati_buff_bvsn_step02 bb
)
select row_number()over()::bigint as gid , * from w_prepare
;
```

```
/* Annexe_9 construction_bd_ocs_local_union_all_v03.sql */
```

```
/*
attention pour eviter des erreurs de topology et la rapidité de création, cette couche
n'est calculée que sur un buffer de 2000 autour du bvsn
*/
set work_mem = '150000';

drop table if exists m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01;
create table m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01

(
gid serial primary key
, data_theme character varying(50)
, data_subtheme character varying(50)
, data_utility character varying(50) []
, source_schema character varying(255)
, source_table character varying(255)
, source_id character varying(255)
, source_data json
, data_create_query text
, data_insert_date date default now()
, data_update_date date default now()
, geom geometry(Multipolygon,2154)
, source_geom geometry(Multipolygon,2154)
)
;
CREATE INDEX sidx_t_ocs_epsn_v01_geom
ON m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01
USING gist
(geom);

CREATE INDEX sidx_t_ocs_epsn_v01_source_geom
ON m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01
USING gist
(source_geom);
/*
Import des données route topo
*/

INSERT INTO m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 (
    data_theme
    , data_subtheme
    , data_utility
    , source_schema
    , source_table
    , source_id
    , source_data
    , data_create_query
    , data_update_date
    , source_geom)

with
w_routes as(
select *
, (case when nature not in ('Chemin', 'Route empierrée') AND largeur = 0 then 5
when nature in ('Chemin', 'Route empierrée', 'Escalier') AND largeur = 0 then 2
else (largeur)::numeric end) as w_largeur
from xx_99_50_10_import_wfs.t_ign_bdtopo_routes
where nature not in ('Bac piéton')
AND st_intersects(geom, (select st_buffer(geom,2000) from
r020_territoire_hydrographie.t_bvsn_actu_carthage))
)

select
'Transport'
, null
, null
, 'xx_99_50_10_import_wfs'
, 't_ign_bdtopo_routes'
, id
```

```
/* Annexe_9 construction_bd_ocs_local_union_all_v03.sql */
```

```
, to_json(a) as source_data
, current_query()
, null
,
ST_Multi(st_Collectionextract(st_makevalid(ST_RemoveRepeatedPoints(st_snaptogrid(st_buffer(geom, (w_largeur)::numeric, 'quad_segs=2'), 0.01))), 3))

from w_routes a
;

/*
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
-- geom =NULL WHERE source_table = 't_ign_bdtopo_routes';
*/
update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
geom = (SELECT
case when st_intersects(t_up.source_geom, st_union(t_in.source_geom)) THEN
st_multi(st_Collectionextract(st_difference(t_up.source_geom,
st_union(t_in.source_geom)), 3))
ELSE st_multi(t_up.source_geom)
END :: geometry
from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_in
WHERE ST_intersects(t_up.source_geom, t_in.source_geom) and t_up.gid > t_in.gid)
where geom is null;

/*
Import des données voie_ferrée topo
*/

INSERT INTO m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 (
data_theme
, data_subtheme
, data_utility
, source_schema
, source_table
, source_id
, source_data
, data_create_query
, data_update_date
, source_geom)

select
'Transport'
, null
, null
, 'xx_99_50_10_import_wfs'
, 't_ign_bdtopo_troncon_voie_ferree'
, id
, to_json(a) as source_data
, current_query()
, null
,
,
ST_Multi(st_Collectionextract(st_makevalid(ST_RemoveRepeatedPoints(st_snaptogrid(st_buffer(geom, 15, 'quad_segs=2'), 0.01))), 3))

from xx_99_50_10_import_wfs.t_ign_bdtopo_troncon_voie_ferree a
WHERE st_intersects(geom, (select st_buffer(geom, 2000) from
r020_territoire_hydrographie.t_bvsn_actu_carthage))
;

/*
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
-- geom =NULL WHERE source_table = 't_ign_bdtopo_troncon_voie_ferree';
*/
update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
```

```
/* Annexe_9 construction_bd_ocs_local_union_all_v03.sql */
```

```
geom = (SELECT
  case when st_intersects(t_up.source_geom, st_union(t_in.source_geom)) THEN
    st_multi(st_collectionextract(st_difference(t_up.source_geom,
    st_union(t_in.source_geom)),3))
  ELSE st_multi(t_up.source_geom)
  END :: geometry
from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_in
WHERE ST_intersects(t_up.source_geom, t_in.source_geom) and t_up.gid > t_in.gid)
where geom is null;
```

```
/*
Import des données surface en eau topo
*/
```

```
INSERT INTO m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01(
  data_theme
  , data_subtheme
  , data_utility
  , source_schema
  , source_table
  , source_id
  , source_data
  , data_create_query
  , data_update_date
  , source_geom)
```

```
select
'Surface en eau'
, null
, null
, 'xx_99_50_10_import_wfs'
, 't_ign_bdtopo_surface_eau'
, id
, to_json(a) as source_data
, current_query()
, null
,
ST_Multi(st_Collectionextract(st_makevalid(ST_RemoveRepeatedPoints(st_snaptogrid(geom,0.01))),
3))
```

```
from xx_99_50_10_import_wfs.t_ign_bdtopo_surface_eau a
WHERE st_intersects(geom, (select st_buffer(geom,2000) from
r020_territoire_hydrographie.t_bvsn_actu_carthage))
;
```

```
/*
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
-- geom =NULL WHERE source_table = 't_ign_bdtopo_surface_eau';
*/
```

```
update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
geom = (SELECT
  case when st_intersects(t_up.source_geom, st_union(t_in.source_geom)) THEN
    st_multi(st_collectionextract(st_difference(t_up.source_geom,
    st_union(t_in.source_geom)),3))
  ELSE st_multi(t_up.source_geom)
  END :: geometry
from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_in
WHERE ST_intersects(t_up.source_geom, t_in.source_geom) and t_up.gid > t_in.gid)
where geom is null;
```

```
/*
Import des données boisement topo
*/
```

```
INSERT INTO m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01(
```

```
/* Annexe_9 construction_bd_ocs_local_union_all_v03.sql */
```

```
data_theme  
, data_subtheme  
, data_utility  
, source_schema  
, source_table  
, source_id  
, source_data  
, data_create_query  
, data_update_date  
, source_geom)
```

```
select
```

```
'Boisement'  
, null  
, null  
, 'xx_99_50_10_import_wfs'  
, 't_ign_bdforet_v2'  
, id  
, to_json(a) as source_data  
, current_query()  
, null  
,  
ST_Multi(st_Collectionextract(st_makevalid(ST_RemoveRepeatedPoints(st_snaptogrid(geom,0.01))),  
3))
```

```
from xx_99_50_10_import_wfs.t_ign_bdforet_v2 a  
WHERE st_intersects(geom, (select st_buffer(geom,2000) from  
r020_territoire_hydrographie.t_bvsn_actu_carthage))  
;
```

```
/*
```

```
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set  
-- geom =NULL WHERE source_table = 't_ign_bdforet_v2';  
*/
```

```
update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set  
geom = (SELECT
```

```
case when st_intersects(t_up.source_geom, st_union(t_in.source_geom)) THEN  
st_multi(st_collectionextract(st_difference(t_up.source_geom,  
st_union(t_in.source_geom)),3))  
ELSE st_multi(t_up.source_geom)  
END :: geometry
```

```
from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_in  
WHERE ST_intersects(t_up.source_geom, t_in.source_geom) and t_up.gid > t_in.gid)  
where geom is null;
```

```
/*
```

```
Import des données RPG
```

```
--- Attention réalisation d'un ST_buffer negatif et positif lors de l'injection de la  
source_geom
```

```
*/
```

```
set work_mem = '150000';
```

```
INSERT INTO m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 (  
data_theme  
, data_subtheme  
, data_utility  
, source_schema  
, source_table  
, source_id  
, source_data  
, data_create_query  
, data_update_date  
, source_geom)
```

```
select
```

```
'Agriculture'  
, null  
, null  
, 'r080_agriculture'
```

```
/* Annexe_9 construction_bd_ocs_local_union_all_v03.sql */
```

```
, 't_rpg_2016_parcelles_graphiques'  
, id_parcel  
, to_json(a) as source_data  
, current_query()  
, null  
,  
st_multi(st_collectionextract(st_makevalid(st_buffer(st_buffer(st_buffer(st_buffer(geom,-0.01,  
'join=mitre mitre_limit=5.0')),0.01,'join=mitre mitre_limit=5.0')),0.01,'join=mitre  
mitre_limit=5.0')),,-0.01,'join=mitre mitre_limit=5.0')),3))  
  
from r080_agriculture.t_rpg_2016_parcelles_graphiques a  
WHERE st_intersects(geom, (select st_buffer(geom,2000) from  
r020_territoire_hydrographie.t_bvsn_actu_carthage))  
;  
update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set  
source_geom  
=ST_Multi(st_collectionextract(st_makevalid(ST_RemoveRepeatedPoints(st_snaptogrid(source_geom,  
0.01))),3)) where source_table = 't_rpg_2016_parcelles_graphiques';  
  
/*  
---delete from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 WHERE source_table =  
't_rpg_2016_parcelles_graphiques'  
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set  
-- geom =NULL WHERE source_table = 't_rpg_2016_parcelles_graphiques';  
*/  
  
set work_mem = '150000';  
update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set  
geom = (SELECT  
case when st_intersects(t_up.source_geom, st_union(t_in.source_geom)) THEN  
st_multi(st_collectionextract(st_difference(t_up.source_geom,  
st_union(t_in.source_geom)),3))  
ELSE st_multi(t_up.source_geom)  
END :: geometry  
from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_in  
WHERE ST_intersects(t_up.source_geom, t_in.source_geom) and t_up.gid > t_in.gid)  
where geom is null;  
  
/*  
Import des données Urbain permeable  
*/  
set work_mem = '150000';  
INSERT INTO m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01(  
data_theme  
, data_subtheme  
, data_utility  
, source_schema  
, source_table  
, source_id  
, source_data  
, data_create_query  
, data_update_date  
, source_geom)  
  
select  
'Urbain'  
, 'Permeable'  
, null  
, '_agent_acoudart'  
, '__bati_buff_perm_bvsn_step03'  
, gid  
, to_json(a) as source_data  
, current_query()  
, null  
,  
ST_Multi(st_collectionextract(st_makevalid(ST_RemoveRepeatedPoints(st_snaptogrid(geom,0.01))),  
3))  
  
from _agent_acoudart.__bati_buff_perm_bvsn_step03 a  
WHERE st_intersects(geom, (select st_buffer(geom,2000) from
```

```
/* Annexe_9 construction_bd_ocs_local_union_all_v03.sql */
```

```
r020_territoire_hydrographie.t_bvsn_actu_carthage))
```

```
;
```

```
/*
```

```
-- delete from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 WHERE source_table =  
'__bati_buff_perm_bvsn_step03'
```

```
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
```

```
-- geom =NULL WHERE source_table = '__bati_buff_perm_bvsn_step03';
```

```
*/
```

```
update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
```

```
geom = (SELECT
```

```
case when st_intersects(t_up.source_geom, st_union(t_in.source_geom)) THEN
```

```
st_multi(st_collectionextract(st_difference(t_up.source_geom,
```

```
st_union(t_in.source_geom)),3))
```

```
ELSE st_multi(t_up.source_geom)
```

```
END :: geometry
```

```
from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_in
```

```
WHERE ST_intersects(t_up.source_geom, t_in.source_geom) and t_up.gid > t_in.gid)
```

```
where geom is null;
```

```
/*
```

```
Import des données Urbain
```

```
*/
```

```
set work_mem = '150000';
```

```
INSERT INTO m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 (
```

```
data_theme
```

```
, data_subtheme
```

```
, data_utility
```

```
, source_schema
```

```
, source_table
```

```
, source_id
```

```
, source_data
```

```
, data_create_query
```

```
, data_update_date
```

```
, source_geom)
```

```
select
```

```
'Urbain'
```

```
, null
```

```
, null
```

```
, '_agent_acoudart'
```

```
, '__bd_urb_v02'
```

```
, gid
```

```
, to_json(a) as source_data
```

```
, current_query()
```

```
, null
```

```
,
```

```
ST_Multi(st_Collectionextract(st_makevalid(ST_RemoveRepeatedPoints(st_snaptogrid(geom,0.01))),
```

```
3))
```

```
from _agent_acoudart.__bd_urb_v02 a
```

```
WHERE st_intersects(geom, (select st_buffer(geom,2000) from
```

```
r020_territoire_hydrographie.t_bvsn_actu_carthage))
```

```
;
```

```
/*
```

```
-- delete from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 WHERE source_table =
```

```
'__bd_urb_v02'
```

```
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
```

```
-- geom =NULL WHERE source_table = '__bd_urb_v02';
```

```
*/
```

```
update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
```

```
geom = (SELECT
```

```
case when st_intersects(t_up.source_geom, st_union(t_in.source_geom)) THEN
```

```
st_multi(st_collectionextract(st_difference(t_up.source_geom,
```

```
st_union(t_in.source_geom)),3))
```

```
ELSE st_multi(t_up.source_geom)
```

```
/* Annexe_9 construction_bd_ocs_local_union_all_v03.sql */
```

```
END :: geometry
from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_in
WHERE ST_intersects(t_up.source_geom, t_in.source_geom) and t_up.gid > t_in.gid)
where geom is null;
```

```
/*
Import de l'OCS Théia
*/
```

```
set work_mem = '150000';
INSERT INTO m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01(
    data_theme
    , data_subtheme
    , data_utility
    , source_schema
    , source_table
    , source_id
    , source_data
    , data_create_query
    , data_update_date
    , source_geom)
```

```
with
-- grille sur les 4 departements
grille as (
    select * from xx_99_utils.f_geo_makegrid2d_table(
        --- 4 depts
        -- (select st_union(geom) from
        r010_administratif_reglementaire.t_departements_2xxx_bd_topo
        --         where "si_admin_metad_version_date" =(select
        max(si_admin_metad_version_date) from
        r010_administratif_reglementaire.t_departements_2xxx_bd_topo)
        --         AND "si_admin_datageom_is_in_bvsn" IS TRUE
        --     )
        (select st_buffer(geom,2000) from r020_territoire_hydrographie.t_bvsn_actu_carthage)
        ,1000,2154)
)
-- decoupe theia avec grille
, decoupe_par_grille as (
    select
        ocs.*
        , ST_intersection(ocs.geom, grille.geom) as geom_intersection_grille
    from
        r020_territoire_occupation_du_sol.t_ocs_ces_2017 ocs
        , grille
    WHERE ST_intersects (grille.geom, ocs.geom)
)
)
```

```
select
CASE a.classe
WHEN 11 THEN 'culture été'
WHEN 12 THEN 'culture hiver'
WHEN 31 THEN 'forêt feuillus'
WHEN 32 THEN 'forêt conifères'
WHEN 34 THEN 'pelouses'
WHEN 36 THEN 'lande ligneuse'
WHEN 41 THEN 'bâti dense'
WHEN 42 THEN 'bâti diffus'
WHEN 43 THEN 'zones ind et com'
WHEN 44 THEN 'surface route'
WHEN 45 THEN 'surfaces minérales'
WHEN 46 THEN 'plages et dunes'
WHEN 51 THEN 'eau'
WHEN 53 THEN 'glaciers ou neige'
WHEN 211 THEN 'prairie'
WHEN 221 THEN 'verger'
```

```
/* Annexe_9 construction_bd_ocs_local_union_all_v03.sql */
```

```
        WHEN 222 THEN 'vigne'
        ELSE '!!!! OCS THEIA non classe !!!!'
    END)::text
, null
, null
, 'r020_territoire_occupation_du_sol'
, 't_ocs_ces_2017'
, ogc_fid
, to_json(a) as source_data
, current_query()
, null
,
st_multi(st_collectionextract(st_makevalid(a.geom_intersection_grille),3))::geometry(multipolygon,2154)
from decoupe_par_grille a
;
```

```
/*
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01
-- set data_theme = (SELECT CASE data_theme::int
--     WHEN 11 THEN 'culture été'
--     WHEN 12 THEN 'culture hiver'
--     WHEN 31 THEN 'forêt feuillus'
--     WHEN 32 THEN 'forêt conifères'
--     WHEN 34 THEN 'pelouses'
--     WHEN 36 THEN 'lande ligneuse'
--     WHEN 41 THEN 'bâti dense'
--     WHEN 42 THEN 'bâti diffus'
--     WHEN 43 THEN 'zones ind et com'
--     WHEN 44 THEN 'surface route'
--     WHEN 45 THEN 'surfaces minérales'
--     WHEN 46 THEN 'plages et dunes'
--     WHEN 51 THEN 'eau'
--     WHEN 53 THEN 'glaciers ou neige'
--     WHEN 211 THEN 'prairie'
--     WHEN 221 THEN 'verger'
--     WHEN 222 THEN 'vigne'
--     ELSE '!!!! OCS THEIA non classe !!!!'
-- END)::text
-- )
-- WHERE source_table = 't_ocs_ces_2017'
```

```
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set geom =NULL WHERE
source_table = 't_ocs_ces_2017';
```

```
update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
geom = st_collectionextract(st_makevalid(geom),3)
WHERE geom is not null;
```

```
DELETE FROM m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01
WHERE source_table = 't_ocs_ces_2017';
```

```
*/
/*
```

```
-- attention pour eviter des erreurs de topology cette couche n'est calculée que sur un
buffer de 2000 autour du bvsn
/*set work_mem = '150000';
update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
geom = (
    SELECT
        case when st_intersects(t_up.source_geom,
            st_collectionextract(st_makevalid(st_union(st_makevalid(t_in.source_geom))),3)) THEN
            st_multi(st_collectionextract(xx_99_utils.st_difference_safe(t_up.source_geom,
            st_makevalid(st_collect(st_makevalid(t_in.source_geom))),3))
                ELSE st_multi(t_up.source_geom)
            END :: geometry
        from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_in
        WHERE ST_intersects(t_up.source_geom, t_in.source_geom) and t_up.gid > t_in.gid
    )
where geom is null ;
```

```
/* Annexe_9 construction_bd_ocs_local_union_all_v03.sql */
```

```
*/
/*
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
-- geom
=ST_Multi(st_Collectionextract(st_makevalid(ST_RemoveRepeatedPoints(st_snaptogrid(source_geom,
0.01))),3)) where source_table = 't_ign_bdtopo_routes';
--
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
-- geom
=ST_Multi(st_Collectionextract(st_makevalid(ST_RemoveRepeatedPoints(st_snaptogrid(source_geom,
0.01))),3)) where source_table = 't_ign_bdtopo_routes';

-- SET postgis.backend = sfcgal;
-- set work_mem = '150000';
-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
-- geom = (
--     SELECT
--         case when st_intersects(t_up.source_geom,
st_collectionextract(st_makevalid(st_union(st_makevalid(t_in.geom))),3)) THEN
st_multi(st_collectionextract(st_difference_safe(t_up.source_geom,
st_makevalid(st_collect(st_makevalid(t_in.geom))),3))
--             ELSE st_multi(t_up.source_geom)
--         END :: geometry
--         from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_in
--         WHERE ST_intersects(t_up.source_geom, t_in.geom) and t_up.gid > t_in.gid
--     )
-- where geom is null ;

-- update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set
-- geom = (SELECT
--     case when st_intersects(t_up.source_geom, st_union(t_in.source_geom)) THEN
st_multi(st_collectionextract(st_difference(t_up.source_geom, st_union(t_in.source_geom)),3))
--     ELSE st_multi(t_up.source_geom)
--     END :: geometry
-- from m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_in
-- WHERE ST_intersects(t_up.source_geom, t_in.source_geom) and t_up.gid > t_in.gid)
-- where geom is null;

-- SET postgis.backend = sfcgal;
--update m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 t_up set geom =NULL WHERE
source_table = 't_ocs_ces_2017';
*/
```

```
create OR REPLACE function
```

```
m060_milieux_continuite_tetes_de_bv.f_decoupe_geom_source_pour_combler_ocs_geom (
_schemaname name
, _tablename name
, _pkname NAME
, _source_geomname name
, _dest_geomname NAME
)
returns text as
```

```
$BODY$
```

```
DECLARE
```

```
/*
```

```
Découpe geom_source_geom pour occuper l'espace libre dans l'ensemble de la table. (utile
pour couche de couverture de type occupation du sol)
```

```
Exemple d'appel :
```

```
SELECT m060_milieux_continuite_tetes_de_bv.f_decoupe_geom_source_pour_combler_ocs_geom (
_schemaname := 'm060_milieux_continuite_tetes_de_bv' -- nom du schema de la table
de couverture
, _tablename := 't_ocs_epsn_v01' -- nom de la table de
couverture
, _pkname:='gid' -- nom de la colonne primary
key
, _source_geomname := 'source_geom' -- nom de la colonne geom à
découper
```

```
/* Annexe_9 construction_bd_ocs_local_union_all_v03.sql */
```

```
    , _dest_geomname := 'geom'                                -- nom de la colonne geom qui
    contiendra le résultat du découpage

)

notes de version :

- V01 : first version

*/
mode_debug BOOLEAN := TRUE ; ---true; -- TRUE renvoi des notices

lst_gids record;
gid_to_up bigint;
i int;
str_sql_lst_gids text;
str_sql_lst_gids_count TEXT;
str_sql_up_geom text;
return_txt text := '';
nb_gid_to_up bigint;

BEGIN

--
-- 1 . Liste des gids à découper :
--
str_sql_lst_gids := format (
    $SQL$
    SELECT %I as gid FROM %I.%I WHERE %I IS NULL

    $SQL$
    , _pkname, _schemaname , _tablename , _dest_geomname
)
;

str_sql_lst_gids_count := format (
    $SQL$
    SELECT count(%I) as nb FROM %I.%I WHERE %I IS NULL

    $SQL$
    , _pkname, _schemaname , _tablename , _dest_geomname
)
;
Execute str_sql_lst_gids_count into nb_gid_to_up;
i:=0;
For lst_gids in EXECUTE str_sql_lst_gids LOOP
i:=i+1;
gid_to_up := lst_gids.gid;

str_sql_up_geom := format (
    $SQL$
    update %I.%I t_up set
    %I= (
        SELECT
            case when st_intersects(t_up.%I,
            st_collectionextract(st_makevalid(st_union(st_makevalid(t_in.%I))),3))
            THEN st_multi(st_collectionextract(st_difference(t_up.%I,
            st_collectionextract(st_makevalid(st_union(st_makevalid(t_in.%I))),3)),3))
            ELSE st_multi(t_up.%I)
            END :: geometry
        from %I.%I t_in
        WHERE ST_intersects(t_up.%I, t_in.%I)
            and t_up.%I > t_in.%I
    )
    where %I = %s;
    $SQL$
    , _schemaname , _tablename
    , _dest_geomname
```

```
/* Annexe_9 construction_bd_ocs_local_union_all_v03.sql */
```

```
, _source_geomname, _source_geomname
, _source_geomname, _source_geomname
, _source_geomname
, _schemaname, _tablename
, _source_geomname, _source_geomname
, _pkname, _pkname
, _pkname, gid_to_up

);
--RAISE NOTICE 'str_sql_up_geom : %', str_sql_up_geom;
RAISE NOTICE '% / % -> gid_to_up : % ', i, nb_gid_to_up, gid_to_up;
begin
  execute str_sql_up_geom;
  EXCEPTION WHEN OTHERS THEN
    str_sql_up_geom := format (
      $SQL$
      update %I.%I t_up set
      %I= (
        SELECT
          case when st_intersects(t_up.%I,
            st_collectionextract(st_makevalid(st_union(st_makevalid(ST_Buffer(t_in.%I
              , 0.00001))))),3))
            THEN st_multi(st_collectionextract(st_difference(t_up.%I,
              st_collectionextract(st_makevalid(st_union(st_makevalid(ST_Buffer(t_in
                .%I,0.00001))))),3)),3))
            ELSE st_multi(t_up.%I)
            END :: geometry
          from %I.%I t_in
          WHERE ST_intersects(t_up.%I, t_in.%I)
            and t_up.%I > t_in.%I
        )
      where %I = %s;
    $SQL$
    , _schemaname, _tablename
    , _dest_geomname
    , _source_geomname, _source_geomname
    , _source_geomname, _source_geomname
    , _source_geomname
    , _schemaname, _tablename
    , _source_geomname, _source_geomname
    , _pkname, _pkname
    , _pkname, gid_to_up

  );
begin
  execute str_sql_up_geom;
  EXCEPTION WHEN OTHERS THEN

  return_txt := return_txt || ',' || gid_to_up::text;
  END;

end ;

END LOOP;

RETURN return_txt;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
```

```
SELECT m060_milieux_continuite_tetes_de_bv.f_decoupe_geom_source_pour_combler_ocs_geom (
  _schemaname := 'm060_milieux_continuite_tetes_de_bv' -- nom du schema de la table
  de couverture
, _tablename := 't_ocs_epsn_v01' -- nom de la table de
  couverture
```

/ Annexe_9 construction_bd_ocs_local_union_all_v03.sql */*

```
, _pkname := 'gid' -- nom de la colonne primary
key
, _source_geomname := 'source_geom' -- nom de la colonne geom à
découper
, _dest_geomname := 'geom' -- nom de la colonne geom qui
contiendra le résultat du découpage

)
```

```
/* Annexe_10 mcd_tbv.sql */
```

```
/*
--
-- PostgreSQL database dump
--

-- Dumped from database version 9.6.9
-- Dumped by pg_dump version 9.6.8

-- Started on 2018-09-07 11:15:34
*/

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET client_min_messages = warning;
SET row_security = off;

SET default_tablespace = '';

SET default_with_oids = false;
/*
--
-- TOC entry 7305 (class 1259 OID 40076238)
-- Name: t_tbv_01_param_carac; Type: TABLE; Schema: m060_milieux_continuite_tetes_de_bv;
Owner: -
--
*/
CREATE SCHEMA if not exists m060_milieux_continuite_tetes_de_bv;

CREATE TABLE m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac (
    id_param integer NOT NULL,
    nom_param character varying(50),
    description_param character varying(255),
    positif boolean,
    path_process text,
    unite text,
    source_data character varying(255),
    commentaire character varying(255)
);

/*
--
-- TOC entry 7304 (class 1259 OID 40076236)
-- Name: t_tbv_01_param_carac_id_param_seq; Type: SEQUENCE; Schema:
m060_milieux_continuite_tetes_de_bv; Owner: -
--
*/
CREATE SEQUENCE m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac_id_param_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

/*
--
-- TOC entry 30550 (class 0 OID 0)
-- Dependencies: 7304
-- Name: t_tbv_01_param_carac_id_param_seq; Type: SEQUENCE OWNED BY; Schema:
m060_milieux_continuite_tetes_de_bv; Owner: -
--
*/

ALTER SEQUENCE m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac_id_param_seq OWNED
BY m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac.id_param;
```

```
/* Annexe_10 mcd_tbv.sql */
```

```
/*  
--  
-- TOC entry 7309 (class 1259 OID 40076288)  
-- Name: t_tbv_01_param_ponde; Type: TABLE; Schema: m060_milieux_continuite_tetes_de_bv;  
Owner: -  
--  
*/
```

```
CREATE TABLE m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_ponde (  
    id_ponde integer NOT NULL,  
    id_param integer,  
    valeur_param_ponde numeric,  
    data_create_query text,  
    scenar character varying  
);
```

```
/*  
--  
-- TOC entry 7308 (class 1259 OID 40076286)  
-- Name: t_tbv_01_ponde_id_ponde_seq; Type: SEQUENCE; Schema:  
m060_milieux_continuite_tetes_de_bv; Owner: -  
--  
*/
```

```
CREATE SEQUENCE m060_milieux_continuite_tetes_de_bv.t_tbv_01_ponde_id_ponde_seq  
    START WITH 1  
    INCREMENT BY 1  
    NO MINVALUE  
    NO MAXVALUE  
    CACHE 1;
```

```
/*  
--  
-- TOC entry 30551 (class 0 OID 0)  
-- Dependencies: 7308  
-- Name: t_tbv_01_ponde_id_ponde_seq; Type: SEQUENCE OWNED BY; Schema:  
m060_milieux_continuite_tetes_de_bv; Owner: -  
--  
*/
```

```
ALTER SEQUENCE m060_milieux_continuite_tetes_de_bv.t_tbv_01_ponde_id_ponde_seq OWNED BY  
m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_ponde.id_ponde;
```

```
/*  
--  
-- TOC entry 7310 (class 1259 OID 40076395)  
-- Name: t_tbv_01_scenar_ponde; Type: TABLE; Schema: m060_milieux_continuite_tetes_de_bv;  
Owner: -  
--  
*/
```

```
CREATE TABLE m060_milieux_continuite_tetes_de_bv.t_tbv_01_scenar_ponde (  
    nom_scenario character varying(255) NOT NULL,  
    description_scenario character varying(255),  
    ordre numeric(5,2),  
    table_dest text,  
    res_id_param integer,  
    filter_end_valeur_type text  
);
```

```
/*  
--  
-- TOC entry 30552 (class 0 OID 0)  
-- Dependencies: 7310  
-- Name: COLUMN t_tbv_01_scenar_ponde.res_id_param; Type: COMMENT; Schema:  
m060_milieux_continuite_tetes_de_bv; Owner: -  
--  
*/
```

```
COMMENT ON COLUMN m060_milieux_continuite_tetes_de_bv.t_tbv_01_scenar_ponde.res_id_param IS  
'id_param du résultat du scenario';
```

```
/* Annexe_10 mcd_tbv.sql */
```

```
/*
--
-- TOC entry 30553 (class 0 OID 0)
-- Dependencies: 7310
-- Name: COLUMN t_tbv_01_scenar_ponde.filter_end_valeur_type; Type: COMMENT; Schema:
m060_milieux_continuite_tetes_de_bv; Owner: -
--
*/
```

COMMENT ON COLUMN

```
m060_milieux_continuite_tetes_de_bv.t_tbv_01_scenar_ponde.filter_end_valeur_type IS 'Filtre
à appliquer en fin de chaine des valeurs type';
```

```
/*
--
-- TOC entry 7587 (class 1259 OID 40209986)
-- Name: t_tbv_01_with_result; Type: TABLE; Schema: m060_milieux_continuite_tetes_de_bv;
Owner: -
--
*/
```

```
CREATE TABLE m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (
  id_tbv_param integer NOT NULL,
  gid_tbv integer,
  id_param integer,
  valeur_type character varying(510) DEFAULT 'Brute'::character varying NOT NULL,
  valeur numeric,
  date_calcul date DEFAULT now(),
  stat_moyenne numeric,
  stat_ecart_type numeric,
  valeur_decompose text,
  stat_nb_valeur numeric,
  stat_valeur_origine_position_borne character varying(255),
  stat_valeur_origine numeric
);
```

```
/*
--
-- TOC entry 30554 (class 0 OID 0)
-- Dependencies: 7587
-- Name: COLUMN t_tbv_01_with_result.stat_nb_valeur; Type: COMMENT; Schema:
m060_milieux_continuite_tetes_de_bv; Owner: -
--
*/
```

COMMENT ON COLUMN m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result.stat_nb_valeur **IS**
'nombre de valeur utilisées pour la serie statistique';

```
/*
--
-- TOC entry 30555 (class 0 OID 0)
-- Dependencies: 7587
-- Name: COLUMN t_tbv_01_with_result.stat_valeur_origine_position_borne; Type: COMMENT;
Schema: m060_milieux_continuite_tetes_de_bv; Owner: -
--
*/
```

COMMENT ON COLUMN

```
m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result.stat_valeur_origine_position_borne
IS 'indique si la valeur se situe en dehors (inf / sup) des bornes statistiques (depend du
paramètre utilisé pour borner : soit : 3. x ecart type (99,7% des individus) ; 1.96 x ecart
type (95%) des individus ; ...)';
```

```
/*
--
-- TOC entry 30556 (class 0 OID 0)
-- Dependencies: 7587
-- Name: COLUMN t_tbv_01_with_result.stat_valeur_origine; Type: COMMENT; Schema:
m060_milieux_continuite_tetes_de_bv; Owner: -
--
```

```
/* Annexe_10 mcd_tbv.sql */
```

```
*/
```

```
COMMENT ON COLUMN
```

```
m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result.stat_valeur_origine IS 'valeur d'origine avant opérations (centrage réduction; centrage_reduction_bornage; centrage_reduction_bornage_range_0_1; ...)';
```

```
/*
```

```
--
```

```
-- TOC entry 7586 (class 1259 OID 40209984)  
-- Name: t_tbv_01_with_result_id_tbv_param_seq; Type: SEQUENCE; Schema:  
m060_milieux_continuite_tetes_de_bv; Owner: -
```

```
--
```

```
*/
```

```
CREATE SEQUENCE m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result_id_tbv_param_seq  
START WITH 1  
INCREMENT BY 1  
NO MINVALUE  
NO MAXVALUE  
CACHE 1;
```

```
/*
```

```
--
```

```
-- TOC entry 30557 (class 0 OID 0)  
-- Dependencies: 7586  
-- Name: t_tbv_01_with_result_id_tbv_param_seq; Type: SEQUENCE OWNED BY; Schema:  
m060_milieux_continuite_tetes_de_bv; Owner: -
```

```
--
```

```
*/
```

```
ALTER SEQUENCE m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result_id_tbv_param_seq  
OWNED BY m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result.id_tbv_param;
```

```
/*
```

```
--
```

```
-- TOC entry 7785 (class 1259 OID 42323730)  
-- Name: tableau_vulnerabilite; Type: TABLE; Schema: m060_milieux_continuite_tetes_de_bv;  
Owner: -
```

```
--
```

```
*/
```

```
CREATE TABLE m060_milieux_continuite_tetes_de_bv.tableau_vulnerabilite (  
id_vulnerabilite integer NOT NULL,  
min_enjeu numeric,  
max_enjeu numeric,  
min_sensi numeric,  
max_sensi numeric,  
vulne_val integer  
);
```

```
/*
```

```
--
```

```
-- TOC entry 7784 (class 1259 OID 42323728)  
-- Name: tableau_vulnerabilite_id_vulnerabilite_seq; Type: SEQUENCE; Schema:  
m060_milieux_continuite_tetes_de_bv; Owner: -
```

```
--
```

```
*/
```

```
CREATE SEQUENCE m060_milieux_continuite_tetes_de_bv.tableau_vulnerabilite_id_vulnerabilite_seq  
START WITH 1  
INCREMENT BY 1  
NO MINVALUE  
NO MAXVALUE  
CACHE 1;
```

```
/*
```

```
--
```

```
-- TOC entry 30558 (class 0 OID 0)  
-- Dependencies: 7784  
-- Name: tableau_vulnerabilite_id_vulnerabilite_seq; Type: SEQUENCE OWNED BY; Schema:
```

```
/* Annexe_10 mcd_tbv.sql */
```

```
m060_milieux_continuite_tetes_de_bv; Owner: -
```

```
--
```

```
*/
```

```
ALTER SEQUENCE
```

```
m060_milieux_continuite_tetes_de_bv.tableau_vulnerabilite_id_vulnerabilite_seq OWNED BY  
m060_milieux_continuite_tetes_de_bv.tableau_vulnerabilite.id_vulnerabilite;
```

```
/*
```

```
--
```

```
-- TOC entry 30209 (class 2604 OID 40076241)
```

```
-- Name: t_tbv_01_param_carac_id_param; Type: DEFAULT; Schema:
```

```
m060_milieux_continuite_tetes_de_bv; Owner: -
```

```
--
```

```
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac ALTER COLUMN
```

```
id_param SET DEFAULT
```

```
nextval('m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac_id_param_seq'::regclass);
```

```
/*
```

```
--
```

```
-- TOC entry 30210 (class 2604 OID 40076291)
```

```
-- Name: t_tbv_01_param_ponde_id_ponde; Type: DEFAULT; Schema:
```

```
m060_milieux_continuite_tetes_de_bv; Owner: -
```

```
--
```

```
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_ponde ALTER COLUMN
```

```
id_ponde SET DEFAULT
```

```
nextval('m060_milieux_continuite_tetes_de_bv.t_tbv_01_ponde_id_ponde_seq'::regclass);
```

```
/*
```

```
--
```

```
-- TOC entry 30211 (class 2604 OID 40209989)
```

```
-- Name: t_tbv_01_with_result_id_tbv_param; Type: DEFAULT; Schema:
```

```
m060_milieux_continuite_tetes_de_bv; Owner: -
```

```
--
```

```
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result ALTER COLUMN
```

```
id_tbv_param SET DEFAULT
```

```
nextval('m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result_id_tbv_param_seq'::regclass)
```

```
;
```

```
/*
```

```
--
```

```
-- TOC entry 30214 (class 2604 OID 42323733)
```

```
-- Name: tableau_vulnerabilite_id_vulnerabilite; Type: DEFAULT; Schema:
```

```
m060_milieux_continuite_tetes_de_bv; Owner: -
```

```
--
```

```
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.tableau_vulnerabilite ALTER COLUMN
```

```
id_vulnerabilite SET DEFAULT
```

```
nextval('m060_milieux_continuite_tetes_de_bv.tableau_vulnerabilite_id_vulnerabilite_seq'::regclass);
```

```
/*
```

```
--
```

```
-- TOC entry 30228 (class 2606 OID 42323738)
```

```
-- Name: tableau_vulnerabilite_id_vulnerabilite_pkey; Type: CONSTRAINT; Schema:
```

```
m060_milieux_continuite_tetes_de_bv; Owner: -
```

```
--
```

```
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.tableau_vulnerabilite
```

```
ADD CONSTRAINT id_vulnerabilite_pkey PRIMARY KEY (id_vulnerabilite);
```

```
/*
```

```
--
```

```
-- TOC entry 30217 (class 2606 OID 40076246)
```

```
/* Annexe_10 mcd_tbv.sql */
```

```
-- Name: t_tbv_01_param_carac t_tbv_01_param_carac_pkey; Type: CONSTRAINT; Schema:
m060_milieux_continuite_tetes_de_bv; Owner: -
--
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac
ADD CONSTRAINT t_tbv_01_param_carac_pkey PRIMARY KEY (id_param);
```

```
/*
--
-- TOC entry 30221 (class 2606 OID 40076296)
-- Name: t_tbv_01_param_ponde t_tbv_01_ponde_pkey; Type: CONSTRAINT; Schema:
m060_milieux_continuite_tetes_de_bv; Owner: -
--
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_ponde
ADD CONSTRAINT t_tbv_01_ponde_pkey PRIMARY KEY (id_ponde);
```

```
/*
--
-- TOC entry 30223 (class 2606 OID 40096613)
-- Name: t_tbv_01_scenar_ponde t_tbv_01_scenar_ponde_pkey; Type: CONSTRAINT; Schema:
m060_milieux_continuite_tetes_de_bv; Owner: -
--
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.t_tbv_01_scenar_ponde
ADD CONSTRAINT t_tbv_01_scenar_ponde_pkey PRIMARY KEY (nom_scenario);
```

```
/*
--
-- TOC entry 30226 (class 2606 OID 40209995)
-- Name: t_tbv_01_with_result t_tbv_01_with_result_pkey; Type: CONSTRAINT; Schema:
m060_milieux_continuite_tetes_de_bv; Owner: -
--
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result
ADD CONSTRAINT t_tbv_01_with_result_pkey PRIMARY KEY (id_tbv_param);
```

```
/*
--
-- TOC entry 30218 (class 1259 OID 40076437)
-- Name: fki_param; Type: INDEX; Schema: m060_milieux_continuite_tetes_de_bv; Owner: -
--
*/
```

```
CREATE INDEX fki_param ON m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_ponde USING
btree (id_param);
```

```
/*
--
-- TOC entry 30229 (class 1259 OID 42323739)
-- Name: idx_id_vulnerabilite; Type: INDEX; Schema: m060_milieux_continuite_tetes_de_bv;
Owner: -
--
*/
```

```
CREATE INDEX idx_id_vulnerabilite ON
m060_milieux_continuite_tetes_de_bv.tableau_vulnerabilite USING btree (id_vulnerabilite);
```

```
/*
--
-- TOC entry 30215 (class 1259 OID 40076247)
-- Name: idx_t_tbv_01_param_carac; Type: INDEX; Schema: m060_milieux_continuite_tetes_de_bv;
Owner: -
--
*/
```

```
CREATE INDEX idx_t_tbv_01_param_carac ON
```

```
/* Annexe_10 mcd_tbv.sql */
```

```
m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac USING btree (id_param);
```

```
/*  
--  
-- TOC entry 30219 (class 1259 OID 40076297)  
-- Name: idx_t_tbv_01_ponde; Type: INDEX; Schema: m060_milieux_continuite_tetes_de_bv;  
Owner: -  
--  
*/
```

```
CREATE INDEX idx_t_tbv_01_ponde ON m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_ponde  
USING btree (id_ponde);
```

```
/*  
--  
-- TOC entry 30224 (class 1259 OID 40209996)  
-- Name: idx_t_tbv_01_with_result; Type: INDEX; Schema: m060_milieux_continuite_tetes_de_bv;  
Owner: -  
--  
*/
```

```
CREATE INDEX idx_t_tbv_01_with_result ON  
m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result USING btree (id_tbv_param);
```

```
/*  
--  
-- TOC entry 30233 (class 2606 OID 40209997)  
-- Name: t_tbv_01_with_result_fk_gid_tbv; Type: FK CONSTRAINT; Schema:  
m060_milieux_continuite_tetes_de_bv; Owner: -  
--  
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result  
ADD CONSTRAINT fk_gid_tbv FOREIGN KEY (gid_tbv) REFERENCES  
m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract(gid) ON UPDATE CASCADE ON  
DELETE RESTRICT;
```

```
/*  
--  
-- TOC entry 30230 (class 2606 OID 40096635)  
-- Name: t_tbv_01_param_ponde_fk_param; Type: FK CONSTRAINT; Schema:  
m060_milieux_continuite_tetes_de_bv; Owner: -  
--  
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_ponde  
ADD CONSTRAINT fk_param FOREIGN KEY (id_param) REFERENCES  
m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac(id_param) ON UPDATE CASCADE ON  
DELETE RESTRICT;
```

```
/*  
--  
-- TOC entry 30232 (class 2606 OID 40210002)  
-- Name: t_tbv_01_with_result_fk_param_carac; Type: FK CONSTRAINT; Schema:  
m060_milieux_continuite_tetes_de_bv; Owner: -  
--  
*/
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result  
ADD CONSTRAINT fk_param_carac FOREIGN KEY (id_param) REFERENCES  
m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac(id_param) ON UPDATE CASCADE ON  
DELETE RESTRICT;
```

```
/*  
--  
-- TOC entry 30231 (class 2606 OID 40096630)  
-- Name: t_tbv_01_param_ponde_t_tbv_01_ponde_scenar_fkey; Type: FK CONSTRAINT; Schema:  
m060_milieux_continuite_tetes_de_bv; Owner: -  
--  
*/
```

```
/* Annexe_10 mcd_tbv.sql */
```

```
ALTER TABLE ONLY m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_ponde  
  ADD CONSTRAINT t_tbv_01_ponde_scenar_fkey FOREIGN KEY (scenar) REFERENCES  
  m060_milieux_continuite_tetes_de_bv.t_tbv_01_scenar_ponde (nom_scenario) ON UPDATE  
  CASCADE ON DELETE RESTRICT;
```

```
/*
```

```
-- Completed on 2018-09-07 11:15:46
```

```
--
```

```
-- PostgreSQL database dump complete
```

```
--
```

```
*/
```

```
/* Annexe_12bis f_polygon_densite_of_geom_table_v03_00.sql */
```

```
/*  
-- Function: xx_99_utils.f_polygon_densite_of_geom_table_v03_00(geometry, geometry,  
geometry, numeric, name, name, name, name, numeric, text, boolean)  
  
-- DROP FUNCTION xx_99_utils.f_polygon_densite_of_geom_table_v03_00(geometry, geometry,  
geometry, numeric, name, name, name, name, numeric, text, boolean);  
*/
```

```
CREATE OR REPLACE FUNCTION xx_99_utils.f_polygon_densite_of_geom_table_v03_00 (  
  _input_geom_recherche geometry,  
  _input_geom_limite geometry,  
  _input_geom_base_ratio geometry,  
  _input_geom_unite numeric,  
  _tcompare_schemaname name,  
  _tcompare_tablename name,  
  _tcompare_geom name,  
  _tcompare_geomtype name,  
  _tcompare_unite numeric,  
  _tcompare_where_clause text DEFAULT NULL::text,  
  _debug_mode boolean DEFAULT false)  
RETURNS numeric AS
```

```
$BODY$
```

```
DECLARE
```

```
/*
```

```
Chaîne de traitement permettant d'intersecter des informations d'une table à une table  
de base (référence) et calculant la densité de ces informations dans la table de base.
```

```
Exemple d'appel :
```

```
SELECT
```

```
xx_99_utils.f_polygon_densite_of_geom_table_v03_00 (  
  _input_geom_recherche :=  
    wkb_geometry -- geom dans laquelle  
    l'intersection sera réalisée  
  , _input_geom_limite :=  
    wkb_geometry -- geom limitant le  
    résultat final (aucun résultat en dehors de cette limite ne sera pris en compte)  
  , _input_geom_base_ratio :=  
    wkb_geometry -- geom utilisée pour  
    le calcul du ratio final  
  , _input_geom_unite :=  
    10000 -- Unité métrique  
    de la géométrie _input_geom_ref_ratio (10000 pour ha...)  
  , _tcompare_schemaname := 'm060_milieus_continuite_dec_agreg'  
  -- Schema de la table contenant les infos à intersecter avec la geom en input  
  , _tcompare_tablename:= 't_zh_agreg'  
  -- Table contenant les infos à intersecter avec la geom en input  
  , _tcompare_geom := 'geom'  
  -- Geom des infos à intersecter avec la geom en input  
  , _tcompare_geomtype := 'POLYGON' -- Type geom de base  
  (POLYGON, LINESTRING, POINT)  
  , _tcompare_unite := 10000 -- Unité métrique des infos  
  à intersecter(10000 pour ha...)  
  , _tcompare_where_clause := NULL -- [optionnel] : Clause Where  
  , _debug_mode :=FALSE -- [optionnel] : TRUE pour afficher la  
  requete dans la console  
  
  ) as dens_zh_ha  
  , *  
  FROM _agent_acoudart._t_test_sangueze_mnt_sinkless_basins_2et1
```

```
notes de version :
```

- V03_00 : intersection safe et union safe
- V02_00 : Ajout d'une geom surfacique de limite et base de ration : _input_geom_limite
et _input_geom_ref_ratio
- V01_02 : second version

```
*/
```

```
--      , _compare_geomtype := 'POLYGON' --  
Type geom de base (POLYGON, LINESTRING, POINT)
```

```
/* Annexe_12bis f_polygon_densite_of_geom_table_v03_00.sql */
```

```
fonction_schemaname name := 'xx_99_utils';  
rcd_return record;  
i int;  
str_sql text;  
/*
```

```
*/  
str_clause_where text;  
str_counter text;
```

```
BEGIN
```

```
/*
```

```
SET client_min_messages = ERROR;  
if _debug_mode THEN  
    SET client_min_messages = NOTICE; -- pour le debug  
END IF;
```

```
*/
```

```
--  
-- 1 .  
--
```

```
IF (_tcompare_where_clause is null or _tcompare_where_clause ilike '')  
    THEN str_clause_where := '';  
    ELSE str_clause_where := ' AND ' || _tcompare_where_clause;  
END IF ;
```

```
CASE WHEN _tcompare_geomtype = 'POLYGON'
```

```
    THEN str_counter := format (  
        $SQL$  
        ST_Area(  
            ST_CollectionExtract(  
                xx_99_utils.st_intersection_safe(  
                    ST_MakeValid(xx_99_utils.st_union_safe(%I.st_intersection_btr(%L,  
                        ST_Makevalid(%I))))  
                , %L  
            )  
            ,3)::geometry)/%s.  
        $SQL$  
        , fonction_schemaname , _input_geom_recherche , _tcompare_geom,  
        _input_geom_limite , _tcompare_unite  
    );
```

```
WHEN _tcompare_geomtype = 'LINESTRING'
```

```
    THEN str_counter := format (  
        $SQL$  
        ST_LENGTH(  
            ST_CollectionExtract(  
                xx_99_utils.st_intersection_safe(  
                    ST_MakeValid(xx_99_utils.st_union_safe(%I.st_intersection_btr(%L,  
                        ST_Makevalid(%I))))  
                , %L  
            )  
            ,2)::geometry)/%s.  
        $SQL$  
        , fonction_schemaname , _input_geom_recherche , _tcompare_geom,  
        _input_geom_limite , _tcompare_unite  
    );
```

```
ELSE str_counter := format (  
    $SQL$  
    COUNT(%L)/%s.  
    $SQL$  
    , _tcompare_geom , _tcompare_unite  
);
```

```
END CASE;
```

```
str_sql := format (  
    $SQL$
```

```
/* Annexe_12bis f_polygon_densite_of_geom_table_v03_00.sql */
```

```
WITH w_prepare AS (  
  SELECT  
    ST_AREA(%L)/%s AS superficie_obj  
  , %s::numeric AS counter  
  FROM %I.%I  
  WHERE ST_INTERSECTS(%L, %I) %s  
  )  
SELECT  
  counter/superficie_obj AS result  
FROM w_prepare  
$SQL$
```

```
,_input_geom_base_ratio, _input_geom_unite  
, str_counter  
,_tcompare_schemaname, _tcompare_tablename  
,_input_geom_recherche , _tcompare_geom , str_clause_where  
);
```

```
if _debug_mode THEN  
  RAISE NOTICE 'str_sql: %', str_sql;  
END IF;
```

```
Execute str_sql into rcd_return; -- le sql doit retourner une ligne avec un numeric qui  
s'appelle result
```

```
return rcd_return.result;
```

```
END;
```

```
$BODY$
```

```
LANGUAGE plpgsql STABLE  
COST 100;
```

```
/* Annexe_12 f_polygon_densite_of_geom_table_v02_00.sql */
```

```
/*  
-- Function: xx_99_utils.f_polygon_densite_of_geom_table_v02_00(geometry, geometry,  
geometry, numeric, name, name, name, name, numeric, text, boolean)  
  
-- DROP FUNCTION xx_99_utils.f_polygon_densite_of_geom_table_v02_00(geometry, geometry,  
geometry, numeric, name, name, name, name, numeric, text, boolean);  
*/
```

```
CREATE OR REPLACE FUNCTION xx_99_utils.f_polygon_densite_of_geom_table_v02_00 (  
  _input_geom_recherche geometry,  
  _input_geom_limite geometry,  
  _input_geom_base_ratio geometry,  
  _input_geom_unite numeric,  
  _tcompare_schemaname name,  
  _tcompare_tablename name,  
  _tcompare_geom name,  
  _tcompare_geomtype name,  
  _tcompare_unite numeric,  
  _tcompare_where_clause text DEFAULT NULL::text,  
  _debug_mode boolean DEFAULT false)  
RETURNS numeric AS
```

```
$BODY$
```

```
DECLARE
```

```
/*
```

```
Chaîne de traitement permettant d'intersecter des informations d'une table à une table  
de base (référence) et calculant la densité de ces informations dans la table de base.
```

```
Exemple d'appel :
```

```
SELECT
```

```
xx_99_utils.f_polygon_densite_of_geom_table_v02_00 (  
  _input_geom_recherche :=  
    wkb_geometry -- geom dans laquelle  
    l'intersection sera réalisée  
  , _input_geom_limite :=  
    wkb_geometry -- geom limitant le  
    résultat final (aucun résultat en dehors de cette limite ne sera pris en compte)  
  , _input_geom_base_ratio :=  
    wkb_geometry -- geom utilisée pour  
    le calcul du ratio final  
  , _input_geom_unite :=  
    10000 -- Unité métrique  
    de la géométrie _input_geom_ref_ratio (10000 pour ha...)  
  , _tcompare_schemaname := 'm060_milieus_continuite_dec_agreg'  
  -- Schema de la table contenant les infos à intersecter avec la geom en input  
  , _tcompare_tablename:= 't_zh_agreg'  
  -- Table contenant les infos à intersecter avec la geom en input  
  , _tcompare_geom := 'geom'  
  -- Geom des infos à intersecter avec la geom en input  
  , _tcompare_geomtype := 'POLYGON' -- Type geom de base  
  (POLYGON, LINESTRING, POINT)  
  , _tcompare_unite := 10000 -- Unité métrique des infos  
  à intersecter(10000 pour ha...)  
  , _tcompare_where_clause := NULL -- [optionnel] : Clause Where  
  , _debug_mode :=FALSE -- [optionnel] : TRUE pour afficher la  
  requete dans la console  
  
  ) as dens_zh_ha  
  , *  
  FROM _agent_acoudart._t_test_sangueze_mnt_sinkless_basins_2et1
```

```
notes de version :
```

```
- V02_00 : Ajout d'une geom surfacique de limite et base de ration : _input_geom_limite  
et _input_geom_ref_ratio  
- V01_02 : second version
```

```
*/
```

```
--      , _compare_geomtype := 'POLYGON' --  
Type geom de base (POLYGON, LINESTRING, POINT)
```

```
/* Annexe_12 f_polygon_densite_of_geom_table_v02_00.sql */
```

```
fonction_schemaname name := 'xx_99_utils';  
rkd_return record;  
i int;  
str_sql text;  
/*  
  
*/  
str_clause_where text;  
str_counter text;
```

```
BEGIN
```

```
/*
```

```
SET client_min_messages = ERROR;  
if _debug_mode THEN  
    SET client_min_messages = NOTICE; -- pour le debug  
END IF;
```

```
*/
```

```
--  
-- 1 .  
--
```

```
IF (_tcompare_where_clause is null or _tcompare_where_clause ilike'' )  
    THEN str_clause_where := '';  
    ELSE str_clause_where := ' AND ' || _tcompare_where_clause;  
    END IF ;
```

```
CASE WHEN _tcompare_geomtype = 'POLYGON'
```

```
    THEN str_counter := format (  
        $SQL$  
        ST_Area(  
            ST_CollectionExtract(  
                ST_Intersection(  
                    ST_MakeValid(ST_Union(%I.st_intersection_btr(%L, ST_Makevalid(%I))))  
                    , %L  
                )  
                ,3)::geometry)/%s.  
        $SQL$  
        , fonction_schemaname , _input_geom_recherche , _tcompare_geom,  
        _input_geom_limite , _tcompare_unite  
    );
```

```
    WHEN _tcompare_geomtype = 'LINESTRING'
```

```
        THEN str_counter := format (  
            $SQL$  
            ST_LENGTH(  
                ST_CollectionExtract(  
                    ST_Intersection(  
                        ST_MakeValid(ST_Union(%I.st_intersection_btr(%L, ST_Makevalid(%I))))  
                        , %L  
                    )  
                    ,2)::geometry)/%s.  
            $SQL$  
            , fonction_schemaname , _input_geom_recherche , _tcompare_geom,  
            _input_geom_limite , _tcompare_unite  
        );
```

```
    ELSE str_counter := format (  
        $SQL$  
        COUNT(%L)/%s.  
        $SQL$  
        , _tcompare_geom , _tcompare_unite  
    );
```

```
END CASE;
```

```
str_sql := format (  
    $SQL$  
    WITH w_prepare as (  
        SELECT  
            ST_AREA(%L)/%s. AS superficie_obj
```

```
/* Annexe_12 f_polygon_densite_of_geom_table_v02_00.sql */
```

```
        , %s::numeric AS counter
        FROM      %I.%I
        WHERE ST_INTERSECTS(%L, %I) %s
    )
    SELECT
    counter/superficie_obj  as result
    FROM w_prepare
$SQL$
```

```
    ,_input_geom_base_ratio, _input_geom_unite
    , str_counter
    ,_tcompare_schemaname, _tcompare_tablename
    ,_input_geom_recherche , _tcompare_geom , str_clause_where
);
```

```
if _debug_mode THEN
    RAISE NOTICE 'str_sql: %', str_sql;
END IF;
```

```
Execute str_sql into rcd_return; -- le sql doit retourner une ligne avec un numeric qui
s'appelle result
```

```
return rcd_return.result;
```

```
END;
```

```
$BODY$
```

```
LANGUAGE plpgsql STABLE
COST 100;
```

```
/* Annexe_13 st_addpoint_at_nearest_seg_on_linestring_v01_01.sql */
```

```
/*
-- Function: xx_99_utils.st_addpoint_at_nearest_seg_on_linestring_v01_01(geometry, geometry)
-- DROP FUNCTION xx_99_utils.st_addpoint_at_nearest_seg_on_linestring_v01_01(geometry,
geometry);
*/
CREATE OR REPLACE FUNCTION xx_99_utils.st_addpoint_at_nearest_seg_on_linestring_v01_01 (
    _geomlinestring geometry,
    _geompointtoadd geometry)
    RETURNS geometry AS
$BODY$
/*
Exemple d'appel :

select
  st_astext (
xx_99_utils.ST_Addpoint_at_nearest_seg_on_linestring_V01_01 (
  _geomLinestring := st_setsrid(St_geomfromtext ('LINESTRING(0 0, 1 1, 2 2)'),2154)
  , _geomPointToAdd := st_setsrid(St_geomfromtext ('POINT( 1.5 1.5)'),2154)
  )::geometry(linestring,2154)
)

*/

with nb_points as (
select
ST_NPoints(geom) as nb_pt
FROM ( select _geomLinestring::geometry as geom) AS foo
)
, lst_points as (
select generate_series(1,nb_points.nb_pt-1) as num_seg
from nb_points
)
, list_seg as (SELECT
    num_seg
    ,st_makeline(ST_pointN(_geomLinestring, num_seg),
    ST_pointN(_geomLinestring,num_seg+1)) geom_seg

    FROM lst_points
)
, seg_to_up as (
select num_seg, geom_seg
from list_seg
ORDER BY St_distance ( geom_seg, _geomPointToAdd) asc limit 1
)
, new_seg as (
Select num_seg, st_makeline(st_startpoint(geom_seg), _geomPointToAdd) as geom_seg from
seg_to_up
union all select
num_seg+0.1 , st_makeline(_geomPointToAdd, st_endpoint(geom_seg) ) as geom_seg from seg_to_up
)
, union_des_segs as (
with union_01 as (
select num_seg::numeric, geom_seg FROM list_seg where num_seg <> (select num_seg from
seg_to_up)
union all
select num_seg, geom_seg from new_seg
)
select * from union_01 order by num_seg asc

)
SELECT St_SetSRID(ST_LineMerge(st_union(geom_seg)),ST_SRID(_geomLinestring)) from
union_des_segs
```

/ Annexe_13 st_addpoint_at_nearest_seg_on_linestring_v01_01.sql */*

;

\$BODY\$

LANGUAGE sql STABLE

COST 100;

```
/* Annexe_14 st_difference_btr.sql */
```

```
/*  
-- Function: xx_99_utils.st_difference_btr(geometry, geometry, boolean)  
-- DROP FUNCTION xx_99_utils.st_difference_btr(geometry, geometry, boolean);  
*/  
CREATE OR REPLACE FUNCTION xx_99_utils.st_difference_btr(  
    _geom_a geometry,  
    _geom_b geometry,  
    _st_makevalid boolean DEFAULT false)  
    RETURNS geometry AS  
$BODY$  
/*  
Pour les gros traitements (btr):  
Optimise le calcul de différence = partie de A qui n'intersecte pas B.  
  
si la géométrie A est totalement exclue de B : retourne la geom A  
ne dispense d'une clause where limitant le nombre de combinaisons à calculer  
  
_st_makevalid permet créer une géométrie valide en sortie (le makevalid ne sera réalisé que  
lors d'un calcul d'intersection)  
cela permet de repousser la validité de la géométrie plus loin dans le cas d'un traitement  
complexe  
*/  
  
SELECT  
case when NOT ST_intersects(_geom_a, _geom_b) -- SI A et B ne se touchent pas  
    THEN _geom_a  
  
ELSE  
    case WHEN _st_makevalid THEN  
        ST_MakeValid(ST_Difference(_geom_a, _geom_b))  
    ELSE  
        ST_Difference(_geom_a, _geom_b)  
    END  
END  
  
;  
$BODY$  
LANGUAGE sql IMMUTABLE  
COST 100;
```

```
/* Annexe_15 st_difference_safe.sql */
```

```
/*
-- Function: xx_99_utils.st_difference_safe(geometry, geometry, numeric)
-- DROP FUNCTION xx_99_utils.st_difference_safe(geometry, geometry, numeric);
*/
CREATE OR REPLACE FUNCTION xx_99_utils.st_difference_safe (
    _geom_a geometry,
    _geom_b geometry,
    _buffervalue numeric DEFAULT 0.000001)
    RETURNS geometry AS
$BODY$
/*
Return St_difference escaping topology error
Return 'GEOMETRY EMPTY' if cannot correct topology by litle buffer

Exemple d'appel :

Select xx_99_utils.st_difference_safe (
    _geom_a := ST_GeomFromText('POLYGON((77.29 29.07,77.42 29.26,77.27 29.31,78.27
    29.31,79.27 29.31,77.29 29.07))')
    , _geom_b := ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27 29.31,77.29 29.07)')
    , _buffervalue :=0.1
    )

Note de version :
V01_02 : Buffer on only specific geom, return geometry type empty
V01_01 : init

Author :
AR (EPTBSN)

*/

DECLARE
geom_a_type TEXT := ST_GeometryType(_geom_a);
geom_b_type TEXT := ST_GeometryType(_geom_b);
srid_geom_a int := ST_SRID (_geom_a);

geom_a_type_without_st TEXT := right (geom_a_type,length(geom_a_type)-3);

-- list of geom type who can be buffered
array_geom_type_who_can_have_buffer TEXT[] :=
array['ST_Polygon','ST_PolygonZ','ST_PolygonZM','ST_MultiPolygon','ST_MultiPolygonZ','ST_Multi
PolygonZM'];

buffering_geom_a boolean := geom_a_type ILIKE ANY (array_geom_type_who_can_have_buffer);
buffering_geom_b boolean := geom_b_type ILIKE ANY (array_geom_type_who_can_have_buffer);
BEGIN

    RETURN ST_difference (_geom_a, _geom_b);
EXCEPTION
    WHEN OTHERS THEN
        BEGIN

            RETURN ST_difference (
                case WHEN buffering_geom_a THEN ST_Buffer(_geom_a, _buffervalue)
                ELSE _geom_a END
                , CASE WHEN buffering_geom_b THEN ST_Buffer(_geom_b, _buffervalue)
                ELSE _geom_b END
            );
        EXCEPTION
            WHEN OTHERS THEN
                RETURN ST_SETSRID(ST_GeomFromText(geom_a_type_without_st || '
                EMPTY'),srid_geom_a); --geom_a_type_without_st ||
                RAISE NOTICE 'PASS TOPOLOGY ERROR :
                ST_difference(%,%)' , _geom_a, _geom_b;

        END;
END
$BODY$
LANGUAGE plpgsql STABLE STRICT
```

```
/* Annexe_15 st_difference_safe.sql */
```

```
COST 100;
```

```
/* Annexe_16 st_intersection_btr.sql */
```

```
/*  
-- Function: xx_99_utils.st_intersection_btr(geometry, geometry, boolean)  
-- DROP FUNCTION xx_99_utils.st_intersection_btr(geometry, geometry, boolean);  
*/  
CREATE OR REPLACE FUNCTION xx_99_utils.st_intersection_btr(  
    _geom_a geometry,  
    _geom_b geometry,  
    _st_makevalid boolean DEFAULT false)  
    RETURNS geometry AS  
$BODY$  
/*
```

```
optimise le calcul d'intersection :  
si la géométrie a ou b est totalement incluse dans l'autre : retourne la geom d'origine  
ne dispense pas de la vérification st_intersects en amont
```

```
_st_makevalid permet créer une géométrie valide en sortie (le makevalid ne sera réalisé que  
lors d'un calcul d'intersection)  
cela permet de repousser la validité de la géométrie plus loin dans le cas d'un traitement  
complexe  
*/
```

```
SELECT  
case when ST_Covers(_geom_a, _geom_b) -- if no point in Geometry B is outside Geometry A  
    THEN _geom_b  
    WHEN ST_CoveredBy (_geom_a, _geom_b) -- if no point in Geometry/Geography A is  
        outside Geometry/Geography B  
    THEN _geom_a  
ELSE  
    case WHEN _st_makevalid THEN  
        ST_MakeValid(ST_intersection(_geom_a, _geom_b))  
    ELSE  
        ST_intersection(_geom_a, _geom_b)  
    END  
END
```

```
;  
$BODY$  
    LANGUAGE sql IMMUTABLE  
    COST 100;  
COMMENT ON FUNCTION xx_99_utils.st_intersection_btr(geometry, geometry, boolean) IS  
'optimise le calcul d'intersection :  
si la géométrie a ou b est totalement incluse dans l'autre : retourne la geom d'origine  
ne dispense pas de la vérification st_intersects en amont
```

```
_st_makevalid permet créer une géométrie valide en sortie (uniquement lors d'un calcul pas  
lors du retour de la géométrie d'origine)  
cela permet de repousser la validité de la géométrie plus loin dans le cas d'un traitement  
complexe  
';
```

```
/* Annexe_17 st_intersection_safe.sql */
```

```
/*  
-- Function: xx_99_utils.st_intersection_safe(geometry, geometry, numeric, boolean)  
-- DROP FUNCTION xx_99_utils.st_intersection_safe(geometry, geometry, numeric, boolean);  
*/  
CREATE OR REPLACE FUNCTION xx_99_utils.st_intersection_safe (  
    _geom_a geometry,  
    _geom_b geometry,  
    _buffervalue numeric DEFAULT 0.000001,  
    _debug_mode boolean DEFAULT false)  
RETURNS geometry AS  
$BODY$  
/*  
Return St_intersection escaping topology error
```

Exemple d'appel :

```
Select xx_99_utils.st_intersection_safe (  
    _geom_a := ST_GeomFromText('POLYGON((77.29 29.07,77.42 29.26,77.27 29.31,78.27  
    29.31,79.27 29.31,77.29 29.07))')  
    , _geom_b := ST_GeomFromText('LINESTRING(77.29 29.07,77.42 29.26,77.27 29.31,77.29 29.07)')  
    , _buffervalue :=0.1  
    )
```

Note de version :

```
V01_02 : Buffer on only specific geom, return geometry type empty  
V01_01 : init
```

```
Author :  
AR (EPTBSN)
```

```
*/
```

```
DECLARE  
geom_a_type TEXT := ST_GeometryType(_geom_a);  
geom_b_type TEXT := ST_GeometryType(_geom_b);  
srid_geom_a int := ST_SRID (_geom_a);  
  
geom_a_type_without_st TEXT := right (geom_a_type,length(geom_a_type)-3);  
  
-- list of geom type who can be buffered  
array_geom_type_who_can_have_buffer TEXT[] :=  
array['ST_Polygon','ST_PolygonZ','ST_PolygonZM','ST_MultiPolygon','ST_MultiPolygonZ','ST_Multi  
PolygonZM'];
```

```
buffering_geom_a boolean := geom_a_type ILIKE ANY (array_geom_type_who_can_have_buffer);  
buffering_geom_b boolean := geom_b_type ILIKE ANY (array_geom_type_who_can_have_buffer);
```

```
BEGIN  
    if _debug_mode THEN  
        raise notice 'geom_a_type %', geom_a_type;  
        raise notice 'geom_b_type %', geom_b_type;  
        raise notice 'array_geom_type_who_can_have_buffer: %',  
        array_geom_type_who_can_have_buffer;  
        raise notice 'buffering_geom_a %', buffering_geom_a;  
        raise notice 'buffering_geom_b %', buffering_geom_b;  
    END IF;  
RETURN ST_intersection(_geom_a, _geom_b);  
EXCEPTION  
    WHEN OTHERS THEN  
        BEGIN  
  
            RETURN ST_intersection(  
                case WHEN buffering_geom_a THEN ST_Buffer(_geom_a, _buffervalue)  
                ELSE _geom_a END  
                , CASE WHEN buffering_geom_b THEN ST_Buffer(_geom_b, _buffervalue)  
                ELSE _geom_b END  
            );  
        EXCEPTION  
            WHEN OTHERS THEN  
                RETURN ST_SETSRID(ST_GeomFromText(geom_a_type_without_st || ' ')
```

```
/* Annexe_17 st_intersection_safe.sql */
```

```
EMPTY'),srid_geom_a); --geom_a_type_without_st ||  
RAISE NOTICE 'PASS TOPOLOGY ERROR :  
ST_intersection(%,%)' , _geom_a, _geom_b;
```

```
END;
```

```
END
```

```
$BODY$
```

```
LANGUAGE plpgsql STABLE STRICT
```

```
COST 100;
```

```
/* Annexe_18 st_union_safe.sql */
```

```
/*  
-- Function: xx_99_utils.st_union_safe(geometry, numeric)  
-- DROP FUNCTION xx_99_utils.st_union_safe(geometry, numeric);  
*/
```

```
CREATE OR REPLACE FUNCTION xx_99_utils.st_union_safe(  
    _geom_a geometry,  
    _buffervalue numeric DEFAULT 0.000001)  
    RETURNS geometry AS
```

```
$BODY$
```

```
/*  
Return St_unionescaping topology error
```

Exemple d'appel :

```
select bv.gid , case when st_area(xx_99_utils.st_union_safe(st_collect(ocs.geom)))  
    IS NOT NULL THEN  
    st_collectionextract(xx_99_utils.st_union_safe(st_collect(ocs.geom),0.000001),3) ELSE  
    bv.geom END as geom  
FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract bv  
left join m060_milieux_continuite_tetes_de_bv.t_ocs_epsn_v01 ocs  
on (  
st_intersects(bv.geom, ocs.geom) AND (  
(ocs.data_theme ilike 'Boisement' AND ocs.source_data#>>'{tfv_g11}'::TEXT[] in('Forêt  
fermée feuillus', 'Forêt ouverte feuillus', 'Forêt fermée mixte', 'Forêt ouverte mixte',  
'Lande'))  
OR ocs.data_theme in ('forêt feuillus', 'lande ligneuse')  
)  
) group by bv.gid, bv.geom
```

Note de version :
V01_01 : init

Author :
AR (EPTBSN)

```
*/
```

```
DECLARE
```

```
BEGIN
```

```
--raise notice 'b value : %s', _buffervalue;  
RETURN ST_union(_geom_a);  
--raise notice 'ok ';  
EXCEPTION  
    WHEN OTHERS THEN  
        BEGIN  
            RAISE NOTICE 'TRY TO USE BUFFER :';  
            RETURN ST_union(ST_Buffer(_geom_a, _buffervalue))  
        ;  
    EXCEPTION  
        WHEN OTHERS THEN  
            RETURN NULL::geometry;  
            RAISE NOTICE 'PASS TOPOLOGY ERROR :';  
END;  
END  
$BODY$  
LANGUAGE plpgsql STABLE STRICT  
COST 100;
```

```
/* Annexe_19 f_decoupe_geom_source_pour_combler_ocs_geom.sql */
```

```
/*
-- Function:
m060_milieux_continuite_tetes_de_bv.f_decoupe_geom_source_pour_combler_ocs_geom(name, name,
name, name, name)

-- DROP FUNCTION
m060_milieux_continuite_tetes_de_bv.f_decoupe_geom_source_pour_combler_ocs_geom(name, name,
name, name, name);
*/
CREATE OR REPLACE FUNCTION
m060_milieux_continuite_tetes_de_bv.f_decoupe_geom_source_pour_combler_ocs_geom (
    _schemaname name,
    _tablename name,
    _pkname name,
    _source_geomname name,
    _dest_geomname name)
RETURNS text AS
$BODY$
    DECLARE
    /*
    Découpe geom_source_geom pour occuper l'espace libre dans l'ensemble de la table. (utile
    pour couche de couverture de type occupation du sol)

    Exemple d'appel :

    SELECT m060_milieux_continuite_tetes_de_bv.f_decoupe_geom_source_pour_combler_ocs_geom (
        _schemaname := 'm060_milieux_continuite_tetes_de_bv'      -- nom du schema de la table
        de couverture
        , _tablename := 't_ocs_epsn_v01'                          -- nom de la table de
        couverture
        , _pkname:='gid'                                          -- nom de la colonne primary
        key
        , _source_geomname := 'source_geom'                       -- nom de la colonne geom à
        découper
        , _dest_geomname := 'geom'                                -- nom de la colonne geom qui
        contiendra le résultat du découpage

    )

    notes de version :

    - V01 : first version

    */
    mode_debug BOOLEAN := TRUE ; ---true;  -- TRUE renvoi des notices

    lst_gids record;
    gid_to_up bigint;
    i int;
    str_sql_lst_gids text;
    str_sql_lst_gids_count TEXT;
    str_sql_up_geom text;
    return_txt text := '';
    nb_gid_to_up bigint;

BEGIN

--
-- 1 . Liste des gids à découper :
--
str_sql_lst_gids := format (
    $SQL$
    SELECT %I as gid FROM %I.%I WHERE %I IS NULL

    $SQL$
    , _pkname, _schemaname , _tablename , _dest_geomname
)
;
```

```

str_sql_lst_gids_count := format (
  $$SQL$
  SELECT count(%I) as nb FROM %I.%I WHERE %I IS NULL

  $$SQL$
  ,_pkname, _schemaname , _tablename , _dest_geomname
)
;
Execute str_sql_lst_gids_count into nb_gid_to_up;
i:=0;
For lst_gids in EXECUTE str_sql_lst_gids LOOP
i:=i+1;
gid_to_up := lst_gids.gid;

str_sql_up_geom := format (
  $$SQL$
  update %I.%I t_up set
  %I= (
    SELECT
      case when st_intersects(t_up.%I,
st_collectionextract(st_makevalid(st_union(st_makevalid(t_in.%I))),3))
      THEN st_multi(st_collectionextract(ST_difference(t_up.%I,
st_collectionextract(st_makevalid(st_union(st_makevalid(t_in.%I))),3)),3))
      ELSE st_multi(t_up.%I)
      END :: geometry
    from %I.%I t_in
    WHERE ST_intersects(t_up.%I, t_in.%I)
      and t_up.%I > t_in.%I
  )
  where %I = %s;
  $$SQL$
  , _schemaname , _tablename
  , _dest_geomname
  , _source_geomname , _source_geomname
  , _source_geomname , _source_geomname
  , _source_geomname
  , _schemaname , _tablename
  , _source_geomname , _source_geomname
  , _pkname, _pkname
  , _pkname, gid_to_up

);
--RAISE NOTICE 'str_sql_up_geom : %', str_sql_up_geom;
RAISE NOTICE '% / % -> gid_to_up : % ', i, nb_gid_to_up, gid_to_up;
begin
  execute str_sql_up_geom;
  EXCEPTION WHEN OTHERS THEN
    str_sql_up_geom := format (
      $$SQL$
      update %I.%I t_up set
      %I= (
        SELECT
          case when st_intersects(t_up.%I,
st_collectionextract(st_makevalid(st_union(st_makevalid(ST_Buffer(t_in.%I
, 0.00001))))),3))
          THEN st_multi(st_collectionextract(ST_difference(t_up.%I,
st_collectionextract(st_makevalid(st_union(st_makevalid(ST_Buffer(t_in
.%I,0.00001))))),3)),3))
          ELSE st_multi(t_up.%I)
          END :: geometry
        from %I.%I t_in
        WHERE ST_intersects(t_up.%I, t_in.%I)
          and t_up.%I > t_in.%I
      )
      where %I = %s;
      $$SQL$
      , _schemaname , _tablename
      , _dest_geomname
      , _source_geomname , _source_geomname
      , _source_geomname , _source_geomname
    );
end;

```

```
/* Annexe_19 f_decoupe_geom_source_pour_combler_ocs_geom.sql */
```

```
        , _source_geomname
        , _schemaname , _tablename
        , _source_geomname, _source_geomname
        , _pkname, _pkname
        , _pkname, gid_to_up

    );
begin
    execute str_sql_up_geom;
    EXCEPTION WHEN OTHERS THEN

        return_txt := return_txt || ',' || gid_to_up::text;
    END;

end ;

END LOOP;

RETURN return_txt;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
```

```
/* Annexe_20 f_geo_linestring_find_z_start_end_from_rast_v01_01.sql */
```

```
/*  
-- Function:  
m060_milieux_continuite_tetes_de_bv.f_geo_linestring_find_z_start_end_from_rast_v01_01(geometry,  
y, name, name, name, name, boolean)  
  
-- DROP FUNCTION  
m060_milieux_continuite_tetes_de_bv.f_geo_linestring_find_z_start_end_from_rast_v01_01(geometry,  
y, name, name, name, name, boolean);  
*/
```

```
CREATE OR REPLACE FUNCTION  
m060_milieux_continuite_tetes_de_bv.f_geo_linestring_find_z_start_end_from_rast_v01_01 (  
    IN _geom geometry,  
    IN _rastschemaname name,  
    IN _rasttablename name,  
    IN _rastzband name,  
    IN _rastcolname name,  
    IN _mode_debug boolean DEFAULT false)  
RETURNS TABLE(z_start numeric, z_end numeric) AS  
$BODY$
```

```
/*
```

Explication fonction:

La fonction prend une polyligne et un raster (MNT - MNE) en paramètre et renvoi le Z de départ (le plus haut) et le Z d'arrivée (Le plus bas) sur cette polyligne

EXEMPLE D'APPEL :

```
SELECT t.*,  
(m060_milieux_continuite_tetes_de_bv.f_geo_linestring_find_z_start_end_from_rast_v01_01(  
    _geom :=t.geom  
, _rastschemaname:='m020_territoire_risque_erosif_et_transfert'  
, _rasttablename:='r_mnt_init'  
, _rastzband := '1'  
, _rastcolname:='rast'  
, _mode_debug := FALSE  
)).*  
  
from _agent_acoudart.t_test_fonctions_lstring t  
WHERE gid = 1
```

Format a string. This function is similar to the C function sprintf; but only the following conversion specifications are recognized:

- %s interpolates the corresponding argument as a string
- %I escapes its argument as an SQL identifier
- %L escapes its argument as an SQL literal
- %% outputs a literal %.

A conversion can reference an explicit parameter position by preceding the conversion specifier with n\$, where n is the argument position. See also

```
*/
```

```
declare
```

```
t_sql text;
```

```
begin
```

```
t_sql := format(  
    $SQL$  
    WITH start_end as (  
        select  
        ST_StartPoint(%L::geometry) as p_start  
        , ST_EndPoint(%L::geometry) as p_end  
        )  
    SELECT  
        max(ST_value(rast1.%I, %s::int , p.p_start, TRUE))::numeric AS z_p_start  
        , max(ST_value(rast2.%I, %s::int , p.p_end, TRUE))::numeric AS z_p_end  
    FROM start_end as p
```

```
/* Annexe_20 f_geo_linestring_find_z_start_end_from_rast_v01_01.sql */
```

```
    , %I.%I rast1
    , %I.%I rast2
WHERE
ST_intersects (p.p_start, rast1.%I)
AND
    ST_intersects (p.p_end, rast2.%I)
;
$SQL$
    , _geom
    , _geom

    , _rastcolname
    , _rastzband

    , _rastcolname
    , _rastzband

    , _rastschemaname
    , _rasttablename

    , _rastschemaname
    , _rasttablename

    , _rastcolname
    , _rastcolname

);
```

```
IF _mode_debug THEN
    RAISE NOTICE 'SQL : %', t_sql;
END IF ;
```

```
RETURN QUERY EXECUTE t_sql;
end;
$BODY$
    LANGUAGE plpgsql STABLE
    COST 100
    ROWS 1000;
```

```
/* Annexe_21 f_geo_point_find_z_from_rast_v01_01.sql */
```

```
/*  
-- Function:  
m060_milieux_continuite_tetes_de_bv.f_geo_point_find_z_from_rast_v01_01(geometry, name,  
name, name, name, boolean)  
  
-- DROP FUNCTION  
m060_milieux_continuite_tetes_de_bv.f_geo_point_find_z_from_rast_v01_01(geometry, name,  
name, name, name, boolean);  
*/
```

```
CREATE OR REPLACE FUNCTION
```

```
m060_milieux_continuite_tetes_de_bv.f_geo_point_find_z_from_rast_v01_01(  
  _geom geometry,  
  _rastschemaname name,  
  _rasttablename name,  
  _rastzband name,  
  _rastcolname name,  
  _mode_debug boolean DEFAULT false)  
RETURNS SETOF numeric AS  
$BODY$
```

```
/*
```

```
Explication fonction:
```

```
Retourne le Z d'un point à partir des données d'un raster d'alti (exemple mnt rge)
```

```
Note de version :
```

```
V01_01 : première version (2018 04 30) , TODO : supprimer SETOF dans le return
```

```
EXEMPLE D'APPEL :
```

```
SELECT t.*, m060_milieux_continuite_tetes_de_bv.f_geo_point_find_z_from_rast_V01_01(  
  _geom :=st_startpoint(t.geom)  
, _rastschemaname:='m020_territoire_risque_erosif_et_transfert'  
, _rasttablename:='r_mnt_init'  
, _rastzband := '1'  
, _rastcolname:='rast'  
, _mode_debug := FALSE  
) as z_startpoint  
, m060_milieux_continuite_tetes_de_bv.f_geo_point_find_z_from_rast_V01_01(  
  _geom :=st_endpoint(t.geom)  
, _rastschemaname:='m020_territoire_risque_erosif_et_transfert'  
, _rasttablename:='r_mnt_init'  
, _rastzband := '1'  
, _rastcolname:='rast'  
, _mode_debug := FALSE  
) as z_endpoint  
  
from _agent_acoudart.t_test_fonctions_lstring t  
WHERE gid = 1
```

```
Format a string. This function is similar to the C function sprintf; but only the following  
conversion specifications are recognized:
```

```
%s interpolates the corresponding argument as a string  
%I escapes its argument as an SQL identifier  
%L escapes its argument as an SQL literal  
%% outputs a literal %.
```

```
A conversion can reference an explicit parameter position by preceding the conversion  
specifier with n$, where n is the argument position. See also
```

```
*/
```

```
declare
```

```
t_sql text;
```

```
begin
```

```
t_sql := format(  
  $SQL$  
  SELECT
```

```
/* Annexe_21 f_geo_point_find_z_from_rast_v01_01.sql */
```

```
    max(ST_value(rast.%I, %s::int , %L::geometry(point), TRUE))::numeric AS z_point
FROM   %I.%I rast
WHERE  ST_intersects(%L::geometry(point), rast.%I)
```

```

;
$SQL$
, _rastcolname
, _rastzband
, _geom

, _rastschemaname
, _rasttablename

, _geom
, _rastcolname
```

```
);
```

```
IF _mode debug THEN
    RAISE NOTICE 'SQL : %', t_sql;
END IF ;
```

```
RETURN QUERY EXECUTE t_sql;
end;
$body$
LANGUAGE plpgsql STABLE
COST 100
ROWS 1000;
```

```
/* Annexe_22 calcul_vulnerabilite.sql */
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur_type, valeur, valeur_decompose)
```

```
WITH
```

```
  w_sensi as (
    SELECT
      gid_tbv
      , valeur::numeric(5,4) val_sensi
    FROM
      m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result
    WHERE
      valeur_type ILIKE '%centr%' AND id_param = 1363
  ),
  w_enjeu as (
    SELECT
      wr.gid_tbv
      , wr.valeur::numeric(5,4) val_enjeu
      , sp.nom_scenario
      , sp.res_id_param
    FROM
      m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result wr
    LEFT JOIN
      m060_milieux_continuite_tetes_de_bv.t_tbv_01_scenar_ponde sp
      on wr.id_param = sp.res_id_param
    where
      nom_scenario ilike '$nom_enjeu' and valeur_type ILIKE '%centr%'
  )
)
```

```
SELECT
```

```
  we.gid_tbv
  , 1364
  , 'expert [vulnerabilite a l enjeu : ' ||we.nom_scenario||']'
  , tv.vulne_val
  , 'valeur enjeu : ' || val_enjeu || ' / valeur sensibilite : ' || val_sensi
FROM
  w_enjeu we
LEFT JOIN
  w_sensi ws
ON
  we.gid_tbv = ws.gid_tbv
JOIN LATERAL (SELECT max(tv.vulne_val) AS vulne_val
FROM m060_milieux_continuite_tetes_de_bv.tableau_vulnerabilite tv
WHERE (we.val_enjeu between tv.min_enjeu and tv.max_enjeu) AND (ws.val_sensi
between tv.min_sensi and tv.max_sensi)) tv ON TRUE
```

```
/* Annexe_23 centrage_reduction_bornage_0_1.sql */
```

```
delete from m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result where valeur_type ilike  
'%_range_0_1[FOR_COMP]' $sql_filter_id_param ;
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract  
(gid_tbv, id_param, valeur_type, valeur,  
stat_moyenne, stat_ecart_type, stat_nb_valeur,  
stat_valeur_origine_position_borne, stat_valeur_origine)
```

```
WITH
```

```
pc_in_borne AS
```

```
(
```

```
WITH lst_pc_in_borne as
```

```
(
```

```
SELECT
```

```
1.96 as fact_stat_ecart_type, 95.::numeric as pc_in , 'semble trop  
faible'::text as commentaire
```

```
UNION ALL
```

```
SELECT
```

```
3. as fact_stat_ecart_type, 99.7.::numeric as pc_in , 'retenu le  
14/08/2018'::text as commentaire
```

```
)
```

```
SELECT * from lst_pc_in_borne where pc_in = 99.7
```

```
)
```

```
, stats AS (
```

```
SELECT
```

```
res.id_param  
, res.valeur_type  
, count(res.valeur) as stat_nb_valeur  
, avg(res.valeur) AS stat_moyenne  
, stddev(res.valeur) as stat_ecart_type
```

```
FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract res  
group by id_param, valeur_type
```

```
)
```

```
, centrage_reduction AS (
```

```
SELECT
```

```
res.gid_tbv  
, res.id_param  
, res.valeur_type || '_centrage_reduction' as valeur_type
```

```
, (res.valeur - stats.stat_moyenne)/stats.stat_ecart_type as valeur  
, null::text as stat_valeur_origine_position_borne  
, stats.stat_nb_valeur  
, stats.stat_moyenne  
, stats.stat_ecart_type  
, res.valeur as stat_valeur_origine
```

```
FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract res  
LEFT JOIN stats on (stats.id_param = res.id_param AND stats.valeur_type =  
res.valeur_type)
```

```
WHERE res.valeur_type not like '%_range_0_1[FOR_COMP]'
```

```
id_param = /*-- A choisir (possibilité de créer une boucle pour calculer  
l'ensemble des paramètres)*/
```

```
)
```

```
, centrage_reduction_bornage as (
```

```
SELECT
```

```
res.gid_tbv  
, res.id_param  
, res.valeur_type || '_bornage' as valeur_type
```

```
, case when res.valeur <= -pc_in_borne.fact_stat_ecart_type THEN  
-pc_in_borne.fact_stat_ecart_type ELSE  
CASE WHEN res.valeur >=pc_in_borne.fact_stat_ecart_type THEN  
pc_in_borne.fact_stat_ecart_type ELSE res.valeur END  
END::numeric as valeur
```

```
, case when res.valeur <= -pc_in_borne.fact_stat_ecart_type THEN 'inf' ELSE  
CASE WHEN res.valeur >=pc_in_borne.fact_stat_ecart_type THEN 'sup' ELSE
```

```
/* Annexe_23 centrage_reduction_bornage_0_1.sql */
```

```
        'ok' END
        END::text as stat_valeur_origine_position_borne
    , res.stat_nb_valeur
    , res.stat_moyenne
    , res.stat_ecart_type
    , res.stat_valeur_origine
FROM centrage_reduction res
    , pc_in_borne
)

, centrage_reduction_bornage_range_0_1 as (
    SELECT
        res.gid_tbv
        , res.id_param
        , res.valeur_type || '%_range_0_1[FOR_COMP]' as valeur_type

        , (res.valeur +
pc_in_borne.fact_stat_ecart_type)/(2*pc_in_borne.fact_stat_ecart_type) as
valeur
        , stat_valeur_origine_position_borne
        , res.stat_nb_valeur
        , res.stat_moyenne
        , res.stat_ecart_type
        , res.stat_valeur_origine
    FROM centrage_reduction_bornage res
        , pc_in_borne
    )

, w_union as (
    SELECT * from centrage_reduction_bornage_range_0_1
    )

SELECT gid_tbv, id_param, valeur_type, valeur,
    stat_moyenne, stat_ecart_type, stat_nb_valeur,
    stat_valeur_origine_position_borne, stat_valeur_origine from w_union
```

```
/* Annexe_24 scenario_ponderation.sql */
```

```
insert into m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result (gid_tbv, id_param,
valeur_type, valeur, valeur_decompose)

With lst_scenar as (
  SELECT *
  FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_scenar_ponde scenar
  WHERE CASE WHEN '$nom_scenar' = '' THEN 1=1 ELSE scenar.nom_scenario ilike
'$nom_scenar%' END
  ORDER BY scenar.ordre ASC)
, pond_pc as
(
  SELECT pond.*
  , pond.valeur_param_ponde/sum(pond.valeur_param_ponde) over (partition by
pond.scenar) as pc_ponde
  FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_ponde pond
)
, w_prepare as (
  SELECT
    bv.gid
  , scenar.nom_scenario
  , param.nom_param
  , val.valeur
  , ponde.valeur_param_ponde
  , sum(CASE WHEN param.positif is TRUE THEN val.valeur ELSE
1.-val.valeur END * ponde.pc_ponde) over (partition by bv.gid,
scenar.nom_scenario, val.valeur_type ) as valeur_indice
  , coalesce(CASE WHEN param.positif is TRUE THEN val.valeur ELSE
1.-val.valeur END * ponde.pc_ponde,0) valeur_pond_by_gid
  , scenar.res_id_param
  FROM

m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract AS bv
LEFT JOIN lst_scenar AS scenar on (1=1)
LEFT JOIN pond_pc AS ponde on (scenar.nom_scenario = ponde.scenar)
LEFT JOIN
m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result AS val
on (bv.gid = val.gid_tbv AND val.id_param = ponde.id_param AND
val.valeur_type ilike '%' ||scenar.filter_end_valeur_type)
LEFT JOIN
m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac AS
param on (param.id_param = ponde.id_param)

ORDER BY
  bv.gid
  , val.gid_tbv
  , scenar.nom_scenario
  , valeur_pond_by_gid desc
)

SELECT
bv.gid
  , bv.res_id_param
  , '[scenar_pond : ' || bv.nom_scenario || ']' as valeur_type
  , valeur_indice
  , string_agg(
    case when bv.valeur_indice = 0. or bv.valeur_indice is
null THEN NULL::TEXT else
    bv.nom_param ||' [val_pond : ' || (100.*
bv.valeur_pond_by_gid / bv.valeur_indice)::numeric(5,2)
|| '% , val_brute : ' || bv.valeur || '%]' END, ' ** ' )
  as decompose_indice_pression

FROM
w_prepare bv

WHERE
```

```
/* Annexe_24 scenario_ponderation.sql */
```

```
valeur_pond_by_gid is not null
```

```
GROUP BY
```

```
bv.gid  
, bv.nom_scenario  
,bv.valeur_indice  
,bv.res_id_param;
```

```
/* Annexe_25 create_VM_TBV.sql */
```

```
create materialized view m060_milieux_continuite_tetes_de_bv.vm_tbv_01_with_result_geom  
as select r.* , p.nom_param,  
  
g.geom  
  
from  
  
m060_milieux_continuite_tetes_de_bv.t_tbv_01_with_result r  
  
LEFT JOIN m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac p on  
(p.id_param=r.id_param)  
  
LEFT JOIN m060_milieux_continuite_tetes_de_bv.t_tbv_01_without_caract g on (g.gid = r.gid_tbv)
```

```
/*
```

```
a jouer et executer le retour :
```

```
*/
```

```
select 'CREATE index if not exists idx_vm_tbv_01_with_result_geom' || id_param || ' ON  
m060_milieux_continuite_tetes_de_bv.vm_tbv_01_with_result_geom using gist (geom) WHERE  
id_param = ' || id_param || ';' ;
```

```
FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac
```

```
union all
```

```
select 'CREATE index if not exists idx_vm_tbv_01_with_result_geom_2_' || id_param || ' ON  
m060_milieux_continuite_tetes_de_bv.vm_tbv_01_with_result_geom using gist (geom) WHERE  
nom_param = ''' || nom_param || ''';
```

```
FROM m060_milieux_continuite_tetes_de_bv.t_tbv_01_param_carac
```

id_param	Sensibilité	Formule	Source donnée	Pondération
1321	Surface de la TBV	Surface TBV en ha = surf total	Délimitation TBV SN via MNT IGN RGE Alt 5x5 & Grass stream.bassin	1
1323	Pente moyenne de la TBV	avg(valeur de pixel) par TBV	MNT IGN RGE Alt 5x5	1
1356	Pente moyenne du chemin le plus long	différence elevation amont-aval / longueur chemin le plus long	MNT IGN RGE Alt 5x5	1
1324	Indice de compacité de Gravélius	kg = périmètre BV / (2*sqrt(pi)*surf)	MNT ; Grass	1
1357	Temps de concentration des pluies (Formule de Passini)	= 0,108 * 3^(surf BV(km2) * Long du plus long chemin eau (km) / sqrt(pente du plus long chemin de l'eau))	MNT ; Grass ; IGN ; DDT	1
1325	Densité du réseau de CE	Dens reseau CE = linéaire CE TBV / superficie TBV	Référentiel hydro DDT + IGN + Tracé théorique	1
1326	Densité du réseau de point bas	Dens reseau theorique = linéaire theorique TBV / superficie TBV	Réseau d'écoulement théorique Grass stream.extract	1
1358	Position de la TBV dans le reseau hydrographique	Apicale = connecté à l'amont d'un tronçon Strahler 3 ; Tributaire amont = Strahler 3 (non amont) et 4 ; Tributaire aval > 4	Référentiel hydro DDT + IGN + Tracé théorique	2

id_param	Qualité des Milieux	Formule	Source donnée	Pondération
1368	Taux de cours d'eau enterrés par TBV	linéaire CE enterrés / linéaire cours d'eau par TBV *100	Données référentiel hydrologique + Données construction du référentiel (t_cours_ref_ddtm_et_ign_corr_detail)	3
1327	Taux de CE impactés par les plans d'eau en barrage	linéaire CE impacter par plan d'eau / longueur CE*100	Données plan d'eau SY SMA (interne EPTB) (>2000m² & conn_terra = oui OR conn_sig = oui)	1
1329	Taux de CE avec ripisylves bande boisé ou boisement	Superficie ripisyle / superficie bande riveraine *100	Données agrégation IGN Bd topo-zone de végétation (nature = haie) + DEC haie si existant	1
1330	Densité de mares dans la bande riveraine	superficie mares / superficie bande riveraine *100	Données plan d'eau SY SMA (interne EPTB) (<2000 m² & conn_terra = 'isolé' ou 'en dérivation')	1
1359	Indice de pression dans la bande riveraine	surface occupation de sol / superficie bande riveraine *100	Données bd topo, ocs théia, rpg 2016, table composite	2
1335	Taux d'artificialisation de la bande riveraine	surface impermeable / superficie bande riveraine *100	Données composite à partir de la Bd topo	1
1333	Densité surfacique de plans d'eau dans la TBV	superficie plan d'eau / superficie bv	Données plan d'eau SY SMA (interne EPTB) (>2000m²)	1
1336	Densité de haies dans la TBV	linéaire de haies / superficie TBV	Données agrégation IGN Bd topo-zone de végétation (nature = haie) + DEC haie si existant	1
1338	Densité de mares dans la TBV	superficie mares / superficie TBV	Données plan d'eau SY SMA (interne EPTB) (<2000 m²)	1
1332	Taux d'artificialisation de la TBV	(superficie transport + urb / superficie TBV)*100	Données composite à partir de la Bd topo	1
1355	Indice de pression sur la TBV	surface occupation de sol / superficie TBV *100	Données bd topo, ocs théia, rpg 2016, table composite	1
1339	Densité de prélèvement dans la TBV	Somme volume prelev / superficie TBV	Données BNPE BRGM - Agences de l'eau	1
1328	Densité Zones Humides dans la bande riveraine	Superficie ZH / superficie bande riveraine*100	Données DEC	1
1334	Densité Zones Humides par TBV	Superficie ZH / superficie TBV *100	Données DEC	1
1337	Densité haies efficaces dans la TBV	linéaires de haies efficaces / superficie TBV	Données DEC	1
/	Pression lié aux STEP	A définir	Données Agence de l'eau + donnée interne + moulinette Norman	A définir

id_param	Qualité de l'Eau	Formule	Source donnée	Pondération
1368	Taux de cours d'eau enterrés par TBV	linéaire CE enterrés / linéaire cours d'eau par TBV *100	Données référentiel hydrologique + Données construction du référentiel (t_cours_ref_ddtm_et_ign_corr_detail)	3
1327	Taux de CE impactés par les plans d'eau en barrage	linéaire CE impacter par plan d'eau / longueur CE*100	Données plan d'eau SY SMA (interne EPTB) (>2000m² & conn_terra = oui OR conn_sig = oui)	1
1329	Taux de CE avec ripisylves bande boisé ou boisement	Superficie ripisyle / superficie bande riveraine *100	Données agrégation IGN Bd topo-zone de végétation (nature = haie) + DEC haie si existant	1
1330	Densité de mares dans la bande riveraine	superficie mares / superficie bande riveraine *100	Données plan d'eau SY SMA (interne EPTB) (<2000 m² & conn_terra = 'isolé' ou 'en dérivation')	0,5
1359	Indice de pression dans la bande riveraine	surface occupation de sol / superficie bande riveraine *100	Données bd topo, ocs théia, rpg 2016, table composite	2
1335	Taux d'artificialisation de la bande riveraine	surface impermeable / superficie bande riveraine *100	Données composite à partir de la Bd topo	1

1333	Densité surfacique de plans d'eau dans la TBV	superficie plan d'eau / superficie bv	Données plan d'eau SY SMA (interne EPTB) (>2000m ²)	1
1336	Densité de haies dans la TBV	linéaire de haies / superficie TBV	Données agrégation IGN Bd topo-zone de végétation (nature = haie) + DEC haie si existant	1
1338	Densité de mares dans la TBV	superficie mares / superficie TBV	Données plan d'eau SY SMA (interne EPTB) (<2000 m ²)	0,5
1332	Taux d'artificialisation de la TBV	(superficie transport + urb / superficie TBV)*100	Données composite à partir de la Bd topo	1
1355	Indice de pression sur la TBV	surface occupation de sol / superficie TBV *100	Données bd topo, ocs théia, rpg 2016, table composite	1
1339	Densité de prélèvement dans la TBV	Somme volume prelev / superficie TBV	Données BNPE BRGM - Agences de l'eau	1
1328	Densité Zones Humides dans la bande riveraine	Superficie ZH / superficie bande riveraine*100	Données DEC	1
1334	Densité Zones Humides par TBV	Superficie ZH / superficie TBV *100	Données DEC	1
1337	Densité haies efficaces dans la TBV	linéaires de haies efficaces / superficie TBV	Données DEC	1
/	Pression lié aux STEP	A définir	Données Agence de l'eau + donnée interne + moulinette Norman	A définir

id_param	Quantité d'Eau	Formule	Source donnée	Pondération
1368	Taux de cours d'eau enterrés par TBV	linéaire CE enterrés / linéaire cours d'eau par TBV *100	Données référentiel hydrologique + Données construction du référentiel (t_cours_ref_ddtm_et_ign_corr_detail)	3
1327	Taux de CE impactés par les plans d'eau en barrage	linéaire CE impacter par plan d'eau / longueur CE*100	Données plan d'eau SY SMA (interne EPTB) (>2000m ² & conn_terra = oui OR conn_sig = oui)	1
1329	Taux de CE avec ripisylves bande boisé ou boisement	Superficie ripisyle / superficie bande riveraine *100	Données agrégation IGN Bd topo-zone de végétation (nature = haie) + DEC haie si existant	0,5
1359	Indice de pression dans la bande riveraine	surface occupation de sol / superficie bande riveraine *100	Données bd topo, ocs théia, rpg 2016, table composite	0,5
1335	Taux d'artificialisation de la bande riveraine	surface impermeable / superficie bande riveraine *100	Donné composite à partir de la Bd topo	1
1333	Densité surfacique de plans d'eau dans la TBV	superficie plan d'eau / superficie bv	Données plan d'eau SY SMA (interne EPTB) (>2000m ²)	1
1336	Densité de haies dans la TBV	linéaire de haies / superficie TBV	Données agrégation IGN Bd topo-zone de végétation (nature = haie) + DEC haie si existant	0,5
1332	Taux d'artificialisation de la TBV	(superficie transport + urb / superficie TBV)*100	Données composite à partir de la Bd topo	1
1355	Indice de pression sur la TBV	surface occupation de sol / superficie TBV *100	Données bd topo, ocs théia, rpg 2016, table composite	0,5
1339	Densité de prélèvement dans la TBV	Somme volume prelev / superficie TBV	Données BNPE BRGM, Agences de l'eau	2
1328	Densité Zones Humides dans la bande riveraine	Superficie ZH / superficie bande riveraine*100	Données DEC	1
1334	Densité Zones Humides par TBV	Superficie ZH / superficie TBV *100	Données DEC	1
1337	Densité haies efficaces dans la TBV	linéaires de haies efficaces / superficie TBV	Données DEC	1
/	Pression lié aux STEP	A définir	Données Agence de l'eau + donnée interne + moulinette Norman	A définir

id_param	Indice de pression	Pondération
1345 / 1346	Forêts de feuillus et mixtes et landes ligneuses	0,1
1347 / 1348	Surfaces toujours en herbe (Prairies permanentes, landes et broussailles)	0,2
1349 / 1350	Prairies temporaires, fourrages, forêts de conifères et peupleraies	0,3
1342 / 1343	Surfaces urbaines perméables (espaces verts, parcs et jardins)	0,4
1351 / 1352	Cultures	0,5
1353 / 1354	Maraîchage, arboriculture et vignes	0,7
1341 / 1344	Surfaces imperméabilisées	1

t_tbv_01_without_caract	
gid	AUTO_INCREMENT
gid_graph_end	BIGINT
max_lenght_ce	DOUBLE PRECISION
strahler_after_outlet	INT
strahler_before_outlet	INT
geom	BLOB
geom_max_path	BLOB
geom_max_path_start_point	BLOB
geom_max_path_end_point	BLOB
geom_ce	BLOB
geom_ce_buffer_10	BLOB

Table des géométries des TBV
+ infos diverses

blob == geom

t_tbv_01_with_result	
id_tbv_param	AUTO_INCREMENT
gid_tbv	INT
id_param	INT
valeur_type	VARCHAR (50)
valeur	NUMERIC
date_calcul	DATE
stat_moyenne	NUMERIC
stat_ecart_type	NUMERIC
valeur_decompose	TEXT
stat_nb_valeur	NUMERIC
stat_valeur_origine_position_borne	VARCHAR (50)
stat_valeur_origine	NUMERIC

Table des résultats : les valeurs caractérisées, pondérées sont stockées ici

gid_tbv == gid (t_tbv_01_without_caract)

id_param == id_param (t_tbv_01_param_carac)

t_tbv_01_param_carac	
id_param	AUTO_INCREMENT
nom_param	VARCHAR (50)
description_param	VARCHAR (255)
positif	BOOL
path_process	TEXT
unite	TEXT
source_data	VARCHAR (255)
commentaire	VARCHAR (255)

paramètres de caractérisation :

différents indicateurs de pression, état, caractérisation.

t_tbv_01_param_ponde	
id_ponde	AUTO_INCREMENT
id_param	INT
valeur_param_ponde	NUMERIC
data_create_query	TEXT
scenar	VARCHAR (255)

Table des paramètres de pondérations :

Sont associés un param à une valeur de pondération

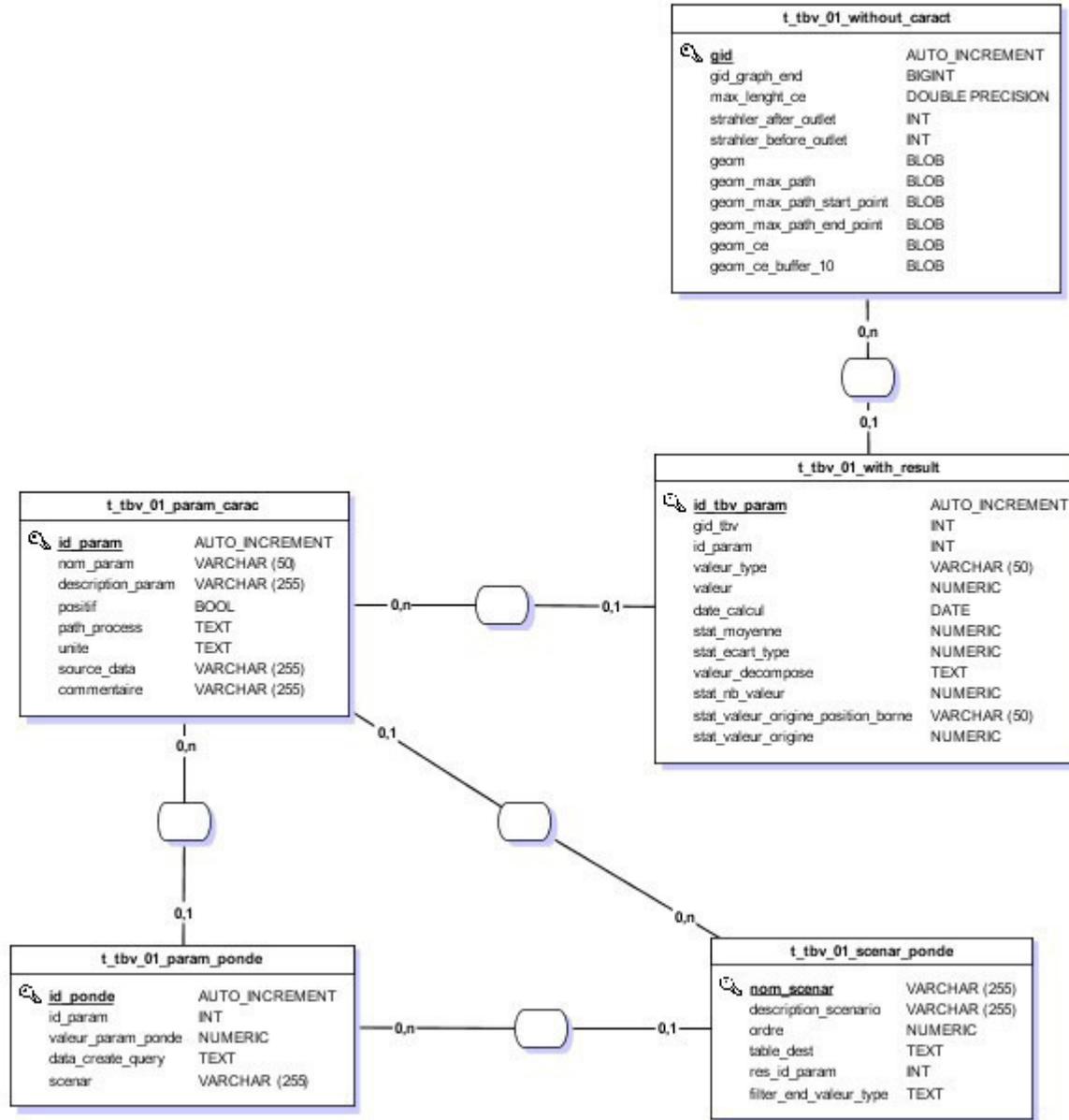
id_param == id_param (t_tbv_01_param_carac)

t_tbv_01_scenar_ponde	
nom_scenar	VARCHAR (255)
description_scenario	VARCHAR (255)
ordre	NUMERIC
table_dest	TEXT
res_id_param	INT
filter_end_valeur_type	TEXT

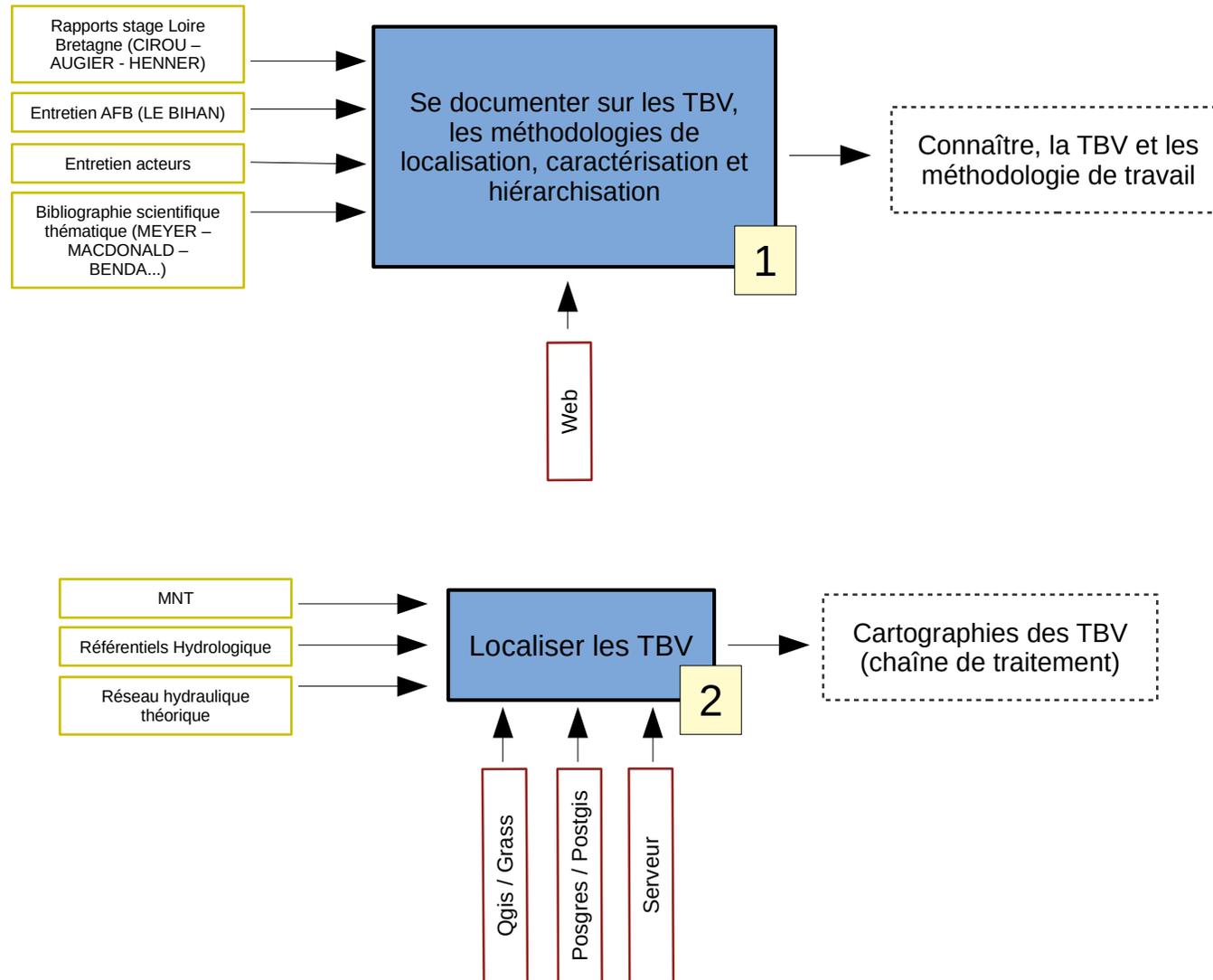
Table des scenarios : regroupe une ensemble de pondération associé à des paramètres de caractérisation

res_id_param == id_param (t_tbv_01_param_carac)

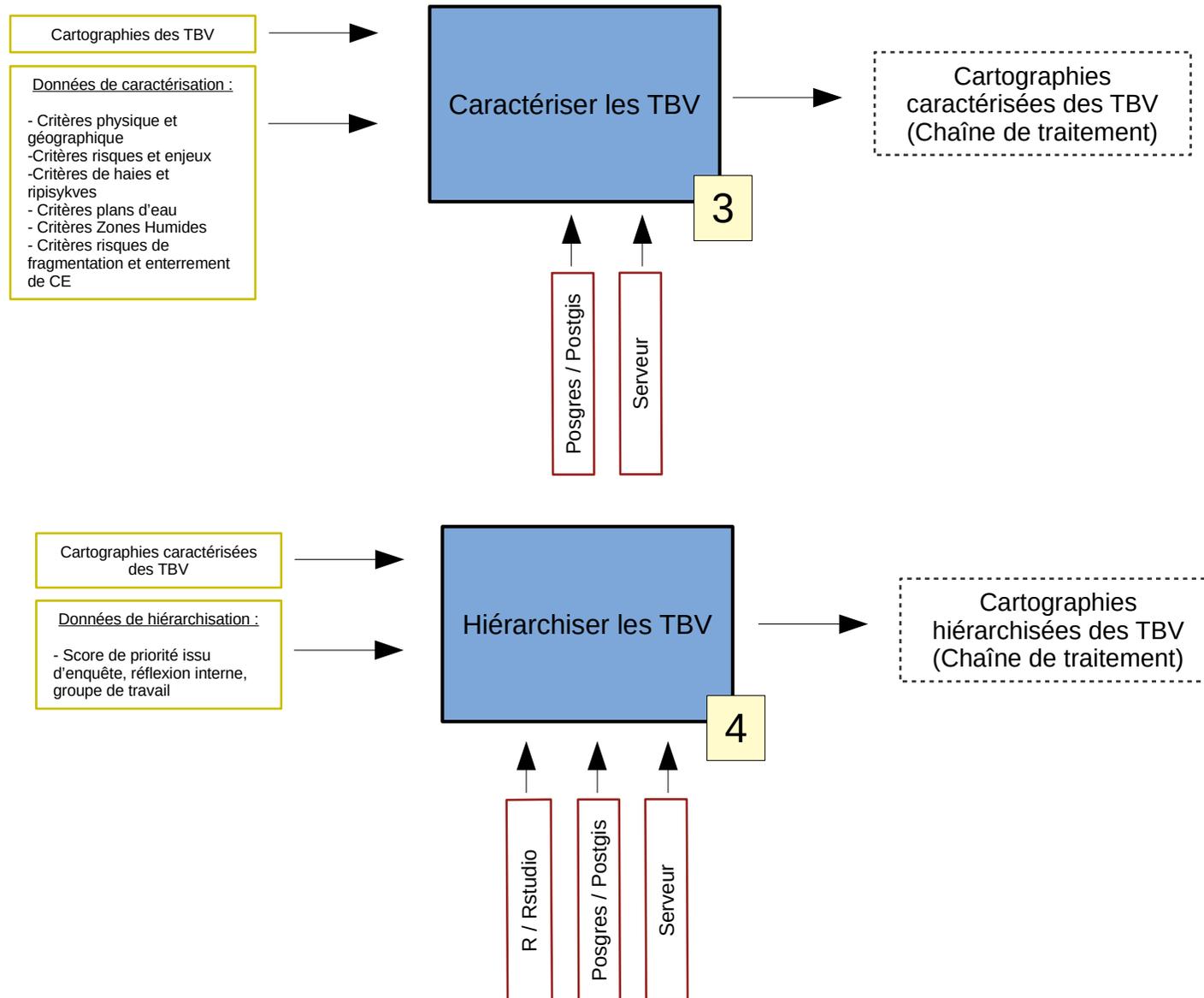
nom_scenario == scenar (t_tbv_param_ponde)



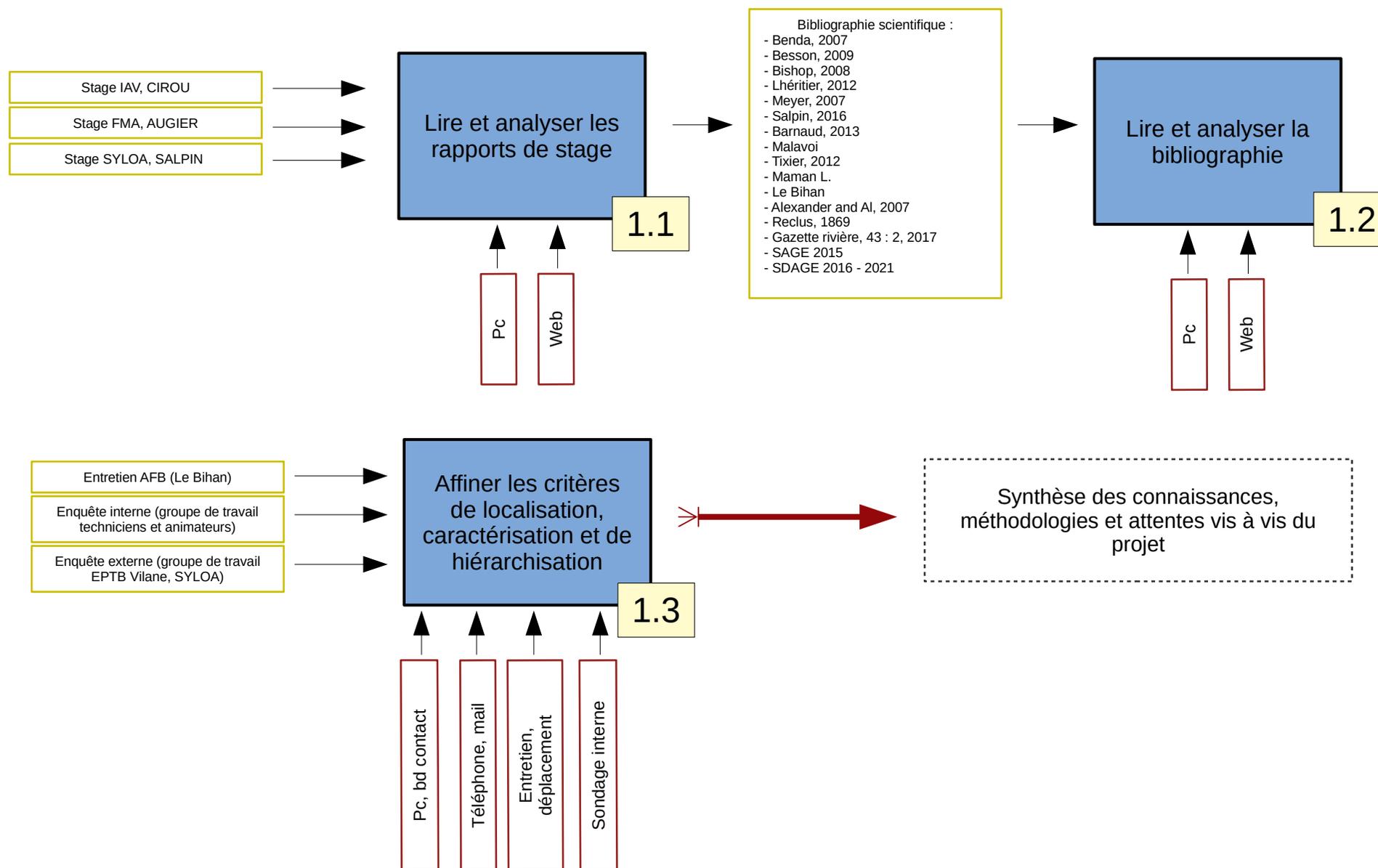
SADT du stage localisation, caractérisation et hiérarchisation des têtes de bassin versant – Niveau 0



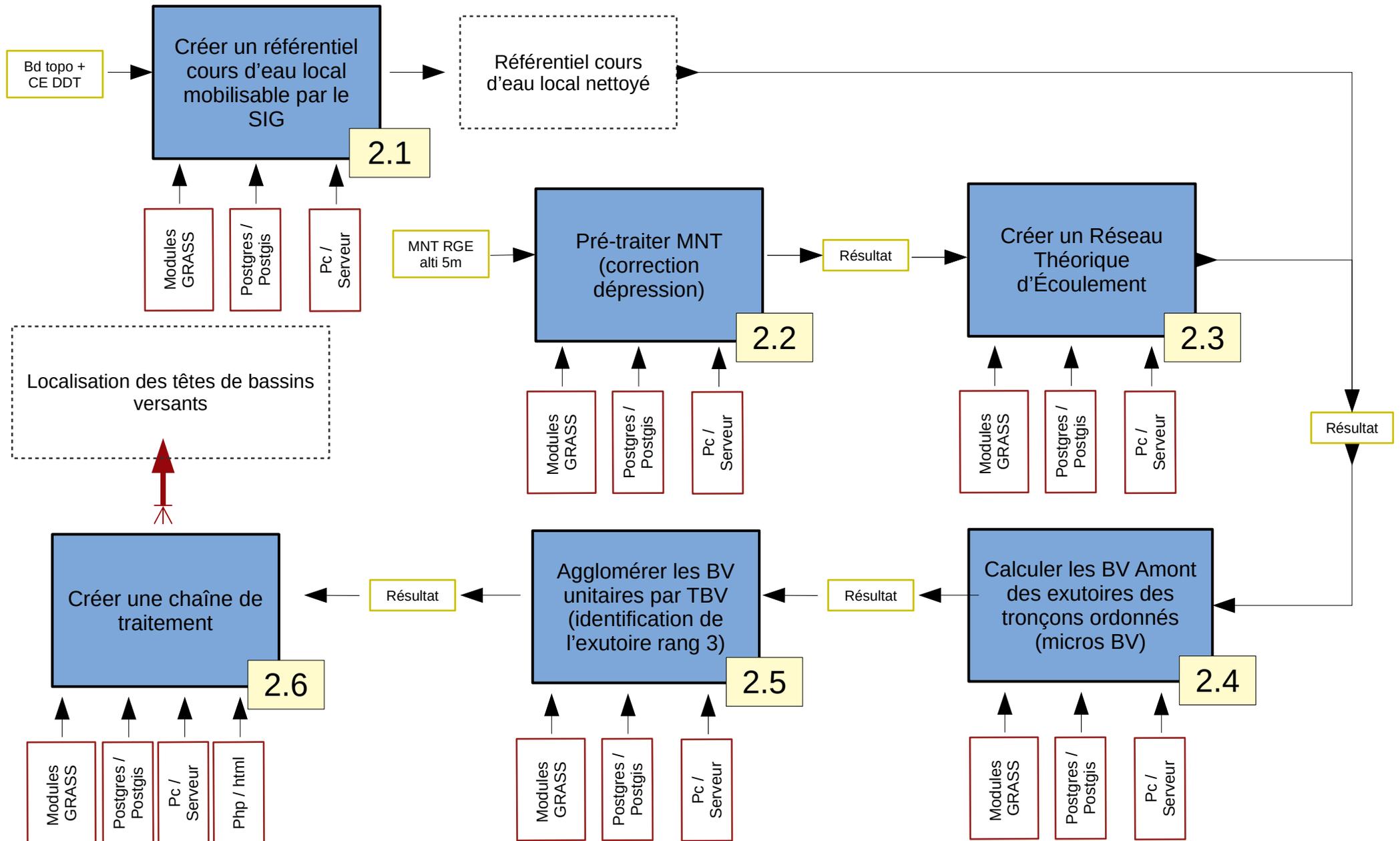
SADT du stage localisation, caractérisation et hiérarchisation des têtes de bassin versant – Niveau 0



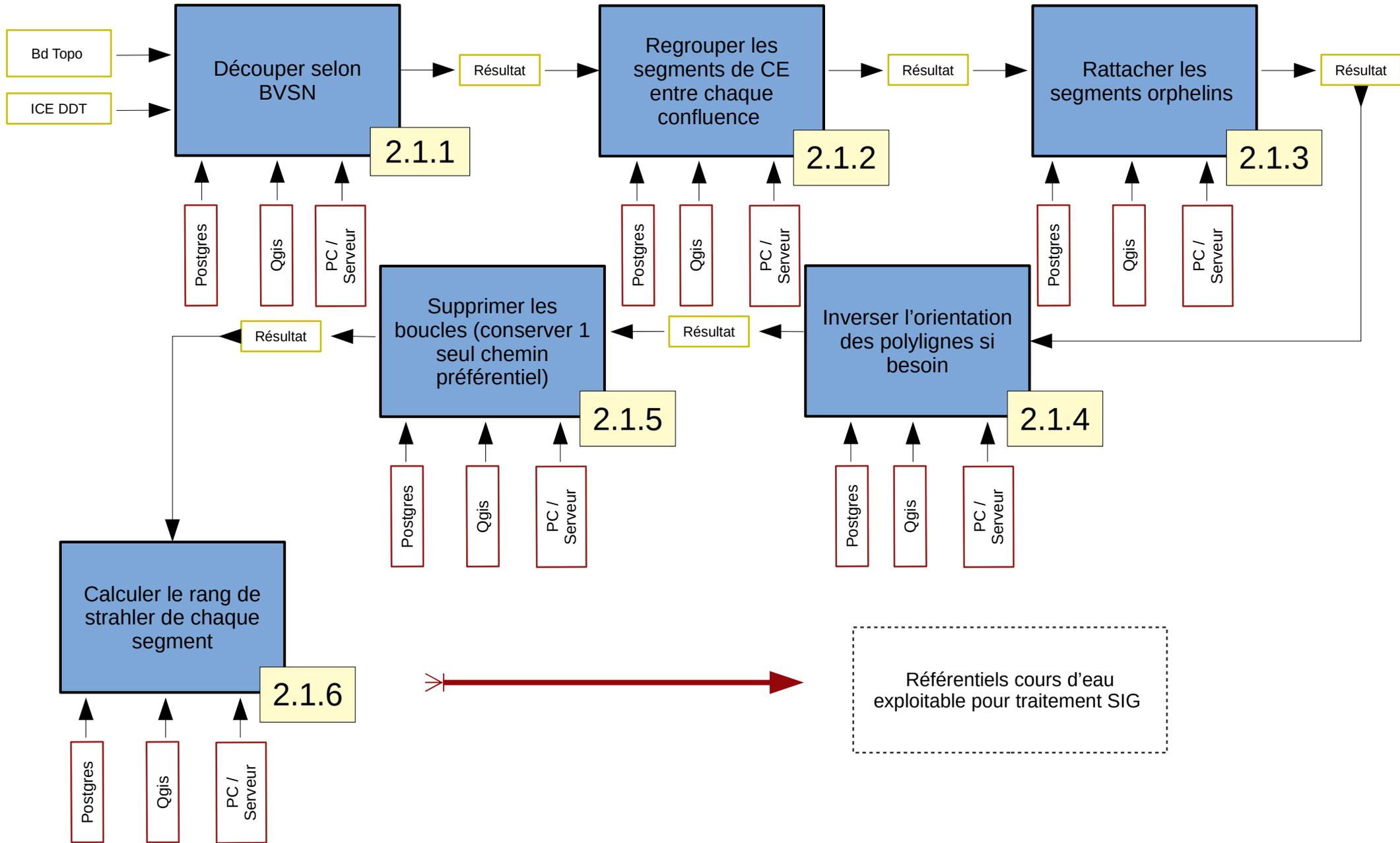
Niveau 1 — Se documenter sur les TBV, les méthodologies de localisation, caractérisation et hiérarchisation



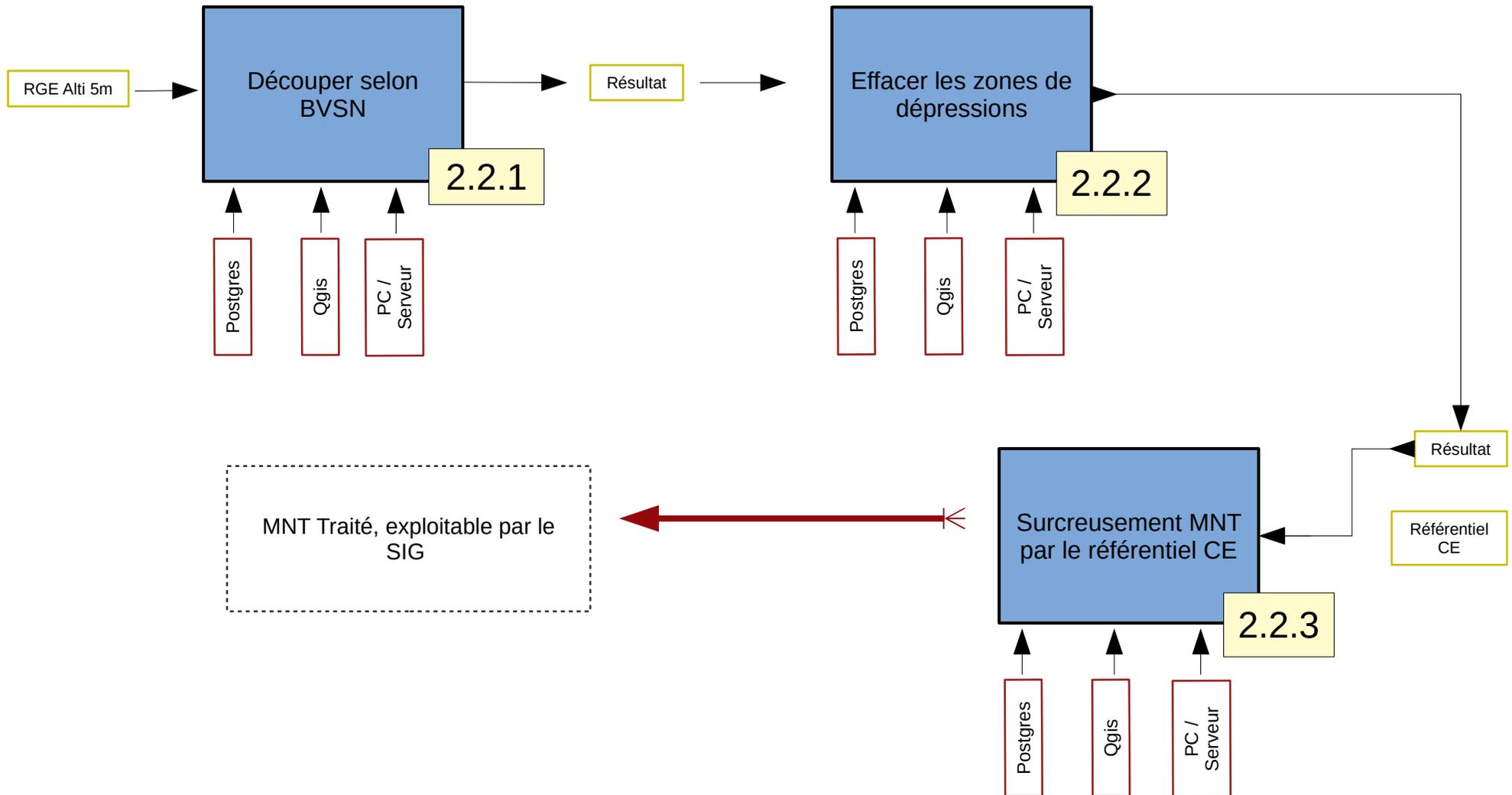
Niveau 1 – Localiser les TBV



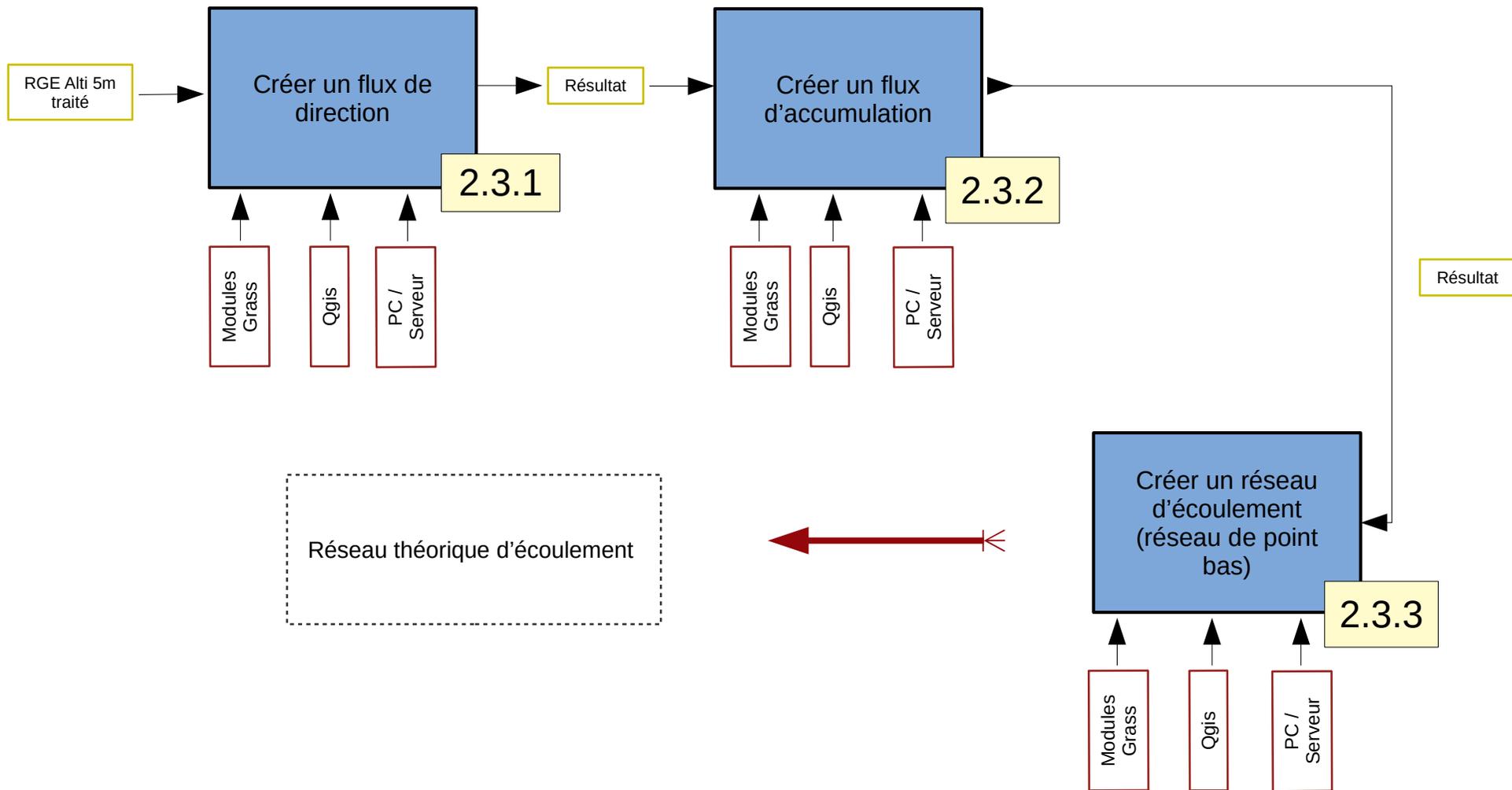
Niveau 2 – Créer Référentiel cours d'eau



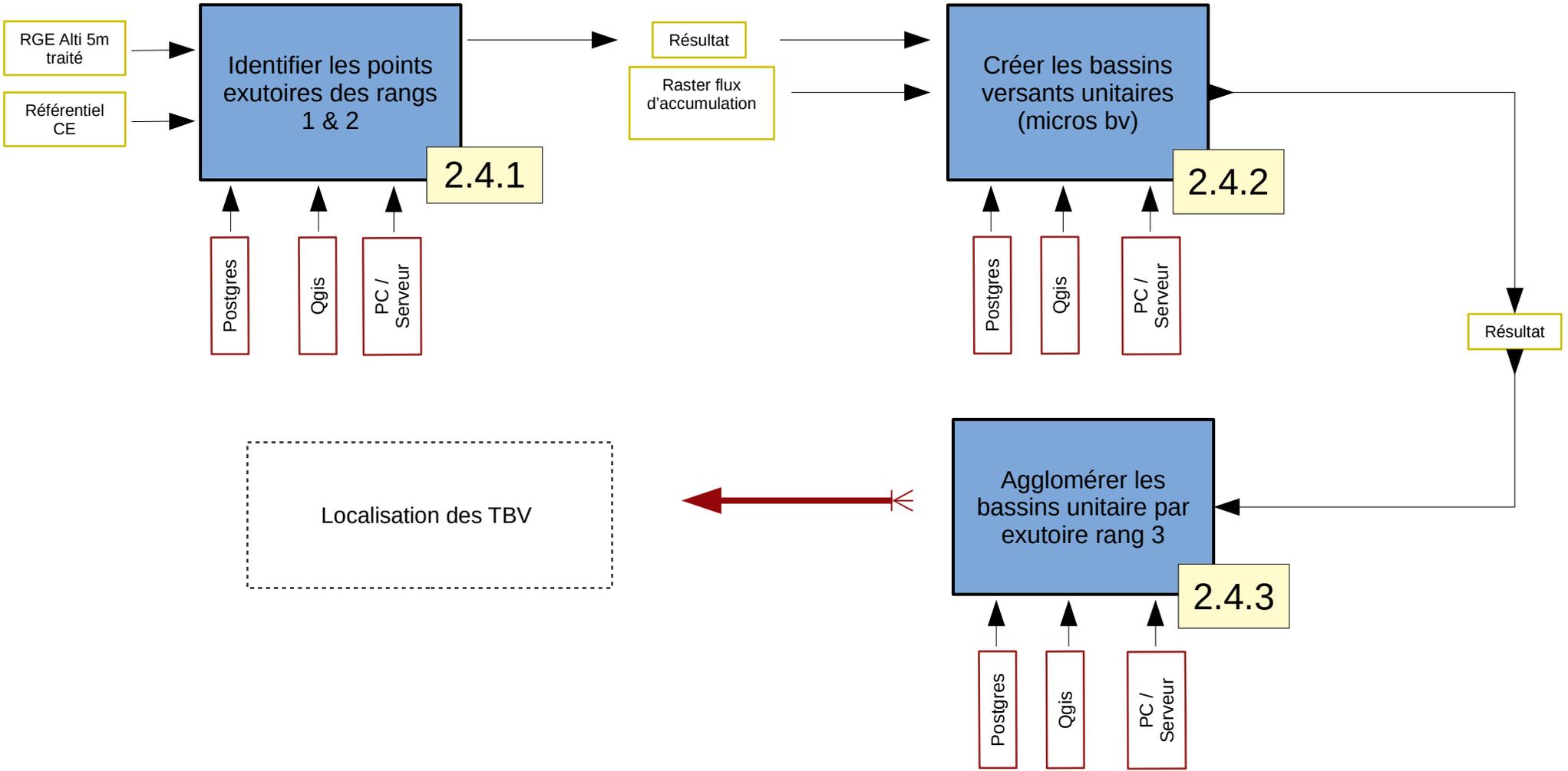
Niveau 2 – Traitement MNT



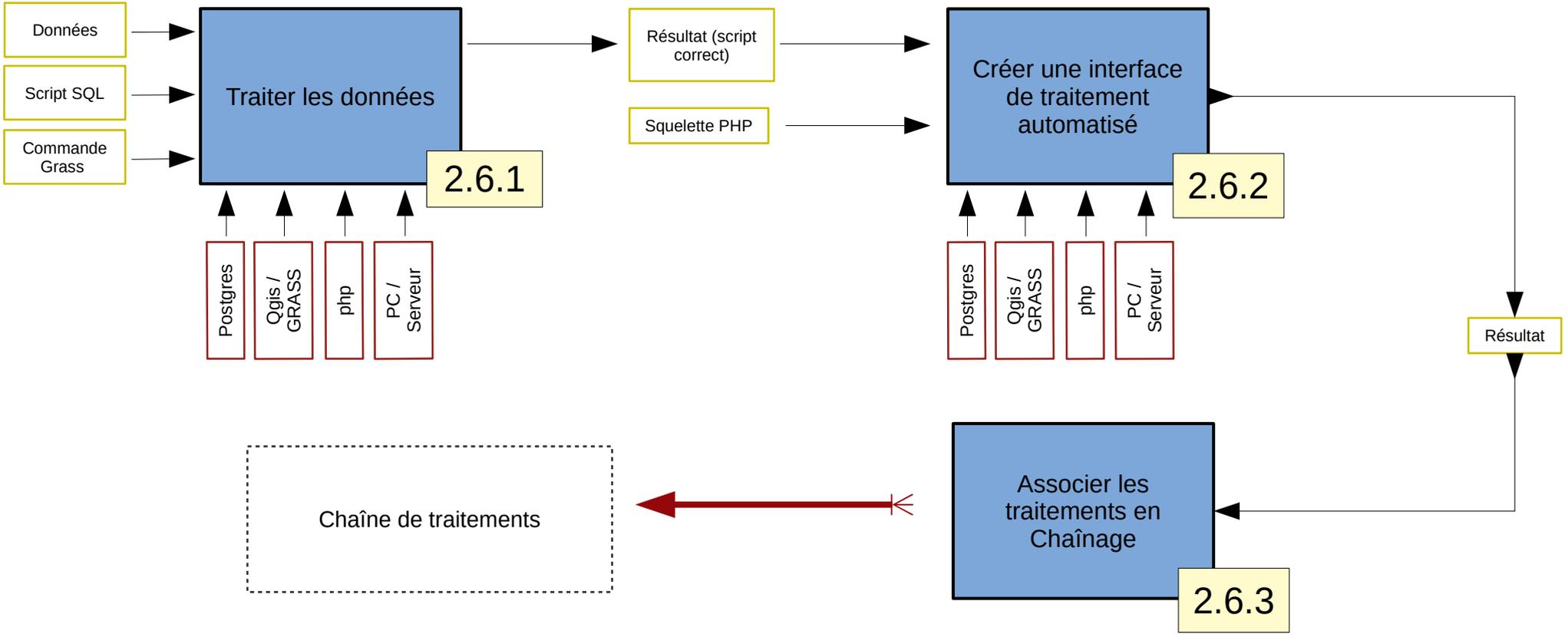
Niveau 2 – Créer un réseau théorique d'écoulement (Utile également pour la création du référentiel CE!)



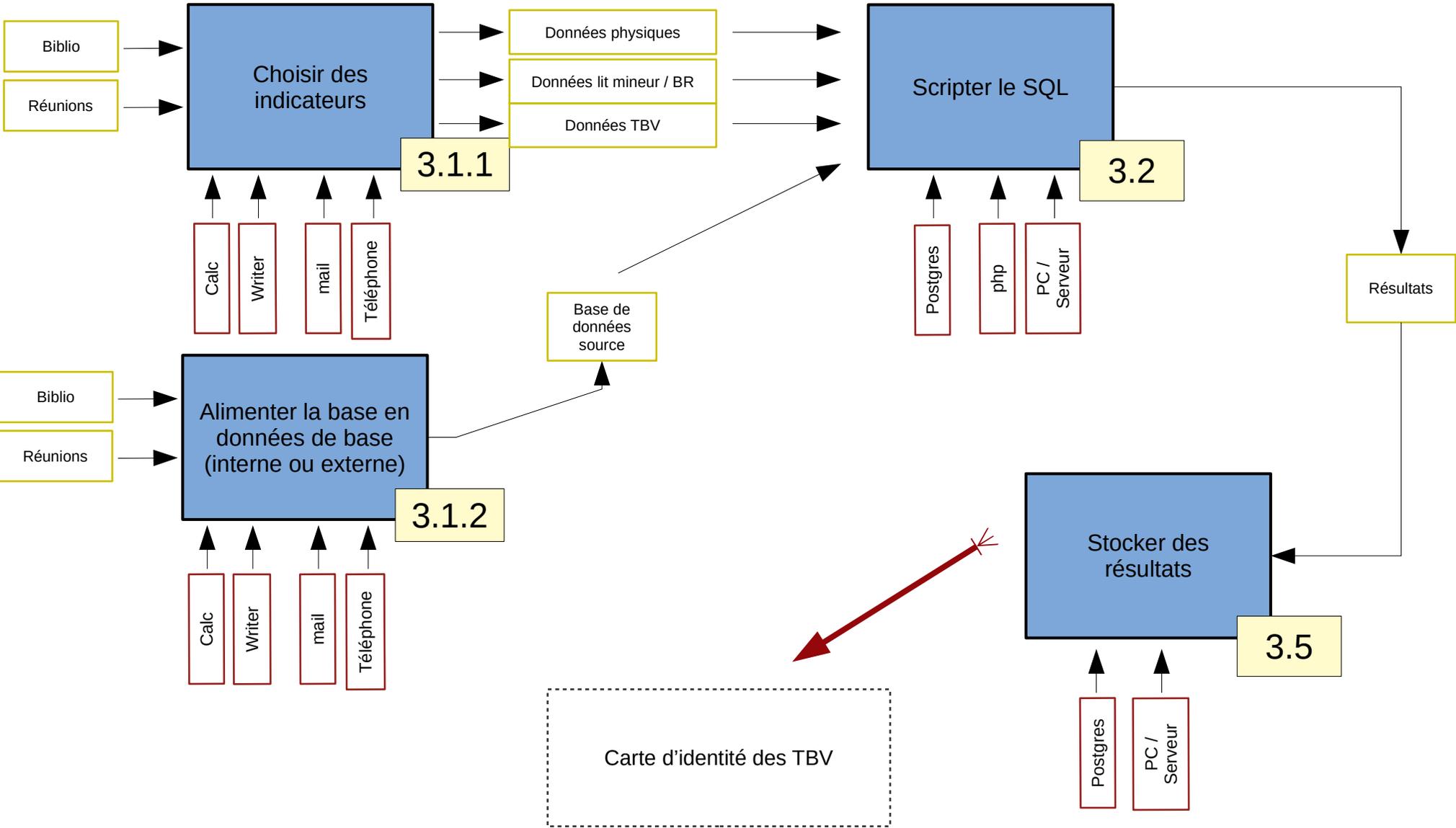
Niveau 2 – Calculer les BV



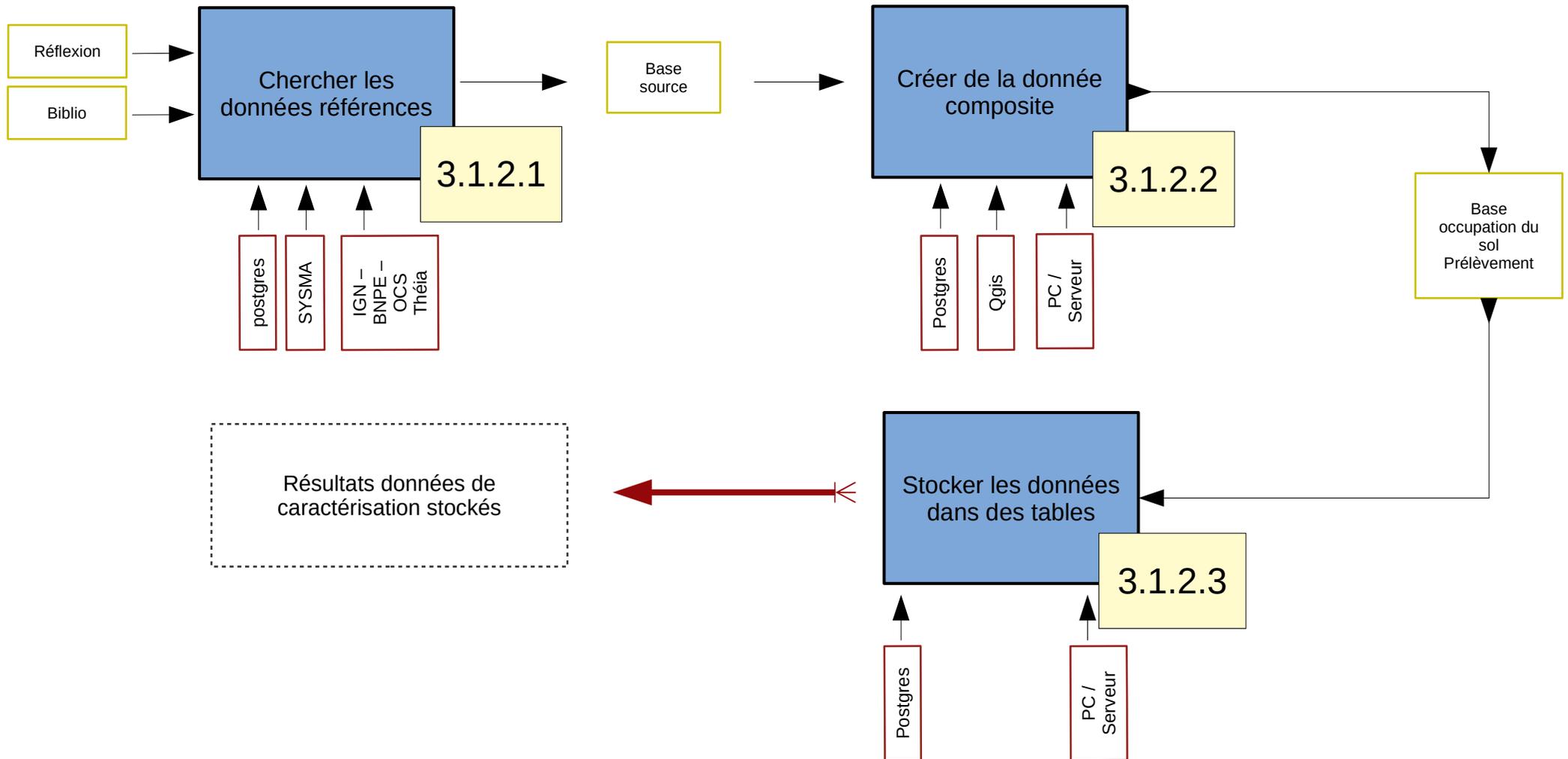
Niveau 2 – Créer une chaîne de traitement



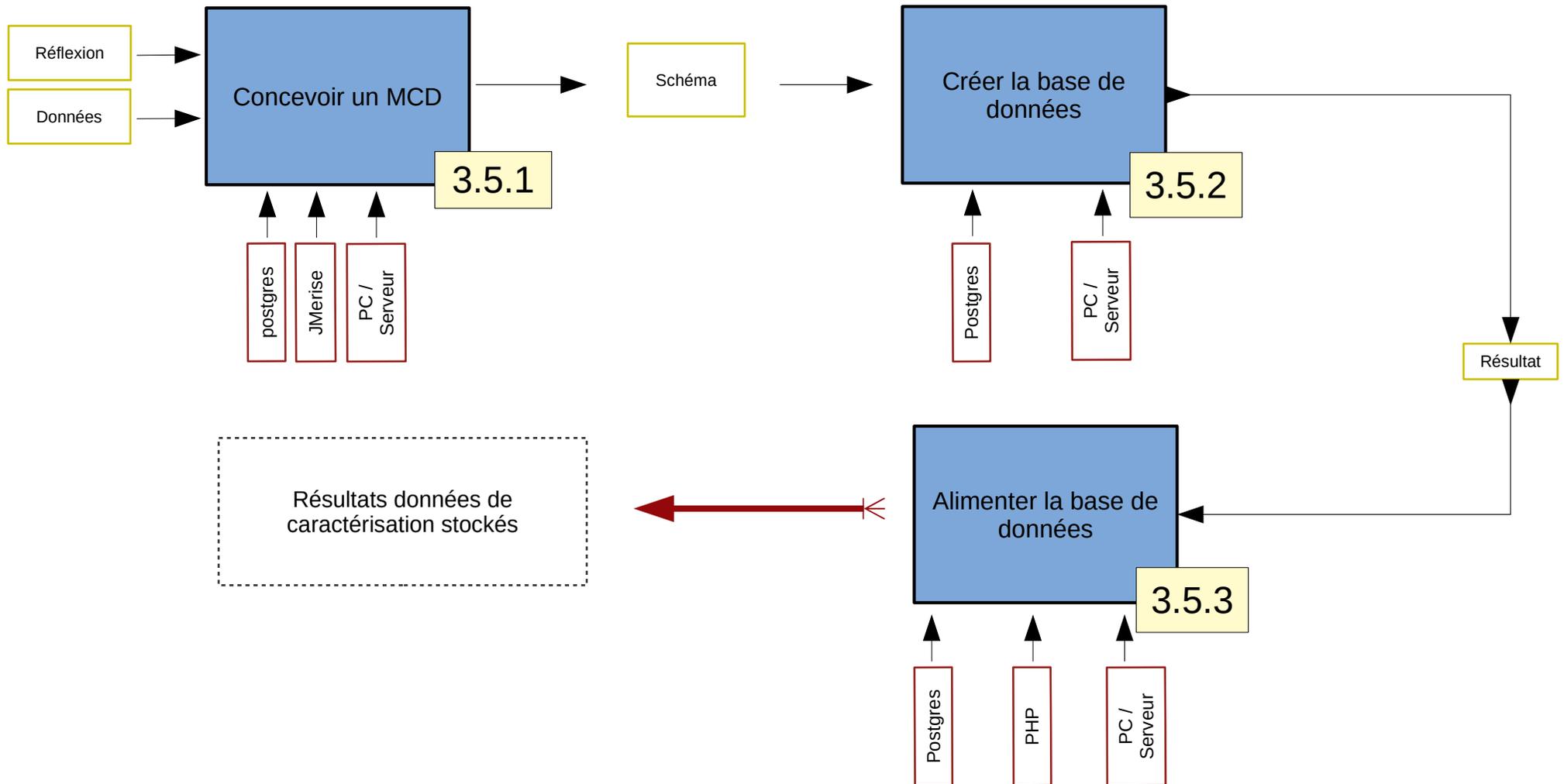
Niveau 1 – Caractériser les TBV



Niveau 2 – Alimenter la base en données de base



Niveau 2 – Stocker les résultats



Niveau 1 – Hiérarchiser les TBV

