

Une base de données topographiques collaborative pour le massif du Folly ?

Par Xavier Robert

1. Introduction

Cet article s'adresse en premier lieu à toutes les personnes actives en exploration et en topographie sur le massif du Folly à Samoëns. Mais il peut aussi intéresser les spéléos qui sont intéressés par mettre en place sur un autre système karstique une base de données topographiques similaire, ou qui traînent leurs bottes dans les cavités péruviennes, étant donné que j'ai aussi mis en place une base de données topographiques collaborative sur les cavités péruviennes. Cet article ne traite pas de l'utilisation de Therion ni de son intérêt.

Au jour d'aujourd'hui, sur le massif karstique du Folly, nous avons :

- 1166 m topographiés et 467 m non topographiés sur les zones A et PV,
- 234 m topographiés et 237 m non topographiés sur les Avoudrues,
- 31992 m topographiés et 3321 m non topographiés sur le système du Jean-Bernard,
- et 19644 m topographiés et 4034 m non topographiés sur le synclinal de la Combe au Puaires.

L'ensemble de ces données est actuellement sous la forme de deux bases de données distinctes :

- La première (dite « base de données Cavités Samoëns »), en cours de migration vers BaseKarst, est constituée de toutes les fiches cavités (coordonnées, historique, description, biblio, image de la topo/schéma).
- La seconde (base de données topographiques) est constituée de l'ensemble des données topographiques, ordonnées et géoréférencées. Ces données sont associées aux dessins de la topographie en plan et en coupe développée.

La base de données Cavités Samoëns est accessible à tous ceux qui en font la demande (voir auprès des administrateurs Stéphane Lips et Antoine Aigueperse). Elle est aussi modifiable par les utilisateurs pour correction d'erreurs, ajout d'informations et création de nouvelles fiches « cavité ». Pour son utilisation, se référer à l'aide de BaseKarst sur le site des développeurs (GIPEK, <http://gipek.fr/>).

Suite à un travail sur plusieurs années et grâce à l'important travail initial de Steph de classification des données, transcription des données et dessins topographiques, j'ai migré l'ensemble de nos

données topographiques sous le logiciel libre de topographie Therion (<https://therion.speleo.sk/>), en les hiérarchisant. Actuellement, après chaque séance topographique sur Samoëns, je m'occupe du report des données, au format Therion, qu'on m'envoie et souvent je dessine aussi la topographie en fonction des carnets topographiques. Cela fonctionne, mais i) demande pas mal de travail et d'aller-retour, ii) le report topographique peut être en conséquence faussé par ma méconnaissance de la zone explorée et topographiée, et iii) implique que je suis la seule personne ressource de cette base de données topographiques, ce qui n'est pas acceptable sur du long terme.

Je propose donc de mettre en place un système de base de données collaborative afin d'alléger le travail et d'ouvrir la possibilité de participer à la maintenance et l'évolution de la base de données topographiques. Par souci de simplicité, j'ai fait le choix de mettre en ligne cette base de données topographiques sur la plateforme *Github* (<https://github.com/>) à l'adresse :

<https://github.com/robertxa/Topographies-Samoens-Folly>. Evidemment, comme pour chaque topographie du club que j'ai travaillée, j'ai mis le travail sous licence libre Créative Commons avec attribution, partage à l'identique et non commerciale (Attribution-ShareAlike-NonCommercial) : <http://creativecommons.org/licenses/by-nc-sa/3.0/>.

2. Description de la base de données

La base de données topographiques a pour buts :

- de construire les topographies de chaque cavité topographiée sur le massif du Folly,
- d'exporter la topographie de n'importe quel sous-réseau appartenant à une cavité plus importante,
- et enfin de construire des plans, cartes et modèles géoréférencés en trois dimensions de chaque système hydrogéologique, de tout le massif ou de cavités sélectionnées à la demande.

J'ai donc tenté de structurer la base de données de façon, d'une part, à répondre aux principaux buts évoqués ci-dessus, et d'autre part à faciliter au maximum son utilisation et sa mise à jour.

2.1. Fichiers contenus dans la base de données

La base de données topographiques a été pensée pour une utilisation avec le logiciel libre Therion parce qu'il permet entre autres le travail sur des systèmes hydro-karstologiques complets. En conséquence, cette base de données contient tous les fichiers nécessaires à la compilation et la construction des différentes topographies :

- Les fichiers de données topographiques ainsi que leurs jointures, si existantes (fichiers avec l'extension *.th*).
- Les fichiers de dessins des plans et des coupes (fichiers avec l'extension *.th2*).
- Les fichiers de compilation qui permettent la compilation des données topographiques et des dessins (fichiers avec l'extension *.thconfig*).

De plus, les topographies ante-2011 avaient été faites au format numérique en utilisant Visual Topo. J'ai gardé en archives toutes les données Visual Topo qui ont été produites avant le passage à Therion (fichiers avec l'extension *.tro*).

Enfin, à la racine, j'ai ajouté un fichier de définition de symboles et de configurations qui est appelé par chacun des fichiers de compilation *.thconfig* (fichier *config.thc*).

Tous ces fichiers-là sont des fichiers textes, qui peuvent être édités avec un éditeur de texte digne de ce nom (i.e. *Notepad++* sous Windows, *TexWrangler*, *Atom*, *Brackets* sous n'importe quel OS).

Enfin, vous trouverez aussi un fichier caché *.gitignore* (décrit dans le paragraphe 4 : Petit Tutoriel pour utiliser GitHub) qui définit les fichiers qui peuvent être produits ou ajoutés dans les dossiers de la base de données, mais qui ne seront pas synchronisés avec la base de données en ligne. Pour des raisons de taille des fichiers et de la base de données, j'ai volontairement exclu tous les fichiers :

- de sortie de Therion (les *.pdf*, *.log*, *.kmz*, *.lox*, *.3d*, *.html*, *.sql*, *.shp...*),
- images, que ce soit les scans des carnets topos ou des dessins à la main ayant servi à dessiner les topographies ;
- qui ont été utilisés pour produire par exemple les modèles numériques de terrain (MNT).

2.2. Structure de la base de données topographiques

J'ai structuré la base de données en systèmes hydrologiques : i) le réseau du gouffre Jean-Bernard, ii) le réseau de la Combe aux Puaires, iii) le réseau des Avoudrues, et iv) le système du A21-A24. Puis, j'ai divisé chaque système selon le système de classification Vulcain en zones que nous utilisons depuis des années. Et évidemment, pour chaque zone, j'ai créé un dossier par cavité, plus un dossier par cavité appartenant à plusieurs zones (e.g. le gouffre Jean-Bernard).

Pour les dossiers ou cavités complexes, afin de garder des dossiers simples, j'ai ajouté :

- un dossier *Outputs* qui contient les sorties de la compilation Therion, à savoir les topographies en pdf, les shapefiles, les modèles 3D, les listes...

- parfois des dossiers *data-th* et *data-th2* qui contiennent respectivement les données topographiques et les dessins.

Si vous modifiez le nom des dossiers, ou si vous en ajoutez, **il est fondamental de ne pas inclure d'espaces ou de caractère accentués/spéciaux**, sous peine d'avoir des difficultés de compilation parce que les chemins des fichiers seront mal reconnus.

2.2.1. A la racine

En plus des fichiers *.gitignore* et *config.thc*, la racine permet de construire le plan et le modèle global de tout le massif du Folly. Pour cela, il y a plusieurs fichiers :

- *Samoens_3D-IGN.thconfig* : permet de construire le plan et le fichier 3D avec les orthophotos et le MNT de l'IGN (voir le wiki de Therion pour une description de la structure de ce type de fichiers),
- *Samoens_3D-Map.thconfig* : permet de construire le plan et le fichier 3D avec les courbes de niveaux issues du MNT de l'IGN, et le MNT de l'IGN,
- *Samoens_3D-Sat.thconfig* : permet de construire le plan et le fichier 3D avec les photos satellites de Google Earth et le MNT de l'IGN,
- *Samoens-GIS/dem/**.th* : ce sont les MNT au format Therion utilisés pour construire les modèles 3D avec la topographie de surface qui est un overlay. Les images de ce dossier sont celles utilisées pour construire les overlays,
- *README.rst* : ce fichier correspond à la description du projet, au format rst.
- *TODO.txt* : fichier texte qui décrit le travail qu'il reste à faire sur la base de données pour l'améliorer.

Ce dossier racine contient aussi un dossier par système hydrogéologique supposé :

- *Avoudrues/* : cavités de tout le système hydrologique des Avoudrues,
- *Combe-Puaires/* : cavités de tout le système hydrologique de la Combe aux Puaires
- *JB/* : cavités de tout le système du gouffre Jean-Bernard
- *Reseau_A21/* : cavités de tout le système du réseau de la Combe aux Puaires.

Il contient aussi un dossier *Samoens-GIS/dem/* avec les MNT utilisés pour construire les modèles en trois dimensions et les images associées.

2.2.2. Dans chaque dossier « Système »

Chaque dossier « système » possède une structure similaire. D'une part, il contient un dossier par zone définie par le groupe Vulcain et appartenant au

système en question. Par exemple, dans le dossier système *JB/*, vous pourrez trouver le dossier *zone-CH/* avec toutes les données associées à cette zone.

En en-tête, il y a aussi a minima deux fichiers Therion. Soit *ZZ* le nom du système, soit *Avoudrues*, *Combe-Puaire*, *Synclinal-JB* ou *Reseau-A21* :

- *ZZ.th* : ce fichier « en-tête » du système *Z* contient un appel à tous les fichiers en-tête des zones du système *ZZ*.
- *ZZ.thconfig* : ce fichier permet de construire le plan et/ou l'atlas du système, en utilisant les zones définies dans le fichier en-tête *ZZ.th*.

Enfin, pour les systèmes où nous connaissons des cavités qui sont à cheval sur plusieurs zones, il y a un dossier correspondant à cette cavité. C'est le cas du gouffre *Jean-Bernard* qui est dans le dossier *JB/JB/*.

2.2.3 Dans chaque dossier « zone »

La structure de chaque dossier « zone » est similaire, et ressemble à la structure des dossiers « système ». Dans chaque dossier « zone », il y a un dossier par cavité topographiée, nommé par le nom de la cavité, comme par exemple le dossier *CH3/* dans le dossier *zone-CH/* lui-même dans le dossier *JB/* qui contient toutes les données topographiques et le dessin de la topographie de la cavité *CH3*.

Comme pour les dossiers « systèmes », en en-tête, il y a aussi a minima trois fichiers Therion. Soit *XX* le nom de la zone :

- *trousXX.th* : Ce fichier Therion contient toutes les données correspondantes aux cavités explorées dans la zone *XX*, mais non topographiées. Les données en question sont le nom de la cavité, ses coordonnées (latitude/longitude/altitude) ainsi que la longueur explorée. Ce fichier est appelé par le fichier *XX.th* du même dossier zone.
- *trousXX.th2* : le fichier *trousXX.th* peut parfois être associé à un fichier de dessin sur lequel sont pointées les entrées avec le figuré « entrée ». Ce fichier est facultatif.
- *XX.th* : ce fichier « en-tête » de la zone *XX* contient un appel à tous les fichiers topographiques et de dessin de chaque cavité topographiée, et au fichier *trousXX.th*. Le fichier *XX.th* est appelé par le fichier en-tête *ZZ.th* du système *ZZ* auquel la zone appartient.
- *XX.thconfig* : fichier de commandes de compilation Therion permettant de construire le plan et le modèle 3D de la zone *XX*.

2.2.4 Dans chaque dossier « cavité »

Soit *xx* le nom de la cavité en question :

- *xx.th* : les fichiers *.th* stockent les données topographiques en elles-mêmes (voir le wiki de Therion pour une description de la structure de ce type de fichier). Ces fichiers peuvent être dans un dossier *data-th/* pour simplifier le dossier de la cavité. Lorsqu'il y a plusieurs fichiers *xx.th*, le fichier *xx-tot.th* permet d'appeler tous les fichiers topographiques, de les joindre entre eux, et de joindre les dessins associés.
- *xx.th2* : ce sont les fichiers de dessins. La plupart du temps, *xx.th2* correspond au plan, et *xx-coupe.th2* à la coupe développée. Pour les cavités complexes, ces fichiers de dessins peuvent être dans un dossier *data-th2/*.
- *xx-maps.th* : ce fichier permet de définir les différentes maps qui permettent dans le *xx.thconfig* de sélectionner la partie de la topographie à exporter.
- *xx.thconfig* : ce fichier contient les commandes de compilation Therion adaptées à la cavité *xx*.

Etant la cavité la plus importante et complexe, le cas du gouffre *Jean-Bernard* est un peu spécifique : pour faciliter la publication dans l'*Echo des Vulcains* de chaque sous-réseau exploré, j'ai créé des sous-dossiers correspondant à chaque sous-réseau dans *Data/Data-th/Crops_reseaux*.

3. Prérequis

Pour pouvoir utiliser cette base de données topographiques et participer à sa maintenance et son évolution, certains points sont prérequis, que j'expose ci-dessous.

3.1. Connaissances de base du logiciel Therion

Cet article n'a pas pour but d'expliquer le fonctionnement du logiciel libre **Therion**, mais comme la base de données est construite autour de ce logiciel, il est nécessaire d'avoir quelques bases pour pouvoir utiliser les données. Vous trouverez toutes les informations nécessaires sur le site internet du logiciel (<https://therion.speleo.sk/>) et sur son wiki (<https://therion.speleo.sk/wiki/doku.php> ; en anglais mais très complet). Afin d'aider à l'utilisation du logiciel, j'ai aussi écrit des templates qui peuvent être utiles et que j'ai mis en ligne sur <https://github.com/robertxa/Th-Config-Xav>.

Normalement, chaque cavité peut être compilée avec le logiciel graphique *xtherion* ou dans le terminal ou la fenêtre console par la commande *therion xx.thconfig* où *xx* correspond au nom de la

cavité (voir le contenu du dossier dans lequel vous travaillez et la documentation Therion).

Pour visualiser les modèles en trois dimensions (fichiers avec l'extension .lox), il y a plusieurs possibilités :

- En installant la suite libre Therion, vous installez aussi le logiciel libre *Loch* qui est le visualisateur 3D de Therion. Dans ce logiciel, une fois un modèle 3D ouvert, il est possible d'importer d'autres modèles de cavités proches avec la fonction *Fichier → Importer*.
- Sinon, l'autre possibilité est d'ouvrir un des modèles 3D sur la page internet du club Vulcain (basé sur l'application CaveView, <https://aardgoose.github.io/CaveView.js/>) ou directement sur le site internet du développeur du visualisateur et, une fois le modèle 3D chargé, effectuer un glisser-déposer du modèle que vous voulez visualiser (<https://www.groupe-speleo-Vulcain.com/explorations/topographies-3d/>).

3.2. Connaissances de base de la plateforme GitHub :

GitHub (<https://github.com/>) est un service web d'hébergement et de gestion de projets qui utilise le logiciel de gestion open-source de version *Git*. Alors que le système traditionnel open-source amène chaque contributeur à télécharger les sources du projet et à proposer ensuite ses modifications à l'équipe du projet, GitHub fonctionne sur le principe du *fork* (branche) par défaut : toute personne *forkant* le projet devient publiquement de facto le chef de son projet portant le même nom que l'original.

3.2.1. Le fork ou la branche

Une branche correspond à une copie du projet principal (« *master* ») à un moment donné, et qui peut être modifiée de manière autonome sans impacter le projet initial. Ceci permet de travailler à plusieurs en parallèle, sans pour autant altérer le projet initial. Après avoir terminé de travailler sur une branche, il est possible d'en faire un projet autonome, ou mieux, de proposer de fusionner les modifications de la branche en question avec le projet initial. C'est ce qu'on appelle proposer une « *pull request* ». Le ou les propriétaire(s) du projet « *master* » accepteront (ou non !) les modifications proposées, une à une.

C'est ce qui fait la force de la plateforme Github.

3.2.2. La gestion de versions

GitHub est basé sur le logiciel Git qui est un logiciel de contrôle de versions. Cela veut dire qu'il gère les modifications d'un projet sans écraser n'importe quelle partie du projet. En pratique, Git

permet d'enregistrer spécifiquement des modifications différentes d'un projet par différentes personnes sans écraser le travail des autres personnes. Plus tard, nous pouvons choisir de fusionner telles ou telles parties sans perdre le travail qui a été fait. En plus, s'il le faut, nous pouvons revenir en arrière parce que Git conserve une copie de tous les changements.

3.2.3. Pourquoi ce choix ?

J'ai choisi d'utiliser cette plateforme d'une part pour les conditions d'utilisations (on conserve la propriété complète de tous les projets déposés), l'hébergement gratuit, l'accessibilité à tous et d'autre part pour les facilités d'utilisation de Git via la plateforme. Via un logiciel multiplateforme d'aide à l'utilisation de Git pour les « non-informaticiens ».

En effet, il est possible d'interagir avec les projets sur la plateforme *GitHub* uniquement avec les commandes en ligne de l'outil *Git*.

Mais le logiciel *GitHub Desktop* (multiplateforme, <https://desktop.github.com/>) facilite grandement le processus. Je vous conseille de l'installer pour travailler sur la base de données.

Il existe d'autres plateformes de ce type (comme *Gitlab* par exemple), mais un peu plus difficiles d'accès pour les non-informaticiens. C'est pourquoi j'ai fait ce choix, mais il serait tout à fait possible de migrer le projet *GitHub* (*actuellement racheté par Microsoft*) vers une plateforme libre.

Comme l'utilisation de l'interface *GitHub* n'est pas forcément intuitive pour tout le monde, je propose ci-dessous un petit tutoriel pour l'utiliser dans le cadre de la maintenance (et du travail collaboratif) sur la base de données de Samoëns.

4. Petit tutoriel pour utiliser *GitHub*

La plateforme *GitHub* est en anglais. Les termes utilisés sont simples et compréhensibles, mais dans la suite, je tente tout de même une traduction des termes les plus importants pour faciliter la compréhension aux personnes fâchées avec la langue de Shakespeare. Il existe beaucoup de documentation sur internet, à la fois en anglais et en français, ne pas hésiter à la chercher si besoin. Et évidemment, comme pour tout, il faut lire toutes les lignes, même les petites !

4.1. Comment télécharger la base de données topographiques et jouer avec

Télécharger la base de données pour usage personnel est simple : sur la page internet du projet (repository en langage Github) <https://github.com/robertxa/Topographies-Samoens-Folly>, cliquez sur le bouton vert « *Clone or download* » (= cloner ou télécharger), puis sur « *Download Zip* » (télécharger le zip). Il ne vous

reste qu'à décompresser l'archive zip dans le dossier où vous voulez travailler, puis à jouer avec !

4.2. Comment travailler collaborativement sur la base de données, la mettre à jour et proposer des modifications

Pour travailler collaborativement, télécharger la base de données sur son ordinateur personnel ne suffit pas. Il est nécessaire de se créer un compte *GitHub*, afin de pouvoir profiter de toutes les options de *Git* et *GitHub*. Je décris ci-dessous comment créer un tel compte, puis comment faire un *fork* de la base de données, comment mettre à jour ce *fork* et enfin comment proposer une *pull request* à un des administrateurs de la base de données pour qu'il révise et intègre les modifications.

Je vous conseille aussi de télécharger le logiciel Github Desktop (<https://desktop.github.com/>).

4.2.1 Créer un compte personnel dans GitHub

Sur la page principale de *GitHub* (<https://github.com/>), cliquez en haut à droite sur « *Sign in* » (= s'enregistrer), puis, sur la nouvelle page « *create a new account* » (= créer un nouveau compte). Remplir les champs demandés et suivre les instructions.

Le compte ainsi créé se trouvera à la page <https://github.com/Username/> où Username est à remplacer par votre identifiant personnel (par exemple, mon Username est robertxa, et mon compte se trouve sur <https://github.com/robertxa/>). Une fois le compte créé, pour se connecter, accéder à n'importe quelle page de Github (la page principale, ou la page d'un des projets existants), et cliquez en haut à droite sur « *Sign in* » (= s'enregistrer). Puis, sur la nouvelle page, entrez l'*username* (=identifiant défini lors de l'ouverture du compte) ou l'*adresse email* utilisée pour ouvrir le compte, et le *password* (= mot de passe défini lors de l'ouverture du compte).

4.2.2. Créer un « fork » et le mettre à jour

Avant tout, lancez le logiciel *Github Desktop*, et connectez vous via ce logiciel à votre compte *Github*. Ensuite, cliquez sur *File* → *Clone Repository* et entrer l'adresse web de la base de données. Avant d'appuyer sur le bouton « cloner », vous devez renseigner votre dossier local (sur votre ordinateur) dans lequel sera clonée la base de données.

Une fois le clonage effectué, vous allez pouvoir travailler directement dans la base de données, dans le dossier local que vous avez défini au moment du clonage. Chaque fois que vous allez effectuer une modification, elle apparaîtra dans le logiciel *Github Desktop*. Pour mettre à jour votre repository Github (votre branche), il faut cliquer sur « Commit » (il faut renseigner obligatoirement le nom du commit),

puis sur Publish. Les modifications seront copiées sur le serveur.

4.2.3. Proposer une « pull request »

Une fois votre contribution prête, il faut l'ajouter au projet initial. Pour cela, il faut demander à ajouter vos modifications et/ou ajouts à la base de données principale. Ceci s'effectue par ce qui s'appelle une « *pull request* ».

Vous pouvez l'effectuer, soit via l'interface en ligne de *Github*, soit via le logiciel *Github Desktop*. Sur votre compte, allez dans votre projet et cliquez sur *Create Pull request*. Dans le logiciel, sélectionnez le projet (votre branche), puis allez dans *Branch* → *Create Pull Request*. Les deux méthodes mènent à la page permettant de renseigner les différents champs et de finaliser la démarche.

4.2.4. Mettre à jour son propre repository via le « Master »

Une fois que la base de données principale a été mise à jour, vous pouvez mettre à jour votre branche en cliquant dans le logiciel *Github Desktop* sur *Repository* → *Pull*.

4.3. Petit mémo des commandes en ligne Git pour les Geeks irréductibles

Sur internet, il y a de nombreux tutoriels pour utiliser Git, je ne vais pas les reprendre, mais voilà quelques commandes importantes au cas où :

- *git init <dossier>* : transformer n'importe quel dossier en un dépôt Git.
- *git clone* : cloner la base de données sur son ordinateur local.
https://github.com/robertxa/Topographies-Samoens_Folly
- *git status* : indique les fichiers que vous avez modifiés.
- *git diff* : indique concrètement ce que vous avez changé.
- *git add nomfichier1 nomfichier2* : ajouter les fichiers à la liste de ceux devant faire l'objet d'un commit.
- *git commit -a* : pour commiter tous les fichiers listés dans *git status* dans les colonnes « changes to be committed » et « Changed but not upadted ».
- *git log* : affiche la liste des commits.
- *git HEAD/HEAD~/HEAD^^/HEAD~2/N°_de_commit* : indique à quel commit on souhaite revenir (HEAD = dernier commit ; HEAD^ = avant-dernier commit ; HEAD^^ = HEAD~2 = avant avant dernier commit ; N°_de_commit).
- *git HEAD/HEAD~/HEAD^^/HEAD~2/N°_de_commit --hard* : annule les commits et perd tous les changements.

- *git pull* : téléchargement des nouveautés depuis le serveur.
- *git push* : envoyer vos commits sur le serveur.
- *git branch* : pour voir toutes vos branches.
- *git branch <nom_branche>* : créer une branche.
- *git branch-d <nom_branche>* : supprimer une branche.
- *git checkout <nom_branche>* : aller dans la branche nom_branche.
- *git merge* : fusionner les changements.

données du massif du Folly, vous avez tout en main, à vous de jouer. Pour l'instant, cette base de données est sur mon compte GitHub. C'est à discuter, mais il serait peut-être possible d'ouvrir un compte pour le groupe Vulcain, avec deux ou trois personnes administratrices.

Pour ceux qui sont intéressés, j'ai créé un projet similaire pour les cavités Péruviennes (https://github.com/robertxa/Mapas_Cavernas_Peru). N'hésitez pas à participer à ces projets, à m'envoyer des commentaires ou des propositions de modifications si besoin, et surtout, n'hésitez pas à me demander de plus amples détails et/ou aide si besoin !

5. Perspectives

Et voilà, vous êtes prêts à utiliser la base de

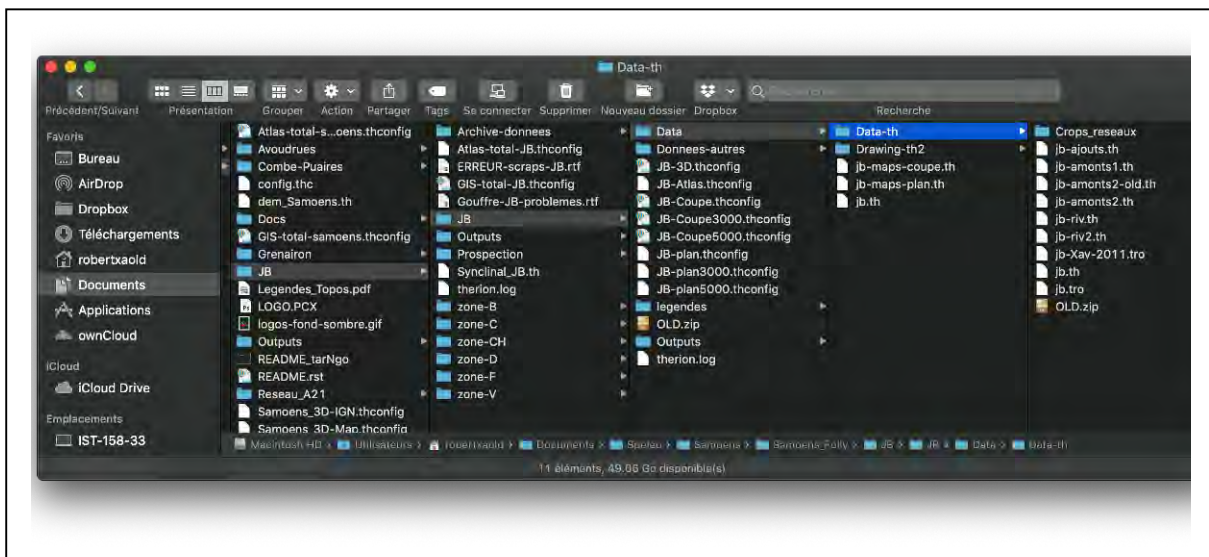


Figure 1 : Exemple de structure de dossier pour le dossier du gouffre Jean-Bernard.



Le vallon des Chambres (FA, 13/09/2019)