Valorisation des données de biodiversité d'une application naturaliste à travers le développement d'un module applicatif " Tableau de bord " et la détection automatique de données atypiques

Elsa Guilley

MINISTERE DE L'AGRICULTURE, DE L'AGROALIMENTAIRE ET DE LA FORET

ECOLE **N**ATIONALE **S**UPERIEURE des **S**CIENCES **A**GRONOMIQUES de **B**ORDEAUX **A**QUITAINE
1, cours du Général de Gaulle – CS 40201 – 33175 GRADIGNAN cedex

MEMOIRE de fin d'études

pour l'obtention du titre

d'Ingénieur de Bordeaux Sciences Agro

Valorisation des données de biodiversité d'une application naturaliste à travers le développement d'un module applicatif « Tableau de bord » et la détection automatique de données atypiques

GUILLEY, Elsa

Spécialisation : AgroTIC – Technologies de l'Information et de la Communication pour l'Agriculture et l'Environnement

Etude réalisée à : Parc national des Ecrins
Domaine de Charance, 05000 GAP

- 2019 -

MINISTERE DE L'AGRICULTURE, DE L'AGROALIMENTAIRE ET DE LA FORET

ECOLE **N**ATIONALE **S**UPERIEURE des **S**CIENCES **A**GRONOMIQUES de **B**ORDEAUX **A**QUITAINE
1, cours du Général de Gaulle – CS 40201 – 33175 GRADIGNAN cedex

MEMOIRE de fin d'études

pour l'obtention du titre

d'Ingénieur de Bordeaux Sciences Agro

Valorisation des données de biodiversité d'une application naturaliste à travers le développement d'un module applicatif « Tableau de bord » et la détection automatique de données atypiques

Optimization of the added-value of biodiversity data of a naturalist application through the development of a data reporting module and the automated detection of outlier data

GUILLEY, Elsa

Spécialisation : AgroTIC – Technologies de l'Information et de la Communication pour l'Agriculture et l'Environnement

Etude réalisée à : Parc national des Ecrins
Domaine de Charance, 05000 GAP

Maître de stage : Camille MONCHICOURT
Tuteur école : Léo PICHON

- 2019 -

Résumé

Le Parc national des Écrins couvre un territoire de hautes montagnes reconnu pour son patrimoine naturel et paysager remarquable. Agir expertement pour la gestion et la sauvegarde de cet espace naturel requiert de le connaître et de le comprendre. Depuis la création du Parc en 1973, les agents collectent chaque jour des données de faune et de flore sur ce territoire, afin de réaliser un inventaire des espèces présentes et suivre leurs évolutions. Ces données, qui sont stockées et gérées dans l'application naturaliste GeoNature, représentent un volume important d'informations hétérogènes en attente d'une validation scientifique. L'objectif de cette étude est d'optimiser la valorisation des données de biodiversité du Parc à travers deux missions : la création d'un module applicatif de reporting de données pour GeoNature et l'implémentation d'un protocole automatique de détection de données atypiques. Pour y parvenir, une analyse des besoins a été réalisée pour chacune des thématiques, permettant de mettre en évidence les enjeux essentiels de ce projet. Les résultats obtenus ont orienté la suite de l'étude vers une analyse de l'existant : les solutions open source *Kibana* et *Metabase* ont été testées pour le volet applicatif, tandis que des protocoles de validation automatique de données naturalistes ont été explorés pour enrichir l'évaluation de la fiabilité des données. Ces états de l'art ont abouti à des choix techniques pertinents pour le Parc, débouchant sur le développement en interne d'un module GeoNature « Tableau de bord » et l'ébauche d'un protocole d'évaluation scientifique de données naturalistes implémentée en base de données. Ces outils sont voués à être renforcés et enrichis par la communauté GeoNature.

Mots clés : parc national, biodiversité, application, open source, base de données, développement web, valorisation des données, validation scientifique

Abstract

The Ecrins national Park covers a high mountain territory recognized as remarkable natural heritage with magnificent landscapes. Safeguarding the Park's biodiversity requires an expert management based on an intimate knowledge and understanding of this natural environment. Since the creation of the Park in 1973, the agents collect on a daily basis data on mountain range fauna and flora, in order to maintain an inventory of existing species and follow their evolutions. The data, which is saved in the naturalist application called GeoNature, represents a significant amount of rich and diverse information pending scientific validation. The purpose of this project is to maximize the added-value of this data through two distinct components : the establishment of a data reporting module for GeoNature and the implementation of an automated protocol to detect outlier data. In this regard, an analysis of needs has been realized for each of these themes, enabling the identification of the key elements to be considered for this project. Based on these findings, the suitability of existing tools has been reviewed : tests have been conducted with the open source softwares *Kibana* and *Metabase* concerning the reporting part, while various automated protocols have been examined for the enhancement of data accuracy. The results of the technical assessments have led the Park to choose to develop internally a new « Dashboard » module for the GeoNature application, as well as to initiate the development of a data assessment automated protocol. These tools will be further developed and enhanced overtime by the GeoNature community.

Key words : national park, biodiversity, application, open source, database, web development, data added-value, scientific data accuracy

Remerciements

Je tiens à remercier mon maître de stage, Camille MONCHICOURT, pour m'avoir offert la possibilité de réaliser mon stage de fin d'études au Parc national des Écrins, dans un domaine qui me tient à cœur. Je suis reconnaissante du soutien et des conseils qu'il m'a apportés, de la confiance et de l'autonomie qu'il m'a accordées, ainsi que de la vision de l'open source qu'il a su me transmettre.

Je souhaite remercier Théo LECHÉMIA pour le temps qu'il a consacré à me former sur la partie développement web, mais également Gil DELUERMOZ pour son soutien technique en base de données.

Je remercie les agents du Parc et les chargés de missions scientifiques qui ont pris de leur temps afin de partager leurs connaissances et participer à mon projet de stage.

Merci aux autres stagiaires et services civiques pour les tours de lac et les pauses café. Ils ont fait de ces 6 mois une expérience humaine mémorable. J'aimerais étendre ces remerciements à l'ensemble des équipes de Charance, qui ont su partager leur bonne humeur au quotidien et qui se sont montrées disponibles en toutes circonstances.

Enfin, j'adresse mes remerciements les plus sincères aux équipes enseignantes de la spécialisation AgroTIC, qui ont fait preuve de beaucoup d'implication et de bienveillance avec nous. Merci de rendre cette formation aussi enrichissante. Je tiens à remercier plus particulièrement mon tuteur école, Léo PICHON, pour l'encadrement pédagogique et l'aide précieuse qu'il m'a apportés tout au long de ce stage.

Table des matières

Introduction **1**

I. Une application naturaliste à l'origine d'un besoin de synthétisation et de validation scientifique de données **3**

1. GeoNature, une application open source et générique de gestion de données naturalistes 3

a) Une application mûrie pendant plusieurs années permettant de saisir, organiser et visualiser des données de faune et de flore 3

b) Une application complexe, à la fois adaptée aux standards nationaux et personnalisable par chaque organisme utilisateur 3

c) Une application basée sur des technologies open source 5

d) Une application en pleine expansion à l'échelle nationale 6

2. Une nécessité de synthétisation et de valorisation des données brutes de GeoNature 6

a) Un module « Tableau de bord » 6

b) Un système de détection automatique des données naturalistes atypiques 6

3. Des besoins identifiés selon plusieurs sources 8

a) Une analyse des besoins complète pour la mise en place d'un module de reporting de données 8

b) Une analyse des besoins partielle concernant la validation scientifique des données 9

4. Des besoins à la fois communs et spécifiques aux différents types d'utilisateurs 9

a) Un module « Tableau de bord » nécessitant de la flexibilité pour répondre à l'ensemble des besoins recueillis 9

b) Un système automatique pour la validation des données ayant déjà été alimenté par des réflexions sur les plans théorique et technique 13

II. Une analyse de l'existant proposant des outils « clé en main » pour la création de tableaux de bord et des protocoles variés de validation automatique de données pour GeoNature 15

1. Des états de l'art destinés à explorer les fonctionnalités offertes par des solutions susceptibles de répondre aux besoins identifiés 15

a) Un module « Tableau de bord » pouvant être développé en interne ou élaboré à partir d'une solution open source existante 15

b) Une analyse de l'existant pour la validation automatique de données reposant sur un travail réalisé par le SINP 16

2. Un module « Tableau de bord » impliquant des besoins précis et variés difficilement accessibles avec des solutions externes 17

a) *Kibana* et *Metabase* : deux outils open source permettant la création de tableaux de bord interactifs 17

b) Des solutions externes ne permettant pas de répondre de manière optimale aux besoins identifiés 20

3. Des méthodes diverses de validation automatique de données naturalistes basées sur des réflexions et traitements similaires 21

a) Une analyse de l'existant basée sur six systèmes d'information traitant des données naturalistes 21

b) Des protocoles de validation automatique structurés par une démarche commune 26

III. Un module « Tableau de bord » et un protocole de détection automatique de données naturalistes atypiques développés en interne par le Parc national des Écrins 30

1. Des mises en œuvre techniques réfléchies et cadrées 30

a) Un module GeoNature nécessitant des technologies et outils spécialisés 30

b) Des protocoles de validation automatique permettant de renforcer la réflexion entamée par le Parc national des Écrins 31

2. Un module « Tableau de bord » interactif et paramétrable, destiné à connaître de nombreuses évolutions 32

a) Un module comportant différents graphiques et cartes flexibles 32

b) Une volonté d'optimisation des performances du module 38

c) Un module perfectible voué à être enrichi par la communauté GeoNature 42

3. Un protocole automatique de notation des données saisies élaboré dans la base de données de GeoNature 44

a) Les profils types de taxon, des référentiels pertinents 44

b) Un calcul de score qui nécessite d'être précisé 45

c) Un protocole automatique exécuté par des actions déclenchées en base de données 46

Conclusion 49

Bibliographie 50

Liste des annexes 52

Liste des figures

Figure 1 : Schéma de l'architecture globale de l'application GeoNature 4
Figure 2 : Schéma décrivant l'insertion de GeoNature dans une chaîne de travail complète, intégrant les référentiels nationaux (source : GeoNature) 5
Figure 3 : Schéma de l'architecture technique de l'application GeoNature 5
Figure 4 : Schéma de la méthodologie générale adoptée pour répondre aux missions du stage 8
Figure 5 : « Dashboard » créé avec le logiciel Kibana 18
Figure 6 : Histogramme réalisé avec le logiciel Metabase 19
Figure 7 : Arbre de décisions pour l'affectation d'un niveau de validité du protocole automatique de l'INPN (source : INPN) 22
Figure 8 : Schéma des cas d'attribution d'un niveau de validité du protocole automatique de SILENE Flore (source : CBNA) 24
Figure 9 : Arbre de décisions pour l'affectation d'un niveau de validité du protocole automatique de Borbonica (source : SINP La Réunion) 25
Figure 10 : Exemple de référentiel établi par taxon (profil type) 27
Figure 11 : Exemples de référentiels établis par paramètre ou critère 27
Figure 12 : Interface d'accueil du module "Tableau de bord" 32
Figure 13 : Histogramme (Graphe 1) développé dans le module "Tableau de bord" 33
Figure 14 : *Cartographie* développée dans le module "Tableau de bord" 34
Figure 15 : Requête SQL de création de la vue matérialisée vm_frameworks 34
Figure 16 : Requête SQL de création de la vue matérialisée vm_synthese 35
Figure 17 : Déclaration du modèle de données de la vue matérialisée vm_synthese dans une classe Python au niveau du back-end avec SQLAlchemy 35
Figure 18 : Fonction Python exécutant la requête SQLAlchemy permettant de récupérer les données du Graphe 1 au niveau du back-end 36
Figure 19 : Service TypeScript (fonction) permettant de récupérer les données du Graphe 1 au niveau du front-end 37
Figure 20 : Code HTML pour l'affichage du Graphe 1 37
Figure 21 : Schéma récapitulatif de l'architecture technique du module "Tableau de bord" 38
Figure 22 : Exemple de résultats attendus pour la requête SQL permettant de retourner les données de la Cartographie 39
Figure 23 : Résultats de la vue matérialisée initialement créée pour la Cartographie 39
Figure 24 : Requête SQL initialement implémentée pour la Cartographie au niveau du back-end . 40
Figure 25 : Requête SQL finale implémentée pour la Cartographie au niveau du back-end 41
Figure 26 : Requête SQL implémentée pour le Graphe 4 au niveau du back-end 42
Figure 27 : Requête SQL de création de la table "reference_validation" (profils types) 46
Figure 28 : Requête SQL de création du trigger permettant la validation automatique des données 46
Figure 29 : Requête SQL de création de la fonction "validation_auto()" 46
Figure 30 : Requête SQL de création de la fonction "calcul_score" 47
Figure 31 : Schéma récapitulatif du protocole automatique d'évaluation scientifique des données de GeoNature 48

Liste des tableaux

Tableau 1 : Liste et description des outils prenant part au projet GeoNature 3
Tableau 2 : Système de hiérarchisation des besoins des utilisateurs de GeoNature 9
Tableau 3 : Bilan de l'enquête ciblant l'identification des différents types d'utilisateurs de GeoNature 10
Tableau 4 : Bilan des besoins identifiés lors de l'analyse des besoins relative au module « Tableau de bord » 12
Tableau 5 : Système de notation des solutions existantes pour le module « Tableau de bord » 16
Tableau 6 : Comparaison des logiciels Kibana et Metabase 20
Tableau 7 : Evaluation des solutions existantes 20
Tableau 8 : Liste et description des systèmes d'information considérés dans l'état de l'art relatif à la validation automatique de données naturalistes 21
Tableau 9 : Liste et description des contrôles réalisés sur les données de l'INPN 22
Tableau 10 : Liste des têtes de réseau du SINP La Réunion (source : SINP La Réunion) 24
Tableau 11 : Liste et description des graphes développés dans le module "Tableau de bord" 32
Tableau 12 : Bilan des tests utilisateurs concernant le module "Tableau de bord" 42
Tableau 13 : Liste et description des paramètres à considérer dans le protocole de détection automatique de données atypiques du PNE 44

Liste des abréviations

AFB = Agence Française pour la Biodiversité

API = Application Programming Interface (interface de programmation)

BDD = Base De Données

CEN = Conservatoire d'Espaces Naturels

CSS = Cascading Style Sheet (feuille de style en cascade)

DEE = Donnée Élémentaire d'Échange

HTML = HyperText Markup Language

HTTP = HyperText Transfer Protocol (protocole de transfert hypertexte)

IGN = Institut National de l'Information Géographique et Forestière

INPN = Inventaire National du Patrimoine Naturel

JSON = JavaScript Object Notation (notation objet issue de JavaScript)

MNHN = Muséum National d'Histoire Naturelle

ORM = Object Relational Mapping (mapping objet-relationnel)

PACA = Provence-Alpes-Côte d'Azur

PNE = Parc National des Écrins

PNR = Parc Naturel Régional

REST = Representational State Transfer

SI = Système d'Information

SIG = Système d'Information Géographique

SINP = Système d'Information sur la Nature et les Paysages

SQL = Structured Query Language (langage de requête structuré)

URL = Uniform Resource Locator (localisateur uniforme de ressource)

Glossaire

API REST : Une API est une solution informatique servant de jonction par laquelle un logiciel offre des services à d'autres logiciels. Elle permet en général la communication et l'échange de données et de services entre deux applications. Elle est offerte par une bibliothèque logicielle ou un service web. Concrètement, il s'agit d'un ensemble normalisé de classes, de méthodes, de fonctions et de constantes qui s'utilise via un langage de programmation. Le style d'architecture REST, qui définit un type d'API, repose sur le protocole HTTP, qui détermine la communication entre les différentes parties du web (clients et serveur) à l'aide de requêtes (GET, PUT, POST, DELETE, etc.) et de réponses HTTP.

Atlas de la Biodiversité : Recensement de l'ensemble des espèces sauvages observées sur un territoire, sous la forme de fiches par espèce et par commune.

Back-end : Partie invisible d'une application web, qui comporte la partie serveur de l'application, le code exécuté par le serveur avant l'envoi des données au client, et la base de données.

Base de données NoSQL : Type de bases de données qui ne repose pas sur une architecture relationnelle classique, permettant de lever certaines contraintes dans le stockage et l'interrogation des données.

Donnée Élémentaire d'Échange : Format standard national de données élaboré par le SINP, permettant notamment de rendre les données de biodiversité interopérables.

Effort de prospection : Notion intimement liée à la pression d'observation, permettant de rendre compte des zones territoriales pour lesquelles il est nécessaire de renforcer les prospections par les agents des espaces naturels.

Framework : Ensemble cohérent d'outils et de composants logiciels permettant d'établir les fondations d'un logiciel ou d'une application. Il s'agit d'une plateforme de développement permettant de faciliter et d'uniformiser le travail des développeurs.

Front-end : Interface visible d'une l'application, avec laquelle les utilisateurs interagissent depuis un navigateur.

GitHub : Logiciel de gestion de versions développé par l'entreprise GitHub, utilisé lors de la phase de développement d'une application.

HABREF : Référentiel national du MNHN sur les habitats naturels et les végétations, utilisé dans les systèmes d'information sur la nature.

INPN : Plateforme nationale du SINP permettant la gestion et le stockage de l'ensemble des données de biodiversité du territoire français.

JSON : Format de données textuelles dérivant de la notation des objets du langage JavaScipt. Ce format permet le stockage de l'information de manière structurée. Le format GeoJSON reprend quant à lui les normes du JSON tout en permettant la considération de données spatiales (points, lignes, polygones, chaînes de caractères).

Open source : Désigne un logiciel ou une application soumis aux possibilités de libre utilisation et redistribution, d'accès au code source et de création de travaux dérivés. Les outils concernés sont généralement le fruit d'une collaboration entre programmateurs.

PostGIS : Extension de la base de données PostgreSQL, permettant de stocker et manipuler des objets géographiques (points, lignes, polygones) sous format binaire.

PostgreSQL : Base de données relationnelle libre.

Pression d'observation : Indicateur permettant de rendre compte de la fréquence de prospection d'une zone par les agents des espaces naturels sur une période donnée. Il s'agit d'une notion assez floue qui peut être évaluée, par exemple, par le nombre d'heures d'observations ou le nombre d'observations réalisées sur la zone et la période considérées.

Reporting de données : Opération consistant à faire rapport des données d'un système d'information.

SINP : Organisation collaborative créée en 2005, impliquant le Ministère en charge de l'écologie et les acteurs de la biodiversité, permettant de mener à bien la mission de structuration et de rassemblement des données de biodiversité à l'échelle nationale (intégration à l'INPN), ainsi que leur diffusion. Ce dispositif est décentralisé puisqu'il comporte un système d'information par région.

SQL : Langage informatique permettant d'exploiter les bases de données relationnelles sous forme de requêtes.

TAXREF : Référentiel taxonomique national du MNHN sur la faune, la flore et la fonge. Ce référentiel est utilisé dans les systèmes d'information sur la nature et permet d'associer un identifiant unique (cd_nom) à chaque taxon.

Taxon : Unité quelconque issue de la hiérarchie taxonomique (règne, classe, famille, espèce, etc.). Généralement, ce terme est employé à tort pour désigner l'espèce ou la sous-espèce.

Taxonomie : Science de la classification hiérarchique des organismes vivants, permettant de les regrouper en entités appelées taxons.

Trigger : Objet de base de données permettant de déclencher l'exécution d'une ou plusieurs instructions lorsqu'une ou plusieurs lignes sont insérées, supprimées ou modifiées dans la table à laquelle il est attaché.

Vue matérialisée : Table particulière de base de données permettant de stocker les résultats d'une requête SQL. A la différence d'une vue standard, les données sont enregistrées, ce qui implique que la requête n'est pas exécutée à chaque consultation des résultats.

Introduction

La France compte aujourd'hui dix parcs nationaux, couvrant près de 9,5 % de son territoire. Ces aires d'exception, créées à partir d'espaces terrestres ou maritimes, sont reconnues, aussi bien à l'échelle nationale qu'internationale, en raison de leur patrimoine naturel, culturel et paysager remarquable. Les objectifs de ces institutions sont la gestion et la protection des richesses naturelles (la faune et la flore essentiellement), ainsi que la sensibilisation du public à la découverte et au respect du patrimoine national (Les parcs nationaux de France, 2017).

Les missions de gestion et de protection du milieu naturel amènent les agents des parcs nationaux à collecter des centaines de milliers de données naturalistes : ils réalisent en permanence l'inventaire des espèces se trouvant sur leur territoire. Historiquement, ces données contribuaient à alimenter des études scientifiques et bénéficiaient d'un usage principalement interne aux parcs. Depuis l'instauration de la « charte » des parcs nationaux en 2006, une dynamique d'ouverture et de diffusion des données a vu le jour. Ces données doivent non seulement être mises à disposition des partenaires institutionnels (ministères, régions, etc.), mais également être accessibles au grand public (Lechémia, 2016). Cette démarche de partage des informations a rapidement soulevé la problématique de l'évaluation des données en termes de fiabilité, indispensable pour garantir une utilisation appropriée de celles-ci au regard des usages (INPN, 2016a).

Au Parc national des Écrins (PNE), la concordance de ces obligations avec l'arrivée de nouvelles technologies et la diversification des protocoles scientifiques de suivi des espèces a entraîné des évolutions dans la collecte, l'organisation et le traitement de l'information. Ce Parc de hautes montagnes a ainsi procédé au regroupement des équipes informatique et géomatique de son établissement dans un pôle « Systèmes d'Information » (SI). Les données anciennement collectées sur papier puis saisies manuellement dans des tableurs ont d'abord transité par des bases de données locales à chaque secteur et agrégées régulièrement. Aujourd'hui, ces données sont rigoureusement recueillies de manière numérique et transférées automatiquement dans des bases de données spatiales centralisées (Lechémia, 2016).

En 2012, le Parc national des Écrins a amorcé le développement de son application naturaliste nommée GeoNature. Il s'agit en fait d'un ensemble d'applications web et mobiles servant d'outil de saisie, de gestion et de diffusion des données multi-protocoles concernant la faune et la flore de son territoire. Ce dispositif correspond au Système d'Information Biodiversité du Parc et regroupe l'ensemble des données naturalistes recueillies depuis sa création en 1973.

Depuis le 1er janvier 2017, le réseau des parcs nationaux de France est rattaché à l'Agence Française pour la Biodiversité (AFB), qui dépend du Ministère de la Transition Écologique. Une de ses missions consiste à créer du lien entre les établissements publics des parcs nationaux pour faciliter les échanges et la mutualisation des outils de gestion, de préservation et de valorisation de leur territoire, tout en préservant le caractère unique de chacun (Les parcs nationaux de France, 2017). En accord avec cette démarche de partage des ressources, GeoNature est une application open source et générique, qui permet malgré tout de répondre à des besoins spécifiques grâce à ses multiples modules. Elle a notamment été déployée au sein d'autres parcs nationaux, comme ceux du Mercantour, des Cévennes, de la Vanoise et de la Guyane. Aujourd'hui, le PNE joue un rôle précurseur dans l'élaboration d'outils informatiques liés à la gestion de la biodiversité et du patrimoine. Il élabore des outils en considérant ses propres besoins mais également ceux d'autres organismes impliqués dans la protection de la biodiversité, et plus particulièrement les attentes du réseau des parcs nationaux.

Une base de données GeoNature peut ainsi rassembler des données provenant de protocoles et sources variés. Les données brutes récoltées sont donc potentiellement très hétérogènes et peuvent représenter un volume conséquent d'informations. Ainsi, il s'avère difficile pour les utilisateurs de garder une vision globale des données et d'en tirer les conclusions nécessaires. De plus, la validation scientifique, qui permet d'évaluer la fiabilité des données, implique aujourd'hui un processus de vérification manuelle chronophage en raison des gros volumes d'informations impliqués, ce qui peut constituer une barrière à leur valorisation.

Comment synthétiser les données brutes hétérogènes de l'application naturaliste GeoNature et optimiser leur valorisation pour répondre aux besoins de l'ensemble des utilisateurs, tout en conservant l'aspect générique de l'outil ?

Le rapport permettra de répondre à cette problématique en trois temps. Tout d'abord, après une présentation de GeoNature, une analyse des besoins sera dressée concernant la valorisation des données de l'application. Cette analyse sera guidée selon deux axes : la création d'un module de reporting de données et la mise en place d'un système de détection automatique de données naturalistes atypiques. Dans un second temps, les états de l'art réalisés en réponse à cette collecte des besoins seront présentés, ainsi que le déroulement de la sélection de solutions adéquates pour le PNE. Enfin, la troisième partie détaillera les travaux techniques qui en ont résulté ainsi que leurs perspectives d'évolution et d'amélioration. Pour chacune des trois parties qui viennent d'être énoncées, une section décrivant la méthodologie adoptée sera d'abord explicitée. Les résultats qui en découlent seront ensuite exposés dans une seconde section.

I. Une application naturaliste à l'origine d'un besoin de synthétisation et de validation scientifique de données

1. GeoNature, une application open source et générique de gestion de données naturalistes

a) Une application mûrie pendant plusieurs années permettant de saisir, organiser et visualiser des données de faune et de flore

Dans les années 2000, le pôle SI du Parc national des Écrins a amorcé un travail de réflexion au sujet des besoins liés à la gestion des données naturalistes. Dans un contexte de diversification des protocoles scientifiques, de diffusion des données et d'innovations informatiques, notamment avec l'arrivée de l'extension *PostGIS* pour *PostgreSQL*, la volonté de créer une base de données spatiale multiple a émergé. Au fil des années, plusieurs petites applications ont été développées en interne pour répondre aux différents protocoles de suivi des espèces, chacune contenant son propre modèle de données. La totalité des informations était agrégée dans une base de données plus globale appelée « synthèse faune-flore ». Peu à peu, un système d'information structuré a vu le jour. Le développement d'une application baptisée GeoNature a ainsi été initié en 2012 par les parcs nationaux des Écrins et des Cévennes.

En plus des protocoles particuliers de suivi de certains taxons, les agents notent quotidiennement les espèces qu'ils observent de manière aléatoire sur le territoire du Parc. Un inventaire des données informatisées, réalisé en janvier 2019 par le PNE, a permis de lister 1 875 464 données de biodiversité (faune et flore), représentant plus de 5 000 espèces différentes observées sur le territoire des Écrins (Maillard, 2019). GeoNature permet de saisir ces observations, de les stocker de manière organisée dans une base de données, et de les visualiser dans un module « Synthèse » sous la forme suivante : « tel observateur a aperçu telle espèce, à tel endroit, à tel moment, à l'aide de tel protocole ».

b) Une application complexe, à la fois adaptée aux standards nationaux et personnalisable par chaque organisme utilisateur

GeoNature constitue un système d'information complet pour la gestion des données faune et flore d'une structure, assurant tout un panel de fonctionnalités nécessaires à leur traitement. Pour cela, l'application est composée de plusieurs outils, présentés dans le Tableau 1, qui peuvent être installés indépendamment :

Tableau 1 : Liste et description des outils prenant part au projet GeoNature

Outil	Fonctionnalités	Spécificités	Utilisateurs
GeoNature-mobile	Saisie des données dans différents protocoles	Saisie mobile	Internes
		Saisie web	Internes
GeoNature	Intégration de données de partenaires		Administrateurs
	Gestion des métadonnées		*Automatique*
	Synthétisation des données des différents protocoles sous forme de Données Elémentaires d'Echange (DEE)		*Automatique*

	Export des données selon les formats attendus par chaque partenaire		Internes
GeoNature-Atlas	Diffusion des données sur un portail web grand public	Sous la forme d'un Atlas de la biodiversité (fiches espèces)	Internes, grand public
TaxHub	Gestion des référentiels	Gestion de la taxonomie à partir de TAXREF	Administrateurs principalement
UsersHub		Gestion des utilisateurs et droits	Administrateurs

Pour répondre aux enjeux de spécialisation des protocoles couplés au besoin croissant d'harmonisation des données, tous les outils de GeoNature ont été reliés à une base de données unique construite sur deux niveaux :

- Un modèle de données par protocole de saisie pour répondre à chaque question scientifique.
- Un modèle de données « synthèse » pour agglomérer les données provenant des différents protocoles et sources et ainsi disposer d'une vue d'ensemble de la connaissance du territoire, facilement diffusable pour des partenaires et le grand public.

Ainsi, un protocole de saisie particulier correspond à un module, ce qui équivaut concrètement à une « sous-application » de GeoNature connectée à un schéma (modèle de données) spécifique dans la base de données. Les données de chaque module sont également intégrées au schéma « synthèse » sur la base des champs communs à tous les protocoles (observateur, espèce observée, lieu, date, nom du protocole) grâce à des triggers et des fonctions SQL. L'organisation des données a aussi été conçue pour autoriser l'intégration régulière de données de partenaires, qui constituent un apport essentiel à la connaissance de la biodiversité du territoire.

L'architecture complète de GeoNature telle qu'elle est développée au PNE est exposée dans la figure ci-dessous (Figure 1). La liste des modules présents n'est pas exhaustive.

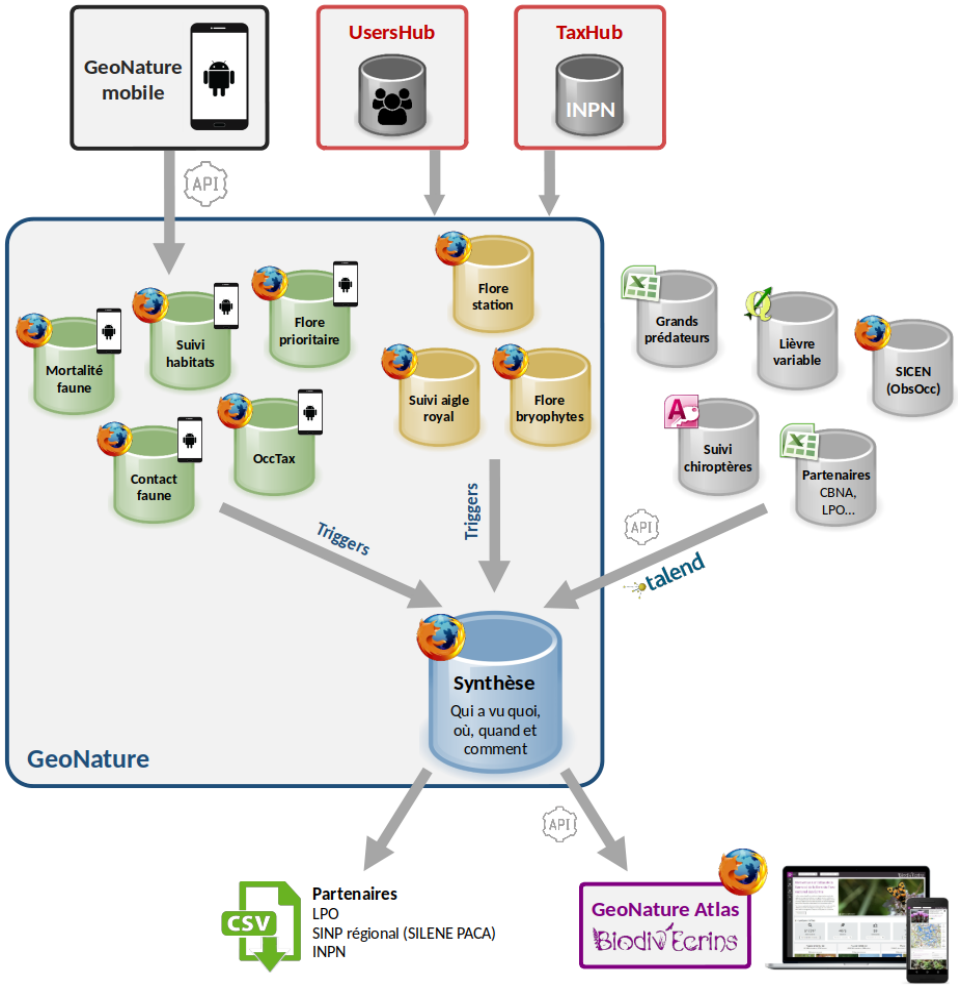


Figure 1 : Schéma de l'architecture globale de l'application GeoNature

GeoNature est construite dans une logique de système d'information s'appuyant sur les référentiels et les standards nationaux, tels que le référentiel taxonomique TAXREF du Système d'Information sur la Nature et les Paysages (SINP) ou les données géographiques de l'Institut National de l'Information Géographique et Forestière (IGN), afin de les intégrer dans une chaîne de travail complète, allant de la collecte des données jusqu'à leur partage et leur diffusion aux échelles locale, régionale et nationale (Figure 2).

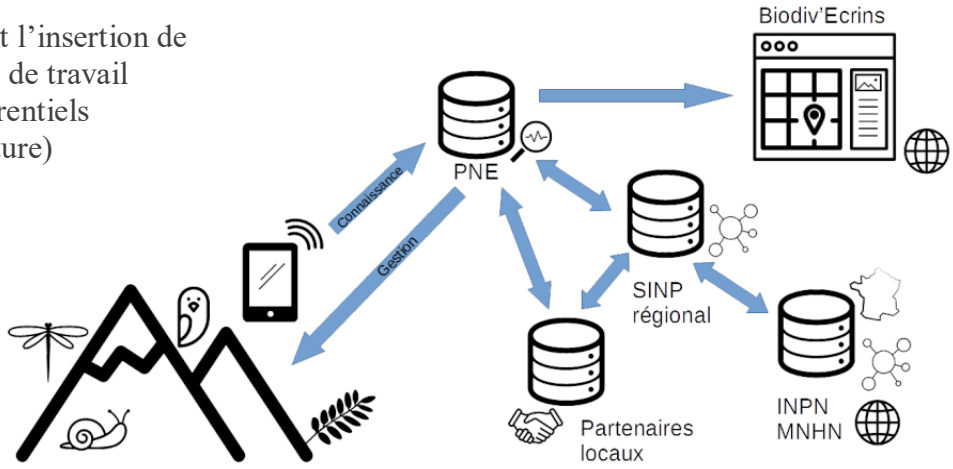


Figure 2 : Schéma décrivant l'insertion de GeoNature dans une chaîne de travail complète, intégrant les référentiels nationaux (source : GeoNature)

c) Une application basée sur des technologies open source

En 2016, le modèle d'organisation des données et les outils de GeoNature ont été retenus par les groupes scientifiques et géomatiques inter-parcs comme système de gestion commun à tous les parcs nationaux français. Cet outil open source a ainsi été déployé dans d'autres parcs nationaux, mais également dans des Parcs naturels régionaux (PNR), des Conservatoires d'Espaces Naturels (CEN), des conservatoires botaniques nationaux et des associations telles que la Ligue pour la Protection des Oiseaux (LPO). Une refonte des technologies utilisées au sein de GeoNature, pilotée par le PNE, a alors été amorcée en 2017 pour rendre l'outil encore plus générique, modulaire, standard et moderne. L'architecture technique actuelle de GeoNature 2 est présentée dans la Figure 3.

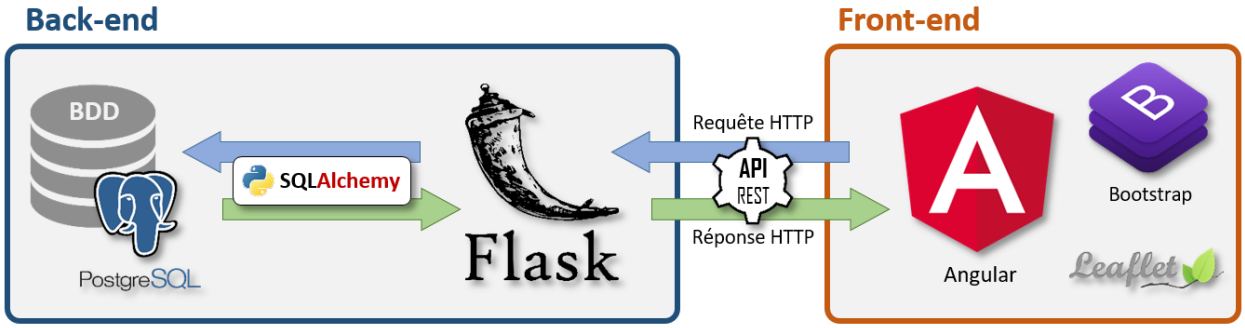


Figure 3 : Schéma de l'architecture technique de l'application GeoNature

Le développement de l'application comporte deux niveaux :

- Le front-end : il s'agit de la partie visible de l'application, avec laquelle les utilisateurs interagissent depuis un navigateur (interface). Pour cela, GeoNature utilise le framework de programmation *Angular*, permettant de créer des applications interactives, et basé sur les langages TypeScript (sur-ensemble de JavaScript), HTML et CSS. L'application fait également appel à la librairie CSS *Bootstrap* pour la création du design et à la librairie JavaScript *Leaflet* pour l'intégration de cartes interactives. Tous ces outils sont open source.
- Le back-end : il s'agit de la « partie immergée de l'iceberg », qui comporte la partie serveur de l'application, le code exécuté par le serveur avant l'envoi des données au client, et la base de données. Le framework open source utilisé pour interagir avec la base de données est *Flask*, écrit en Python (Blondin, 2013).

d) Une application en pleine expansion à l'échelle nationale

Le projet GeoNature a entraîné un engouement important et la formation d'une communauté d'utilisateurs et de développeurs (animée par le PNE) provenant de différentes structures utilisant l'application, et mutualisant leurs ressources autour de ce logiciel libre. Par ailleurs, certains modules ont été développés par d'autres structures que le PNE, à l'image du module « Validation OneByOne » qui est actuellement en cours d'élaboration par l'association *Picardie Nature*.

Depuis la loi pour la reconquête de la biodiversité, de la nature et des paysages promulguée en 2016, les bureaux d'études environnementaux sont contraints de rendre publiques les données générées lors de leurs études d'impact. Ces données doivent donc être remontées à l'INPN en répondant aux formats standards du SINP. GeoNature a été sélectionnée pour aider les bureaux d'études à réaliser cette tâche. Ainsi, une instance nationale de l'application est actuellement déployée par le PNE, le Ministère de la Transition Écologique, le Muséum National d'Histoire Naturelle (MNHN), l'Agence Française pour la Biodiversité (AFB) et l'IGN pour proposer aux maîtres d'ouvrage un outil de saisie des données brutes de biodiversité (Monchicourt, 2018).

2. Une nécessité de synthétisation et de valorisation des données brutes de GeoNature

a) Un module « Tableau de bord »

Les données intégrées à la « Synthèse » de GeoNature peuvent provenir de divers modules de saisie et partenaires. Ces données sont donc variées et hétérogènes, en termes de périmètres taxonomiques et géographiques par exemple. Dans le cadre de la refonte de l'application vers une version 2, la création d'un module « Tableau de bord » a été envisagée, l'objectif étant de fournir une vision d'ensemble des données de la base à travers des graphiques et des cartes appropriés qui répondent aux besoins des utilisateurs.

Une telle démarche de synthétisation des données avait déjà été entreprise dans la version 1 de GeoNature à travers la représentation de statistiques simples, telles que des histogrammes du nombre total d'observations selon les années, selon les différents groupes taxonomiques ou selon les différents programmes scientifiques (Annexe 1). Ces graphiques avaient été implémentés sans réelle consultation au préalable de la communauté. La mission de stage a donc consisté à :

- Prolonger ce travail en réalisant une analyse des besoins complète, afin que les représentations graphiques et cartographiques soient génériques, en accord avec les attentes de tous les utilisateurs internes au PNE, mais également adaptées au grand public. En effet, le nouveau module sera, à termes, accessible à tous sans authentification dans une optique de sensibilisation et de diffusion.
- Réaliser un état de l'art des outils existants pour répondre à ces besoins et déterminer la solution la plus adaptée.
- Développer une première version fonctionnelle de ce module avec la mise en place de quelques graphes et cartes correspondant aux besoins identifiés.

b) Un système de détection automatique des données naturalistes atypiques

Depuis une dizaine d'années, la tendance nationale est à la valorisation et la diffusion auprès du grand public des données de biodiversité. En effet, la méconnaissance de l'état et de la localisation des richesses naturelles entraîne l'installation d'aménagements souvent néfastes pour la biodiversité. De ce fait, il semble indispensable de diffuser ces données naturalistes afin que celles-ci soient considérées dans les projets de conservation, par la recherche et pour l'information du public (INPN, 2016c).

Le SINP a été créé en 2005 pour favoriser la collaboration entre les différents producteurs de données de biodiversité et pour encadrer le partage et l'accès libre à ces données. Cette organisation a détecté plusieurs enjeux liés à la diffusion et l'utilisation des données, qui nécessitent la mise en place de traitements et de contrôles de celles-ci avant leur intégration à l'INPN :

- Conformité de la donnée : une nouvelle donnée doit respecter les règles fixées par les formats standards de données et de métadonnées. Doivent être contrôlés, en l'occurrence : le renseignement des champs obligatoires, le format, l'utilisation des référentiels de nomenclatures et de valeurs, etc.
- Cohérence de la donnée : les informations transmises au travers d'une nouvelle donnée et de ses métadonnées doivent respecter une certaine logique combinatoire. Par exemple, une date d'observation est obligatoirement inférieure à la date du jour.
- Validation scientifique de la donnée : une nouvelle donnée doit être soumise à des processus de vérification dans le but de déterminer son degré de confiance. Par exemple, si une espèce est observée en dehors de son aire de répartition connue, il est possible qu'une erreur d'identification ou de saisie ait été commise. Ce niveau de fiabilité peut être estimé selon des bases de connaissance ou l'expertise de naturalistes. Le SINP a retenu les niveaux de validité suivants pour les données naturalistes : « certain – très probable », « probable », « douteux », « invalide », « non réalisable », « non évalué » (INPN, 2016a).

L'application GeoNature assure la conformité et la cohérence des données du PNE avec les standards du SINP au moyen de formulaires de saisie, comportant essentiellement des listes déroulantes et des champs de recherche à choix imposés. De plus, certaines informations ne sont jamais demandées à l'utilisateur mais uniquement calculées à partir d'autres données, telles que l'altitude d'observation qui est déterminée à partir de la localisation, ce qui limite les sources d'erreurs.

En revanche, l'application ne présente pas de processus de validation scientifique des données. En général, cette étape est réalisée de manière manuelle. Dans ce cas, elle se révèle très chronophage puisque les validateurs sont contraints de considérer les données une par une. Ainsi, certaines données sont validées plusieurs années après leur date de production, ce qui constitue une barrière à leur utilisation et leur diffusion. Le service scientifique du Parc, conscient de l'intérêt d'évaluer la pertinence de ses données, souhaiterait mettre en place dans GeoNature un système de détection automatique de données atypiques afin de concentrer les efforts de validation manuelle sur ce type d'informations, et ainsi accélérer l'étape de validation scientifique. Pour ce service, une donnée atypique correspond à une donnée dont les informations sont inhabituelles, ne correspondant pas à la moyenne des données ordinairement récoltées pour le taxon concerné. A titre d'exemple, une donnée d'espèce observée hors de sa zone de répartition connue pourrait correspondre à une erreur de saisie ou d'identification, et mériterait une attention particulière. De plus, dans le contexte actuel de changement climatique, les espèces ont tendance à s'adapter et à évoluer. De ce fait, les données atypiques peuvent également être reliées à ces changements, et non à des erreurs, et leur détection permettrait d'alimenter les études à ce sujet.

Ainsi, cette deuxième mission de stage a consisté à :

- Récolter les avis et les attentes de certains utilisateurs de GeoNature au sein du service scientifique du PNE concernant la mise en place d'un protocole automatique de détection de données atypiques. En raison de la courte période de temps accordée à cette mission, aucune analyse des besoins complète n'a pu être menée.
- Réaliser un état de l'art des solutions existantes relatives à cette thématique, ou plus généralement aux processus automatiques de validation scientifique de données naturalistes.
- Proposer des pistes de travail pour l'implémentation d'un tel système au sein de GeoNature à l'aide des informations récoltées lors de la phase précédente, le but étant de détecter les méthodes les plus pertinentes et les plus adaptées aux attentes du service scientifique.

Le schéma suivant résume la méthodologie qui a été employée pour répondre à chacune des deux missions (Figure 4) :

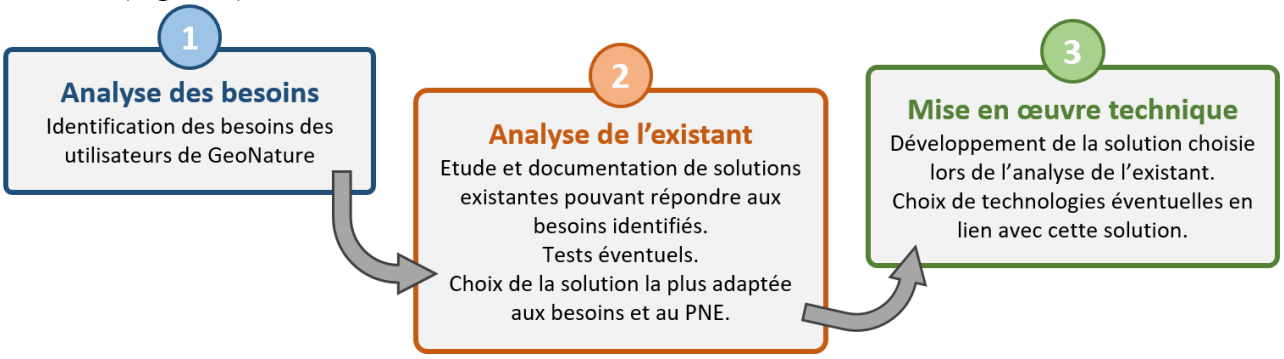


Figure 4 : Schéma de la méthodologie générale adoptée pour répondre aux missions du stage

3. Des besoins identifiés selon plusieurs sources

a) Une analyse des besoins complète pour la mise en place d'un module de reporting de données

Les sections qui suivent exposent les différentes étapes méthodologiques qui ont été adoptées pour réaliser l'analyse des besoins complète.

Identification des utilisateurs

L'identification des différents types d'utilisateurs de GeoNature et de leurs interactions avec l'application a constitué la première étape de l'analyse des besoins. A l'échelle du PNE, un petit groupe de travail, nommé « données faune flore » et composé d'agents utilisant GeoNature, avait été formé dans le but de suivre le projet et d'alimenter les réflexions à ce sujet. Dans ce groupe ont été identifiés trois types de profil : des gardes moniteurs, des techniciens patrimoine et des chargés de mission scientifiques. Il faut savoir qu'un parc national est divisé en plusieurs secteurs géographiques en raison de l'ampleur de son territoire. Sur chaque secteur, on trouve un chef de secteur, qui encadre le personnel et coordonne les actions, un technicien patrimoine, qui s'occupe principalement de la gestion de la faune et de la flore (soutien au chef de secteur), et des gardes moniteurs. La démarche empruntée a consisté à interviewer, au sein de ce groupe de travail, au minimum une personne par type de profil, afin de comprendre, dans un premier temps, son métier et les actions qu'elle réalise sur GeoNature.

Recueil des besoins des utilisateurs

Dans un second temps, les besoins de ces utilisateurs ont été récoltés. Pour ce faire, le travail s'est orienté selon quatre axes :

- Le recueil de l'opinion de mon maître de stage : Camille MONCHICOURT, en tant que chef du pôle SI, possède une vision d'ensemble de GeoNature et de ses utilisateurs. Il a donc un avis précis sur la nécessité d'un tel module.
- Le recueil des propositions de la communauté GeoNature : les développeurs participant à l'amélioration de l'application sur le GitHub de GeoNature postent régulièrement des tickets concernant de nouveaux besoins.
- Le recueil des besoins des utilisateurs directs : cette collecte s'est organisée à travers des entretiens directs ou téléphoniques avec les personnes ayant participé à l'identification des utilisateurs (groupe de travail « données faune flore »). Un questionnaire a été élaboré préalablement, découlant des résultats de l'identification des utilisateurs.

- La prise en compte des besoins présumés du grand public.

Hiérarchisation des besoins identifiés

Afin de hiérarchiser les besoins lors de leur recueil, une note leur a été attribuée. Cette note est fonction du nombre de personnes qui ont énoncé le besoin et de leur profil. A chaque expression d'un besoin particulier, la note de ce dernier est incrémentée d'un certain nombre de points selon les règles détaillées dans le tableau suivant (Tableau 2) :

Tableau 2 : Système de hiérarchisation des besoins des utilisateurs de GeoNature

Provenance du besoin			Incrémentation de la note (points)
Besoin exprimé par le maître de stage			+1
Besoin exprimé par la communauté GeoNature			+1
Utilisateurs directs	Besoin exprimé par un utilisateur	1ère priorité	+4
		2ème priorité	+3
		3ème priorité	+2
		Pas de priorité	+2
	Besoin présumé du grand public		+3

Les notions de priorité concernant les besoins émis par les utilisateurs directs découlent du questionnaire qui leur a été soumis. En effet, certaines questions requièrent de fournir 2 à 3 idées classées dans un ordre de priorité.

b) Une analyse des besoins partielle concernant la validation scientifique des données

Cette thématique n'a pas pu être traitée de manière aussi poussée que la précédente, par manque de temps. Un aperçu des besoins du service scientifique a pu être dressé à travers l'approfondissement de deux axes :

- Le recueil des attentes des naturalistes du service scientifique : il s'agit d'experts concernant la fiabilité des données. Eux seuls possèdent les connaissances scientifiques nécessaires pour juger la validité d'une observation. Ces personnes ont été interviewées en direct.
- Analyse des travaux déjà réalisés au sein du pôle SI : avant le commencement du stage, le pôle SI du Parc avait déjà profité de temps de travail pour réfléchir à une validation automatique des données. Ces travaux ont été recueillis et analysés.

Ces besoins n'ont pas pu être classés ni hiérarchisés par la suite.

4. Des besoins à la fois communs et spécifiques aux différents types d'utilisateurs

a) Un module « Tableau de bord » nécessitant de la flexibilité pour répondre à l'ensemble des besoins recueillis

Des utilisateurs aux profils variés qui ne présentent pas le même rapport à la donnée

Le nombre de personnes concernées par l'enquête visant à identifier les types d'utilisateurs de GeoNature s'est élevé à quatre, ce qui est faible mais la charge de travail des agents des parcs nationaux étant très importante, ces personnes avaient très peu de temps à consacrer au projet. J'ai

ainsi pu m'entretenir avec un garde moniteur du secteur du Champsaur (Marc CORAIL), un technicien patrimoine du secteur de l'Embrunais (Michel BOUCHE) et deux chargés de mission, vertébrés et invertébrés, travaillant au siège du PNE (Ludovic IMBERDIS et Damien COMBRISSON), du groupe de travail « données faune flore ». Les informations récoltées sont résumées dans le tableau suivant (Tableau 3):

Tableau 3 : Bilan de l'enquête ciblant l'identification des différents types d'utilisateurs de GeoNature

Poste	Description métier	Interactions avec GeoNature		
		Saisie de données	Consultation de données	Export de données
Garde moniteur	- Faire appliquer la réglementation : constatation des infractions des visiteurs et dressage de procès-verbaux. - Accueil, accompagnement, information et sensibilisation auprès du public (guide-animateur). - Entretien des équipements du Parc : refuges, sentiers de randonnée… - Recensement de la faune et de la flore à travers une contribution aux travaux de collecte de données (inventaires, comptages, protocoles de suivis scientifiques) pour enrichir la connaissance des milieux et des espèces (Les parcs nationaux de France, 2014).	Tous les jours	Souvent	Occasionnel
Technicien patrimoine	- Assurer l'animation de tous les protocoles scientifiques : planifier les comptages/opérations, les mettre en place, y participer, synthétiser les résultats pour les remettre au service scientifique. - Gestion de l'espace à travers la participation à des programmes en collaboration avec *Natura 2000* par exemple, ou à des mesures agro-environnementales en lien avec les communes. - Missions de garde moniteur. - Missions transversales à l'échelle du Parc (capture et parcage du Bouquetin des Alpes, suivis sanitaires…).	Occasionnel	Souvent (porter à connaissance)	Souvent
Chargé de mission	- Coordination et mise en place du suivi des espèces : protocoles de collecte de données. - Traitement/analyse des données récoltées et rédaction de rapports. - Mise en place de mesures de protection.	Occasionnel (été)	Souvent	Souvent

Les différents types d'utilisateurs internes au PNE n'interagissent pas de la même manière avec les données naturalistes : certains les récoltent, d'autres les consultent pour orienter leur travail, d'autres les exportent pour les analyser sur des logiciels de traitement tels que R et QGis, et d'autres

encore réalisent ces trois actions à la fois à des fréquences distinctes. Sachant que le module « Tableau de bord » doit également être adapté au grand public, ces indications permettent de présupposer que les attentes des différents utilisateurs concernant cet outil ne vont pas être les mêmes, et que ce dernier nécessitera de préserver à la fois les aspects générique et spécifique propres à l'application.

Des besoins à la fois communs et spécifiques

Recueil de l'opinion du maître de stage

Selon lui, les agents qui déposent des données sur GeoNature ont besoin d'un retour direct sur ces données, d'avoir une vision d'ensemble et concrète de la base. Cela pourrait les influencer par exemple pour aller prospecter des zones où certaines espèces ont été peu observées.

En outre, certains jeux de données de l'INPN sont accompagnés d'analyses statistiques représentées sous forme de graphiques (Annexe 2). Les graphes à implémenter sur le module « Tableau de bord » pourraient s'inspirer de ces exemples intéressants.

Recueil des propositions de la communauté GeoNature

Certains développeurs ont manifesté, sur le GitHub de GeoNature, leur volonté de mettre en évidence la pression d'observation et donc l'effort de prospection pour chaque espèce au sein de GeoNature-Atlas. Cet outil étant majoritairement destiné au grand public, ces idées sont plutôt vouées à être mises en place dans le module « Tableau de bord » qui sera davantage professionnel.

Donovan MAILLARD, anciennement chargé de mission invertébrés au PNE, a beaucoup pris part au projet GeoNature lors de ses fonctions au Parc. Il s'agit, encore aujourd'hui, d'un membre particulièrement actif de la communauté GeoNature. C'est pourquoi un entretien téléphonique a été conclu avec lui afin de récolter son avis sur l'intérêt d'un tel module ainsi que ses suggestions de contenus à inclure. Ces informations sont considérées dans la partie regroupant tous les besoins.

Recueil des besoins des utilisateurs directs

Le questionnaire complet soumis aux utilisateurs du groupe de travail « données faune flore » est présent en Annexe 3. Les questions posées ont été orientées vers la conception d'un module supposé très flexible avec la mise en place de nombreux filtres, afin de prendre en compte les profils variés des utilisateurs. Ces informations sont considérées dans la partie regroupant tous les besoins.

Prise en compte des besoins présumés du grand public

Le grand public représentant un type d'utilisateur à part entière du futur module « Tableau de bord », il est nécessaire de considérer ses besoins. Une étude complète n'a pas pu être menée faute de temps, mais il a été supposé que ces utilisateurs seraient davantage intéressés par une interface simple d'utilisation, esthétique et interactive. Les connaissances en biodiversité pouvant varier considérablement d'un utilisateur à l'autre pour ce type de profil, il a également été convenu que l'outil devrait être particulièrement intelligible, avec l'insertion d'encarts explicatifs concernant certains termes et contenus scientifiques.

Regroupement des besoins identifiés

Les besoins identifiés ne relèvent pas tous du même niveau de précision, c'est pourquoi ils ont été classés selon trois catégories : les besoins relatifs aux objectifs du module, les besoins énonçant des représentations graphiques ou cartographiques précises, et les besoins concernant les

filtres à mettre en place. Le Tableau 4 présente ces besoins hiérarchisés par catégorie selon leur note finale. Le détail du calcul des notes attribuées se trouve en Annexe 4.

Tableau 4 : Bilan des besoins identifiés lors de l'analyse des besoins relative au module « Tableau de bord »

Besoin identifié		Note
Objectifs généraux du module		
Visualiser la pression d'observation pour orienter les efforts de prospection		13
Obtenir un état des lieux des connaissances présentes dans la base de données : combien d'espèces sont présentes, lesquelles précisément et où ?		12
Obtenir des informations sur les espèces patrimoniales (protégées, menacées, rares ou ayant un intérêt scientifique ou symbolique (INPN, 2016b))		9
Pouvoir faire du reporting auprès des communautés, des collègues et du grand public avec un module ergonomique, interactif et pédagogique		9
Exporter certains graphiques sous format Excel avec les données brutes		3
Identifier les programmes et cadres d'acquisition à dynamiser		1
Propositions précises de représentations et statistiques à implémenter		
Carte de présence/absence (et classes intermédiaires) d'une espèce par zonage (pour une échelle donnée)		12
Carte du nombre d'observations et du nombre de taxons par zonage (pour une échelle donnée)		9
Graphiques réalisés dans GeoNature 1 et sur le site de l'INPN : camembert de la répartition des observations par grands groupes		5
Nombre de nouvelles espèces observées chaque année par grand groupe		3
Carte du statut de reproduction (nicheur certain, nicheur probable, nicheur possible) par maille		3
Moyennes d'altitude en fonction du temps pour une espèce		2
Échelles permettant d'ajuster les représentations graphiques et cartographiques		
Échelle spatiale	Pouvoir visualiser des données seulement pour des zones précises : mailles de différentes tailles (100mx100m, 1km^2, 5km^2, 10km^2, etc.), sites définis par l'administrateur, communes.	9
Échelle temporelle	Pouvoir visualiser des données seulement pour une échelle de temps précise : mois, année, décennie.	8
Taxonomie	Pouvoir visualiser des données seulement pour un rang taxonomique précis : espèce, famille, groupe INPN, etc.	8

Les exigences énoncées par les différents utilisateurs sont globalement propres à chacun, mais il a été possible de les regrouper sous la forme de besoins plus globaux. Cela souligne une nouvelle fois la nécessité de développer un outil à la fois générique et modulable pour chaque utilisateur. Le module « Tableaux de bord » doit donc être constitué d'une interface interactive et flexible proposant divers filtres afin de simplifier l'ajustement des graphes selon les besoins de chacun.

Ces résultats démontrent également que les attentes des utilisateurs internes au PNE sont plus poussées et plus variées que les possibilités offertes par GeoNature 1. En effet, les graphiques instaurés dans cette version paraissent adéquats mais nécessitent de pouvoir être adaptés à différents

niveaux d'échelles, particulièrement les échelles temporelle et taxonomique. De plus, un volet cartographique semble indispensable pour mettre en évidence la pression d'observation et la présence d'espèces particulières dans certaines régions.

b) Un système automatique pour la validation des données ayant déjà été alimenté par des réflexions sur les plans théorique et technique

Recueil des attentes des naturalistes du service scientifique du PNE

La collecte de ces besoins a été réalisée à travers les interviews de deux naturalistes utilisant GeoNature, ayant participé par ailleurs à l'alimentation des réflexions concernant le module « Tableau de bord ». Il s'agit de Ludovic IMBERDIS, chargé de mission vertébrés, et Michel BOUCHE, technicien patrimoine, travaillant au PNE. Leurs avis étant très similaires, il a été possible de les regrouper et les synthétiser.

Il serait nécessaire de faire ressortir toutes les observations atypiques saisies dans la base de données. Pour cela, l'analyse doit se centrer au niveau de l'espèce et considérer trois échelles :

- Calendaire : Une donnée doit être mise en valeur si la date d'observation diffère de celles des autres données déjà récoltées pour le taxon concerné.
- Géographique : Une donnée doit être mise en valeur si la zone d'observation diffère de celles des autres données déjà récoltées pour le taxon concerné.
- Altimétrique : Une donnée doit être mise en valeur si la plage d'altitude de l'observation diffère de celles des autres données déjà récoltées pour le taxon concerné.

Un avertissement pourrait être affiché à l'écran lors de la saisie d'une observation anormale par un utilisateur. Ce dernier serait libre d'en tenir compte en corrigeant les informations, ou non. Afin de vérifier la fiabilité des données dès leur production, il serait également intéressant de mettre en place un système d'alerte (notification sur l'application ou mail) destiné à un réseau de validateurs, déclenché lors de la détection d'une donnée hors normes. Ces validateurs pourraient par la suite avoir accès aux éléments précis de la donnée et déterminer manuellement le type d'erreur, si nécessaire. Il est très important de ne pas supprimer les données identifiées comme atypiques. En effet, en raison du changement climatique, des observations dîtes exceptionnelles se produisent de manière plus régulière, comme la présence d'oiseaux américains en Europe.

Par ailleurs, afin de détecter plus facilement les adaptations comportementales des espèces dues au climat, il serait notamment pertinent de mettre en évidence les observations pour lesquelles les informations sont situées dans les valeurs extrêmes (5%) des données d'altitude et de période d'observations habituelles pour un taxon donné. Ces données remarquables pourraient être soulignées à travers un autre type d'alerte.

Analyse des travaux déjà réalisés au sein du pôle SI du PNE

GeoNature ne comportant pas de jeu de données identifiées comme erronées, il avait été convenu par le pôle SI du Parc que les observations saisies devraient plutôt être comparées aux données qualifiées comme correctes. Pour ce faire, un « profil type » par taxon avait été amorcé en base de données. Cela a été réalisé à travers une vue matérialisée fournissant différents éléments pour chaque taxon (une entrée par taxon) sur la base des données de la « Synthèse » : zone de répartition estimée avec des fonctions de calcul spatiales, jours minimal et maximal d'observation dans l'année, altitudes minimale et maximale d'observation, nombre total d'observations. Le travail doit être complété et amélioré, mais ces « fiches d'identité » par espèce servant de référence permettraient de mettre en évidence les données hors normes. En outre, certaines données anciennes sont très peu précises et peuvent fausser les résultats. Par exemple, certaines observations sont seulement associées

à une année et non à une date exacte. Il serait nécessaire d'implémenter des seuils de précision pour exclure ces données dans l'élaboration des « profils types ».

Une autre démarche intéressante consisterait à retourner un pourcentage de confiance ou une note de validité pour chaque donnée saisie, étant donné que le PNE n'a pas paramétré de niveaux de validation dans GeoNature. Les critères pris en compte dans ce calcul pourraient être associés à des poids différents, selon leur importance. Ils pourraient notamment concerner, par analogie avec les « fiches d'identité », la date d'observation, la localisation de l'observation, l'altitude de l'observation, l'observateur.

Bilan des besoins

L'idéal pour le PNE serait de pouvoir combiner l'élaboration d'un profil type par espèce et l'attribution d'un pourcentage de confiance pour chaque donnée. Il serait possible, par exemple, de comparer chaque nouvelle donnée au profil type du taxon concerné, et de déterminer un pourcentage de correspondance. Les données ne seraient pas supprimées quel que soit le résultat obtenu, mais les notes calculées seraient associées aux observations dans la base de données et visualisables sur GeoNature. Un système d'alerte, déclenché lors de la saisie mais aussi après soumission de la donnée par l'observateur, permettrait d'informer les utilisateurs sur le doute détecté.

Les réflexions sont déjà bien amorcées, mais elles nécessiteraient d'être davantage cadrées et complétées par une analyse de l'existant. En effet, des méthodes tout à fait différentes pourraient également s'avérer efficaces et pertinentes pour le PNE.

L'analyse des besoins complète concernant la mise en place d'un module « Tableau de bord » a permis de dresser une liste de critères et caractères indispensables à implémenter dans le module : reporting de données, état des lieux des connaissances, interactivité, flexibilité, filtres, graphiques, cartes. La deuxième partie sera donc consacrée à une analyse de l'existant ciblant les outils permettant de répondre à ces exigences.

L'analyse des besoins partielle relative à l'élaboration d'un processus automatique de détection de données naturalistes atypiques a permis quant à elle d'identifier les attentes des naturalistes utilisant GeoNature. La combinaison de ces avis avec les travaux déjà amorcés à ce sujet par le pôle SI ont abouti aux idées suivantes : profil type, pourcentage de confiance, paramétrage scientifique, système d'alerte. La deuxième partie permettra ainsi de prospecter les protocoles automatiques de validation scientifique déjà conçus et de compléter les idées listées ou bien de les réorienter.

II. Une analyse de l'existant proposant des outils « clé en main » pour la création de tableaux de bord et des protocoles variés de validation automatique de données pour GeoNature

1. Des états de l'art destinés à explorer les fonctionnalités offertes par des solutions susceptibles de répondre aux besoins identifiés

a) Un module « Tableau de bord » pouvant être développé en interne ou élaboré à partir d'une solution open source existante

Les modules GeoNature existants ont tous été développés avec les mêmes technologies que l'application elle-même. Certains sont incorporés au cœur de GeoNature, c'est-à-dire fournis lors de l'installation de l'application, et d'autres doivent être installés indépendamment.

La synthétisation des données sous la forme de tableaux de bord est un enjeu courant au sein des structures publiques et privées qui en génèrent de gros volumes, c'est pourquoi il existe de nombreux logiciels permettant d'atteindre cet objectif.

Le développement du module « Tableau de bord » peut ainsi s'opérer de deux manières :

- Ce module correspondant à un outil de valorisation des données générées et non à un protocole de saisie particulier, il peut être codé intégralement en interne et intégrer directement le cœur de GeoNature, comme expliqué précédemment.
- Ce module peut également être conçu à partir d'une solution déjà existante.

Le recours à la deuxième méthode permettrait potentiellement de gagner un temps considérable en évitant le développement en interne des fonctionnalités nécessaires. Un état de l'art centré sur ce type de solutions a donc été réalisé. Les sections qui suivent décrivent la méthodologie qui a été adoptée pour répondre à cette mission.

Orientation de l'état de l'art

Afin de suivre la même dynamique que GeoNature, je me suis concentrée uniquement sur des outils open source. Mon maître de stage réalise en permanence de la veille technologique au sujet des dernières innovations informatiques liées au développement web, aux technologies open source et aux SIG. Cela lui a permis de détecter trois logiciels open source pouvant répondre aux besoins.

Au vu du temps imparti, seulement deux solutions ont pu être étudiées et testées : *Kibana* et *Metabase*. Après installation et expérimentation de chacun des logiciels (création de graphiques), les solutions ont été comparées au moyen de critères tels que la qualité de la documentation et la facilité d'installation et de prise en main.

Détermination de critères pertinents pour le choix d'une solution

La détermination d'une solution de développement pour le module « Tableau de bord » s'est effectuée par affectation d'un score à chaque outil. L'objectif a été d'orienter la décision vers la solution présentant le score le plus élevé.

Pour ce faire, la démarche s'est opérée en deux temps. Tout d'abord, des critères de sélection ont été établis, illustrant tous les éléments qu'il serait préférable de retrouver dans le futur module. A chaque critère a ensuite été associé un nombre de points fixe ou variable reflétant son poids, c'est-à-

dire son importance dans le processus de détermination de la solution. Plus le critère est nécessaire au module, plus son poids, et donc son nombre de points, doit être important. Les points variables permettent de prendre en compte les différents niveaux de complétude qui peuvent coexister pour un critère (« faible », « moyen », « élevé », « très élevé »). A titre d'illustration, il est possible qu'une solution soit en accord avec l'un des critères de manière plus poussée que ne le sont les autres solutions. Les critères à points fixes n'offrent quant à eux aucune ambiguïté : soit ils se retrouvent dans la solution considérée (« présent »), soit ils ne s'y retrouvent pas (« absent »).

Cette première phase a été réfléchie avec l'aide de mon maître de stage. Les critères qui ont été déterminés ainsi que leur nombre de points respectif sont listés dans le Tableau 5 :

Tableau 5 : Système de notation des solutions existantes pour le module « Tableau de bord »

Critères	**Points à attribuer à la solution selon le critère**					
	Points fixes		Points variables			
	Présent	*Absent*	*Faible*	*Moyen*	*Élevé*	*Très élevé*
Open source	2	0				
Unité avec GeoNature	1					
Facilité de développement			0	1	2	3
Facilité d'installation du module final						
Accessibilité pour tous les types d'utilisateurs	2	0				
Capacité à répondre aux besoins identifiés			0	1	2	3
Facilité de contribution de la communauté (perspectives d'évolution)						
Autres éléments	1	0				

La deuxième étape a consisté à croiser dans un tableau la liste des critères de sélection avec la liste des solutions envisagées, et de déterminer un score total pour chaque outil.

b) Une analyse de l'existant pour la validation automatique de données reposant sur un travail réalisé par le SINP

En 2015, le SINP a formé un groupe de travail « Validation » dans le but de déterminer une démarche commune de validation des données naturalistes pour les opérateurs en charge de cette procédure au sein du SINP même. Son objectif était de pouvoir fournir un niveau de confiance aux différents utilisateurs des données de l'INPN, afin que celles-ci soient utilisées à bon escient en fonction des usages. Pour cela, le SINP a réalisé un recensement de l'existant. Le travail s'est porté sur 17 organismes disposant d'un protocole de validation des données pour leur système d'information interne (Robert, 2015).

Ces protocoles, qui concernaient aussi bien la conformité et la cohérence des données que la validation scientifique, n'étaient pas détaillés dans le document de synthèse rédigé par le groupe de travail. Toutefois, ce document a permis de dresser une liste de systèmes intéressants sur lesquels baser l'état de l'art devant être réalisé pour le PNE. Les organismes ayant des missions et objectifs semblables à ceux du PNE ont été ciblés en priorité. Les recommandations de mon maître de stage et

de mon tuteur école ont complété cette base de travail. Les informations concernant les différents protocoles de validation étudiés ont été récoltées à l'aide de la documentation et d'entretiens.

2. Un module « Tableau de bord » impliquant des besoins précis et variés difficilement accessibles avec des solutions externes

a) *Kibana* et *Metabase* : deux outils open source permettant la création de tableaux de bord interactifs

Kibana d'ElasticSearch

ElasticSearch est un moteur de recherche permettant l'indexation et la recherche de données personnalisées. Ce logiciel, écrit en Java et distribué sous licence Apache, est gratuit, libre et open source. Concrètement, il s'agit d'une base de données NoSQL pouvant stocker de gros volumes de documents (sous format JSON), que l'on peut interroger de manière précise grâce à un langage de requêtes HTTP (architecture REST) offrant de nombreuses fonctionnalités. Ainsi, cet outil met à disposition des utilisateurs une API d'accès aux données et s'utilise donc avec n'importe quel langage de programmation.

ElasticSearch utilise la bibliothèque *Lucene*, également open source et écrite en Java. Elle offre la possibilité d'indexer et de chercher des données, essentiellement de type texte. Il s'agit d'une couche intermédiaire entre les données et les programmes. Les index attribués aux documents, de formats divers et variés, permettent d'accélérer et d'améliorer la recherche de contenu.

ElasticSearch s'inscrit dans la suite *Elastic* qui comprend également le module *Kibana*. Cette interface permet de visualiser les données *ElasticSearch* sous forme de représentations graphiques créées par l'utilisateur lui-même : histogrammes, graphes linéaires, diagrammes en secteurs… sans une seule ligne de code. Il est également possible de visualiser des données géographiques sur une carte. Les diagrammes réalisés peuvent être regroupés au sein de tableaux de bord personnalisés, destinés à être partagés.

Enfin, le module *Logstash*, qui a également été testé dans cette étude, est un logiciel capable d'importer des données en provenance d'un grand nombre de sources, de les uniformiser en les remaniant et de les envoyer vers un système de stockage (*ElasticSearch* en l'occurrence).

Metabase

Metabase est un outil de reporting de données qui se différencie des autres solutions similaires notamment par sa simplicité d'installation et d'utilisation. Cette application open source repose sur un système de questions/réponses : l'utilisateur interroge ses données sous forme de « questions » (formatées par l'application grâce à des filtres), qui sont en réalité des requêtes SQL cachées, et obtient des « réponses » simples sous forme de tableaux tout d'abord, puis au moyen de graphiques en tout genre selon ses envies. Cette solution est très pratique pour les utilisateurs qui ne possèdent pas de compétences en SQL. Toutefois, il est possible de produire des requêtes SQL pour accéder à des résultats plus avancés.

Les données peuvent provenir de bases de données variées : *MySQL, PostgreSQL, SQL Server, Oracle*... Comme pour *Kibana*, les diagrammes créés peuvent être exposés dans des tableaux de bord et partagés, grâce à des liens, dans d'autres programmes.

Kibana couplé à ElasticSearch et Logstash

Les outils *ElasticSearch*, *Logstash* et *Kibana* de la suite *Elastic* ont été testés avec des données provenant de la base de données de GeoNature. Il a d'abord fallu procéder à leur installation sous le système d'exploitation Linux (Ubuntu). La documentation *ElasticSearch* étant très vaste et peu intelligible à ce sujet, j'ai dû me référer à un tutoriel trouvé sur internet, provenant de la communauté *DigitalOcean* (hébergeur web). L'installation complète des trois outils avec la configuration appropriée a pris plusieurs jours. Les différentes étapes sont détaillées dans l'article « Connecter des outils de DataViz à GeoNature » que j'ai écrit pour le Parc, accessible à l'adresse suivante : https://si.ecrins-parcnational.com/blog/2019-04-dataviz-geonature.html.

La deuxième étape de l'expérimentation a consisté à importer les données de « synthèse » de GeoNature dans *ElasticSearch* par le biais de *Logstash*. Pour ce faire, un fichier a été créé au sein du répertoire *Logstash* afin de configurer la connexion à la base de données *PostgreSQL* de GeoNature, et d'appliquer la requête SQL adéquate de récupération des données d'intérêt. Une nouvelle fois, la documentation *ElasticSearch* n'étant pas assez précise, cette étape s'est avérée chronophage.

La troisième étape est la plus intéressante en lien avec la valorisation des données de GeoNature. Il s'agit de la prise en main de l'outil *Kibana*. Les données importées au sein d'*ElasticSearch* ont été récupérées sur l'interface de *Kibana*. Cette dernière étant relativement intuitive, et avec l'aide de la documentation, je suis rapidement parvenue à créer des graphes semblables à ceux développés sur GeoNature 1 : histogramme du nombre d'observations par année, camembert de la répartition des observations par rang taxonomique, histogramme horizontal du nombre d'observations par taxon. J'ai pu organiser ces graphes au sein d'un même « dashboard » (Figure 5), autrement dit tableau de bord, diffusable via un lien. J'ai également exploré quelques options concernant l'implémentation de filtres sur ce « dashboard ». En revanche, les fonctionnalités cartographiques se sont avérées moins concluantes au regard des besoins. Je n'ai pas réussi à insérer des polygones sur les cartes proposées. Cette fonctionnalité se révèle pourtant indispensable pour représenter les données de GeoNature selon le type de zonage, tel que les mailles ou les communes.

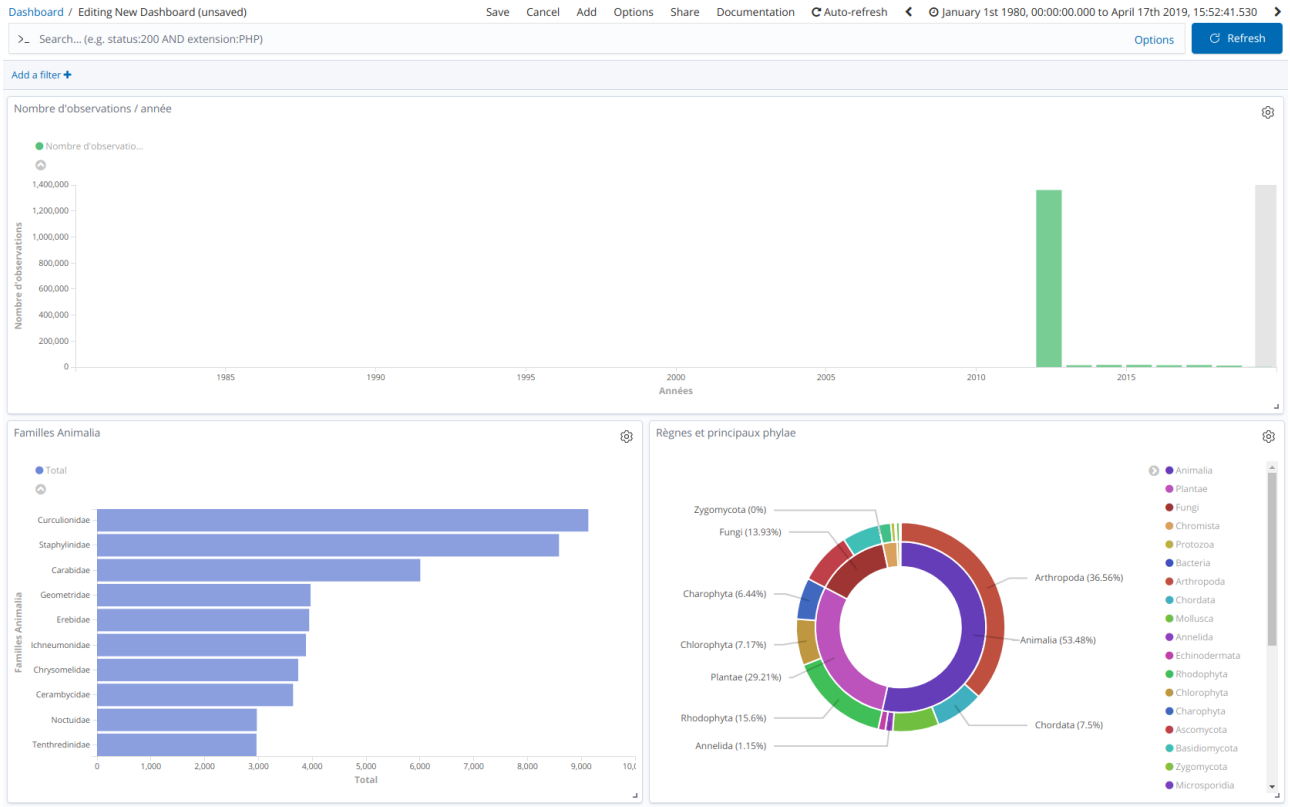


Figure 5 : « Dashboard » créé avec le logiciel *Kibana*

L'API REST d'*ElasticSearch* étant réputée rapide et puissante, j'ai également expérimenté les possibilités de requêtage des données importées au sein de ce logiciel. *ElasticSearch* possède son propre langage de requêtes, adaptable dans différents langages de programmation tels que Java, Python, PHP… J'ai réalisé ces tests sous *Angular*, avec l'aide de la documentation. Les possibilités de requêtage se sont avérées très vastes et complexes. La compréhension d'une infime partie d'entre elles a nécessité plusieurs jours.

Metabase

De même que pour les outils de la suite *ElasticSearch*, le logiciel *Metabase* a été installé sous Linux. La documentation de ce logiciel est, à l'inverse de celle d'*ElasticSearch*, synthétique et efficace, se suffisant à elle-même. L'installation, qui s'est avérée beaucoup plus simple, a été réalisée en moins de deux heures.

La connexion à la base de données GeoNature a pu être établie très facilement grâce au remplissage d'un formulaire simple (hôte et port, nom de la base, nom d'utilisateur et mot de passe...) via l'interface de *Metabase*. Le logiciel a ainsi accès à toutes les données présentes dans la base. Les différents schémas, tables et champs, ainsi que les données, sont consultables sur l'interface de *Metabase*.

L'élaboration de graphiques, similaires à ceux créés avec *Kibana* (Figure 6), au sein d'un « dashboard » a été particulièrement efficace. Le requêtage de données sous forme de « questions/réponses » facilite considérablement la prise en main du logiciel. Le passage d'une représentation tabulaire des réponses à une représentation graphique est proprement explicité. *Metabase* permet notamment l'insertion de polygones sur ses cartes grâce à la spécification d'URL (adresse web) de fichiers GeoJSON. Malheureusement, cette fonctionnalité n'a pas pu être étudiée de manière poussée et n'a pas abouti.

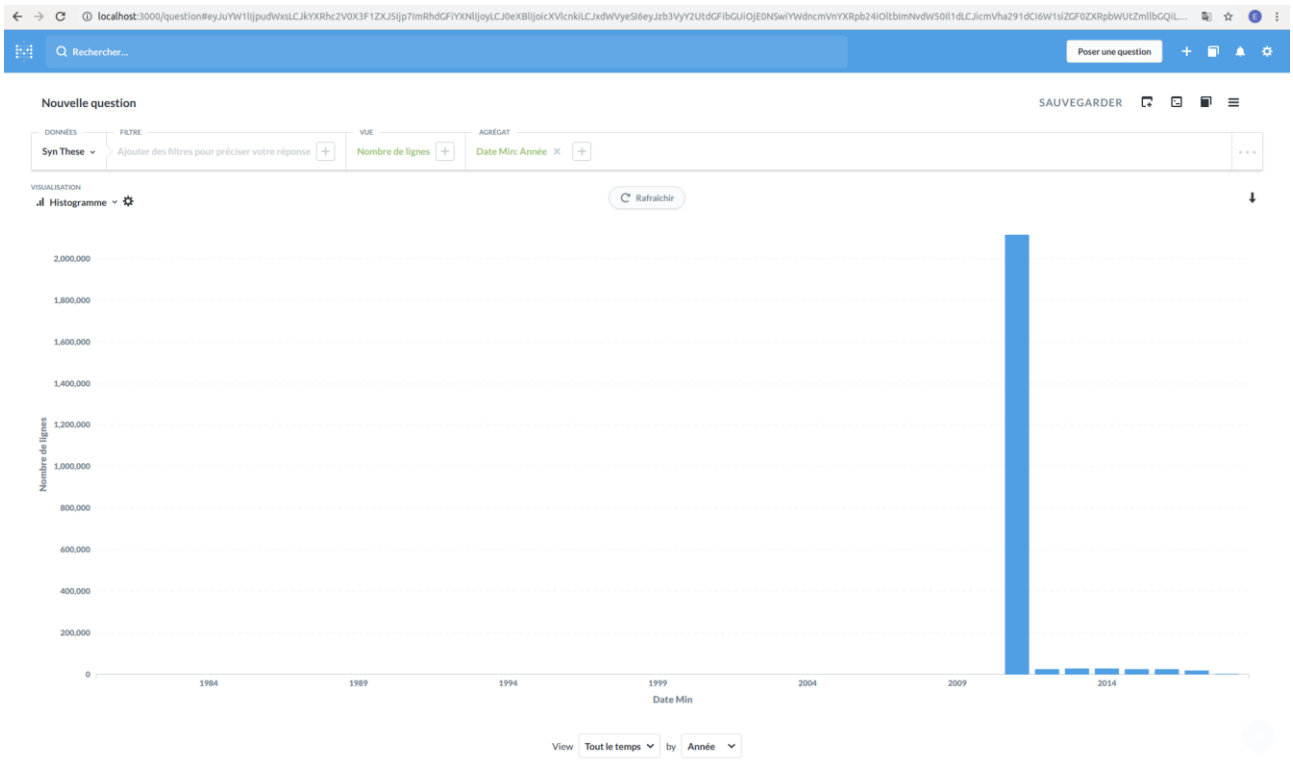


Figure 6 : Histogramme réalisé avec le logiciel *Metabase*

Toutes les étapes de ces expérimentations et les résultats obtenus sont détaillés dans l'article mentionné précédemment.

Comparaison des deux solutions

Kibana et *Metabase* sont des logiciels permettant tous deux de créer des graphiques et cartes personnalisables en lien avec des données contenues dans une base. Cependant, les phases de test ont mis en lumière des différences, notamment en termes d'installation et de facilité d'utilisation, entre ces deux outils. Le récapitulatif de ces divergences est présenté dans le tableau suivant (Tableau 6) :

Tableau 6 : Comparaison des logiciels *Kibana* et *Metabase*

Caractéristiques		Kibana	Metabase
Qualité de la documentation		- Longue - Peu intelligible	- Concise - Claire
Caractéristiques de l'installation		- 3 outils à installer - Nombreuses étapes pointilleuses	Une seule étape simple
Connexion à la base de données GeoNature		Nécessitant la création et la configuration d'un fichier	Guidée via un formulaire sur l'interface
Graphiques/ cartes et tableau de bord	Prise en main du logiciel pour la conception	- Guidée par une interface esthétique mais complexe - Assez chronophage - Création difficile de cartes personnalisées	- Guidée par une interface esthétique et simple - Plutôt rapide - Création difficile de cartes personnalisées
	Diversité des graphiques	- Très importante	- Importante
	Customisation des graphiques	- Parfois limitée	- Dense

b) Des solutions externes ne permettant pas de répondre de manière optimale aux besoins identifiés

Les résultats du croisement des critères de sélection avec la liste des solutions envisagées sont présentés dans le Tableau 7, avec des éléments d'explication pour certaines affectations de points :

Tableau 7 : Evaluation des solutions existantes

Critères	Kibana	Metabase	Développement en interne
Open source	2	2	2
Unité avec GeoNature	0	0	1
Facilité de développement	1	2	3
	Phase d'apprentissage nécessaire		Compétences de développement déjà maîtrisées
Facilité d'installation du module final	0	1	3
	Installation du logiciel + configuration de la connexion à la base de données + élaboration du « dashboard »		Installation standard au même titre que les autres modules (rapide)
Accessibilité pour tous les types d'utilisateurs	2	2	2
Capacité à répondre aux besoins identifiés	2	2	3
	Les besoins sont peut-être tous couverts mais la faible connaissance des outils ne permet pas de l'assurer.		En développement pur, toutes les fonctionnalités sont réalisables.

Facilité de contribution de la communauté (perspectives d'évolution)	0	1	3
	Phase d'apprentissage nécessaire pour la communauté également		Compétences de développement déjà maîtrisées
Autres éléments	1	0	0
	API rapide de requêtage des données		
SCORE TOTAL	**8**	**10**	**17**

Les résultats de cette analyse ont permis de désigner la méthode de développement en interne (utilisation des frameworks *Flask* et *Angular*) comme solution la plus adaptée pour l'élaboration du module « Tableau de bord ». Les facilités de développement et d'installation, la capacité à répondre aux besoins identifiés et la facilité de contribution de la communauté sont les arguments principaux qui font de cette solution la plus pertinente à adopter.

Comme cela a déjà été précisé précédemment (page 18), un article au sujet des recherches réalisées pendant le stage sur *Kibana* et *Metabase* a été publié sur le portail du SI du Parc. Ainsi, les utilisateurs de GeoNature peuvent prendre en main ces outils rapidement et les manipuler eux-mêmes pour créer leurs propres représentations graphiques et traiter des données s'ils en ont besoin.

3. Des méthodes diverses de validation automatique de données naturalistes basées sur des réflexions et traitements similaires

a) Une analyse de l'existant basée sur six systèmes d'information traitant des données naturalistes

Durant six semaines d'études, je suis parvenue à récolter les protocoles de validation automatique de six organismes, à travers la documentation ou des entretiens. La liste des systèmes analysés ainsi que leurs descriptions sont présentées dans le Tableau 8 (Robert, 2015) :

Tableau 8 : Liste et description des systèmes d'information considérés dans l'état de l'art relatif à la validation automatique de données naturalistes

Organisme	Outil	Description du système
SINP national	INPN	Plateforme nationale du SINP : outil public permettant de consulter les données de biodiversité du territoire français.
Conservatoire d'Espaces Naturels du Languedoc-Roussillon (CEN LR)	Atlas des papillons de jours et des libellules du Languedoc-Roussillon	Projet participatif de recensement de deux groupes d'espèces d'insectes : les papillons de jours et les libellules.
SINP Provence-Alpes-Côte d'Azur (PACA)	SILENE (Système d'Information et de Localisation des Espèces Natives et Envahissantes)	Plateforme régionale (PACA) du SINP : outil public permettant de consulter les données de biodiversité de la région PACA.
SINP La Réunion	Borbonica	Plateforme régionale (La Réunion) du SINP : outil public permettant de consulter les données de biodiversité de l'île de La Réunion.
Tela Botanica	Flora Data	Programme national de sciences participatives permettant à tous les botanistes, novices comme experts, d'améliorer les connaissances

		sur la flore, notamment via la saisie d'observations (outil *Carnet en ligne*).
Picardie Nature	Clicnat	Logiciel libre permettant à tous les Picards de saisir et gérer leurs observations de faune sauvage et de consulter ces informations (cartes de répartition des espèces).

Tous les protocoles étudiés sont synthétisés dans les sections qui suivent. Le niveau de détail obtenu n'est pas le même pour toutes les études. En effet, les informations concernant la mise en œuvre technique n'ont pas toujours été fournies.

INPN

La procédure de validation automatique de l'INPN réalise plusieurs contrôles sur la donnée saisie, en se basant sur deux référentiels : TAXREF et l'Atlas de la Biodiversité Départementale et des Secteurs Marins (ABDSM). L'ABDSM est un programme du MNHN ayant pour but de créer des cartes nationales de répartition géographique d'espèces sous la forme de présence/absence, expertisées par département. Les contrôles effectués sur les données intégrées sont les suivants (Tableau 9) :

Tableau 9 : Liste et description des contrôles réalisés sur les données de l'INPN

Référentiel concerné	Nom du contrôle	Description du contrôle
TAXREF	Contrôle de reconnaissance par TAXREF (R)	Vérifie l'existence du cd_nom (identifiant unique de taxon) dans TAXREF.
	Contrôle du statut biogéographique (SB)	Vérifie que la donnée d'observation est cohérente par rapport aux informations relatives au statut biogéographique contenues dans TAXREF.
	Contrôle de l'habitat (H)	Vérifie que l'habitat associé au taxon est cohérent avec la situation géographique de la maille 10x10km où est située l'observation.
ABDSM	Contrôle ABDSM	Vérifie que le taxon est situé dans son aire de répartition ou dans les environs.

Une fois ces contrôles effectués, un niveau de validité est attribué à la donnée à l'aide d'un arbre de décisions (Figure 7) prenant en compte les résultats des contrôles (Robert et al., 2017) :

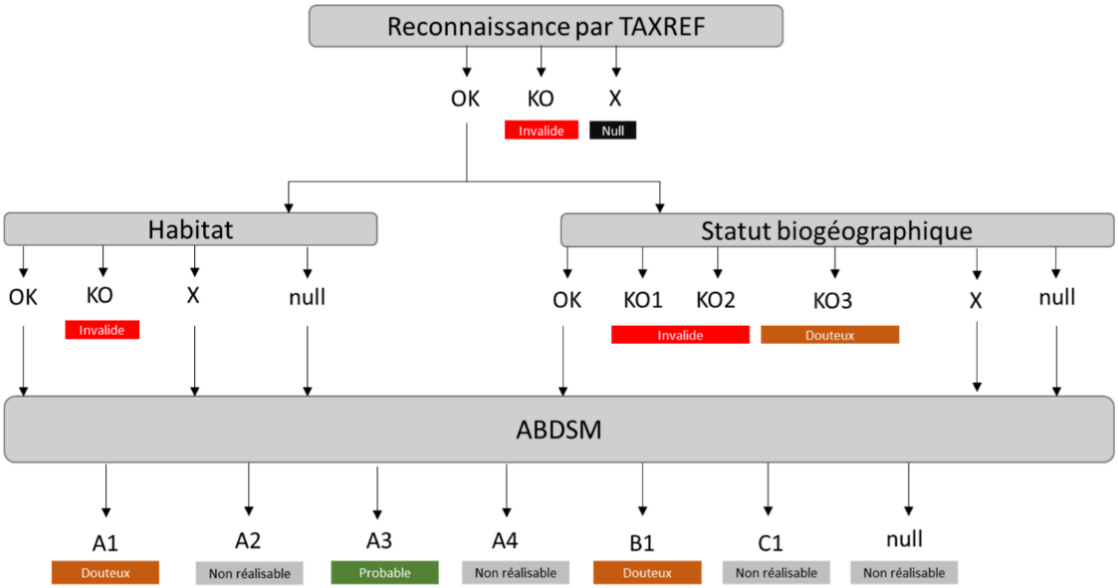


Figure 7 : Arbre de décisions pour l'affectation d'un niveau de validité du protocole automatique de l'INPN (source : INPN)

Atlas des papillons de jours et des libellules du Languedoc-Roussillon

Le Conservatoire d'Espaces Naturels du Languedoc-Roussillon caractérise chaque taxon selon les connaissances actuelles déjà validées dans la base de données de l'*Atlas*. Pour ce faire, les références suivantes ont été calculées pour chaque taxon dans une nouvelle table de la base de données (une entrée par taxon) :

- Liste des communes pour lesquelles le taxon a déjà été observé
- Liste des entités et ensembles paysagers pour lesquels le taxon a déjà été observé
- Liste des semaines et décades d'observation
- Liste des observateurs ayant déjà une donnée validée pour le taxon
- Répartition altitudinale d'observation : liste de plages de 100m d'altitude, correspondant à la division entière de l'altitude par 100

Chaque saisie de donnée (ou chaque donnée modifiée) est confrontée aux listes de référence calculées pour le taxon concerné. Cette opération est permise par l'implémentation d'un trigger déclenché à chaque insertion d'une donnée dans la base. Le trigger lance une fonction de validation, qui ajoute un score dans le champ « décision de validation » associé à l'observation fraîchement intégrée. Cette fonction de validation fait elle-même appel à une fonction de comparaison permettant de calculer ce score, qui correspond en fait à une suite de 0 et de 1. Cette deuxième fonction compare en fait chaque liste de référence du taxon concerné aux champs de l'observation. Si l'information est contenue dans la liste, alors la fonction ajoute 1 au score, et si ce n'est pas le cas, alors la fonction ajoute 0. Les validateurs peuvent ensuite filtrer les données selon ce score.

Chaque nouvelle donnée validée entraîne la mise à jour du calcul des références pour le taxon concerné. Cette opération a également lieu lorsqu'une donnée validée redevient invalide. C'est un nouveau trigger qui déclenche une fonction de calcul de référence, permettant de supprimer l'ancienne ligne de référence et d'ajouter la nouvelle après l'avoir calculée (Bossaert, 2015).

SILENE

Le système d'information *SILENE* comporte deux déclinaisons : *SILENE Flore* et *SILENE Faune*. Ces deux outils ne sont pas constitués du même protocole automatique de validation de données. *SILENE Faune* n'a pas pu être étudié.

Le processus de validation automatique de *SILENE Flore* a été conçu par le Conservatoire Botanique National Alpin (CBNA). Il se concentre sur trois niveaux :

- La cohérence territoriale : cette analyse vérifie si le taxon concerné par l'observation a déjà été aperçu sur le même territoire, à l'échelle de la commune, de la maille de 5km, du district naturel (petite région géographique) ou bien du département.
  Pour ce faire, chaque donnée est comparée aux listes suivantes, qui ont été extraites de la base de données à partir des observations validées : liste des taxons observés par commune, liste des taxons observés par maille de $5km^2$, liste des taxons observés par district naturel et liste des taxons observés par département.
- La cohérence altitudinale : cette analyse vérifie que l'altitude de l'observation est comprise entre les altitudes minimale et maximale calculées par analyse statistique sur les données validées.
- Taxons à vérifier obligatoirement : cette analyse permet de déterminer les observations qui doivent systématiquement être soumises à un contrôle par un botaniste référent, en raison du statut du taxon concerné. Les types de taxons impliqués sont : les taxons protégés (à l'échelle nationale, régionale et départementale), les taxons dont le statut IUCN (liste rouge) est particulièrement critique (en danger d'extinction), les taxons rares, les taxons difficiles à déterminer ou mal connus.

Le traitement des données et l'attribution d'un niveau de validité sont ensuite effectués à l'aide d'une liste de cas modélisable sous forme de schéma (Figure 8) (Genis et al., 2017).

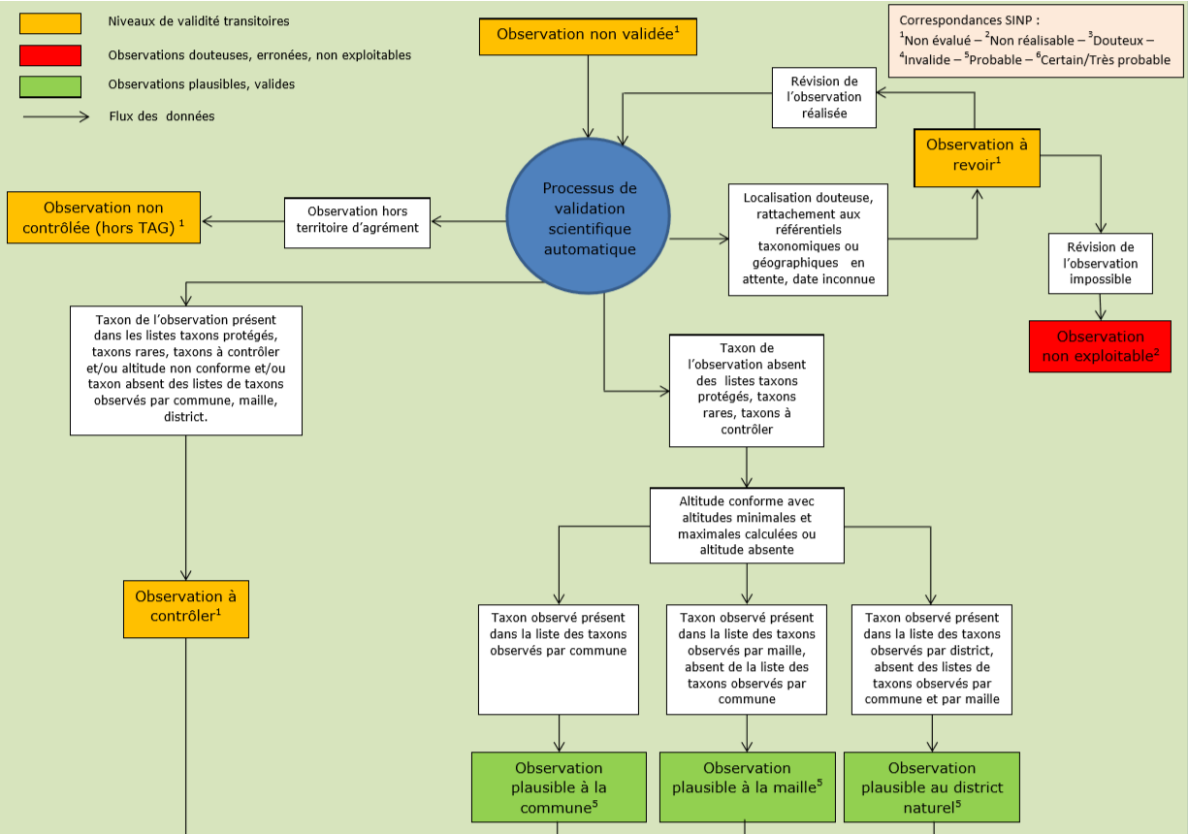


Figure 8 : Schéma des cas d'attribution d'un niveau de validité du protocole automatique de SILENE Flore (source : CBNA)

Borbonica

La validation automatique régionale du SINP La Réunion consiste à tester un certain nombre de règles sur la donnée et à en déduire un niveau de validité. Une règle est construite grâce à la combinaison des éléments suivants :

- Un taxon ou un groupe de taxons
- Le(s) champ(s) testé(s)
- La condition à remplir pour ce(s) champ(s) (donnée de référence)
- Le niveau de validité à attribuer lorsque la condition est remplie

Elles permettent de comparer certains champs de l'observation par rapport à des bases de connaissance et des données de référence calculées à partir de la base de données, mais elles considèrent aussi l'existence de preuves (photos, enregistrements).

L'élaboration de ces règles scientifiques est déléguée aux têtes de réseau du SINP La Réunion, qui correspondent aux structures énumérées dans le tableau ci-contre (Tableau 10) :

Tableau 10 : Liste des têtes de réseau du SINP La Réunion (source : SINP La Réunion)

Pôle thématique	Tête de réseau	Validateurs
Flore et habitats naturels	Conservatoire botanique national de Mascarin (CBNM)	Frédéric Picot Marie Lacoste
Insectes et arachnides	Unité mixte de recherche « protection des végétaux et biologie des milieux tropicaux » (UMR PVBMT)	Samuel Nibouche Bernard Reynaud
Oiseaux	Société d'études ornithologiques de La Réunion (SEOR)	Martin Riethmuller Laurent Brillard Alexandre Boyer Jérôme Dubos Marc Salamolard
Reptiles et amphibiens	Nature Océan Indien (NOI)	Mickaël Sanchez
Tortues marines	Kelonia	Claire Jean Stéphane Ciccione Mayeul Dalleau
Chiroptères	Groupe chiroptères Océan Indien (GCOI)	Gildas Monnier Sarah Fourasté
Mammifères terrestres non volants	Office national de la chasse et de la faune sauvage (ONCFS)	Sarah Caceres
Cétacés	Globice	Violaine Dulau Vanessa Estrade
Poissons et macro-crustacés d'eau douce	*Pôle thématique non encore mis en place. Dans l'attente, rédaction du protocole de validation confiée à Ocea (Pierre Valade, Laëtitia Faivre, Henri Grondin) et Nexa (Raphaël Lagarde)*	

Chaque pôle thématique possède ainsi ses propres règles. A l'échelle de l'ensemble des pôles, 405 règles concernant 4 786 taxons ont été identifiées. Le niveau de détail des règles est donc très important, même si celles-ci peuvent être regroupées par type : observation d'une espèce commune en dehors de la période habituelle, observation d'une espèce commune en dehors de son habitat préférentiel, observation d'une espèce en dehors des gammes d'altitude connues… Ces règles sont traduites en langage SQL pour pouvoir être mises en œuvre dans *Borbonica* par les administrateurs de données.

Les données de référence utilisées extraites de l'analyse des 64 types de règles sont les suivantes :

- Périodes d'observation de l'espèce
- Aire de répartition de l'espèce
- Habitats préférentiels de l'espèce
- Gammes d'altitude de l'espèce
- Caractéristiques de l'espèce : non décrite dans TAXREF, non commune à La Réunion, non connue à La Réunion, rare ou jamais observée dans la zone, indigène ou naturalisée, éteinte ou disparue de La Réunion, migratrice, nicheuse, reproductrice.
- Complexité d'identification de l'espèce

Il existe également des éléments bien plus précis qui ne sont testables que pour certaines espèces : profondeur, distance à la côte, phénologie…

Un arbre de décisions (Figure 9 – cas des tortues marines) est construit pour chaque pôle thématique, prenant en compte chacune des règles établies au sein du pôle, afin d'attribuer un niveau de validité à chaque donnée (Le Tellier, 2019) :

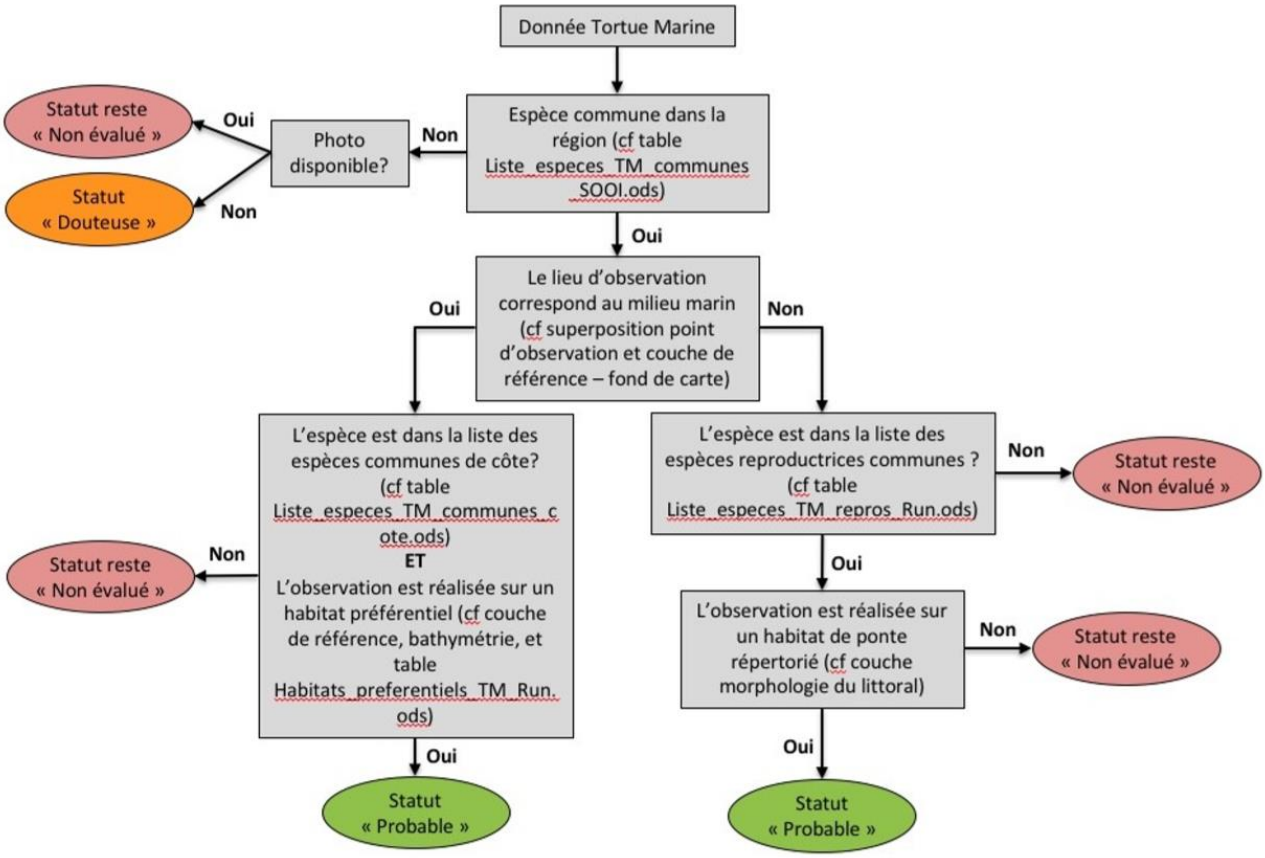


Figure 9 : Arbre de décisions pour l'affectation d'un niveau de validité du protocole automatique de *Borbonica* (source : SINP La Réunion)

Flora Data

Carnet en ligne possède un système d'alerte, à l'intention de l'utilisateur lors de la saisie, qui se déclenche lorsqu'un taxon est déclaré hors de sa zone de répartition. Cette zone de répartition est tirée d'un « référentiel de chorologie départementale ». Ce référentiel a été dressé grâce à des bénévoles du réseau, et recense les espèces présentes sur chaque département. Toutefois, il est de moins en moins actualisé. Ainsi, l'objectif à l'avenir serait de baser les tests sur les données des Conservatoires Botaniques Nationaux (départementales, voire communales).

Un indicateur de fiabilité des données est également en cours d'élaboration. Il serait destiné à séparer les données en plusieurs classes selon leur degré de complétude, de cohérence et de fiabilité. Les critères de cette fiabilité de l'identification des espèces reposent sur :

- La certitude de l'identification renseignée par l'auteur de l'observation : « à déterminer », « douteuse » ou « certaine ».
- La note de l'identification sur la plateforme collaborative *IdentiPlante* : les membres du réseau peuvent se prononcer pour ou contre une identification. Une note très négative, c'est-à-dire contenant beaucoup de « contre », signifie souvent une erreur d'identification, et inversement pour une note très positive.

La zone de répartition n'est pas prise en compte de manière volontaire dans l'établissement de la fiabilité : une observation dont la fiabilité semble élevée mais dont le taxon n'est pas cohérent avec les données départementales est probablement une donnée intéressante, alors qu'une observation dont la fiabilité semble faible et dont le taxon n'est pas cohérent avec les données départementales est probablement une erreur.

Clicnat

Picardie Nature réfléchit actuellement à la mise en place d'un système de validation automatique pour son outil de saisie *Clicnat*. L'idée générale consisterait à établir une grille de données de références importantes par groupe de taxons en tenant compte du statut biologique (notamment pour les oiseaux nicheurs/hivernants), grâce aux données déjà validées. Chaque nouvelle donnée serait confrontée à la grille de référence du groupe de taxons concerné par le biais de tests. Un statut « CERTAIN » serait attribué si tous les tests sont validés. Les résultats des tests seraient également affichés dans l'interface de validation manuelle à titre informatif.

Les tests utilisés seraient les suivants :

- Effectif : Validé si l'effectif est inférieur ou égal à celui de X % des observations (percentile) ou si effectif non spécifié.
- Capacité de l'observateur : Validé si l'observateur a déjà identifié l'espèce.
- Répartition connue : Validé si l'espèce a déjà été vue à proximité (distance en mètres).
- Période d'observation : Validé si l'espèce a déjà été vue à la même période (± X jours).
- Comportement : Validé si le comportement de l'espèce observé est dans la liste.
- Méthode d'observation : Validé si la méthode d'observation est dans la liste.
- Méthode de détermination : Validé si la méthode de détermination est dans la liste.

b) Des protocoles de validation automatique structurés par une démarche commune

L'étude des précédents protocoles semble démontrer qu'il n'existe pas de règles précises concernant la validation automatique de données. Chacun des organismes considérés a créé un protocole de validation qui lui est propre. Cependant, il est possible de mettre en évidence une trame globale généralisable à tous les processus.

Définition de bases de référence

Tous les protocoles étudiés sont capables de valider automatiquement les données du système d'information concerné en s'appuyant sur différentes bases de référence. En effet, chaque donnée saisie nécessite d'être comparée à ce qui s'apparente à une « norme ».

Ces bases peuvent être des référentiels nationaux ou départementaux, tels que TAXREF pour la taxonomie ou les couches de répartition des taxons déclinables selon différentes résolutions géographiques (maille, commune, département…). Ces données de référence peuvent également être déterminées à partir des données déjà validées du système d'information, par le biais de traitements. Dans ce cas, deux tendances opposées se dessinent :

- Un référentiel est établi par taxon ou par groupe de taxons (Figure 10) : il forme ce qui s'apparente à un profil type ou une fiche d'identité, comme ce qui a été réalisé pour l'*Atlas des papillons de jours et des libellules* (CEN LR). Chaque élément de référence caractérisant le taxon/groupe (zone de répartition, dates d'observation, altitudes d'observation, etc.) est associé à une valeur ou une liste de valeurs de référence estimées grâce aux données déjà validées pour ce taxon/groupe. Ces « normes » peuvent être simplement extraites de la base de données à l'aide de requêtes SQL ou bien calculées à l'aide de fonctions statistiques ou spatiales.

Nom du taxon	Communes d'observation	Périodes d'observation	Gammes d'altitude d'observation	…
Ablette	Liste de communes	Liste de périodes	Liste de gammes	…
Bouquetin des Alpes	Liste de communes	Liste de périodes	Liste de gammes	…
Loup commun	Liste de communes	Liste de périodes	Liste de gammes	…
Lynx boréal	Liste de communes	Liste de périodes	Liste de gammes	…
…	…	…	…	…

Figure 10 : Exemple de référentiel établi par taxon (profil type)

Les éléments considérés peuvent être des paramètres, c'est-à-dire des éléments comparés directement aux informations des nouvelles données saisies (valeurs de champs) pour les valider intrinsèquement, ou bien des critères d'orientation, c'est-à-dire des éléments permettant de focaliser les efforts de validation manuelle sur des données nécessitant une attention particulière.

- Un référentiel est établi par paramètre ou critère considéré comme pertinent dans la caractérisation des observations de taxons (Figure 11) : c'est la méthode employée par le CBNA pour *SILENE Flore*. Une liste de taxons est associée à chaque valeur possible de l'élément. A titre d'illustration, un référentiel établi à l'échelle des communes correspondrait aux listes de taxons déjà observés par commune.

Communes d'observation	
Valeurs possibles	**Taxons concernés**
Aiguille	Liste de taxons
Ancelle	Liste de taxons
Barcelonnette	Liste de taxons
Briançon	Liste de taxons
…	…

Périodes d'observation	
Valeurs possibles	**Taxons concernés**
Semaine 1	Liste de taxons
Semaine 2	Liste de taxons
Semaine 3	Liste de taxons
Semaine 4	Liste de taxons
…	…

Gammes d'altitude d'observation	
Valeurs possibles	**Taxons concernés**
0-100m	Liste de taxons
100-200m	Liste de taxons
200-300m	Liste de taxons
300-400m	Liste de taxons
…	…

Figure 11 : Exemples de référentiels établis par paramètre ou critère

Établissement de règles de comparaison

Une fois le(s) référentiel(s) sélectionné(s) ou établi(s), il est possible d'élaborer des règles de comparaison. Comme cela a été précisé par le SINP La Réunion, une règle permet de confronter une des valeurs des champs de l'observation à la base de référence qui lui est associée. Une valeur de seuil peut également être implémentée si nécessaire. C'est le cas de *Picardie Nature* par exemple qui

souhaite tester le champ du lieu d'observation en définissant une distance en mètres par rapport à l'observation la plus proche du même taxon. L'application d'une règle sur une donnée sera appelée « test ». Le résultat d'un test est obligatoirement de la forme « conforme » ou « non conforme ».

Les règles et les seuils peuvent être déterminés pour l'ensemble des taxons (cas de l'*Atlas des papillons de jours et des libellules*), ou bien être déclinés par groupe taxonomique (cas de *Borbonica*). En effet, certains champs importants ne sont applicables qu'à un nombre limité de taxons, tels que le statut de reproduction « nicheur » pour les oiseaux.

L'ensemble des paramètres et critères (éléments de référence) identifiés lors des études et utilisés au sein des règles sont détaillés dans la liste suivante :

- Aire de répartition de l'espèce
- Périodes d'observation de l'espèce
- Plages d'altitude de l'espèce
- Types d'habitats de l'espèce
- Effectifs observés de l'espèce
- Observateurs ayant une donnée validée pour l'espèce
- Statuts spécifiques de l'espèce : protégée, en danger, rare, peu connue
- Difficulté de détermination de l'espèce
- Niveau de certitude de l'identification renseigné par l'observateur
- Comportement de l'espèce

Qualification scientifique des données saisies

Les résultats des différents tests appliqués à la donnée saisie doivent être combinés les uns aux autres afin de qualifier la donnée en termes de fiabilité. Pour cette étape également, deux démarches sont possibles :

- La mise en place d'un arbre de décisions : il s'agit d'une suite logique de choix représentée sous la forme d'un arbre, les extrémités des branches correspondant aux différentes décisions finales possibles. Dans le cas de la validation automatique de données, les extrémités représentent les différents niveaux de validation existants : « certain – très probable », « probable », « douteux », « invalide », « non réalisable », « non évalué » (cas de l'INPN). Les embranchements de l'arbre correspondent aux choix découlant des résultats des tests. Cette méthode nécessite donc d'organiser les tests et leurs résultats les uns par rapport aux autres.
- Le calcul d'un score : les résultats des tests, « conforme » et « non conforme », peuvent également se traduire de manière numérique pour constituer un score final de la donnée. Ainsi, cette démarche nécessite d'attribuer une note aux résultats de chaque test. Le score final peut être formé de la somme des notes de tous les tests ou bien juste de leur concaténation. Il peut ensuite être affiché tel quel lors de la visualisation des données, ou bien servir d'indicateur pour l'attribution de niveaux de validité.

Mise en œuvre technique

Les seules précisions techniques mentionnées dans l'étude des six protocoles sont des commandes exécutées dans la base de données des outils. Le CEN Languedoc-Roussillon créé ses référentiels sous format SQL. Les données de référence sont extraites de la base pour chaque taxon et stockées dans une nouvelle table.

L'implémentation des tests de l'*Atlas des papillons de jours et des libellules* et de *Borbonica* s'opère sous la forme de requêtes SQL. Dans le cas de l'*Atlas*, ces requêtes renvoient les résultats des

tests. Elles sont écrites dans des fonctions activées grâce à des triggers, se déclenchant dès l'insertion ou la modification d'une donnée dans la base.

Le CEN Languedoc-Roussillon assure la mise à jour de ses référentiels de la même manière. Un trigger est déclenché lorsqu'une nouvelle donnée est validée, entraînant le déroulement d'une fonction de calcul de données de référence. Ce système est appelé « rétroaction » : les données validées servent à produire des couches de référence, qui sont elles-mêmes utilisées pour la validation.

L'état de l'art concernant la mise en place d'un module de reporting de données a mis en évidence deux solutions logicielles spécialisées dans la réalisation de tableaux de bord interactifs. Ces outils ne se sont pas avérés suffisamment adaptés aux besoins et simples d'utilisation pour constituer des solutions techniques exploitables pour GeoNature. Le module « Tableau de bord » sera donc développé en interne par le PNE, avec les technologies *Angular* et *Flask*. Ainsi, la troisième partie sera dédiée à la présentation du développement du module, aux difficultés rencontrées lors de cette phase et aux améliorations qui seront possible d'apporter.

L'analyse de l'existant relative aux systèmes de validation automatique de données naturalistes a permis quant à elle d'esquisser, à travers l'étude de solutions diverses et variées, une architecture générale de protocole adaptable aux données du PNE. La troisième partie sera donc consacrée à la complétion du travail amorcé par le Parc à l'aide de cette trame et des scripts SQL du CEN Languedoc-Roussillon, qui sont particulièrement en accord avec les aspirations du service scientifique.

III. Un module « Tableau de bord » et un protocole de détection automatique de données naturalistes atypiques développés en interne par le Parc national des Écrins

1. Des mises en œuvre techniques réfléchies et cadrées

a) Un module GeoNature nécessitant des technologies et outils spécialisés

Après l'installation de l'application GeoNature sur mon poste en local, une partie du stage a consisté à amorcer le développement du module « Tableau de bord » avec la solution identifiée durant la phase précédente. Les sections suivantes décrivent la méthodologie qui a été adoptée pour répondre à cette mission.

Choix des graphes à implémenter

La sélection des représentations graphiques et cartographiques à mettre en place était fonction de leur facilité de déploiement et de leur correspondance avec les besoins identifiés les plus importants. Trois représentations ont été imaginées avant le lancement du développement : un histogramme du nombre d'observations et du nombre de taxons identifiés par année, une carte du nombre d'observations et du nombre de taxons identifiés par zonage, et un diagramme en secteurs de la répartition des observations par rang taxonomique. Ces éléments seront décrits plus précisément dans la présentation du module final. Ils ont été codés de manière successive, les uns après les autres. Le temps accordé à cette phase de développement a été suffisamment long pour permettre la réalisation de deux autres graphes : des courbes du nombre d'observations par année pour chaque cadre d'acquisition et un camembert de la répartition des taxons recontactés, non recontactés et nouveaux par année.

Mise en forme des données

La première étape du développement consiste à réfléchir aux données indispensables pour la construction de chaque graphe et de l'interface en générale. Le but est d'identifier les cas pour lesquels il est nécessaire de créer une vue matérialisée en base de données. En effet, une vue matérialisée correspond au stockage des résultats d'une requête SQL sous la forme d'une table. A la différence d'une vue standard, les données sont enregistrées, ce qui implique que la requête n'est pas exécutée à chaque consultation des résultats. En revanche, une vue matérialisée nécessite d'être rafraîchie régulièrement pour rester à jour. Ce type de vue permet d'augmenter les performances de manière considérable lorsque la requête concernée est complexe, qu'elle comporte plusieurs jointures ou fonctions de calcul. Trois vues matérialisées, qui seront décrites par la suite, ont été créées.

Familiarisation avec Angular et Flask

Je n'avais jamais eu recours aux frameworks *Angular* et *Flask* avant cette mission de stage. J'ai pu connaître et prendre en main ces outils essentiellement à l'aide de tutoriels *Openclassroom*, la référence en ligne pour les cours d'informatique. Théo LECHÉMIA, le développeur du pôle SI du PNE, était également présent pour m'apporter des éléments complémentaires et faciliter mon apprentissage.

Il faut savoir que le framework *Angular*, assurant le codage du front-end, se divise principalement en deux types de fichiers : les fichiers TypeScript et les fichiers HTML. Chaque fichier TypeScript permettant de coder la partie dynamique d'une page est relié à un fichier HTML à l'origine de l'ergonomie et de l'agencement des éléments sur la page. Les deux fichiers, qui forment un

composant *Angular*, peuvent communiquer entre eux et s'envoyer des informations. C'est ce qu'on appelle le data-binding. Cette notion a été particulièrement explorée dans le but d'appréhender la mise en place de formulaires pour l'implémentation de filtres.

Au sein du back-end de GeoNature, l'utilisation du framework *Flask* est couplée à celle de la librairie Python *SQLAlchemy*. Il s'agit d'un Object Relational Mapping (ORM), c'est-à-dire un programme qui assure le lien entre une application et une base de données relationnelle afin de modéliser une base de données orientée objet. *SQLAlchemy* permet de manipuler la base de données GeoNature (connexion à la base, déclaration de modèles de données, production de requête SQL) sous forme de classes et d'objets Python. La partie concernant la récupération des données par le back-end et leur envoi à l'interface a été particulièrement approfondie.

Choix des librairies à installer pour réaliser les représentations

Une librairie pour l'instauration de graphiques

L'implémentation des graphiques a été possible grâce à l'utilisation de la librairie *ng2-charts*, nécessitant elle-même l'installation de la librairie *chart.js*. Cette librairie de création de graphiques interactifs fait appel aux langages JavaScript et HTML. Elle a été sélectionnée parmi les autres librairies existantes en raison de son caractère open source, sa popularité, sa documentation importante et son design.

Une librairie pour la création de cartes personnalisées

La partie cartographique du module a été développée avec la librairie JavaScript *Leaflet*. Cette librairie open source permet la création de cartes interactives. Elle a été choisie car elle est simple d'utilisation et déjà employée dans d'autres modules de GeoNature. En effet, l'application propose des composants *Angular* assurant l'affichage simplifié de cartographies avec *Leaflet*.

Planification de tests utilisateurs

Afin d'évaluer le module une fois le développement achevé, et dans le but de connaître le ressenti utilisateur, une phase de test a été organisée. Le module a été expérimenté par deux types de profils :

- Des utilisateurs directs de GeoNature : Ludovic IMBERDIS et Donovan MAILLARD ont pu tester le module afin de donner leur avis sur les améliorations à apporter à chaque graphe, et fournir d'éventuelles nouvelles idées de représentations. Cette phase n'était pas guidée. Les testeurs ont navigué au sein du « Tableau de bord » à leur guise et émettaient des commentaires et des suggestions lorsqu'ils le souhaitaient.
- Des utilisateurs « grand public » : 8 salariés du PNE n'utilisant pas GeoNature ont manipulé le module afin de rendre compte de la facilité de compréhension et d'utilisation de l'outil, ainsi que son ergonomie. Les testeurs ont pu prendre en main le « Tableau de bord » de manière libre, sans limite de temps, puis ont ensuite été soumis à un petit questionnaire ayant pour but d'interroger les données de la base à travers le module pour en tirer des conclusions. Ce questionnaire est présent en Annexe 5. Une note sur 5 a d'abord été attribuée à chaque testeur selon ses réponses, puis la moyenne de toutes les notes a été calculée.

b) Des protocoles de validation automatique permettant de renforcer la réflexion entamée par le Parc national des Écrins

L'analyse des processus de validation automatique étudiés vient confirmer les idées exprimées par le PNE concernant la mise en place de son propre protocole. La démarche du Parc, qui consiste à

comparer les nouvelles données saisies aux fiches d'identité des taxons (données de référence) établies en base de données, correspond tout à fait au bilan dressé. De ce fait, l'état de l'art réalisé va permettre de compléter et d'enrichir la base de travail du service scientifique du Parc.

Pour ce faire, les éléments érigés dans chaque étape de la trame globale extraite de l'analyse de l'existant ont été reconsidérés un à un. Les points les plus pertinents ont été retenus pour compléter les sections manquantes du protocole du PNE, et les éléments en accord avec les idées du Parc ont été retenus pour enrichir et améliorer le protocole. Concernant la partie technique, des fonctions et triggers SQL ont été construits dans la base de données GeoNature, s'inspirant fortement de ceux implémentés par le CEN Languedoc-Roussillon.

2. Un module « Tableau de bord » interactif et paramétrable, destiné à connaître de nombreuses évolutions

a) Un module comportant différents graphiques et cartes flexibles

Description de l'interface réalisée pendant le stage

Le module développé se présente sous la forme d'une interface unique contenant des blocs dépliables au clic (Figure 12) :

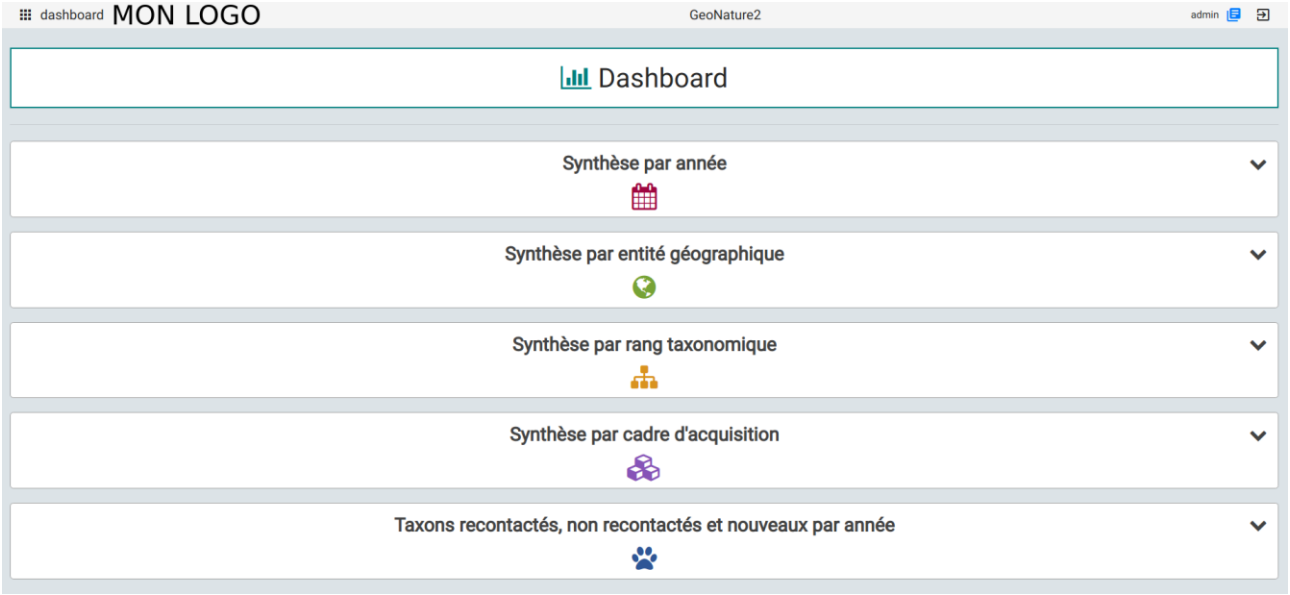


Figure 12 : Interface d'accueil du module "Tableau de bord"

Chaque bloc contient un graphe ou une carte que l'utilisateur peut afficher et masquer à sa guise. Quatre graphiques et une carte ont ainsi pu être déployés avec les librairies JavaScript adaptées. Ils sont décrits dans le tableau suivant (Tableau 11) :

Tableau 11 : Liste et description des graphes développés dans le module "Tableau de bord"

Titre de la représentation	Détails	Filtres
Graphe 1 : Histogramme du nombre d'observations et du nombre de taxons identifiés par année (Figure 13)	Histogramme à deux échelles	Rang taxonomique + Taxon
Cartographie : Carte du nombre d'observations et du nombre de	Carte des zonages (mailles, communes, départements, etc.) colorés selon leur	Type de zonages Période

taxons identifiés par zonage (Figure 14)	nombre d'observations ou de taxons (établissement d'une légende en classes)	Rang taxonomique + Taxon
Graphe 2 : Camembert de la répartition des observations par rang taxonomique (Annexe 6)	Diagramme en secteurs de la répartition des observations selon les taxons d'un rang taxonomique donné	Période Rang taxonomique
Graphe 3 : Courbes du nombre d'observations par année pour chaque cadre d'acquisition (Annexe 7)	Courbes multiples. Les cadres d'acquisition correspondent à des projets scientifiques.	-
Graphe 4 : Camembert de la répartition des taxons recontactés, non recontactés et nouveaux par année (Annexe 8)	Diagramme en secteurs de la répartition des observations de taxons par année selon leur statut. Un taxon recontacté est un taxon observé qui avait déjà été observé au moins une fois lors des années précédentes. Un taxon non recontacté n'est pas observé lors de l'année considérée alors qu'il avait été observé au moins une fois lors des années précédentes. Un nouveau taxon n'avait jamais été observé lors des années précédentes.	Année

Les filtres implémentés permettent de rendre les graphes et la carte dynamiques. Les données affichées sont ajustées instantanément à chaque changement des valeurs des filtres par l'utilisateur. Le détail de ces filtres est explicité ci-après :

- *Rang taxonomique* : liste déroulante contenant les options « Groupe 1 INPN », « Groupe 2 INPN », « Règne », « Phylum », « Classe », « Ordre », « Famille », « Rechercher une espèce... ». *Taxon* : liste déroulante dépendante du résultat de *Rang taxonomique*, contenant les noms des taxons pour lesquels il y a au moins une observation.
- *Type de zonages* : liste déroulante contenant les noms des types de zonages (communes, mailles 10x10, mailles 5x5, etc.).
- *Période* : slider à double curseur (pas = 1 an) permettant de sélectionner une période comprise entre l'année de la toute première observation et l'année de la toute dernière observation.
- *Année* : liste déroulante contenant les années pour lesquelles il y a au moins une observation.

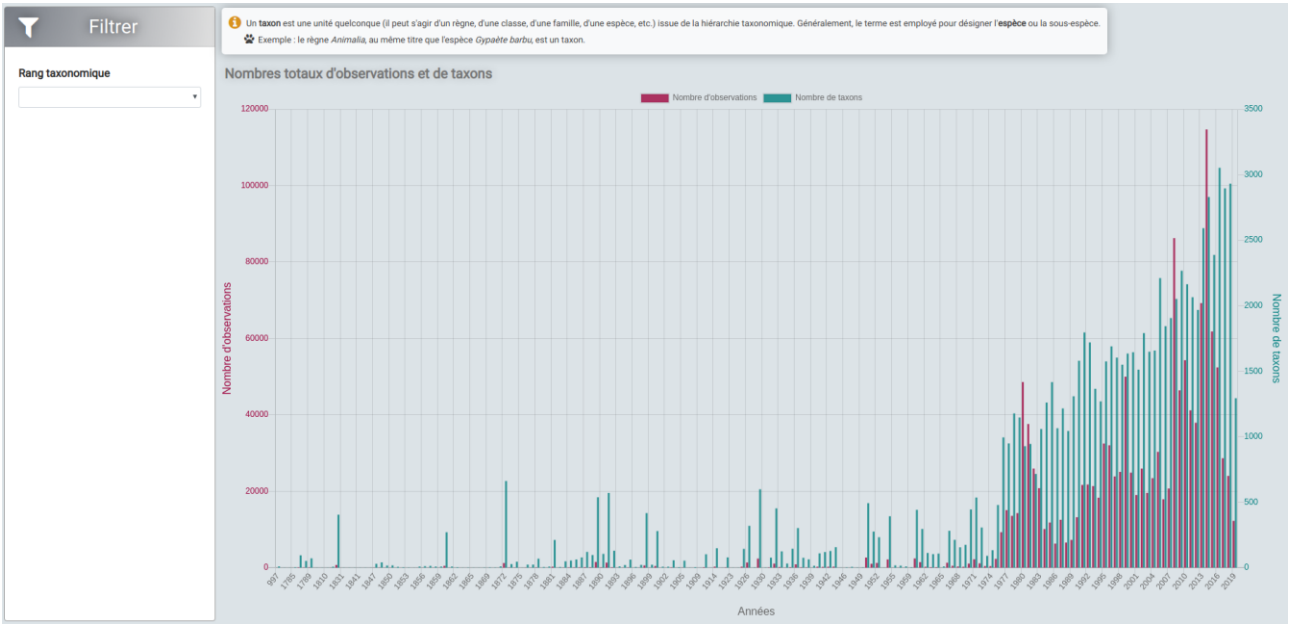


Figure 13 : Histogramme (*Graphe 1*) développé dans le module "Tableau de bord"

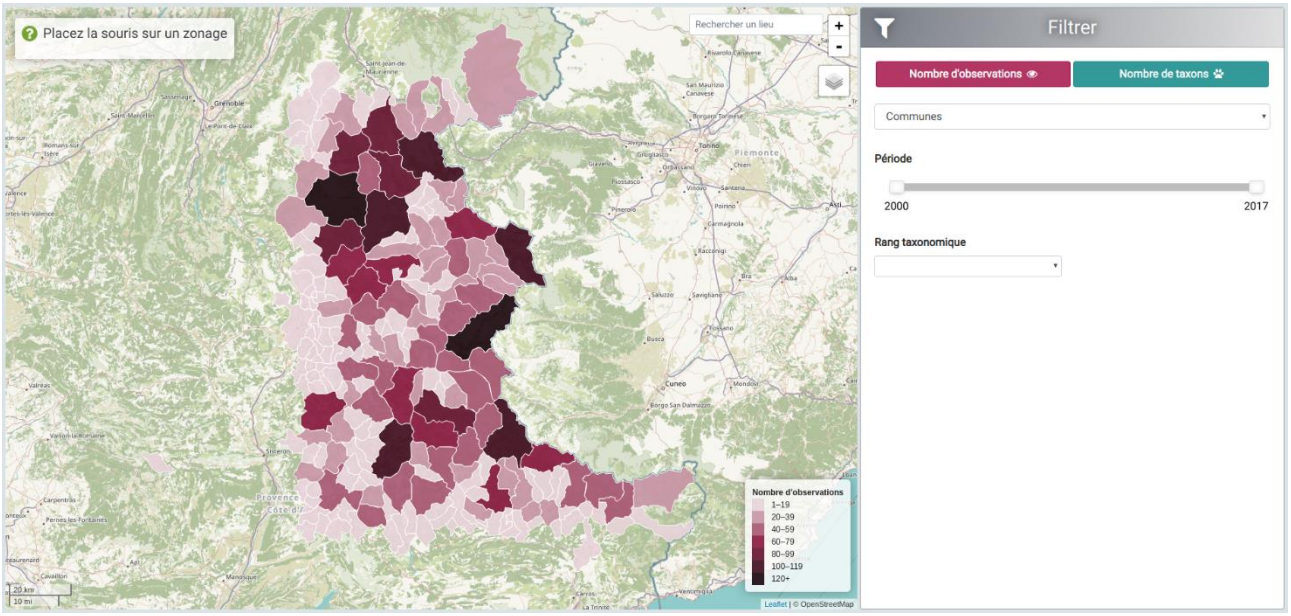


Figure 14 : *Cartographie* développée dans le module "Tableau de bord"

Mise en œuvre technique

La réalisation de cette interface graphique utilisant les données de GeoNature a nécessité aussi bien du développement front-end que du travail en back-end.

Développement back-end

Le développement back-end, qui s'appuie sur le framework Python *Flask*, permet de relier l'interface du module à la base de données GeoNature, et de récupérer les informations nécessaires à l'établissement des graphes. Ceci est assuré entre autres par la mise en place d'une web API d'accès à la base de données, qui est ensuite appelée par le front-end.

Tout d'abord, dans une optique d'optimisation des performances (diminution du temps de récupération des données par l'interface), trois vues matérialisées ont été créées dans un nouveau schéma « gn_dashboard » de la base de données GeoNature :

- *vm_synthèse* : cette vue est une jointure des tables « synthèse » et « taxref », sur la base du cd_nom (identifiant unique de taxon), permettant d'associer toutes les informations concernant la taxonomie (règne, phylum, famille, etc.) à chaque observation de taxon. Elle a été mise en place pour les *Graphes 1* et *2*. Le code SQL correspondant à la création de cette vue matérialisé est présent en Figure 16.
- *vm_synthese_frameworks* : cette vue est une jointure des tables « synthèse » et « acquisition_frameworks » permettant d'obtenir le nombre d'observations par cadre d'acquisition et par année, pour alimenter le *Graphe 3*. Elle fait appel à la fonction de calcul SQL « count » qui dénombre les observations, et à la clause « GROUP BY » qui permet d'effectuer les regroupements nécessaires (Figure 15).
- *vm_taxonomy* : cette dernière vue matérialisée permet d'enregistrer les noms de taxons qui ont été observés au moins une fois pour chaque rang taxonomique. Elle a été implémentée notamment pour alimenter les listes déroulantes servant de filtres pour les graphes.

```
CREATE MATERIALIZED VIEW gn_dashboard.vm_synthese_frameworks AS
 SELECT DISTINCT af.acquisition_framework_name,
    date_part('year'::text, s.date_min) AS year,
    count(*) AS nb_obs
   FROM gn_synthese.synthese s
     JOIN gn_meta.t_datasets d ON d.id_dataset = s.id_dataset
     JOIN gn_meta.t_acquisition_frameworks af ON af.id_acquisition_framework = d.id_acquisition_framework
  GROUP BY af.acquisition_framework_name, (date_part('year'::text, s.date_min))
  ORDER BY af.acquisition_framework_name, (date_part('year'::text, s.date_min))
WITH DATA;
```

Figure 15 : Requête SQL de création de la vue matérialisée *vm_frameworks*

```sql
CREATE MATERIALIZED VIEW gn_dashboard.vm_synthese AS
 SELECT s.id_synthese,
    s.id_source,
    s.id_dataset,
    s.id_nomenclature_obj_count,
    s.count_min,
    s.count_max,
    s.cd_nom,
    t.cd_ref,
    s.nom_cite,
    t.id_statut,
    t.id_rang,
    t.regne,
    t.phylum,
    t.classe,
    t.ordre,
    t.famille,
    t.sous_famille,
    t.group1_inpn,
    t.group2_inpn,
    t.lb_nom,
    t.nom_vern,
    t.url,
    s.altitude_min,
    s.altitude_max,
    s.date_min,
    s.date_max
   FROM gn_synthese.synthese s
     JOIN taxonomie.taxref t ON s.cd_nom = t.cd_nom
WITH DATA;
```

Figure 16 : Requête SQL de création de la vue matérialisée *vm_synthese*

En raison des nombreux filtres qui impliquent de garder différents niveaux de détail sur les données (année, lieu, taxonomie complète), certains graphes ne se prêtaient pas à la création d'une vue matérialisée, c'est pourquoi le nombre de vues est inférieur au nombre de graphiques/carte. En effet, la plupart des représentations nécessitent de faire appel à des requêtes comportant des groupements de données (par année, par taxon…) avec la clause SQL « GROUP BY ». Or, cette commande suppose une perte d'informations lors de l'affichage des résultats dans des vues matérialisées, empêchant la mise en place de filtres (conditions « WHERE ») par la suite.

Dans un second temps, les données nécessaires à l'interface ont été récupérées. La démarche a eu lieu en deux étapes. Tout d'abord, les modèles de données correspondant aux trois vues matérialisées ont été déclarés avec *SQLAlchemy* dans des classes Python, permettant de créer des nouveaux types d'objets. Pour chacune des vues, le nom du schéma, de la vue et des colonnes qu'elle contient ont été renseignés (Figure 17). Les modèles relatifs aux tables de GeoNature nécessaires dans ce module avaient déjà été construits dans le back-end général de l'application.

Figure 17 : Déclaration du modèle de données de la vue matérialisée *vm_synthese* dans une classe Python au niveau du back-end avec *SQLAlchemy*

```python
# vm_synthese
@serializable
class VSynthese(DB.Model):
    __tablename__ = "vm_synthese"
    __table_args__ = {"schema": "gn_dashboard"}
    id_synthese = DB.Column(DB.Unicode, primary_key=True)
    count_min = DB.Column(DB.Integer)
    count_max = DB.Column(DB.Integer)
    cd_nom = DB.Column(DB.Unicode)
    cd_ref = DB.Column(DB.Unicode)
    nom_vern = DB.Column(DB.Unicode)
    id_rang = DB.Column(DB.Unicode)
    regne = DB.Column(DB.Unicode)
    phylum = DB.Column(DB.Unicode)
    classe = DB.Column(DB.Unicode)
    ordre = DB.Column(DB.Unicode)
    famille = DB.Column(DB.Unicode)
    group1_inpn = DB.Column(DB.Unicode)
    group2_inpn = DB.Column(DB.Unicode)
    altitude_min = DB.Column(DB.Integer)
    altitude_max = DB.Column(DB.Integer)
    lon = DB.Column(DB.Unicode)
    lat = DB.Column(DB.Unicode)
    date_min = DB.Column(DB.DateTime)
    date_max = DB.Column(DB.DateTime)
```

Ensuite, les requêtes nécessaires à la récupération de ces données ont été implémentées, chacune dans une fonction Python. Au total, 8 requêtes ont été produites, avec des niveaux de complexité différents. La plupart d'entre elles ont nécessité la mise en place de conditions variables (clause SQL « WHERE »), afin de répondre aux contraintes possibles des différents filtres. Certaines ont également recours à des fonctions de calcul, parfois spatiales (*Cartographie*). Un exemple de requête est présenté dans la Figure 18. La fonction *SQLAlchemy* « DB.session.query » construit la requête SQL et l'applique à la base de données GeoNature, qui renvoie les résultats au back-end. Le paramétrage de la connexion à la base est effectué dans la configuration générale de l'application.

```python
blueprint = Blueprint("dashboard", __name__)

# Obtenir le nombre d'observations et le nombre de taxons pour chaque année
# vm_synthese
@blueprint.route("/synthese", methods=["GET"])
@json_resp
def get_synthese_stat():
    params = request.args
    q = DB.session.query(
        label("year", func.date_part("year", VSynthese.date_min)),
        func.count(VSynthese.id_synthese),
        func.count(distinct(VSynthese.cd_ref)),
    ).group_by("year")
    if ("selectedRegne" in params) and (params["selectedRegne"] != ""):
        q = q.filter(VSynthese.regne == params["selectedRegne"])
    if ("selectedPhylum" in params) and (params["selectedPhylum"] != ""):
        q = q.filter(VSynthese.phylum == params["selectedPhylum"])
    if "selectedClasse" in params and (params["selectedClasse"] != ""):
        q = q.filter(VSynthese.classe == params["selectedClasse"])
    if "selectedOrdre" in params and (params["selectedOrdre"] != ""):
        q = q.filter(VSynthese.ordre == params["selectedOrdre"])
    if "selectedFamille" in params and (params["selectedFamille"] != ""):
        q = q.filter(VSynthese.famille == params["selectedFamille"])
    if ("selectedGroup2INPN" in params) and (params["selectedGroup2INPN"] != ""):
        q = q.filter(VSynthese.group2_inpn == params["selectedGroup2INPN"])
    if ("selectedGroup1INPN" in params) and (params["selectedGroup1INPN"] != ""):
        q = q.filter(VSynthese.group1_inpn == params["selectedGroup1INPN"])
    if ("taxon" in params) and (params["taxon"] != ""):
        q = q.filter(VSynthese.cd_ref == params["taxon"])
    return q.all()
```

Figure 18 : Fonction Python exécutant la requête *SQLAlchemy* permettant de récupérer les données du *Graphe 1* au niveau du back-end

Comme on peut le voir dans la Figure 18, chaque fonction Python appelant une requête est associée à une URL structurée qui permet de récupérer les résultats. Dans le cas de la fonction « get_synthese_stat » présente ci-dessus et appliquée à la base de données GeoNature du PNE, les données sont accessibles à l'adresse suivante :

https://geonature.ecrins-parcnational.fr/dashboard/synthese

Ce système correspond à une API REST. Il s'agit d'un standard d'échange de données permettant la communication entre deux applications. Dans le cas de GeoNature, l'API permet l'échange de données entre la base de données et le front-end. Le standard REST est basé sur le protocole HTTP, présent au cœur du web, qui détermine la communication entre les clients et les serveurs à l'aide de requêtes (GET, PUT, POST, DELETE, etc.) et de réponses dîtes HTTP (Reese, 2018).

En bref, chaque fonction Python instaure à la fois le requêtage de données auprès de la base de données GeoNature, et définit également un chemin URL (route) pour permettre au front-end d'accéder aux résultats. Ces derniers sont présentés sous le format de données JSON (ou GeoJSON pour les informations géographiques). Ce type de format permet de représenter les informations de manière simple et très structurée.

Développement front-end

Le développement front-end, basé sur le framework JavaScript *Angular*, permet de produire et d'afficher les graphes et les cartes, ainsi que de coder la présentation de l'interface.

L'accès aux données à partir du front-end est opéré par la définition de services (fonctions TypeScript) qui font appel aux URL définies dans le back-end. Cette procédure est rendue possible par l'utilisation d'un service *Angular* appelé « HttpClient », permettant de construire et de lancer des appels HTTP (Alexander, 2019). Dans le cas du module « Tableau de bord », les requêtes HTTP ont été exécutées en mode GET : récupération de données dans la base. Une option d'ajout de paramètres avec la méthode « queryString » propre aux URL a également été implémentée, afin d'autoriser la personnalisation des requêtes (sélection de la clause SQL « WHERE » adéquate) selon les filtres renseignés sur l'interface (Figure 19).

```
getDataSynthese(params?) {
    let queryString = new HttpParams();
    if (params) {
        for (const key in params) {
            if (params[key]) {
                queryString = queryString.set(key, params[key]);
            }
        }
    }
    return this.httpClient.get<any>(AppConfig.API_ENDPOINT + "/" + ModuleConfig.MODULE_URL + "/synthese", { params: queryString })
}
```

Figure 19 : Service TypeScript (fonction) permettant de récupérer les données du *Graphe 1* au niveau du front-end

Chaque graphe ou carte a été codé dans un composant *Angular*. Tous les composants sont ensuite appelés dans un même composant parent pour l'affichage sur l'interface.

La librairie *ng2-charts* permettant de créer des graphiques s'utilise de manière simple : les graphes sont appelés dans le fichier HTML grâce à des composants *Angular* (balises HTML) nécessitant le renseignement de certains paramètres, comme la liste des données ou la liste des étiquettes du graphe - « barChartData » et « barChartLabels » sur la Figure 20. Ces listes sont quant à elles établies dans le fichier TypeScript avec les données récupérées par l'API. L'exemple d'utilisation pour l'affichage du *Graphe 1* est exposé dans la figure suivante (Figure 20) :

```
<div style="display: block">
    <canvas baseChart [datasets]="barChartData" [labels]="barChartLabels" [options]="barChartOptions"
        [colors]="barChartColors" [chartType]="barChartType">
    </canvas>
</div>
```

Figure 20 : Code HTML pour l'affichage du *Graphe 1*

L'utilisation de *Leaflet* sous *Angular* a été légèrement complexe car la documentation associée n'est écrite qu'en JavaScript. *Leaflet* a permis d'insérer des couches GeoJSON sur une carte afin de visualiser les mailles et les communes contenues dans la base de données de l'application. En effet, l'extension *PostGIS* de *PostgreSQL* permet le stockage d'objets géographiques (points, lignes, polygones) sous format binaire. Ces géométries ont été récupérées à l'aide d'une fonction Python de requêtage (back-end) renvoyant les données en GeoJSON. Celles-ci ont ensuite été insérées dans un composant *Angular* de GeoNature déjà existant permettant de les afficher sur une carte. La librairie *Leaflet* a également permis de colorer les zonages selon leur nombre d'observations ou leur nombre de taxons, grâce à la définition de classes de données. La carte comprend aussi une légende et un système d'affichage d'informations spécifiques au passage de la souris sur un zonage particulier.

Chaque graphique a été associé à un formulaire contenant les filtres qui permettent d'ajuster les données affichées. Les éléments de chaque formulaire, c'est-à-dire les listes déroulantes et éventuels sliders, ainsi que les fonctions d'accès aux informations transmises par les formulaires, ont

été déclarés dans le fichier TypeScript. Les formulaires sont affichés sur l'interface par des balises HTML, prenant en paramètres ces fonctions TypeScript. A chaque modification d'un élément du formulaire, l'information est ainsi envoyée du fichier HTML au fichier TypeScript, qui traite l'information et relance l'API avec les paramètres adéquats. Les nouvelles données récupérées sont ensuite insérées dans les balises HTML des graphes ou le composant de la carte, qui sont actualisés sur l'interface.

Le design de l'interface a été codé avec le langage CSS et la librairie *Bootstrap*. Cet outil très populaire est open source, et permet de gérer le style des boutons, des formulaires, des alertes, etc. *Bootstrap* a notamment été utilisé au sein du module « Tableau de bord » pour customiser les formulaires contenant les filtres et les cadres dans lesquels ils sont insérés.

La Figure 21 résume l'ensemble des éléments qui ont été exposés dans la mise en œuvre technique, ainsi que les liens qui les unissent :

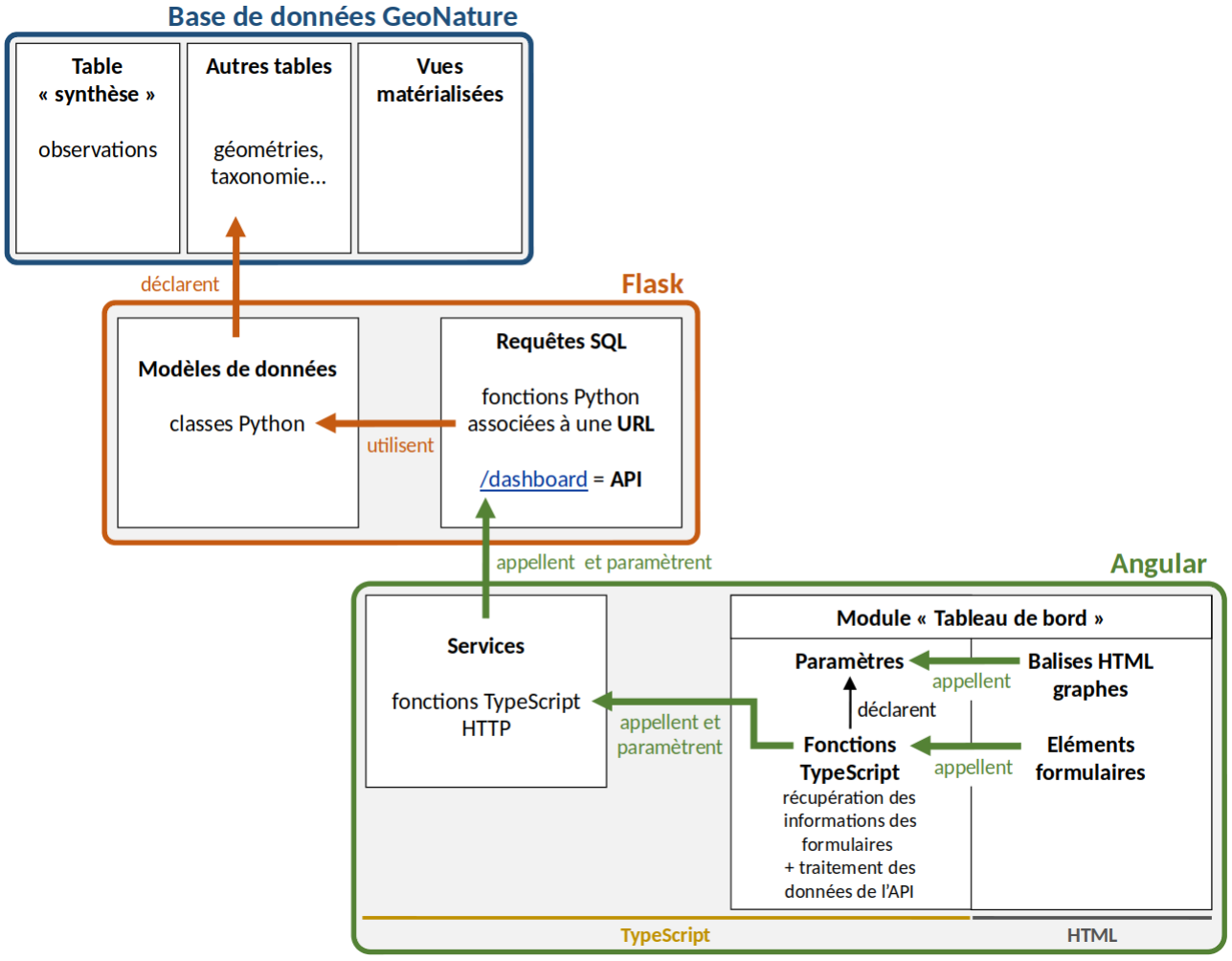


Figure 21 : Schéma récapitulatif de l'architecture technique du module "Tableau de bord"

b) Une volonté d'optimisation des performances du module

La difficulté la plus importante rencontrée lors du développement du module « Tableau de bord » a été la maximalisation des performances concernant le temps d'exécution des requêtes. En effet, la base de données GeoNature du PNE comporte plus de 1 300 000 données de faune et flore. De ce fait, certaines requêtes qui impliquent des traitements complexes nécessitaient, avant leur optimisation, plusieurs secondes, voire plusieurs minutes, pour afficher les résultats sur l'interface.

Afin de se rapprocher au maximum des conditions réelles et de tenir compte d'éventuelles montées en charge, j'ai travaillé en local avec un gros jeu de données issu de la base de production.

Une requête complexe relative au volet cartographique, impliquant des fonctions de traitements et de nombreuses conditions variables

La requête qui a posé le plus de problèmes en termes de performances est celle reliée à la *Cartographie*, car elle implique le traitement de données géospatiales plus lourdes que des données standards. Elle est aussi associée à un nombre de filtres plus important. Cette requête nécessite de fournir la liste des entrées de type : « nom du zonage (maille, commune, etc.) - géométrie du zonage (en GeoJSON) – nombre d'observations – nombre d'espèces observées ». Les zonages ne comportant aucune observation ne doivent pas être retournés. La Figure 22 présente un exemple partiel de résultats attendus :

	nom_zonage character varying(250)	geometrie_zonage text	nb_observations bigint	nb_especes bigint
1	Abriès-Ristolas	{"type":"MultiPolygon","coordinates":[[[[6.92252041658607	104	9
2	Aiglun	{"type":"MultiPolygon","coordinates":[[[[6.11111108638205	8	6
3	Aiguilles	{"type":"MultiPolygon","coordinates":[[[[6.8282820990321,	27	9
4	Albiez-Montrond	{"type":"MultiPolygon","coordinates":[[[[6.37950090162425	14	6
5	Allons	{"type":"MultiPolygon","coordinates":[[[[6.57835505897866	4	4
6	Allos	{"type":"MultiPolygon","coordinates":[[[[6.54528192086073	63	9
7	Ancelle	{"type":"MultiPolygon","coordinates":[[[[6.16953474965466	27	8
8	Archail	{"type":"MultiPolygon","coordinates":[[[[6.29991573054288	8	5
9	Arvieux	{"type":"MultiPolygon","coordinates":[[[[6.68156727742798	49	9
10	Authon	{"type":"MultiPolygon","coordinates":[[[[6.19649036043131	28	9

Figure 22 : Exemple de résultats attendus pour la requête SQL permettant de retourner les données de la *Cartographie*

Les étapes d'optimisation de cette requête sont explicitées dans les sections suivantes.

Création d'une vue matérialisée

J'ai tout d'abord pensé à la création d'une vue matérialisée afin de réaliser les traitements de données lourds dans la base. Cette vue affiche les résultats suivants pour le cas des communes (Figure 23) :

	nom_zonage character varying(250)	geometrie_zonage text	annee double precision	nb_observations bigint	nb_especes bigint
1	Abriès-Ristolas	{"type":"MultiPolygon","coordinates":[[[[6.	2001	1	1
2	Abriès-Ristolas	{"type":"MultiPolygon","coordinates":[[[[6.	2002	3	2
3	Abriès-Ristolas	{"type":"MultiPolygon","coordinates":[[[[6.	2003	1	1
4	Abriès-Ristolas	{"type":"MultiPolygon","coordinates":[[[[6.	2008	4	4
5	Abriès-Ristolas	{"type":"MultiPolygon","coordinates":[[[[6.	2009	2	2
6	Abriès-Ristolas	{"type":"MultiPolygon","coordinates":[[[[6.	2010	1	1
7	Abriès-Ristolas	{"type":"MultiPolygon","coordinates":[[[[6.	2011	2	2
8	Abriès-Ristolas	{"type":"MultiPolygon","coordinates":[[[[6.	2014	1	1
9	Abriès-Ristolas	{"type":"MultiPolygon","coordinates":[[[[6.	2015	3	3
10	Abriès-Ristolas	{"type":"MultiPolygon","coordinates":[[[[6.	2016	1	1

Figure 23 : Résultats de la vue matérialisée initialement créée pour la *Cartographie*

J'ai rapidement réalisé que cette vue ne permet pas d'appliquer les filtres *Rang taxonomique* et *Taxon* dans la requête *SQLAlchemy* du back-end. Ceci sous-entend que, si une vue matérialisée doit être créée, celle-ci doit comporter le même niveau de détail que celui de la table « synthèse ». En effet, cette table permet de connaître aussi bien l'année de chaque observation, que toute la taxonomie qui lui est associée, jusqu'au rang taxonomique le plus bas : l'espèce. Or, l'espèce fait partie du filtre *Rang taxonomique*, elle doit donc être enregistrée. De plus, cette requête est également spécialisée pour le cas des communes, ce qui aurait supposé de devoir construire une vue matérialisée par type de zonage.

Il a donc été conclu que la mise en place d'une vue matérialisée pour cette requête n'était pas adaptée, et qu'il était nécessaire d'écrire la requête complète avec les traitements de données lourds au niveau du back-end.

Utilisation d'une fonction de calcul spatiale simplifiant les géométries des objets géographiques

La première requête implémentée au niveau du back-end est exposée en Figure 24. Dans un souci de clarification, cette figure ne présente pas les clauses conditionnelles « WHERE » de la requête qui répondent aux contraintes des filtres *Rang taxonomique*, *Taxon* et *Période*. Le filtre *Type de zonage* est compris dans la variable « :type_code » (« COM » pour communes, « M1 » pour mailles 1km^2...). La première jointure interroge une table dans laquelle les intersections géographiques de chaque observation de la « synthèse » sont pré-calculées, afin d'alléger les temps de calcul. La jointure suivante donne accès aux géométries des zonages. La troisième jointure permet d'accéder au champ « type_code » nécessaire pour le choix du type de zonage. Enfin, la dernière jointure permet d'inclure toutes les informations taxonomiques relatives à chaque observation.

Figure 24 : Requête SQL initialement implémentée pour la *Cartographie* au niveau du back-end

```
SELECT a.area_name as nom_zonage,
       st_asgeojson(st_transform(a.geom, 4326)) as geometrie_zonage,
       count(s.id_synthese) as nb_observations,
       count(distinct t.cd_ref) as nb_especes
  FROM gn_synthese.synthese s
    JOIN gn_synthese.cor_area_synthese cor ON s.id_synthese=cor.id_synthese
    JOIN ref_geo.l_areas a ON cor.id_area=a.id_area
    JOIN ref_geo.bib_areas_types bib ON a.id_type=bib.id_type
    JOIN taxonomie.taxref t ON s.cd_nom=t.cd_nom
WHERE bib.type_code = :type_code
GROUP BY a.area_name, a.geom
```

Son temps d'exécution et d'affichage est d'environ 6 secondes avec la base de données GeoNature locale et le cas des communes. Ces entités géographiques sont particulièrement longues à afficher car leur géométrie peut être très complexe. Le GeoJSON renvoyé par l'API est très lourd en raison du nombre de sommets très important de chaque commune. C'est pourquoi il a été envisagé de recourir à la simplification du contour des zonages.

Pour ce faire, j'ai utilisé la fonction *PostGIS st_simplify* permettant de « simplifier » la géométrie de polylignes et de polygones. Son principe repose sur l'algorithme de Douglas-Peucker, qui supprime certains nœuds de la polyligne ou du polygone selon des règles détaillées en Annexe 9. La fonction *st_simplify* prend comme arguments la géométrie de l'élément à simplifier et le degré de simplification appelé tolérance, qui est un paramètre de l'algorithme de Douglas-Peucker. La requête est restée sensiblement la même, la fonction étant appliquée simplement au champ « geom » :

```
st_asgeojson(st_transform(st_simplify(a.geom, 50), 4326)) as geometrie_zonage
```

Le temps de chargement est descendu à environ 3 secondes et demie. Avec l'implémentation d'un spinner sur l'interface (indicateur de chargement des données), nous avons estimé que ce temps d'exécution était acceptable. Le degré de simplification de la fonction *st_simplify* a été ajouté dans les paramètres globaux du module, modifiables par l'administrateur.

Utilisation d'une sous-requête

Après avoir implémenté cette requête sur la véritable instance de l'application GeoNature du PNE, nous nous sommes rendu compte que l'affichage des résultats nécessitait plus d'une minute. Il aurait donc été préférable de tester le code sur des jeux de données réels, et non pas juste sur des jeux volumineux. Le gestionnaire de base de données m'a donc aidée à optimiser davantage la requête.

Une « sous-requête » a d'abord été insérée avec la commande « WITH ». Cette clause permet de segmenter une requête complexe en requêtes plus simples. Chaque instance « WITH » enregistre les résultats de la requête qu'elle contient dans une sorte de table temporaire qui n'existe que pour la

requête initiale, ce qui permet d'augmenter les performances (PostgreSQL, 2009). La sous-requête créée permet entre autre le calcul du nombre d'observations et du nombre d'espèces. Une deuxième sous-requête, faisant appel à une fonction, a également été implémentée dans le « WHERE », permettant de supprimer le recours à la 3ème jointure. Cette fonction, appelée « get_id_area_type », permet de récupérer directement l'identifiant d'un type de zonage en renseignant son « type_code ».

Pour améliorer également la requête dans les cas où plusieurs filtres précis sont renseignés, un champ indexé a été utilisé dans la clause « GROUP BY ». La création d'index sur certains champs est un concept qui permet encore une fois d'optimiser les temps d'exécution. Ce principe fonctionne comme le sommaire d'un livre : il est beaucoup plus facile de trouver les pages qui nous intéressent en consultant les sections du sommaire et leur numéro de page. Les champs « area_name » et « geom » présents dans la clause de regroupement ne sont pas indexés, c'est pourquoi nous avons arrangé la requête de sorte que le champ « id_area », pourtant inutile à la requête de base, se retrouve dans le « GROUP BY ».

Ces derniers changements ont abouti à la requête finale exposée en Figure 25 :

```
WITH count AS (
  SELECT cor.id_area, count(distinct s.id_synthese) as nb_obs, count(distinct t.cd_ref) as nb_esp
    FROM gn_synthese.cor_area_synthese cor
      JOIN gn_synthese.synthese s ON s.id_synthese=cor.id_synthese
      JOIN taxonomie.taxref t ON s.cd_nom=t.cd_nom
  WHERE cor.id_area IN (SELECT id_area FROM ref_geo.l_areas WHERE id_type = ref_geo.get_id_area_type(:type_code))
  GROUP BY cor.id_area
)
SELECT a.area_name as nom_zonage,
       st_asgeojson(st_transform(st_simplify(a.geom, 50), 4326)) as geometrie_zonage,
       c.nb_obs as nb_observations,
       c.nb_esp as nb_especes
  FROM ref_geo.l_areas a
    JOIN count c ON a.id_area = c.id_area
```

Figure 25 : Requête SQL finale implémentée pour la *Cartographie* au niveau du back-end

Le temps d'exécution et d'affichage est descendu à environ 6 secondes.

Ce résultat restant perfectible, une piste d'amélioration qui pourrait être explorée concerne le partitionnement des tables en base de données. Ce processus permet de découper une table volumineuse en plusieurs « tables » plus petites de manière organisée et structurée, optimisant ainsi les performances des requêtes en diminuant le nombre de lignes à inspecter (PostgreSQL, 2017).

JavaScript, un langage asynchrone

JavaScript, le langage utilisé par *Angular*, est un langage dit asynchrone. Cela signifie qu'il « n'attend pas » que les lignes de code aient terminé leur exécution pour lancer les suivantes. Cette notion a posé problème pour l'établissement du *Graphe 4*.

En effet, ce graphique nécessite de récupérer trois types de données : le nombre de taxons recontactés, le nombre de taxons non recontactés et le nombre de nouveaux taxons (cf. Tableau 11 pour les définitions). Chacune de ces informations est complexe à obtenir car elle nécessite de comparer les données d'une année considérée à celles des années précédentes. Ceci est réalisable avec les clauses SQL « INTERSECT » et « EXCEPT » qui ne seront pas détaillées ici.

J'ai tout d'abord écrit trois requêtes différentes, chacune associée à sa propre URL. Lors de l'élaboration du graphique, les données doivent être mentionnées dans un ordre précis afin d'être associées au bon élément de légende. Le requêtage HTTP en JavaScript étant asynchrone, il est impossible de savoir quel service (URL) renverra ses données en premier, quel que soit l'ordre dans lequel ils sont appelés. Le graphique s'affiche donc rarement de manière correcte.

Dans un second temps, j'ai donc construit une requête permettant de fournir à elle seule les trois types de données. Celle-ci est présente dans la figure suivante (Figure 26) :

```
WITH recontactees AS
    (SELECT DISTINCT cd_ref FROM gn_synthese.synthese s JOIN taxonomie.taxref t ON t.cd_nom=s.cd_nom WHERE date_part('year', date_min) < :selectedYear
    INTERSECT
    SELECT DISTINCT cd_ref FROM gn_synthese.synthese s JOIN taxonomie.taxref t ON t.cd_nom=s.cd_nom WHERE date_part('year', date_min) = :selectedYear),
non_recontactees AS
    (SELECT DISTINCT cd_ref FROM gn_synthese.synthese s JOIN taxonomie.taxref t ON t.cd_nom=s.cd_nom WHERE date_part('year', date_min) < :selectedYear
    EXCEPT
    SELECT DISTINCT cd_ref FROM gn_synthese.synthese s JOIN taxonomie.taxref t ON t.cd_nom=s.cd_nom WHERE date_part('year', date_min) = :selectedYear),
nouvelles AS
    (SELECT DISTINCT cd_ref FROM gn_synthese.synthese s JOIN taxonomie.taxref t ON t.cd_nom=s.cd_nom WHERE date_part('year', date_min) = :selectedYear
    EXCEPT
    SELECT DISTINCT cd_ref FROM gn_synthese.synthese s JOIN taxonomie.taxref t ON t.cd_nom=s.cd_nom WHERE date_part('year', date_min) < :selectedYear)
SELECT count(cd_ref) FROM recontactees
UNION ALL
SELECT count(cd_ref) FROM non_recontactees
UNION ALL
SELECT count(cd_ref) FROM nouvelles
```

Figure 26 : Requête SQL implémentée pour le *Graphe 4* au niveau du back-end

Cependant, cette requête difficilement simplifiable possède un temps d'exécution très long. Je n'ai pas eu le temps d'optimiser ce sujet, mais il serait intéressant d'avoir recours aux *Promises* d'*Angular*, qui permettraient de gérer les temps de récupération des données de trois requêtes indépendantes. En effet, les « états » des *Promises* pourraient autoriser le lancement d'une requête seulement après l'exécution complète d'une autre. Les requêtes HTTP seraient donc synchrones.

c) Un module perfectible voué à être enrichi par la communauté GeoNature

Les résultats des tests utilisateurs ont été analysés, regroupés et classés dans le tableau suivant (Tableau 12) :

Tableau 12 : Bilan des tests utilisateurs concernant le module "Tableau de bord"

Commentaires sur le module	Moyenne des notes au questionnaire
Intuitif, paramétrable	4,75/5
Améliorations globales	
Permettre à l'administrateur de cacher certains graphes.	
Accéder au module sur la tablette (pour les agents).	
Ajouter des options dans le filtre *Rang taxonomique* : pouvoir filtrer par guildes (ex : oiseaux forestiers), espèces patrimoniales.	
Intégrer des métadonnées dans les graphes : objectifs, types de financement…	
Améliorations sur le *Graphe 1*	
Les deux échelles (nombre d'observations et nombre de taxons) sont perturbantes. Ajouter un filtre plutôt qui permettrait de passer d'une représentation à l'autre.	
Ajouter la courbe cumulée des observations.	
Pouvoir zoomer sur les données.	
Améliorations sur la *Cartographie*	
Pouvoir cliquer sur un zonage et accéder à des informations plus précises : habitats, liste des dernières observations…	
Ajouter une légende dynamique, qui s'ajuste de manière statistique (méthode des quantiles) aux données qui sont renvoyées suite aux changements de filtres. Probablement possible à implémenter au niveau du back-end avec les fonctions statistiques de l'extension *Numpy* pour Python.	
Améliorations sur le *Graphe 2*	
Ajouter un deuxième disque avec le nombre de taxons (comme sur le site de l'INPN).	
Mentionner les données brutes et pas seulement les pourcentages, car certaines parts sont trop petites et donc non visibles.	
Améliorations sur le *Graphe 3*	
Ajouter les jeux de données également, en plus des cadres d'acquisition. Ajouter un filtre éventuellement pour passer de l'un à l'autre.	
Pouvoir zoomer sur les données.	

Réaliser plutôt un histogramme, car scientifiquement ce n'est pas correcte de réaliser une courbe avec des années en abscisses.
Améliorations sur le *Graphe 4*
Mentionner les données brutes et pas seulement les pourcentages, car certaines parts sont trop petites et donc non visibles.
Pouvoir sélectionner une période avec un slider (plusieurs années) et pas seulement une année.
Réaliser plutôt un histogramme des pourcentages par année. Les barres doivent être paramétrées à 100% et sont divisées selon les trois catégories de données : taxons recontactés, taxons non recontactés et nouveaux taxons. Dans ce cas on peut ajouter les filtres *Rang taxonomique* et *Taxon*.
Améliorations ergonomiques
Créer une page d'accueil avec des blocs.
Ajouter des encadrés d'aide à l'utilisation (ex : « Cliquez sur les éléments de légende pour les faire apparaître/disparaître sur le graphe »).
Ajouter des encadrés avec des définitions pour les termes scientifiques.
Diminuer la largeur des contours des zonages sur la *Cartographie*.
Ne pas afficher des légendes à l'envers sur les camemberts.
Nouvelles idées
Carte de présence pour une espèce donnée avec une année seuil : 2 classes « présent avant » et « présent après » avec couleurs ou sigles différents (exemple en Annexe 10).
Histogramme du nombre d'observations par observateur, pour les 10 observateurs les plus actifs.
Histogramme du nombre d'observations par espèce, pour les 5 espèces les plus observées et les 5 espèces les moins observées.
Histogramme du nombre d'observations et nombre de taxons par milieu (forêts, prairies…).
Camembert de la répartition des observations selon les différents modules.
Camembert de la répartition des données validées/non validées pour une année ou sur une période.
Courbe d'évolution de l'altitude en fonction du temps pour une espèce donnée.
Créer un « Tableau de bord » par module de saisie afin d'accéder à des représentations plus précises, et rendre possible la création de cartes sous la forme présence/absence (il n'y a que les données protocolées qui comportent des informations d'absence).
Faire en sorte que certains filtres soient applicables à l'ensemble des graphes en même temps.

Les remarques concernant l'ergonomie de l'outil ont été considérées immédiatement.

Ces résultats montrent que le module « Tableau de bord » répond aux besoins, mais reste perfectible, à tous les niveaux. Les performances sont encore à optimiser pour certains graphes, comme cela a été étudié dans la partie précédente.

Les nouvelles idées apportées par certains utilisateurs, mais également les besoins qui n'ont pas pu être considérés dans le développement, prouvent que l'outil est voué à être enrichi par la communauté GeoNature et les structures impliquées dans la protection de la biodiversité qui l'utiliseront. Un projet plus ambitieux serait de créer un module encore plus souple, permettant aux utilisateurs de construire leurs propres graphiques et cartes en direct, à l'aide d'une suite d'étapes d'élaboration à suivre : type de graphe, valeurs en abscisses, valeurs en ordonnées…

Réalisation de tests d'installation

Les fichiers de configuration du module ont été complétés afin de rendre l'outil installable. Les requêtes SQL permettant de créer les vues matérialisées ont été renseignées dans un fichier SQL qui s'exécute lors de l'installation. Le front-end de chaque module étant packagé comme une librairie JavaScript, les noms des librairies installées (*ng2-charts* pour les graphes et *ng2-nouislider* pour les

sliders) ont également été déclarées dans le fichier listant les dépendances du module (*package.json*), afin qu'elles soient installées en même temps que celui-ci.

Enfin, afin de vérifier le bon déroulement de l'installation, le module a été supprimé de mon poste et réinstallé avec la procédure GeoNature destinée à l'ajout d'un module. L'opération s'est déroulée sans problème. Ceci a été réalisé avec l'utilisation du GitHub du module présent à l'adresse suivante : https://github.com/PnX-SI/gn_module_dashboard.

3. Un protocole automatique de notation des données saisies élaboré dans la base de données de GeoNature

a) Les profils types de taxon, des référentiels pertinents

Le protocole de validation du CEN Languedoc-Roussillon est celui qui se rapproche le plus de la voie empruntée par le PNE. Cette association de protection de la nature est, par ailleurs, d'envergure plutôt similaire au Parc et se charge de missions semblables : gestion et protection d'espaces naturels et de leur biodiversité. Il est donc pertinent de s'inspirer des travaux réalisés par cet organisme. De plus, l'efficacité de ce protocole est avérée. Les experts vérifiant les données de la plateforme en sont satisfaits, affirmant que la méthode permet de faire ressortir les données marginales ou originales.

Pour des raisons supplémentaires, la caractérisation de chaque taxon par l'établissement d'un profil type basé sur les données validées semble constituer un référentiel fiable. En effet, la comparaison des données saisies à des référentiels nationaux ou départementaux ne permet pas toujours de tenir compte par exemple des disparités géologiques, topographiques ou climatiques des zones qu'ils englobent. De plus, l'élaboration d'un référentiel par champ, en comparaison avec un référentiel par taxon, limite le nombre de champs testés, car elle implique de réfléchir à toutes les valeurs ou plages de valeurs possibles du champ et de créer une table par champ considéré. Ainsi, les fiches d'identité mises en place par le PNE doivent être conservées et surtout complétées.

Des profils types pouvant être enrichis

Les paramètres et critères considérés comme intéressants par le PNE dans l'établissement des profils types de taxons sont peu nombreux. L'analyse de l'existant a permis d'identifier d'autres éléments de référence pertinents pour compléter cette liste. Le Tableau 13 énumère les éléments que le PNE devrait prendre en compte ainsi que le format de valeurs possible pour chacun. Ils ont été choisis selon leur possibilité de mise en œuvre au sein de la base de données du Parc.

Tableau 13 : Liste et description des paramètres à considérer dans le protocole de détection automatique de données atypiques du PNE

Elément de référence	Valeur(s) de l'élément pour un taxon considéré	Seuil envisageable	Commentaires
Paramètres			
Lieu d'observation	Liste de communes	-	Implémenter plutôt une fonction spatiale hors du profil permettant de créer une zone tampon circulaire autour de la donnée (*st_buffer*) et vérifier si une observation du même taxon est comprise à l'intérieur.

Périodes d'observation	Liste de semaines	± X jours	
Altitudes d'observation	Liste de plages d'altitude de 100m	± X mètres	
Habitats préférentiels	-	-	A l'avenir : utiliser le référentiel HABREF du SINP
Effectif (nombre d'individus) de l'observation	Plage d'effectifs	± X	
Observateur	Liste d'observateurs ayant au moins une donnée validée pour le taxon considéré	-	
Critères d'orientation			
Nombre d'observations validées total	Calcul du nombre d'observations validées total pour le taxon considéré	< X	Orienter la validation manuelle sur les taxons ayant peu de données validées.
Statut spécifique du taxon	Protégé, en danger, rare, peu connu	-	Orienter la validation manuelle sur les observations de taxons à statut critique.
Difficulté de détermination du taxon	% de difficulté	-	Orienter la validation manuelle sur les observations de taxons habituellement difficiles à déterminer (risques d'erreur plus élevés).
Niveau de certitude de l'identification renseigné par l'observateur	% de certitude	-	Orienter la validation manuelle sur les observations douteuses pour l'observateur.

Ces éléments de référence forment une base à appliquer à chaque taxon. Cependant, l'état de l'art a mis en évidence l'intérêt de décliner ces éléments, valeurs et seuils par pôle thématique (groupe de taxons) dans le but d'obtenir une meilleure précision concernant le score attribué aux données. Cette opération doit être réalisée avec la collaboration de chargés de mission scientifiques ou de naturalistes.

b) Un calcul de score qui nécessite d'être précisé

L'état de l'art a prouvé que l'attribution d'un score à chaque donnée est une alternative possible à l'affectation d'un niveau de validité. Les résultats de chaque test, « conforme » et « non conforme », peuvent être associés à une note. Ce processus doit, une nouvelle fois, être effectué par des experts. Ces derniers doivent décider si tous les paramètres et critères considérés dans les tests peuvent être d'importance égale, ou bien si des poids différents doivent être établis. L'élaboration du score final à partir des résultats de chaque test est également à prévoir. La tendance serait la suivante : plus le score est élevé, plus il est probable que la donnée soit fiable. Enfin, les chargés de missions scientifiques doivent déterminer un seuil en dessous duquel un score est estimé trop faible. Les scores inférieurs à ce seuil seront mis en évidence à travers une alerte, une notification ou un signe distinctif sur l'application.

Ainsi, cette section reste encore très incomplète et méritera d'être approfondie par les réflexions de scientifiques.

c) Un protocole automatique exécuté par des actions déclenchées en base de données

La partie technique du protocole de validation automatique peut être entièrement réalisée en base de données, comme ce qui a été établi par le CEN Languedoc-Roussillon.

Dans un premier temps, les fiches d'identité de taxon sont élaborées dans une nouvelle table à l'aide d'une requête SQL complexe (Figure 27), prenant en compte les éléments de référence listés dans le Tableau 13. Cette requête nécessitera d'être complétée.

```
CREATE TABLE gn_synthese.references_validation AS
  SELECT cd_ref,
         count(s.id_synthese) as nb_observations_total,
         array_agg(DISTINCT id_area) AS liste_communes,
         array_agg(DISTINCT extract(week FROM COALESCE(date_min, date_max))) AS liste_semaines,
         array_agg(DISTINCT (altitude_min::integer/100) ORDER BY (altitude_min::integer/100) ASC) AS liste_altitudes,
         MIN(count_min) AS borne_inf_eff,
         MAX(count_max) AS borne_sup_eff,
         array_agg(DISTINCT observers) AS liste_observateurs
    FROM gn_synthese.synthese s
      JOIN gn_synthese.cor_area_synthese cor ON s.id_synthese=cor.id_synthese
      JOIN taxonomie.taxref t ON s.cd_nom=t.cd_nom
  WHERE cor.id_area IN (SELECT id_area FROM ref_geo.l_areas WHERE id_type = ref_geo.get_id_area_type('COM'))
  GROUP BY cd_ref
ALTER TABLE gn_synthese.references_validation ADD CONSTRAINT reference_validation_pkey PRIMARY KEY(cd_ref)
```

Figure 27 : Requête SQL de création de la table "reference_validation" (profils types)

Les paramètres « habitats préférentiels », « difficulté de détermination du taxon » et « niveau de certitude de l'identification renseigné par l'observateur » n'ont pas été pris en compte car ils ne sont actuellement pas implémentés dans la base de données GeoNature. Le « statut spécifique du taxon » est un cas plus complexe qui n'a pas pu être exploré dans le temps imparti. Ces éléments seront donc à considérer dans les futures réflexions. De plus, cette requête reste à améliorer car, comme cela a été précisé dans la première partie, certaines données de la base sont très anciennes et donc peu précises. Les résultats affichés dans la table sont parfois aberrants.

Ensuite, la création d'un trigger permet de comparer chaque donnée saisie au profil type du taxon concerné (Figure 28) :

Figure 28 : Requête SQL de création du trigger permettant la validation automatique des données

```
CREATE TRIGGER validation_auto
  AFTER INSERT /* on exécute la procédure à chaque insertion */
  ON gn_synthese.synthese
  FOR EACH ROW
  EXECUTE PROCEDURE gn_synthese.validation_auto()
```

La fonction « validation_auto() » exécutée par le trigger est présente en Figure 29. Elle permet d'associer le score final à la donnée, dans un champ particulier de la « synthèse » :

```
CREATE OR REPLACE FUNCTION gn_synthese.validation_auto()
  RETURNS trigger AS
$BODY$
BEGIN
UPDATE gn_synthese.synthese SET score_validation = gn_synthese.calcul_score(id_synthese) WHERE id_synthese = NEW.id_synthese;
RETURN NULL;
END;
$BODY$
  LANGUAGE plpgsql VOLATILE
```

Figure 29 : Requête SQL de création de la fonction "validation_auto()"

Cette première fonction fait elle-même appel à une deuxième fonction « calcul_score() » chargée de calculer ce score et de le retourner (Figure 30).

```
CREATE OR REPLACE FUNCTION gn_synthese.calcul_score(integer)
  RETURNS text AS
$BODY$
DECLARE
   var_id_synthese alias FOR $1;
   var_score text;
BEGIN
  WITH
  /* L'observation qui nous intéresse, et les colonnes utiles à son examen */
  observation AS (
    SELECT s.id_synthese, cd_ref, id_area, date_min, date_max,
           altitude_min, altitude_max, count_min, count_max, observers
      FROM gn_synthese.synthese s
        JOIN gn_synthese.cor_area_synthese cor ON s.id_synthese=cor.id_synthese
        JOIN taxonomie.taxref t ON s.cd_nom=t.cd_nom
    WHERE cor.id_area IN (SELECT id_area FROM ref_geo.l_areas WHERE id_type = ref_geo.get_id_area_type('COM')) AND s.id_synthese = var_id_synthese),
  /* Les valeurs de référence pour le taxon concerné */
  reference AS (
    SELECT cd_ref, nb_observations_total, liste_communes, liste_semaines,
           liste_altitudes, borne_inf_eff, borne_sup_eff, liste_observateurs
      FROM gn_synthese.references_validation
      JOIN observation USING(cd_ref)),
  /* La confrontation des valeurs saisies aux valeurs de référence */
  bilan AS (
    SELECT id_synthese,
           CASE WHEN nb_observations_total < X THEN 1 ELSE 0 END AS nb_observations,
           CASE WHEN id_area = ANY(liste_communes) THEN 1 ELSE 0 END AS commune,
           CASE WHEN EXTRACT(week FROM COALESCE(date_min, date_max)) = ANY(liste_semaines) THEN 1 ELSE 0 END AS semaine,
           CASE WHEN (altitude_min/100)::integer = ANY(liste_altitudes) AND altitude_min IS NOT NULL THEN 1 ELSE 0 END AS altitude,
           CASE WHEN count_min > borne_inf_eff AND count_max < borne_sup_eff THEN 1 ELSE 0 END AS effectif,
           CASE WHEN observers = ANY(liste_observateurs) THEN 1 ELSE 0 END AS observateur
      FROM observation JOIN reference USING(cd_ref))
  /* Mise en forme du résultat retourné */
  SELECT CONCAT('Nombre observations: ', nb_observations,
                ', Commune: ', commune,
                ', Semaine: ', semaine,
                ', Altitude: ', altitude,
                ', Effectif: ', effectif,
                ', Observateur: ', observateur) into var_score
    FROM bilan;
RETURN var_score;
END;
$BODY$
  LANGUAGE plpgsql VOLATILE;
```

Figure 30 : Requête SQL de création de la fonction "calcul_score"

L'utilisation de la future fonction spatiale permettant de vérifier si une observation du même taxon a été opérée dans les alentours (*st_buffer*) devra être appelée dans cette fonction « calcul_score() ».

Par ailleurs, un autre trigger devra être implémenté afin de mettre à jour régulièrement la table « references_validation » contenant l'ensemble des profils types. En effet, les nouvelles données saisies et validées devront être prises en compte dans les calculs. Cette partie pourra également s'inspirer du travail réalisé par le CEN Languedoc-Roussillon.

La Figure 31 synthétise l'ensemble des étapes techniques qui viennent d'être énumérées et les liens qui les unissent :

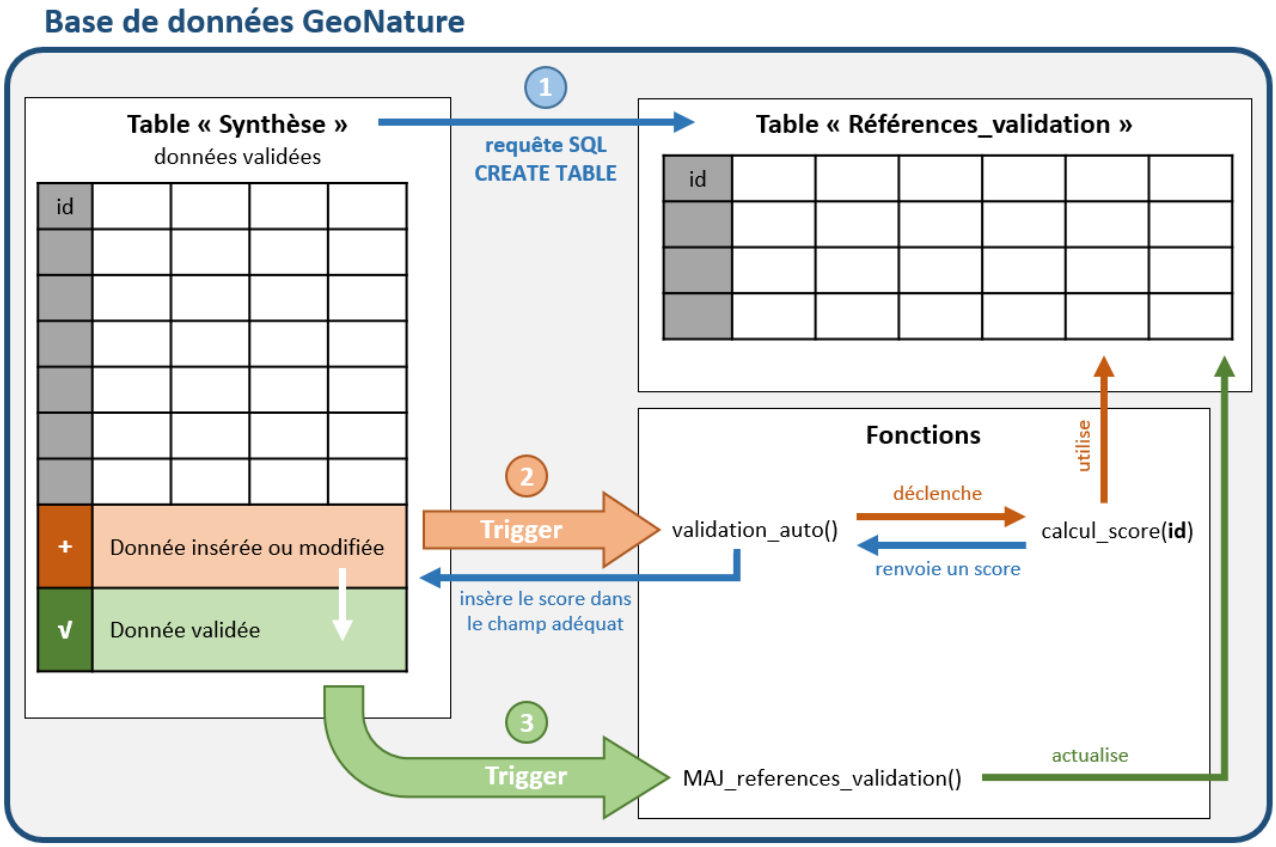


Figure 31 : Schéma récapitulatif du protocole automatique d'évaluation scientifique des données de GeoNature

A l'avenir, il serait intéressant de développer une API niveau back-end permettant de calculer le score d'une observation. Cela permettrait d'envoyer une alerte à l'observateur avant la validation d'une saisie de donnée afin que celui-ci reconsidère son observation. Une deuxième API pourrait également être implémentée afin de récupérer les données de la fiche d'identité de chaque taxon pour les afficher dans le module « Tableau de bord » par exemple, à titre d'information.

Le module « Tableau de bord » développé est un outil interactif et paramétrable grâce à l'utilisation d'un framework dynamique et l'implémentation de nombreux filtres. Ce caractère modulable qui est une force représente l'enjeu principal de son élaboration. En effet, les requêtes SQL utilisées peuvent être lourdes et complexes, impliquant ainsi une optique permanente d'optimisation des performances. Ce point a beaucoup été travaillé, mais il est voué à être enrichi par la communauté GeoNature, tout comme l'ensemble du module. En effet, ce « Tableau de bord » a été créé pour être manipulé également par d'autres structures de conservation de la biodiversité, qui exprimeront probablement d'autres besoins à son sujet.

Le protocole de détection automatique de données atypiques du PNE a été validé et complété par l'état de l'art réalisé. Sur le plan théorique, l'établissement des fiches d'identité a été enrichi par de nouveaux éléments de référence. Concernant le volet technique, des scripts SQL ont été entamés en base de données. En revanche, la détermination d'un calcul de score et la déclinaison du protocole par pôle thématique restent à définir par des experts.

Conclusion

Les données de l'application naturaliste GeoNature représentent une base de données complexe et dense, avec des informations hétérogènes. Les besoins identifiés relatifs à la synthétisation de ces données ont montré que le développement d'un module de reporting de données, présenté sous la forme d'un tableau de bord interactif, est adapté pour obtenir une vision claire des données saisies. Le module GeoNature amorcé présente des graphiques et une carte ajustables à l'aide de différents filtres, afin de fournir des informations à différentes échelles : groupes taxonomiques, espèces, communes, mailles, périodes précises… Cet outil conserve l'aspect générique propre à l'application tout en s'adaptant à n'importe-quelle structure productrice de données et à n'importe-quel utilisateur (scientifique ou grand public). Intégré au cœur de l'application et mis à la disposition de tous, ce module encore perfectible est destiné à être optimisé et enrichi par la communauté GeoNature, de par sa structuration en composants. Il mériterait notamment d'être décliné par protocole de saisie, afin d'obtenir des informations de synthèse plus précises, telles que des cartes de présence/absence permettant d'enrichir les rapports de suivi des espèces.

La dynamique de partage des données de biodiversité lancée par le SINP a poussé les organismes de protection de la nature à élaborer des indicateurs de fiabilité pour qualifier leurs données, afin de garantir une meilleure utilisation de celles-ci. La validation manuelle des informations étant très chronophage, plusieurs structures ont imaginé des systèmes de validation automatique dans le but de concentrer les efforts sur les données atypiques. L'étude de ces protocoles existants a permis au Parc national des Écrins de compléter son propre travail sur le sujet et d'amorcer un processus automatique de notation scientifique des observations dans la base de données de GeoNature. Ce projet nécessite néanmoins d'être renforcé par des connaissances scientifiques sur les espèces et les groupes taxonomiques, afin de gagner en efficacité.

Les projets d'aide au suivi de l'évolution de la biodiversité, tels que GeoNature, sont en expansion à l'international. Il y a plusieurs années, une organisation non gouvernementale aux États-Unis appelée *Wild Me* a dévoilé son projet de plateforme open source développée pour lutter contre l'extinction des espèces. Cet outil repose sur un algorithme d'intelligence artificielle capable d'analyser des photos ou des vidéos d'individus et de déterminer une sorte d'empreinte digitale propre à chacun. Le Parc national des Écrins assure déjà le suivi de certaines espèces à l'aide de colliers ou balises GPS, telles que le Bouquetin des Alpes ou les Gypaètes Barbus, mais ce processus lui permettrait d'aller plus loin en recensant les individus de plusieurs espèces pour lesquelles le suivi GPS n'est pas possible et d'assurer un suivi de la dynamique des populations (Wild Me, 2015).

Bibliographie

ALEXANDER, Will, 2019. Développez des applications web avec Angular. In : *Openclassrooms Cours* [en ligne]. 27 août 2019. Disponible à l'adresse : https://openclassrooms.com/fr/courses/4668271-developpez-des-applications-web-avec-angular.

BLONDIN, Alexis, 2013. Article : différence entre le développeur front-end et le développeur back-end. In : *Alticreation Articles* [en ligne]. 11 juillet 2019. Disponible à l'adresse : https://www.alticreation.com/difference-developpeur-front-end-et-developpeur-back-end/.

BOSSAERT, Mathieu, 2015. Le SI du CEN LR : validation automatique de données. In : *Conservatoire d'espaces naturels Languedoc-Roussillon Articles* [en ligne]. 7 août 2019. Disponible à l'adresse : https://si.cenlr.org/validation_automatique_de_donnee.

GENIS J.M. et al., 2017. Le processus de gestion des observations floristiques au CBNA : de la réception des données à leur diffusion dans les SINP. Version 1.0. In : *CBNA Portail documentaire* [en ligne]. 7 août 2019. Disponible à l'adresse : http://www.cbn-alpin-biblio.fr/Record.htm?idlist=3&record=19180465124919086479.

INPN, 2016a. Données d'observation sur les espèces – Validation : la validation des données de l'INPN. In : *Données d'observation sur les espèces. Références. Validation* [en ligne]. 29 juillet 2019. Disponible à l'adresse : https://inpn.mnhn.fr/programme/donnees-observations-especes/references/validation.

INPN, 2016b. Glossaire. In : *Glossaire* [en ligne]. 23 juillet 2019. Disponible à l'adresse : https://inpn.mnhn.fr/informations/glossaire/liste/e.

INPN, 2016c. Données d'observation sur les espèces – Sensibilité : les données sensibles : un cas particulier de restriction de la diffusion. In : *Données d'observation sur les espèces. Références. Sensibilité* [en ligne]. 29 juillet 2019. Disponible à l'adresse : https://inpn.mnhn.fr/programme/donnees-observations-especes/references/sensibilite.

LE TELLIER, Valentin, 2019. Protocole de validation des données du Système d'Information sur la Nature et les Paysages de La Réunion (SINP 974) - Volet occurrences de taxons. Version 1.1.0. In : *Nature France. La Réunion. Protocole de validation* [en ligne]. 9 août 2019. Disponible à l'adresse : http://www.naturefrance.fr/la-reunion/protocole-de-validation.

LECHÉMIA, Théo. 2016. *Rapport de stage : création d'un atlas dynamique de la faune et de la flore au Parc national des Écrins*. Grenoble : Université Grenoble-Alpes ; Parc national des Écrins

Les parcs nationaux de France, 2014. Le métier de garde-moniteur. In : *Des connaissances. Protection et réglementation* [en ligne]. 16 juillet 2019. Disponible à l'adresse : http://www.parcsnationaux.fr/fr/des-connaissances/protection-et-reglementation/le-metier-de-garde-moniteur.

Les parcs nationaux de France, 2017. Les 10 parcs nationaux et le projet de parc national : fiche d'identité. In : *Des découvertes. Les parcs nationaux de France* [en ligne]. 20 juin 2019. Disponible à l'adresse : http://www.parcsnationaux.fr/fr/des-decouvertes/les-parcs-nationaux-de-france/les-10-parcs-nationaux-et-le-projet-de-parc-national.

MAILLARD, Donovan, 2019. La gestion des données de biodiversité du PNE. In : *GeoNature Documents* [en ligne]. 5 juin 2019. Disponible à l'adresse : https://geonature.fr/documents/2019-04-CS-SI-Biodiversite-PNE.pdf.

MONCHICOURT, Camille, 2018. GeoNature, un outil open source développé par les Parcs nationaux français. In : *GeoNature Documents* [en ligne]. 26 mai 2019. Disponible à l'adresse : https://geonature.fr/documents/2018-01-GeoNature.pdf.

PostgreSQL, 2009. Requêtes WITH (Common Table Expressions). In : *PostgreSQL Documentation 8.4. Langage SQL. Requêtes. Requêtes WITH* [en ligne]. 28 août 2019. Disponible à l'adresse : https://docs.postgresql.fr/8.4/queries-with.html.

PostgreSQL, 2017. Partitionnement de tables. In : *PostgreSQL Documentation 10. Langage SQL. Définition des données. Partitionnement de tables* [en ligne]. 29 août 2019. Disponible à l'adresse : https://docs.postgresql.fr/10/ddl-partitioning.html.

REESE, Emily, 2018. Utilisez des API REST dans vos projets web. In : *Openclassrooms Cours* [en ligne]. 22 août 2019. Disponible à l'adresse : https://openclassrooms.com/fr/courses/3449001-utilisez-des-api-rest-dans-vos-projets-web.

ROBERT, Solène, 2015. GT Validation des données d'occurrence du SINP : recensement de l'existant pour la validation des données d'occurrence du SINP. In : *Nature France Ressources* [en ligne]. 5 août 2019. Disponible à l'adresse : http://www.naturefrance.fr/sites/default/files/fichiers/ressources/pdf/spn_2015_-_44_-_recensementexistantvalidationdonneesoccurrence_sinp_v1.pdf.

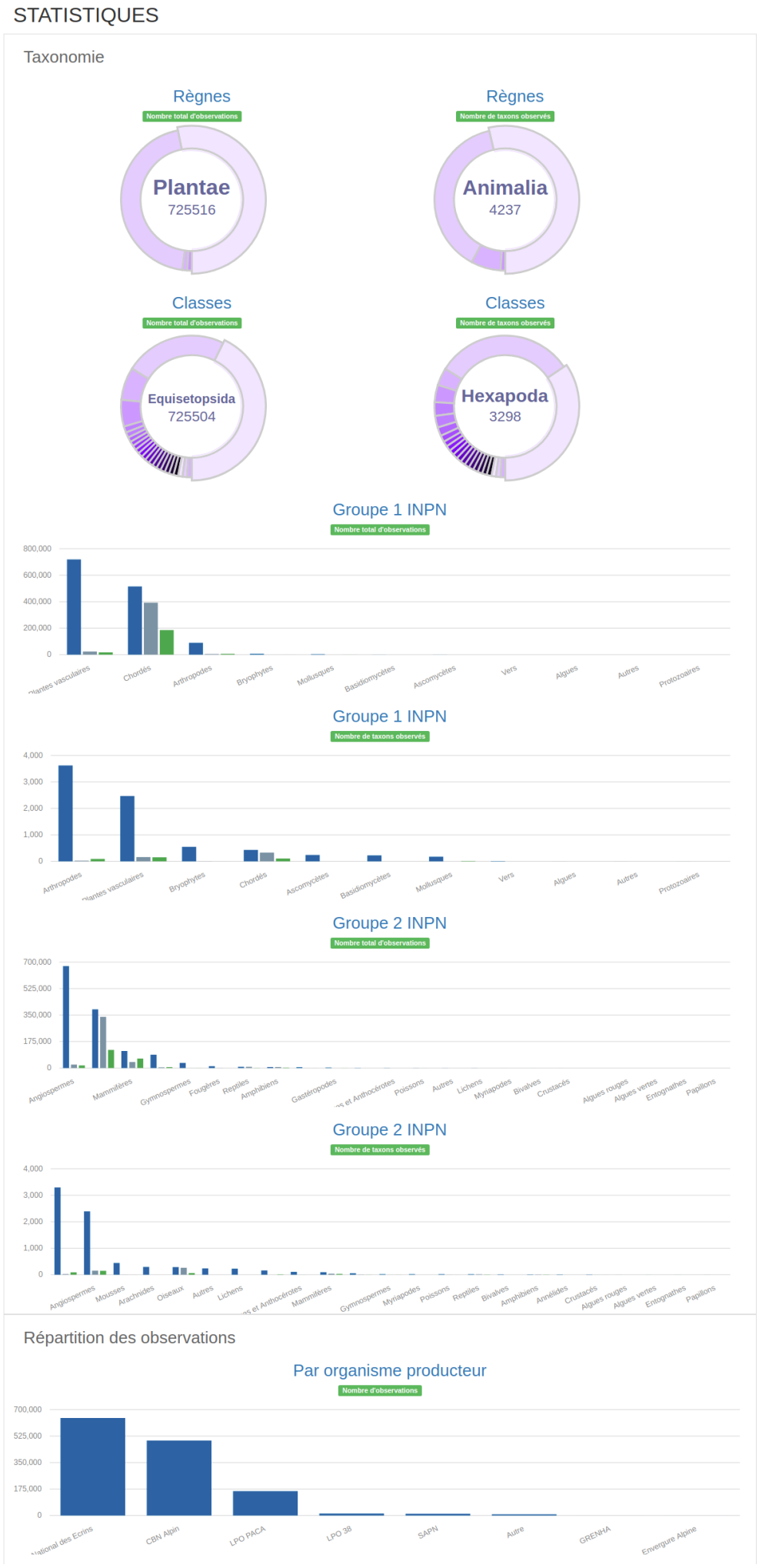
ROBERT S., DUPONT P., DE MAZIERES J., PONCET L., TOUROULT J., 2017. Procédure nationale de validation scientifique des données élémentaires d'échange du SINP pour les occurrences de taxons. Version 1. In : *MNHN Rapports* [en ligne]. 6 août 2019. Disponible à l'adresse : http://spn.mnhn.fr/spn_rapports/archivage_rapports/2017/SPN%202017%20-%202%20-%20procedure_validation_scientifique_INPN_V1.pdf.

Wild Me, 2015. Wildbook : software to combat extinction. In : *Wildbook Home Page* [en ligne]. 2 septembre 2019. Disponible à l'adresse : http://wildbook.org/doku.php.

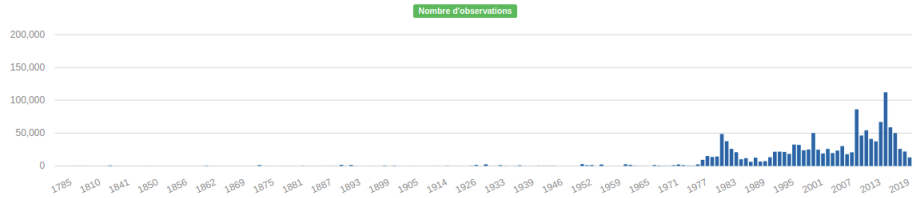
Liste des annexes

Annexe 1 : Représentations graphiques implémentées sur GeoNature 1 i
Annexe 2 : Représentations graphiques de l'INPN pour le jeu de données du Parc national des Écrins du 18/12/2017 (présentes à l'adresse https://inpn.mnhn.fr/espece/jeudonnees/8937) iii
Annexe 3 : Questionnaire complet de recueil des besoins concernant le module « Tableau de bord » destiné aux utilisateurs directs de GeoNature iv
Annexe 4 : Bilan des besoins identifiés lors de l'analyse des besoins relative au module « Tableau de bord », avec le détail du calcul des notes v
Annexe 5 : Questionnaire d'évaluation de la facilité d'utilisation du module « Tableau de bord » .. vi
Annexe 6 : Camembert de la répartition des observations par rang taxonomique (*Graphe 2*) du module « Tableau de bord » vi
Annexe 7 : Courbes du nombre d'observations par année pour chaque cadre d'acquisition (*Graphe 3*) du module « Tableau de bord » vii
Annexe 8 : Camembert de la répartition des taxons recontactés, non recontactés et nouveaux par année (*Graphe 4*) du module « Tableau de bord » vii
Annexe 9 : Principe de fonctionnement de l'algorithme de Douglas-Peucker pour la simplification de polylignes et de polygones (source : Wikipédia) viii
Annexe 10 : Carte de présence avec année seuil pour l'espèce Parnassius apollo (papillon) ix

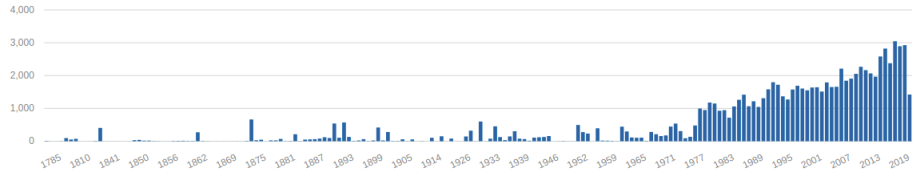
Annexe 1 : Représentations graphiques implémentées sur GeoNature 1



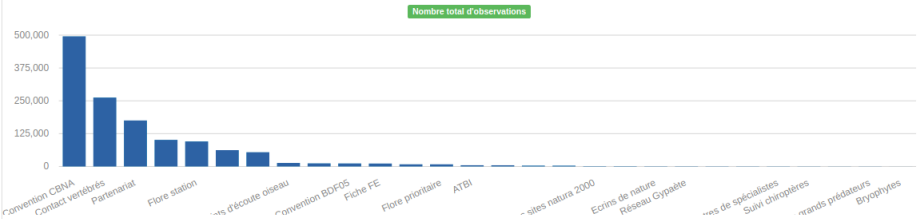
Par année



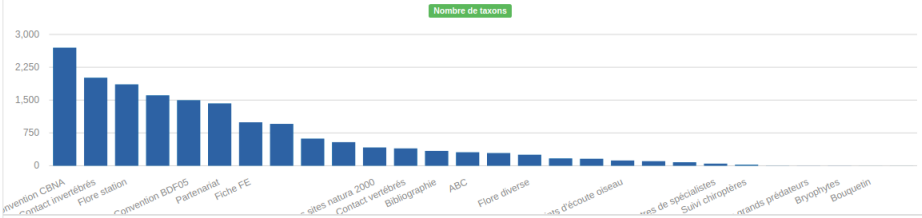
Par année



Par programme

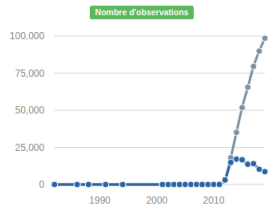


Par programme

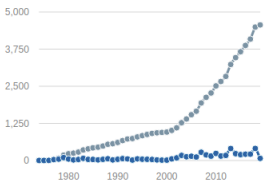


Protocoles GeoNature

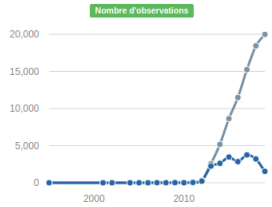
Contact faune vertébrée



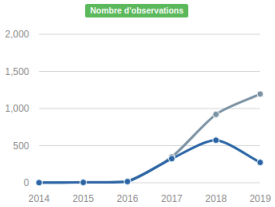
Mortalité



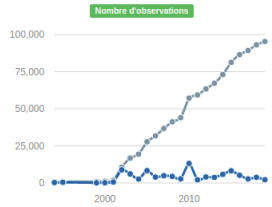
Contact faune invertébrée



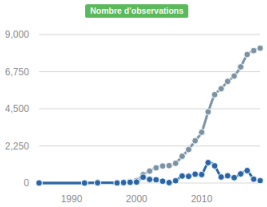
Contact flore



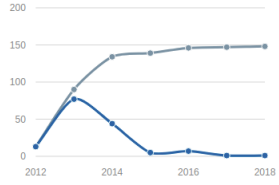
Flore station



Flore prioritaire



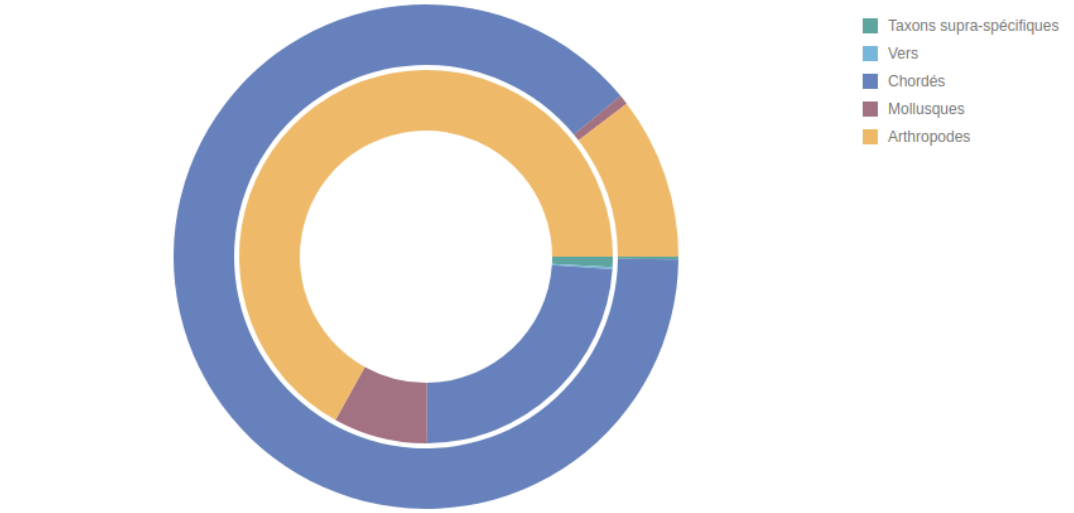
Bryophythes



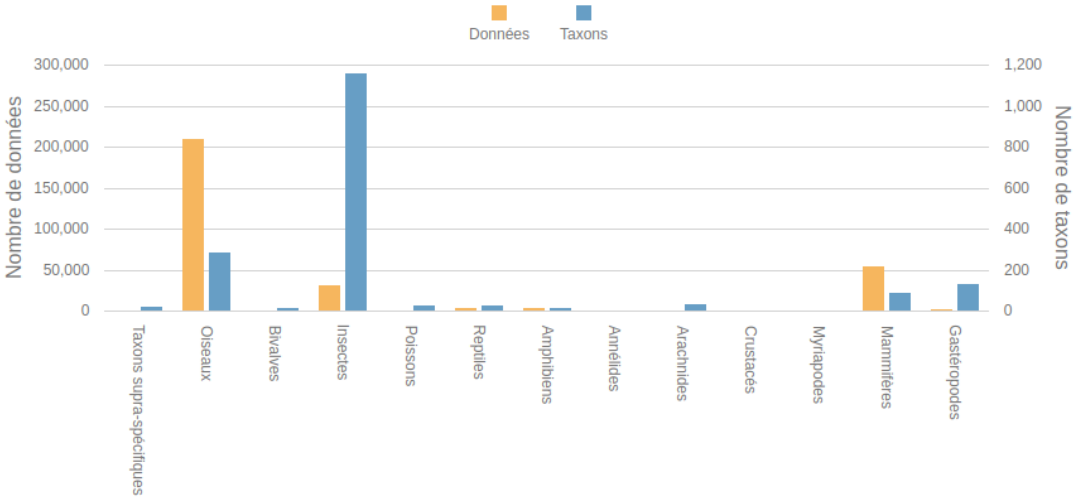
<u>Annexe 2</u> : Représentations graphiques de l'INPN pour le jeu de données du Parc national des Écrins du 18/12/2017 (présentes à l'adresse https://inpn.mnhn.fr/espece/jeudonnees/8937)

Répartition des données par groupes taxonomiques :

Groupes vernaculaires de premier ordre

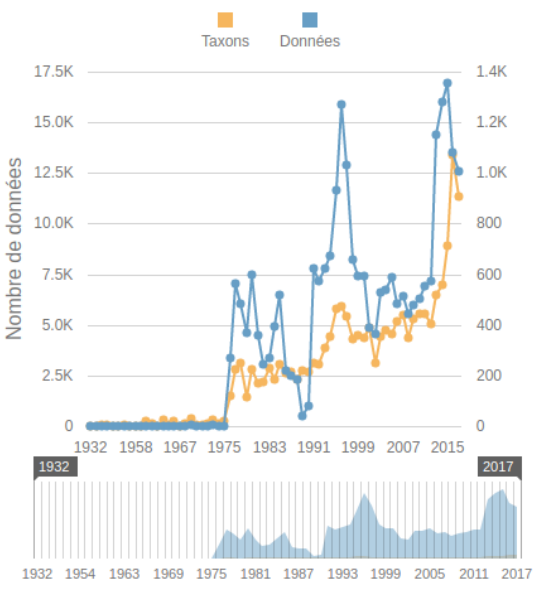


Groupes vernaculaires de second ordre

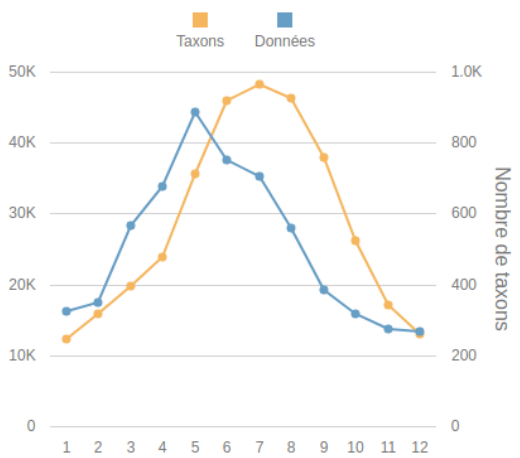


Répartition temporelle:

Distribution par année



Distribution par mois



Annexe 3 : Questionnaire complet de recueil des besoins concernant le module « Tableau de bord » destiné aux utilisateurs directs de GeoNature

1- En quoi consiste votre poste ? Quelles sont vos principales missions ?
2- Utilisez-vous GeoNature ? Si oui, quelle utilisation en faîtes-vous ? Quelles actions principales réalisez-vous sur l'application ?
3- Quels outils de traitement de données utilisez-vous en dehors de GeoNature ? Quels types de traitement de données réalisez-vous : quels types de graphes ? de cartes ? etc.
4- A quel point seriez-vous intéressé par un module "Tableau de bord" dans GeoNature, fournissant des graphes/cartes relativement basiques mais très flexibles (avec de nombreux filtres pour ajuster les représentations) sur les données contenues dans la « Synthèse » de GeoNature ?
5- Quelles tendances/informations principales auriez-vous besoin de faire ressortir ? Celles-ci peuvent tout à fait être en lien avec les traitements de données que vous m'avez indiqués à la question 3. Pouvez-vous m'en citer deux ou trois dans un ordre de priorité ? A titre d'exemple, vous pourriez avoir besoin de connaître l'espèce qui a été la plus observée en 2019.
6- Découlant de la question précédente, quels graphiques/cartes aimeriez-vous trouver précisément dans ce module ? De même, pouvez-vous m'en citer deux ou trois dans un ordre de priorité ?
7- Quels types de filtres aimeriez-vous trouver pour jouer sur ces graphiques/cartes ? Par filtres, j'entends quels niveaux de précision ? Temporel : année, mois ? Spatial : commune, taille de maille ? Taxonomie : règne, famille, espèce ? etc.

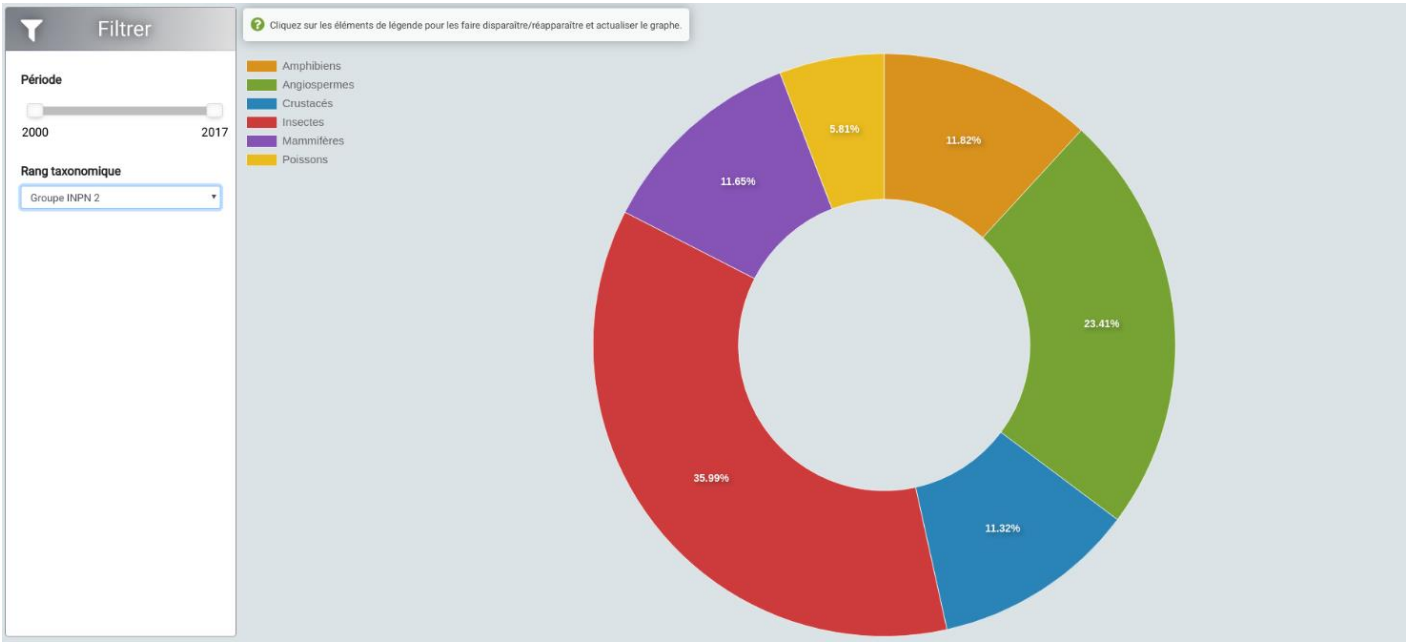
Annexe 4 : Bilan des besoins identifiés lors de l'analyse des besoins relative au module « Tableau de bord », avec le détail du calcul des notes

Besoin identifié		Calcul de la note
Objectifs généraux du module		
Visualiser la pression d'observation pour orienter les efforts de prospection		Monchicourt 1 + communauté 1 + Imberdis 4 + Bouche 4 + Corail 3 = 13
Obtenir un état des lieux des connaissances présentes dans la base de données : combien d'espèces sont présentes, lesquelles précisément et où ?		Maillard 1 + Monchicourt 1 + Bouche 2 + Corail 2 + Combrisson 4 + Imberdis 2 = 12
Obtenir des informations sur les espèces patrimoniales (protégées, menacées, rares ou ayant un intérêt scientifique ou symbolique)		Bouche 3 + Corail 4 + Combrisson 2 = 9
Pouvoir faire du reporting auprès des communautés, des collègues et du grand public avec un module ergonomique, interactif et pédagogique		Imberdis 3 + Bouche 2 + Corail 1 + grand public 3 = 9
Exporter certains graphiques sous format Excel avec les données brutes		Combrisson 3 = 3
Identifier les programmes et les cadres d'acquisition à dynamiser		Maillard 1 = 1
Propositions précises de représentations et statistiques à implémenter		
Carte de présence/absence (et classes intermédiaires) d'une espèce par zonage (pour une échelle donnée)		Bouche 3 + Imberdis 3 + Corail 4 + Combrisson 2 = 12
Carte du nombre d'observations et du nombre de taxons par zonage (pour une échelle donnée)		Maillard 1 + Combrisson 4 + Bouche 4 = 9
Graphiques réalisés dans GeoNature 1 et sur le site de l'INPN : camembert de la répartition des observations par grands groupes		Monchicourt 1 + Imberdis 4 = 5
Nombre de nouvelles espèces observées chaque année par grand groupe		Combrisson 3 = 3
Carte du statut de reproduction (nicheur certain, nicheur probable, nicheur possible) par maille		Corail 3 = 3
Moyennes d'altitude en fonction du temps pour une espèce		Bouche 2 = 2
Échelles permettant d'ajuster les représentations graphiques et cartographiques		
Échelle spatiale	Pouvoir visualiser des données seulement pour des zones précises : mailles de différentes tailles (100mx100m, 1km^2, 5km^2, 10km^2, etc.), sites définis par l'administrateur, communes.	Maillard 1 + Bouche 2 + Imberdis 2 + Corail 2 + Combrisson 2 = 9
Échelle temporelle	Pouvoir visualiser des données seulement pour une échelle de temps précise : mois, année, décennie.	Bouche 2 + Imberdis 2 + Combrisson 2 + Corail 2 = 8
Taxonomie	Pouvoir visualiser des données seulement pour un rang taxonomique précis : espèce, famille, groupe INPN, etc.	Bouche 2 + Imberdis 2 + Combrisson 2 + Corail 2 = 8

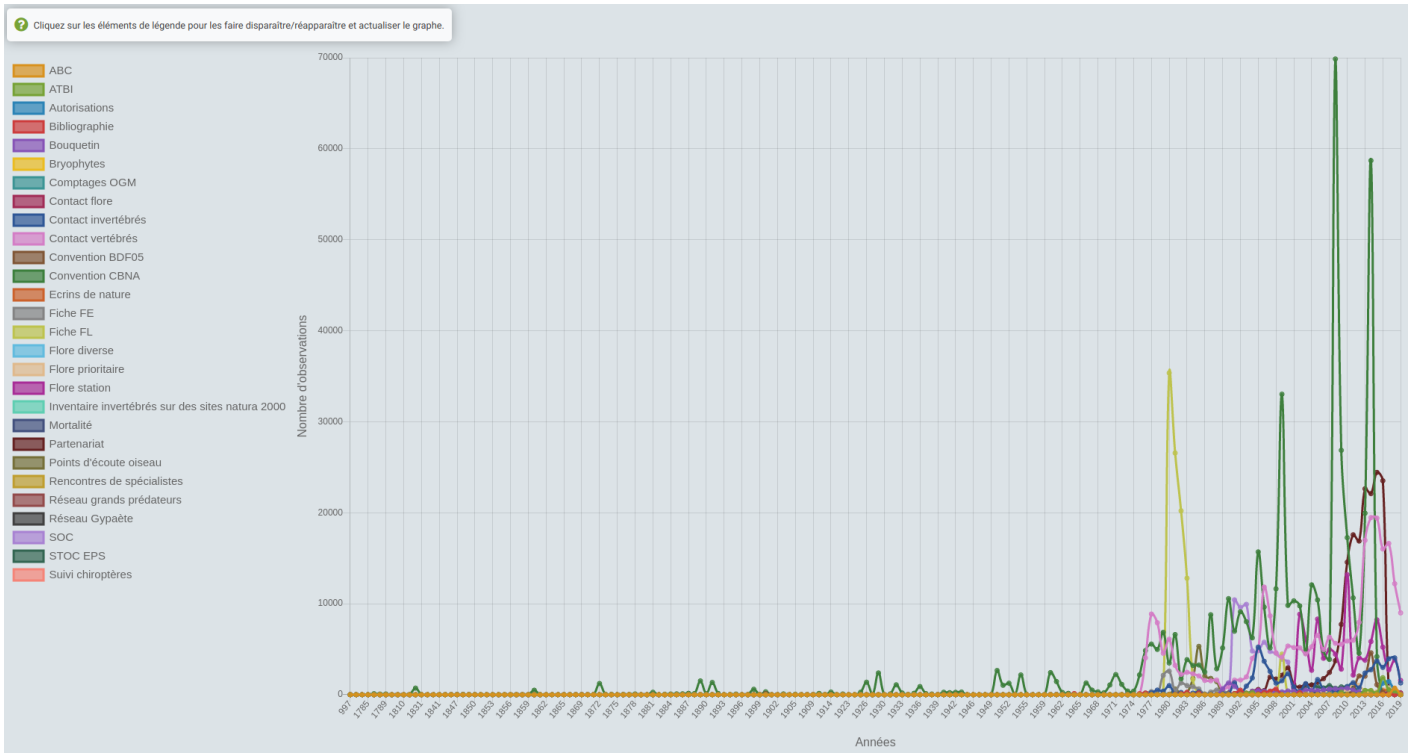
<u>Annexe 5</u> : Questionnaire d'évaluation de la facilité d'utilisation du module « Tableau de bord »

1- Dans quelle commune le Bouquetin des Alpes a-t-il été le plus observé entre 2009 et 2019 ?
2- Combien de taxons ont été observés en 2018 ?
3- Quel est le nombre d'observations du Gypaète Barbu en 2018 ?
4- Combien de nouveaux taxons ont été observés en 2018 ?
5- Quel est le pourcentage d'oiseaux observé entre 2009 et 2019 sur la totalité des groupes INPN 2 représentés sur cette période ?

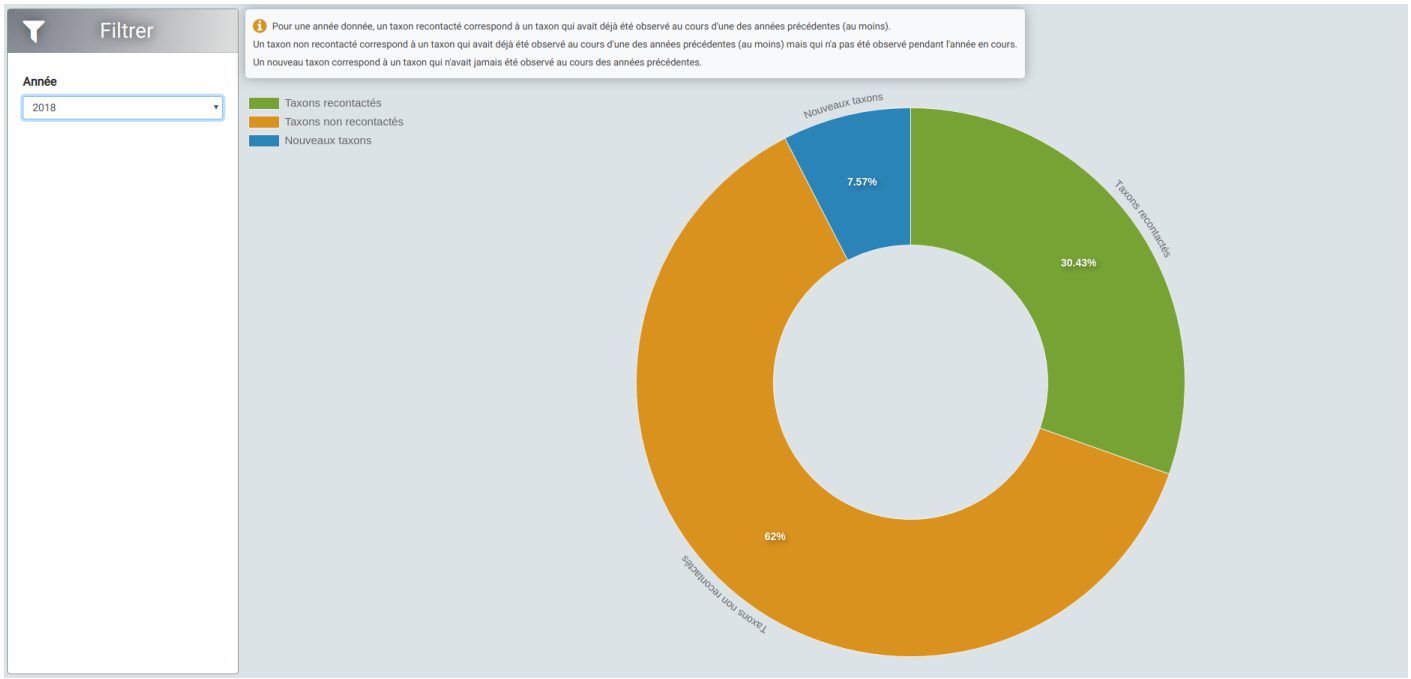
<u>Annexe 6</u> : Camembert de la répartition des observations par rang taxonomique (*Graphe 2*) du module « Tableau de bord »



<u>Annexe 7</u> : Courbes du nombre d'observations par année pour chaque cadre d'acquisition (*Graphe 3*) du module « Tableau de bord »



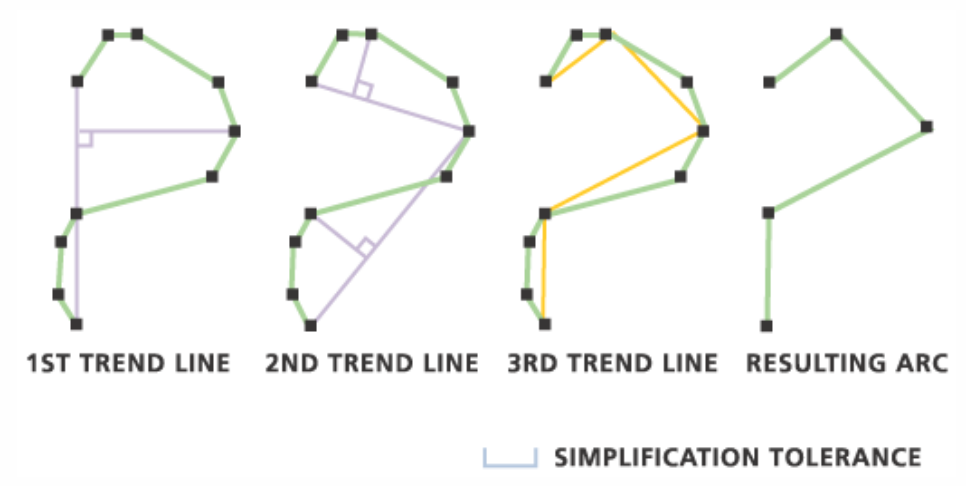
<u>Annexe 8</u> : Camembert de la répartition des taxons recontactés, non recontactés et nouveaux par année (*Graphe 4*) du module « Tableau de bord »



<u>Annexe 9</u> : Principe de fonctionnement de l'algorithme de Douglas-Peucker pour la simplification de polylignes et de polygones (source : Wikipédia)

« L'algorithme travaille de manière récursive. À l'initialisation, on sélectionne le premier et le dernier nœud (cas d'une polyligne), ou un nœud quelconque (cas d'un polygone). Ce sont les bornes.
À chaque étape on parcourt tous les nœuds entre les bornes et on sélectionne le nœud le plus éloigné du segment formé par les bornes :

1. s'il n'y a aucun nœud entre les bornes, l'algorithme se termine,
2. si cette distance est inférieure à un certain seuil on supprime tous les nœuds entre les bornes,
3. si elle est supérieure à ce seuil, la polyligne n'est pas directement simplifiable. On appelle de manière récursive l'algorithme sur deux sous-parties de la polyligne : de la première borne au nœud distant, et du nœud distant à la borne finale. »



<u>Annexe 10</u> : Carte de présence avec année seuil pour l'espèce *Parnassius apollo* (papillon)