Evolution du portail WIDDE dédié à l'exploration de la diversité génétique des populations d'animaux d'élevage

Yuwen Xu, Laurence Flori, Guilhem Sempéré

Evolution du portail WIDDE dédié à l'exploration de la diversité génétique des populations d'animaux d'élevage

Nom, Prénom : XU Yuwen
Branche : RT
Tuteur UTT : DURAND Francine
Semestre : Automne 2023

Résumé (148 mots)

Ce stage s'est déroulé dans l'entreprise INRAE, au sein de l'unité de recherche SELMET.
Il a consisté à poursuivre le développement du portail WIDDE dédié à l'exploration de la diversité génétique des animaux d'élevage, qui s'appuie sur une application JEE, tout en implémentant des règles de sécurité web, afin de permettre à des personnes non-expertes en informatique d'y importer des données génomiques et de les configurer pour les rendre accessibles aux utilisateurs finaux.
Les étapes fondamentales de ce stage étaient de :

- Développer le backend et les pages de l'interface d'import
- Créer un nouveau format d'export
- Mettre en place des procédures pour sécuriser les pages
- Optimiser une procédure d'analyse de données
- Développer des pages et des fonctionnalités backend pour configurer l'accès aux données

L'enjeu a été d'améliorer l'efficacité du traitement des données tout en garantissant la sécurité et la convivialité de l'interface du back-office.

Entreprise : INRAE
Lieu : 2 Pl. Pierre Viala, 34000 Montpellier
Tuteurs : Laurence FLORI et Guilhem SEMPÉRÉ

Mots clés :

— Recherche appliquée, développement
— Agriculture, sylviculture, pêche
— Génie logiciel; Informatique
— Logiciels - Recherche

utt UNIVERSITÉ DE TECHNOLOGIE TROYES

INRAE la science pour la vie, l'humain, la terre

cirad

Remerciements

Les acronymes, explicités p.39 sont consultables par simple clic en version pdf.

Je tiens à exprimer ma sincère gratitude à l'Institut National de Recherche pour l'Agriculture, l'Alimentation et l'Environnement (INRAE) et plus particulièrement à l'unité mixte de recherche (UMR) Systèmes d'élevage méditerranéens et tropicaux (SELMET) pour m'avoir offert l'opportunité d'effectuer ce stage enrichissant. Je remercie tout particulièrement mon encadrant M. Guilhem Sempéré et mon tuteur Mme Laurence Flori pour leur accueil chaleureux, leur soutien et leur suivi pendant toute la durée de ce stage, et pour avoir proposé un tel sujet enrichissant tant techniquement qu'humainement.

Je souhaite exprimer ma sincère gratitude envers Mme. Laurence Flori pour son précieux soutien tout au long du stage. Sa patience dans l'explication des concepts biologiques et son intérêt passionné pour la recherche sur les bovins et les ovins ont considérablement enrichi mon bagage de connaissances. Sa présentation claire des fonctionnalités de WIDDE a non seulement élargi mes connaissances en génomique, mais elle a également joué un rôle essentiel pour que je comprenne pleinement la raison pour laquelle je m'investissais dans ce travail. Ces insights ont été d'une grande aide dans la réalisation des tâches et ont profondément impacté ma perspective sur mon travail actuel et futur.

Je tiens également à exprimer ma reconnaissance envers M. Guilhem Sempéré pour son accompagnement quotidien. Il a joué un rôle essentiel en me guidant, en me soutenant et en me prodiguant des conseils judicieux tout au long du stage. Sa disponibilité, son aide précieuse lors des obstacles rencontrés et ses encouragements après la réalisation des missions ont grandement contribué à mon expérience. Au-delà de m'enseigner des connaissances techniques approfondies, Guilhem m'a également facilité une intégration rapide au sein des équipes de recherche impliquées. Mes remerciements sincères à Guilhem pour sa générosité, son professionnalisme et son impact positif sur mon parcours.

Je souhaite exprimer ma reconnaissance envers tous les agents de l'UMR SELMET ainsi qu'à Alice Boizet de l'UMR Amélioration génétique et adaptation des plantes méditerranéennes et tropicales (AGAP) qui travaille sur l'adaptation des plantes, et qui m'a apporté une assistance technique précieuse lors de mes premiers jours.

Enfin, merci de même à ma responsable de stage Francine Durand et mon suiveur Florent Retraint au sein de l'Université de Technologie de Troyes. Leur réactivité et leur expertise ont grandement facilité mon adaptation et ma réussite au cours de ce stage.

Sommaire

1 Introduction **2**

2 Présentation de l'entreprise **3**
2.1 INRAE . . . 3
2.2 CIRAD . . . 4
2.3 UMR SELMET . . . 4
2.4 UMR Intertryp . . . 5

3 Contexte et présentation de WIDDE **6**
3.1 Les marqueurs moléculaires et leur utilité en génétique . . . 6
3.2 Nouveaux outils de génotypage et de séquençage . . . 7
3.3 Présentation de l'application WIDDE . . . 7
3.3.1 Objectif . . . 7
3.3.2 Structure et architecture globale du projet . . . 8
3.3.3 Environnement de fonctionnement de la version actuelle . . . 13
3.3.4 Présentation succincte des technologies . . . 13

4 Stage **15**
4.1 Le sujet . . . 15
4.1.1 Sujet défini avant mon arrivée . . . 15
4.1.2 Sujet réel . . . 15
4.2 Place (rôle, fonction) occupée dans le service . . . 16
4.3 Positionnement et mise en œuvre des tâches . . . 17
4.3.1 Interface d'import . . . 17
4.3.2 Création d'un nouveau format d'export . . . 21
4.3.3 Implémentation de la sécurité . . . 23
4.3.4 Optimisation de code existant . . . 25
4.3.5 Interface de gestion des "datasets" . . . 27
4.4 Planning des Activités . . . 30

5 Conclusion **31**
5.1 Synthèse des résultats sur le travail . . . 31
5.2 Perspectives . . . 31

A Projets **32**

B Captures d'écran et exemples de fichiers exportés **33**

C Liste des acronymes **39**

Chapitre 1

Introduction

Ce document constitue la synthèse du travail que j'ai effectué durant la période du 4 septembre 2023 au 15 janvier 2024, au sein de SELMET, une UMR entre INRAE, Le centre de Coopération Internationale en Recherche Agronomique pour le Développement (CIRAD) et l'Institut Agro de Montpellier.

Il est à noter que la rédaction de ce rapport a été effectuée avant la fin de mon stage, qui est prévue le 1er mars 2024. L'objectif principal de ce stage était d'améliorer et d'élargir l'application du portail de données WIDDE pour l'exploration de la diversité génétique des populations d'animaux d'élevage.

Pendant ce stage, j'ai eu l'opportunité de m'immerger pendant cinq mois dans un environnement atypique de bioinformatique, aux côtés d'ingénieurs et de chercheurs aux horizons variés. Il convient de souligner que mon travail s'inscrit dans le cadre d' une collaboration entre l'INRAE et le CIRAD. Ainsi, j'ai eu pu travailler simultanément avec des collègues de l'INRAE et d'échanger avec des collègues du CIRAD. Cette expérience a été enrichissante à la fois professionnellement, humainement et scientifiquement.
Au cours de ce stage, j'ai renforcé mon expérience pratique du développement Java et appliqué mes connaissances professionnelles en développement et en sécurité. En tant qu'étudiante étrangère, ce stage m'a permis d'acquérir une expérience professionnelle en France et d'élargir mes connaissances dans des domaines aussi divers que la biologie et l'informatique.

La préparation de ce rapport est basée principalement sur mon expérience quotidienne de travail et sur les enseignements que j'ai pu tirer de l'observation du travail des autres.

Dans ce rapport, je vais présenter un aperçu du contenu de mon stage, ainsi qu'une introduction de l'entreprise et du projet sur lequel j'ai travaillé.

Chapitre 2

Présentation de l'entreprise

J'ai travaillé à INRAE (Fig. 1) au sein de l'UMR CIRAD-INRAE-L'Institut Agro SELMET en étant accueillie sur un des sites d'implantation de l'UMR au CIRAD à Montpellier. J'ai été encadrée par Guilhem Sempéré de l'UMR INTERTRYP, ingénieur au CIRAD et Laurence Flori de l'UMR SELMET, chercheuse à INRAE. Je presenterai donc successivement l'INRAE et le CIRAD, puis les UMR SELMET et INTERTRYP.

2.1 INRAE

FIGURE 1 – Logo d'INRAE

L'INRAE, Institut national de recherche pour l'agriculture, l'alimentation et l'environnement, est le premier organisme de recherche au nveau mondial spécialisé sur ses trois domaines scientifiques. Il est issu de la fusion entre l'INRA, Institut national de la recherche agronomique, et l'Irstea, Institut national de recherche en sciences et technologies pour l'environnement et l'agriculture. Grâce à la complémentarité des disciplines et des compétences à la fois technologiques et scientifiques présentes au sein des deux structures scientifiques, l'Institut national de recherche pour l'agriculture, l'alimentation et l'environnement a pour objectif de devenir l'un des leaders mondiaux de la recherche pour répondre aux enjeux sociétaux :

- de sécurité alimentaire et nutritionnelle
- de transition des agricultures (agroécologie, réduction de la chimie)
- d'érosion de la biodiversité
- et d'économie circulaire et de risques naturels

INRAE regroupe 8 229 agents titulaires dont 2 005 chercheurs, 3 179 ingénieurs et assistants ingénieurs et 3 045 techniciens, répartis dans 18 centres de recherche et 14 départements de recherche. INRAE est engagé dans 33 sites universitaires et collabore avec plus de 450 partenaires socio-économiques.

2.2 CIRAD

FIGURE 2 – Logo du CIRAD

Le CIRAD, Centre de coopération Internationale en Recherche Agronomique pour le Développement est un Établissement Public à caractère Industriel et Commercial (EPIC) français de recherche agronomique et de coopération internationale pour le développement durable, en particulier des régions méditerranéennes et tropicales (Fig. 2). Son objectif principal est de construire une agriculture durable, adaptée aux changements climatiques, capable de nourrir 10 milliards d'êtres humains en 2050, tout en préservant l'environnement.

Placé sous la tutelle du ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation et du ministère de l'Europe et des Affaires étrangères depuis sa fondation en 1984, le CIRAD mène des missions et des partenariats dans de nombreux pays de la zone intertropicale. Ceux-ci se localisent notamment en Afrique où près de la moitié des projets se concentrent, mais également en Amérique Latine et dans les Outre-mer français.

Le CIRAD comprend 1 650 salariés, dont 800 chercheurs, qui travaillent dans 33 unités de recherche. Il mène des activités dans plus de 100 pays et avec plus de 200 institutions et comprend 13 directions régionales à travers le monde.

2.3 UMR SELMET

FIGURE 3 – Logo de SELMET

L'unité de recherche SELMET (Fig. 3) constitue un collectif d'envergure internationale positionné sur l'élevage en zones méditerranéennes et tropicales. L'unité regroupe une centaine d'agents permanents et contractuels. Ils sont issus du CIRAD, de l'INRAE et de l'Institut Agro Montpellier. SELMET produit des recherches, des enseignements et des expertises afin d'accompagner les transitions durables des activités d'élevage. L'unité génère des connaissances pour accompagner les transitions durables des élevages familiaux en zones méditerranéennes et tropicales. Elle s'intéresse en particulier aux élevages pastoraux et aux exploitations mixtes agriculture-élevage.

Dans ces régions spécifiques, les éleveurs de ruminants sont confrontés à des défis particuliers liés à l'accès aux ressources fourragères, en plus des changements économiques, écologiques et sociaux significatifs. La mission de SELMET consiste à comprendre, évaluer et soutenir ces transformations, dans le but ultime de renforcer le rôle essentiel de l'élevage dans les modes de vie ruraux, l'approvisionnement des marchés, ainsi que dans l'aménagement et la gestion des espaces agricoles et pastoraux.

Pour analyser les dynamiques de ces élevages et accompagner leur transition, l'unité SELMET développe une conception systémique des activités d'élevage. L'objectif est de mener des travaux à différentes échelles, de l'animal au territoire. Il s'agit de favoriser la complémentarité entre des approches dites « intégratrices » replaçant l'élevage dans leur environnement, et des approches dites « bio-techniques » qui permettent d'analyser les éléments constituants des systèmes d'élevage.

2.4 UMR Intertryp

FIGURE 4 – Logo d'Intertryp

Interactions hôtes-vecteurs-parasites dans les infections par trypanosomatidae (INTERTRYP) (Fig. 4) compte plus de 60 personnes en 2022, dont 35 titulaires, avec 22 agents IRD, 13 Cirad, auxquels il faut ajouter 3 émérites, des post-doctorants, des ingénieurs et de nombreux doctorants. Environ 2/3 du personnel permanent sont des chercheurs. Intertryp compte 15 personnels HDR sur 35 titulaires dont 25 CR-DR ou équivalents.

Les compétences des agents regroupent diverses disciplines telles que la parasitologie, la génomique, l'épidémiologie, l'écologie, l'entomologie, la géographie. . .

Intertryp se concentre sur la recherche intégrée des éléments constitutifs du cycle des trypanosomatidés, notamment l'hôte, le parasite, le vecteur, la microbienne et l'environnement, ainsi que leurs interactions et leurs impacts sur la santé et la société. Les domaines de recherche comprennent eco-épidémiologie et lutte contre les MTN-Tryp et leurs vecteurs dans un contexte de transitions plurielles , bases moléculaires et cellulaires de l'infection par les Trypanosomatidés, nouvelles approches préventives, diagnostiques, et thérapeutiques.

L'UMR basée à Montpellier est Centre Collaborateur de l'OMS pour la Trypanosomiase Humaine Africaine.

Intertryp est laboratoire de référence de l'OIE sur les "trypanosomiases animales d'origine africaine".

Chapitre 3

Contexte et présentation de WIDDE

3.1 Les marqueurs moléculaires et leur utilité en génétique

Les marqueurs génétiques, définis comme des portions de l'Acide Désoxyribo-Nucléique (ADN) de l'organisme étudié, plus précisément un gène ou une séquence polymorphe d'ADN aisément détectable situé à un emplacement connu sur un chromosome, sont fréquemment utilisés pour analyser les ressources génétiques animales et végétales. Ils peuvent servir de balise sur le génome afin par exemple d'étudier la structuration de leur diversité génétique ou d'identifier des régions de leur génome associées avec certains caractères d'intérêt. Ils peuvent également être utilisés dans les programmes d'amélioration génétique.

L'ADN, est une macromolécule biologique présente dans presque toutes les cellules ainsi que chez de nombreux virus, qui contient toute l'information génétique, appelée génome, permettant le développement, le fonctionnement et la reproduction des êtres vivants. Il est composé de deux brins antiparallèles qui forment une double hélice. Ces deux brins sont composés d'unités structurales appelées nucléotides, elles même formées de trois unités : une base azotée, un sucre et un groupe phosphate. Il existe quatre sortes de nucléotides formant l'ADN : l'adénine (A), la guanine (G), la thymine (T) et la cytosine (C). Les deux brins qui composent l'ADN sont reliés entre eux par les nucléotides qui forment des paires complémentaires : l'adénine avec la thymine (A-T) et la guanine avec la cytosine (G-C).

Il existe différents marqueurs génétiques mais nous nous intéresserons uniquement aux marqueurs de type Single Nucleotide Polymorphism (SNP). Ces marqueurs, aussi appelés variants dans la suite de ce rapport, repèrent l'emplacement de polymorphismes résultant de la variation d'un seul nucléotide à une position spécifique du génome parmi les individus d'une même espèce (Fig. 5). La plupart des SNP sont bialléliques c'est à dire que les individus peuvent avoir deux formes possibles à la position du SNP sur un brin de l'ADN (par exemple un A ou un G). Un allèle (abréviation d'allélomorphe) est ainsi une version de cette position polymorphe. Le génotype d'un individu pour un SNP donné décrit les deux allèles que possède l'individu à la position donnée du SNP (par exemple dans le cas d'un SNP entraînant le remplacement d'un A par un G à une position donnée, les génotypes des individus d'une même espèce à cette position pourraient être AA, AG ou GG). Les SNP peuvent fournir des informations subtiles et riches sur la diversité génétique d'une espèce. En analysant les SNP, les chercheurs peuvent notamment mieux comprendre le patrimoine génétique et l'histoire évolutive d'une espèce ou d'une population (ensemble des individus appartenant à la même espèce qui occupent un espace géographique déterminé à un moment donné).

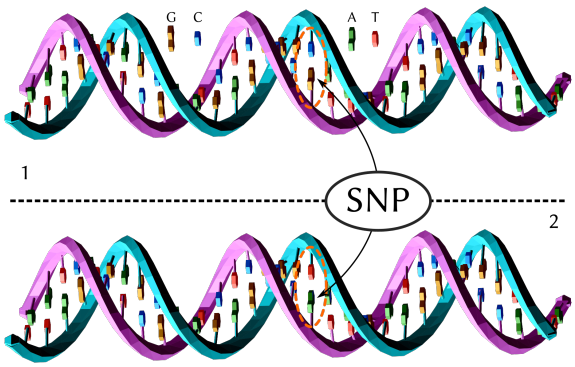

FIGURE 5 – Représentation d'un marqueur SNP entre deux séquences homologues

3.2 Nouveaux outils de génotypage et de séquençage

L'étude de ces marqueurs sur l'ensemble du génome a été facilitée par le développement d'outils de génotypage et de séquençage à haut-débit. Les technologies de Séquençage de Nouvelle Génération (NGS) et de Génotypage de Nouvelle Génération (NGG) ont apporté une révolution dans le génotypage des polymorphismes, permettant désormais une caractérisation génomique à grande échelle et économique pour un nombre croissant d'espèces. En effet, grâce aux progrès technologiques accomplis ces 30 dernières années, le séquençage d'un génome entier est devenu possible. Le projet « génome humain », débuté en 1990, avait ainsi pour but de séquencer l'intégralité du génome d'un homme avec une fiabilité supérieure à 99,99%, ce qui fut accompli en 2004. Il a permis de faire progresser les technologies de séquençage en les rendant plus fiables, plus rapides et moins coûteuses. Vers 2005 sont apparues les techniques de séquençage NGS basées sur le séquençage simultané de millions de fragments d'ADN. Grâce à ces techniques, le coût de séquençage d'un million de paires de bases est aujourd'hui de l'ordre de 0.05$, et celui-ci peut être réalisé en quelques jours. La démocratisation du séquençage a engendré une explosion de la quantité de données génotypiques disponibles. Le traitement ne pouvant se faire à la main, de nouvelles techniques s'appuyant sur l'informatique ont été développées, contribuant grandement à l'essor de la bio-informatique.

Les outils de génotypage ont également considérablement évolué ces dernières années chez les espèces d'élevage. Ainsi, le séquençage complet des génomes d'animaux d'élevage s'est accompagné du développement d'outils de génotypage à faible, moyenne puis haute densité sous la forme de puces de génotypage.

3.3 Présentation de l'application WIDDE

Le stockage efficace des ensembles des données génomiques massives produites chez les animaux d'élevage, leur gestion, leur partage et leur exploration courante, reste un défi majeur.

C'est pour répondre à cette problématique qu'a été développé WIDDE (3), une base de données NoSQL accessible à l'url `http://widde.toulouse.inra.fr` (l'interface utilisateur est illustrée par l'annexe B Fig. 16 et Fig. 17), spécifiquement conçue pour le stockage et la gestion de jeux de données de génotypage denses (jusqu'à des centaines de milliers de marqueurs génotypés sur des milliers d'individus).

3.3.1 Objectif

WIDDE est équipé de divers outils conviviaux pour :

- la sélection de données,
- l'exploration de données,
- l'export vers divers formats populaires,
- l'assignation de jeux d'individus externes à des populations (en général assimilables à des races) de référence.

Grâce à une interface web qui gère l'accès aux données publiques (accessibles librement) et privées (accessibles via un identifiant et un mot de passe), les utilisateurs peuvent sélectionner des sous-ensembles de données (basés sur la population et/ou la localisation des marqueurs), effectuer des vérifications élémentaires de la qualité des données et réaliser des analyses génétiques de population standard, telles que les tests d'équilibre de Hardy-Weinberg et l'Analyse en Composantes Principales (ACP). De plus, ils peuvent exporter les ensembles de données résultants vers divers formats utilisés en génomique des populations.

Les utilisateurs ont également la possibilité d'analyser conjointement leurs propres données de génotypage avec des sous-ensembles de données de WIDDE, afin d'explorer la proximité génétique entre leurs animaux et les populations dont les données sont présentes dans la base par le calcul de la distance de partage d'allèles (ASD), une ACP, et une classification supervisée permettant l'estimation de la composition ancestrale des échantillons et l'assignation des individus étudiés à une ou plusieurs populations.

Dans le cadre d'une publication scientifique parue en 2015 présentant la première version de WIDDE, ses fonctionnalités de sont illustrées à travers un ensemble de données bovines vaste et évolutif, représentatif de la diversité génétique mondiale des bovins. À ce jour, des données ovines sont également prises en compte dans le système et des données caprines sont en cours d'intégration. Pour chaque espèce, la base de données contient un nombre spécifique d'individus, regroupés par population, et de SNPs pour lesquels des données de génotypages sont stockées. Cela permet une analyse génétique détaillée et approfondie des espèces concernées.

En outre, il est prévu de continuer à enrichir la base WIDDE en intégrant des données provenant d'autres espèces d'animaux d'élevage, notamment les porcs. Cet enrichissement sera mis en œuvre à travers la création de nouveaux modules (équivalents à des bases de données) dédiés à ces nouvelles espèces. L'ajout de ces données contribuera à élargir le champ d'application de WIDDE, en offrant aux chercheurs et aux éleveurs des outils plus exhaustifs pour l'étude de la génétique et de l'amélioration des races d'animaux d'élevage.

3.3.2 Structure et architecture globale du projet

La liste des projets est détaillée p.32.

WIDDE est une application Web conçue pour stocker et analyser des données de génotypage. Son architecture se compose de quatre composants principaux : la base de données, le backend, le frontend et une instance Sun Grid Engine dédiée aux calculs lourds.

a. Base de données

WIDDE utilise MongoDB comme base de données NoSQL, une solution extensible et flexible idéal pour stocker des données génétiques à grande échelle. La base de données WIDDE est constituée d'un ensemble de collections dont chacune stocke des entités de différents types, couvrant tous les concepts mis en jeux, des données de génotypage elles-mêmes aux détails relatifs aux projets ayant mené à leur production, en passant par des liens informatifs sur les différentes races.

Dans MongoDB, l'utilisation de noms de champs courts vise à réduire l'occupation de l'espace de stockage. MongoDB est une base de données orientée document qui utilise le format BSON (Binary JSON) pour stocker les données. Dans cette structure, chaque nom de champ est répété dans chaque enregistrement. Si les noms de champs sont longs, les documents occuperont davantage d'espace. Par conséquent, l'utilisation de noms de champs courts est une stratégie efficace pour réduire la taille globale des données. Cette approche contribue à optimiser l'efficacité du stockage, en particulier dans les scénarios impliquant de grandes quantités de données. Le chargement des données en mémoire (occasionné par les recherches de variants ou les exports de génotypes) se trouvent également accéléré.

Les collections dans MongoDB sont similaires à des tables dans les bases de données relationnelles, et stockent un ensemble de documents. Dans notre modèle de données, elles sont structurées de la manière suivante :

- **variants :**
  Description : Cette collection stocke des informations sur les variants génétiques.
  Champs :
  - `"_id"` : identifiant unique de chaque variant
  - `"sy"` : nom de synonyms
  - `"am"` : nom de méthode d'analyse
  - `"p"` : position de variants
  - `"ty"` : type de variants
  - `"ai"` : information additional

- **individuals :**
  Description : Stocke des détails sur les individus.
  Champs :
  - `"_id"` : identifiant unique de chaque individu
  - `"po"` : population d'individu
  - `"pb"` : problème de donnée d'individu
  - `"ai"` : information additional

- **samples :**
  Description : Contient des données liées aux échantillons. La collection Samples stocke des informations sur les échantillons, où chaque échantillon est identifié de manière unique par _id de sample. Un même individu peut être représenté par plusieurs échantillons dans la collection Samples, reflétant différentes occasions où des tests SNP ont été effectués sur cet individu.
  Champs :
  - `"_id"` : identifiant unique de chaque échantillon
  - `"nm"` : nom de échantillon
  - `"in"` : identifiant d'individu pour cet échantillon
  - `"pj"` : nom du projet auquel appartient cet échantillon

- **projects :**
  Description : Regroupe des entités représentant un projet de recherche qui a abouti à l'obtention de données de génotypage. Les données d'un projet sont constituées des génotypes acquis dans le cadre d'un ou plusieurs "runs", chacun matérialisé par un fichier de génotypes

importé dans le système. Les entités de type projet contiennent également des champs servant de cache afin de pouvoir faire fonctionner l'interface de manière optimisée (e.q., liste des chromosomes impliqués pour chaque assemblage).

Champs :

- `_id` : Identifiant unique pour chaque projet
- `nm` : nom de projet
- `te` : technologie utilisée dans le projet, i.e., type de puce de génotypage.
- `pl` : niveau de ploïdie
- `rn` : "runs" de projet
- `ch` : liste des chromosomes impliqués
- `c` : contigs (Les contigs désignent des séquences d'ADN continues assemblées à partir de fragments de séquences d'ADN issus de données de séquençage. Suite à un assemblage finalisé, les contigs représentent chacun un chromosome)
- `ac` : nombre d'allèles
- `vt` : type de variante
- `ea` : annotations d'effet

- **populations :**

  Description : Stocke des données relatives aux populations d'animaux.

  Champs :

  - `"_id"` : identifiant unique de chaque population
  - `"nm"` : nom de population
  - `"ds"` : le lien réseau correspondant à ce population
  - `"pg"` : le groupe auquel appartient ce population

- **populationGroups :**

  Description : Contient des informations sur les groupes de populations.

  Champs :

  - `"_id"` : identifiant unique de chaque groupe de population
  - `"nm"` : nom de groupe de population

- **assemblies :**

  Description : Stocke la référence des assemblages génomiques considérés. L'assemblage d'un génome consiste à reconstruire la séquence génomique complète originale à partir des lectures obtenues (fragments d'ADN) après séquençage de cette même séquence, grâce des des techniques bioinformatiques d'alignement et/ou de fusion de ces fragments.

  Champs :

  - `"_id"` : identifiant unique de chaque assemblage
  - `"nm"` : nom d'assemblage

- **counters :**

  Description : Collection utilitaire sur laquelle l'application s'appuie pour la génération d'identifiants de BDD de type "auto-increment" (par exemple pour les entités de type assembly, projec et sample.)

  Champs :

  - `"_id"` : type d'entité concernée

  - "seq" : dernier ID séquentiel généré pour ce type d'entité

- **datasets :**
  Description : Stocke des informations sur les ensembles de données. Son objectif est d'organiser et de classer efficacement les données, permettant des requêtes et des présentations flexibles selon les besoins des utilisateurs et des ensembles de données spécifiques, afin de personnaliser l'affichage des données pour différents utilisateurs.
  Champs :
  - "_id" : identifiant unique de chaque ensemble de données
  - "in" : l'ensemble de tous les identifiants d'individus
  - "rf" : la publication de ce cet ensemble de données
  - "pj" : identifiant du projet auquel appartient cet échantillon
  - "vg" : groupes de variants appartenant à cet ensemble de données
  - "pb" : élément permettant de stocker la valeur "true" ou "false" indiquant si ces données sont publiques ou non

b. Backend

Le backend de WIDDE gère le traitement des demandes utilisateur et renvoie des réponses. Il est écrit en Java, utilisant le framework Spring. Le backend fait également le lien avec l'invocation sur HPC de deux fonctionnalités : l'ACP et l'assignation à des populations de référence d'individus extérieurs à la base à travers une classification supervisée.

WIDDE, outre l'application principale 'widdeProject', intègre plusieurs sous-projets indépendants mais interconnectés, contribuant de manière collaborative à une architecture distribuée complexe. Les principaux projets impliqués comprennent :

- widdeProject : Description : WIDDE est une application d'exploration de données dans le domaine de l'écologie. Fonction : Fournit des fonctionnalités riches d'analyse et d'exploration de données. Technologies utilisées : Java, HTML et JavaScript pour le développement front-end et back-end.
- Role-Manager : Description : Projet chargé de gérer les rôles et les autorisations des utilisateurs. Fonction : Assure la sécurité des données et gère les droits d'accès des utilisateurs au système. Technologies utilisées : Java, HTML et Javascript, intégré à un cadre d'authentification et d'autorisation.
- Mgdb (Mongo Genotype DataBase) : Description : Projet clé pour le stockage et la gestion des données. Fonction : Stocke de manière structurée et optimisée les données de génotypage par variant et par échantillon, ainsi que les concepts annexes. Technologies utilisées : Java, Utilisation de MongoDB comme principale base de données NoSQL.
- MgdbExport : Description : Projet axé sur l'export de données. Fonction : Fournit des fonctionnalités d'export de données de génotypage, pouvant impliquer des conversions de format et des stratégies d'optimisation. Technologies utilisées : Java pour la logique d'export, interagissant avec MongoDB.
- Individual-Assignment : Description : Projet dédié à l'affectation d'individus. Fonction : Gère et affecte des données individuelles. Technologies utilisées : Java.

Relation globale : Ces projets interagissent pour former un système complexe de traitement de données écologiques. WIDDE, en tant qu'application principale, intègre ces sous-projets en utilisant des interfaces et des protocoles clairement définis pour communiquer et réaliser des fonctionnalités globales.

c. Front-end

Le frontend de WIDDE assure l'interaction avec l'utilisateur. Il est composé de fichiers JSP, HTML, CSS et JavaScript, s'appuyant sur les bibliothèques jQuery et D3.js. Les différentes pages de l'interface utilisateur sont conçues à travers les fichiers JSP et HTML, qui utilisent CSS pour formatter les contenus. JavaScript est fortement employé pour dynamiser l'affichage, que ce soit en réponse directe à des actions de l'utilisateur, ou à travers des aller/retour invoquant le backend via des requêtes asynchrones (AJAX).

Le frontend propose également une interface permettant à l'utilisateur de visualiser les résultats de l'ACP de façon dynamique grâce à la librairie D3.js. Enfin, le "back-office" (interface d'administration) de WIDDE est géré à travers la librairie Role-Manager citée ci-dessus, qui, développée sous la forme de web-fragment, s'intègre de faon modulaire à des applications JEE de plus grande envergure.

d. Trousse à outils Opal

Opal Toolkit est un outil utilisé pour exécuter des tâches sur des grappes de calcul. WIDDE utilise Opal Toolkit pour lancer et suivre les calculs de l'ACP et d'assignation d'individus extérieurs à la base à des populations de référence via de la classification supervisée.

e. Architecture du système

Le schéma ci dessous (Fig. 6), tiré de l'article présentant la première version de WIDDE, est toujours d'actualité, et illustre les différents composants du système ainsi que le type d'interactions qu'ils partagent.

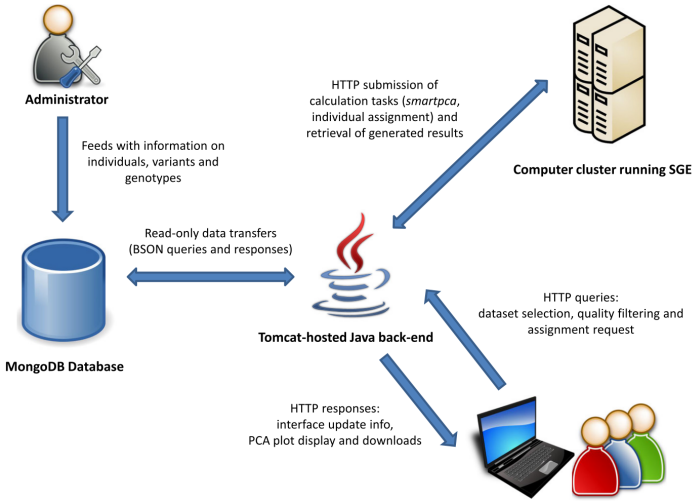

FIGURE 6 – Architecture de WIDDE

L'architecture choisie présente deux avantages principaux. Le premier est lié à son extensibilité. En effet, la flexibilité de MongoDB garantit que la base de données WIDDE peut répondre à l'évolution des besoins au fil du temps. Deuxièmement, WIDDE emploie du calcul distribué : Opal Toolkit a pour objet de déléguer l'exécution des tâches potentiellement lourdes sur des grappes de calcul, évitant que les procédures de ACP et d'assignation ne s'exécutent sur le serveur web au risque de le surcharger.

3.3.3 Environnement de fonctionnement de la version actuelle

Pour fonctionner, le système nécessite l'installation des logiciels suivants :

- Environnement d'exécution : Java JDK 17
- Environnement tomcat : Tomcat 9.0
- Environnement matériel : Linux Ubuntu
- Base de données : MongoDB v4.2.24
- Environnement HPC : Une grappe de calcul Sun Grid Engine ou SLURM, dont le nœud maître est doté d'un serveur Tomcat hébergeant Opal Toolkit 2.5

A titre indicatif, en tant que contributrice au projet j'ai utilisé au quotidien Eclipse comme plateforme de développement et Studio3T comme gestionnaire de bases MongoDB.

3.3.4 Présentation succincte des technologies

Afin de pouvoir contribuer à l'évolution de WIDDE, j'ai dû m'approprier un certain nombre de technologies de développement informatique. Les principales sont les suivantes :

a. Maven

Maven est un outil de gestion et d'automatisation de production de logiciels Java (Fig. 7). Il permet d'automatiser les tâches lors du développement d'un logiciel, comme par exemple tester des fonctionnalités, nettoyer l'espace de travail, effectuer les opérations lors de la compilation ou gérer le déploiement du logiciel. Mon maître de stage l'utilise pour l'instant pour sa fonctionnalité de dépôts lui permettant de gérer facilement ses dépendances. Il souhaite que j'élargisse son utilisation aux autres possibilités qu'offre Maven.

FIGURE 7 – Logo de Maven

b. Spring Framework

Le framework Spring (Fig. 8) est un outil essentiel pour le développement d'applications Java en entreprise, simplifiant le processus de création, de test et de maintenance des applications. Dans notre projet, nous avons pleinement exploité le framework Spring à travers trois de ses bibliothèques clés : Spring-Data, Spring-Security et Spring-WebMVC.

Spring-Data simplifie l'accès aux données via un mapping document-objet, offre une puissante fonction de requêtage et prend en charge diverses sources de données, facilitant la gestion et la manipulation des données.

Spring-Security se concentre sur l'authentification et l'autorisation, garantissant la sécurité de l'application en ligne et protégeant les informations sensibles.

Spring-WebMVC est utilisé pour construire des applications Web en facilitant l'implémentation du modèle MVC, fournissant des contrôleurs, des résolveurs de vues et une gestion des requêtes

pour créer une interface Web conviviale.

Ces trois bibliothèques interagissent dans notre projet, offrant des outils puissants qui simplifient le processus de développement, améliorent les performances et la sécurité, et garantissent que nous répondons aux besoins du projet.

FIGURE 8 – Logo de Spring

c. MongoDB

À ce jour, la Base de données (BDD) distribuée MongoDB est très populaire (Fig. 9). Elle est classée 5e BDD toutes catégories confondues et la plus répandue des BDD de type NoSQL. MongoDB étant de type NoSQL, elle est particulièrement performante dans la gestion de gros volumes. Elle est donc très bien adaptée pour la gestion d'une base de données génomiques, et son aspect distribué permet de gérer l'évolutivité des volumes de données via le "horizontal scaling" : on peut distribuer son jeu de données à travers de multiples nœuds grâce aux Replica-Sets et au Sharding. D'autre part, MongoDB n'impose pas l'utilisation de structure fixe ni de typage (NoSQL), ce qui est bien adapté aux données génomiques parfois hétérogènes. Elle stocke en effet ses données sous la forme de documents au format JSON basé sur le modèle de "clé" : "valeur". Enfin, le framework d'aggrégation offert par MongoDB est un puissant outil de requêtage qui permet de remodeler les données à volonté et de leur appliquer des filtres d'une grande précision.

FIGURE 9 – Logo de MongoDB

Chapitre 4

Stage

4.1 Le sujet

4.1.1 Sujet défini avant mon arrivée

Développé autour d'un modèle de données NoSQL structuré mais flexible, le portail WIDDE est aujourd'hui reconnu et utilisé quotidiennement par des chercheurs du monde entier. L'objectif principal du stage était de lui apporter des évolutions qui facilitent sa maintenance, et de nouvelles fonctionnalités qui le rendent encore plus utile et attractif.

Les objectifs spécifiques avant mon arrivée étaient les suivants :

- La mise au point de scripts et fonctions de vérification de cohérence de données (Java, Javascript)
- La création d'une interface WEB (Javascript, HTML, Java) autour de procédures d'import de données existantes
- La prise en charge de nouveaux formats d'export (Java)
- La recherche de pistes d'optimisation des temps de réponse
- Évolution du modèle de données pour prendre en charge des métadonnées à l'échelle des populations ou races

4.1.2 Sujet réel

A mon arrivée dans l'UMR SELMET, mes objectifs spécifiques ont été redéfinis comme suit :

- Concevoir une interface conviviale permettant à des personnes non-expertes en informatique d'ajouter des données dans le système,prenant en charge l'import d'entités de plusieurs types : les génotypes, les populations, les individus , ainsi que la configuration de "datasets" qui permettent de définir avec précision quelles données sont visibles par quels utilisateurs.
- Développer un nouveau format d'export, BayPass, afin de fournir aux chercheurs la possibilité d'obtenir directement en sortie du système des comptages d'allèles par populations.
- Mettre en œuvre la parallélisation dans l'outil d'assignation de manière à en améliorer les temps de réponse.
- Assurer la sécurité de l'API pour garantir un environnement de travail fiable et protégé.

Au moment de mon intervention, l'instance officielle de WIDDE ne contenait que des données bovines et ovines, et l'intégration de données caprines était en cours. La mise en œuvre de l'interface

d'import avait notamment pour objectif de faciliter l'intégration des nouvelles données génomiques disponibles pour d'autres espèces (porcine en particulier), offrant ainsi une plateforme plus complète pour la communauté scientifique. Ces objectifs s'inscrivent dans une approche globale visant à faciliter l'accessibilité des données, à répondre aux besoins spécifiques des chercheurs et à améliorer l'efficacité opérationnelle, tout en veillant à la sécurité des processus et des interfaces.

4.2 Place (rôle, fonction) occupée dans le service

Durant mon stage, j'ai exercé la fonction de "Développeur Full Stack", qui correspond à un poste de développeur de logiciels généraliste capable d'appréhender un large éventail de développements frontaux et dorsaux. Ces profils possèdent les compétences et les connaissances requises pour développer des applications complètes, y compris la conception de l'interface utilisateur, le développement frontal, la programmation côté serveur et l'administration de la base de données. Ce sont, plus précisément, des professionnels du développement logiciel qui maîtrisent un panel de compétences leur permettant d'assumer la responsabilité de l'ensemble du processus de développement d'une application, couvrant à la fois le front-end et le back-end. Leurs responsabilités professionnelles incluent, mais ne se limitent pas à :

- **Développement front-end :** Créer des interfaces conviviales pour garantir une excellente expérience utilisateur. Mettre en œuvre des conceptions frontales à l'aide de technologies telles que HTML, CSS et JavaScript.
- **Développement du back-end :** Construire la logique côté serveur de l'application pour gérer les données et la logique métier. Utiliser des langages de programmation côté serveur tels que Java, Python, Node.js, etc.
- **Administration de bases de données :** Concevoir et maintenir des bases de données pour assurer un stockage et une récupération efficaces des données. Maîtrise des bases de données relationnelles et non relationnelles telles que MySQL, MongoDB, etc.
- **Architecture du système :** Comprendre l'architecture globale de l'application, en veillant à la collaboration harmonieuse entre le front-end et le back-end. Prendre en compte des aspects tels que les performances, la sécurité et la facilité de maintenance.
- **Collaboration avec l'équipe :** Travailler en étroite collaboration avec d'autres développeurs, concepteurs et chefs de projet pour assurer le succès des projets.
- **Apprentissage de nouvelles technologies :** Dans un paysage technologique en constante évolution, les développeurs Full Stack doivent constamment acquérir de nouvelles compétences et connaissances pour rester compétitifs.

Les développeurs Full Stack opèrent généralement dans des start-ups, des équipes restreintes ou des environnements de développement exigeant flexibilité et polyvalence. Leur ensemble de compétences leur permet de développer des applications ou des systèmes de manière autonome. Dans le cadre d'un rapport de stage, les compétences multidisciplinaires et l'adaptabilité des développeurs Full Stack peuvent être mises en avant, tout comme leurs contributions spécifiques à un projet.

Grâce à ce stage, au cours duquel j'ai été confronté à la plupart de ces rôles, j'ai acquis une compréhension approfondie des exigences et des défis du développement Full Stack, j'ai perfectionné mes compétences et j'ai appliqué mes connaissances dans des projets concrets. Cette expérience m'a permis d'acquérir une expertise complète en développement, me préparant à aborder avec confiance et compétence les futures missions de développement Full Stack.

4.3 Positionnement et mise en œuvre des tâches

4.3.1 Interface d'import

Afin d'améliorer la facilité d'utilisation de l'import vers les bases de données pour le personnel non informaticien, nous avons développé une série d'interfaces d'import et de pages correspondantes. Cela comprend les tâches clés suivantes :

- **Conception et interaction de la page :** Concevoir une page d'import de données intuitive, garantissant que les non-techniciens puissent la comprendre et l'utiliser facilement. À travers la couche de services applicatifs, nous gérons la logique d'interaction des données de la page, assurant ainsi la convivialité du système pour l'utilisateur.
- **Traitement de téléchargement de fichiers :** Implémenter la fonctionnalité de téléversement de fichiers, permettant aux utilisateurs de sélectionner et de téléverser des fichiers de données via l'interface. La couche de services applicatifs est responsable de la réception, de la validation et du traitement des fichiers téléversés.
- **Validation et nettoyage des données :** Pendant le processus d'import de données, la couche de services applicatifs effectue des opérations de validation et de nettoyage des données, garantissant que les données importées respectent les spécifications prédéfinies, en effectuant des corrections automatiques lorsque cela est possible.
- **Traitement asynchrone :** En tenant compte du fait que l'import de grandes quantités de données peut prendre un certain temps, nous utilisons un mécanisme de traitement asynchrone. Cela permet aux utilisateurs de rester informés de l'avancement des opérations pendant que l'import de données est en cours en arrière-plan.

a. Présentation détaillée des interfaces d'import :

Import d'un fichier descriptif des variants :

Cette interface permet aux utilisateurs d'importer de nouvelles informations sur les variants (SNP) dans la base de données, comme les coordonnées du variant sur chaque assemblage (ou génome) de référence pris en charge, la liste des puces SNP incluant le variant ou les différents identifiants ayant été attribués à ce variant selon le contexte de génotypage ou de pré-analyse. De plus, ils peuvent également spécifier un lien URL pour importer des fichiers directement à partir de leur propre ordinateur ou d'une URL de fichier déjà stockée sur le serveur de WIDDE. Cela offre une flexibilité accrue aux utilisateurs pour choisir la source de leurs données de manière pratique (cette fonctionnalité est illustrée par l'annexe B Fig. 19).

Import des génotypes des individus pour chaque SNP :

Cette interface permet à l'utilisateur d'importer des données de génotypes décrivant le profil génomique d'un individu (cette fonctionnalité est illustrée par l'annexe B Fig. 20).

Création de populations :

Cette interface permet à l'utilisateur de créer de nouvelles populations assimilables à des races ou variétés, c'est-à-dire de définir une collection d'individus présentant des caractéristiques génétiques similaires. Une population regroupe un ensemble d'individus qui ont une origine commune, présentent des caractéristiques phénotypiques partagées et une certaine proximité génétique (cette fonctionnalité est illustrée par l'annexe B Fig. 22).

Import des métadonnées sur les individus :

Cette interface permet à l'utilisateur d'importer des informations détaillées sur les individus, y compris des informations de base (âge, sexe, parenté, localisation géographique...), des caractères ou phénotypes (p.ex. paramètres morphologiques ou de coloration, paramètres de production), etc. Les données relatives aux individus contiennent diverses informations qui permettent de regrouper les individus en fonction de caractères d'intérêt (cette fonctionnalité est illustrée par l'annexe B Fig. 21).

b. Fourniture d'interface de programmation d'application (API) :

Afin de permettre la communication avec le frontend, nous avons mis en place des services d'import de données via une API Spring, comprenant des interfaces de téléchargement de fichiers et de validation des données, etc. Nous mettons en place un contrôle d'accès au niveau de l'API pour nous assurer que seuls les utilisateurs autorisés peuvent effectuer des opérations d'import de données.

Liste détaillée des API :

API	Paramètres	Description
initialVariantImport	module : nom de base de données processId : ID de process snpFile : fichier de SNP	import des données SNP initiales
STDVariantImport	module : nom de base de données processId : ID de process sProject : projet pour l'import run : run pour l'import STDFile : fichier de génotypes stdFileUri : URL de fichier assemblyName : nom de l'assemblage	import des données génotypes sur la base du fichier/URL téléchargé au format STD (format interne)
PlinkVariantImport	module : nom de base de données processId : ID de process sProject : projet pour l'import run : run pour l'import mapFile : fichier identifiant les SNPs pedFile : fichier de génotype mapDataUri : URL de fichier .map pedDataUri : URL de fichier .ped assemblyName : nom de l'assemblage	import des données génotypes sur la base du fichier/URL téléchargé au format Plink (standard génétique des populations)
GSGTVariantImport	module : nom de base de données processId : ID de process sProject : projet pour l'import run : run pour l'import CORRIDFile : fichier de nommage des échantillons CsvFile : fichier de génotype CORRIDFileUri : URL de fichier de nommage CsvFileUri : URL de fichier de génotypes assemblyName : nom de l'assemblage	import des données génotypes sur la base du fichier/URL téléchargé au format GSGT (Genome Studio)

TABLE 4.1 – Détails de l'API

API	Paramètres	Description
PopulationCreator	module : nom de base de données processId : ID de process mFile : fichier tabulé regroupant les individus par population, ou fichier XLS équivalent sheetName : nom de la feuille dans le cas d'un fichier XLS	Création de populations
importMetaData	module : nom de base de données processId : ID de process uploadedFile : fichier tabulé de métadonnées	import de métadonnées pour des individus
isDatabaseEmpty	module : nom de base de données	Vérifier si la base de données est vide
getProjectNames	module : nom de base de données	Obtenir la liste des noms de projets présents dans une base de données
clearDatabaseVariant	module : nom de base de données	Effacer la base de données

TABLE 4.2 – Détails de l'API

c. Mécanisme de gestion des exceptions :

Retours utilisateurs : Pour faire face aux exceptions pouvant survenir pendant le processus d'import de données, la couche de service applicatif a été conçue avec un mécanisme de retour convivial pour les utilisateurs. Les messages d'erreur seront renvoyés de manière compréhensible, aidant les utilisateurs à mieux comprendre et résoudre les problèmes.

Journalisation : Les informations sur les exceptions seront consignées dans les journaux pour faciliter le dépannage et la surveillance du système.

d. Progression de l'import :

La progression de l'import est gérée par le backend. Ce dernier reçoit en paramètre un identifiant de processus unique à chaque exécution d'une procédure d'import. Au cours du processus d'import, il met périodiquement à jour les informations relatives à la progression et les transmet à l'indicateur de progression (cette fonctionnalité est illustrée par l'annexe B Fig. 23).

Affichage frontal : Le front-end peut afficher la progression de l'import sur la page en récupérant les informations de progression.

Gestion des erreurs : Si une erreur survient au cours du processus d'import, le backend transmet les informations d'erreur au ProgressIndicator et le frontend affiche le message d'erreur en conséquence. Celui-ci peut inclure des informations telles que la cause de l'erreur, l'emplacement, etc. afin que l'utilisateur puisse comprendre le problème.

e. Étapes de Test :

Pour garantir une interaction efficace et sans erreur avec l'utilisateur, des étapes de test ont été intégrées pour identifier et corriger les actions inappropriées des utilisateurs. Ces tests comprennent :

La vérification de la conformité des entrées textuelles : Lorsque l'utilisateur saisit des données textuelles, le système vérifie automatiquement leur conformité aux formats attendus. En cas de saisie non standard ou d'erreurs de format, une alerte est immédiatement générée pour informer l'utilisateur et demander une correction.

Le contrôle de format des fichiers téléchargés : Le système vérifie le format des fichiers téléchargés pour s'assurer qu'ils correspondent aux spécifications requises. Si un fichier est dans un format non pris en charge ou incorrect, une alerte est déclenchée pour informer l'utilisateur de l'erreur et indiquer le format correct à utiliser.

La vérification des champs obligatoires : Une autre étape cruciale du test consiste à vérifier si tous les champs obligatoires ont été correctement remplis par l'utilisateur. Si des champs obligatoires sont laissés vides ou incomplets, le système déclenche une alerte pour rappeler à l'utilisateur de fournir toutes les informations nécessaires (Fig. 10).

Ces mécanismes de test et de validation jouent un rôle crucial dans la prévention des erreurs et la facilitation d'une expérience utilisateur fluide et sans erreur. Ils contribuent également à minimiser les erreurs de données et à améliorer l'efficacité du processus d'import de données.

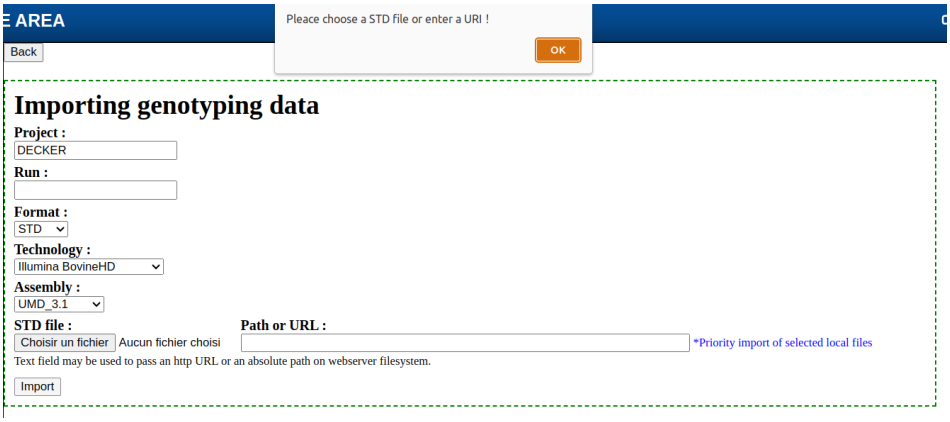

FIGURE 10 – Exemple d'alerte donnée lorsque le chemin vers le fichier STD n'a pas été renseigné lors de l'import des données de génotypage

f. Problématiques rencontrées et dolutions apportées :

Durant la phase de développement, notre équipe a été confrontée à plusieurs défis, notamment en ce qui concerne la gestion des Threads et la synchronisation des données entre le serveur (backend) et l'interface utilisateur (frontend). Nous avons identifié et adressé les problèmatiques suivantes :

Gestion des paramètres extérieurs aux threads : Des difficultés ont été rencontrées lors de l'utilisation de paramètres externes au sein des Threads, ce qui a entraîné des incohérences dans les données retournées. Pour remédier à cela, nous avons privilégié l'emploi de structures de données sécurisées pour les Threads et des méthodes de synchronisation adaptées. Cette approche a consisté à utiliser des variables locales à l'intérieur des Threads et à synchroniser leur état avec les paramètres externes après achèvement des opérations. Cette stratégie a permis de préserver l'intégrité et la cohérence des données traitées par différents Threads.

Suivi de progression des tâches longues : Dans le cas de l'import de fichiers contenant des données volumineuses (typiquement, des millions de génotypes) et dont le traitement peut durer

plusieurs secondes à plusieurs minutes, il était nécessaire de mettre en place un système de suivi de progression de l'import. Ceci a été accompli en faisant exécuter le code d'import de données dans une Thread asynchrone, de manière à permettre au contrôleur de répondre immédiatement au client. Ce dernier peut alors se concentrer sur l'interrogation d'une URL fournissant des informations sur le degré d'avancement du processus principal géré par la Thread citée ci-dessus. Cette manière de procéder permet une mise à jour dynamique et précise de l'interface utilisateur.

Ces améliorations ont considérablement renforcé la robustesse et la fiabilité de notre application, garantissant une gestion optimisée des Threads et une synchronisation efficace des données entre le backend et le frontend. La résolution de ces défis particuliers a renforcé mes compétences en programmation concurrente et en gestion de processus parallèles, tout en soulignant l'importance d'une conception méticuleuse des systèmes multithreads. Cette expérience a été cruciale pour ma compréhension des mécanismes assurant le fonctionnement harmonieux et fiable des applications web modernes.

4.3.2 Création d'un nouveau format d'export

Afin d'enrichir les formats d'export des données génomiques disponibles dans WIDDE, j'ai ajouté un nouveau format d'export, le format entrée du logiciel Baypass (2), qui est utilisé en génétique des populations. Ce logiciel permet de détecter des signatures de sélection dans le génome et des associations génome-environnement, en utilisant des méthodes statistiques bayésiennes.

a. Présentation des fichiers

Pour ce faire, j'ai converti les données de génotypage stockées dans les bases de données existantes au format de fichier requis par BayPass.

Ce format comprenait trois fichiers :

Le fichier ".genobaypass" :
C'est un fichier de comptage allélique. Il est structuré de manière à ce que chaque ligne représente un SNP distinct. Les colonnes correspondent au comptage des allèles (allele 1, allele 2) pour chaque population étudiée. Cette structure permet une analyse précise et détaillée de la fréquence allélique au sein des populations (ce type de fichier est illustré par l'annexe B Fig. 29).

Dans l'exemple suivant de fichier ".genobaypass" , les données se présentent comme suit :

— début du fichier —
81 19 86 14 2 98 8 92 32 68 23 77
89 11 81 19 9 91 1 99 27 73 27 73
89 11 91 9 0 0 15 85 77 23 80 20
[...97 lignes supplémentaires...]
— fin du fichier —

Dans cet exemple, il y a 6 populations et 100 marqueurs SNP. Pour le premier SNP, dans la première population, il y a 81 copies du premier allèle et 19 copies du second. Dans la deuxième population, il y a 86 copies du premier allèle et 14 du second, etc. Notez que les deux allèles du troisième SNP dans la troisième population ont 0 copie. Ce marqueur sera traité comme une donnée manquante dans la population correspondante.

Le fichier ".popname" :

Ce fichier contient les noms de chaque population étudiée, avec chaque nom inscrit sur une ligne distincte. L'ordre des noms de population dans ce fichier correspond exactement à l'ordre de présentation des populations dans le fichier ".genobaypass". Cette correspondance assure une cohérence et une facilité d'interprétation des données entre les deux fichiers (ce type de fichier est illustré par l'annexe B Fig. 27).

Dans l'exemple suivant de fichier ".popname" qui contient les noms de chaque population, les données se présentent comme suit :

— début du fichier —
ACE
ANB
BAD
BNA
HAN
QIC
— fin du fichier —

Dans cet exemple, il y a 6 populations. Chaque ligne de ce fichier est le nom d'une population

Le fichier ".snp_code" :
Ce fichier .snpcode est utilisé pour enregistrer des informations détaillées sur chaque SNP. Chaque ligne du fichier représente un SNP unique. Les colonnes, disposées de gauche à droite, fournissent plusieurs informations cruciales, notamment l'identifiant du SNP, le chromosome sur lequel il se trouve, sa position génomique, ainsi que les valeurs des deux allèles (parmi A, G, T, C).

Dans l'exemple suivant de fichier ".snp_code" qui contient les noms de chaque population, les données se présentent comme suit (ce type de fichier est illustré par l'annexe B Fig. 28) :

— début du fichier —

Hapmap48407-BTA-60956 26 730090 G A
Hapmap39418-BTA-61060 26 759786 C A
Hapmap36306-SCAFFOLD246909_16859 26 840268 C A
BTA-61940-no-rs 26 920919 A G
Hapmap51853-BTA-70134 0 0 A G
BTB-00918897 26 970808 A G
Hapmap45824-BTA-21209 26 1067267 A T
BTB-01735218 26 1202126 A G
ARS-BFGL-NGS-36475 26 1272311 T A
ARS-BFGL-NGS-29458 26 1292511 A G

[...90 lignes supplémentaires...]

Dans cet exemple, il y a 100 marqueurs SNP. Au premier SNP, son identifiant est Hapmap48407-BTA-60956, il se trouve sur le chromosome 26, sa position est 730090, et les valeurs des deux allèles sont "G" et "A".

b. Présentation des étapes clés

La création du nouveau format d'export s'est déroulée en plusieurs étapes clés :

- **Accès et révision des données :** Accéder à la base de données génétiques et procéder à une révision détaillée des données SNP stockées. La tâche comprend l'évaluation de la

qualité, de l'intégrité et de l'applicabilité des données pour s'assurer qu'elles conviennent aux traitements ultérieurs.

- **Prétraitement des données :** Cette phase impliquait le nettoyage et la normalisation des données pour qu'elles correspondent aux spécifications requises par BayPass. Cela incluait la gestion des valeurs manquantes ou aberrantes, l'élimination des données superflues et la préservation des informations essentielles pour l'analyse de BayPass.
- **Itération des données et écriture des fichiers :** Les données prétraitées ont ensuite été écrites dans un fichier de manière itérative. Pour ce faire, un comptage ligne par ligne a été effectué, processus rendu plus efficace par l'emploi de threads. Cette méthode a permis une organisation structurée et efficace des données en vue de leur traitement ultérieur.
- **Validation et test :** Après la conversion des données, réaliser une vérification et un test rigoureux. S'assurer que les fichiers exportés ne respectent pas seulement à première vue les exigences de format, mais peuvent aussi effectivement être correctement lus et traités par BayPass.

c. Test de la nouvelle fonctionnalité d'export

Une procédure de test rigoureuse a été mise en place pour vérifier les données exportées dans le format 'BayPass'. Cette procédure comprend :

Le contrôle de conformité du format : Des procédures automatisées ont été développées pour assurer que le format du fichier exporté correspond aux normes définies par BayPass. Ces contrôles incluent la validation de la structure du fichier, la vérification du type de données, ainsi que l'organisation des colonnes et des lignes.

L'assurance de l'intégrité des données : Nous procédons à des examens approfondis pour garantir l'intégralité et l'exactitude des données exportées. Ces examens portent sur la détection de valeurs manquantes, l'identification d'incohérences et la correction d'éventuelles anomalies.

La comparaison avec des jeux de données de référence : Pour valider l'exactitude des transformations et des calculs, les résultats d'export ont été comparés à des ensembles de données de référence préalablement établis et validés.

4.3.3 Implémentation de la sécurité

Deux tâches principales axées sur la consolidation de la sécurité de l'application à travers le framework Spring Securityont permis l'amélioration de la sécurité. La première tâche a consisté à configurer le fichier application Context-security.xml pour renforcer la sécurité des URL. Cette configuration avait pour but de s'assurer que l'accès aux URL privées, comme celles commençant par /private, nécessite une authentification utilisateur. En cas d'accès non authentifié, les utilisateurs sont automatiquement redirigés vers la page de connexion, ce qui empêche l'accès non autorisé aux sections sensibles de l'application.

La seconde tâche a consisté en l'implémentation d'une fonction "mayManageModule" nécessitant une prise en charge plus fine des rôles affectés aux utilisateurs. Cette fonction joue un rôle crucial dans le contrôle des permissions, car elle est invoquée avant chaque opération d'import de données. Elle permet de vérifier si l'utilisateur actuel dispose des autorisations nécessaires pour procéder à l'action. Cette approche garantit que les opérations d'import de données sont sécurisées.

a. Rôles des utilisateurs

Dans le cadre de ce projet, un système de gestion des rôles et des permissions a été mis en place pour assurer une sécurité optimale de l'application. Trois rôles principaux ont été définis : admin, supervisor et reader, chacun ayant des niveaux d'accès et des responsabilités distincts.

- **Rôle ADMIN :** Le rôle ADMIN est unique et représente le niveau le plus élevé d'autorités dans l'application. L'administrateur a un contrôle total sur la gestion de tous les utilisateurs et de toutes les données. Il peut accéder, modifier et gérer l'ensemble de l'application sans restrictions.
- **Rôle SUPERVISOR :** Le rôle de SUPERVISOR est similaire à celui de l'admin, mais à hauteur d'une base de données. Il peut gérer les projets et les datasets au sein de cette base de données spécifique.
- **Rôle READER :** Les utilisateurs avec le rôle READER ont des permissions limitées, principalement orientées vers l'accès en lecture. Le rôle de READER est utilisé sur les datasets pour donner accès en lecture à des données privées à certains utilisateurs authentifiés.
- **Utilisateurs anonymes :** Les utilisateurs anonymes représentent ceux qui n'ont pas de compte authentifié dans le système. Ils ont uniquement accès aux données publiques.

b. Configuration de sécurité des URL privées

Afin de s'assurer que l'accès aux URL privées nécessite une authentification, et rediriger vers une page d'erreur en cas d'authentification, nous avons utilisé des configurations spécifiques dans le fichier applicationContext-security.xml de Spring Security (Fig. 11).

```
<secur:http request-matcher="regex" pattern="^.*/403\.jsp.*$" security="none" />
<secur:http request-matcher="regex" pattern="^.*/private/index.jsp.*$" security="none" />


<secur:http realm="WIDDE WebApp Security Administration" request-matcher="regex"
access-decision-manager-ref="accessDecisionManager">
  <secur:intercept-url pattern="^.*/private/.*$" access="isFullyAuthenticated()" />
  <secur:access-denied-handler error-page="/WEB-INF/jsp/error/403.jsp" />
</secur:http>
```

FIGURE 11 – Le fichier applicationContext-security.xml

Explication du code :

- **Accès aux ressources de connexion et d'erreur :** Les patterns pour `login.json`, `403.jsp` et `private/index.jsp` sont marqués comme `security="none"`, permettant un accès non restreint à ces ressources. Cela assure que les utilisateurs peuvent atteindre la page de connexion et la page d'erreur sans authentification.
- **Sécurisation des URL privées :** La directive <secur :intercept-url pattern="*/private/.*$" access="isFullyAuthenticated()" /> est utilisée pour s'assurer que toutes les URL sous `/private` nécessitent une authentification complète de l'utilisateur.
- **Gestion des accès refusés :** En cas d'accès non autorisé, les utilisateurs sont redirigés vers une page d'erreur personnalisée (`403.jsp`) grâce à <secur :access-denied-handler error-page="/WEB-INF/jsp/error/403.jsp" />.

Résultats :

Cette configuration permet de sécuriser efficacement les URL privées de l'application, tout en offrant une expérience utilisateur claire en cas de tentatives d'accès non autorisées. Elle garantit que seuls les utilisateurs dûment authentifiés peuvent accéder aux ressources sensibles, tout en

fournissant des indications claires en cas d'échec d'authentification ou de permissions insuffisantes. Ce type de configuration universelle à travers un fichier XML présente l'avantage de permettre d'appliquer les règles de sécurité sur un ensemble d'URL sans écrire de code spécifique, et de court-circuiter totalement l'exécution du code visé en cas d'invocation non autorisée.

c. Implémentation de la fonction mayManageModule

Objectif : L'objectif principal de la fonction mayManageModule est de contrôler et de valider les droits d'accès des utilisateurs aux différents modules de l'application (Fig. 12). Cette fonction est cruciale pour assurer que seuls les utilisateurs autorisés puissent effectuer des actions sur des modules spécifiques, renforçant ainsi la sécurité globale de l'application.

Implémentation technique : Voici la description de son implémentation :

```
public boolean mayManageModule(@RequestParam("module") String sModule, HttpServletResponse response) throws Exception {
    Authentication authentication = SecurityContextHolder.getContext().getAuthentication();
    Collection<? extends GrantedAuthority> authorities = authentication.getAuthorities();
    if (authorities.contains(new SimpleGrantedAuthority(IRoleDefinition.ROLE_ADMIN)))
        return true;
    HashSet<String /*module*/> allowedModules = userDao.getSupervisedModules(authorities);
    if (allowedModules.contains(sModule))
        return true;
    response.setStatus(HttpServletResponse.SC_FORBIDDEN);
    return false;
}
```

FIGURE 12 – Implémentation de la fonction mayManageModule

Explication du code

- **Obtention des autorités de l'utilisateur :** La fonction récupère d'abord l'objet Authentication de l'utilisateur actuel pour accéder à ses autorités (roles).
- **Vérification du rôle administrateur :** Elle vérifie si l'utilisateur possède le rôle d'administrateur. Si oui, l'utilisateur a automatiquement la permission de gérer tous les modules.
- **Vérification des modules autorisés :** Pour les utilisateurs non-administrateurs, la fonction vérifie si le module demandé fait partie des modules qu'ils sont autorisés à gérer.
- **Réponse en cas d'accès non autorisé :** Si l'utilisateur n'a pas les autorisations nécessaires, la fonction renvoie un statut HTTP de SC_FORBIDDEN, indiquant un refus d'accès.

Résultats :

Cette méthode permet de maintenir la sécurité de l'application, en s'assurant que seuls les utilisateurs autorisés peuvent accéder à des fonctions spécifiques (modification des données). Elle joue un rôle crucial dans la prévention des accès non autorisés et assure une gestion fine des droits d'accès.

4.3.4 Optimisation de code existant

Cette section décrit une tâche spécifique assignée dans le cadre de mon stage impliquant l'optimisation d'un programme Java utilisé pour l'estimation de l'ascendance des individus à partir de données génétiques. Cette fonctionnalité permet d'estimer l'ascendance (ancestry) des individus dont les génotypes sont importés par l'utilisateur. Ces calculs sont executés dans un programme externe à la webapp, sur un cluster de calcul (HPC). Le programme actuel n'utilise pas le multithreading. L'objectif de cette tâche a consisté à modifier le programme pour qu'il puisse exécuter des calculs parallèles, augmentant ainsi son efficacité.

a. Optimisation par multithreading

L'optimisation par multithreading consiste à exécuter plusieurs tâches ou portions de code en parallèle pour améliorer l'efficacité et réduire le temps d'exécution global d'une application. En identifiant soigneusement les zones où le multithreading peut être appliqué efficacement, on peut significativement améliorer les performances d'une application tout en maintenant la sécurité et la fiabilité du traitement des données.

- **Identification des "itérations indépendantes" :** L'identification de boucles for ou while contenant du code dans lequel chaque itération peut être considérée comme indépendante est cruciale. En effet, le multithreading ne peut être mis en place efficacement que lorsque les itérations de la boucle peuvent s'exécuter simultanément sans dépendre les unes des autres. Cela signifie que le résultat d'une itération ne doit pas influencer ou dépendre des résultats des autres itérations.
- **Charge de travail :** Évaluer la charge de travail des boucles permet de déterminer si le gain potentiel en performances avec le multithreading justifie la complexité supplémentaire. Les boucles qui effectuent des tâches lourdes ou qui sont des goulots d'étranglement en termes de performance sont des candidates idéales pour le multithreading.
- **Gestion des données :** Une gestion sécurisée des données partagées entre les threads est essentielle pour éviter les problèmes de concurrence, tels que les conditions de course. Cela implique souvent l'utilisation de structures de données thread-safe et de techniques de synchronisation pour assurer l'intégrité des données lorsque plusieurs threads y accèdent ou les modifient simultanément.

b. Gestion des threads

La gestion efficace des threads est fondamentale pour optimiser le multithreading. Elle nécessite une approche réfléchie pour maximiser l'utilisation des ressources CPU tout en évitant les problèmes liés aux accès concurrents.

- **Identification des ressources disponibles :**
  La méthode Runtime.getRuntime().availableProcessors() est utilisée pour déterminer le nombre de cœurs de processeur disponibles.
- **Utilisation de la classe ExecutorService :** ExecutorService permet de gérer de manière plus efficace et moins laborieuse les threads. Au lieu de nécessiter de lancer et suivre manuellement chaque thread, cette classe fournit un cadre pour exécuter des tâches de manière asynchrone et gérer leur cycle de vie.
- **Gestion des tâches en parallèle :** Simplification de la gestion des threads avec ExecutorService pour automatiser la création et la gestion des tâches en parallèle.

c. Méthode de test

Les procédures suivantes ont été mises en œuvre afin de vérifier et valider les modifications apportées :

- **Validation de la conformité :** Utilisation de la commande diff pour comparer les fichiers de sortie avant et après l'optimisation, afin de garantir qu'il n'y ait aucune modification dans les données.
- **Tests de performance :** Mesure du temps d'exécution pour évaluer l'efficacité de l'optimisation.
- **Analyse des résultats :** Observation des temps de traitement pour chaque tâche exécutée en parallèle afin d'identifier les améliorations ou les éventuels goulots d'étranglement.

d. Résultats

Des tests comparatifs ont été réalisés autour d'un jeu de données type, sur un ordinateur doté de 8 cœurs.

- **Conformité des données :** Les résultats des tests ont confirmé que les fichiers de sortie (.tsv) sont identiques à ceux produits avant l'optimisation, assurant ainsi l'intégrité des données après l'implémentation du multithreading.
- **Amélioration des performances :** Une réduction significative du temps de traitement a été constatée. Le temps moyen nécessaire pour les calculs est passé de 94.8 secondes avant l'optimisation à 50.6 secondes après, indiquant une amélioration notable de l'efficacité.

Le fichier progressIndicator.txt a indiqué une durée totale de calcul notablement inférieure, prouvant l'efficacité de l'optimisation par multithreading (Fig. 13 et 14).

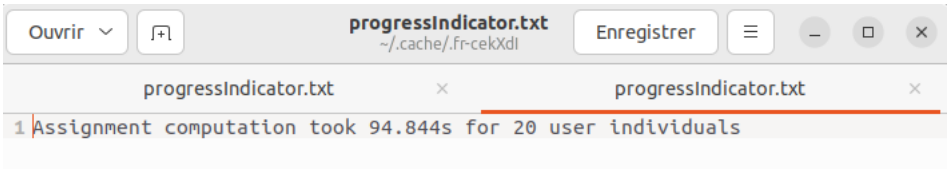

FIGURE 13 – Affichage de la durée du calcul avant l'optimisation

FIGURE 14 – Affichage de la durée du calcul après l'optimisation

4.3.5 Interface de gestion des "datasets"

Comme déjà introduit dans la section 3.3.2 "Structure et architecture globale du projet", les datasets sont des ensembles organisés de données. Ils représentent un élément clé dans la gestion et le traitement de l'information car ils matérialisent des entités couvrant tout ou partie des données liées à un projet (par la sélection d'individus et de chromosomes), qui peuvent être définies comme publiques (accessibles sans authentification) ou privées (avec possibilité de préciser quels utilisateurs y ont accès).

Dans cette partie, je vais présenter le développement d'une interface dédiée à la création et à la modification de ces datasets. Cette tâche est cruciale pour améliorer la gestion des données, offrant une solution flexible et conviviale pour manipuler des ensembles de données variés. Elle implique la conception d'une interface utilisateur intuitive, facilitant la création de nouveaux datasets et l'ajustement de ceux existants, pour mieux répondre au besoin de gérer dynamique ces données. Ce travail contribue à l'objectif global d'optimiser l'organisation des données et de faciliter leur accès et personnalisation pour les utilisateurs (la liste de datasets est illustrée par l'annexe B Fig. 24).

a. La création et modification de dataset

La création de dataset

La création d'un nouveau dataset dans l'interface dédiée implique plusieurs étapes essentielles

pour garantir une organisation et une personnalisation adéquates des données. Voici les étapes détaillées pour la création d'un nouveau dataset dans l'interface (cette fonctionnalité est illustrée par l'annexe B Fig. 25 et 26 :

- **Sélection d'une base de données existante :** Choisir une base de données déjà présente au sein de laquelle le dataset sera créé.
- **Choix d'un projet existant :** Sélectionner un projet existant dans la base de données choisie.
- **Nom du nouveau dataset :** Entrer le nom du nouveau dataset.
- **Définition de la visibilité :** Décider si le dataset sera public ou privé.
- **Sélection des groupes de variants :** Choisir les groupes de variants (i.e., chromosomes) appropriés pour chaque assemblage.
- **Choix de la population :** Sélectionner une population parmi celles disponibles dans le projet actuel.
- **Sélection des individus :** Choisir des individus spécifiques au sein de chaque population sélectionnée.
- **Références bibliographiques :** Fournir une liste de références indiquant les articles dans le cadre desquels chaque population a été génotypée.

La modification de dataset

Pour la modification d'un dataset existant, les points suivants sont à considérer :

- **Éléments fixes :** La base de données, le projet, et le nom du dataset ne peuvent pas être modifiés.
- **Éléments modifiables :** Les utilisateurs peuvent ajuster les données à inclure dans un dataset existant. Cela implique la possibilité de changer les groupes de variants pour les assemblages, de sélectionner ou de modifier les populations et les individus au sein de celles-ci, ainsi que de mettre à jour les références bibliographiques.

Cette flexibilité permet de tenir à jour les datasets en fonction de l'évolution des besoins ou des informations sans perturber leur structure fondamentale.

b. Processus opérationnel

Il est important et intéressant de distinguer deux cas d'utilisation différents de cette interface :

Action de création :

- L'utilisateur clique sur le bouton "Créer" sur la page web.
- L'interface est dirigée vers la page de gestion des datasets : /private/editDataset.do.
- Côté serveur, le contrôleur Spring génère un objet de type org.springframework.web.servlet.ModelAndView et le renvoie "vierge" à la vue JSP.

Action de modification :

- L'utilisateur clique sur le bouton "Modifier" sur la page web.
- L'interface est dirigée vers URL /private/editDataset.do, avec des paramètres supplémentaires, notamment la base de données et le projet. Ces paramètres supplémentaires permettent au backend d'identifier le dataset spécifique à éditer.
- Le contrôleur utilise alors ModelAndView pour réunir les données existantes de ce dataset depuis la base de données, et les met à disposition de la vue JSP constituant l'interface de modification, où certaines informations restent non modifiables pour maintenir la cohérence des données.

La page JSP utilisée pour créer et modifier un dataset est donc la même. En mode modification, la page est pré-remplie avec les données existantes, et certaines parties sont figées pour éviter des modifications non autorisées.

Soumission des données :

Une classe DataSetDTO a été créée pour servir de "data-transfer-object", simplifiant la transmission de l'ensemble des données constituant un dataset via l'encapsulation de ses contenus (Fig.15).

- Après avoir rempli le formulaire, l'utilisateur soumet les données en cliquant sur "create".
- Une requête AJAX est envoyée à l'URL de traitement (DatasetCreator.json), avec pour body l'équivalent d'un DataSetDTO en version JSON.
- Le backend traite ces données et les enregistre dans la base de données.

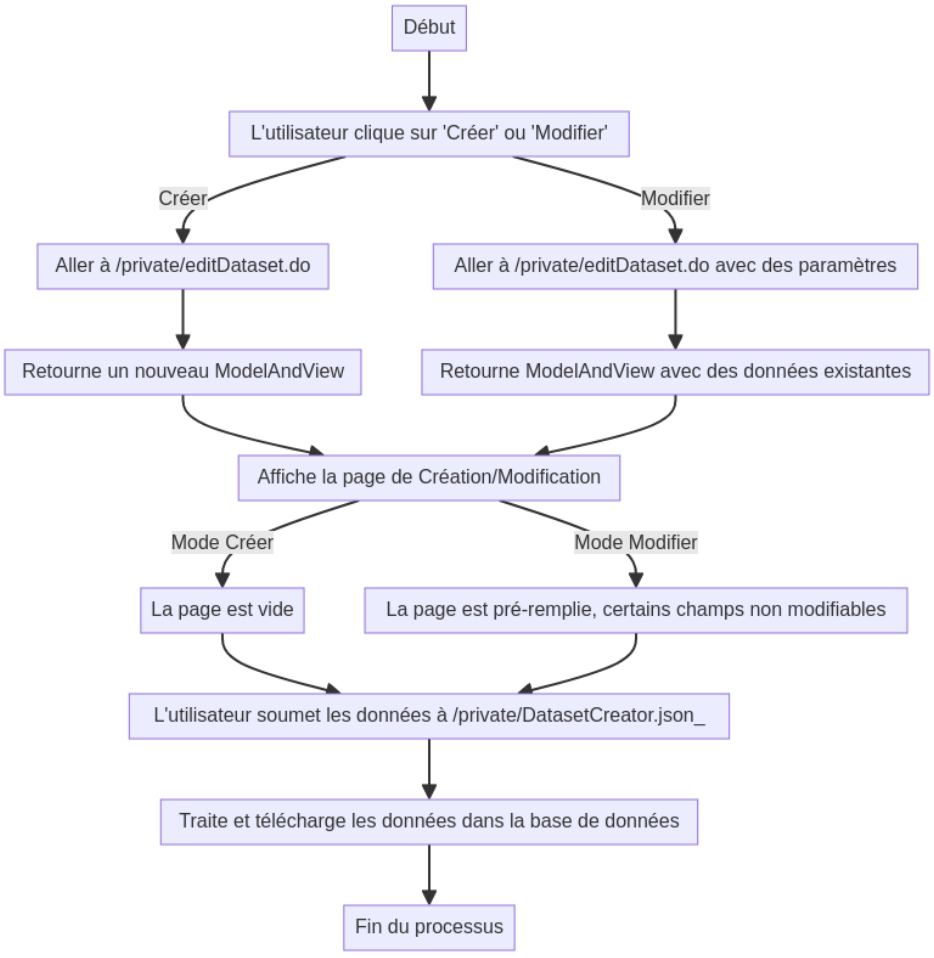

FIGURE 15 – Processus Opérationnel

c. Concepts et conventions utilisés :

Il me semble utile de donner ici en exemple quelques unes des conventions et bonnes pratiques appliquées dans l'ensemble du projet et plus particulièrement dans cette partie.

- **URLs *.do :** Traditionnellement utilisées dans les servlets Java pour gérer les requêtes de page web. Les URL se terminant par .do sont généralement associées à des actions qui renvoient une vue complète, souvent utilisées pour charger des pages web ou des interfaces utilisateur.
- **URLs *.json :** Utilisées pour les requêtes qui renvoient des données au format JSON, souvent employé pour des appels d'API ou des requêtes AJAX. Ces requêtes sont typiquement utilisées pour transférer des données entre le serveur et le client sans recharger toute la page.
- **ModelAndView :** Il s'agit d'une classe fournie par Spring MVC qui contient à la fois les données (modèle) et la présentation (vue) à renvoyer en réponse à une requête. ModelAndView est utilisé pour faire afficher la vue appropriée, chargée avec les données nécessaires pour la création ou pré-remplie pour la modification.
- **DataSetDTO :** DTO signifie Data Transfer Object. datasetDTO sert à encapsuler les données de dataset et les simplifier pour le transfert et le traitement.
- **Interaction entre front-end et back-end :** Les requêtes en .do et en .json facilitent une communication fluide entre le Front-End et le Back-End. Dans la gestion des datasets, editDataset.do est utilisé pour initialiser et afficher la page de création ou de modification de dataset, tandis que DatasetCreator.json est utilisé pour réceptionner les données du dataset après leur saisie, traitant la requête en arrière-plan et renvoyant une réponse au format JSON. ModelAndView et DataSetDTO assurent une gestion adéquate des données et une expérience utilisateur cohérente.

4.4 Planning des Activités

Le tableau ci-dessous donne un aperçu du temps consacré à chacune des tâches accomplies.

Section	**Période**
Apprentissage de la biologie et la technologie, téléchargement de logiciels et de projets	4/9 - 8/9
Création d'interface pour les procédures d'import	8/9 - 10/10
Création d'un nouveau format d'export	6/10 - 19/10
Optimisation	19/10 – 6/11
Implémentation de la sécurité	1/11 - 10/11
Partie dataset	10/11 - 22/12

TABLE 4.3 – Planning des activités

Chapitre 5

Conclusion

5.1 Synthèse des résultats sur le travail

Grâce à l'intégration d'une interface utilisateur intuitive, WIDDE est désormais entièrement gérable par une personne sans compétences informatiques spécifiques, facilitant ainsi les procédures d'import. Nous avons également renforcé la sécurité de l'accès aux programmes et aux données en améliorant la fonctionnalité qui empêche les utilisateurs non autorisés d'accéder à des pages sensibles. Pour répondre aux besoins des chercheurs, une nouvelle fonctionnalité a été mise en place permettant l'export de fichiers au format Baypass, un atout majeur pour leur travail. Enfin, l'efficacité du système a été optimisée par l'ajout de multiples fils d'exécution, assurant ainsi une performance accrue et une meilleure réactivité de la fonctionnalité d'assignation. Ces améliorations contribuent à une expérience utilisateur plus fluide et à une gestion des données et des fonctionnalités de sécurité plus robustes, tout en contribuant à fournir les ressources nécessaires pour appuyer les travaux de recherche avancés.

Cette expérience enrichissante m'a non seulement permis de maîtriser l'architecture d'une application trois-tiers mais aussi d'affiner mon utilisation des frameworks Spring pour le backend, Bootstrap pour le frontend, et MongoDB pour la gestion de données. J'ai renforcé aussi ma capacité à gérer plusieurs projets de front, amélioré mes aptitudes en communication et coordination, et aiguisé ma faculté d'apprentissage rapide. Les échanges autour du suivi de mon travail et la rédaction de ce rapport m'ont également aidée à améliorer ma capacité de synthèse et ma pratique de la langue française notamment à l'écrit. Ces compétences transversales, acquises parallèlement aux connaissances techniques, constituent un atout indéniable pour mon développement professionnel.

5.2 Perspectives

Ce rapport devant être rendu avant la fin de mon stage, il me restera plus d'un mois de travail. Dans les semaines qui viennent, le portail qui s'appuie sur notre application sera enrichi par la création d'un module porcin, qui permettra de mettre à l'épreuve en conditions réelles l'ensemble des interfaces d'import nouvellement développées. Le système doit aussi sous peu être migré vers une nouvelle machine virtuelle plus performante, et ce changement sera l'occasion de mettre en place un sécurisation via le protocole https.

En parallèle, nous avons le projet de démarrer le développement d'un prototype web qui pourrait faciliter l'interopérabilité avec la plateforme Galaxy(1) à travers l'utilisation du client d'API blend4j (`https://github.com/galaxyproject/blend4j`). Celui-ci devrait permettre d'offrir aux utilisateurs la possibilité de lancer facilement de nombreux types d'analyses en ligne à partir de données exportées de WIDDE, ouvrant ainsi le service à de nouveaux cas d'utilisation.

Annexe A

Projets

Ci-après, sont listés les liens vers les dépôts des projets sur lesquels j'ai travaillés pendant mon stage, où se trouvent le code source et l'historique de mon travail.

Veuillez noter que dans ce document, aucun lien direct vers Web-Interfaced next generation Database for genetic Diversity Exploration (WIDDE) n'est fourni. Afin de respecter les accords de confidentialité et de sécurité des informations, l'accès à WIDDE est strictement réglementé et non accessible publiquement. Par conséquent, les détails spécifiques ou les liens directs vers ce système ne peuvent être inclus dans ce rapport. Sous-modules de WIDDE, par ordre d'importance dans les travaux que j'ai mené au cours de ce stage :

- **Mgdb2 :** Implémentation générique pour le modèle de données utilisé pour exploiter des données de génotypage dans des bases MongoDB
  o `https://github.com/SouthGreenPlatform/Mgdb2`
- **role_manager :** Implémentation générique de la partie administration et gestion des utilisateurs
  o `https://github.com/SouthGreenPlatform/role_manager`
- **Mgdb2Export :** Procédures d'export à partir d'une base Mgdb2
  o `https://github.com/SouthGreenPlatform/Mgdb2Export`
- **Mgdb2BrapiModel :** Modèles de données pour BrAPI v2
  o `https://github.com/SouthGreenPlatform/Mgdb2BrapiModel`
- **GenotypeFileManipulation :** Utilitaires pour la manipulation de certains formats de fichiers
  o `https://github.com/SouthGreenPlatform/GenotypeFileManipulation`
- **OpalClient :** Implémentation pour l'invocation des services de la boîte à outils Opal
  o `https://github.com/GuilhemSempere/OpalClient`

Annexe B

Captures d'écran et exemples de fichiers exportés

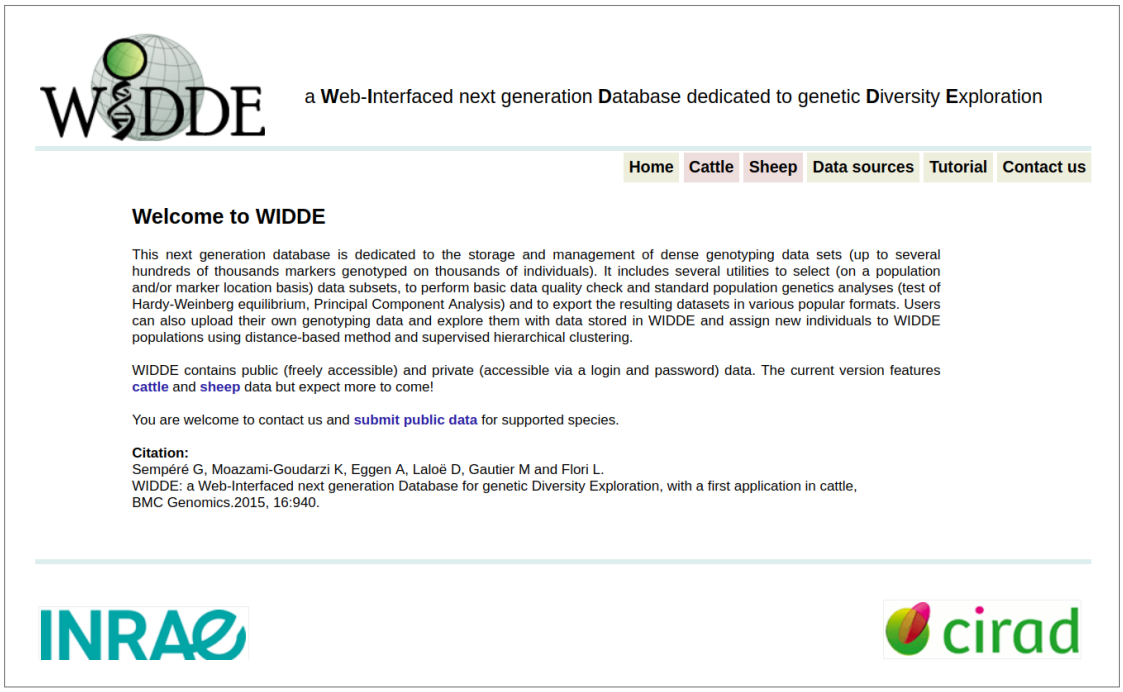

FIGURE 16 – Page d'accueil de WIDDE

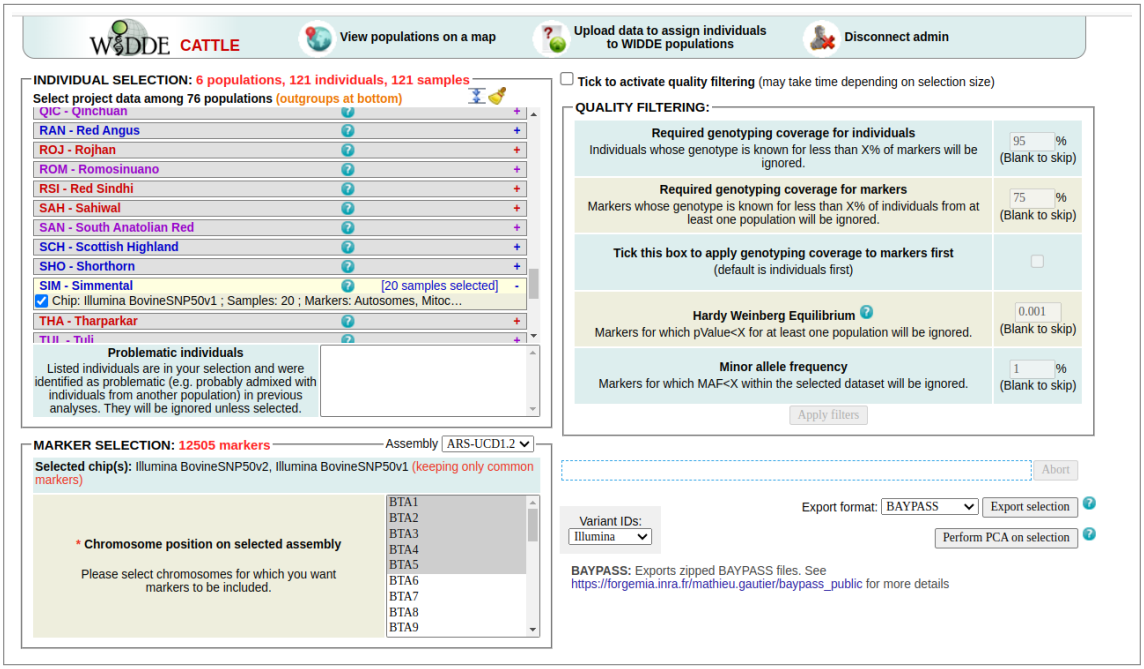

FIGURE 17 – Interface utilisateur principale de WIDDE

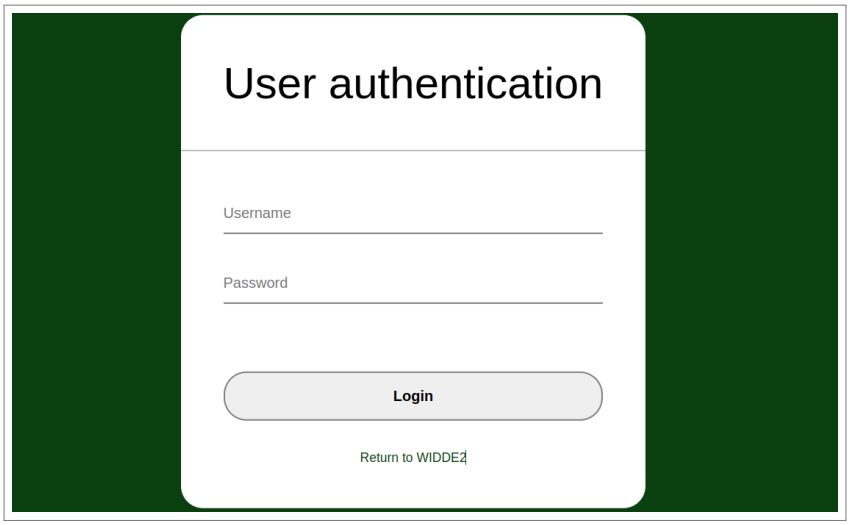

FIGURE 18 – Page d'authentification

FIGURE 19 – Interface d'import des variants

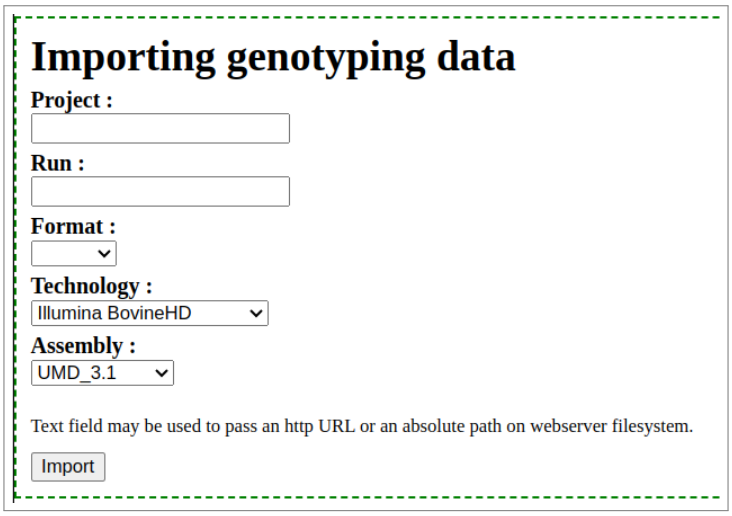

FIGURE 20 – Interface d'import des génotypes

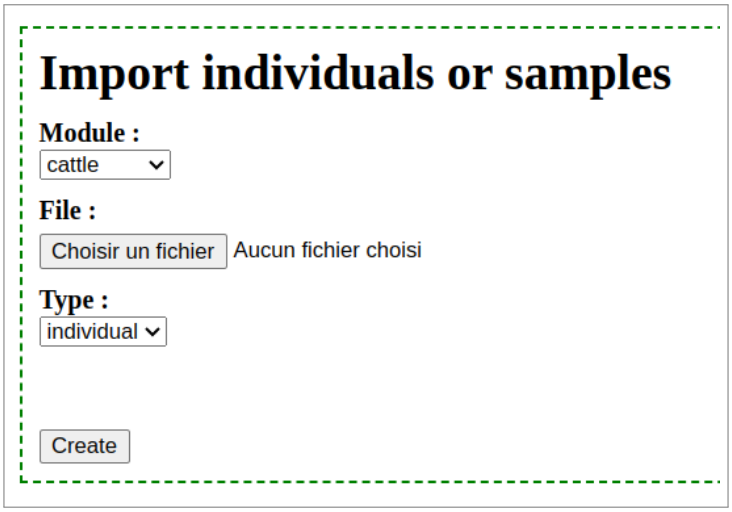

FIGURE 21 – Interface d'import des métadonnées

FIGURE 22 – Page de création de population

FIGURE 23 – Interface de suivi de progression

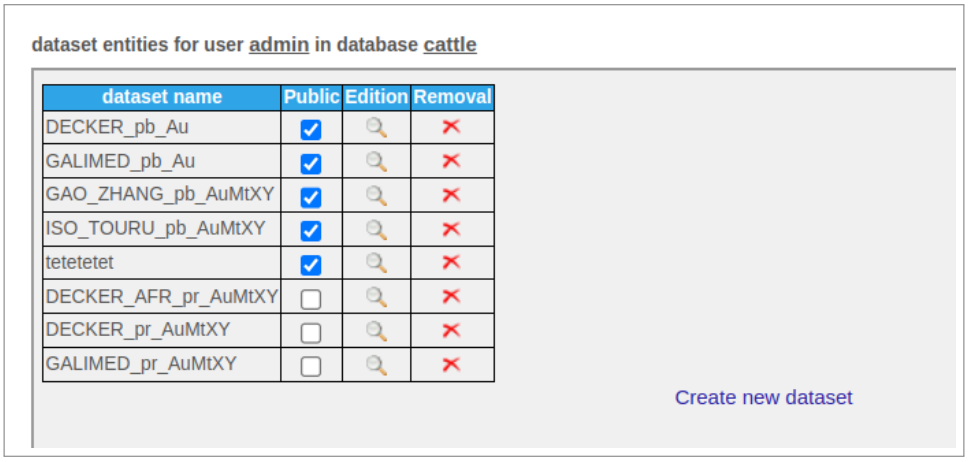

FIGURE 24 – Page de la liste de datasets

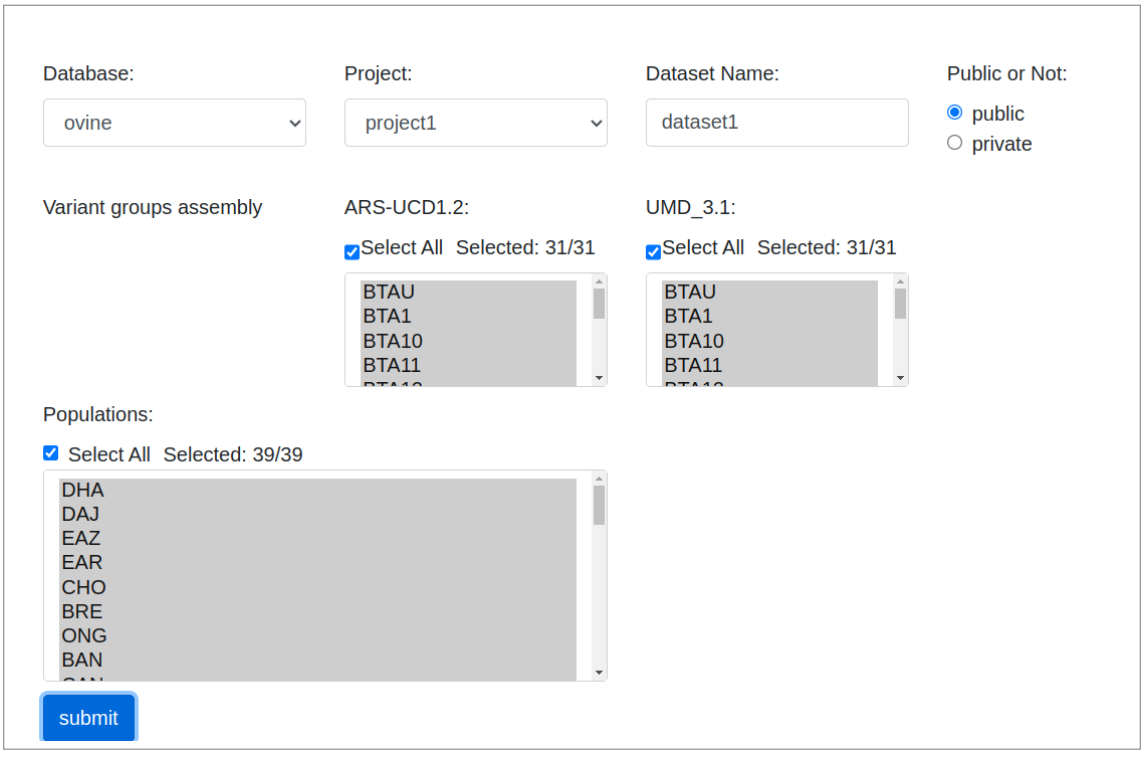

FIGURE 25 – Interface de création de dataset

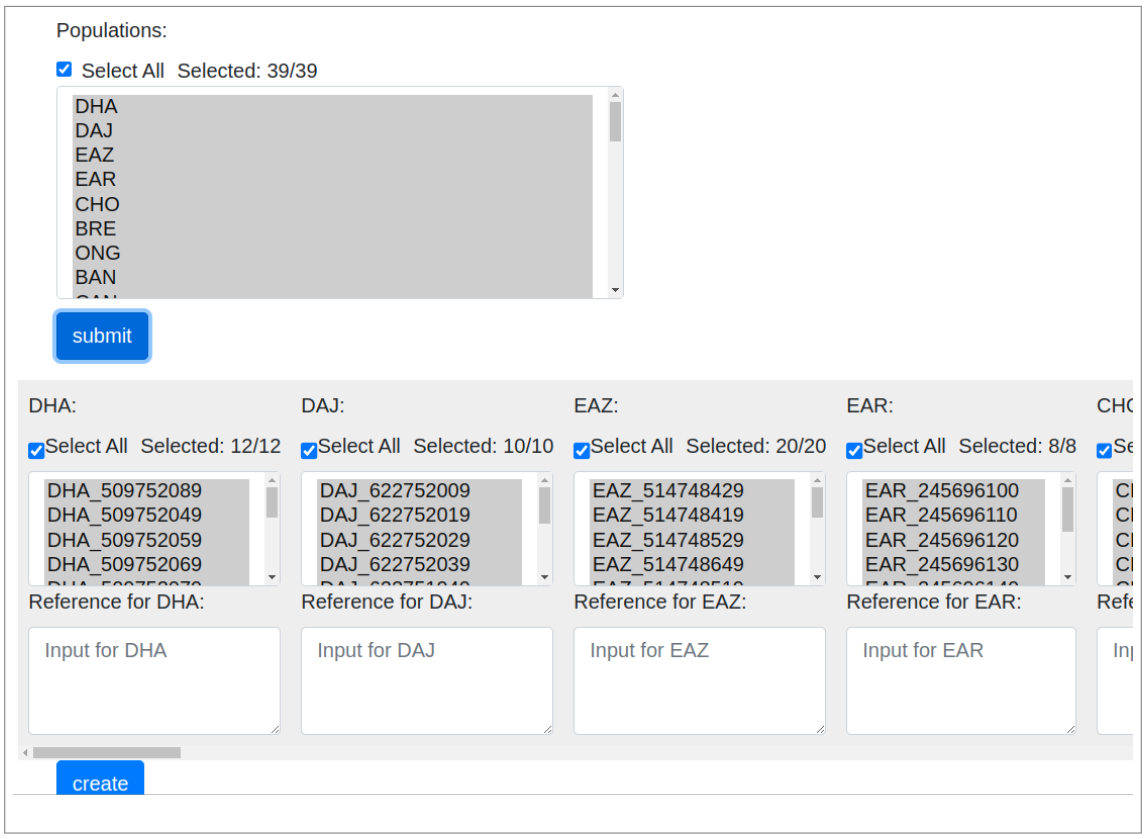

FIGURE 26 – Interface de création de dataset

FIGURE 27 – Exemple de fichier *.popname

```
VariantID Chromo Position Allele1 Allele2
Hapmap43437-BTA-101873 1 776231 G A
ARS-BFGL-NGS-16466 1 907810 G A
Hapmap34944-BES1_Contig627_1906 1 1032564 C A
ARS-BFGL-NGS-98142 1 1110393 G A
Hapmap53946-rs29015852 1 1150763 A G
ARS-BFGL-NGS-66449 1 1204825 A G
ARS-BFGL-BAC-32770 1 1566539 A G
ARS-BFGL-NGS-65067 1 1604940 G A
ARS-BFGL-BAC-31497 1 1626677 A
ARS-BFGL-BAC-32722 1 1650662 A G
ARS-BFGL-BAC-34682 1 1671886 G A
ARS-BFGL-NGS-3964 1 1695632 G A
ARS-BFGL-NGS-98203 1 1730547 C A
ARS-BFGL-BAC-36895 1 1798608 A G
ARS-BFGL-BAC-2376 1 1834781 G A
ARS-BFGL-BAC-31722 1 1909740 A G
BTA-49284-no-rs 1 1929664 C G
ARS-BFGL-BAC-6557 1 1954528 C A
ARS-BFGL-BAC-7196 1 1984725 C A
```

FIGURE 28 – Exemple de fichier *.snp_code

```
14 2 7 11 44 36 24 16 21 27 30 10
12 4 14 4 49 31 22 18 37 7 28 12
8 8 18 0 49 31 33 7 47 1 38 2
12 4 17 1 36 44 35 5 34 14 20 20
6 10 3 15 5 75 9 31 13 35 3 37
15 1 16 2 76 4 20 20 44 4 35 5
4 12 0 18 1 79 0 40 5 39 0 40
13 3 16 2 47 33 16 24 35 11 28 12
16 0 18 0 80 0 40 0 48 0 40 0
14 2 17 1 62 18 40 0 46 2 29 11
12 4 17 1 78 2 40 0 46 2 29 11
12 4 17 1 79 1 38 2 47 1 40 0
7 9 7 11 53 27 18 22 24 24 24 16
16 0 18 0 0 0 40 0 0 0 0 0
15 1 17 1 35 45 38 2 37 11 25 15
12 4 18 0 80 0 40 0 0 0 0 0
15 1 16 2 79 1 39 1 40 8 40 0
14 2 18 0 56 24 40 0 42 6 38 2
14 2 16 0 51 29 32 8 0 0 0 0
9 7 13 5 8 72 23 17 0 0 0 0
```

FIGURE 29 – Exemple de fichier *.genobaypass

Annexe C

Liste des acronymes

SELMET Systèmes d'élevage méditerranéens et tropicaux

UMR unité mixte de recherche

INRAE l'Institut National de Recherche pour l'Agriculture, l'Alimentation et l'Environnement

CIRAD Le centre de Coopération Internationale en Recherche Agronomique pour le Développement

WIDDE Web-Interfaced next generation Database for genetic Diversity Exploration

INTERTRYP Interactions hôtes-vecteurs-parasites dans les infections par trypanosomatidae

ADN Acide Désoxyribo-Nucléique

SNP Single Nucleotide Polymorphism

NGS Séquençage de Nouvelle Génération

NGG Génotypage de Nouvelle Génération

ACP Analyse en Composantes Principales

BDD Base de données

API interface de programmation d'application

AGAP Amélioration génétique et adaptation des plantes méditerranéennes et tropicales

Bibliographie

[1] The Galaxy Community : The galaxy platform for accessible, reproducible and collaborative biomedical analyses : 2022 update. *Nucleic Acids Research*, 50(W1):W345–W351, 2022.

[2] Mathieu Gautier : Genome-wide scan for adaptive divergence and association with population-specific covariates. *Genetics*, 201(4):1555–1579, 2015.

[3] Guilhem Sempéré, Katayoun Moazami-Goudarzi, André Eggen, Denis Laloë, Mathieu Gautier et Laurence Flori : Widde : a web-interfaced next generation database for genetic diversity exploration, with a first application in cattle. *BMC genomics*, 16(1):1–8, 2015.