



UNIVERSITÉ DE MONTPELLIER

FACULTÉ DES SCIENCES

Navigation dans les règles
d'implication extraites de
connaissances agroécologiques en
santé animale et végétale pour l'aide
à la décision

Master 2 Informatique
Parcours GL

Effectué au LIRMM
du 1^{er} février au 31 août 2022
par GUILHÈM BLANCHARD

Tutrice universitaire :
MADALINA CROITORU

Encadré par :
MARIANNE HUCHARD
ARNAUD SALLABERRY
PIERRE MARTIN

Année Universitaire 2021 - 2022

Remerciements

Je tiens tout d'abord à remercier sincèrement Marianne Huchard, avec qui j'avais des cours au premier semestre, qui a pu noter mon assiduité et mes rendus de TP, avec qui j'ai pu discuter et qui a pu reconnaître mes capacités à faire un travail sérieux. Effectivement, sans lien direct avec un des membres de l'équipe proposant le stage, ma candidature aurait pu avoir plus de mal à aboutir à la simple lecture du CV, étant donné la durée qu'il m'a fallu pour valider ma licence, cela n'étant pas dû à de l'absentéisme, mais à ma dyslexie/dysorthographe.

Je souhaite aussi remercier profondément Pascal Poncelet et Arnaud Sallaberry qui ont pu me guider au quotidien dans le développement de mon application, tant sur les choix techniques que graphiques. Ils m'ont sans cesse indiqué les points à améliorer au cours de nos nombreuses réunions pour atteindre le meilleur résultat possible.

Je souhaite également remercier Pierre Martin, qui a su d'une part expliquer la lecture des règles d'implications, mais aussi y apporter des modifications pour faciliter mon travail.

Je tiens enfin à remercier les équipes ADVANSE et MAREL dans lesquelles j'ai été accueilli. J'ai travaillé seul sur mon projet, mais j'ai fait des rencontres agréables au cours des réunions d'équipes et séminaires du vendredi après midi, ainsi que l'équipe ICAR avec qui j'ai partagé de nombreux repas.

Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre du programme d'Investissements d'Avenir portant la référence ANR-16-CONV-0004.

Table des matières

I	Introduction	3
1.	Objectifs du stage	3
2.	Contexte	3
3.	Organisation du rapport	3
II	Règles d'implication sur des données multi-relationnelles	4
1.	Le projet Knomana	4
2.	Les règles d'implication sur des données multi-relationnelles	5
III	État de l'art	8
1.	Graphes	8
2.	Tables	10
IV	RCAvizIR	12
1.	Principes de la visualisation	12
2.	Implémentation	16
	a. Navigation	16
	b. Parseur	17
V	Gestion de projet	19
VI	Conclusion	21

I Introduction

1. Objectifs du stage

Au cours de mon stage, j'ai été accueilli au sein du LIRMM afin de produire un prototype d'application permettant une visualisation et une navigation au sein de règles d'implication. Ces règles, qui sont fournies par le CIRAD, explicitent des relations connues dans une culture agricole entre une espèce cultivée, un agresseur et une espèce protectrice. Ces règles indiquent quelles sont leurs interactions et précisent aussi d'autres propriétés (usage médical, alimentaire, ...). L'outil que j'ai produit permet d'offrir un accès intelligible à l'ensemble des règles sous forme d'une application web afin de permettre l'optimisation des cultures.

2. Contexte

Au LIRMM, j'ai été accueilli par les équipes ADVANSE et MAREL. L'équipe ADVANSE travaille sur la fouille de données, la visualisation analytique et l'apprentissage automatique. Ces axes, et plus précisément celui portant sur la visualisation, sont au cœur de ce sujet de stage. L'équipe MAREL travaille quant à elle sur les axes de manipulation de modèles et de programmes et les méthodologies de développement pour et par la réutilisation. Elle contribue également au domaine de l'analyse de concepts formels (FCA : Formal Concept Analysis) et de l'analyse de concepts relationnels (RCA : Relational Concept Analysis), qui sont deux méthodes centrale au sujet de ce stage, les règles fournies étant basées sur des constructions issues de RCA.

Le CIRAD est l'organisme français de recherche agronomique et de coopération internationale pour le développement durable des régions tropicales et méditerranéennes. C'est lui qui développe le projet Knomana (<http://agents.cirad.fr/index.php/pierre+martin/Knomana>) qui a suscité ce travail. Le CIRAD nous fournit des données et des règles sur ces données et souhaite les visualiser sous forme graphique.

3. Organisation du rapport

Ce rapport présentera les règles d'implication dans le cadre de RCA, la problématique de la visualisation de ces règles d'implication, les premières maquettes du projet, les choix techniques et visuels qui ont été fait et l'avancement de la réalisation.

Le chapitre II présente brièvement le projet Knomana puis développe la problématique rencontrée sur la visualisation de règles dans le contexte de RCA. Le chapitre III présente un état de l'art sur la visualisation des règles. Le chapitre IV présente les principes suivis et le prototype réalisé ainsi que son implémentation. La gestion du projet est abordée au chapitre V. Le rapport se conclut au chapitre VI par un résumé des réalisations et des perspectives.

II Règles d'implication sur des données multi- relationnelles

1. Le projet Knomana

Le projet Knomana (CIRAD/IRD) a constitué un recueil d'informations sur les plantes à effet pesticide et antibiotique. Ce recueil doit servir à accompagner les experts pour proposer des solutions permettant de se passer des pesticides de synthèse.

Les données sont recueillies dans un document excel (voir figure 1). Ce document comporte actuellement plus de 46000 descriptions d'usages de plantes contre des agresseurs de culture et on comprend aisément qu'il est difficile à utiliser tel quel.

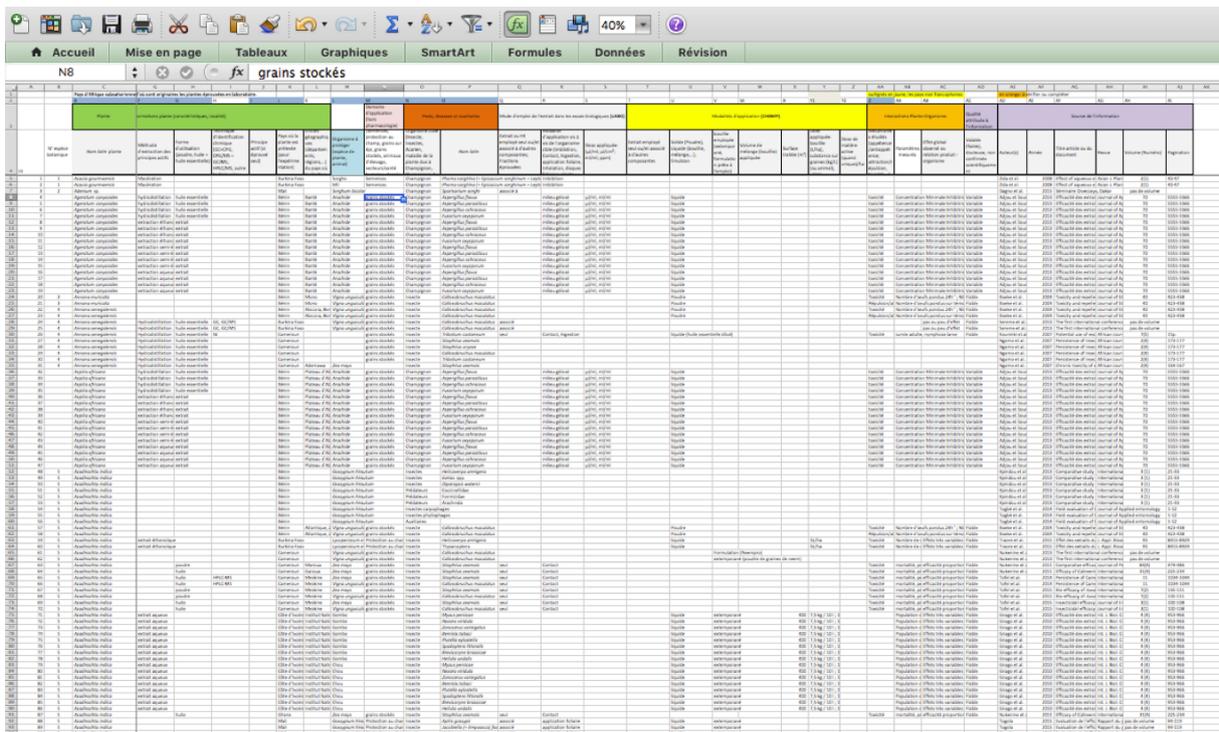
The image shows a screenshot of an Excel spreadsheet. The title bar at the top indicates the file name is 'grains stockés'. The ribbon at the top includes 'Accueil', 'Mise en page', 'Tableaux', 'Graphiques', 'SmartArt', 'Formules', 'Données', and 'Révision'. The spreadsheet itself has a grid with columns labeled A through Z and rows numbered 1 through 46000. The data is organized into several distinct sections with colored headers: a green header for 'Nom des plantes', a blue header for 'Nom des agresseurs', a yellow header for 'Nom des cultures', and a purple header for 'Source de l'information'. Each row contains detailed text entries corresponding to these categories.

FIGURE 1 – Aperçu d'une feuille excel contenant les données du projet Knomana

Pour être analysées, l'ensemble de ces données, ou plus souvent une sélection de ces données, est exporté(e) et structuré(e) selon un schéma entité-relation. Ce schéma entité-relation est présenté à la figure 2. Il comporte des systèmes de protection (*ProtSystem*) qui comprennent (*uses*) des plantes (*Plant*) qui contrôlent (*treats*) des agresseurs (*Pest*) pour protéger (*protects*) des cultures (*Crops*). Les plantes, les cultures et les agresseurs sont eux-même caractérisés (associations *xx_CharactBy*) par des informations, rassemblées dans *OrganismInfo*, comme l'espèce, le genre, la famille, et si l'organisme peut être consommé *food* ou utilisé en soin médical *medical*. Ce schéma est typique de l'analyse de données dans de nombreux domaines applicatif et fait apparaître de multiples relations.

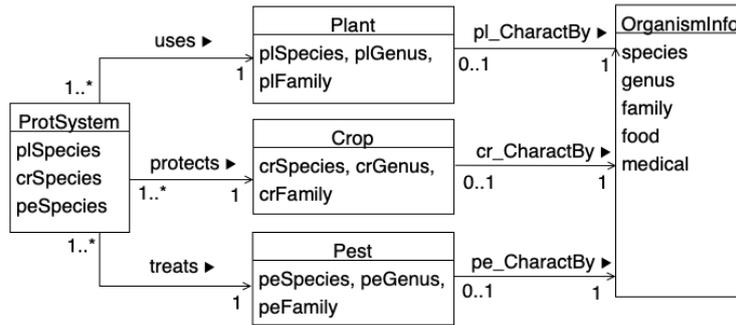


FIGURE 2 – Exemple de schéma entité-relation pour exporter puis analyser des données de Knomana

Dans ce travail, nous nous intéressons plus précisément à une analyse des données utilisant des règles d'implication extraites dans des structures construites par Analyse Relationnelle de Concepts. Cette extraction est décrite par Marach *et al.* [MGH⁺21].

2. Les règles d'implication sur des données multi-relationnelles

Une règle d'implication est une formule logique sous un format contraint. Elle comprend une prémisse et une conclusion, et se présente sous cette forme :

prémisse \Rightarrow *conclusion*

La prémisse et la conclusion sont constituées d'attributs descriptifs des entités, par exemple :

- $Food_X$ est un attribut *simple* qui indique qu'une entité peut être consommée, $Food_$ est un attribut *simple* qui indique qu'une entité ne peut pas être consommée,
- $\exists protects(Medical_X)$ est un attribut *relationnel* qui indique qu'un système de protection protège au moins une culture qui est utilisée en soin médical. \exists peut être omis si l'analyse ne comporte que ce quantifieur.

Nous donnons trois exemples de règles dans les figures 3, 4 et 5. Dans les sorties des algorithmes d'extraction, les attributs sont séparés par des virgules correspondant au "et" logique. Les légendes des figures donnent une lecture de la signification de la règle.

```
<006> protects(Medical_X),protects(Food_X),
protects(Species_CropS&Genus_CropG&Family_CropF),
treats(Food_),treats(Medical_),treats(Family_Noctuidae),treats(Genus_Helicoverpa),
treats(Species_HelicoverpaArmigera),uses(Medical_) => uses(Food_)
```

FIGURE 3 – Règle 1. 6 objets (où 1 objet = 1 description de l'utilisation d'une plante) supportent cette connaissance : Lorsque l'on protège une espèce qui est utilisée en soin médical et consommée, de l'espèce CropS, du genre CropG et de la famille CropF, contre un agresseur non alimentaire, non utilisé en soin médical, de la famille Noctuidae, de genre Helicoverpa et de l'espèce Helicoverpa Armigera, en utilisant un protecteur non utilisé en soin médical, alors le protecteur est non consommé.

```
<023> treats(PeFamily_Noctuidae),uses(PlFamily_Annonaceae),
protects(cr_CharactBy(Food_X)),protects(cr_CharactBy(Medical_X)) =>
protects(CrSpecies_CropS&CrGenus_CropG&CrFamily_CropF)
```

FIGURE 4 – Règle 2. 23 objets supportent cette connaissance : Quand on contrôle l'agresseur Noctuidae, en utilisant un protecteur qui est de la famille Annonaceae, si on protège une culture à usage médical et alimentaire alors on protège une culture de l'espèce CropS, du genre CropG et de la famille CropF.

```
<036> treats(PeFamily_Noctuidae),uses(PlFamily_Meliaceae),uses(pl_CharactBy(Food_)),
uses(pl_CharactBy(Medical_)) => treats(PeGenus_Spodoptera),protects(cr_CharactBy(Food_X)),
protects(cr_CharactBy(Medical_X))
```

FIGURE 5 – Règle 3. Cette règle est plus complexe : elle contient deux relations imbriquées. 36 objets supportent cette connaissance : Quand on contrôle l'agresseur Family_Noctuidae, en utilisant un protecteur qui est de la famille Meliaceae, qui n'est ni médical ni alimentaire, alors on contrôle l'agresseur PeGenus_Spodoptera, et on protège des plantes alimentaires et médicales.

Dans le cadre de la visualisation, deux métriques nous seront utiles sur des ensembles de règles respectant un certain format :

- Nombre d'implications
- **Max scope**, le *Scope* est le nombre d'objets pour lesquels la prémisse d'une règle est vérifiée. Il s'agit donc là d'avoir ce nombre maximum pour les règles vérifiant le format.

Par exemple, avec les règles présentées en figure 6 qui respectent le format *treats(...)* \Rightarrow *protects(...)* nous obtiendrons les valeurs de métriques suivantes :

- Nombre d'implications = 3
- **Max scope** = 22 car les trois règles ont pour scope respectif 17, 17 et 22.

```
<017> treats(Food_),treats(Medical_),treats(Family_Noctuidae),
treats(Genus_Trichoplusia) => protects(Medical_),protects(Family_Brassicaceae)

<017> treats(Food_),treats(Medical_),treats(Family_bbb),
treats(Genus_uuu) => protects(Medical_),protects(Family_xxx)

<022> treats(Food_),treats(Medical_),treats(Family_ccc),
treats(Genus_vvv) => protects(Medical_),protects(Family_yyy)
```

FIGURE 6 – Liste de règles pour illustrer le calcul des métriques.

L'accès aux règles de manière uniquement textuelle est assez difficile :

- Elles peuvent être en grand nombre, par exemple 4532 règles sont mentionnées dans l'article Marach *et al.* [MGH⁺21].
- Elles peuvent contenir de nombreux éléments de description (attributs et attributs relationnels).

- Trouver les règles correspondant à un certain format qui nous intéresse, par exemple, les règles donnant des informations sur des relations comme "uses" et des plantes utilisées en soin médical est délicat.

Pour toutes ces raisons, nous visons une restitution par une visualisation graphique dont nous pensons qu'elle pourrait aider l'expert dans son analyse.

Les besoins de la visualisation ont été définis comme suit :

- B1 Naviguer dans les entités relationnelles, selon des contraintes sur les prémisses et les conclusions.
- B2 Afficher la quantité de règles pour chaque format, puis les attributs qui les composent, et enfin les règles.
- B3 Permettre de naviguer dans un grand ensemble de règles de manière intuitive.
- B4 Trouver une règle à partir de son format.

Les règles seront calculées pour ce projet avec FCA4J (<https://www.lirmm.fr/fca4j/Introduction.html>) [GHM22].

III État de l'art

Dans cette partie, nous présentons des articles relatifs à la visualisation de règles qui nous ont inspirés pour nos travaux.

Il existe des outils permettant de naviguer dans des règles basés sur des composants classiques tels que les listes ou les boutons [SAVN⁺13]. Un exemple est présenté dans la figure 1. Il peut cependant être intéressant d'étudier comment les techniques de visualisation actuelles concernant les graphes et les données tabulaires peuvent répondre aux besoins soulevés précédemment.

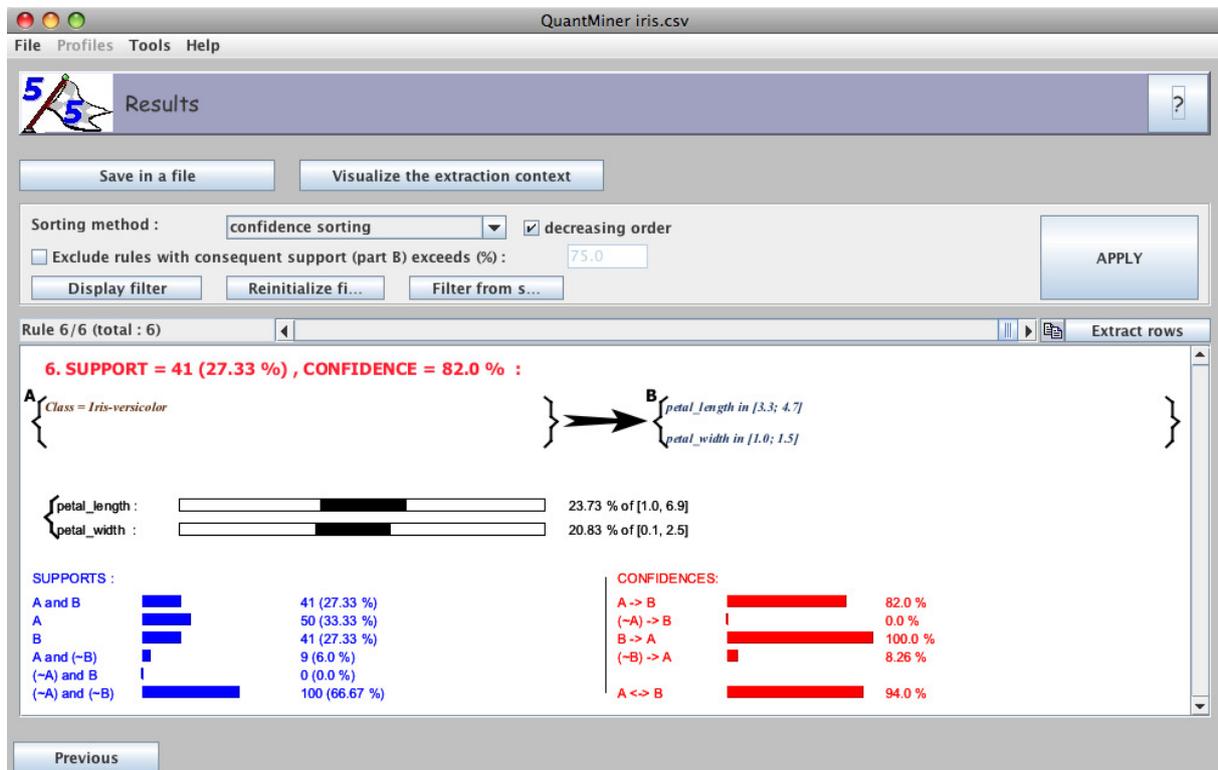


FIGURE 1 – Visualisation classique [SAVN⁺13].

1. Graphes

Les règles peuvent se modéliser sous forme de graphes dans lesquels les sommets représentent les prémisses et les conclusions, et les liens orientés représentent les implications. Les approches de visualisation les plus utilisées pour représenter les graphes sont les diagrammes nœuds-liens et les matrices d'adjacence [Mun14]. Il existe d'autres familles pour les arbres basées sur des régions imbriquées (e.g. tree-maps) ou adjacentes (e.g. sunbursts) mais on ne s'y intéressera pas ici car elles nécessiteraient une structuration redondante des données (voir [Sch11]).

Un diagramme nœuds-liens est composé d'objets graphiques ponctuels représentant les sommets et de lignes (ou de polygones) représentant les arêtes. Les représentations

à gauche de la figure 2 montrent deux exemples de graphes dessinés sous cette forme. [BGB03] et [Hah17] utilisent une telle approche (voir figure 3).

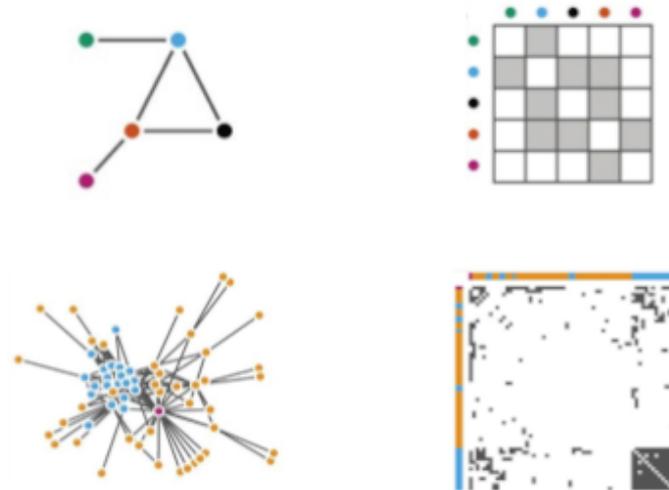


FIGURE 2 – Deux exemples de graphes dessinés à l’aide de deux techniques de visualisation : diagrammes nœuds-liens à gauche et matrices d’adjacence à droite [Mun14].

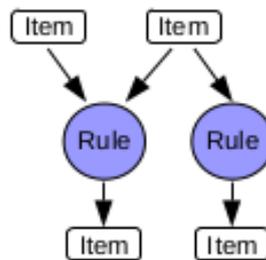


FIGURE 3 – Exemple de visualisation de règles avec un diagramme nœuds-liens [BGB03, Hah17].

La visualisation matricielle consiste à dessiner une matrice d’adjacence dans laquelle les arêtes correspondent à des cases colorées. Les représentations à droite de la figure 2 montrent deux exemples de graphes dessinés sous cette forme. Les matrices permettent de représenter un grand nombre de règles sans superposition. [Hah17] propose une telle approche (voir figure 4).

Les deux représentations sont équivalentes pour la visualisation du voisinage. Les diagrammes nœuds-liens ont pour principal intérêt de permettre de visualiser plus facilement les communautés et les chemins entre sommets dans un graphe peu dense, ce dont nous n’avons pas besoin ici dans un premier temps. Nous avons donc choisi des représentations de type matrices d’adjacence pour leur capacité à passer à l’échelle et le fait qu’elles évitent les phénomènes d’encombrement visuel.

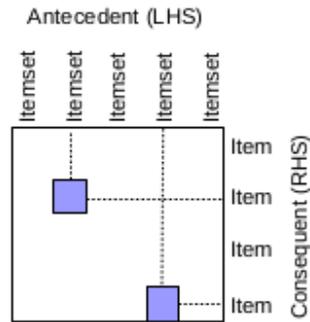


FIGURE 4 – Exemple de visualisation de règles avec une matrice d’adjacence [Hah17]

2. Tables

On peut vouloir visualiser des statistiques sur les règles comme des indices de confiance ou le support. Dans ce cas on peut utiliser les techniques de visualisation de données tabulaires [Mun14] telles que les nuages de points (comme dans [Hah17, LSZ⁺12] par exemple) avec des versions simples (voir figure 5) ou plus avancées (voir figure 6). Nous ne nous intéresserons pas à ce besoin dans RCAvizir dans un premier temps.

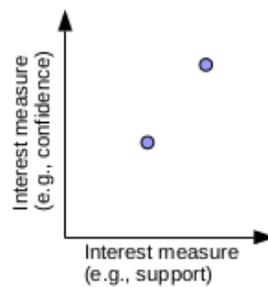


FIGURE 5 – Visualisation à l’aide d’un nuage de points simple [LSZ⁺12, Hah17].

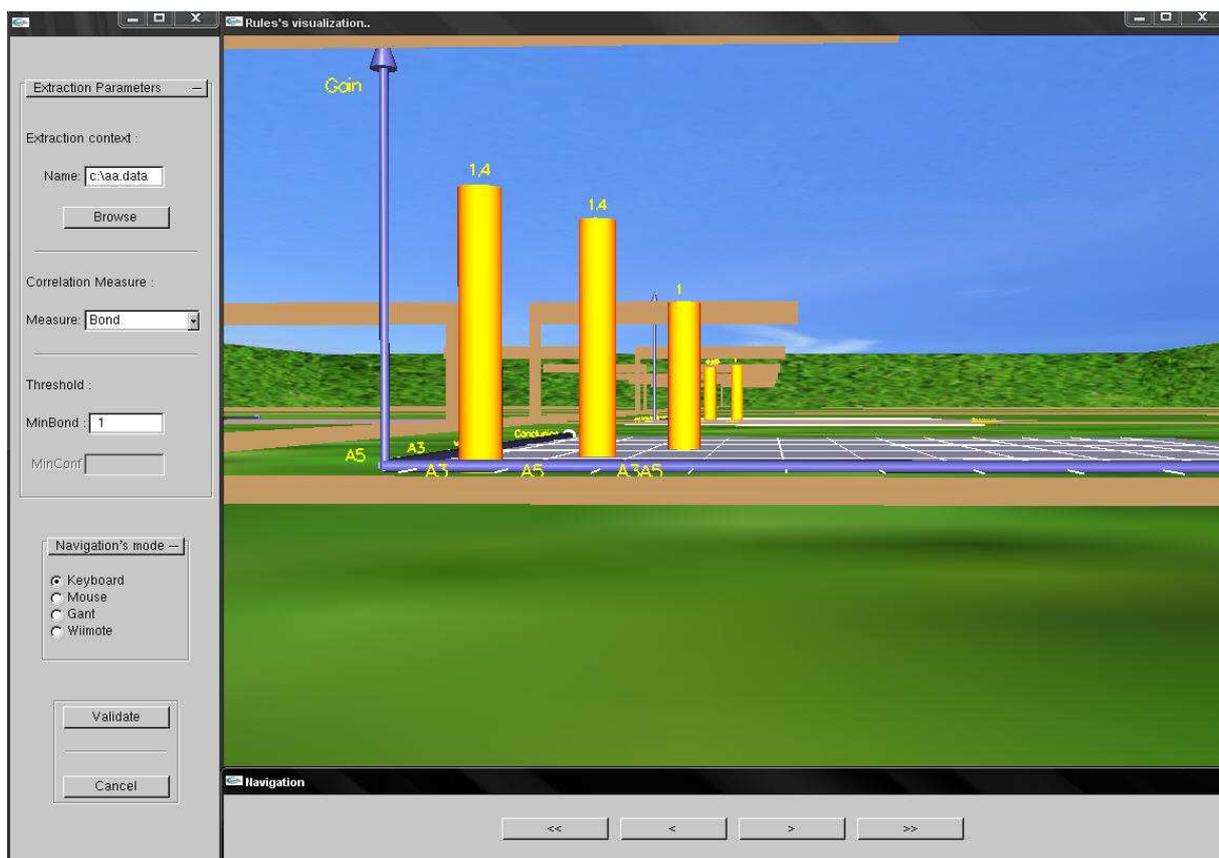


FIGURE 6 – Visualisation à l’aide d’un nuage de points en 3D : dans cet exemple, outre les positions x et y, la hauteur et le diamètre des cylindres représentent aussi des métriques [BGB03, MY10].

IV RCAvizIR

Dans ce chapitre nous présentons le prototype réalisé de façon à satisfaire les besoins/attendus présentés au chapitre II.

1. Principes de la visualisation

RCAvizIR est une application en ligne qui permet de visualiser et d'explorer des règles d'implications sur une page web. Un aperçu est présenté en figure 1. Le fond de la page comportera toute la partie graphique de base (en reprenant le thème graphique de RCAviz, réalisé en stage l'année dernière par Emile Muller <https://rcaviz.lirmm.fr/>). La barre du haut comprend les logos et un bouton About permettant d'avoir des informations sur l'application. Nous y ajouterons des boutons pour accéder à de exemples et à une documentation. La fenêtre principale se compose d'un volet de légende escamotable et trois volets de navigation.

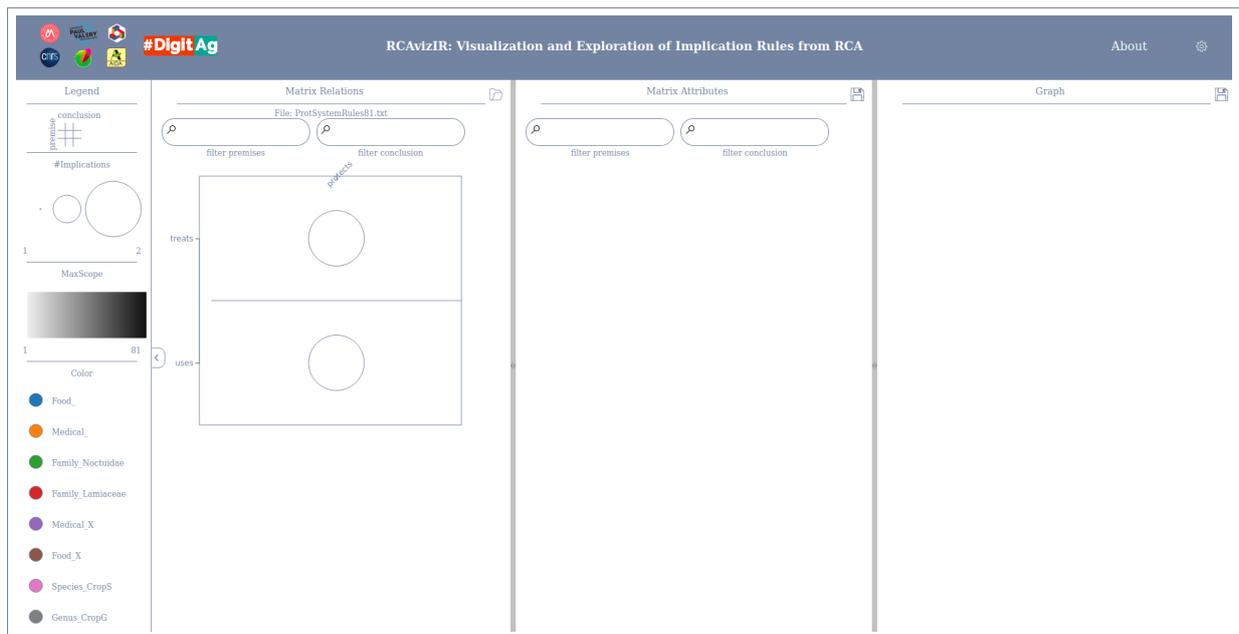


FIGURE 1 – Aperçu général de l'application prévue

La légende est détaillée en figure 2, elle contient :

1. un rappel du sens de lecture (prémisse en ligne et conclusion en colonne),
2. une échelle qui porte sur le nombre d'implications ($\#Implications$) qui est représenté par la taille des cercles (petit = 1, grand = 2 dans l'exemple),
3. une échelle de couleur qui porte sur le Max scope qui est représenté par un niveau de gris ou autrement dit de luminosité et qui est utilisé dans les cercles des visualisations des deux premiers volets de navigation présentés ci-dessous (clair = 1, foncé = 81 dans l'exemple),

- une couleur correspondant à la teinte utilisée sur le deuxième volet de navigation présenté ci-dessous et qui associe chaque catégorie à une couleur fixe.



FIGURE 2 – Détails du volet légende

Nous présentons en figure 3 un fichier de règles qui va servir à illustrer les autres volets. Il comprend quatre règles présentant des supports et des formats différents.

```

Éditeur de texte 25 août 13:32
Ouvrir ProtSystemRules20.txt Document 1 sans titre ProtSystemRules9.txt Prot351925.txt Enregistrer
*ProtSystemRules20.txt *Document 1 sans titre *ProtSystemRules9.txt *Prot351925.txt
1 Rules calculated from file: D:/projects/LinaProject/TwoTables/results/OrganismInfo.cex
2 <028> treats(Food_),treats(Medical_),treats(Family_Noctuidae) => protects(Food_X)
3 <050> treats(Food_),treats(Medical_),treats(Family_Noctuidae),uses(Family_Verbenaceae) =>
  protects(Medical_X),protects(Food_X)
4 <020> treats(Food_),treats(Medical_),treats(Family_Noctuidae),uses(Family_Myrtaceae) =>
  protects(Food_X)
5 <020>
  protects(Medical_X),treats(Food_),treats(Medical_),treats(Family_Noctuidae),uses(Family_Solanaceae)
  => protects(Food_X)

```

FIGURE 3 – Fichier de règles pour illustrer l'interface

Le premier volet de visualisation est le volet *Matrix Relation* présenté en figure 4. On retrouve les relations sur les entêtes des lignes et des colonnes. Avec les règles de l'exemple,

on va retrouver les relations `treats` et `uses` en lignes (elles apparaissent en prémisse) et la relation `protects` en colonne (elle apparaît en conclusion). La visualisation met en valeur le fait que dans cet exemple, il y a plus de règles au format “`treat \implies protect`” que de règles au format “`uses \implies protect`”; ce qui est traduit par un cercle plus grand dans le premier cas.

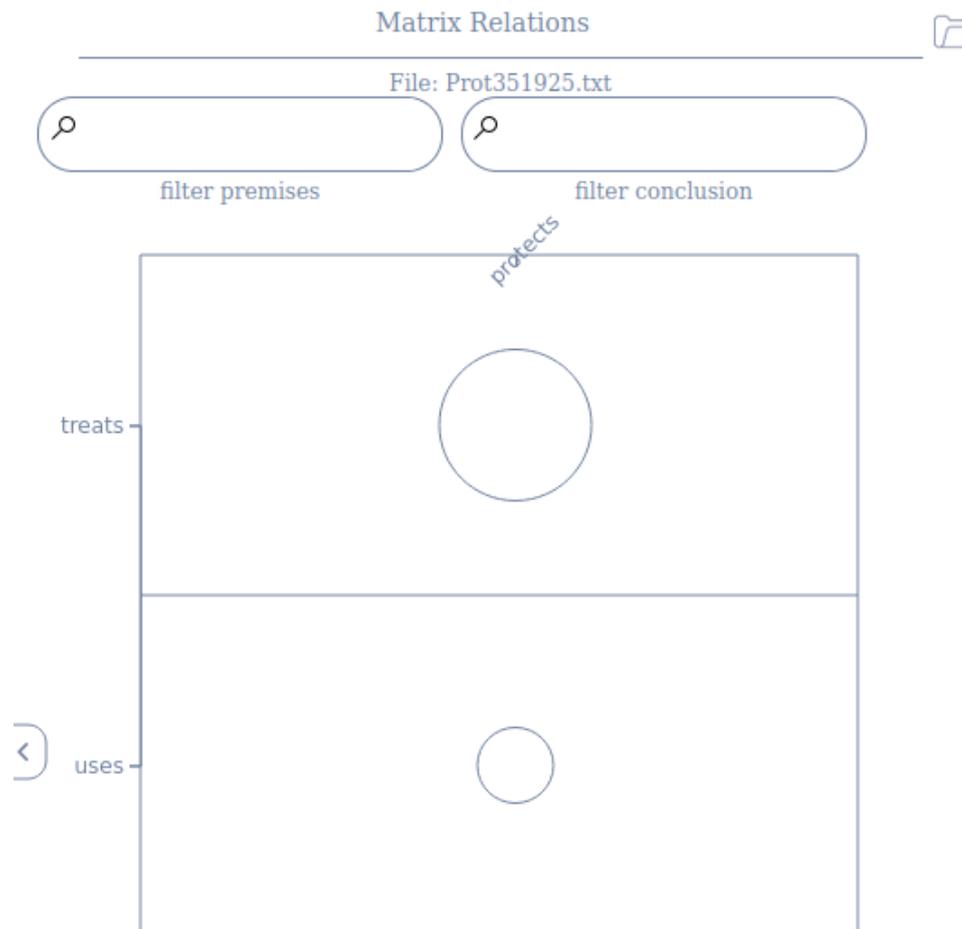


FIGURE 4 – Volet *Matrix Relation*

Le second volet est le volet *Matrix Attribute*, présenté en figure 5. Les entêtes de lignes et de colonnes sont des noms d’attributs comme `Food` et `Medical` et non plus des noms de relation. Ce qui est présenté concerne seulement les règles sélectionnées dans la première matrice en cliquant sur l’un des cercles. On visualise ainsi la répartition des éléments concernés par le format de règle de la première matrice.

Enfin le troisième volet, qui n’est pas encore développé, s’affichera après que l’on ait cliqué sur un cercle de la deuxième matrice. Il présentera les règles correspondantes. Nous présentons un schéma indicatif de ce qui sera réalisé en figure 6. On visualisera ici les règles filtrées par format de règle (par exemple de la forme `Protect \implies Treat`) et par élément (par exemple : `a \implies ?`).

En synthèse, lorsque l’on peut cliquer sur un cercle d’une matrice pour mettre à jour la suite de la visualisation, cela répond aux besoins B1, B3 et B4. En effet cela permet de filtrer progressivement les règles. On navigue progressivement en sélectionnant les contraintes une à une, le nombre de règles restantes diminue donc à chaque étape

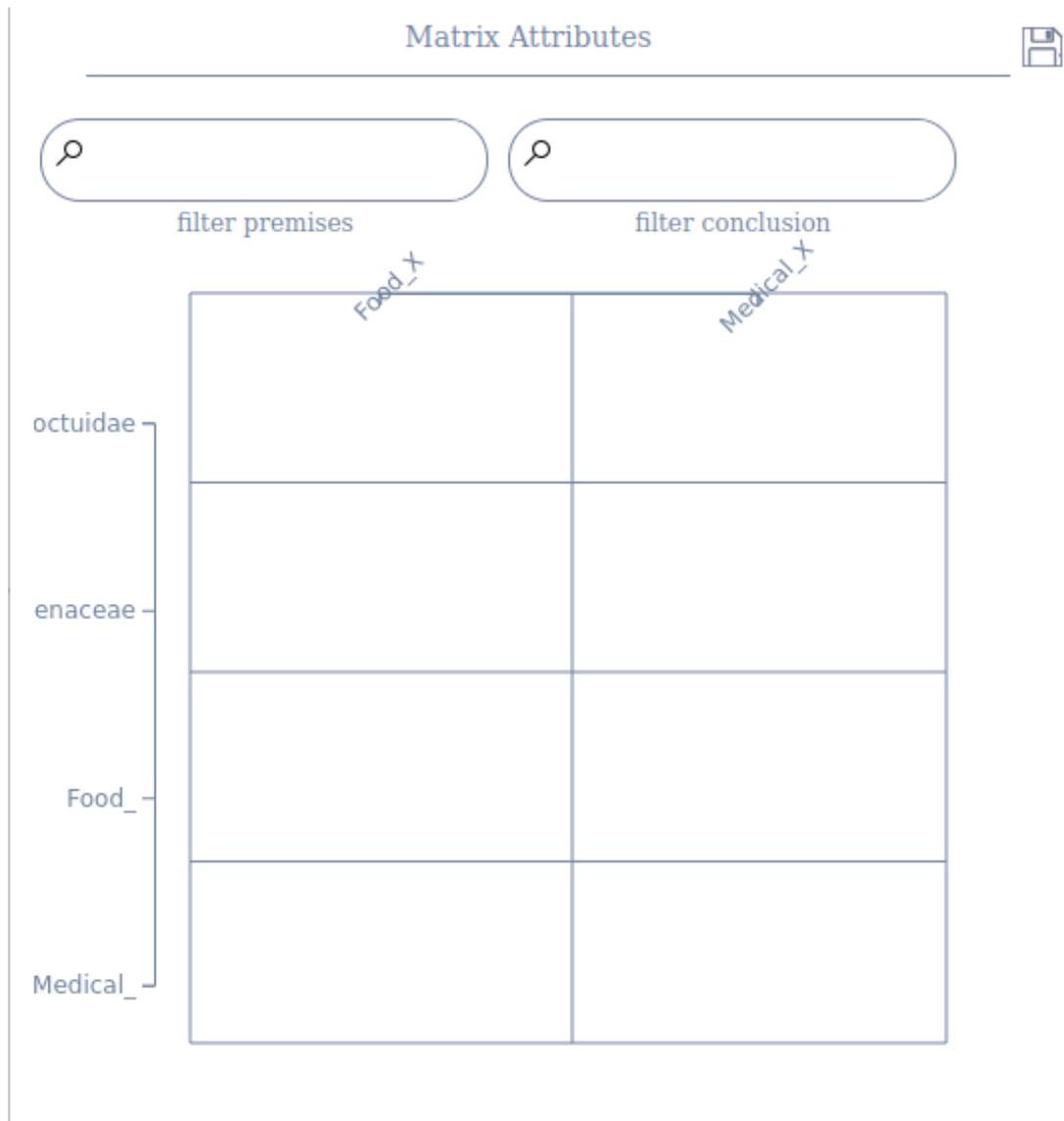


FIGURE 5 – Volet *Matrix Attribute*

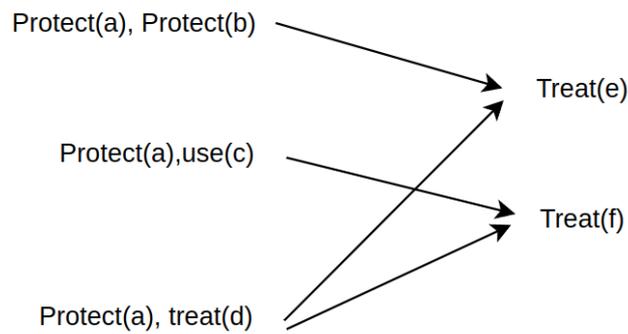


FIGURE 6 – Volet *Graph*

et permet donc de naviguer dans un grand ensemble de règles. On construit ainsi petit à petit le format de la règle recherché. La taille de chaque cercle ainsi que la légende

indiquent le nombre de règles concernées par chaque format. Cela répond au besoin B2. Il sera possible de filtrer les formats de règles et attributs via les champs de texte situés au dessus des matrices, afin réduire le nombre de lignes et de colonnes dans les matrices. Cela permettra de naviguer plus facilement dans un grand ensemble de règles. Cela répond aux besoins B3 et B1.

2. Implémentation

Le logiciel développé est une application web utilisable dans un navigateur ; ce qui la rend indépendante des systèmes d'exploitation des utilisateurs. La charte graphique est cohérente avec RCAViz (charte graphique et organisation). Le projet est développé en HTML, JavaScript (avec la bibliothèque D3.js [BOH11]) et CSS. Les calculs sont fait en JavaScript côté client en non en PHP coté serveur. Les données à visualiser sont fournies via un chargement de fichier et non issues d'une base de données. Il n'y a de ce fait aucun besoin d'authentification ni de gestion d'accès à des données sensibles. Les temps de calcul étant très faible, la génération de 3 images fixes en 2D côté client est interactive et ne nécessite donc pas l'utilisation d'un code côté serveur qui pourrait générer des temps de latence dus au temps de connexion. La bibliothèque D3.js a été utilisée pour réaliser :

- l'en-tête
- la légende
- les zones d'affichage interactives

a. Navigation

En cliquant sur la roue dentée en haut à droite de l'application, on peut changer la palette de couleur et télécharger un fichier contenant des données de test. Des images en SVG sont construites à la volée à partir des données du fichier d'entrée utilisées pour la navigation. L'affichage des matrices avec une image vectorielle (svg) (voir figure 8) a été préféré à l'utilisation d'un tableau HTML classique (table, tr, hr) (voir figure 7). Cela permet de faire un rendu plus pratique à lire (inclinaison des en-tête à 45°).

Après avoir ouvert un fichier, la légende se met à jour. La première section indiquant le sens de lecture ne change pas, les valeurs associés aux tailles de cercles et de niveaux de gris sont calculées en fonction du nombre maximum de règles et de valeurs de support rencontrées dans le fichier. Les différentes catégories d'item sont associées à une couleur dans la palette par défaut. Cette légende pourra être masquée au besoin. Dans la colonne à droite de la légende, une première matrice nous indique par des cercles le nombre de règles en fonction de leur format. Si 18 règles existent au format "protect \implies uses", un cercle ayant pour rayon celui associé à 18 dans la légende prend place sur la ligne de protect et dans la colonne de uses. En cliquant sur l'un d'entre eux, la 2e matrice affichera les attributs concernés par ces relations. Si parmi ces 18 règles, trois contiennent l'attribut Food_X en prémisses et Noctuidae en conclusion, un cercle ayant pour rayon la taille associée à 3 dans la légende prendra place dans la ligne de Food_X et dans la colonne de Noctuidae. En cliquant sur la 2e matrice, la liste des règles concernées par le format de relations et les attributs sélectionnés apparaîtra dans la 3e section de l'application sous forme de graphe.

	Lamiaceae	Rutaceae	Solanaceae	Food_N	Food_Y	Annona	CropG	Spodoptera	Medic
Annonaceae									
CropF			23						
Fabaceae									
Meliaceae									
Noctuidae									
Poaceae									
Rutaceae									
Food_N			17						
Food_Y									
Agrotis									
Capsicum									
Citrus									
CropG									
Melia									
Thymus									
Vitex									
Medical_N									
Medical_Y									
AgrotisSpp.									
CropS									

FIGURE 7 – Tableau HTML

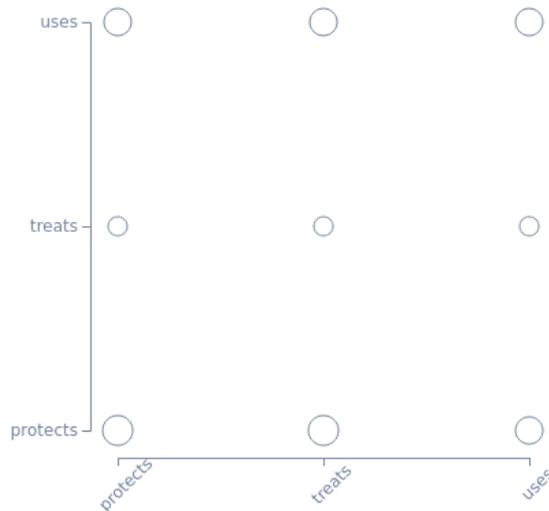


FIGURE 8 – Image vectorielle

b. Parseur

Un parseur a été réalisé pour lire les fichiers de règles qui sont dans un format de fichier textuel. La prémisse est séparée de la conclusion par les deux caractères ASCII "=>". L'analyse des règles est effectuée avec la méthode `split` de JavaScript plutôt qu'en utilisant des `RegEx`. La méthode `split` permet de découper la chaîne selon des caractères indiqués. Nous avons trois formats de règles :

- Règles sans attribut relationnel
Exemple : <004> Family_Malvaceae,Food_ => Medical_
- Règles avec attribut relationnel
<019> protects(Family_Brassicaceae),treats(Food_),
Exemple : treats(Medical_),treats(Family_Noctuidae) => protects(Medical_)

- Règles avec attributs relationnels imbriqués (nombre max. de niveaux considérés égal à 2)

Exemple : `<006> treats(PeFamily_Noctuidae),uses(PlFamily_Cupressaceae) => protects(cr_CharactBy(Food_X))`

A partir de l'analyse de la règle, on construit un objet JSON qui est un arbre représentant ses différentes parties : prémisse, conclusion, relation, attribut, etc. Ainsi la règle de la figure 9 devient :

```
{
  "id":1,
  "premRaw": "protects(Family_Fabaceae),treats(Food_),treats(Medical_),treats(Family_Noctuidae)",
  "concRaw": "protects(Medical_),protects(Food_X)",
  "support":9,
  "rankPremUses":-2,
  "rankPremTreats":0,
  "rankPremProtects":0,
  "rankConcUses":-2,
  "rankConcTreats":-2,
  "rankConcProtects":-2,
  "premProtects": "Family_Fabaceae",
  "premTreats": "Family_Noctuidae",
  "relPrem": ["protects","treats"],
  "relConc": ["protects"],
  "attPrem": ["Family_Fabaceae","Family_Noctuidae","Food_","Medical_"],
  "attConc": ["Food_X","Medical_"],
  "format": "protects,treats => protects"},
  {"protects,treats => protects":1},
  {"protects,treats => protects":9}
```



FIGURE 9 – Fichier de règles

Il est ainsi plus pratique d'accéder aux différents éléments : la chaîne de texte brut a été décomposée en une structure comportant le support, la prémisse et la conclusion, et pour ces deux dernières, l'ensemble des attributs qui les composent.

v Gestion de projet

Ce projet a été géré avec la méthode Agile. Des réunions plénières ont été organisées régulièrement (toutes les 2 à 3 semaines, étant donné le nombre de personnes à réunir elle ne pouvaient pas être plus fréquentes), accompagné d'un compte rendu listant les points abordés et les futures tâches à réaliser. Entre les réunions, nous faisons des points avec une partie du groupe de travail. La figure 1 présente une maquette qui a été réalisée avant de produire l'interface graphique.



FIGURE 1 – Maquette de l'application

Le code a été régulièrement déposé sur le git du lirmm. Les différentes tâches sont tracées dans le diagramme de Gantt de la figure 2. Le développement a été fait sur un serveur web local de mon PC, et une version a été déposée à l'adresse <https://rcavizir.s3-website.fr-par.scw.cloud/> pour des démonstrations intermédiaires avant de livrer la version finale sur un serveur du LIRMM.

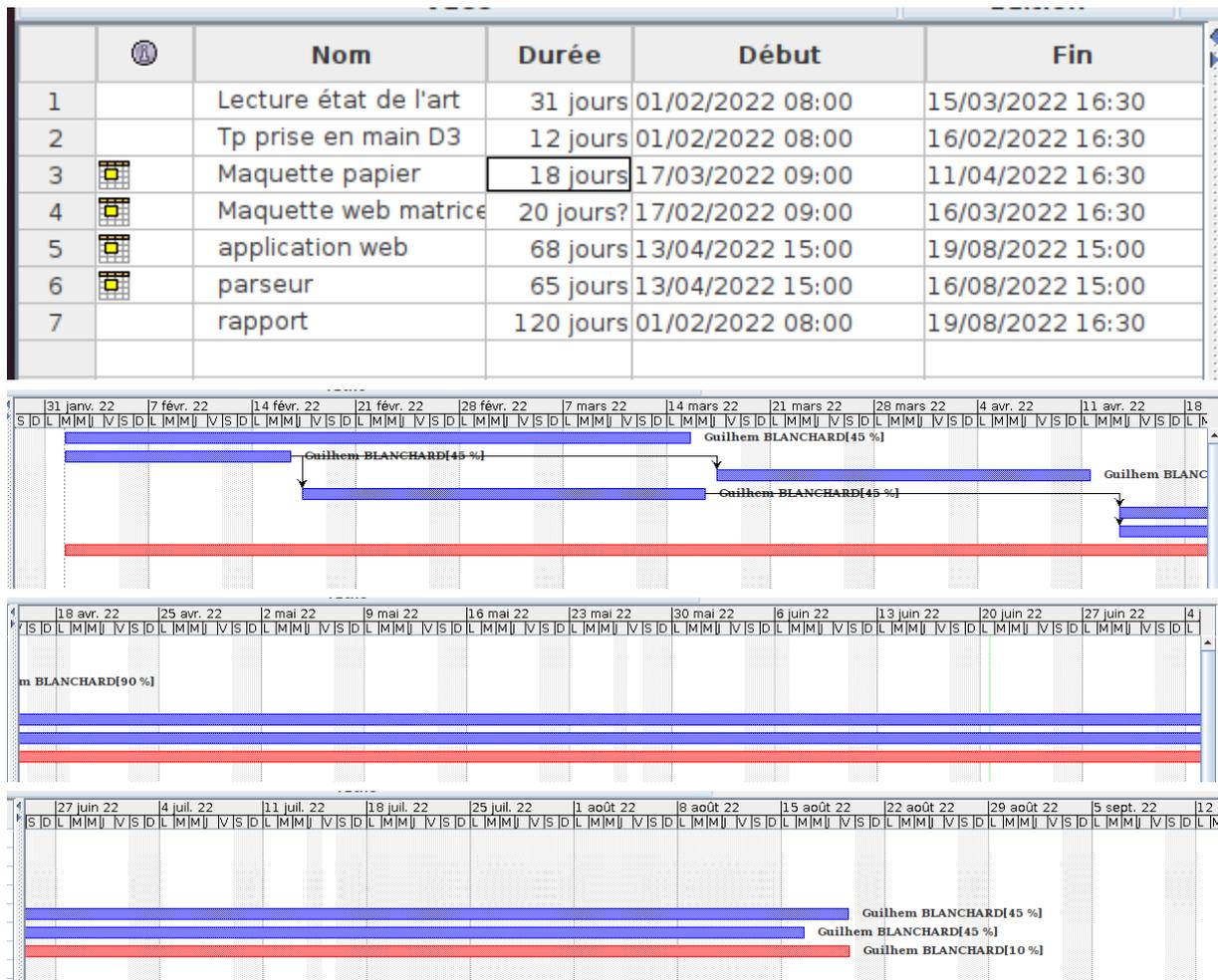


FIGURE 2 – Diagramme de Gantt indiquant les durées approximatives de chaque tâche

VI Conclusion

Lors de ce stage, j'ai pu réaliser une application web de visualisation de règles d'implication calculées pour un ensemble de données multi-relationnelles, et organisées avec RCA. Pour cela, j'ai utilisé des matrices et proposé un filtrage et des moyens de sélection. Deux points importants ont été la création du parseur de formule logique et la génération des images SVG à la volée pour la navigation. J'avais déjà réalisé un parseur pour un langage de programmation mais jamais pour des formules logiques.

Les travaux envisagés à court terme sont les suivants :

- la mise en place du filtrage par relations ou par attributs,
- la finalisation de la 3e colonne de l'interface (graphes),
- la considération des attributs non relationnels dans les deux matrices,
- la réalisation des tests systématiques avec des utilisateurs finaux.

Les travaux à prévoir à moyen ou long terme sont :

- analyser la complexité théorique et pratique du parseur,
- étudier des règles dans le cas de modèles de données comportant des circuits, avec des mises à jour de la matrice de droite vers la matrice de gauche (et non pas seulement l'inverse comme actuellement),
- rendre l'interface adaptative pour des attributs relationnels imbriqués avec plus de 2 niveaux (par ex. $\exists serve(\exists hasForIngredient(\exists conformToDiet(C))))$),
- gérer des quantificateurs logiques autres que le quantificateur existentiel.

Ce stage m'a permis de me former sur le fonctionnement du travail académique (travaux avec des chercheurs et recherche d'articles pour l'état de l'art) et de progresser notamment en développement web. J'ai pu approfondir mes connaissances en JavaScript et dans l'usage de la librairie D3.js ainsi que dans l'analyse de concepts formels.

À l'issue de mon stage, en août, je poursuivrai ce travail un mois et demi de plus, dans le cadre d'un CDD, afin d'améliorer mon produit ; ce qui permettra au CIRAD d'avoir le meilleur résultat possible et ce qui me permettra aussi de le présenter dans mon portfolio avec mon CV pour mes futures recherches d'emploi.

Bibliographie

- [BGB03] Julien Blanchard, Fabrice Guillet, and Henri Briand. Exploratory visualization for association rule rummaging. In *KDD-03 workshop on multimedia data mining (MDM-03)*, volume 3, 2003.
- [BOH11] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12) :2301–2309, 2011.
- [GHM22] Alain Gutierrez, Marianne Huchard, and Pierre Martin. FCA4J : A Java Library for Relational Concept Analysis and Formal Concept Analysis. In Alexandre Bazin, Karel Bertet, Christophe Demko, Pierre Martin, and Ants Torim, editors, *Workshop ETAFCA'2022@CLA2022 : Existing Tools and Applications for Formal Concept Analysis*, pages 209–214, Tallinn, Estonia, June 2022. CLA Conference Series (cla.inf.upol.cz).
- [Hah17] Michael Hahsler. arulesViz : Interactive visualization of association rules with r. *The R Journal*, 9(2) :163, 2017.
- [LSZ⁺12] Guimei Liu, Andre Suchitra, Haojun Zhang, Mengling Feng, See-Kiong Ng, and Limsoon Wong. Assocexplorer : an association rule visualization system for exploratory data analysis. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1536–1539, 2012.
- [MGH⁺21] Lina Mahrach, Alain Gutierrez, Marianne Huchard, Priscilla Keip, Pascal Marnotte, Pierre Silvie, and Pierre Martin. Combining implications and conceptual analysis to learn from a pesticidal plant knowledge base. In Tanya Braun, Marcel Gehrke, Tom Hanika, and Nathalie Hernandez, editors, *Graph-Based Representation and Reasoning - 26th International Conference on Conceptual Structures, ICCS 2021, Virtual Event, September 20-22, 2021, Proceedings*, volume 12879 of *Lecture Notes in Computer Science*, pages 57–72. Springer, 2021.
- [Mun14] Tamara Munzner. *Visualization Analysis and Design*. A.K. Peters visualization series. A K Peters, 2014.
- [MY10] Amira Mouakher and Sadok Ben Yahia. Anthropocentric visualisation of optimal cover of association rules. In *CLA*, pages 211–222. Citeseer, 2010.
- [SAVN⁺13] Ansaf Salleb-Aouissi, Christel Vrain, Cyril Nortet, Xiangrong Kong, Vivek Rathod, and Daniel Cassard. Quantminer for mining quantitative association rules. *The Journal of Machine Learning Research*, 14(1) :3153–3157, 2013.
- [Sch11] Hans-Jorg Schulz. Treevis. net : A tree visualization reference. *IEEE Computer Graphics and Applications*, 31(6) :11–15, 2011.