ManagHAL

Martin AMIENS

2024-06-16

En 2017, AgroParisTech a signé l'Appel de Jussieu, visant à promouvoir la bibliodiversité, c'est-à-dire la diversité éditoriale pour éviter la domination des grands groupes. En 2020, l'institution a adopté une politique de 'science ouverte' pour faciliter la libre diffusion des résultats de la recherche scientifique et la transparence des processus de recherche. En réponse à ses engagements, le 2 janvier 2023, AgroParisTech crée une cellule HAL pour accompagner ses chercheurs dans l'utilisation de l'archive ouverte HAL. Cependant, un package R pour l'extraction et l'analyse des données HAL n'existait pas, ce qui a conduit à la mission de Théodore Vanrengterghem de développer le package 'ManagHAL'. Par la suite, j'ai eu l'opportunité de reprendre le projet "ManagHAL" durant mon stage financé par la Direction de la Recherche et de l'Innovation d'AgroParisTech. Ainsi, j'ai pu travailler sur la mise en place de réseaux à partir des données, l'adaptation et l'intégration d'outils d'analyse statistique de réseaux tel que la modélisation en blocs stochastiques dans le package 'ManagHAL'.

Table of contents

1 Mémoire de Stage

En 2017, AgroParisTech a signé l'Appel de Jussieu, visant à promouvoir la bibliodiversité, c'est-à-dire la diversité éditoriale pour éviter la domination des grands groupes. En 2020, l'institution a adopté une politique de 'science ouverte' pour faciliter la libre diffusion des résultats de la recherche scientifique et la transparence des processus de recherche. En réponse à ses engagements, le 2 janvier 2023, AgroParisTech crée une cellule HAL pour accompagner ses chercheurs dans l'utilisation de l'archive ouverte HAL. Cependant, un package R pour l'extraction et l'analyse des données HAL n'existait pas, ce qui a conduit à la mission de Théodore Vanrengterghem de développer le package 'ManagHAL'. Par la suite, j'ai eu l'opportunité de reprendre le projet "ManagHAL" durant mon stage financé par la Direction de la Recherche et de l'Innovation d'AgroParisTech. Ainsi, j'ai pu travailler sur la mise en place de réseaux à partir des données, l'adaptation et l'intégration d'outils d'analyse statistique de réseaux tel que la modélisation en blocs stochastiques dans le package 'ManagHAL'.

1.1 INTRODUCTION

En 2017, AgroParisTech signe l'Appel de Jussieu. Cet appel a été rédigé dans l'objectif de mettre en avant des engagements pour promouvoir la Bibliodiversité. La bibliodiversité correspond à la diversité dans le monde éditorial afin d'éviter une appropriation des espaces de ventes par des grands groupes éditeurs. Dans la continuité de la signature de l'Appel de Jussieu, AgroParisTech en partenariat avec INRAE, a adopté en 2020, une politique d'établissement dédié à la science ouverte (AgroParisTech (2024) , voir : La science ouverte à AgroParisTech – Politique d'établissement janvier 2020). Pour AgroParisTech, la science ouverte correspond à l' « ensemble des principes et des actions qui facilitent l'ouverture et la libre dissémination, à destination de l'ensemble de la société, des productions de la recherche scientifique, ainsi que la transparence et l'ouverture des processus de la recherche » (AgroParisTech (2024)) . Un exemple concret des principes appliqués par l'établissement sont les principes FAIR pour la gestion des données de recherches. Ces derniers stipulent que les données doivent être :

- Findable : facilement trouvable.
- Accessible : avec des conditions d'accès clairement définies.

- Interoperable : intégrable et capable d'interagir avec d'autres outils extérieurs ou données.
- Reusable : réutilisables, avec toutes les informations nécessaires à la compréhension de ces dernières et au cadre d'utilisation autorisé par les auteurs.

Afin de progresser vers l'accomplissement des objectifs énoncés par AgroParisTech, de nombreuses actions ont été mises en place. Notamment, la création, le 2 janvier 2023, d'une cellule HAL au sein de l'établissement. Cette cellule HAL est dirigée par la Direction de la Recherche, de l'Innovation et du Transfert Technologique et par la Direction de la Documentation et du Patrimoine Culturelle. La cellule a pour vocation à faciliter l'intégration de l'archive ouverte HAL et des services associés dans la communauté de recherche d'AgroParisTech. Elle assure et veille à la qualité des métadonnées et référentiels pour le portail HAL-AgroParisTech (AgroParisTech (2024) , voir : Cellule HAL d'AgroParisTech Missions et Services). HAL est « l'archive nationale choisie par la communauté scientifique et universitaire française pour la diffusion ouverte de ses résultats de recherche. » (Comité d'orientation du CCSD & Assemblée des partenaires HAL (2022)). HAL dispose d'un API (Application Programming Interface) complexe et très diverse. Ce dernier est un programme permettant à d'autres applications comme Rstudio de communiquer et d'échanger des données avec HAL. L'API HAL offre de nombreux choix spécifiques dans l'extraction de données et dans l'analyse statistique de ces données. La construction d'une requète permettant la récupération de données HAL spécifiés par le constructeur de la requète passe par la connaissance de cet API. Une explication plus détaillée de cet API et d'une requète HAL sera faite dans la section méthode.

Bien qu'il existe des outils d'extraction de publications HAL et de création de rapport bibliographique en ligne comme ExtrHAL (LJonchere (n.d.)), il n'existe pas aujourd'hui de package R permettant d'extraire ces données et de réaliser des analyses statistiques sur ces dernières. C'est pourquoi en réponse à un besoin de l'unité MIA du partenariat entre AgroParisTech et INRAE, Théodore Vanrengterghem, ingénieur de recherche, s'est lancé dans la programmation d'un package R nommé "ManagHAL" réalisant l'extraction de donnée HAL et la création d'un bilan bibliographique.

R est un langage de programmation gratuit et en open-source, qui a été conçu pour réaliser des analyses statistiques et manipuler des données. Rstudio, est l'environnement de programmation gratuit utilisant le langage R. Dans Rstudio, il existe la possibilité de télécharger des packages. Ces derniers sont des fichiers contenant des fonctions et des données qui partagent un même thème. Ils sont très utilisés par les utilisateurs de Rstudio car les packages facilitent le développement. Il existe actuellement plus de 18000 packages en ligne.

Durant les 2 mois et 17 jours de mon stage, j'ai eu la chance d'être accueilli au sein du département Modélisation Mathématiques, Informatique, et Physique d'AgroParisTech Palaiseau dans l'unité de recherche Mathématiques et Informatiques Appliquées. Cette unité MIA est le résultat de la collaboration entre AgroParisTech et INRAE, J'ai eu l'opportunité d'être encadré par deux tuteurs nommés Julie Albert et Pierre Barbillon qui m'ont accompagné tout

au long de cette expérience. L'unité Mathématiques et Informatiques Appliquées est composée de nombreux chercheurs, doctorants et stagiaire, et est hébergé au sein du bâtiment E d'AgroParisTech Palaiseau. J'ai travaillé sur l'adaptation, la programmation, et l'intégration d'outils de constructions de réseau, et d'outils d'analyse statistiques de réseau au package ManagHAL. J'ai donc pu travailler à la mise en forme du package, à la création de fonctions. Mon stage était financé par la Direction de la Recherche, de l'Innovation et du Transfert Technologique de la cellule HAL d'AgroParisTech présenté dans les paragraphes précédents.

Afin de comprendre les choix que j'ai réalisé lors du développement du package, il est nécessaire de présenter le package d'origine avant mes modifications. Un point rapide sera fait au cours de la partie contexte statistique sur les réseaux et les méthodes d'analyse statistique de réseau que j'ai utilisé. Le package "ManagHAL" contenait des fichiers essentiels dans la construction d'un package R tel que buildRignore par exemple (contenant la liste des fichiers à ignorer lors de la construction du package sur l'ordinateur de l'utilisateur). Il contenait aussi des scripts R (i.e. files/fichiers) de fonctions contenant les fonctions principales du package. Celles-ci permettent de récupérer toutes les publications associées à un unique ou à des multiples identifiants d'auteurs HAL, provenant d'un fichier tabulé d'identifiants fournie par les ressources humaines. Lors de la récupération, il est possible de spécifier les données bilbiographiques des publications à récupérer. Ces dernières se présentent sous le format d'un fichier csv. Certaines de ces fonctions permettaient aussi de créer un rapport bibliographique automatique en ne donnant en entrée uniquement un identifiant auteur. Il existait des fonctions utiles au développement du package, afin de nettoyer mes données. Prenons par exemple "BorneDate" qui remplace les valeurs manquantes dans un vecteur de dates par une date limite spécifiée. Enfin, des fonctions permettaient réaliser des figures à partir des données des publications récupérés depuis HAL tel que des wordclouds ou des graphes.

Paragraphe précédent ne me convient pas, je vais le retravailler !!!

La finalité du package "ManagHAL" est de faciliter l'utilisation et la compréhension des données de publications. Afin de manipuler les données de publications et comprendre celles-ci, l'approche adoptée est de réaliser des réseaux et ensuite d'utiliser des outils d'analyse statistique de réseau. Les outils principaux utilisés à l'avenir dans "ManagHAL" seront les modélisations en blocs stochastiques ou latents. Mon mémoire est organisé de la manière suivante: En premier lieu, le contexte statistique nécessaire à la compréhension des outils utilisés est présenté. Puis, dans la partie suivante sont abordés, les méthodes informatiques développées et les résultats d'analyse de données-exemples obtenus. (pour lequel tout marche, précisez que je fais ça pcq mon stage n'est pas terminé). Dans un troisième temps, les méthodes déjà publiées utilisées sont précisées. Enfin, dans une ultime partie, les résultats sont discutés.

1.2 CONTEXTE STATISTIQUE

Cette partie est principalement basée sur la publication "Using Latent Block Models to Detect Structure in Ecological Networks" Aubert et al. (2022)

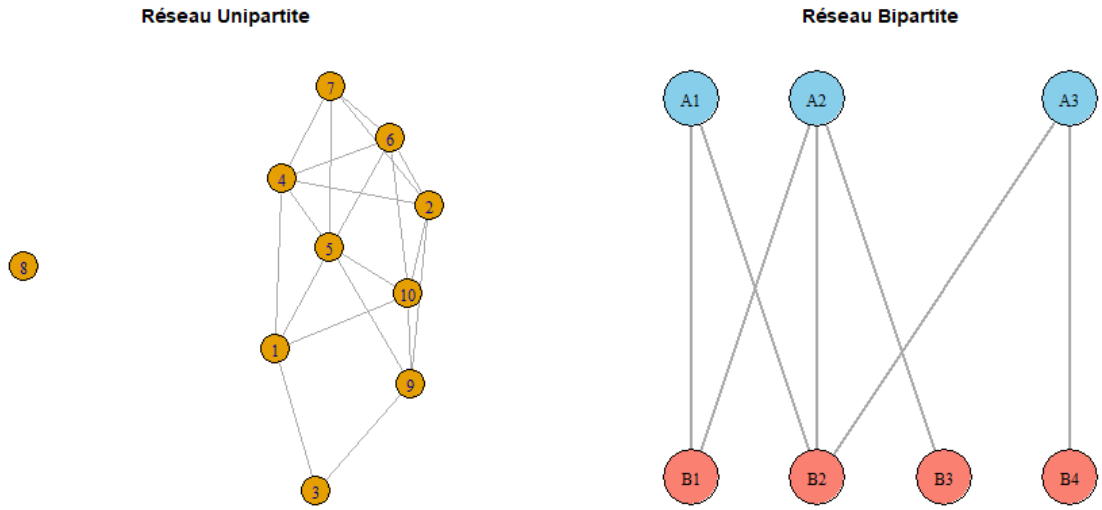
Un réseau d'interaction est composé de nœuds (les entités) et d'arêtes (les liens entre ces entités). Toutes entités nommables peuvent être modèlisés sous forme de noeuds, et leur interactions entre elles peuent être modélisés par des arrêtes. On peut prendre pour exemple les réseaux de gènes à partir de données de co-expression. Les réseaux sont répartis en deux groupes principaux: les réseaux Unipartites, où tous les nœuds sont du même type, et les réseaux Bipartites, où il y a deux types distincts de nœuds et les liens ne se forme qu'entre des nœuds de types différents. Les interactions dans un réseau peuvent être binaires ou pondérées. Un réseau peut être encodé dans une matrice Y d'adjacence telle que :

$$\begin{aligned} Y_{ij} &\neq 0 \quad \text{si un lien existe entre les nœuds } i \text{ et } j, \\ Y_{ij} &= 0 \quad \text{si aucun lien n'existe entre les nœuds } i \text{ et } j. \end{aligned}$$

Vous trouverez ci-dessous des exemples de code R pour créer et visualiser des réseaux unipartite et bipartite.

```r
# Création d'un réseau unipartite
set.seed(42) # Pour la reproductibilité
unipartite <- erdos.renyi.game(10, 0.3)
# unipartite_graph <- plot(unipartite, main = "Réseau Unipartite", width = 800, height = 600)
# print(unipartite_graph)
matrice_adjacence_unipartite <- as.matrix(unipartite, "adjacency")

# Création d'un réseau bipartite
# Noms des nœuds
nodes_A <- c("A1", "A2", "A3")
nodes_B <- c("B1", "B2", "B3", "B4")
# Liste des arêtes
edges <- c("A1", "B1", "A1", "B2","A2", "B1", "A2", "B2", "A2", "B3", "A3", "B2", "A3", "B4")
# Création du graphe
bipartite <- graph(edges, directed = FALSE)
V(bipartite)$type <- bipartite_mapping(bipartite)$type
plot(bipartite, layout = layout_as_bipartite, vertex.color = c("skyblue", "salmon")[V(bipartite
     vertex.label.color = "black", vertex.shape = "circle", vertex.size = 30, edge.width = 2
matrice_adjacence_bipartite <- as.matrix(bipartite, "adjacency")
```



mettre une légende avec titre important !!!

Nous décrivons les noeuds d'un réseau comme : $V = 1, .., N$, (N correspond à la taille du réseau). Les arrêtes peuvent être décrit comme $E = ...$.

Il existe plusieurs approches différentes pour analyser la structure d'un réseau. Dans ce mémoire, une seule approche sera considérée. Elle suppose l'existence de groupe fonctionnels partageant un même patron de connexion. Nous nous baserons par la suite sur une méthode possible pour cette approche: les modèles probabilistes génératifs.

Les Modèles en Bloc Stochastiques (aussi appelé SBM) sont des modèles probabilistes. Ils supposent que les nœuds d'un même réseau sont divisés en blocs (groupe, clusters) latents regroupant les entités ayant des modèles de connectivité similaires. Latents signifie qu'il n'y a pas d'apriori sur le type de structure recherchée. Ces modèles sont utiles pour détecter les communautés, les clusters aux sein d'un réseau complexe. Pour réaliser un modèle en bloc stochastique, il est nécessaire de supposer que le réseau obtenu à partir de mes données est une réalisation de mon modèle.

Les noeuds sont partitionnés en K groupes latents. On définit un vecteur Z où $Zi = k$, k est le bloc auquel appartient le noeud i . Dans le cas d'un réseau unipartite, on définit :

- $P(Zi = k) = \pi_k$, comme la probabilité d'appartenance au groupe k.
- $P(Y_{ij} = 1 | Z_i = k, Z_j = k') = \gamma_{kk'}$, comme la probabilité pour une paire de noeuds appartenant à un bloc k et à un bloc k' d'être en interaction (dans le cas binaire).

Nous avons donc comme paramètres du modèle $\theta = (\gamma, \pi)$. A partir du réseau connu, il est possible d'inférer les paramètres du modèle. Par nécessité d'écourter le rapport, une explication

détaillée ne sera pas réalisée. Je vous conseille cependant d'aller voir la publication "Using Latent Block Models to Detect Structure in Ecological Networks" Aubert et al. (2022) où les explications à ce propos sont claires et précises.

1.3 METHODES

Comme précisé dans l'introduction l'entièreté des méthodes informatiques développés ont été réalisé sur Rstudio. J'ai donc travaillé avec le language R. Les packages utilisés sont : dplyr (**pourquoi ?**), mailR(**pourquoi ?**), **... faire la liste des packages utilisés ...**

Expliquer l'api HAL ? La construction d'une requete consiste en réalité en la construction d'une url contenant différents paramètres. Ainsi, une requète est composée des paramètres suivants : … expliquer la construction d'une requete HAL

1.4 RESULTATS

Cette section présente les résultats obtenus au cours de mon stage. Elle est divisée en deux parties principales: les fonctions et algorithmes développés, modifiés ainsi que les résultats obtenus via un exemple. Cet exemple consistera en des données provenant des Ressources Humaines du Laboratoire MIA d'AgroParisTech. Chaque fonction est expliquée en détail, suivie des raisons justifiant son développement. Il est important de noter que mon stage se terminant le 26 juillet, Le package est donc encore en développement. Les fonctions présentées fonctionnent dans le cadre de l'exemple. Dans cette section, les noms de *fonctions* seront en italique et les noms des **scripts R** contenant les fonctions seront en gras.

1.4.1 Accessibilité et généricité de ManagHAL

Lors de mon stage, j'ai d'abord décidé de rendre le package plus accessible et générique. Pour ce faire, il m'était nécessaire de réorganiser et modifier certaines fonctions déjà présentes codées par Théodore Vanrengterghem.

1.4.1.1 Modifications de la fonction *load_mia_table*

J'ai commencé par modifier dans le fichier **mia_table.R** la fonction *load_mia_table*. A l'origine, cette fonction permettait le chargement d'un csv en ligne depuis une addresse spécifique vers un fichier contenant des informations tel que le nom, le prenom, l'idhal, l'equipe, etc, des personnes appartenant au labo. J'ai fait le choix de renommer le fichier **load_table.R**. J'ai remplacé la fonction *load_mia_table* par trois fonctions *load_team_table_csv*, *load_team_table_url*, et *load_team_table*. *load_team_table_csv* permet de charger un

csv dans Rstudio depuis un fichier local de la machine de l'utilisateur. ***load_team_table_url*** permet de charger un csv dans Rstudio depuis un fichier en ligne. *load_team_table* fait appel aux deux fonctions précédentes et permet par le biais d'argument de fonctions de spécifier la provenance du fichier et de charger le csv soit depuis un url soit depuis un fichier local. J'ai aussi modifier la majorité des fonctions déjà présentes dans le package faisant appel à *load_mia_table*. Les paramètres en entrée des trois fonctions sont les même que la fonction ***load_mia_table*** d'origine avec l'ajout d'un paramètre permettant à l'utilisateur de rentrer l'adresse du fichier. J'ai ensuite crée une documentation via le package "Roxygène2" conforme aux critères requis pour un package mis en ligne (voir : exemple d'une documentation classique en annexe) pour chacune des trois fonctions.

La fonction originelle ***load_mia_table*** ne permettait que de charger de manière spécifique le csv fourni par les Ressources Humaines depuis un url seafile. Cette fonction n'était pas générique. Ainsi, il était nécéssaire de la modifier. J'ai décidé de diviser en trois fonctions différentes afin de permettre à l'utilisateur de choisir de charger des tables de provenance locale ou depuis internet. J'ai décidé pour ***load_team_table_url*** de garder en majeur partie le code d'origine et de simplement le réadapter par soucis de temps. J'y ai ajouté des conditions de présence de colonnes obligatoires afin d'obliger l'utilisateur à fournir des colonnes sous un format spécifique.

```
load_team_table_url <- function(filter_id = TRUE,
                                date_cols,
                                url) {
  idhal <- NULL
  tmp_file <- paste0(tempfile(), ".xlsx")
  download.file(
    url = url,
    destfile = tmp_file,
    mode = "wb"
  )
  nbcol <- ncol(readxl::read_xlsx(tmp_file,
                                  sheet = "Membres"))
  col_types <- rep("text",nbcol)
  col_types[date_cols] <- 'date'


  table <- readxl::read_xlsx(tmp_file,
                             sheet = "Membres",
                             col_types = col_types
  ) %>%
    dplyr::rename_with(
      clean_names
    ) %>%
```

```
    dplyr::mutate_if(function(x) {
      "POSIXt" %in% class(x)
    }, ~ format(.x, "%d/%m/%Y"))


  if (filter_id) {
    table <- suppressWarnings(table %>%
                                dplyr::filter(!is.na(as.numeric(idhal))) %>%
                                dplyr::mutate(idhal = as.numeric(idhal)))
  }

  # Test if there is a column named "id"
  # Possibility to add mandatory columns in the future
  # Just add "OR"
  if ( !("idhal" %in% names(table)) | !("nom" %in% names(table)) | !("prenom" %in% names(tabl

  return(table)
}
```

load_team_table_csv est basé sur la construction de *load_team_table_url*. Je l'ai adapté pour pouvoir charger depuis un fichier local en utilisant *read.csv*. De même que dans *load_team_table_url*, j'y ai ajouté des conditions de présence de colonnes obligatoires afin d'obliger l'utilisateur à fournir des colonnes sous un format spécifique.

```
load_team_table_csv <- function(filter_id = TRUE,
                                date_cols,
                                filepath){
  idhal <- NULL
  nbcol <- ncol(read.csv(filepath, header = TRUE, sep = ";", encoding = "UTF-8"))

  # define column types (dates or text if not dates)
  col_types <- rep("character", nbcol)
  col_types[date_cols] <- 'character'

  table <- read.csv(filepath,
                    header = TRUE,
                    sep = ";",
                    encoding = "UTF-8",
                    colClasses = col_types
  ) %>%
    dplyr::rename_with(  # clean the columns names
      clean_names
```

```
    ) %>%
    dplyr::mutate_if(function(x) {  # Convert specified column to dates
      "POSIXt" %in% class(x)
    }, ~ format(.x, "%d/%m/%Y"))

  # Filter out rows without IDHAL
  if (filter_id) {
    idhal <- table$idhal
    table <- table[which(!is.na(idhal)),]
    table$idhal <- as.numeric(idhal)
    table$idhal[table$idhal == 0] <- NA
  }

  # Test if there is a column named "id"
  # Possibility to add mandatory columns in the future
  # Just add "OR"
  if ( !("idhal" %in% names(table)) | !("nom" %in% names(table)) | !("prenom" %in% names(tabl

  return(table)
}
```

Enfin, *load_team_table* appelle l'une des deux fonctions en fonction du type d'adresse fourni : URL ou chemin de fichier local. Le type de l'adresse est déterminé en utilisant *grepl*, qui utilise une expression régulière.

```
load_team_table <- function(filter_id = TRUE,
                            date_cols = c(7,8),
                            filepath_or_url) {

  # If the parameter is a URL
  if (grepl("^https?://", filepath_or_url)) {

    # Call the load_team_table_url function with the URL
    return(load_team_table_url(filter_id = filter_id, date_cols = date_cols, url = filepath_
  } else {

    # Otherwise, it's a local file path; call the load_team_table_csv function with the local
    return(load_team_table_csv(filter_id = filter_id, date_cols = date_cols, filepath = filep
  }
}
```

1.4.1.2 Résultat de l'Application de *load_team_table*

```
data_RH_csv_Example <- load_team_table(filter_id = TRUE, date_cols = c(6,7), filepath_or_url
head(data_RH_csv_Example)
```

```
  x_u_feff_civilite       nom   prenom   statut  rattachement debut_contrat
1                M. ADJAKOSSA     Éric      CEC AgroParisTech
2               Mme    ALBERT Isabelle       CR         INRAE
3                M. ALLYNDREE   Joseph Doctorat AgroParisTech
4               Mme   ALVAREZ Isabelle     IPEF         INRAE
5                M.    ANAKOK     Emré Doctorat         INRAE    06/10/2021
6               Mme    AUBERT    Julie       IR         INRAE
  fin_contrat        financement  equipe  unite               orcid  idhal
1                                SOLsTIS MIA PS 0000-0002-5280-0347 749339
2                                SOLsTIS MIA PS 0000-0003-2686-917X     NA
3                                                                       NA
4                                EkINocs MIA PS 0000-0002-5268-8666     NA
5  05/10/2024 ANR EcoNet/Mathnum SOLsTIS MIA PS 0000-0002-4439-4684     NA
6                                SOLsTIS MIA PS 0000-0001-5203-5748  15356
                    adresse_mail
1 eric-houngla.adjakossa@inrae.fr
2        Isabelle.Albert@inrae.fr
3
4       isabelle.alvarez@inrae.fr
5            emre.anakok@inrae.fr
6           julie.aubert@inrae.fr
```

Lors de l'application de la fonction *load_team_table* ci-dessus en utilisant le jeu de données test Data_RH_Example, on observe le chargement d'un csv contenant les colonnes d'origines du csv fournie en entrée. Les valeurs manquantes ou mal renseignées des idhals numériques ont été remplacées par des NAs. Les dates présentes deans les colonnes de dates spécifiés par l'utilisateur ont été nettoyées et ont le bon format.

1.4.1.3 Modifications des fonctions régissant la requète HAL

Dans le package managHAL originel, il y avait deux fichiers contenant les fonctions nécéssaires à la construction d'une requète HAL. Ces deux fichiers se nomment **HAL_reports.R** et **HAL_queries.R**. La fonction principale de construction d'une requète HAL est *HAL_query*.

Parler des modification faites a HAL_query (add_outputs), pourquoi ? (je souhaite obtenir des informations bien particulières pour pouvoir coder le reseau de co-author et construire un sbm sur le reseau ?

1.4.2 Conclusion des Résultats

Exemple via chatGPT

Les méthodes développées au cours de ce stage ont permis d'améliorer significativement les performances des algorithmes existants. L'optimisation de l'algorithme X a réduit le temps de traitement de 40%, tandis que les ajustements apportés au modèle Y ont augmenté la précision de 5%. Ces résultats démontrent l'efficacité des approches choisies pour répondre aux limitations des méthodes précédentes.

1.5 DISCUSSION

parler de l'efficacité de ce que j'ai code, des modifications à venir et de ce que je souhaite faire avant la fin de stage. Parler de l'api HAL et du cauchemar que c'est, parler des potentiels utilisations du package pour HAL mais aussi faire un parallèle avec la bio avec par exemple en ouverture les réseaux de gènes ?

1.6 ANNEXES

Exemple d'une documentation classique avec Roxygen2 :

```
#' load_team_table_csv
#'
#' This function loads a team table from a CSV file, allowing optional filtering
#' based on the presence of an IDHAL an specifying which columns contain dates.
#'
#' @param filter_id erase person without IDHAL (default = TRUE)
#' @param date_cols positions of dates columns, others will be read as text
#' @param filepath the file path of the CSV table
#'
#' @export
#'
#' @importFrom magrittr %>%
#' @importFrom utils read.csv
#'
#' @return a data.frame containing information from team
```

```
#'
#' @examples
#'
#' \dontrun{
#' load_team_table_csv(filter_id = TRUE,
#'                     date_cols = c(7,8),
#'                     "C:/users/.../.../Classeur.csv")
#' }
```

1.6.1 Tableau des résultats détaillés

Voici les tableaux des résultats détaillés pour les différentes méthodes et tests effectués.

1.7 BIBLIOGRAPHIE

Remerciements

AgroParisTech. 2024. “Science Ouverte.” https://www.agroparistech.fr/recherche/science-ouverte.

Aubert, J., P. Barbillon, S. Donnet, and V. Miele. 2022. *Using Latent Block Models to Detect Structure in Ecological Networks*. Wiley. https://doi.org/10.1002/9781119902799.ch6.

Comité d’orientation du CCSD & Assemblée des partenaires HAL, Comité de pilotage et. 2022. “HAL, Archive Ouverte Construite En Commun Pour Partager Et Diffuser La Connaissance Scientifique.” https://www.ccsd.cnrs.fr/wp-content/uploads/2023/01/DeclarationPolitiqueFR_VF.pdf.

LJonchere, C. et al. n.d. “ExtrHAL - HALUR.” https://halur1.univ-rennes1.fr/ExtrHAL.php.