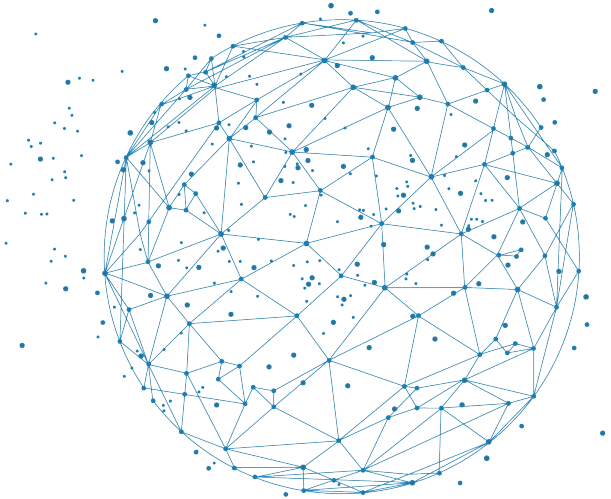
Synthèse des hackings



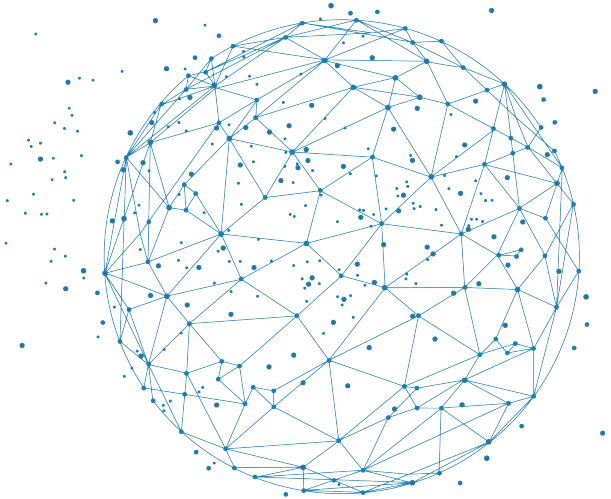
Hackathon "IA pour la biologie-santé", Villejuif du 1 au 3 juin 2026

Liste des sujets proposés

1. Microbiome metadata : compléter les métadonnées de jeux de données publics avec les articles scientifiques
2. EDAM terms recommender
3. Utilisation d'une IA pour faciliter et améliorer l'annotation des concepts d'une ontologie
4. AI Assistant for tool selection in Biosphere
5. Spatial RAG for earth virome exploration
6. Benchmarking : comparaison de solutions pour implémentation d'un workflow nextflow (Seqera IA, Albert, GitHub Copilot, etc.)
7. FAIR-Checker MCP
8. Utilisation d'une IA pour sécuriser un service bioinformatique exposé sur le web

8. Microbiome Metadata

Hélène Chiapello, Nicolas Pons



Description synthétique

Contexte

- Réutilisation de jeux de données publics de microbiomes dans le cadre de méta-analyses

Problématique

- Métadonnées incomplètes, erronées & hétérogènes dans les jeux de données publics des microbiomes
- Checklists ENA utiles mais mal utilisées pour décrire précisément les caractéristiques échantillons
- Les métadonnées sont dispersées dans différentes ressources (ENA, github, articles scientifiques, bases de données spécialisées,...)

Objectifs du hacking (et pour quelques années)

- Utiliser les articles complets en XML de EPMC (EBI), pour identifier les sources de métadonnées pertinentes
- Récupérer les métadonnées
- Automatiser la standardisation les métadonnées à l'aide d'un portail dédié (Ontology Lookup Service)

Approches et moyens envisagés pour le hacking

- On a besoin d'aide pour clarifier le use case et le rendre faisable
- Par contre on a un jeu de données pour évaluer les résultats vs une 'curation' manuelle : https://entrepot.recherche.data.gouv.fr/dataset.xhtml?persistentId=doi:10.57745/7IVO3E

Notes hackathon :
https://docs.numerique.gouv.fr/docs/d750fb4a-54bc-4583-a331-edb653e05de1/

Conclusion du UC microbiome metadata

Réalisations

- Formalisation du protocole (voir notes dans docs)
- Tests de plusieurs Bioprojects du dataset CRC (Nicolas)
- Tests de plusieurs IA
- Premiers résultats sur lesquels évaluer la qualité
  - Perplexity education / Mistral : A partir d'un article et d'un bioproject, génération d'un tableau des métadonnées des biosamples d'un bioproject en mixant les métadonnées soumises (ENA) et les métadonnées trouvées dans le SM de l'article
  - API Albert : RAG construit à partir de l'url du pdf d'un bioproject ; interrogation du RAG et formatage dans un tableau de la liste des datasets publiés et/ou utilisés dans la publication

Difficultés rencontrées

- Ecriture du prompt
- Selon les bioprojects et les articles les métadonnées ne sont pas toutes accessibles de la même manière → généricité du prompt ?
- Choix du modèle adapté ?

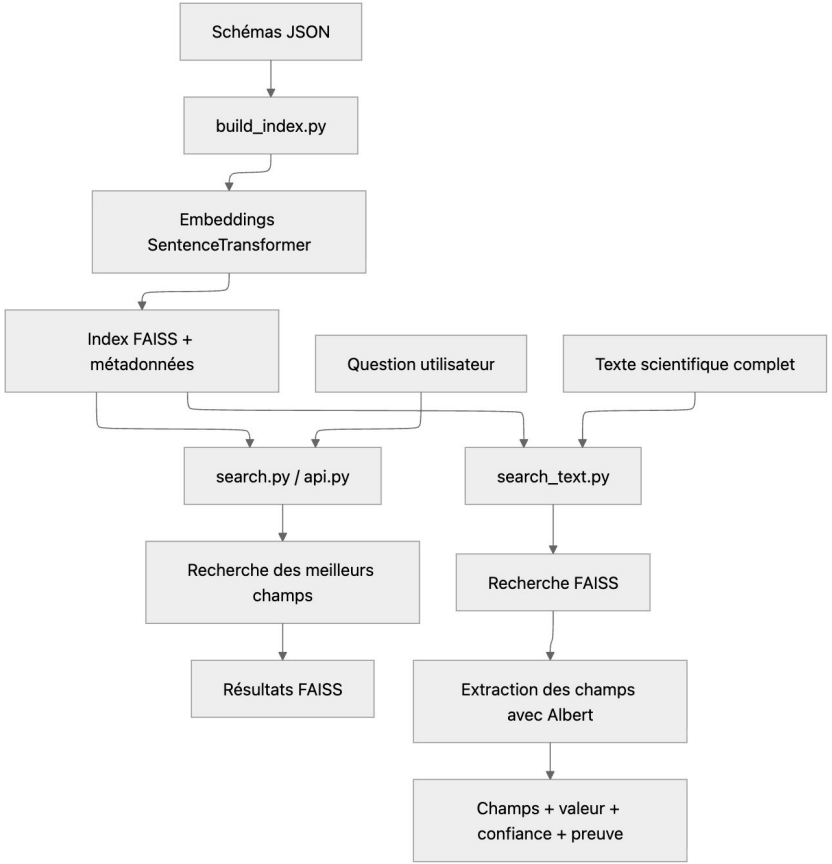
Conclusion

- Faisabilité démontrée du UC
- Généralisation pas forcément triviale (cas où ce sera impossible, attention aux hallucinations)
- Lien avec plusieurs travaux en cours
  - Alban / BIRD : score de complétude des métadonnées d'une Checklist d'un Biosample avec FAIRChecker
  - Nicolas / MGP : Base de données MIASSM Cloud4Sams
  - Liliana / MNHN : Référentiel des métadonnées de Biodiversité et d"identification des espèces à partir de séquences génétiques
  - Hélène / INRAE : projet MicrobiomeSchemas de schemas de métadonnées partagés pour les données de microbiome
  - Thomas & Imane / IFBcore : aide à la sélection de métadonnées dans madbot

Lien du dépot

- https://docs.numerique.gouv.fr/docs/d750fb4a-54bc-4583-a331-edb653e05de1/
- https://forge.inrae.fr/nicolas.pons/microbiome_metadata_with_albert

Aide à la section de métadonnées dans madbot



Ce que nous avons fait

- Embedding
- Construction d'un index
- Exploration de l'index
- Affinage de la réponse via LLM (albert)
- Création d'un endpoint API (fastAPI)

Ce que nous avons testé

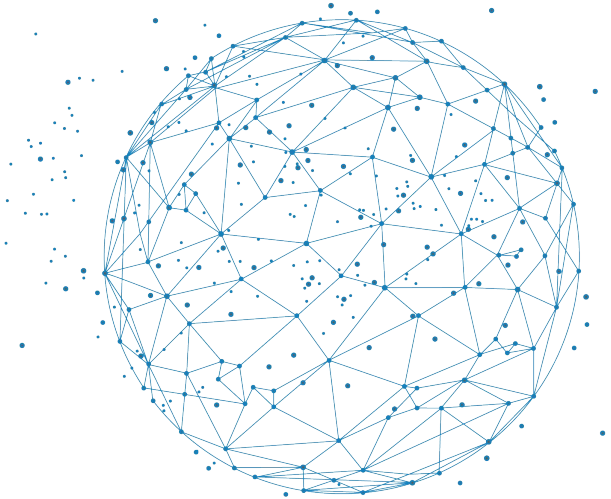
- Question simple : "comment je donne l'état de santé de mon patient ?"
- Demande d'extraction des métadonnées depuis un abstract et passage dans albert pour préparation de la réponse

Ce que nous aimerions tester maintenant

- Mise en place d'agent associé à des ontologies et des taxonomies

EDAM-terms-reco

Alban Gaignard



Description synthétique

Contexte

- EDAM est une ontologie utilisée pour annoter les ressources bioinformatiques
- Peu d'experts Bioinfo connaissent l'ontologie EDAM
- Processus d'annotation très chronophage

Problématique

- Comment proposer des annotations EDAM pertinentes

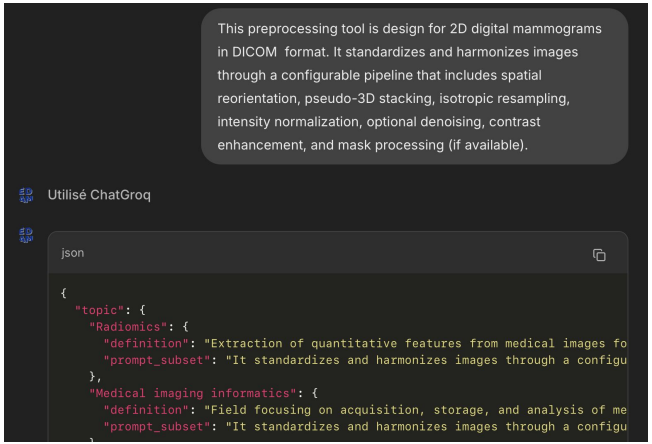
Objectifs du hacking

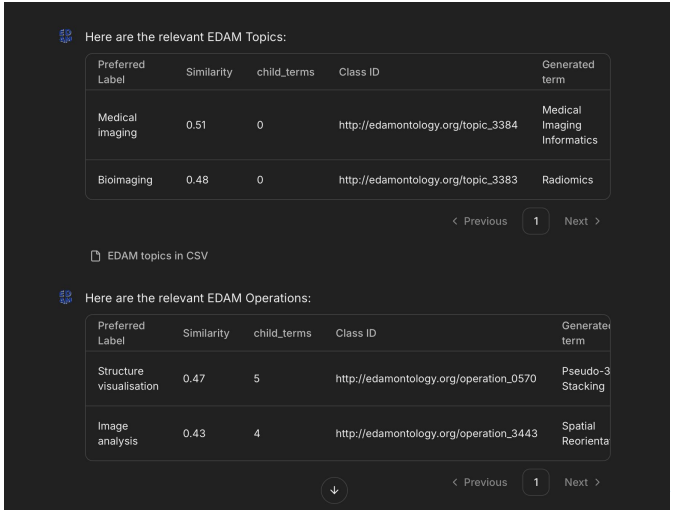
- Réutiliser une infra LLM souveraine (Albert ? Glicid ?
- Évaluer la prédiction de termes EDAM sur une ground truth / comparer avec des outils NLP ?
- Proposer une interface MCP (EDAM-MCP)

Approches et moyens envisagés pour le hacking

- Ontologie EDAM
- RAG (FAISS, ChromaDB)
- Chainlit pour l'interface

https://docs.numerique.gouv.fr/docs/6b054483-f3b9-488b-8e8d-f647dcf1724a/





Conclusion

Réalisations

- Un adaptateur pour facilement basculer entre
  - LLM local (biomistral)
  - API Albert (LLM académique)
  - API Groq (LLM commercial)
- Un notebook pour évaluer les prédictions
- Tests “à la main” de différentes stratégies d’annotation
  - 2 exemples: Kraken2, Deseq2
  - Edam-terms-reco VS Edam-map ?
  - Bio.tools VS bioconductor VS Wikipedia ?
  -

Difficultées rencontrées

- Code généré par IA parfois trop verbeux et inefficace → difficile à refactorer à la main
- Difficulté d’évaluer la pertinence des annotations EDAM produites (précision / rappel / F1-score)
  - Besoin de prendre en compte la hiérarchie des classes EDAM
- Besoin de plus de temps … :-)

Conclusion

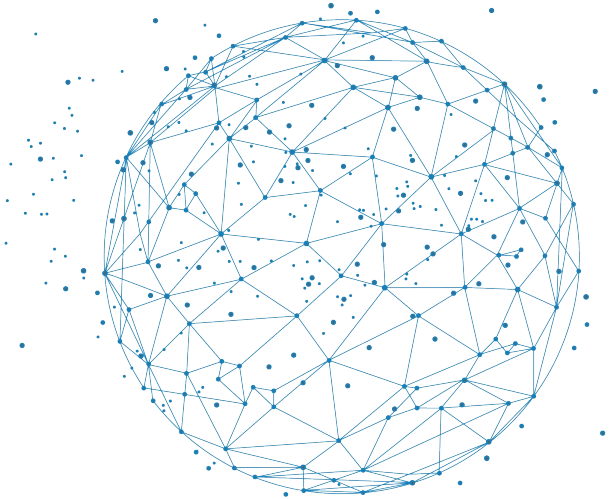
- La GT représente les connaissances des experts
- Pas de document textuel de référence pour l’annotation
- La page Wikipedia de DESEQ2 permet d’obtenir de meilleurs résultats VS descriptions bio.tools & bioconductor (plus courtes) VS cookbook (trop long)

Lien du dépot

- https://github.com/albangaignard/edam-terms-reco

Utilisation d'une IA pour faciliter et améliorer les définitions de concepts dans l'ontologie EDAM

Jacques van Helden



Utilisation d'une IA pour faciliter et améliorer les définitions de concepts dans l'ontologie EDAM

Contexte

- Les ontologies sont censées dresser un inventaire exhaustif des concepts d'un domaine de recherche.
  - Termes (noms, synonymes)
  - Définition
  - Liens entre concepts
  - Liens avec d'autres ontologies
- Elles assurent la cohérence de la terminologie utilisée dans différents contextes (annotations biologiques, outils logiciels)
- L'ontologie EDAM, qui recense les concepts centraux de la bioinformatiques : types et formats de données biologiques, opérations, thématiques ("topics"), est utilisée par un nombre croissant d'applications externes.

Problématique

- L'annotation des concepts d'EDAM requiert un travail considérable, et repose sur une très petite équipe.

Objectifs du hacking

- Evaluer l'apport d'une IA générative pour

Approches et moyens envisagés pour le hacking

- Charger l'IA avec la version actuelle de EDAM (format owl)
- S'intéresser à un domaine précis (ex: régulation transcriptionnelle)
- Demander à l'IA d'extraire tous les termes liés à ce domaine
- Sélectionner un ensemble raisonnable de ces termes, les plus représentatifs (ex: TF, TFBM, TFBS, découverte de motifs, recherche de motifs)
- Demander à l'IA de suivre la procédure suivante
  - Collecter les informations pertinentes auprès de différentes sources considérées comme références
  - Evaluer le pour et le contre de chaque définition
  - Proposer une définition qui capture le meilleur
- Expertise humaine : demander à un groupe d'experts d'évaluer les propositions et de proposer d'éventuelles améliorations

Conclusion

Réalisations

- Tentatives infructueuses d'utiliser l'API d'Albert
- Tentative avec Ollama + mistral peu encourageante
- Analyse ChatGPT pertinente mais incomplète:
  - Dialogue pré-analyse pour préciser les attentes et les exigences (no hallucination, no modification of the EDAM-extracted columns…)
  - Tableau de résultats contient essentiellement des concepts pertinents mais il manque des termes essentiels

Difficultées rencontrées

- Albert
  - ne digère pas un tableau de 3.400 lignes (EDAM) dans le prompt
  - Tentative infructueuse de lui faire générer un RAG: tableau de données trop gros
  - Réduction au minimum (3 colonnes, 300 lignes) mais autre erreur
- Mistral
  - me propose de sélectionner moi-même les concepts pertinents ⟶ pas poussé plus loin

Conclusion

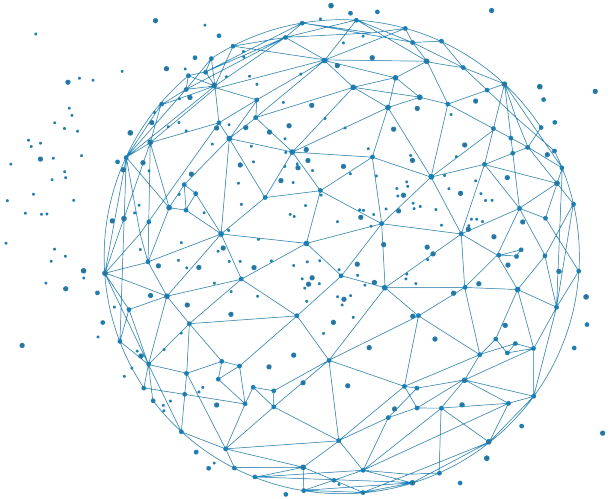
- ChatGPT trop fort
- Révisions d'EDAM : comment soumettre les résultats (édition via Protege, PR github, issues github…) ?
- A creuser: possibilité de mener cette analyse sur des IA souveraines
  - Albert (arriver à faire un RAG)
  - CNRS (mistral) ?
  - INRAE (chatGPT) ?
  - NNCR IFB ?
  - Jean Zay ?
- Reste à faire : utilisation pour améliorer les défiintions

Lien du dépôt

- Dossier partagé: 03_EDAM-definitions
- Notes: notes_hacking-03_EDAM-definitions.docx
- Tableau de résultats
  - ChatGPT_EDAM_transcriptional_regulation_terms.xlsm
  - Propositions de ChatGPT
  - Annotation des propositions (Jacques van Helden)
  - Onglet ajouté : termes manquants

3. AI Assistant for tool selection in Biosphere

Christophe BLANCHET, Matis ZOUARI



Description synthétique

Contexte

- Biosphere : Bioinformatics cloud services to analyze life science data

Problématique

- Users with limited knowledge about existing tools miss out on services provided by existing Biosphere appliances

User stories

- “As a scientist working on human gut metagenomics, I would like to identify tools/biosphere-apps to analyse my data.”

Objectifs du hacking

- Creating a chatbot to redirect users towards appliances with the right tools for their analysis

Approches et moyens envisagés pour le hacking

- Albert API
- Agentic AI
- Query to the RAINBio and Bio.Tools catalogues (MCP?)
- Use of EDAM (and its MCP)

Conclusion

Définition des questions modèles

- “As a scientist working on human gut metagenomics, I would like to identify tools/biosphere-apps to analyse my data.”
- “I am proficient with R and would like to find an appliance to analyse my metagenomic data.”

Réalisations

- Interaction entre Albert API (python) et une version dev du catalogue RainBIO (fichiers .yaml local).
  - Modèle : ***mistralai/Mistral-Small-3.2-24B-Instruct-2506***
- Utilisation du dépôt git RSEc pour la récupération des biotools et topics associés.
- Etablissement de plusieurs requêtes successives afin d’améliorer la pertinence de la réponse (CLI).

Participants

- Matis ZOUARI, Audrey BIHOUEE, Christophe BLANCHET, Hervé MENAGER

Difficultés rencontrées

- Base de données réduite -> évaluation de la qualité de la réponse moins évidente
- Difficultés du prompting -> a pu provoquer des bugs de formatage (.json), et de manière générale ça a été le plus gros travail pour améliorer la qualité de la réponse du modèle
- Hallucinations possibles, potentiellement liées à la taille et au format de la base de données (passage sur un graphe potentiellement intéressant)

Conclusion

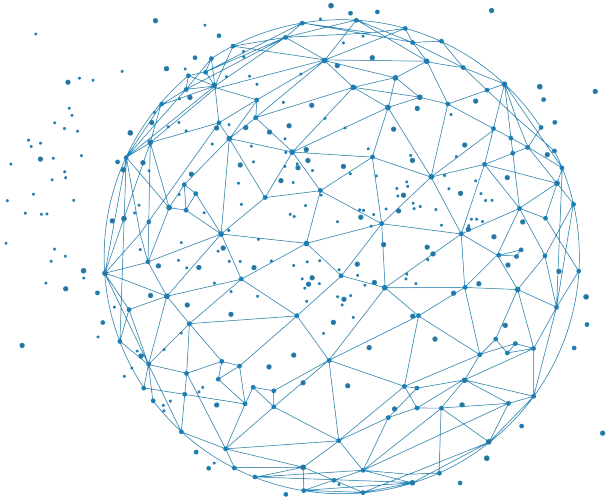
- Ca marche plutôt bien :)
- Beaucoup de potentiel pour les usagers de Biosphère
- Un bon cas d’usage pour RSEc qui donne du poids au projet

Lien du dépôt

- https://gitlab.in2p3.fr/ifb-biosphere/ifb-hack-ia/biosphere-chat

4. FAIR-Checker MCP

Alban Gaignard



Description synthétique

Contexte

- FAIR-checker is a tool to evaluate FAIRness from semantic metadata (schema.org, DCTerms, PROV, etc.)
- An API returns evaluation scores with textual recommendations

Problématique

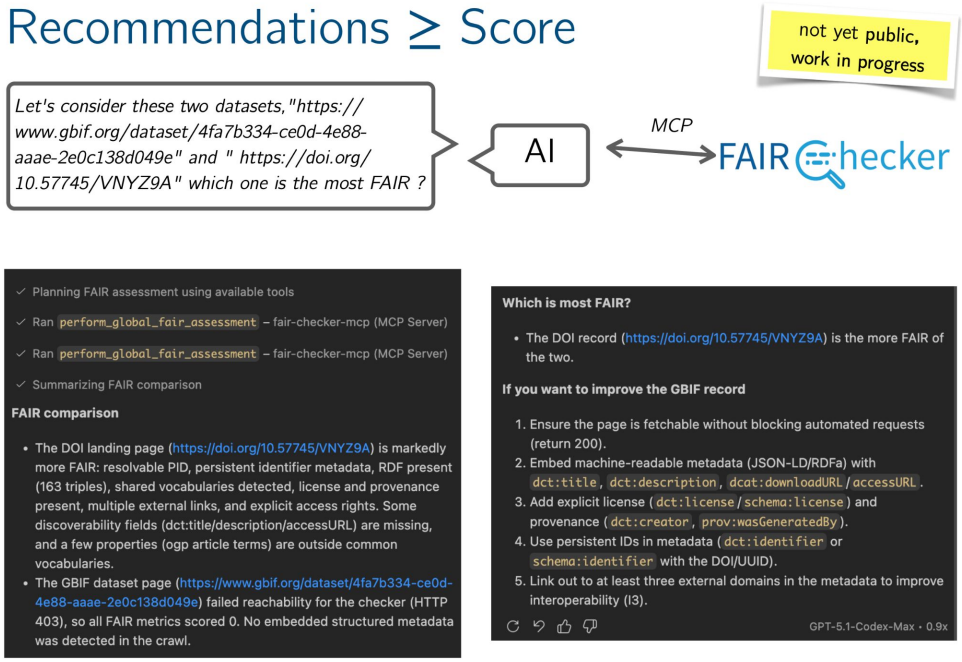
- Can we interact with FAIRChecker in natural language ?
- Can an LLM agent propose relevant and valid metadata based on the recommendations ?

Objectifs du hacking

- Develop an MCP endpoint for FAIR-Checker
  - FAIR assessment
  - Proposal of missing metadata
- Interact with the MCP endpoint through Albert

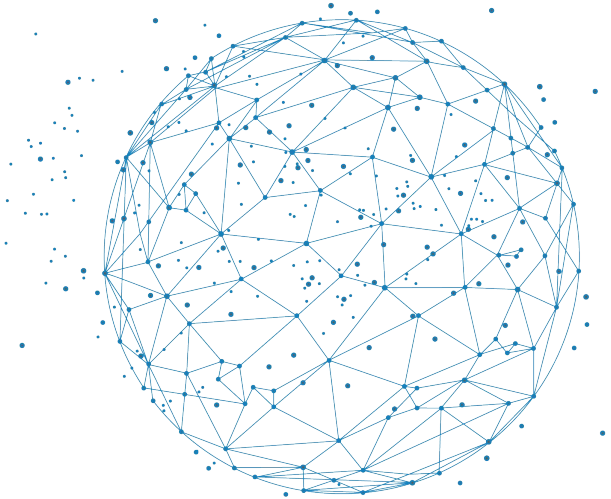
Approches et moyens envisagés pour le hacking

- FAIR-Checker
- FAIR-Checker-MCP
- Albert API



5. Spatial RAG for earth virome exploration

Paul TISSOT



Description synthétique

Contexte

- Virome@tlas project : large scale description and exploration of earth virome
- Curated dataset with localisation and viral information

Problématique

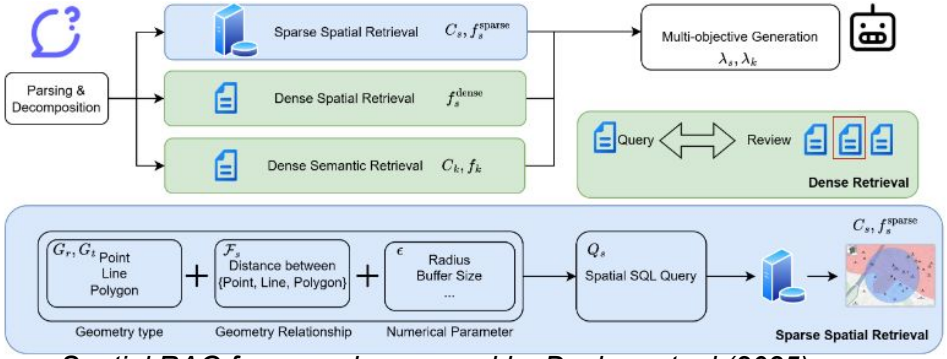
- Dataset spatial query with classical GIS query languages (different of human language, experts oriented)
- Spatial relationships are difficult to represent and query

Objectifs du hacking

- Implement a Spatial RAG architecture on virome@tlas dataset

Approches et moyens envisagés pour le hacking

- (Spatial)RAG implementation (with Albert API)
- Neo4J implementation (Knowledge graph approach)
- Streamlit interface



Spatial RAG framework proposed by Dazhou et. al (2025)

Conclusion

Réalisations

- Création d'un agent avec Albert interrogeant un fichier tabulaire de métadonnées BioSample (6M lignes, ~65 colonnes)
- **1er outil** (Paul) : donne une description textuelle des métadonnées à partir d'un ID BioSample

  Ex: The BioSample SAMN13153374 is an environmental metagenome from soil in California, United States, sequenced using Illumina platform with Whole Genome Sequencing (WGS) assay type and METAGENOMIC library source. The sample is located on land, 0.48 km from the nearest river and 25.24 km from the coastline. No host information is provided, which is consistent with the environmental origin of the sample. The ENVO ontology labels the sample as "saprolite" with medium confidence. The sample has been sequenced, and the sequencing method is well-documented. There are no apparent incompatible values in the metadata.
- **2ème outil** (Pauline) : donne un résumé des descriptions pour un pays donné

  Ex: Based on the curated dataset, the biosample metadata from France primarily consists of samples collected from various environments, including soil, water, and host-associated sources. The most common organisms identified in these samples are humans (Homo sapiens), followed by various species of bacteria such as Escherichia coli and Staphylococcus aureus. Additionally, there are samples from plants like Arabidopsis thaliana and animals like Mus musculus, reflecting the diverse range of biological research conducted in France.
- **3ème outil** (Romuald) : Rechercher les virus présents à "proximité" d'une ville

  Ex: Retourne un tableau des N plus proche échantillons d'une ville (calcul distance haversine) si l'user le demande.

Difficultés rencontrées

- Pas possible de faire un RAG en donnant un fichier .csv
- Prise en main de l'API Albert
- Compréhension du prompting
- Compréhension du mode agentique
- Quelques erreurs côté Albert

Conclusion

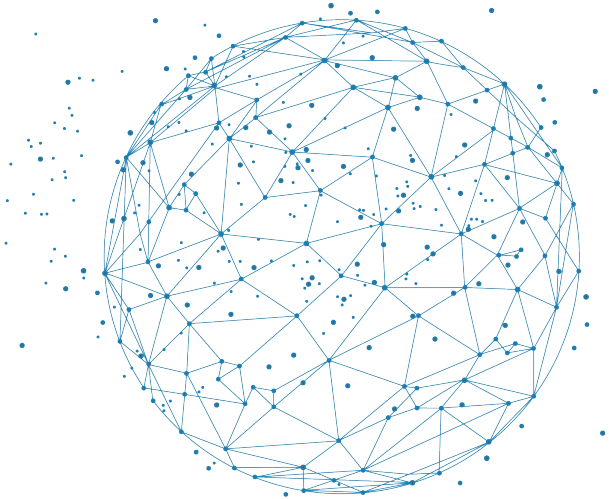
- Réalisation d'un objectif secondaire (Fiche résumé d'un BioSample à partir des métadonnées "curées" par le projet Virome@tlas)
- Interrogation Spatiale complexe :
  - Nécessité d'un dataset avec relation spatiales clairement définies entre samples et environnement
  - Création d'un agent (ou agents multiples) capable de décider quelles opérations spatiales il doit réaliser selon la demande utilisateur (agent spatial + agent sémantique + fusion réponses)

Lien du dépôt

- https://gitlab.in2p3.fr/ifb-biosphere/ifb-hack-ia/viromeatlas-rag

6. Benchmarking: a comparison of solutions for implementing a Nextflow workflow (Seqera AI, Albert, GitHub Copilot, etc.)

Philippe Hupé, Frédéric Jarlier, Nicolas Servant, Corentin Raoux, Baptiste Roelens, Fabrice Leclerc, Quentin Duvert



Description synthétique

Contexte

- Nextflow is a workflow managers use to create bioinformatics pipelines
- The Nextflow language is evolving rapidly, and pipeline development is time-consuming for an analyst

Problématique

- How to develop and keep up to date with the NextFlow language?

Objectifs du hacking

- Enable a bioinformatician to quickly create a NextFlow pipeline by specifying the tools
- Enable a computer scientist to create a pipeline without prior knowledge of the tools
- Update an existing pipeline to keep pace with NextFlow's developments

Approches et moyens envisagés pour le hacking

- Use of open-source AI (Albert)
- Use of a specific AI (SeqeraIA)
- Comparison of results and user time

Conclusion

Réalisations

- Configuration env VS Code + continue
- Création d'un prompt pour le benchmark
- Test du prompt avec différentes modèles d'albert
- Test du prompt avec Seqera IA
- Test du prompt avec Claude (latest)
- Test du prompt avec Gemini 3.1

Difficultée rencontrée

- Mise en place assez longue pour paramétrer continue avec Albert
- Relecture du code parfois aléatoire en fonction du résultat
- Limite de token à l'instant T ce qui fait planter l'exécution

Conclusion

- Plus le modèle est gros et récent meilleur est le résultat
- Faire un plan avec l'IA permet d'avoir un meilleur résultat
- Modèle commerciaux plus à jour => meilleur résultat

Lien du dépot

- Aucun pas de pipeline fonctionnel
- https://docs.numerique.gouv.fr/docs/4bab9f58-f28f-49f3-96b3-8293130aa74b/

I would like to implement a single-cell RNA-sequencing bioinformatics pipeline using nextflow. It should take as input data : the output directory of the command line `cellranger count`

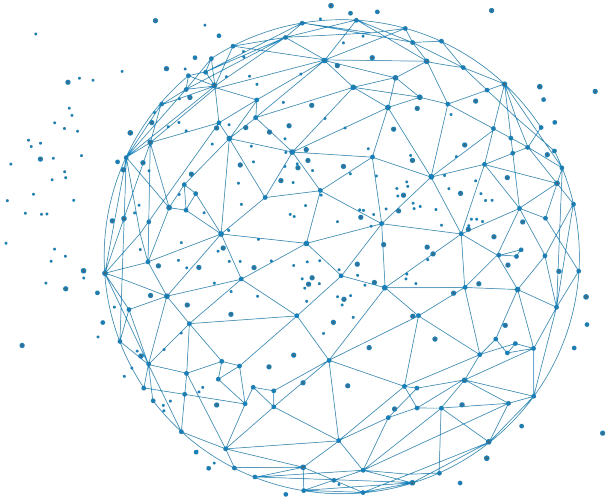
The pipeline will carry out the following steps:

- Load the data from the cellranger output directory
- Apply the first state-of-the-art quality control steps
- Detect empty droplets
- Remove the Ambient RNA contamination
- Apply a thresholding cut-off on different variables (Number of counts, minimal number of cells expressing a specific genes, maximal ratio of mitochondrial and ribosomal genes)
- Detect and remove the doublets
- Normalize and scale the data
- clusterize the data
- Apply reduction dimension methods as PCA and UMAP
- Perform automatic single-cell annotation
- Generate a final user-friendly HTML report reporting all quality controls, versions and command lines

The pipeline:

- will be launched on a computing cluster with the SLURM scheduler and apptainer. Each tool must be available within a sif container.
- should include a dry-run test with a `stub` section within each process.
- should include a test profile with a toy data set (10X format)
- whenever necessary, use the python language
- the user must be able to set the value of the most essential parameters from the nextflow command line
- should include the documentation to install and user it

7. Sécurisation de l'interface web d'un portail bioinformatique



Jacques van Helden

Description synthétique

Contexte

- Augmentation de la cybercriminalité, du hacking, et de la puissance de nuisance
- Certaines ressources bioinformatiques ont été conçues il y a longtemps, sur base de technologies pas forcément idéales pour résister au hacking
- Nécessité de renforcer la sécurité de tous les services déployés sur le Web

Problématique

- Regulatory Sequence Analysis Tools (http://rsat.eu/), suite logicielle déployée depuis 1998 sans interruption.
- 6 serveurs sur 3 sites : Mexique (bactéries), Espagne (Plantes) IFB (Metazoa, Protists, Fungi, Teaching)
- IFB et Mexique hackés en octobre 2025 → interruption des services. Seule l'Espagne résiste encore
- Travail entrepris par les équipes Mexicaine et Espagnole pour sécuriser et relancer les services

Objectifs du hacking

- Utiliser une IA pour renforcer la sécurité de l'interface Web de RSAT

Approches et moyens envisagés pour le hacking

- Charger une IA avec
  - Le code RSAT
  - Le manuel d'installation
- Demander de détecter les points de faiblesse
- Proposer un scénario de sécurisation du code actuel
  - Actions à entreprendre, avec évaluation des coûts / bénéfices
  - Priorisation des actions
- Evaluer les pistes pour une réimplémentation moderne des interfaces web

Conclusion

Réalisations

- Pas pu démarrer

Difficultées rencontrées

- C.B. n'a pas accepté de travailler au restaurant, ni dans le métro ligne 7 (malgré la “pression" ;-).

Conclusion

- Reste à faire
- …

Lien du dépôt

- Dépôt **privé** sur github, réservé à l'équipe de déploiement RSAT (Mexique + Espagne) pour éviter d'exposer publiquement les failles qu'on détecte et les stratégies de résolution.