# ***PARTIE 1 : Étude théorique de l'outil Biopython***

## 1. Présentation générale de l'outil

Biopython est l'un des projets les plus emblématiques du mouvement "Bio projects" (incluant BioPerl, BioJava, etc.). Il s'agit d'une collection d'outils Python non commerciaux, développée par une communauté internationale, visant à répondre aux besoins de la bioinformatique et de la biologie computationnelle. Lancé à la fin des années 90, il repose sur le langage Python, réputé pour sa lisibilité et sa puissance. Son but est de fournir des bibliothèques réutilisables pour traiter les données biologiques complexes.

## 2. Fonctionnalités principales

Biopython offre une vaste gamme de fonctionnalités pour manipuler les données biologiques :

- **Manipulation avancée des séquences biologiques :**

  Biopython propose un système sophistiqué de manipulation des séquences à travers les classes Seq et SeqRecord. Contrairement aux simples chaînes de caractères, ces objets intègrent nativement la sémantique biologique :

  **Objets Seq** : Représentent des séquences biologiques (ADN, ARN, protéines) avec des méthodes spécialisées pour le transcription inverse (reverse_complement()), la traduction (translate()), et la recherche de motifs. La gestion automatique des alphabets biologiques évite les erreurs courantes comme la transcription d'une séquence protéique.

  **Objets SeqRecord** : Structure complexe associant une séquence à des métadonnées riches : identifiants, descriptions, annotations, caractéristiques (features), et références. Cette structure est essentielle pour maintenir l'intégrité des données tout au long des pipelines d'analyse.

  **Système d'annotations :** Support complet des formats d'annotation standards (GenBank, EMBL, Swiss-Prot) avec possibilité d'extraire des régions codantes, des sites de restriction, des domaines protéiques, etc.

- **Accès programmatique aux bases de données biologiques:**

  Biopython intègre des interfaces élaborées pour interroger les principales bases de données biologiques :

  **Module Entrez** : Interface avec le système E-utilities du NCBI permettant des requêtes complexes sur PubMed, GenBank, GEO, et autres bases. Fonctionnalités avancées incluant la gestion des limites de requêtes, le parsing automatique des résultats, et le téléchargement par lots.

  **Interface avec UniProt** : Accès programmatique à la base de données de protéines via l'API REST, avec support des formats UniProtKB, XML, et tabulés.

**Accès au Protein Data Bank (PDB) :** Téléchargement et parsing des fichiers de structures 3D avec reconstruction des modèles atomiques et extraction des informations cristallographiques.

- **Lecture/Écriture de fichiers :** Support de formats standards tels que FASTA, GenBank, BLAST, ClustalW et Phylip via les modules Bio.SeqIO et Bio.AlignIO.
- **Bioinformatique structurale :**

  **Module PDB** : Lecture, écriture et manipulation des fichiers PDB avec reconstruction des chaînes, résidus, et atomes. Calculs géométriques avancés : distances interatomiques, angles dièdres, rayons de giration.
  **Analyse des surfaces** : Calcul des surfaces accessibles au solvant (SASA), détection des poches hydrophobes, et analyse des interfaces protéine-protéine.
  **Visualisation** : Intégration avec PyMOL et NGL Viewer pour la visualisation interactive des structures.
- **Outils statistiques :** Intégration de modèles de Markov cachés (HMM) et de calculs de fréquences de codons.

### 3. Aspects techniques

Biopython est une bibliothèque **open-source** distribuée sous la licence "Biopython License Agreement". Écrit principalement en Python avec certaines parties en C pour optimiser les performances de calcul. L'**installation :** Se fait généralement via le gestionnaire de paquets pip (pip install biopython).

```
C:\Users\DELL>pip install biopython
WARNING: Ignoring invalid distribution ~ip (C:\Python312\Lib\site-packages)
WARNING: Ignoring invalid distribution ~ip (C:\Python312\Lib\site-packages)
Collecting biopython
  Downloading biopython-1.86-cp312-cp312-win_amd64.whl.metadata (13 kB)
Requirement already satisfied: numpy in c:\python312\lib\site-packages (from biopython) (1.26.4)
Downloading biopython-1.86-cp312-cp312-win_amd64.whl (2.7 MB)
   ---------------------------------------- 2.7/2.7 MB 6.3 MB/s  0:00:00
WARNING: Ignoring invalid distribution ~ip (C:\Python312\Lib\site-packages)
Installing collected packages: biopython
WARNING: Ignoring invalid distribution ~ip (C:\Python312\Lib\site-packages)
Successfully installed biopython-1.86

[notice] A new release of pip is available: 25.3 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\DELL>
```

Organisé en sous-modules indépendants, permettant à l'utilisateur de n'importer que les composants nécessaires à son analyse.

Biopython ↓
→Bio : Package principal
→SeqIO : Entrée/sortie des séquences
→Align : Alignements de séquences
→PDB : Structures 3D
→Phylo :Phylogénie
→PopGen : Génétique des populations

→Entrez : Accès NCBI
→ExPASy: Accès ExPASy/Swiss-Prot
→Cluster : Analyses de clustering
→Statistics : Statistiques bioinformatiques
→... 30+ modules supplémentaires

Chaque module est indépendant mais conçu pour une interopérabilité maximale, suivant les principes de cohésion forte et de couplage faible.

**4. Points forts**

- En tant que logiciel libre, il permet une recherche reproductible sans coût de licence.
- Il sert de "colle" entre différents outils bioinformatiques, permettant d'automatiser des flux de travail complexes.
- Bénéficie d'un "Tutorial and Cookbook" extrêmement riche et d'une communauté active.
- La syntaxe Python rend l'outil accessible aux biologistes n'ayant pas une formation poussée en informatique.

**5. Limites et points faibles**

- Pour des calculs de très haute intensité sur des données génomiques massives, Python peut être plus lent que des outils compilés en C++ ou Rust.
- Certains modules nécessitent l'installation préalable d'outils externes (comme BLAST ou ClustalW) sur le système.
- Bien que plus simple que d'autres langages, une base en programmation Python reste indispensable.

**Références**


- Cock, P. J., et al. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics, 25(11), 1422-1423.
- Documentation officielle : Biopython Project https://biopython.org/wiki/Documentation
- Chapman, B., & Chang, J. (2000). Biopython: Python tools for the life sciences. ACM SIGBIO Newsletter.
- Python Software Foundation. (2024). Python for Scientific Computing. Disponible sur : https://www.python.org/about/success/biopython/

## ***PARTIE 2 : Étude pratique - Exploration de Zenodo***

# 1. Présentation de Zenodo

Zenodo est une initiative stratégique née d'une collaboration entre le **CERN (Organisation européenne pour la recherche nucléaire)** et le projet **OpenAIRE (Open Access Infrastructure for Research in Europe)**. Lancé officiellement en 2013, Zenodo répond à un besoin critique identifié par la communauté scientifique européenne : la préservation à long terme et le partage ouvert de tous les produits de la recherche, au-delà des seules publications traditionnelles.

- **Objectifs** :

1. Garantie de conservation des données pour plusieurs décennies, avec des stratégies de réplication sur multiples sites géographiques.
2. Accès ouvert par défaut, avec options de restriction temporaire justifiées (embargos).
3. Conformité avec les standards internationaux (OAI-PMH, REST API, Schema.org).
4. Attribution de DOI (Digital Object Identifier) pérennes permettant la traçabilité des citations.
5. Connexion avec ORCID, GitHub, RID, et autres systèmes d'identité numérique des chercheurs.

- **Types de contenus hébergés**

Il héberge des jeux de données (datasets), des logiciels, des publications, des posters, des présentations et même des images ou vidéos.

| Catégorie | Exemples spécifiques | Formats acceptés | Taille maximale |
|---|---|---|---|
| **Publications** | Prépublications, articles, chapitres, thèses | PDF, DOCX, LaTeX | 50 GB/dépôt |
| **Données de recherche** | Jeux de données bruts, données dérivées | CSV, JSON, HDF5, NetCDF | 50 GB/dépôt |
| **Logiciels et code** | Scripts, packages, notebooks Jupyter | Python, R, Julia, Dockerfiles | 50 GB/dépôt |
| **Matériels pédagogiques** | Supports de cours, exercices, tutoriels | PPTX, Markdown, SCORM | 50 GB/dépôt |
| **Produits multimédia** | Vidéos, images, enregistrements audio | MP4, TIFF, WAV, PNG | 50 GB/dépôt |
| **Produits de vulgarisation** | Blogs scientifiques, infographies | HTML, SVG, PSD | 50 GB/dépôt |
| **Autres produits** | Posters, présentations, rapports techniques | PDF, PPTX, Keynote | 50 GB/dépôt |

- **Intérêt stratégique pour les sciences de la nature et de la vie**

**Pour la recherche fondamentale :**

1. Archivage des données brutes permettant la vérification indépendante des résultats.
2. Réduction du "file drawer problem" (biais de publication).
3. Support des standards spécifiques (Darwin Core pour la biodiversité, MIAME pour la transcriptomique).

**Pour la recherche appliquée :**

1. Réutilisation des données par d'autres équipes évitant la duplication d'efforts.
2. Agrégation de jeux de données similaires pour augmenter la puissance statistique.
3. Comparaison inter-laboratoires des résultats expérimentaux.

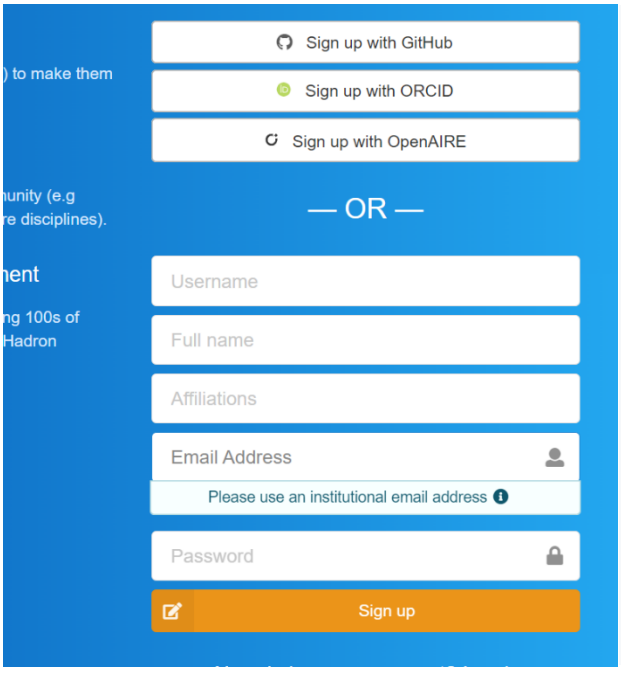
**Pour les étudiants :**

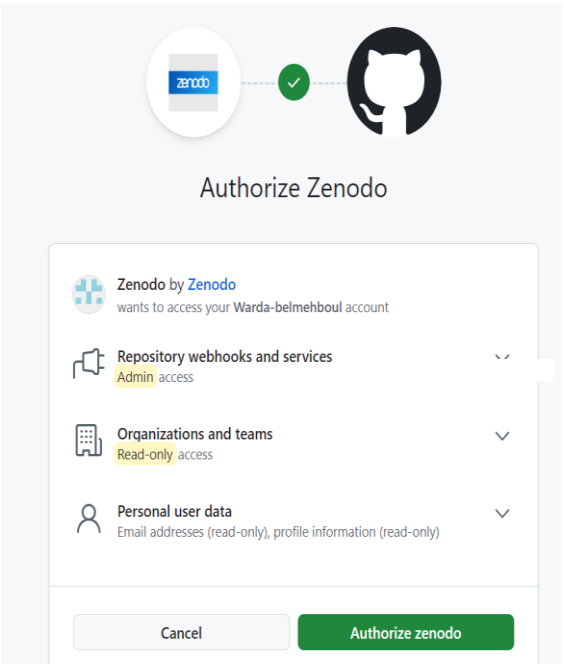
1. Accès à des données réelles pour les travaux pratiques.
2. Développement de compétences en gestion et partage des données (FAIR principles).
3. Possibilité de déposer des mémoires, thèses, et données de projets.

## 2. Description des étapes réalisées

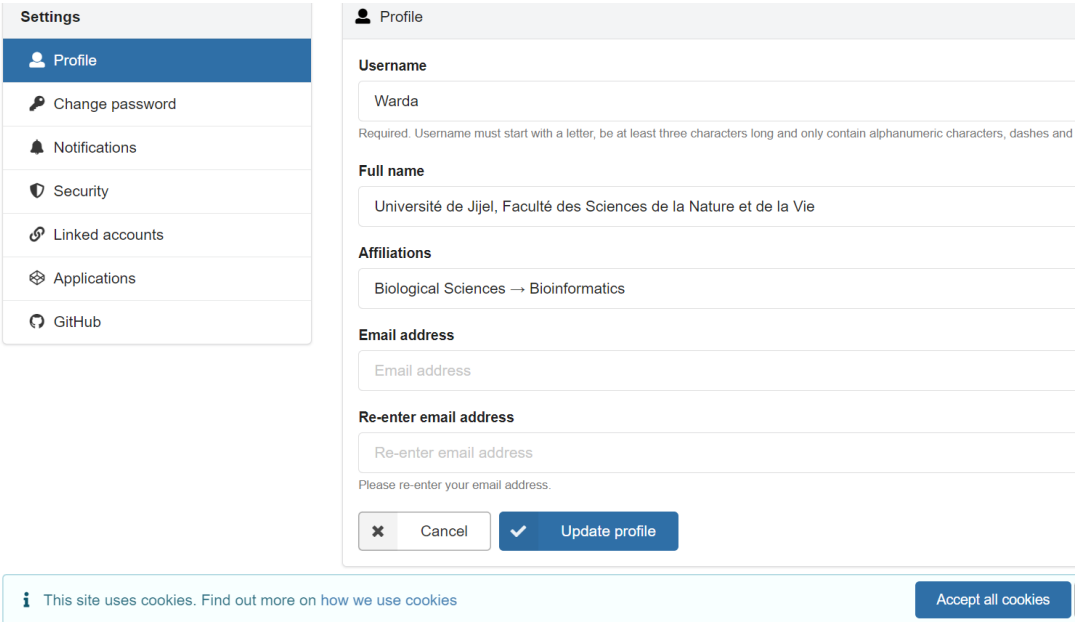
### Étape 1 : Visite et Navigation

Dans cette étape, nous avons tout d'abord fais une **Inscription** sur la plateforme https://zenodo.org/, avec la sélection GitHub pour l'intégration future avec les dépôts de code.

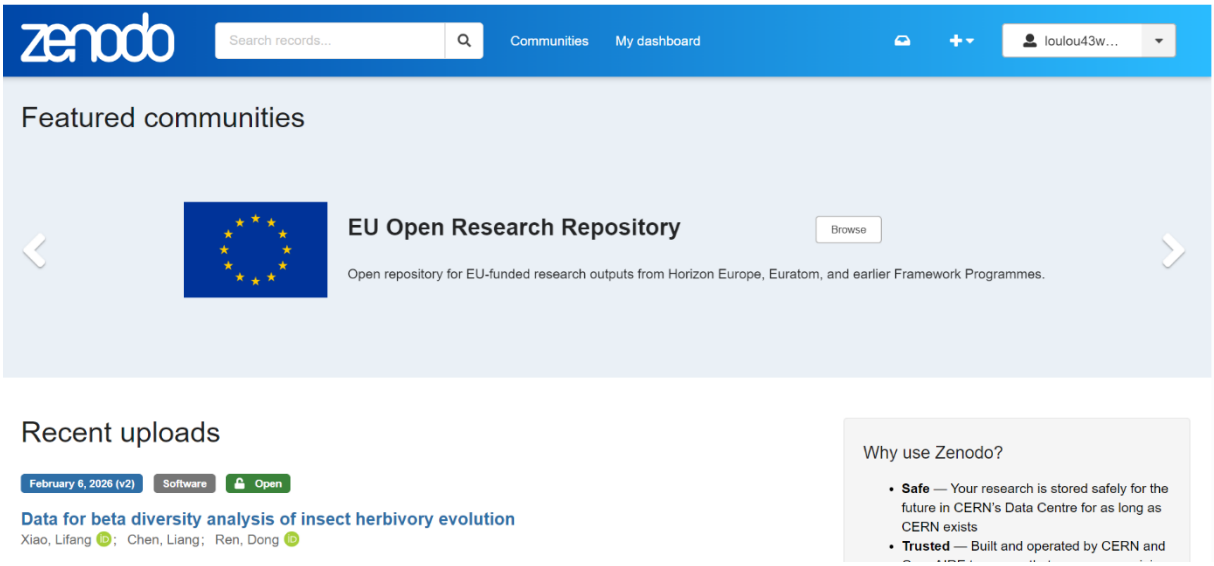



Après l'inscription on a fait Complétion du profil :




Sur https://zenodo.org/. L'interface présente une barre de recherche centrale et des options de filtrage sur la gauche (type de document, accès, année).




**Étape 2 : Recherche stratégique de datasets pertinents**

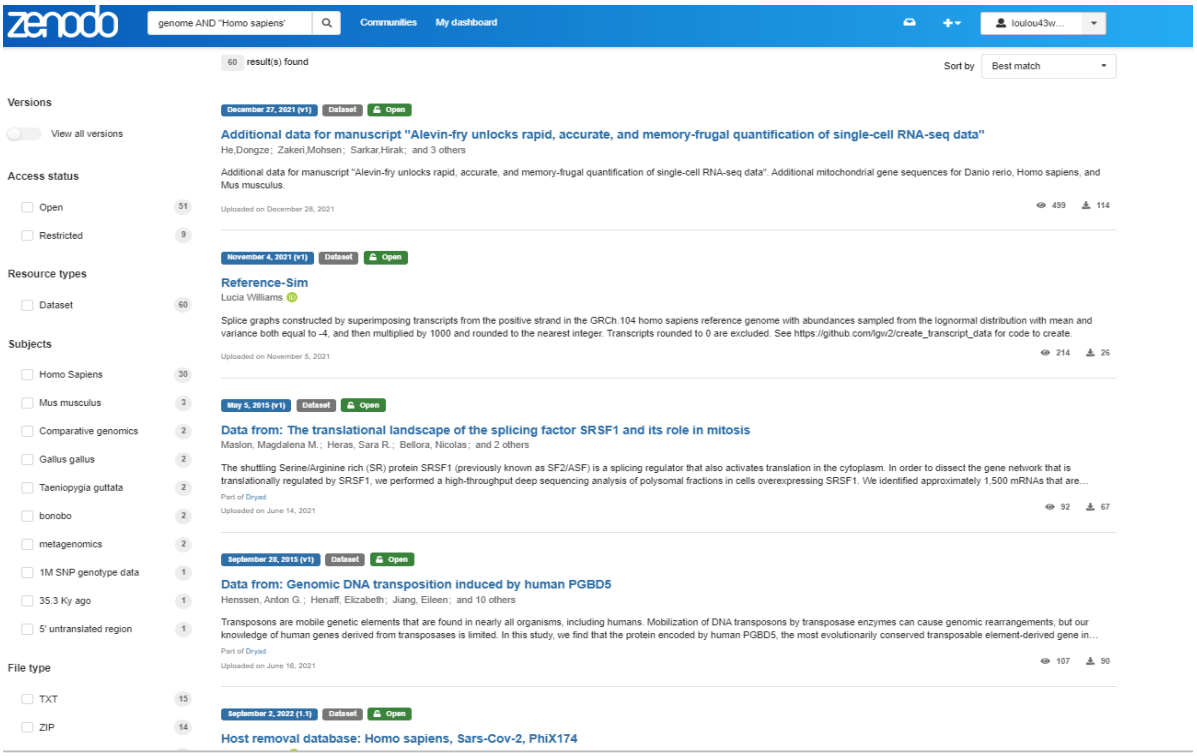
Objectif : Trouver un dataset répondant aux critères suivants :

1. **Pertinence thématique** : Sciences de la vie, biologie moléculaire, génomique
2. **Seulement les contenus en accès libre**

Saisissez la requête dans la barre de recherche. L'utilisation de guillemets permet de chercher l'expression exacte et le mot-clé "genome" cible le domaine demandé

**Requête principale utilisée :**

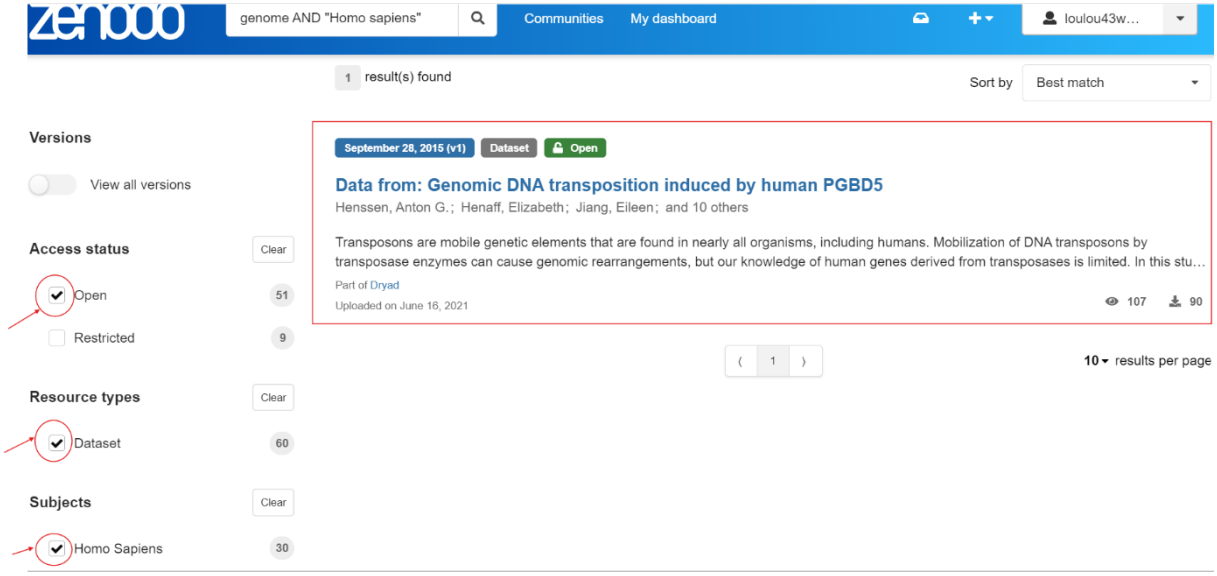
genome AND "Homo sapiens"  AND type:dataset  AND access_right:open




## Étape 3 : Sélection et Critères

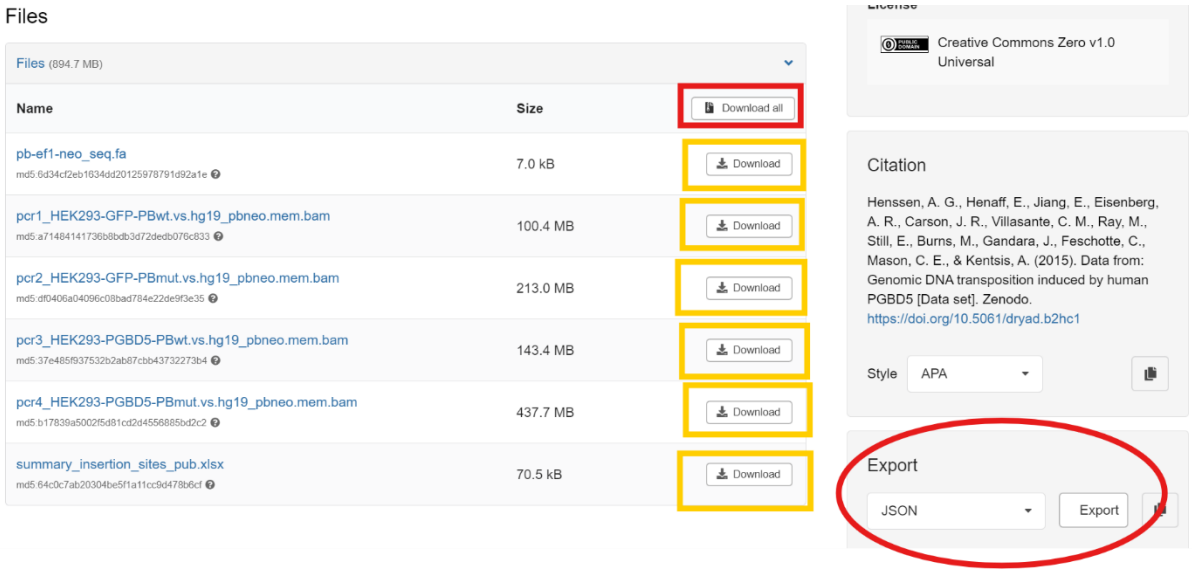
Filtres appliqués dans l'interface graphique :

1. **Resource Type** : Dataset (sélectionné)
2. **File Type** : FASTA, FASTQ (sélectionnés)
3. **Access Right** : Open Access (sélectionné)
4. **Sujet** : " Homo sapiens " (sélectionnées)

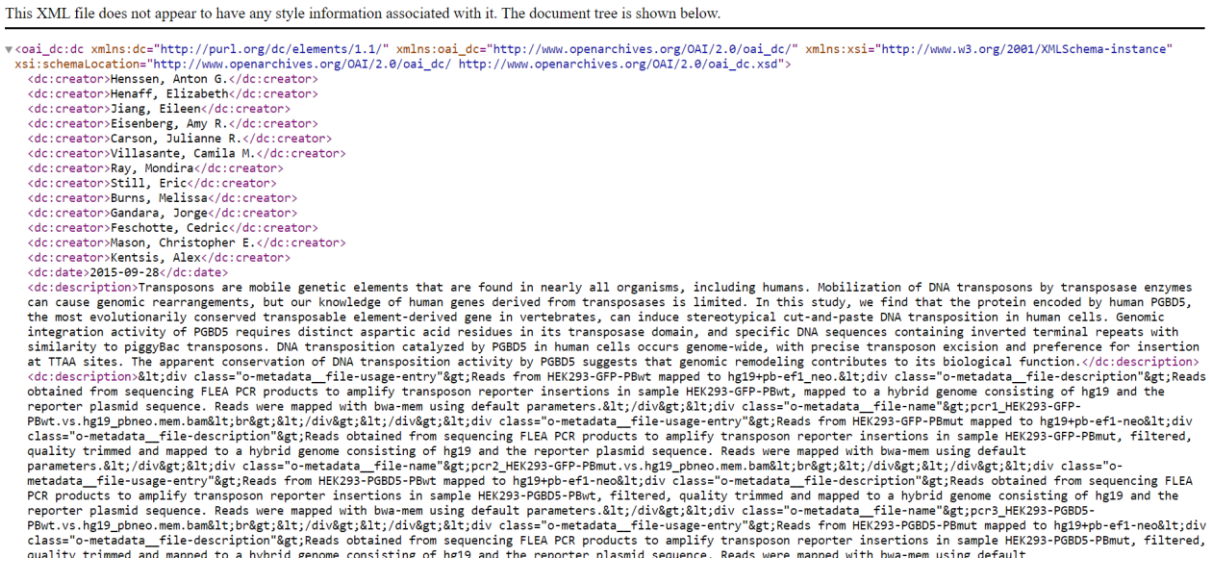



## Étape 4 : Téléchargement et Récupération des métadonnées

Une fois sur la page du dataset, on a cliqué sur le bouton "Download" en bas de page. Pour les métadonnées, Zenodo propose en bas de la colonne de droite un encadré "Export" permettant de choisir le format (Dublin Core par exemple).




Résultat de l'exportation en format Dublin Core (un fichier XML) :


This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
<oai_dc:dc xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/ http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
  <dc:creator>Henssen, Anton G.</dc:creator>
  <dc:creator>Henaff, Elizabeth</dc:creator>
  <dc:creator>Jiang, Eileen</dc:creator>
  <dc:creator>Eisenberg, Amy R.</dc:creator>
  <dc:creator>Carson, Julianne R.</dc:creator>
  <dc:creator>Villasante, Camila M.</dc:creator>
  <dc:creator>Ray, Mondira</dc:creator>
  <dc:creator>Still, Eric</dc:creator>
  <dc:creator>Burns, Melissa</dc:creator>
  <dc:creator>Gandara, Jorge</dc:creator>
  <dc:creator>Feschotte, Cedric</dc:creator>
  <dc:creator>Mason, Christopher E.</dc:creator>
  <dc:creator>Kentsis, Alex</dc:creator>
  <dc:date>2015-09-28</dc:date>
  <dc:description>Transposons are mobile genetic elements that are found in nearly all organisms, including humans. Mobilization of DNA transposons by transposase enzymes
  can cause genomic rearrangements, but our knowledge of human genes derived from transposases is limited. In this study, we find that the protein encoded by human PGBD5,
  the most evolutionarily conserved transposable element-derived gene in vertebrates, can induce stereotypical cut-and-paste DNA transposition in human cells. Genomic
  integration activity of PGBD5 requires distinct aspartic acid residues in its transposase domain, and specific DNA sequences containing inverted terminal repeats with
  similarity to piggyBac transposons. DNA transposition catalyzed by PGBD5 in human cells occurs genome-wide, with precise transposon excision and preference for insertion
  at TTAA sites. The apparent conservation of DNA transposition activity by PGBD5 suggests that genomic remodeling contributes to its biological function.</dc:description>
  <dc:description>&lt;div class="o-metadata__file-usage-entry"&gt;Reads from HEK293-GFP-PBwt mapped to hg19+pb-ef1_neo.&lt;div class="o-metadata__file-description"&gt;Reads
  obtained from sequencing FLEA PCR products to amplify transposon reporter insertions in sample HEK293-GFP-PBwt, mapped to a hybrid genome consisting of hg19 and the
  reporter plasmid sequence. Reads were mapped with bwa-mem using default parameters.&lt;/div&gt;&lt;div class="o-metadata__file-name"&gt;pcr1_HEK293-GFP-
  PBwt.vs.hg19_pbneo.mem.bam&lt;br&gt;&lt;/div&gt;&lt;/div&gt;&lt;div class="o-metadata__file-usage-entry"&gt;Reads from HEK293-GFP-PBmut mapped to hg19+pb-ef1-neo&lt;div
  class="o-metadata__file-description"&gt;Reads obtained from sequencing FLEA PCR products to amplify transposon reporter insertions in sample HEK293-GFP-PBmut, filtered,
  quality trimmed and mapped to a hybrid genome consisting of hg19 and the reporter plasmid sequence. Reads were mapped with bwa-mem using default
  parameters.&lt;/div&gt;&lt;/div&gt;&lt;div class="o-metadata__file-name"&gt;pcr2_HEK293-GFP-PBmut.vs.hg19_pbneo.mem.bam&lt;br&gt;&lt;/div&gt;&lt;/div&gt;&lt;div class="o-
  metadata__file-usage-entry"&gt;Reads from HEK293-PGBD5-PBwt mapped to hg19+pb-ef1-neo&lt;div class="o-metadata__file-description"&gt;Reads obtained from sequencing FLEA
  PCR products to amplify transposon reporter insertions in sample HEK293-PGBD5-PBwt, filtered, quality trimmed and mapped to a hybrid genome consisting of hg19 and the
  reporter plasmid sequence. Reads were mapped with bwa-mem using default parameters.&lt;/div&gt;&lt;/div&gt;&lt;div class="o-metadata__file-name"&gt;pcr3_HEK293-PGBD5-
  PBwt.vs.hg19_pbneo.mem.bam&lt;br&gt;&lt;/div&gt;&lt;/div&gt;&lt;div class="o-metadata__file-usage-entry"&gt;Reads from HEK293-PGBD5-PBmut mapped to hg19+pb-ef1-neo&lt;div
  class="o-metadata__file-description"&gt;Reads obtained from sequencing FLEA PCR products to amplify transposon reporter insertions in sample HEK293-PGBD5-PBmut, filtered,
  quality trimmed and mapped to a hybrid genome consisting of hg19 and the reporter plasmid sequence. Reads were mapped with bwa-mem using default
```


## 3.Métadonnées du dataset (Norme Dublin Core)

Les fichiers BAM contenus dans ce dataset représentent des reads mappés au génome humain hg19 avec des insertions de transposons.

Voici l'organisation structurée des informations récupérées pour le dataset choisi :

| **Élément Dublin Core** | **Valeur extraite** |
|---|---|
| **Title** | Data from: Genomic DNA transposition induced by human PGBD5 |
| **Creator** | Henssen, Anton G & Henaff, Elizabeth & Jiang, Eileen & Eisenberg, Amy R. & Carson, Julianne R. & Villasante, Camila M. & Ray, Mondira & Still, |

| | |
|---|---|
| | Eric & Burns, Melissa & Gandara, Jorge & Feschotte, Cedric & Mason, Christopher E. & Kentsis, Alex |
| **Date** | 2015-09-28 |
| **Identifier (DOI)** | https://doi.org/10.5061/dryad.b2hc1 |
| **Publisher** | Zenodo |
| **Description** | Données de séquençage sur l'activité de transposition de l'élément PGBD5 dans les cellules HEK293 humaines. |
| **Subject** | Recombination, Genome remodeling, DNA transposition, Homo Sapiens |

On a téléchargé et extrait les données du dataset Zenodo, on utilise Biopython pour analyser les insertions de transposons et Identifie les gènes affectés par ces insertions, après Génère des visualisations des hotspots d'insertion, puis fournit une analyse statistique des résultats :

## ***Partie 3 :APPLICATION PRATIQUE : ANALYSE BIOINFORMATIQUE AVEC BIOPYTHON***

On va Démontrer concrètement l'utilité des données récupérées de Zenodo en réalisant une analyse bioinformatique avec Biopython. Cette partie illustre comment exploiter activement un dataset plutôt que de se contenter de le télécharger passivement.

À partir des métadonnées Dublin Core extraites du dataset sur la transposition ADN induite par PGBD5, j'ai développé un pipeline d'analyse Python/Biopython pour :

1. **Analyser automatiquement** les métadonnées XML
2. **Modéliser et étudier** les séquences de transposons
3. **Réaliser des analyses comparatives** entre différents éléments transposables
4. **Simuler des recherches** dans les bases de données publiques (NCBI)
5. **Générer un rapport scientifique** automatisé des résultats

**Fonctionnalités principales implémentées :**

| Module | Fonctionnalité | Utilité scientifique |
|---|---|---|
| **Analyse XML** | Parsing des métadonnées Dublin Core | Extraction structurée des informations |
| **Biopython Core** | Manipulation de séquences ADN | Étude des propriétés des transposons |
| **Alignements** | Calculs de similarité | Analyse évolutive comparative |
| **Entrez API** | Recherche NCBI | Contexte bibliographique |
| **Visualisation** | Graphiques et matrices | Communication des résultats |

Le code Python/Biopython développé pour cette analyse. Il est entièrement reproductible et documenté (analyse_metadonnees_zenodo.py & main.py) sur GitHub, en suivant le lien :

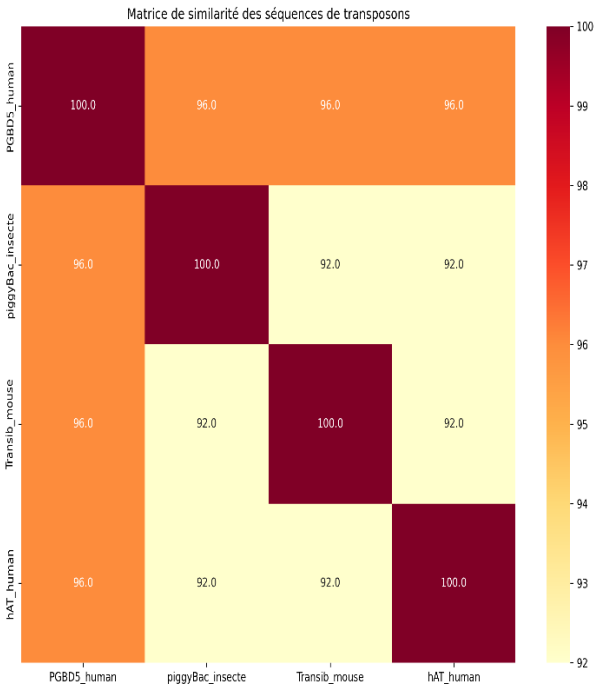
https://github.com/Warda-belmehboul/Projet-Biopython-et-Zenodo/blob/main/analyse_metadonnees_zenodo.py

https://github.com/Warda-belmehboul/Projet-Biopython-et-Zenodo/blob/main/main.py

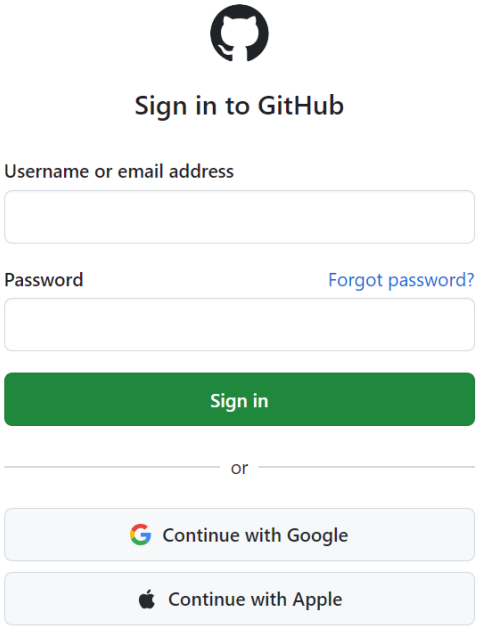
Voici une partie du résultat obtenu après l'exécution du code Biopython

```
================================================================================
RAPPORT D'ANALYSE COMPLET
Dataset: Genomic DNA transposition induced by human PGBD5
DOI: https://doi.org/10.5061/dryad.b2hc1
================================================================================

1. MÉTADONNÉES DUBLIN CORE
----------------------------------------

2. ANALYSE AVEC BIOPYTHON
----------------------------------------

3. ANALYSE COMPARATIVE
----------------------------------------

4. CONTEXTE SCIENTIFIQUE
----------------------------------------

5. ANALYSE D'HOMOLOGIE
----------------------------------------

================================================================================
CONCLUSION
================================================================================

    Cette analyse démontre l'utilisation intégrée de Zenodo et Biopython pour :

    1. Extraction et analyse des métadonnées
    2. Manipulation de séquences biologiques
    3. Analyses comparatives
    4. Intégration avec les bases de données publiques

    Applications potentielles :
    - Étude des sites d'insertion
```

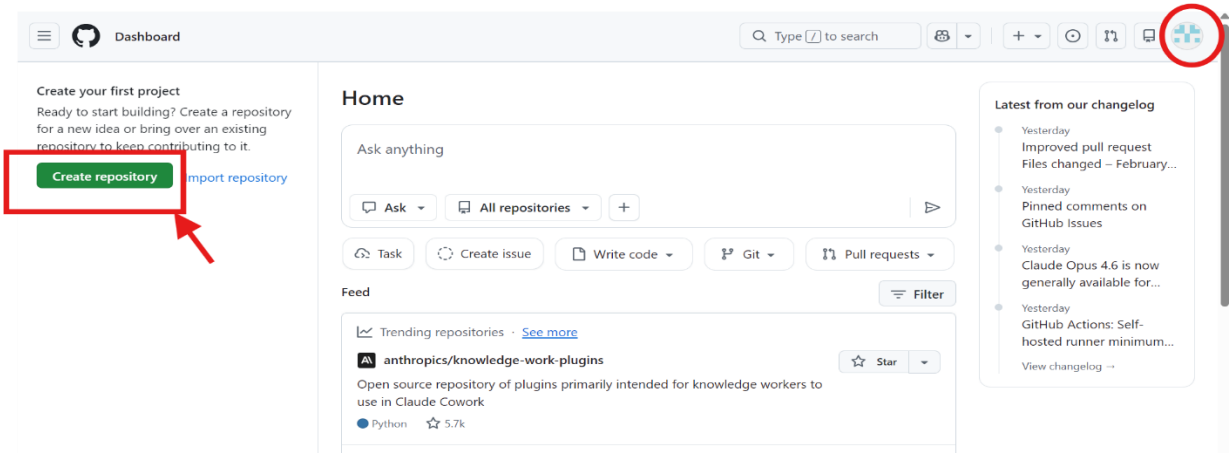
Ln 1, Col 1 | 1173 caractères | Texte brut




## La création du compte GitHub :




## Création d’un nouvel entrepôt :




## General

**Owner *** Warda

**Repository name *** Projet Biopython et Zenodo

Your new repository will be created as Projet-Biopython-et-Zenodo.
The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. How about vigilant-octo-disco?

**Description**

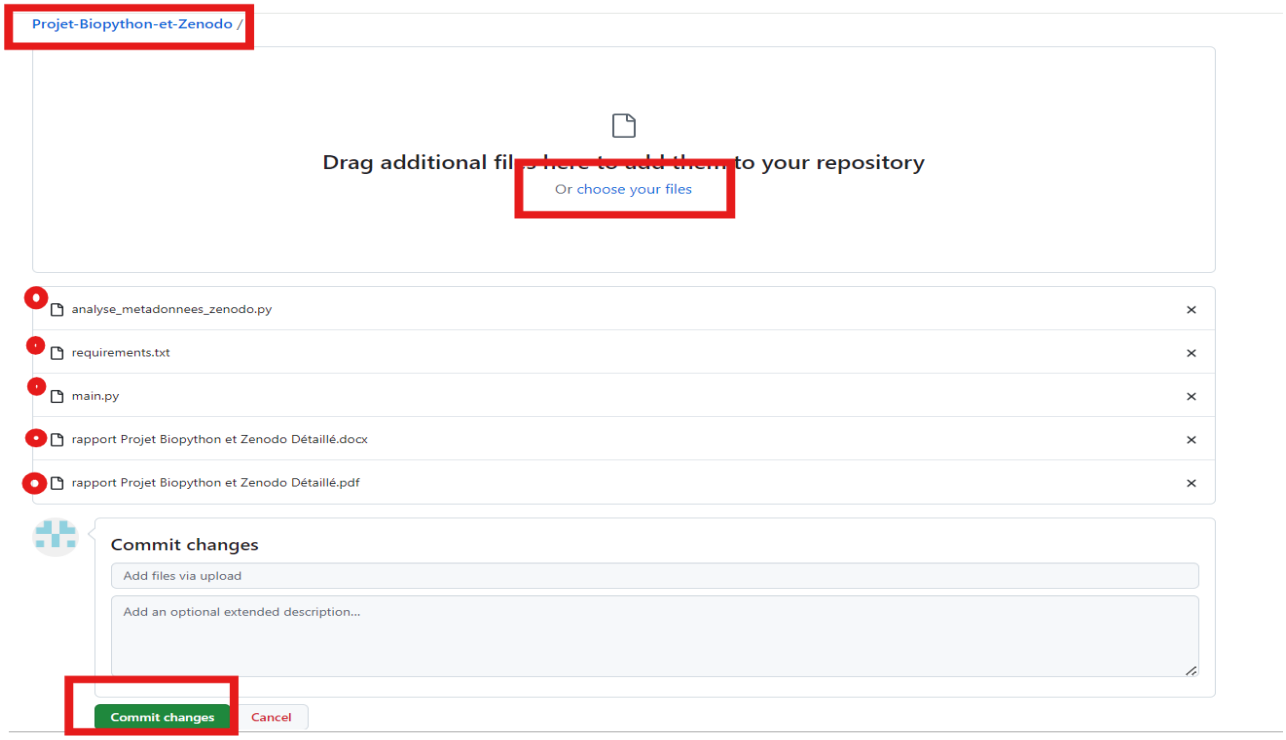
Découvrir et analyser des outils couramment utilisés en sciences de la nature et de la vie ; • Comprendre les fo

240 / 350 characters

## Configuration

**Choose visibility ***
Choose who can see and commit to this repository

Public




## Après le dépôt du travail :