

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Jijel

Faculté des Sciences de la Nature et de la Vie

Année universitaire : 2025 / 2026

TRAVAIL PRATIQUE

Logiciels Libres et Open Source

*Étude théorique et pratique des logiciels libres
en sciences de la nature et de la vie*

Outil étudié : **Biopython**



Réalisé par :

SOUKKOU Mohammed Antar

Encadré par :

Dr Bensalem Amel

SCIENCE OUVERTE

L'évolution fulgurante des technologies de l'information a profondément transformé le paysage de la recherche scientifique, particulièrement dans le domaine des sciences de la nature et de la vie. Aujourd'hui, l'exploitation des données biologiques, qu'elles soient génomiques, moléculaires ou écologiques, repose indéniablement sur la performance des outils informatiques utilisés pour leur analyse et leur interprétation.

LOGICIELS LIBRES

REPRODUCTIBILITÉ

Dans ce contexte, le mouvement des Logiciels Libres et Open Source s'est imposé comme un pilier fondamental de la science moderne. Contrairement aux solutions propriétaires, ces outils offrent une transparence algorithmique, une flexibilité d'adaptation et une accessibilité qui favorisent la reproductibilité des résultats, un critère éthique et scientifique majeur. Cette philosophie de partage s'étend désormais à la gestion des données de recherche à travers le concept de la Science Ouverte (Open Science), visant à rendre les connaissances scientifiques disponibles et réutilisables par tous. L'objectif de ce travail est de s'immerger dans cet écosystème numérique en explorant deux axes complémentaires. Dans un premier temps, nous porterons une analyse théorique sur un outil bioinformatique de référence afin d'en saisir les fonctionnalités techniques et les enjeux pratiques. Dans un second temps, nous aborderons la dimension pratique de la gestion des données via la plateforme Zenodo, en mettant l'accent sur l'importance des métadonnées pour la structuration de l'information scientifique. À travers ce rapport, il s'agira de démontrer comment l'articulation entre logiciels libres et plateformes de partage de données constitue aujourd'hui un levier stratégique pour l'avancement des sciences de la nature et de la vie.

OPEN SCIENCE

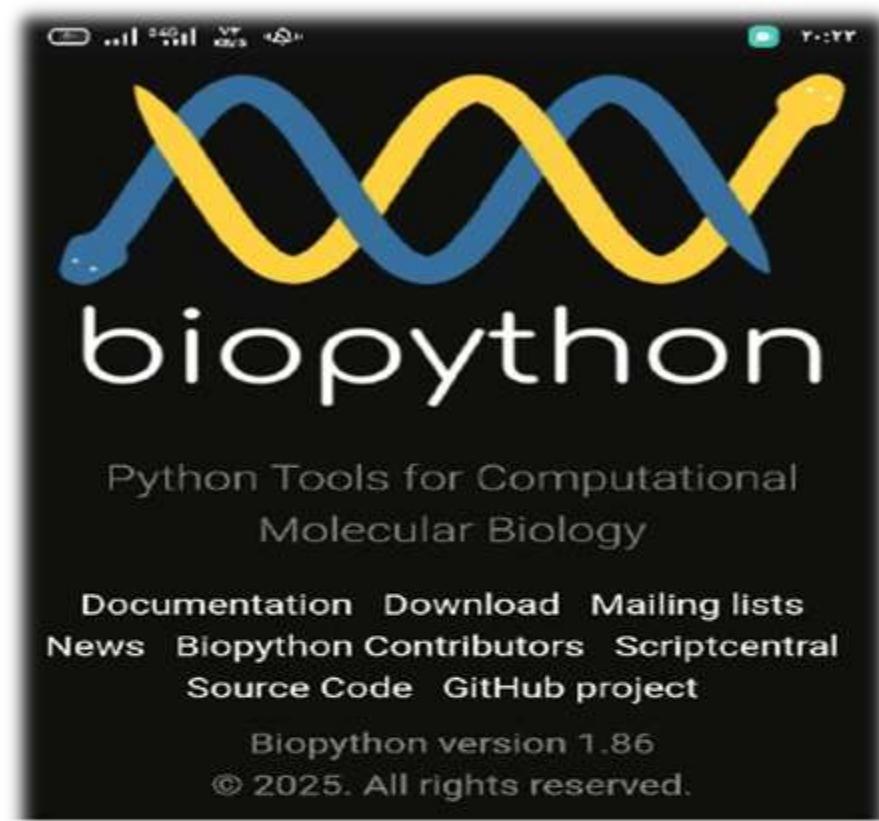
PARTIE 1_ étude théorique de Biopython

1_Présentation générale de l'outil Biopython

est une bibliothèque open source constituée de modules écrits en langage Python, destinée à la biologie computationnelle et à la bioinformatique. Elle fournit un ensemble d'outils permettant de manipuler, analyser et interpréter des données biologiques telles que les séquences d'ADN, d'ARN et de protéines [1].

Le projet Biopython a été lancé au début des années 2000 et est développé par une communauté internationale de chercheurs et de développeurs bénévoles, dans le but de rendre les outils bioinformatiques plus accessibles aux biologistes et aux étudiants [2].

Biopython est distribué sous une licence open source, ce qui signifie que son code source est librement accessible, modifiable et réutilisable aussi bien à des fins académiques que scientifiques [2].



Biopython

See also our [News feed](#).

Introduction

Biopython is a set of freely available tools for biological computation written in [Python](#) by an international team of developers.

Biopython:

est un outil important en sciences de la vie car il permet de traiter efficacement de grandes quantité de données biologiques, notamment les données issues du séquençage génomique, qui sont de plus en plus volumineuses dans la recherche moderne [1].

Il propose des fonctions permettant la lecture, l'écriture et la conversion de plusieurs formats standards de données biologiques tels que FASTA GenBank et EMBL, facilitant ainsi l'échange et l'analyse des données entre chercheurs [1].

Biopython intègre également des modules permettant l'accès direct à des bases de données biologiques internationales, comme celles du National Center for Biotechnology Information (NCBI), ce qui favorise l'automatisation des recherches et l'extraction rapide d'informations biologiques pertinentes [3]

Tutorial

The Biopython Tutorial and Cookbook ([HTML](#), [PDF](#)) contains the bulk of our documentation. See [Documentation](#) for more links.

Quick example

Try executing this in python:

```
from Bio.Seq import Seq  
  
# create a sequence object  
my_seq = Seq("CATGTAGACTAG")  
  
# print out some details about it  
print("seq %s is %i bases long" % (my_seq))  
print("reverse complement is %s" % my_seq.reverse_complement())  
print("protein translation is %s" % my_seq.translate())
```

You should get the following output:

```
seq CATGTAGACTAG is 12 bases long  
reverse complement is CTAGTCTACATG  
protein translation is HVD*
```

This was a very quick demonstration of Biopython's [Seq](#) (sequence) object and some of its methods.

Reading and writing Sequence Files



Rôle de Biopython en biologie :

Biopython joue un rôle central dans la biologie computationnelle en permettant aux biologistes et bioinformaticiens de programmer des analyses de séquences biologiques, de réaliser des alignements, d'analyser des structures moléculaires et de créer des pipelines d'analyse reproductibles [3].

Grâce à sa structure modulaire et à sa documentation détaillée, Biopython est utilisé aussi bien dans l'enseignement universitaire, pour l'apprentissage de la bioinformatique, que dans la recherche scientifique, pour l'analyse de données réelles issues d'expériences biologiques [4].



Table of contents

- Biopython Tutorial & Cookbook
 - Introduction
 - Quick Start – What can you do with Biopython?
 - Sequence objects
 - Sequence annotation objects
 - Sequence Input/Output
 - Sequence alignments
 - Pairwise sequence alignment
 - Multiple Sequence Alignment objects
 - Pairwise alignments using pairwise2
 - BLAST (new)
 - BLAST (old)
 - BLAST and other sequence search tools
 - Accessing NCBI's Entrez databases
 - Swiss-Prot and ExPASy
 - Going 3D: The PDB module
 - Bio.PopGen: Population genetics
 - Phylogenetics with Bio.Phylo
 - Sequence motif analysis using Bio.motifs
 - Cluster analysis
 - Graphics including GenomeDiagram
 - KEGG
 - Bio.phenotype: analyze phenotypic



2_Fonctionnalités principales de Biopython :

Biopython offre un large ensemble de fonctionnalités destinées à faciliter le travail des biologistes et des bioinformaticiens dans l'analyse et la gestion des données biologiques.

2.1 Manipulation et analyse des séquences biologiques :

Biopython permet la manipulation directe des séquences d'ADN, d'ARN et de protéines, notamment le calcul de la longueur des séquences, la recherche de motifs, la traduction ADN → protéine et la manipulation de séquences complémentaires [1].

```
>>> from Bio.Seq import Seq  
>>> my_seq = Seq("GATCG")  
>>> for index, letter in enumerate(my_seq)  
...     print("%i %s" % (index, letter))  
...  
0 G  
1 A  
2 T  
3 C  
4 G  
>>> print(len(my_seq))  
5
```

You can access elements of the sequence in the same way as for strings (but remember, Python counts from zero!):

```
>>> print(my_seq[0]) # first letter  
G  
>>> print(my_seq[2]) # third letter  
T  
>>> print(my_seq[-1]) # last letter  
G
```

The `Seq` object has a `.count()` method, just like a string. Note that this means that like a Python string, this gives a non-overlapping count:

```
>>> from Bio.Seq import Seq  
>>> "AAAA".count("AA")  
2  
>>> Seq("AAAA").count("AA")  
-
```

2.2 Lecture et écriture des formats biologiques standards:

L'outil prend en charge la lecture, l'écriture et la conversion de nombreux formats de fichiers biologiques, tels que FASTA, GenBank, EMBL et Swiss-Prot, ce qui facilite l'échange et la standardisation des données biologiques [1].



```
Simple FASTA parsing example

If you open the lady slipper orchids FASTA file ls_orchid.fasta in your favorite text editor, you'll see that the file starts like this:

>G1|2765658|emb|Z78533.1|C1Z78533 C. irape
CGTAACAAGGTTCCGTAGGTGAACCTGGAAAGGAAGGA
+ ...

It contains 94 records, each has a line
starting with > (greater-than symbol)
followed by the sequence on one or more
lines. Now try this in Python:

>>> from Bio import SeqIO
>>> for seq_record in SeqIO.parse("ls_orc
...     print(seq_record.id)
...     print(repr(seq_record.seq))
...     print(len(seq_record))
...
You should get something like this on your
screen:
```



```
Simple GenBank parsing example

Now let's load the GenBank file
ls_orchid.gbk instead - notice that the code
to do this is almost identical to the snippet
used above for the FASTA file - the only
difference is we change the filename and
the format string:

>>> from Bio import SeqIO
>>> for seq_record in SeqIO.parse("ls_orc
...     print(seq_record.id)
...     print(repr(seq_record.seq))
...     print(len(seq_record))
...
This should give:
```

```
Z78533.1
SeqC 'CGTAACAAGGTTCCGTAGGTGAACCTGGAAAGGA
740
+
Z78439.1
Seq( 'CATTGTTGAATCACATAATAATTGATCGAGTTAAT
592

You'll notice that a shorter string has been
used as the seq_record.id in this case.

I love parsing – please don't
```



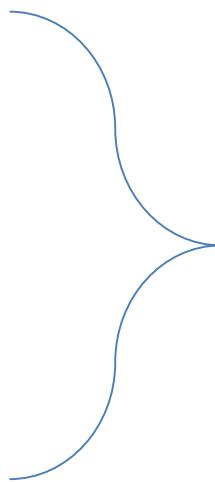
Parsing sequence file formats

A large part of much bioinformatics work involves dealing with the many types of file formats designed to hold biological data. These files are loaded with interesting biological data, and a special challenge is parsing these files into a format so that you can manipulate them with some kind of programming language. However the task of parsing these files can be frustrated by the fact that the formats can change quite regularly, and that formats may contain small subtleties which can break even the most well designed parsers.

We are now going to briefly introduce the `Bio.SeqIO` module - you can find out more in Chapter Sequence Input/Output. We'll start with an online search for our friends, the lady slipper orchids. To keep this introduction simple, we're just using the NCBI website by hand. Let's just take a look through the nucleotide databases at NCBI, using an Entrez online search (<https://www.ncbi.nlm.nih.gov/nucleotide/?term=Cypripedioideae>) for everything mentioning the text Cypripedioideae (this is

2.3 Accès aux bases de données biologiques:

Biopython fournit des modules permettant un accès programmatique à des bases de données biologiques en ligne, notamment celles du NCBI (GenBank, PubMed, BLAST). Cela permet d'automatiser la recherche, le téléchargement et l'analyse de données biologiques [3].



The screenshot shows a terminal window with the title "EInfo: Obtaining information about the Entrez databases". The window contains Python code demonstrating how to use the `einfo` function from the `Bio.Entrez` module to retrieve information about NCBI databases. It also shows the resulting XML output containing a list of database names.

```
>>> from Bio import Entrez
>>> Entrez.email = "A.N.Other@example.com"
>>> stream = Entrez.einfo()
>>> result = stream.read()
>>> stream.close()

>>> print(result)
<?xml version="1.0"?>
<!DOCTYPE eInfoResult PUBLIC "-//NLM//DTD
  "https://www.ncbi.nlm.nih.gov/entrez/que
<eInfoResult>
<DbList>
  <DbName>pubmed</DbName>
  <DbName>protein</DbName>
  <DbName>nucleotide</DbName>
  <DbName>nuccore</DbName>
  <DbName>nucgss</DbName>
  <DbName>nucest</DbName>
  <DbName>structure</DbName>
```

2.4 Outils d'analyse bioinformatique:

La bibliothèque inclut des fonctionnalités pour réaliser des alignements de séquences, des analyses phylogénétiques et le traitement de structures moléculaires, ce qui en fait un outil complet pour la bioinformatique [3].

There is also an optional `seq_count` argument which is discussed in Section **Ambiguous Alignments** below for dealing with ambiguous file formats which may contain more than one alignment.

Single Alignments

As an example, consider the following annotation rich protein alignment in the PFAM or Stockholm file format:

```
# STOCKHOLM 1.0
##=GS COATB_BPIKE/30-81 AC P03620_1
##=GS COATB_BPIKE/30-81 DR PDB: 1ifl : 1-
##=GS Q9TQ08_BPIKE/1-52 AC Q9TQ08_1
##=GS COATB_BPI22/32-83 AC P15416_1
##=GS COATB_BPM13/24-72 AC P69541_1
##=GS COATB_BPM13/24-72 DR PDB: 2cpb : 1-
##=GS COATB_BPM13/24-72 DR PDB: 2cps : 1-
##=GS COATB_BPZJ2/1-49 AC P03618_1
##=GS Q9TQ09_BPFD/1-49 AC Q9TQ09_1
##=GS Q9TQ09_BPFD/1-49 DR PDB: 1nh4 A: 1
##=GS COATB_BPIF1/22-73 AC P03619_2
##=GS COATB_BPIF1/22-73 DR PDB: 1itk : 1-
COATB_BPIKE/30-81 AEPNAATNYAT
##=GR COATB_BPIKE/30-81 SS -HHHHHHHHHH
Q9TQ08_BPIKE/1-52 AEPNAATNYAT
COATB_BPI22/32-83 DGTSTATSYAT
COATB_BPM13/24-72 AEGDDP...AK
##=GR COATB_BPM13/24-72 --S-T...CH
COATB_BPZJ2/1-49 AEGDDP...AK
Q9TQ09_BPFD/1-49 AEGDDP...AK
##=GR Q9TQ09_BPFD/1-49 SS -----
COATB_BPIF1/22-73 FAADDATSQAK
##=GR COATB_BPIF1/22-73 SS XX-HHHH--HH
```

2.5 Intégration dans des pipelines d'analyse :

Grâce à son intégration avec le langage Python, Biopython permet de créer des scripts et des pipelines d'analyse reproductibles, combinant plusieurs étapes de traitement des données biologiques de manière automatique [2]

BLAST and other sequence search tools

Biological sequence identification is an integral part of bioinformatics. Several tools are available for this, each with their own algorithms and approaches, such as BLAST (arguably the most popular), FASTA, HMMER, and many more. In general, these tools usually use your sequence to search a database of potential matches. With the growing number of known sequences (hence the growing number of potential matches), interpreting the results becomes increasingly hard as there could be hundreds or even thousands of potential matches. Naturally, manual interpretation of these searches' results is out of the question. Moreover, you often need to work with several sequence search tools, each with its own statistics, conventions, and output format. Imagine how daunting it would be when you need to work with multiple sequences using multiple search tools.

Site officiel:

<https://biopython.org/docs/latest/Tutorial/index.html>

3. Aspects techniques de l'outil Biopython :

Les aspects techniques de Biopython permettent de comprendre son fonctionnement informatique ainsi que les conditions nécessaires à son utilisation dans un environnement scientifique.

3.1 Langage de programmation :

Biopython est entièrement développé en langage Python, un langage de programmation de haut niveau largement utilisé en sciences pour sa simplicité, sa lisibilité et sa compatibilité avec de nombreux outils scientifiques [1].

3.2 Licence et caractère open source :

Biopython est distribué sous une licence open source permissive, autorisant librement l'utilisation, la modification et la redistribution du code source. Cette licence favorise la collaboration scientifique et le partage des connaissances [2].

3.3 Systèmes d'exploitation compatibles :

Biopython est multiplateforme, ce qui signifie qu'il peut être utilisé sur différents systèmes d'exploitation tels que Windows, Linux et macOS, à condition que Python soit installé sur le système [2].

3.4 Installation et dépendances :

L'installation de Biopython est simple et peut être réalisée à l'aide du gestionnaire de paquets Python pip. L'outil ne nécessite que peu de dépendances externes, ce qui facilite son déploiement dans des environnements académiques et de recherche [3].

3.5 Intégration avec d'autres outils scientifiques :

Biopython peut être intégré avec d'autres bibliothèques scientifiques Python telles que NumPy ou SciPy, permettant ainsi de réaliser des analyses bioinformatiques plus avancées et des traitements de données complexes [1].

4 Points forts de Biopython:

Biopython présente plusieurs points forts qui expliquent son adoption vaste expérience dans la bioinformatique et des sciences de la vie.

4.1 Logiciel gratuit et open source Biopython est un logiciel entièrement gratuit et open source, ce qui permet à tous les chercheurs, enseignants et étudiants d'y accéder sans contraintes financières. Le caractère open source favorise également la transparence scientifique et la reproductibilité des travaux de recherche [1].

4.2 Communauté scientifique active Biopython est soutenu par une communauté internationale active de développeurs et de chercheurs, qui contribue continuellement à son amélioration, à la correction des erreurs et à l'enrichissement de ses fonctionnalités [2]

4.3 Documentation riche et bien structurée :

L'outil dispose d'une documentation officielle complète et régulièrement mise à jour, comprenant des tutoriels, des exemples et des explications détaillées, ce qui facilite son apprentissage et son utilisation par les débutants comme par les utilisateurs avancés [3].

4.4 Large champ d'application en biologie:

Biopython est utilisé dans de nombreux domaines des sciences de la vie, notamment la génomique, la transcriptomique, la protéomique et la phylogénie, ce qui en fait un outil polyvalent et adapté à différents contextes de recherche biologique [1].

4.5 Intégration facile avec l'écosystème :

Python Grâce à son intégration avec l'écosystème Python, Biopython peut être combiné avec d'autres bibliothèques scientifiques (NumPy, SciPy, pandas), permettant ainsi de réaliser des analyses complexes et des pipelines de traitement de données efficaces [2]

5_Limites et points faibles de Biopython :

- Nécessite des connaissances de base en Python pour l'utilisation des fonctions avancées [1]
- Moins adapté pour traiter des volumes de données très importants (ex. génomes volumineux, centaines de milliers de séquences) [3]
- Pas d'interface graphique native ; fonctionne uniquement via du code Python [2]

6_ Conclusion :

Biopython est un outil open source essentiel en bioinformatique, largement utilisé en recherche et en enseignement. Grâce à sa richesse fonctionnelle et à son intégration dans l'écosystème Python, il constitue une solution efficace pour l'analyse des données biologiques. Malgré certaines limites techniques, Biopython reste un choix pertinent et fiable pour les applications en sciences de la vie

Références

- [1] Wikipedia – Biopython <https://en.wikipedia.org/wiki/Biopython>
- [2] Site officiel de Biopython <https://biopython.org>
- [3] Documentation officielle de Biopython – Introduction
https://biopython.org/docs/latest/Tutorial/chapter_introduction.html
- [4] [Bio.dev](#) – Présentation de Biopython <https://bio.dev/tools/biopython>

PARTIE 2_ étude pratique :exploration de Zenodo

1_Présentation de la plateforme Zenodo :

Zenodo est une plateforme de dépôt ouverte et gratuite destinée au stockage et au partage de résultats de recherche scientifique. Elle a été développée par le CERN dans le cadre du projet européen OpenAIRE dans le but de permettre aux chercheurs de publier facilement leurs données, publications, logiciels et autres ressources numériques liées à la recherche.

Objectifs de la plateforme Zenodo :

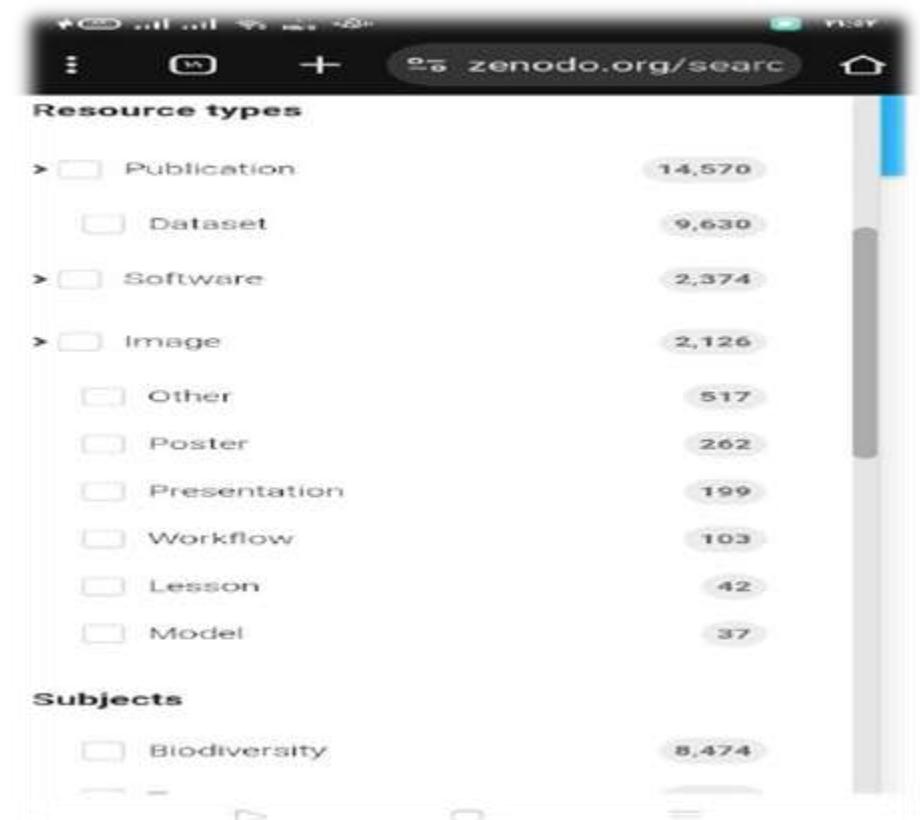
- Archivage et diffusion ouverte des résultats scientifiques, y compris les datasets, publications, logiciels et autres contenus numériques [1][2].
- Fournir un identifiant pérenne (DOI) à chaque dépôt, facilitant la citation et la réutilisation dans d'autres recherches [1][3].
- Encourager la conformité avec les politiques de science ouverte, en particulier celles qui exigent la diffusion libre des données issues de travaux scientifiques [3][4].



Types de contenus hébergés :

Zenodo accepte différents types de contenus scientifiques, notamment :

- Datasets (jeux de données scientifiques), très utiles pour les analyses en biologie, génomique, écologie, etc. [1][2]
- Publications (articles, rapports, posters) [1]
- Logiciels et codes de recherche [1]
- Présentations multimédias (audio, images, vidéos, posters) [1]
- Autres fichiers numériques liés à la recherche scientifique [2].
- Chaque dépôt inclut des métadonnées détaillées, qui facilitent la compréhension et l'usage ultérieur du contenu [2].



Importance de Zenodo en science ouverte et pour la biologie :

Dans le contexte de la science ouverte :

- Zenodo offre un accès libre et gratuit à toutes les ressources déposées, sans barrière d'abonnement ou paiement [2][4].
- La plateforme assure la préservation à long terme des données, même si les projets de recherche ne sont plus actifs [3].
- Elle relie les données à des plateformes plus larges comme OpenAIRE, augmentant la visibilité et l'accès global [5]



Pour la recherche en biologie :

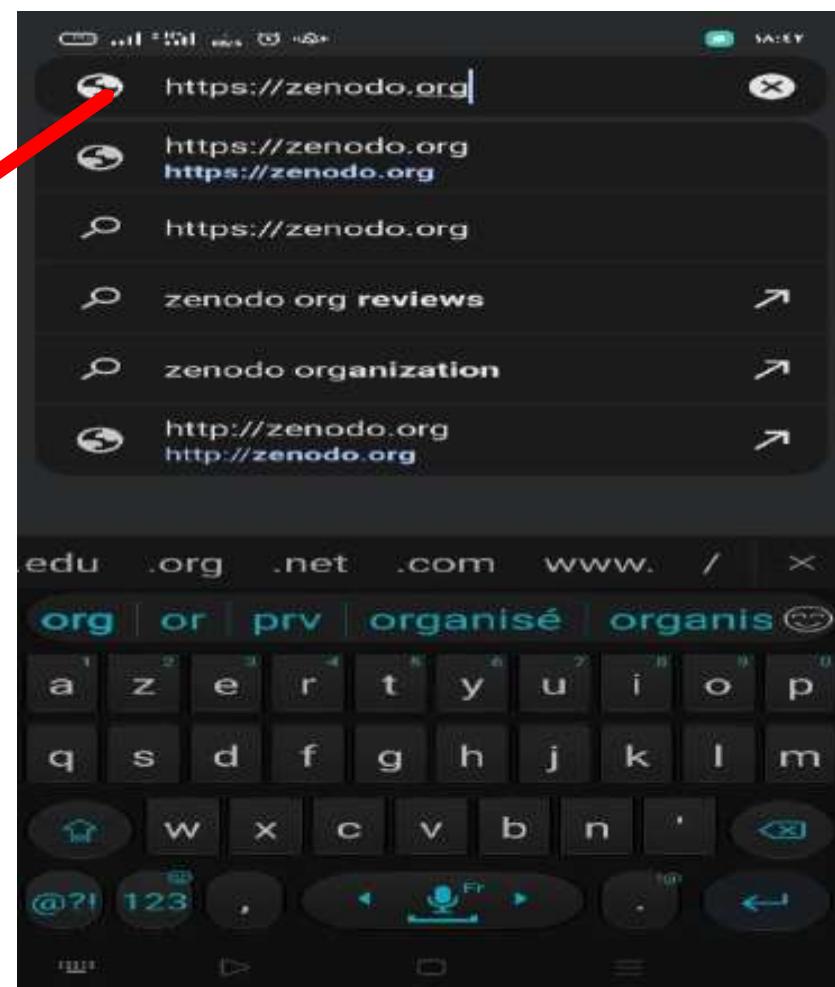
- Zenodo permet de partager et réutiliser des jeux de données biologiques, comme des séquences génomiques, données écologiques ou enregistrements d'espèces [1][2].
- Les données publiées peuvent être réexaminées, reproduites ou combinées avec d'autres données, renforçant la rigueur scientifique et la collaboration internationale [4].
- Les métadonnées riches facilitent les analyses avancées et l'intégration dans des bases de données biologiques [2]

2>Description des étapes réalisées:

Accès à la plateforme Zenodo

Dans un premier temps, nous avons accédé à la plateforme Zenodo via le navigateur web en utilisant l'adresse officielle suivante : <https://zenodo.org>

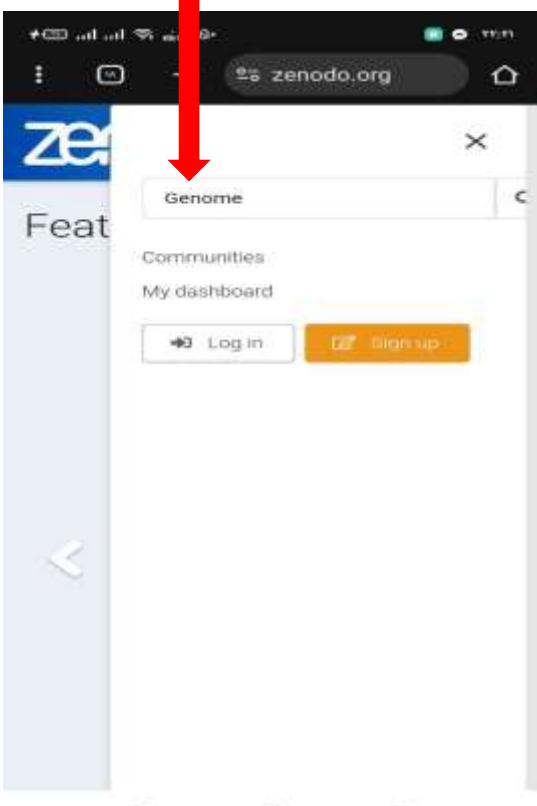
Zenodo est une plateforme ouverte qui permet la consultation, le partage et le téléchargement de données scientifiques sans obligation de créer un compte utilisateur pour la recherche ou le téléchargement.



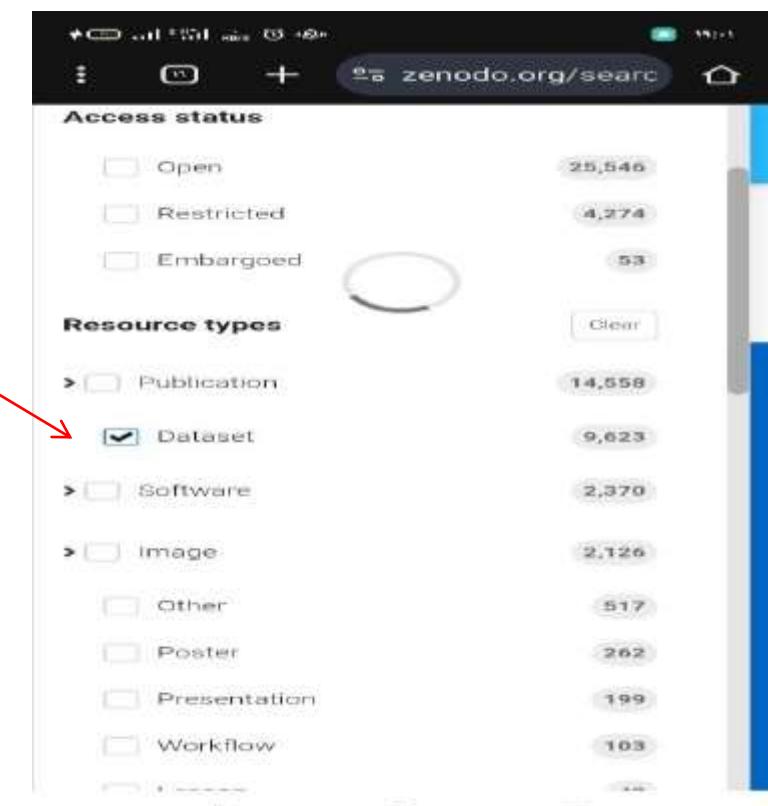
2_Par la suite , nous procémons à l'accès à la plateforme , telle qu'elle se présente actuellement:



3_Effectuer une recherche de dataset à l'aide d'une requete contenant au moins un des mots-clés comme cell ,tessu , genome



4_Sélectionner un **Dataset** pertinent



5_ Télécharger le Dataset choisi

6_. Sélection du Dataset Parmi les résultats obtenus, un Dataset pertinent a été sélectionné selon plusieurs critères :

- La pertinence par rapport au domaine de la biologie et de la biodiversité
- La disponibilité d'un Darwin Core Archive
- La présence de métadonnées complètes (titre, auteurs, description, DOI, licence)
- Dataset sélectionné : Abundance of waterbirds at Lake Miedwie, Poland (2002–2024)

The screenshot shows a dataset page with the following details:

- Title:** Abundance of waterbirds at Lake Miedwie, Poland (2002–2024) (Version 2) – Darwin Core Archive
- Author:** Marchowski, Dominik (Project leader)
- Description:** This dataset contains long-term waterbird count data from Lake Miedwie, Poland, collected between 2002 and 2024. Counts were performed during three periods each season (November, January, and March) and include both game and protected species. The dataset provides seasonal abundance data useful for population trend analyses and conservation planning.
- Files:** miedwie_dwca_updated.zip

Description

The screenshot shows a dataset page with the following details:

- Title:** R codes for quantifying a mass mortality event in freshwater wildlife within the Lower Odra River
- Author:** Marchowski, Dominik
- Description:** In the summer of 2022, the Odra River in Europe was struck by an ecological calamity. Toxins fro...
- Statistics:** Uploaded on November 2, 2023 | Published in: Science of the Total Environment, 907(1), 167898; ISSN: 0049-9679, 2023. 64 43
- Version:** April 16, 2025 (v2) Dataset Open
- Dataset Title:** Abundance of waterbirds at Lake Miedwie, Poland (2002–2024) (Version 2) – Darwin Core Archive
- Author:** Marchowski, Dominik
- Description:** This dataset contains long-term waterbird count data from Lake Miedwie, Poland, collected...
- Statistics:** Uploaded on April 16, 2025. 51 24
- Version:** November 8, 2022 (v1) Dataset Open
- Dataset Title:** Drones, automatic counting tools and artificial neural networks in wildlife population censusing
- Author:** Marchowski, Dominik
- Description:** Drones, automatic counting tools and artificial neural networks in wildlife population censusing

7_ Présenter les informations :

The screenshot shows a Zenodo dataset page with several sections highlighted by red arrows:

- DOI**: Points to the DOI field in the "Details" section, which contains the value [10.5281/zenodo.15231820](https://doi.org/10.5281/zenodo.15231820).
- License**: Points to the "License" section, which displays the Creative Commons Attribution 4.0 International license logo and text.
- creators**: Points to the "Citation" section, which lists the creator as Marchowski, D. (2025).
- Files**: Points to the "Files" section, which shows a zip file named "miedwie_dwca_updated.zip" containing three XML files: eml.xml, event.txt, and meta.xml.

Details

DOI
DOI [10.5281/zenodo.15231820](https://doi.org/10.5281/zenodo.15231820)

Resource type
Dataset

Publisher
Zenodo

Languages
English

Rights

License
Creative Commons Attribution 4.0 International

Citation
Marchowski, D. (2025). Abundance of waterbirds at Lake Miedwie, Poland (2002–2024) (Version 2) – Darwin Core Archive [Data set]. Zenodo.
<https://doi.org/10.5281/zenodo.15231820>

Style APA

Export JSON Export

Technical metadata
Created April 16, 2025
Modified April 16, 2025

Files

miedwie_dwca_updated.zip

<input type="checkbox"/>	miedwie_dwca_updated.zip	2.6 kB
<input type="checkbox"/>	eml.xml	188.2 kB
<input type="checkbox"/>	event.txt	1.2 kB
<input type="checkbox"/>	meta.xml	

8_Critères de sélection d'un Dataset:

- Pertinence par rapport au sujet Le Dataset doit être directement lié au thème de l'étude ou de la recherche. Exemple : Comme le sujet concerne le génome ou les données biologiques, nous avons choisi un Dataset contenant des informations sur des génomes humains ou animaux, ou des données biologiques conformes au standard Darwin Core.
- Disponibilité complète des métadonnées Le Dataset doit contenir des métadonnées claires et complètes, telles que :

Titre (Abundance of waterbirds at lake meidwie)

Auteurs / Créateurs (Dominik Marchowski)

Description du Dataset (2002-2024)

DOI ou identifiant numérique (DOI) :10.5281/zenodo.15231820

Licence (creative commons attribution)

Ceci facilite la présentation des informations dans le rapport selon la norme Darwin Core.

- Accessibilité et facilité de téléchargement Le fichier doit être disponible en téléchargement direct depuis Zenodo, sans nécessiter d'outils particuliers ni de compte obligatoire. La taille des fichiers doit être raisonnable pour permettre une manipulation facile pendant l'exercice pratique.
- Type et format des fichiers Les fichiers doivent être dans un format adapté à l'analyse ou à la présentation dans le rapport : CSV, TSV, FASTA, ou Darwin Core Archive (DwC-A). Exemple : l'utilisation d'un Darwin Core Archive permet d'appliquer directement le standard Darwin Core.
- Fiabilité de la source Le Dataset doit provenir de Zenodo, une plateforme reconnue et fiable pour les données scientifiques ouvertes. Il est préférable que le Dataset soit associé à un projet connu ou à des chercheurs renommés.
- Actualité et période de collecte Le Dataset doit être récent ou couvrir une période suffisamment actuelle pour garantir que les données sont pertinentes et fiables

Métdonnées (Metadata) du Dataset -Lake Miedwie:

Champ Darwin Core	Contenu du Dataset
Dataset Name	Abundance of waterbirds at Lake Miedwie, Poland (2002–2024)
creator / author	Dominik Marchowski (Project leader)
Date Published	2024
Identifier / DOI	10.5281/zenodo.15231820
description	Dataset sur l'abondance des oiseaux aquatiques au lac Miedwie en Pologne, collecté entre 2002 et 2024. Contient des occurrences d'espèces, dates, et coordonnées géographiques.
keywords	birds, waterbirds, Poland, Lake Miedwie, ecology, biodiversity
License	CC BY 4.0 (Creative Commons Attribution 4.0 International)
Format	Darwin Core Archive (ZIP contenant CSV et XML)
Language	English
location / country	Poland
Georeference	Latitude / Longitude disponibles dans les fichiers event.csv et occurrence.csv

Références:

- [1] Wikipedia – Zenodo. Zenodo. Disponible sur : <https://en.wikipedia.org/wiki/Zenodo>
- [2] FORRT – Glossaire (Zenodo, science ouverte). Disponible sur :
https://forrt.org/glossary/arabic/zenodo/?utm_source=chatgpt.com
- [3] Zenodo – Horizon Zen project. Disponible sur : https://about.zenodo.org/projects/horizon-zen/?utm_source=chatgpt.com
- [4] ResearchGate – FAIR & Zenodo. Disponible sur :
https://www.researchgate.net/publication/391580267_ZENODO_A_PLATFORM_FOR_OPEN_ACCESS_AND_SUSTAINABLE_DIGITAL_RESEARCH_REPOSITORY?utm_source=chatgpt.com

Partie 3: GitHub

<https://github.com/mokasoko/rapport-logiciel-libre.git>