



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Universitaire de Mila
Institut des Sciences de la Nature et la Vie
Département des science biologiques et agricoles
Master académique
Spécialité : Biochimie Appliquée



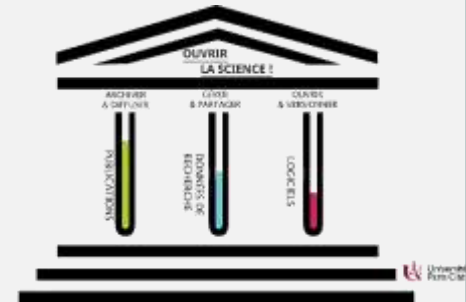
Intitulé de la matière : Logiciels libres et Open-Sources 2

1

CHAPITRE 03:

OUTILS OPEN SOURCE ET APPLICATION EN BIOLOGIE

Préparé par M. Kamel Belmehboul



Année univ. 2025-2026

1. Analyse des séquences génomiques avec Biopython

2

- Qu'est-ce que Biopython ?

Biopython est une bibliothèque Python destinée à la bio-informatique. Elle permet de manipuler facilement des séquences d'**ADN**, d'**ARN** et de protéines. [Selon le *Biopython Tutorial & Cookbook*](#), elle fournit des outils standard pour lire des fichiers biologiques, analyser des séquences et interagir avec les bases de données comme **NCBI**.



1. 1 Pourquoi analyser le génome ?

3

D'après *Molecular Biology of the Gene (Watson et al.)*,

l'analyse génomique permet :

- d'identifier les gènes et les régions codantes.
- de détecter les mutations.
- de comparer des génomes entre espèces.
- de comprendre l'évolution et les maladies génétiques.
- d'extraire des informations fonctionnelles sur l'**ADN**.

1. 2 Fonctionnalités principales de Biopython.

4

- **Lecture des fichiers FASTA et GenBank:**

Avec le module SeqIO, il est possible de charger rapidement des séquences :

python

```
from Bio import SeqIO
record = SeqIO.read("genome.fasta", "fasta")
```

- **Analyse du contenu de la séquence:**

Biopython calcule la longueur, la teneur en GC (**guanine** + **cytosine**) et les cadres de lecture (ORFs).

```
from Bio.SeqUtils import gc_fraction

gc_fraction(record.seq)
```

1. 2 Fonctionnalités principales de Biopython.

5

- **Traduction ADN → Protéine :**

La fonction **translate()** permet d'obtenir la séquence protéique, essentielle pour étudier les fonctions biologiques.

```
protein = record.seq.translate()
```

python

- **Alignement des séquences :**

L'alignement, considéré par *Peuzner* comme essentiel en bio-informatique, se fait via **ClustalW** ou **MUSCLE** grâce à Biopython.

1. 2 Fonctionnalités principales de Biopython.

6

- **Recherche BLAST (NCBI) :**

Biopython permet d'envoyer des séquences directement à BLAST afin de trouver des similarités et prédire la fonction d'un gène.

python

```
from Bio.Blast import NCBIWWW
```

Biopython est un outil simple et puissant pour l'analyse génomique. Il permet d'importer, analyser, traduire et comparer des séquences, tout en offrant un accès direct aux bases de données biologiques. Grâce à sa flexibilité, il constitue un pilier de l'enseignement et de la recherche en bio-informatique.

2. Traitement des données avec EMBOSS

7

- **EMBOSS** (European Molecular Biology Open Software Suite) est une suite gratuite et open-source dédiée à la bio-informatique.

Elle permet l'analyse de séquences ADN, ARN et protéines.

Utilisée largement dans les laboratoires, universités et projets de génomique.

- **Pourquoi l'utiliser ?**

- Compatible avec presque tous les formats biologiques (FASTA, GenBank...).
- Offre plus de 200 outils pour : analyser des séquences, chercher des motifs, traduire de l'ADN, faire des alignements, extraire des informations

2.1 Traitement des données avec EMBOSS

8

Fonctions principales d'EMBOSS:

- **Fonctionnalités clés :**

- **Analyse des séquences** : longueur, composition, %GC.
- **Recherche de motifs** (motifs, sites CpG, répétitions...).
- **Traduction ADN → protéine.**
- **Alignement de séquences** pour comparer ADN/protéines.
- **Conversion de formats** (utile pour préparer les données).

- **Outils courants :**

- **infoseq** → informations sur une séquence
- **getorf** → recherche d'ORF (cadres de lecture)
- **transeq** → traduction d'une séquence
- **needle** → alignement global

2.2 Traitement des données avec EMBOSS

9

- **Exemple simple avec EMBOSS:**

Objectif :

Analyser une séquence ADN et afficher :

1. sa longueur.
2. son % de GC.
3. sa traduction en protéine.

- **Exemple de séquence (FASTA) :**

```
>Sequence_demo
```

```
ATGCGTACGTTAGC
```

shell

2.2 Traitement des données avec EMBOSS

10

- **Étapes :**

1. Obtenir les informations de base:

Commande :

```
$
```

```
infoseq sequence.fasta
```

```
nginx
```

Résultat attendu :

- Longueur : 14 bases
- Composition GC : ~57%

2.2 Traitement des données avec EMBOSS

11

2. Traduire l'ADN en protéine:

Commande :

```
transeq sequence.fasta protein.pep
```

Résultat (exemple):

MRTV

(**M** = Méthionine, **R** = Arginine, **T** = Thréonine, **V** = Valine)

3. Visualisation d'arbres phylogénétiques

12

- Définition :
 - Un arbre phylogénétique est une représentation graphique qui montre les relations évolutives entre différentes espèces, gènes ou séquences.
 - Les branches représentent l'évolution et les distances indiquent le degré de similarité ou divergence.
 - Utilisé en biologie moléculaire, génomique, évolution et classification des organismes

3. 2 Visualisation d'arbres phylogénétiques

13

- **Pourquoi visualiser ces arbres ?**
 - Comprendre l'origine d'une séquence ou d'un organisme.
 - Comparer des séquences ADN/ARN/protéines.
 - Identifier les ancêtres communs.
 - Vérifier les résultats d'alignements et d'analyses évolutives.
- **Outils courants de visualisation :**
 - **MEGA, FigTree, iTOL** (Interactive Tree Of Life).
 - **Biopython.Phylo** (visualisation programmée).

3. 3 Visualisation d'arbres phylogénétiques

14

- 1) Exemple d'un petit arbre (format **Newick**):
Format standard utilisé pour stocker les arbres :

```
(A, (B, C));
```

- A est une espèce/séquence isolée
- B et C partagent un ancêtre commun proche

(Réf. : MEGA & Newick Format Specification)

3. 4 Visualisation d'arbres phylogénétiques

15

2. Visualisation avec Biopython (exemple simple):

```
from Bio import Phylo
tree = Phylo.read("arbre.newick", "newick")
Phylo.draw(tree)
```

Ce code :

- ❖ lit un arbre au format **Newick**
- ❖ l'affiche graphiquement

4. Modélisation de l'expression génique

16

- **Définition:**

La modélisation de l'expression génique consiste à utiliser des modèles mathématiques ou informatiques pour décrire, prédire ou simuler comment un gène est exprimé (transcrit en ARN puis traduit en protéine)

- **Pourquoi modéliser ?**

- 1) Comprendre comment les gènes s'activent ou se désactivent.
- 2) Prédire la quantité d'ARN ou de protéines produites.
- 3) Étudier l'impact de facteurs externes (température, nutriments, stress...)

4. 1 Modélisation de l'expression génique

17

- **Étapes de l'expression génique:**

Transcription : $\text{ADN} \rightarrow \text{ARN}$

Traduction : $\text{ARN} \rightarrow \text{Protéine}$

Régulation : activation ou inhibition du gène.

- **Exemple simple**

Si un gène G s'active lorsqu'une molécule M est présente :

Beaucoup de $M \rightarrow$ expression forte du gène

Peu de $M \rightarrow$ expression faible

4. 2 Modélisation de l'expression génique

18

- **Types de modèles usuels :**

- 1) Modèles mathématiques simples:**

- équations linéaires pour estimer la quantité d'ARN.

- 2) Modèles dynamiques:**

- équations différentielles pour prédire la variation du niveau d'ARN/protéine dans le temps.

- 3) Modèles statistiques / Machine Learning:**

- prédiction de l'expression génique selon plusieurs facteurs (température, nutriments, mutations...).

5. Simulation de réseaux cellulaires avec COPASI

19

- COPASI (COmplex PAthway SIimulator) est un logiciel open source dédié à la modélisation et à la simulation des réseaux cellulaires et biochimiques. Il permet de représenter mathématiquement les interactions entre molécules (réactions enzymatiques, régulation génique, voies métaboliques) et d'analyser leur comportement dynamique à l'aide d'équations différentielles.

5. 1. Simulation de réseaux cellulaires avec COPASI

20

Intérêt et applications de COPASI

- COPASI offre des outils de simulation temporelle, d'analyse de stabilité et d'estimation de paramètres.

Il est largement utilisé en biologie systémique pour :

- comprendre le fonctionnement des réseaux cellulaires complexes.
- prédire l'évolution des concentrations moléculaires.
- tester des hypothèses biologiques sans expérimentation coûteuse.

6. Modélisation de la dynamique avec CellDesigner

21

- **CellDesigner** est un logiciel spécialisé dans la modélisation graphique et mathématique des réseaux biologiques.

Il permet de représenter de manière intuitive les interactions moléculaires (réseaux de signalisation, régulation génique, voies métaboliques) en utilisant des standards reconnus comme **SBML**

6.1. Modélisation de la dynamique avec CellDesigner

22

Intérêt et applications de CellDesigner

- CellDesigner facilite l'analyse de la dynamique des systèmes biologiques complexes en combinant visualisation et modélisation mathématique.

Il est utilisé pour :

- décrire et visualiser clairement les réseaux cellulaires,
- étudier la dynamique temporelle des interactions biologiques,
- servir de base à des simulations avancées via des outils comme COPASI.

7. Analyse intégrée des données multi-omiques avec Galaxy

23

- **Galaxy** est une plateforme web open source dédiée à l'analyse intégrée des données **multi-omiques** (génomique, transcriptomique, protéomique, métabolomique).

Elle permet d'exécuter des pipelines complexes de manière reproductible, sans nécessiter de compétences avancées en programmation, tout en assurant la traçabilité et le partage des analyses scientifiques.

8.1 Statistique et visualisation en R

24

- **R** est un langage de programmation et un environnement statistique open source, largement utilisé pour l'analyse de données, les statistiques avancées et la visualisation scientifique.

Il offre un large éventail de fonctions pour le traitement des données, la modélisation statistique et la production de graphiques de haute qualité grâce à des bibliothèques spécialisées telles que **ggplot2**, **dplyr** et **tidyr**.

6.2. Exemple de visualisation en R

25

- Dans cet exemple, **R** est utilisé pour analyser un jeu de données et représenter graphiquement la relation entre deux variables.

À l'aide de la bibliothèque **ggplot2**, il est possible de créer des graphiques clairs et interprétables (histogrammes, courbes, diagrammes de dispersion), facilitant ainsi l'exploration des données et la communication des résultats statistiques.

Fin cours