Bioinformatique avec Python et Biopython

Bioinformatique avec Python et Biopython

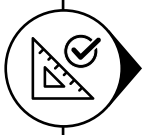
Emmanuel Jaspard
Université d'Angers

Gilles Hunault
Université d'Angers

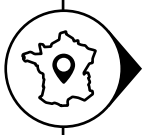
DUNOD

NOUS NOUS ENGAGEONS EN FAVEUR DE L'ENVIRONNEMENT :

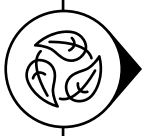
Nos livres sont imprimés sur des papiers certifiés pour réduire notre impact sur l'environnement.



Le format de nos ouvrages est pensé afin d'optimiser l'utilisation du papier.



Depuis plus de 30 ans, nous imprimons 70 % de nos livres en France et 25 % en Europe et nous mettons tout en œuvre pour augmenter cet engagement auprès des imprimeurs français.



Nous limitons l'utilisation du plastique sur nos ouvrages (film sur les couvertures et les livres).

11, rue Paul Bert, 92240 Malakoff
www.dunod.com
ISBN 978-2-10-085887-3

À Gwilherm Jaspard.

Pour Raphaël et Timothée.

Table des matières

Introduction

La bioinformatique est une discipline essentielle de la biologie. Il existe différentes acceptions du terme « bioinformatique » : elle peut être définie comme l'ensemble des théories et des méthodes issues de plusieurs disciplines, notamment l'informatique, dédiées aux traitements des données biologiques.

Les domaines dits en « omique » (dont la génomique, la métagénomique, la transcriptomique, la protéomique, la métabolomique, l'interactomique, la fluxomique...) nécessitent des outils bioinformatiques. Compte tenu de la très grande quantité de données biologiques acquises à une vitesse croissante et en regard de leur complexité, notamment la diversité des types de données, il est impossible d'en effectuer une analyse globale et exhaustive sans programme informatique.

Certains langages de programmation sont donc des outils indispensables pour l'acquisition des données biologiques et leur intégration dans des bases de données, la transformation des types de données, la fouille de données et leur classification, l'analyse (numérique, statistique, textuelle, visuelle...) des données selon des modèles mathématiques ou physiques et bien d'autres types de traitements.

Nous avons choisi `Python` et **`Biopython`** pour les raisons suivantes :

- Le langage `Python` est un langage orienté objet particulièrement utilisé en bioinformatique : c'est un langage évolué relativement simple à apprendre, qui est adopté par une proportion croissante de chercheurs pour développer des applications en bioinformatique, avec une multiplicité de bibliothèques et de scripts libres de droits...
- **`Biopython`** est un ensemble d'outils, librement disponibles et issus d'un effort collaboratif, pour développer des bibliothèques et des applications `Python` pour les besoins des travaux en bioinformatique.

La première partie de cet ouvrage a trait à l'analyse de certaines propriétés et caractéristiques des chaînes de nucléotides (**`ADN`** et **`ARN`**). La seconde partie traite de l'analyse textuelle (fouille de données de type chaînes de caractères) avec un outil remarquable, les expressions régulières. La troisième partie est dédiée à l'analyse des chaînes polypeptidiques et aux caractéristiques physico-chimiques des chaînes latérales des acides aminés. Elle illustre aussi l'alignement de séquences en utilisant la puissance des expressions régulières dans l'outil **`PHI-BLAST`**.

Nous explicitons dans cet ouvrage certaines interfaces de programmation **`Biopython`** (**`API`**) afin que chacun comprenne comment communiquer avec telle ou telle base de données et avec les interfaces *Web* des grands sites internationaux de bioinformatique.

Bon nombre de scripts équivalents aux scripts de cet ouvrage existent sous de multiples formes et sont accessibles librement dans des dépôts tels que **`GitHub`**. Même si nos scripts sont complets, fonctionnels et téléchargeables, le but principal de cet ouvrage n'est pas de fournir des scripts « clé en main », mais bien d'en comprendre la logique et

d'établir un lien avec le concept biologique afférent afin que le lecteur puisse développer ses propres scripts ou adapter ceux existants. Pour continuer l'analogie automobile, nous fournissons des scripts détaillés et expliqués pour que les lecteurs et lectrices aillent « voir sous le capot ».

Contenu type d'un chapitre

Tous les chapitres sont structurés de la même façon : on trouve d'abord un rappel sur les notions biologiques liées au sujet du chapitre. Ensuite, les objectifs du script sont présentés, avec un exemple de résultat de son exécution. Le code du script est alors fourni, accompagné d'explications et du code de l'auto-test utilisé pour vérifier les principaux cas d'exécution. Lorsque le script est un peu long, les sous-programmes (fonctions) développés pour le script sont listés et expliqués. Enfin, une synthèse du chapitre précède une courte bibliographie et une liste de sites et pages *Web* susceptibles d'éclairer ou d'approfondir le chapitre. En fin de chapitre, des questions complètent aussi bien les aspects biologiques que le code `Python` utilisé. Les réponses détaillées à ces questions sont disponibles après le chapitre 13.

Aucun des scripts ne fournit d'interface de saisie utilisateur, excepté au chapitre 1. Les scripts implémentent toujours des fonctions paramétrées. Le chapitre 1 décrit comment réaliser une interface minimale avec une lecture au clavier, sachant que ce n'est pas toujours la meilleure solution. Une interface *Web*, par exemple, ou une lecture dans un fichier (local ou sur Internet) sont parfois plus adaptées. C'est pourquoi nous fournissons dans chaque chapitre l'auto-test lié au script afin de savoir comment interfacer (au sens d'utiliser) les fonctions implémentées.

Progression pédagogique : du plus simple au plus compliqué

Dans le chapitre 1, nous nous intéressons à des calculs simples : dénombrement de nucléotides, calcul du pourcentage de `GC`, de la température de fusion **`Tm`**. Nous effectuons directement ces calculs en `Python` avant d'expliciter comment utiliser les fonctions équivalentes de **`Biopython`**. Nous montrons aussi comment profiter de l'auto-test *via* le test **`if __name__=="__main__"`** afin de tester et d'exposer le comportement des fonctions.

Le chapitre 2 traite de la lecture et de l'écriture de séquences d'acides nucléiques (**`ADN`** ou **`ARN`**) d'une séquence lue dans un fichier local.

Le chapitre 3 est consacré aux expressions régulières et à la recherche de motifs dans des fichiers au format **`FASTA`**.

Le chapitre 4 présente une utilisation typique de **`Biopython`** : on utilise les méthodes fournies par le package **`Entrez`** afin de trouver des articles dans la base de données bibliographiques du site **`PubMed`**. La connexion au serveur *Web* du **`NCBI`** et

la gestion des fichiers **XML** renvoyés y sont « transparentes » pour le développeur parce que le package implémente des structures de données et des méthodes qui les gèrent de façon automatique.

Le chapitre 5 présente les profils de familles de protéines et la recherche de familles **PFAM** sur le site **EBI-Interpro**. Le script commence par rechercher le numéro identifiant d'une famille de protéines à partir d'un mot-clé avant de rapatrier le profil associé.

Le chapitre 6 détaille le profil d'hydrophobicité d'une protéine dans une fenêtre glissante de taille choisie. Ce calcul est effectué une première fois en `Python` « pur », avant d'être réalisé par l'appel d'une fonction du package **Bio.SeqUtils.ProtParam** de **Biopython**. Les fichiers utilisés, au format **FASTA**, sont soit locaux, soit lus sur le site **Uniprot**.

Le chapitre 7 expose la recherche de la position de cystéines impliquées dans un ou plusieurs ponts disulfures pour un fichier structure d'une protéine lu dans la **PDB**.

Le chapitre 8 explicite le calcul du barycentre des carbones α d'une chaîne polypeptidique dont la structure est lue dans la **PDB**.

Les chapitres 9 à 13 illustrent l'analyse et la comparaison de structures de protéines :

- diagramme de Ramachandran (chapitre 9) ;
- matrice des distances et carte des contacts d'une structure **3D** pour une protéine de la **PDB** (chapitre 10) ;
- superposition de structures protéiques et calcul du score **RMSD** à l'aide du module **Bio.PDB.Superimposer** de **Biopython** (chapitre 11) ;
- calcul de l'indice de repliement nommé *Foldindex* d'un ensemble de séquences de protéines lues dans un fichier **FASTA** à partir du site `https://fold.proteopedia.org` (chapitre 12) ;
- alignement de séquences et recherche de protéines homologues à l'aide de **PHI-BLAST** *via* le module **NCBIWWW** de **Biopython** (chapitre 13).

Importance des questions en fin de chapitre

Chaque chapitre se termine par des questions dont les réponses sont fournies en fin d'ouvrage après le chapitre 13. L'ensemble des questions et réponses représente environ un cinquième de cet ouvrage : c'est un élément clé pour la réflexion du lecteur.

Ces questions et réponses constituent des rappels des notions développées dans chaque chapitre mais également des compléments et des illustrations. Elles permettent d'enrichir et de compléter les notions et les scripts tant du point de vue biologique qu'informatique. En effet, les scripts présentés dans le corps des chapitres sont courts et vont à l'essentiel.

Par exemple, au chapitre 1, pour obtenir un pourcentage, on effectue une division sans vérifier que le dénominateur n'est pas égal à zéro. Une des questions du chapitre 1

vient corriger ce problème et rappeler qu'il faut mettre systématiquement en place ce genre de vérifications.

De même, au chapitre 2, le script est censé lire une séquence dans un fichier dont le nom est passé en paramètre. Si le fichier n'existe pas, l'exécution du script aboutit à une erreur d'exécution. Ce comportement est volontaire et unique afin d'avoir un code `Python` court à lire. Une des questions du chapitre 2 a pour but de résoudre ce problème. Ensuite, tous les autres scripts qui utilisent un nom de fichier vérifient que le fichier existe en utilisant la technique indiquée dans la solution du chapitre 2.

Publics visés

Cet ouvrage s'adresse à tous les étudiants, chercheurs, ingénieurs qui désirent approfondir leurs connaissances dans l'analyse de données biologiques avec des outils `Python` et **`Biopython`**. Les notions requises en biologie sont essentiellement celles relatives à la composition et à la structure tridimensionnelle des acides nucléiques et des protéines (nucléotides, acides aminés, chaînes polypeptidiques).

Les étudiants et étudiantes de premier cycle y trouveront une aide à l'utilisation de `Python` pour la bioinformatique tandis qu'il pourra servir de manuel de référence pour les étudiants de cycle supérieur, pour les chercheurs et ingénieurs qui débutent en bioinformatique et qui ont besoin de modèles de scripts pour gérer les séquences, interfacer les sites internationaux de bioinformatique, parcourir les fichiers de séquences, de structures **`3D`** ou les fichiers de résultats de requêtes, au format **`JSON`** ou **`XML`**.

Enfin, le dernier public concerné est celui, de plus en plus nombreux, de nos collègues enseignantes et enseignants en bioinformatique car ils trouveront au fil des chapitres des sujets de T.P. avec corrigés prêts à l'emploi et faciles à adapter ou compléter.

Prérequis

La compréhension des scripts nécessite des connaissances élémentaires des instructions et fonctions du langage `Python`. Ces notions doivent également inclure les tableaux, les dictionnaires, les expressions régulières, les imports de packages et les appels de méthodes objets. Un rappel de ces connaissances `Python` se trouve dans les annexes, page 183, après le chapitre 13. Une présentation détaillée des expressions régulières et leur utilisation en `Python` est aussi fournie en annexe, page 191.

Aucune connaissance de **`Biopython`** n'est requise. Une présentation de **`Biopython`** est fournie en annexe (page 197). La liste des scripts écrits, avec leurs dépendances et la liste des modules et packages `Python` et **`Biopython`** utilisés sont également présentes dans cette annexe, respectivement pages 199 et 203.