Traitement de données biologiques complexes avec Python

Cours

Prérequis

- H2, BIF-1001 : introduction à la bio-informatique

ET

- A1, IFT-1004 : introduction à la programmation
- **Ou**
- GLO-1901 : introduction à la programmation avec python

Les deux cours IFT-1004 et GLO-1901 sont donnés en python

Recommandés

- H4, BIF-4004 : génomique computationnelle
- A5, IFT-2004 : modèle et langages des bases de données

Positionnement

Le cours serait donné à H6 pour le programme de baccalauréat bioinformatique (prochains : H2019-H2020)

Informations générales sur le cours

- Fait en Python 3.6
- Travaux réalisés avec des jeux de données biologiques réels provenant des bases de données biologiques connues (GenBank, Uniprot, ...)

Base de connaissances demandée

Sont considérés connus :

- Aspect conceptuel
  - La décomposition fonctionnelle d'un problème
  - La notion d'objet : encapsulation, héritage, polymorphisme
  - La notion de complexité des algorithmes
  - La notion d'exception
  - La notion de tests unitaires
- Aspect technique
  - La manipulation de base d'un système d'exploitation Unix/Linux
  - La syntaxe du langage Python
  - L'architecture d'un programme
  - L'utilisation de modules (import)
  - La création de conditions, de boucles, de fonctions et de classes (if/elif/else, for, while, def, class)
  - Le mécanisme d'itération
  - La gestion des exceptions (try/except)
  - La documentation du code

Ces bases seront abordées rapidement lors du premier cours

Objectifs

Le but de ce cours est de permettre l'analyse de données massives biologique grâce à Python. Entre autres, ce cours permet d'approfondir les connaissances déjà acquises, de développer de bonnes pratiques pour améliorer l'efficacité du code et diminuer l'impact sur l'utilisation des ressources.

Les points abordés ci-après permettront d'améliorer autant le savoir que le savoir-faire de l'étudiant par la pratique dans un cadre applicatif orienté sur les données massives biologiques.

Vue d'ensemble

- Comprendre ce que sont les données massives (Big Data) en biologie
- Manipuler efficacement les données et les outils
- Optimiser les traitements et l'utilisation des ressources

Vue détaillée

1. Accéder aux données
    - Manipuler un fichier csv/tsv/json/gff3/fasta/...
    - Manipuler les bases de données
    - Gestion des exceptions et validation des données (checksum, schémas, etc.)
2. Traiter/manipuler les données biologiques
    - Connaitre/employer les bonnes bibliothèques logicielles
    - Estimer les temps et coûts des traitement (complexité des algorithmes, tqdm, ...)
    - Savoir répondre à la question : quelle bibliothèque utiliser dans quel contexte ?
    - Construire un pipeline
    - Visualisation des données sans interface graphique : matplotlib
3. Raffiner l'algorithme et bibliothèques utilisées
    - Utiliser les structures de données (dict, list, set, tuple, ordereddict, namedtuple,)
    - Utiliser les générateurs, itertools, lambda, décorateurs, énumérations (Enum)
    - Manipuler la compression/décompression de données
    - Parallélisation des taches : multithreading, multiprocessing
    - Utilisation de buffers : téléchargement, écriture fichier, etc.
    - Surcharger les opérateurs Object : __str__, __init__, __lg/le/gt/ge/ne/eq__, ...

   - Interagir dynamiquement avec une classe : hasattr, getattr et setattr

4. Manipuler les lib de data science et de visualisation
   - Calcul scientifique : numpy, scipy
   - Manipulation de données et matrices : pandas
   - Datamining et extraction de pattern grâce au machine learning : Tensorflow, pytorch, scikit-learn
   - Visualisation des résultats : matplotlib, bokeh, plotly

Plan

Cours 1 : Rappel des bases de GLO-1901

- Variable et affectation
- String manipulation et format mini-langage
- Import de modules
- Structure de données de base
  - List/dict/set/tuple
  - List/dict comprehension
  - Slice notation
- Notion objet
  - Héritage et polymorphisme
  - Surcharge d'operateurs
- Lambda
- Transtypage
- Nouveautés majeures de python3.6
  - Notions abordées plus tard mais introduites ici
  - PEP 526 : variable annotation
  - PEP 498 & PEP 515 : f-string, and underscore in numeric literals
  - PEP 525 & PEP 530: asyncio stable, async generators & async comprehensions

Cours 2 : Manipuler un fichier

- Ouvrir un fichier (open) + mode + close
- Par ordre de difficulté croissant :
  - Json
  - Csv/tsv
  - Fasta/gff3/XML
- Méthodes de parcours
  - For line in file
  - While 1 -> fin de fichier
    - Iter () + next ()
    - Exception StopIteration
- Content management (with)
- Compresser des données

Cours 3 : Interroger une base de données

- Travail sur db
  - Petite db SQL : sqlite3
  - Petite db NoSQL : Berkeley dB ?
  - Notion d'API
- Builts-in methods : any, all
- Envoyer des données
- Notion d'asynchronicité, syn vs asyn
- Algorithmes de compression pour envoi (msgpack) ?

Cours 4 : Faciliter l'accès aux ressources avec BioPython

- Documentation en ligne simple et détaillée
- Revoir les notions d'ouverture de fichier avec cette bibliothèque (Module SeqIO)

- Ouvrir une base de données avec les drivers BioPython (SQLite3)
- Utiliser les scripts de la communauté disponibles sur le site: ScriptCentral
- Lire une séquence depuis différents format (FASTA/FASTQ, GenBank, SwissProt...) et la ranger dans une structure adaptée (list, dictionnaire, tuple ...)
- Manipuler une séquence et ses transformations les plus classiques (reverse, complément, recherche de motif, transcription/traduction avec la norme IUPAC, ...)
- Aligner des séquences (simple ou multiple) selon les algorithmes MUSCLE, ClustalW et pairwise2. Faire appel à un alignement BLAST

Cours 5 : Calcul scientifique avec Numpy et Scipy

- Manipulation de tableaux multidimensionnels (matrices)
- Numpy et Scipy

Cours 6 : Opérations matricielles de haut niveau avec Pandas

- Matrice d'alignement de séquences
- Matrices de distances pour clustering

Cours 7 : Calcul statistique avec StatsModels

- StatsModels
- Génétique des populations

Cours 8 : Mini-projet individuel

- Application de toutes les notions vues
- Temps à déterminer
- Jeu de données réel

Cours 9 : Limites des ressources

- Reprise des exercices déjà vus avec de très gros fichiers (plusieurs Go) et de multiples sources (base de données, fichiers, etc.)

- Montrer l'impossibilité de traitement sans cluster de calcul avec les méthodes classiques
- Utiliser les structures de données pertinentes
- Utiliser les outils générateurs, itertools, lambda, décorateurs, énumérations
- Alignement matching
- Être critique sur les données à utiliser

Cours 10 : Limites des ressources 2

- Manipuler la compression/décompression de données
- Paralléliser ses calculs
- Utilisation de buffers : téléchargement, écriture fichier, etc.

Cours 11 : Visualisation de données

- Test des deux niveaux de bibliothèques :
  - matplotlib: graphiques 2D, interface similaire à MATLAB
  - bokeh et plotly: graphiques interactifs pour le web

Cours 12 : Machine learning et réseau de neurones

- Outils de machine learning et de réseaux de neurones :
  - scikit-learn
  - Tensorflow
  - Pytorch

Idées

- Cheatsheets a la fin de chaque cours pour résumer les notions vues