|  |  |
| --- | --- |
|  |  |

**Stage Python pour biologistes : programme détaillé**

La formation alternera cours et TP. Les stagiaires se familiariseront avec chaque notion présentée en créant ou manipulant des codes.

Le programme ainsi que le découpage des journées pourra évoluer en fonction du public.

**Jour 1 : Présentation non approfondie du Langage Python**

Les stagiaires manipuleront le langage à travers l'interprète puis via des scripts.

L'objectif est de pouvoir écrire des scripts simples sur un fichier à la fin de la journée.

- Historique

- Python comme Interprète de commande

- Types de base, opérateurs : présentation et manipulation

- Aiguillages

- Hello world

- Premiers scripts

- Listes

- Boucles

- Fonctions

- Utilisation de librairies

- Introduction au entrées/sorties

**Jour 2 : Présentations de Git**

Les stagiaires, qui peuvent désormais écrire des scripts simples, les manipuleront de façon collaborative à l'aide de Git.

- Présentations des principaux types de systèmes de versionnage

- Création et utilisation d'un dépôt local

- Utilisation comme système de préservation d'historique

- Introduction aux branches

- Clonage et synchronisation avec un dépôt existant

- Mode multi-joueur

- Présentations des bonnes pratiques pour des collaborations harmonieuses

- Avoir un compte sur GitHub

- Authentification

- Gestion des clefs

- Requêtes de merge

- Discussions

**Jour 3 : Python "avancé"**

Les fonctionnalités que les stagiaires seront amenées à rencontrer seront présentées de l’intérieurs. Ceci afin qu'ils puissent avoir un certain recul par rapport à des fonctionnalités même s'ils ne seront pas amenés à les mettre en œuvre. Les exercices seront toujours effectués à travers Git et de façon collaborative afin que les stagiaires se familiarisent avec ce mode de travail.

- Structures de données

- Listes avancées

- Ensembles, dictionnaire, tuple

- Boucles avancées

- Modules

- Exceptions

- Classes

- Définition, attributs, méthodes

- Itération

**Jour 4 : Le module Biopython**

L'objectif pour les stagiaires est de manipuler des séquences et de fichiers de séquences à l'aide du module BioPython et d'adapter le traitement suivant le type d'analyse

- L'objet Seq (Bio.Seq)

- Séquences nucléotidique et protéique, reverse complément, Transcription,Translation (TP en relation avec le chapitre)

- L'objet SeqRecord (Bio.SeqRecord)

- FASTA-Genbank, Features, Positions and Location (TP en relation avec le chapitre)

- Alignement/mapping de séquences (Bio.AlignIO)

- Utilisation de BLAST, ClustalW (TP en relation avec le chapitre)

- Parser et lire, convertir des séquences (Bio.SeqIO) (Récupération des données nécessaires à l'analyse (TP en relation avec le chapitre)

- À partir d'un fichier ou des bases de données en ligne (Entry NCBI database (Bio.Entrez) et Swiss-prot database (Bio.SwissProt))

- Nettoyage et filtrage des fichiers fastq

- Conversion de format (fastq/fasta)

**2 Idées de TP**

1- Aligner séquences protéiques, faire un bootstrapping, trouver matrices de distances et créer un arbre phylogénétique en utilisant le neighbor-joining.

2- Récupérer des résultats chip-seq ou micro-array (fasta) matchant un motif particulier, retrouver les régions en amont de la séquence correspondant au motif, et chercher pour des motifs conservés.