Abderrahim Tabta
Niko Kwok Hiu Fung

5 décembre 2022
TP3

1. Résumé sur l'implémentation

Nous avons programmé PLAST dans le langage Python. Toute la logique se trouve dans le fichier **plast.py**. Cela dit, il est important que le fichier **FastaFile.py** soit dans le même projet puisque ce code gère l'ouverture des fichiers fasta. Enfin, l'installation de la librairie externe numpy est nécessaire pour le bon fonctionnement du programme.

Nous avons utilisé l'IDE Pycharm pour écrire notre code. Vous pouvez l'utiliser aussi pour compiler le code ou lire nos commentaires. Sinon, il suffit d'exécuter une commande comme la suivante dans le dossier du projet :

```
python plast.py -i "CGTAGTCGGCTAACGCATACGCTTGATAAGCGTAAGAGCC" -db "tRNAs.fasta" -E 5
```

Le résultat à cette commande sur notre console est le suivant :

```
C:\Users\leade\Desktop\IFT3295-TP1\Pycharm Project>python plast.py -i "CGTAGTCGGCTAACGCATACGCTTGATAAGCGTAAGAGCC" -db "tRNAs.fasta" -E 5
I|gat|Mesostigma_viride
# Best HSP score: 78.0 , bitscore:  24 , evalue:  0.00017642974853515625
13 CGCATACGCTTGATAAGCGTA 33
23 AGTATACGCCTGATAAGCGTA 43

--------------------------------------
Total : 1

C:\Users\leade\Desktop\IFT3295-TP1\Pycharm Project>
```

Ce n'est pas demandé dans l'énoncé, mais il est aussi possible d'activer le mode debug de notre code en modifiant la ligne 247 de plast.py :

```
244    ##########################################################{ TEST }##########################################################
245
246     # fasta tests
247    debug = False                                                #  TODO SET TO TRUE TO DEBUG !!!
```

Cela permet par exemple d'exécuter le programme plast sur chacune des séquences dans le fichier **unknown.fasta**. Biensûr, il n'est pas nécessaire de le faire puisque nous allons expliciter tous nos résultats dans la suite du rapport. Voici à quoi ressemble le debug :

```
------------------------------ TEST 1.1
Ouverture du fichier fasta et affichage de la première séquence :
R|tct|Mesostigma_viride
ACATTCTTAGCTCAGTTGGATAGAGCAACGGCCTTCTAAGCTGTAGGTCACAGGTTCAAATCCTGTAGAATGTA

------------------------------ TEST 1.2
Génération de kmers de seed de taille 11 de cette première séquence:
['ACATTCTTAGC', 'CATTCTTAGCT', 'ATTCTTAGCTC', 'TTCTTAGCTCA', 'TCTTAGCTCAG', 'CTTAGCTCAGT', 'TTAGCTCAGTT', 'TAGCTCAGTTG

------------------------------ TEST 1.3
Recherche de la séquence (input) dans le fichier fasta:
Input: CGTAGTCGGCTAACGCATACGCTTGATAAGCGTAAGAGCC
Kmer:  TGATAAGCGTA
Séquence fasta:  GGGCCTATAACTCAGTTGGTTAGAGTATACGCCTGATAAGCGTATTGTCAGTAGTTCAAATCTGCTTGGGCCCA
Indexe de kmer dans fasta:  [33]

------------------------------ TEST 1.4
Extension de la séquence voulue dans une séquence fasta:
([kmerIndex, wordIndex, wordOfKmerStart, wordOfKmerEnd, wordStart, wordEnd])
[[23, 6, 13, 33, 23, 43, 78.0]]
kmer:  TGATAAGCGTA
full kmer:  CGTAGTCGGCTAACGCATACGCTTGATAAGCGTAAGAGCC
fasta seq:  GGGCCTATAACTCAGTTGGTTAGAGTATACGCCTGATAAGCGTATTGTCAGTAGTTCAAATCTGCTTGGGCCCA
-------
Score:  78.0
I|gat|Mesostigma_viride
13 CGCATACGCTTGATAAGCGTA 33
23 AGTATACGCCTGATAAGCGTA 43

------------------------------ TEST 1.5
Algorithme complet sur toutes les séquences dans unknown.fasta:
M|cat|Carica_papaya
# Best HSP score: 256.0 , bitscore:  73 , evalue:  5.797940523955686e-19
21 TACTCATCAGGCTCATGACCTGAAGACTGCAGGTTCGAATCCTGTCCCCGCCT 73
```

2. Questions sur la mise en pratique

1. Donnez l'output de votre programme pour chacune des séquences du fichier *unknown.fasta* qui contient des séquences d'ARNts dont la nature est inconnue. On vous demande d'utiliser les paramètres par défaut (-ss = 0.001, -E = 4, seed = '11111111111').

M|cat|Carica_papaya

Best HSP score: 256.0 , bitscore: 73 , evalue: 5.797940523955686e-19

21 TACTCATCAGGCTCATGACCTGAAGACTGCAGGTTCGAATCCTGTCCCCGCCT 73

21 GACTCATCAGGCTCATGACCTGAAGACTGCAGGTTCGAATCCTGTCCCCGCCT 73

R|tcg|Marchantia_polymorpha

Best HSP score: 297.0 , bitscore: 85 , evalue: 1.4155128232313686e-22

0 ACATCCTTAGCTCAGTAGGATAGAGCAACAGCCTTCTAAGCTGGTGGTCACAGGTTCAAATCCTGTAGGATG 71

0 GCATTCTTAGCTCA

GTTGGATAGAGCAACAACCTTCGAAGTTGATGGTCACAGGTTCAAATCCTGTAGGATG 71

P|tgg|Oryza_sativa_Japonica_Group

Best HSP score: 137.0 , bitscore: 40 , evalue: 4.980392986908555e-09

43 GAGGTCACGGGTTCAAATCCTGTCATCCCTA 73

43 AATGTCACGGGTTCAAATCCTGTCATCCCTA 73

R|tct|Marchantia_polymorpha

Best HSP score: 347.0 , bitscore: 99 , evalue: 8.522853220007812e-27

0 GCATTCTTAGCTCAGCTGGATAGAGCAACAACCTTCTAAGTTGAAGGTCACAGGTTCAAATCCTGTAGGATGC 72

0 GCATTCTTAGCTCAGTTGGATAGAGCAACAACCTTCTAAGTTGAAGGTCACAGGTTCAAATCCTGTAGAATGC 72

Abderrahim Tabta
Niko Kwok Hiu Fung

5 décembre 2022
TP3

Image de la console :

```
M|cat|Carica_papaya
# Best HSP score: 256.0 , bitscore:  73 , evalue:  5.7979405239556686e-19
21 TACTCATCAGGCTCATGACCTGAAGACTGCAGGTTCGAATCCTGTCCCCGCCT 73
21 GACTCATCAGGCTCATGACCTGAAGACTGCAGGTTCGAATCCTGTCCCCGCCT 73

R|tcg|Marchantia_polymorpha
# Best HSP score: 297.0 , bitscore:  85 , evalue:  1.4155128232313686e-22
0 ACATCCTTAGCTCAGTAGGATAGAGCAACAGCCTTCTAAGCTGGTGGTCACAGGTTCAAATCCTGTAGGATG 71
0 GCATTCTTAGCTCAGTTGGATAGAGCAACAACCTTCGAAGTTGATGGTCACAGGTTCAAATCCTGTAGGATG 71

P|tgg|Oryza_sativa_Japonica_Group
# Best HSP score: 137.0 , bitscore:  40 , evalue:  4.980392986908555e-09
43 GAGGTCACGGGTTCAAATCCTGTCATCCCTA 73
43 AATGTCACGGGTTCAAATCCTGTCATCCCTA 73

R|tct|Marchantia_polymorpha
# Best HSP score: 347.0 , bitscore:  99 , evalue:  8.522853220007812e-27
0 GCATTCTTAGCTCAGCTGGATAGAGCAACAACCTTCTAAGTTGAAGGTCACAGGTTCAAATCCTGTAGGATGC 72
0 GCATTCTTAGCTCAGTTGGATAGAGCAACAACCTTCTAAGTTGAAGGTCACAGGTTCAAATCCTGTAGAATGC 72
```

2. En déduire, **si possible** la nature de chacune des séquences. Notez que l'identifiant des séquences de la banque de données est sous la forme : Amino-Acide|Anticodon|Espèce. Vous pouvez au besoin jouer avec le seuil de signification (-ss) pour affiner votre recherche.

Première séquence : Amino-Acide : **M** | Anticodon : **cat**

Deuxième séquence : Amino-Acide : **R** | Anticodon : **tcg**

Troisième séquence : Amino-Acide : **P** | Anticodon : **tgg**

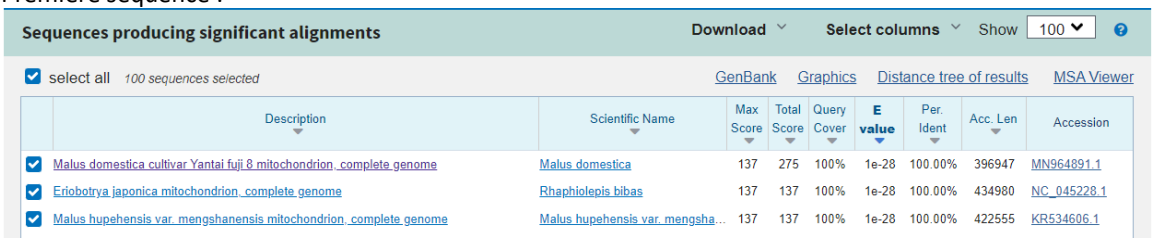
Dernière séquence : Amino-Acide : **R** | Anticodon : **tct**

Abderrahim Tabta
Niko Kwok Hiu Fung

5 décembre 2022
TP3

3. Vérifiez vos résultats en vous servant du véritable outil BLAST pour nucléotide (BLASTN) disponible sur NCBI ==> `https://blast.ncbi.nlm.nih.gov/Blast.cgi` . On vous demande de comparez les deux résultats.

Première séquence :



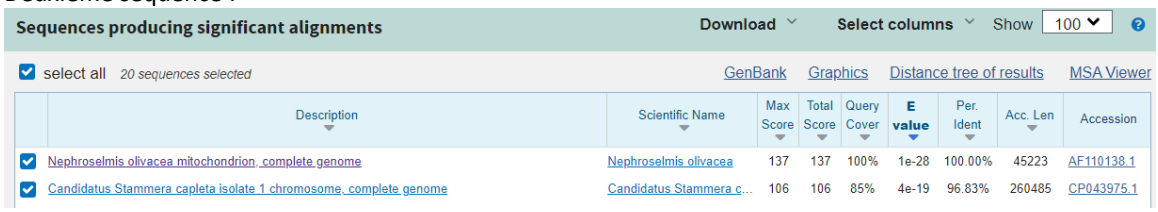
Sequences producing significant alignments — Download — Select columns — Show 100

select all *100 sequences selected* — GenBank — Graphics — Distance tree of results — MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Malus domestica cultivar Yantai fuji 8 mitochondrion, complete genome	Malus domestica	137	275	100%	1e-28	100.00%	396947	MN964891.1
Eriobotrya japonica mitochondrion, complete genome	Rhaphiolepis bibas	137	137	100%	1e-28	100.00%	434980	NC_045228.1
Malus hupehensis var. mengshanensis mitochondrion, complete genome	Malus hupehensis var. mengsha...	137	137	100%	1e-28	100.00%	422555	KR534606.1

Nous avons obtenu un max score beaucoup trop grand comparé au score du véritable blast (256>137).

Deuxième séquence :



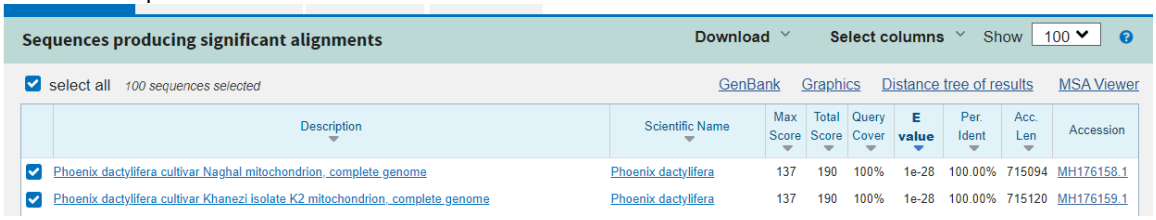
Sequences producing significant alignments — Download — Select columns — Show 100

select all *20 sequences selected* — GenBank — Graphics — Distance tree of results — MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Nephroselmis olivacea mitochondrion, complete genome	Nephroselmis olivacea	137	137	100%	1e-28	100.00%	45223	AF110138.1
Candidatus Stammera capleta isolate 1 chromosome, complete genome	Candidatus Stammera c...	106	106	85%	4e-19	96.83%	260485	CP043975.1

Encore une fois, le max score est beaucoup moins élevé que ce que nous avons obtenu (297>137)

Troisième séquence :



Sequences producing significant alignments — Download — Select columns — Show 100

select all *100 sequences selected* — GenBank — Graphics — Distance tree of results — MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Phoenix dactylifera cultivar Naghal mitochondrion, complete genome	Phoenix dactylifera	137	190	100%	1e-28	100.00%	715094	MH176158.1
Phoenix dactylifera cultivar Khanezi isolate K2 mitochondrion, complete genome	Phoenix dactylifera	137	190	100%	1e-28	100.00%	715120	MH176159.1

Ici, nous avons obtenu le même max score (137). Cela dit, la E Value est différente.

Dernière séquence :

Sequences producing significant alignments — Download — Select columns — Show 100

select all *100 sequences selected* — GenBank — Graphics — Distance tree of results — MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Nitella hyalina mitochondrion, complete genome	Nitella hyalina	137	137	100%	1e-28	100.00%	80193	NC_017598.1
Chara braunii S276 mitochondrial DNA, complete sequence	Chara braunii	137	137	100%	1e-28	100.00%	67059	AP018556.1

Ici, notre score maximum est beaucoup trop élevé comparé au vrais score (347>137)

4. **Bonus :** Qu'arrive t'il lorsque vous utilisez des graines plus longues ? plus courtes ? (on vous demande l'impact sur la vitesse, la précision et la sensibilité de PLAST)

Suites à plusieurs tests sur notre programme, nous avons remarquer que l'exécution du programme est beaucoup plus lente avec des seeds plus court. En d'autres mots, réduire la taille des seed peut nuire aux performances (**vitesse**) du programme. Tandis qu'utiliser des seed plus grand améliore la performance du programme. Cela dit, il est raisonnable de croire qu'en utilisant des seeds trop grand, nous pouvons manquer certaines occurrences, mais ce n'est pas nécessairement négatif, puisque nous allons retenir seulement les sous-séquences les plus longues. Donc, cela peut améliorer la **précision** du programme.