Table des matières

Table des figures

1 Introduction

Le but de ce document est de présenter les différents outils développés, pour permettre à un utilisateur externe aux projet de pouvoir l'utiliser. La présentation se fera dans l'ordre chronologique des développements, on commencera tout d'abord par la construction de la base de données en locale, puis par le comptage en fréquence de kmers et on terminera par la validation croisée.

2 Espace de travail et prérequis

On suppose que l'espace de travail par défaut est celle qui contient les dossiers présentés en figure :

```
  $ ls -1
count_kmer/  cpp/  create_db/  data_mining/  docs/  filter/ generate_data/  generate_graph/
generate_learn/  HOWTO/  qwery/ REQUIRED
```

FIGURE 1 – Espace de travail par défaut

- qwery/ : Voir section 3.1,
- create_db/ : voir section 3.2,
- generate_data/ : voir section 3.3,
- cpp : voir section 4.1,
- generate_graph : voir seciton 5.1.

3 Construction de la base de données locale

3.1 Récupération du Genbank

3.1.1 Prérequis

Dans l'espace de travail par défaut se trouve un fichier REQUIRED, qui un script d'installation de paquet Debian nécessaire pour l'utilisation des scripts perl, voir figure 2

```
$ ls -1
count_kmer/  cpp/  create_db/  data_mining/  docs/  filter/ generate_data/  generate_graph/
generate_learn/  HOWTO/  qwery/ REQUIRED

$ sh REQUIRED
```

FIGURE 2 – Installation des bibliothèque perl

On peut à présent se placer dans le dossier qwery pour effectuer notre requete.

```
$ cd qwery/

$ ls
qweryNCBI.pl
```

3.1.2 Résultats attendus

On s'attend ici à récupérer un fichier au format Genbank du ncbi [1] des génomes mitochondriaux complet.

La requête exécutée par défaut si l'id = 2759 est :

```
txid2759[Organism:exp] AND (mitochondria[Title] OR mitochondrion[Title] OR mitochondrial[Title])
```

3.1.3 Utilisation

Les options

- Obligatoires :
  - id : taxid du taxon à récupérer (-id 2759)
  - m : email de l'utilisateur (-m myemail@mail.com)
  - out : nom de sortie sans extension (-out eukaryota)
- Optionelles :
  - path : dossier où sauvegarder le genbank (/home/me/bdd/)
  - not : liste d'id à ne pas récupérer (-not 33630,33258)

1. http ://www.ncbi.nlm.nih.gov/

- more : ajouter des précisions à la requêtes par défaut (-more "NOT HOMO")
- mine : spécifier sa propre requête (-mine txid2759[Organism :exp])
- help : Affiche l'aide.

3.1.4 Exemples

Utilisation basique voir figure 7 :

```
$ ls
qweryNCBI.pl

$ ./qweryNCBI.pl -id 2759 -m toto@mail.com -out eukaryota

$ ls
qweryNCBI.pl eukaryota.gb
```

FIGURE 3 – Utilisation minimale de qweryNCBI.pl

3.2 Création des dossier

```
$ cd create_db
$ ls
bdd/ Eukaryota_krona.html  generateDirectories.pl
get_dump_file.sh  get_leaf.sh  install_krona.sh
```

3.2.1 Prérequis

Un fichier genbank est nécessaire pour la création de la base de donnée. Les requêtes sont par défaut effectuées en locale il est donc d'avoir la structure arborescente du NCBI dans des fchiers plats, ces fichiers sont versionnés sur la forge et sont présent dans create_db/bdd/, mais on peut également les récupérer ou mettre à jour via le script *get_ dump_ file.sh*

```
$ sh get_dump_file.sh
```

3.2.2 Résultats attendus

Le script produit une base de donnée avec une structure arborescente de dossier. Il produit également, la structure de la base dans un format newick, un fichier xml au format adéquat pour l'outil *krona*, et un script pour le nettoyage de la base.

Le script *install_krona.sh* est prévu pour l'installation de krona, si on souhaite l'utiliser voir figure 4.

Enfin il va générer deux fichier dans le dossier generate_data pour la génération des données pour l'étape 3.3.

```
14:47:09 [jeremy][create_db]$ ./install_krona.sh

**********************
Debut install krona
**********************


Cloning into 'krona'...
remote: Counting objects: 788, done.
remote: Compressing objects: 100% (598/598), done.
remote: Total 788 (delta 364), reused 305 (delta 149)
Receiving objects: 100% (788/788), 528.43 KiB | 395 KiB/s, done.
Resolving deltas: 100% (364/364), done.
Creating links...

Installation complete.

To use scripts that rely on NCBI taxonomy, run updateTaxonomy.sh to build the
local taxonomy database.


**************************
Fin install krona, exemple
**************************


**********************
ktImportXML krona_Eukaryota.xml
**********************


14:47:19 [jeremy][create_db]$
```

FIGURE 4 – Installation de krona

3.2.3 Utilisation

Les options

- Obligatoires :
  - id : taxid du taxon (-id 2759)
  - gen : fichier genbank (-gen file.gb)
  - bound : nombre de sequence pour creer un dossier (-bound 10)
- Optionelles :
  - path : dossier où créer la base de donnée (/home/me/bdd/)
  - time : temps pris pour la création de la base (-time)
  - help : Affiche l'aide.

3.2.4 Exemples

Utilsation de base voir figure 5 :

```
$ ls
bdd/  Eukaryota_krona.html  generateDirectories.pl  get_dump_file.sh
get_leaf.sh  install_krona.sh

$ ls ../generate_data
conf  extractGenbank.pl  fillAll_v2.sh

$ ./generateDirectories.pl -id 33630 -gen ../qwery/alveolata.gb -bound 10

$ ls
Alveolata__33630/  bdd/  Eukaryota_krona.html  generateDirectories.pl  get_dump_file.sh
get_leaf.sh  install_krona.sh  krona_Alveolata.xml  script_clean_Alveolata.sh
tree_Alveolata.newick

$ ls ../generate_data
conf  extractGenbank.pl  fillAll_v2.sh generateGenbank_Alveolata.sh  listGenbank.txt
```

FIGURE 5 – Utilisation minimale de generateDirectories.pl

3.3 Création des dossier

```
$ cd generate_data
```

```
$ ls
conf  extractGenbank.pl  fillAll_v2.sh generateGenbank_Alveolata.sh  listGenbank.txt
```

3.3.1 Prérequis

Le fichier listGenbank, généré grâce au script présenté en 3.2 est nécessaire pour la génération des données aux bons endroits. En effet ce fichier contient des listes d'accessions propre à un dossier.

3.3.2 Résultats attendus

On s'attend, après avoir exécuté ce scripts, l'ensemble des données dans la base. C'est à dire les genbanks correspondants, les génomes, les séquences codantes (selon le fichier conf)...

3.3.3 Utilisation, exemple

```
$ sh generateGenbank_Alveolata.sh
```

FIGURE 6 – Génération des données

Attention, ce script produit un affichage sur stdout montrant l'évolution du script dans ces trois étapes :

```
Traitement 1 / 3
Avancement: 100 %
Traitement 2 / 3 (Génération des donnees aux feuilles)
Traitement 3/3 (linkage)
```

4 Fréquence de kmers

4.1 Calcul de fréquence en c++

```
$ cd cpp/count_kmer/

$ ls
qweryNCBI.pl
```

Il existe également une version de calcul de fréquence de kmer en C, mais pour le projet on a opté pour le c++ pour son côté objet. Les versions développés en Perl, Java et C ont servi de test pour la version c++.

4.1.1 Utilisation

Les options

- *listFasta* fichier contenant une liste de chemins vers des fichiers fasta,
- *fasta* fichier fasta,
- *wsize* taille du read,
- *noData* libérer la mémoire suite au chargement des données,
- *kmer* fichier contenant les patterns de kmers,
- *output* nom du fichier de sortie,
- *version* version du programme,
- *test* lance une batterie de tests unitaires,
- *intra* lance un test sur le taxon Intramacronucleata,
- *key* sur quels séquences compter : cox1,cox2,genomes...,
- *root* racine où on doit établir l'apprenstissage,
- *jump* taille du décalage de la fenetre,
- *learn* taille de l'apprentissage pour la validation croissée,
- *start* debut de la taille du read dans la prédiction pour la validation croissée,
- *end* fin de la taille du read dans la prédiction pour la validation croissée,
- *step* taille du pas pour les différentes fenetre pour la prédiction,
- *sample* nombre de sequences à considérer par taxon,
- *weka* imprime sur la sortie le format weka,
- *help* affiche l'aide.

4.1.2 Exemples

```
$ makefile

$ ./count_kmer -f seq.fasta -k pattern.txt -o out.arff -l -1 --weka

$ cat out.arff
@RELATION freqKmer

@ATTRIBUTE aaaa NUMERIC
@ATTRIBUTE aaac NUMERIC
@ATTRIBUTE aaag NUMERIC
@ATTRIBUTE aaat NUMERIC
...
@ATTRIBUTE tttg NUMERIC
@ATTRIBUTE tttt NUMERIC
@ATTRIBUTE class {null}

@DATA
0.25,0,...,null %Datat(0,0)
```

FIGURE 7 – Produit un fichier weka avec count_kmer

5 Validation croisée

5.1 Validation croisée à un niveau taxonomique

5.1.1 Prérequis

5.1.2 Résultats attendus

5.1.3 Utilisation

5.1.4 Exemples