

MANUEL D'UTILISATION

Auteur : Jérémy FONTAINE

Table des matières

1	Introduction	3
2	Espace de travail et prérequis	4
3	Construction de la base de données locale	5
3.1	Récupération du Genbank	5
3.1.1	Prérequis	5
3.1.2	Résultats attendus	5
3.1.3	Utilisation	5
3.1.4	Exemples	6
3.2	Création des dossier	6
3.2.1	Prérequis	6
3.2.2	Résultats attendus	6
3.2.3	Utilisation	8
3.2.4	Exemples	8
3.3	Création des dossier	9
3.3.1	Prérequis	9
3.3.2	Résultats attendus	9
3.3.3	Utilisation, exemple	9
4	Fréquence de kmers	10
4.1	Calcul de fréquence en c++	10
4.1.1	Utilisation	10
4.1.2	Exemples	10
5	Validation croisée	13
5.1	Validation croisée à un niveau taxonomique	13
5.1.1	Résultats attendus	13
5.1.2	Utilisation	13
5.1.3	Exemples	13
5.2	Génération d'une courbe à partir d'un fichier tabulé	14
5.2.1	Prérequis	15
5.2.2	Résultats attendus	15
5.2.3	Exemples	15

Table des figures

1	Espace de travail par défaut	4
2	Installation des bibliothèque perl	5
3	Utilisation minimale de queryNCBI.pl	6
4	Installation de krona	7
5	Utilisation minimale de generateDirectories.pl	8
6	Génération des données	9
7	Produit un fichier weka avec count_kmer	11
8	Produit un fichier weka avec count_kmer en précisant un dossier de taxon	12
9	Utilisation du script execAndEval.pl	13
10	Contenu du fichier list.txt	14
11	Contenu du fichier result.log	14
12	Utilisation du script make_graph.pl	15

1 Introduction

Le but de ce document est de présenter les différents outils développés, pour permettre à un utilisateur externe au 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 1 :

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

FIGURE 1 – Espace de travail par défaut

- query/ : Voir section [3.1](#),
- create_db/ : voir section [3.2](#),
- generate_data/ : voir section [3.3](#),
- cpp/ : voir section [4.1](#),
- cpp/ : afin de lancer des validation croisée [5.1](#)
- generate_graph/ : voir section [5.2](#).

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 est un script d'installation de paquets Debian et de modules Perl nécessaires pour l'utilisation des scripts perl, voir [figure 2](#)

```
$ ls -l
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])\
AND "complete genome"[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 l'extension (-out eukaryota)
- Optionnelles :

1. [http ://www.ncbi.nlm.nih.gov/](http://www.ncbi.nlm.nih.gov/)

- path : dossier où sauvegarder le genbank (/home/me/bdd/)
- not : liste d'id à ne pas récupérer (-not 33630,33258)
- 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 3 :

```
$ ls
queryNCBI.pl

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

$ ls
queryNCBI.pl eukaryota.gb
```

FIGURE 3 – Utilisation minimale de queryNCBI.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 nécessaire 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 fichiers 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 données (/home/me/bdd/)
 - time : temps pris pour la création de la base (-time)
 - help : Affiche l'aide.

3.2.4 Exemples

Utilisation 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/
```

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,
- *wsiz*e 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 quelles séquences compter : cox1,cox2,genomes...,
- *root* racine où on doit établir l'apprentissage,
- *jump* taille du décalage de la fenêtre,
- *learn* taille de l'apprentissage pour la validation croisée,
- *start* début de la taille du read dans la prédiction pour la validation croisée,
- *end* fin de la taille du read dans la prédiction pour la validation croisée,
- *step* taille du pas pour les différentes fenêtre pour la prédiction,
- *sample* nombre de séquences à considérer par taxon,
- *weka* imprime sur la sortie le format weka,
- *help* affiche l'aide.

4.1.2 Exemples

Pour produire un fichier de fréquence au format weka avec un fichier fasta voir la figure 7. Cependant utiliser un fichier simple ne produira pas de classe à prédire pour cela il faut fournir un dossier 8 avec l'architecture établi par le programme présenté en 3.2.

```
$ make

$ ./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 %Data(0,0)
...
```

FIGURE 7 – Produit un fichier weka avec count_kmer

```

$ makefile

$ ./count_kmer -f seq.fasta
                -k pattern.txt
                -o out.arff
                -l -1
                --weka
                --root ~/trunk/create_db/Eukaryota__2759/Alveolata__33630

$ 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 {5794,5878,others}

@DATA
0.15,0.64,...,5878 %Data(0,0)

```

FIGURE 8 – Produit un fichier weka avec count_kmer en précisant un dossier de taxon

5 Validation croisée

5.1 Validation croisée à un niveau taxonomique

```
$ cd cpp/count_kmer/
```

5.1.1 Résultats attendus

Un fichier au format tabulé avec

- La première colonne : la taille des reads de l'apprentissage
- la seconde : la taille des reads de l'espace à prédire
- et la dernière : le taux de vrais positifs

Ce fichier se nomme `result.log`.

5.1.2 Utilisation

Les options

- *noData* libérer la mémoire suite au chargement des données,
- *kmer* fichier contenant les patterns de kmers,
- *key* sur quels séquences compter : `cox1,cox2,genomes...`,
- *root* racine où on doit établir l'apprentissage,
- *learn* taille de l'apprentissage pour la validation croisée,
- *start* début de la taille du read dans la prédiction pour la validation croisée,
- *end* fin de la taille du read dans la prédiction pour la validation croisée,
- *step* taille du pas pour les différentes fenêtres pour la prédiction,
- *sample* nombre de séquences à considérer par taxon,
- *list* une liste de taille pour la taille de l'apprentissage

5.1.3 Exemples

Utilisation de base voir figure 9

```
$ perl execAndEval.pl --root "../create_db/Eukaryota__2759"
--start "100"
--end "300"
--step "50"
--list list.txt
--kmer "pattern.txt"
--sample "20"
```

FIGURE 9 – Utilisation du script `execAndEval.pl`

Si on lance la commande de la figure 9 et si list.txt correspond au fichier de la figure 10 alors notre fichier aura la forme de celui présenté en figure 11

```
$ cat list.txt
-1
1000
200
```

FIGURE 10 – Contenu du fichier list.txt

```
$ cat list.txt
complete    100    61.7
complete    150    63.1
complete    200    64.7
complete    250    73.9
complete    300    77.3
1000         100    68.9
1000         150    44.3
1000         200    50.7
1000         250    66.9
1000         300    62.9
200          100    91.3
200          150    98.4
200          200    97.3
200          250    98.4
200          300    99.0
```

FIGURE 11 – Contenu du fichier result.log

5.2 Génération d'une courbe à partir d'un fichier tabulé

```
$ cd generate_graph
```

5.2.1 Prérequis

Un fichier au format tabulé avec à la première colonne la taille du read de l'espace d'apprentissage, à la suivante la taille du read de l'espace de prédiction et à la dernière colonne au taux de vrais positifs

5.2.2 Résultats attendus

L'outil génère un fichier au format pdf représentant le graphe de vrai positif en fonction de la taille du read.

5.2.3 Exemples

Utilisation de base voir [figure 12](#)

```
$ perl ./make_graph.pl -in ../cpp/count_kmer/result.log -title eukaryotaComplete
```

FIGURE 12 – Utilisation du script make_graph.pl