Compte Rendu de Projet

Informations Importantes

Le programme « CoronaWrapper » est le projet de premier semestre réalisé par SAIDI Adel, RICHEZ Elie et TAVOILLOT Antoine.

Dans le dossier fourni, sont présents :

- Un package python contenant le fichier principal (App.py), des fichiers de classes python et des ressources (icônes, etc.),  
- Ce fichier, le compte rendu de projet.

Toutes les signatures RSA sont produites avec SHA-512 vérifiable avec la clé publique d’empreinte « 680A 1D42 AF6A 6313 A7B1 814D F527 954F 3054 4C37 » au nom de « Licence 1 – Info – Groupe B » avec en description, les noms des personnes ayant travailler sur le projet.

La signature ci-dessous permet de vérifier l’authenticité de ce compte-rendu :

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA512

Signature de la Documentation du Projet CoronaScrapper,

En date du 9 Décembre 2022,

Par Saidi Adel, Richez Elie et Tavoillot Antoine.

Empreinte de clé publique : 680A 1D42 AF6A 6313 A7B1 814D F527 954F 3054 4C37

-----BEGIN PGP SIGNATURE-----

Version: BCPG C# v1.8.9.0

iQKtBAEBCgCXBQJji5EdkBxMaWNlbmNlIEwxIEluZm8gLSBHcm91cGUgQiAoUmlj

aGV6IEVsaWUsIFRhdm9pbGxvdCBBbnRvaW5lLCBTYWlkaSBBZGVsIDsgR3JvdXBl

IEIsIExpY2VuY2UgMSBJbmZvcm1hdGlxdWUsIE1vbnRwZWxsaWVyLCAyMDIyLCBP

ZmZpY2lhbCBSU0EgS2V5KQAKCRD1J5VPMFRMN9GGEACRDGwiPMZPfO0Co9GnzGEE

FJqQFEUY3bCBG5o23BWm79rJdaJtnBbXKynFyaoTg4r89NY7kzbxGWgSqKH3khlp

3s1OPWQc0aEDn9a1Jj3YdZ9/ZBkwkdU5CCDFcVcCA7jzG0BZIHI0FOWt9x7jVQAt

xbNH/vjKujeWQ7zukP+Sn8q5WXrwRKdJ/Wk0zB2Kt4qpe/JuexdRPjseuFdM2N2r

bbzZz41NN3SZOLeCCdfq+9QK29u2wkhnJKYcAQQ1A3arxWvKnAoQvmCIjxkG4Y/G

QmwSxcVUdN4ERkNZfWhDNQQuV2EDBXTAySjy5PLuIa3itBF9ATJASjXHeZA2ZITn

TSVkvX5HihEB1uPYAQyEFwCoHkP+BPMAWEcn0+5sN8o2q6ZRK9T4U5+9wUed38Iz

0WlHTk8i1HBIxwo5rAZo7k2KwhS/ocauw9hik+mAXj7q0nEoo8RX0nZu78lKWnTc

Z/Nfd8BivKt65mCWBMg/vry8XvyuL2a+J9EXGpoC3AJx4nDIfg833ja2TKIo7Tgk

YkccJz8tIqZrGOookOei5AsN48tSjegnzCcKICnr4+oB/eAQDHSGah55J9SnP/W2

N9POEdPGACjkCdntJScCvHxzkKoYbmtUI9cvPz4YWX8OO2HH/r2sNV0ZvsyWqZcX

pPqc06YcXSd8TvyYOqSZaw==

=+/zz

-----END PGP SIGNATURE-----

Table des matières

[Fonctionnement du logiciel 3](#_Toc121506119)

[Localisation des Etapes 3](#_Toc121506120)

[Etapes 3](#_Toc121506121)

[Spécificité du Logiciel 4](#_Toc121506122)

[Code Source 4](#_Toc121506123)

[Bibliothèques 4](#_Toc121506124)

[Développement Graphique 4](#_Toc121506125)

[Perfomance moyenne 4](#_Toc121506126)

[Organisation du répertoire du logiciel 5](#_Toc121506127)

[Répertoire par défaut 5](#_Toc121506128)

[Modification du répertoire lors de l’utilisation de CoronaWrapper 5](#_Toc121506129)

[Exceptions et Erreurs possibles 6](#_Toc121506130)

[Exception de corruption de mémoire sur ZSH 6](#_Toc121506131)

[Levée d’exception lors de l’analyse impossible d’un gène 6](#_Toc121506132)

[Erreurs de données sur le Graph 6](#_Toc121506133)

[Paramètres du Logiciel 7](#_Toc121506134)

[Localisation du fichier 7](#_Toc121506135)

[Modification d’une règle 7](#_Toc121506136)

[Liste des règles 7](#_Toc121506137)

[Logs et historique d’utilisation 8](#_Toc121506138)

[Activation/Désactivation de l’écriture des Logs 8](#_Toc121506139)

[Création et écriture du fichier 8](#_Toc121506140)

# Fonctionnement du logiciel

## Localisation des Etapes

L’ensemble des étapes se trouvent dans les fichiers suivants :

- Récupération de données : Scrapper.py

- Ecriture de données : WriteData.py

## Etapes

Les étapes d’écriture et de lecture sont nommées par lettres dans l’ordre alphabétique.

Seulement, l’Etape A (récupération) est nommée GetData, de même que l’étape (K) qui est nommée GetGenSeqGB.

L’étape d’alignement est incluse dans ManualAlign.py mais n’est pas fonctionnelle pour le moment.

Note importante :

La fonction d’alignement manuelle est implémentée bien que donnant des résultats non-satisfaisant. De fait, le fichier d’alignement manuellement généré n’est pas celui utilisé pour générer les données graphiques.

# Spécificité du Logiciel

## Code Source

« CoronaWrapper » est un logiciel d’analyse et d’étude des séquences du virus *Coronavirus* chez l’homme, la chauve-souris et le Pangolin.

Il est codé en Python 3.7.10, initialement développé sur un processeur ARM (Apple Chipset M1) et portable sur les processeurs ARM et X86 en 64 bits.

Le code source contient les élements ci-dessous :

- App.py (fichier de démarrage de l’application),

- Scrapper/ScrapData.py (acquisition des données depuis la banque du NCBI) et Scrapper/WriteData.py (écriture des différents fichiers GenBank et Fasta et analyse de ces derniers),

- Window/Intro.py (fenêtre d’introduction à l’ouverture du logiciel),Window/MainWindow.py (fenêtre d’étude des données) et Window/Graph.py (fenêtre d’un graph)

- Resources/ (répertoire de différentes resources du logiciel).

## Bibliothèques

CoronaWrapper utilise les librairies suivantes :

- BioPython (librairie d’analyse de données biologiques),  
- Entrez (librairie du wrapper de données sur la banque du NCBI),  
- Tkinter (librairie de fonctions graphiques et d’affichages),  
- PIL (librarie de fonctions graphiques sur les images)

## Développement Graphique

CoronaWrapper est un logiciel possédant une interface graphique comprenant quatre fenêtres :

- Fenêtre d’introduction (Intro.py),  
- Fenêtre principale (MainWindow.py) codé en programmation orienté objet et plus complexe que les autres,  
- Fenêtre d’un graphique (Graph.py) servant à représenter certaines données dans un graphiques,  
- Fenêtre des paramètres (Settings.py) servant à intéragir avec l’utilisateur pour modifier les paramètres de l’application.

L’API graphique utilisée est Tkinter. Le fichier MainWindow.py contient du code orienté objet.

Etant donné que CoronaWrapper est initialement développé sur OSX, l’expérience d’utilisation sera meilleure sur une machine Apple (x86 où ARM).

## Perfomance moyenne

Les performances suivantes ont été obtenus sur plusieurs tests sur une machine Apple MacAir, M1, 16gb de RAM connectée en réseau à distance à une fibre optique :

- Temps de récupération des fichiers depuis la banque de données du NCBI : Entre 2,4 et 3,8 secondes,

- Affichage de la première fenêtre : Entre 0,6 et 1,2 seconde

- Affichage des résultats dans le tableau (à partir du clic sur un bouton) : Entre 0,4 et 0,6 seconde

- Génération du graph : Entre 0,5 et 0,7 secondes

# Organisation du répertoire du logiciel

## Répertoire par défaut

Le logiciel est fourni dans un répertoire par défaut ; Aucun fichier n’a été pour le moment créé où modifié par CoronaWrapper.

Ce répertoire se compose :

- D’un dossier *Ressources*, composé de divers ressources graphiques utilisés par le logiciel,

- D’un dossier *Scrapper*, composé d’un fichier de récupération des données (ScrapData.py) et d’un fichier de manipulation des données (WriteData.py)

- D’un dossier *Window*, composé de quatre fichiers d’affichage graphique : Intro.py pour la fenêtre d’introduction, MainWindow.py pour la fenêtre principale, Settings.py pour la fenêtre des réglages et Graph.py pour la fenêtre du graphique.

- D’un fichier *Settings*, composé de trois fichiers, CheckFiles un fichier de classe créant les fichiers permettant le fonctionnant du logiciel, ReadSettings.py qui permet la lecture de Settings.txt et WriteSettings qui réécris Settings.txt lorsque les paramètres sont changés.

## Modification du répertoire lors de l’utilisation de CoronaWrapper

Lors de l’utilisation du logiciel, un dossier « Fichiers Externes » est créé. Il permet de stocker l’ensemble des fichiers qui vont par la suite être écris par CoronaWrapper.

Ce nouveau répertoire contiendra :

- Le fichier seq\_covid.gb,

- Le fichier info\_seq\_covid.txt,

- L’ensembles de dossiers relatifs aux gènes,

- Un dossier Log, qui contient des fichiers relatifs à l’utilisation du logiciel.

# Exceptions et Erreurs possibles

CoronaWrapper peut être soumis à des exceptions fatales notamment lors de certains calculs. Ces exceptions ne concernent pas des fonctions à proprement dits et sont minimes ; Seules deux exceptions fatales ont été relevées.

La librarie tkinter gère une exception sur la partie graphique en continuant à faire fonctionner le programme tout en la signalant dans le terminal. Concernant les exceptions matériels, comme des exceptions de la mémoire où de calculs (division par zéro, etc.), le programme enverra un signal d’extinction et crash.

Pour éviter le risque d’erreurs et de crash, CoronaWrapper procédera à une levée d’exception et le marquera dans les logs.

## Exception de corruption de mémoire sur ZSH

A plusieurs reprises sur OSX 13.0 (Ventura), des exceptions de corruption de mémoire « *trace trap* » se sont produits avec ZSH. Il semble que cette exception soit lié à une écriture dans un bloc de mémoire non-alloué pour cette fonction.

Cependant, cette erreur n’a pas été relevée sur d’autres shell. Il est probable que cette erreurs soit liés à des défauts d’optimisation de Python sur OSX 13.0 où que cela soit due à une mauvaise gestion de la mémoire de Python sur des chipset ARM.

## Levée d’exception lors de l’analyse impossible d’un gène

Lors de la comparaison d’un gène apparaissant dans plusieurs organismes, si celui-ci est orthographié de plusieurs façons différentes, CoronaWrapper retournera une levée d’exception.

D’après les essais effectués, la levée d’exception concerne le gène ORF1AB. Ce gène ne sera pas analysable ni dans le graphique, ni dans la comparaison des nucléotides. Cependant, étant donné que la liste des identifiants du gène ORF1AB pour les trois virus sont connus, seul le fichier « orf1ab.gb » sera disponible dans le répertoire Fichier Externes/Gènes ORF1AB.

## Erreurs de données sur le Graph

Pour que le graphique affiche l’ensemble de données, l’ensemble des fichiers doit être téléchargés et les données extraites.

Pour éviter ces complications, la règle *DownloadAtOpening* (demandant à CoronaWrapper de télécharger l’ensemble des fichiers

# Paramètres du Logiciel

## Localisation du fichier

CoronaWrapper utilise le fichier Settings.txt pour démarrer. Ce fichier contient des règles permettant au logiciel de correctement fonctionner.

Il est recommandé de ne pas directement modifier le fichier, de le déplacer de son répertoire, de le supprimer où de l’employer pour une autre tâche en raison d’un risque de corruption.

## Modification d’une règle

Dans la fenêtre d’introduction, cliquez sur le bouton « Paramètres » et accédez à l’interface. Lorsque vous avez modifié les paramètres, vous devez cliquez sur « Sauvegarder » pour modifier le fichier Settings.txt.  
Une fois que les règles ont été changés, redémarrez le programme et executez le normalement.

Les règles se modifient via une case à cocher. Si la case est cochée, la règle est considérée comme vraie. Sinon, comme fausse.

## Liste des règles

La suivante établie l’ensemble des règles que CoronaWrapper utilisera pour fonctionner et leurs valeurs par défaut lors de la première utilisation :

*Using MafftCommandLine*

Valeur par défaut : True

Description : Règle pour savoir si l’outil MafftCommandLine doit être utilisé comme outil principal d’alignement des séquences.

Pour accéder à l’étape K, cette règle doit être désactivée.

*Using Logs*

Valeur par défaut : True

Description : Création de fichiers de Logs ; True or False.

*Rewriting File*

Valeur par défaut : True

Description : Si les fichiers ont déjà été créés et qui doivent seulement être analysés, vous pouvez désactiver cette règle pour éviter de réécrire les fichiers à chaque fois.

*Keep OldLogs*

Valeur par défaut : False

Description : Conserve où non les anciens logs. Réinitialise les fichiers de logs dans le cas où la règle est définie sur False.

*Create All Files*

Valeur par défaut : True

Description : Lors de la première utilisation où lorsque le répertoire des gènes n’est pas détecté, CoronaWrapper télécharge et créer l’ensemble des fichiers d’un seul coup.

Attention : Télécharger les fichiers en une fois peut prendre un certains temps.

Si cette règle est désactivée, seul le gène S (Spike) sera téléchargé et les fichiers relatifs aux gènes seront créés.

# Logs et historique d’utilisation

CoronaWrapper génère un fichier de Logs qui contient l’historique des actions effectués par le logiciel.

Ce fichier est localisable dans Fichiers Externes/Logs/Logs.txt.

## Activation/Désactivation de l’écriture des Logs

La règle *UsingLogs* contrôle l’utilisation où non du fichier de Logs. Par défaut, la règle est activée.

Note : Etant donné le faible volume de données traitées (centaines d’octets à quelques ko), il est probable que l’activation de l’écriture des logs n’ai qu’un impact infime sur les performances du disque de stockage.

## Création et écriture du fichier

Le fichier est créé lors du premier démarrage du logiciel, même si vous ne souhaitez pas l’utiliser.

A chaque action, le fichier est réécrit. Si une modification à eu lieu, vous devez fermer et rouvrir le fichier.

Si la règle *keepOldLogs* est désactivé, le fichier sera vidé à chaque nouvelle exécution du logiciel.