

# Logiciel d'analyse de séquence: rapport

année 2021-2022

BOSSUT Noémie  
BEGHIN Romane



# Sommaire :

[Méthode de partage](#)

[Organisation des modules](#)

[Réalisation des modules](#)

[Difficultés rencontrées](#)

[Difficultés générales](#)

[Difficultés propres à chaque module](#)

[Module 1 :](#)

[Module 3:](#)

[Modules 6 & 7:](#)

[Améliorations possibles](#)

## Méthode de partage

Nous avons créé des modules suivant l'énoncé en nous aidant de Github pour partager nos avancements et mettre le projet au propre.

## Organisation des modules

Notre projet s'organise en modules qui peuvent être appelés par un menu. Le menu, appelé dans le main, permet une interface avec l'utilisateur claire et facile d'utilisation.

## Réalisation des modules

Nous avons fait les modules 1 à 6 et avons traité le module 7 en partie.

Les modules 3 et 4 nécessitent l'ouverture de fichiers et la lecture de séquences fasta simplifiées contenues dedans. Pour cela, nous avons créé la fonction *recupereSeq1Seq2* dans utils : cette fonction (appelant *extract\_sequence*) récupère les séquences et vérifie si elles ont bien la même taille. (voir code partiel ci-dessous)

```
14
73 Seq1Seq2 recupereSeq1Seq2(const char* path_input1, const char* path_input2) {
74     char seq1[10000];
75     char seq2[10000];
76     extract_sequence(path_input1, seq1);
77     extract_sequence(path_input2, seq2);
78
79     Seq1Seq2 sequences;
80     sequences.seq1=seq1;
81     sequences.seq2=seq2;
82
83     int lenseq1=strlen(seq1);
84     int lenseq2=strlen(seq2);
85
86     if (lenseq1==lenseq2)
87         sequences.memeLongueur='O';
88     else
89         sequences.memeLongueur='N';
90
91     return sequences;
92 }
```

## Difficultés rencontrées

### 1) Difficultés générales

Ayant eu de nombreuses lors de la conception générale de l'application, seuls les 6 premiers modules ont été entièrement réalisés. Une ébauche du module 7 est toutefois visible.

Au sein même de l'application, nous avons rencontré lors de l'ouverture du fichier FASTA de P2RX7, alors que les fichiers test (d'une centaine de lignes maximum) ne posaient aucun problèmes. L'espace de certaines chaînes de caractère a donc été augmenté en dépit du conseil donné dans la consigne, passant de 10000 à 100000, mais résolvant ledit problème.

Lors de la demande du chemin contenant le fichier à traiter, le chemin relatif ne semble pas fonctionner. Nous nous sommes donc résignées à écrire le chemin absolu à chaque fois, ce qui rend l'utilisation du logiciel plus pénible.

Enfin, par manque de temps, les erreurs d'entrées de l'utilisateur peuvent faire crasher, ou boucler le programme (exemple ci-dessous: écriture d'un caractère plutôt que d'un chiffre).

```
Vous avez du faire une erreur, seuls les chiffres 1, 2 et 3 sont autorisés, veuillez recommencer :)
Vous avez du faire une erreur, seuls les chiffres 1, 2 et 3 sont autorisés, veuillez recommencer :)
Vous avez du faire une erreur, seuls les chiffres 1, 2 et 3 sont autorisés, veuillez recommencer :)
Vous avez du faire une erreur, seuls les chiffres 1, 2 et 3Z
[5] + 21097 suspended ./ProgrammeNBRB
```

## 2) Difficultés propres à chaque module

### Module 1 :

Au stade final, ce module trouve la même plus grande séquence codante que le site ORFfinder (<https://www.ncbi.nlm.nih.gov/orffinder/>), mais n'indique pas la même position de départ et de fin ni le même nombres d'acides aminés. Ce problème n'a pas été réglé.

### Module 3:

Le module 3 présente un comportement imprévisible, sans que nous en ayons trouvé la source. En effet, une première utilisation peut donner la suite d'acides aminés identiques à celle trouvée sur Orffinder, mais une autre utilisation va trouver des codons stops en milieu de séquence. Souvent, les ~30 premiers acides aminés sont exacts, puis le cadre de lecture change. Un printf a été ajouté à la fonction, pour vérifier que le cadre de lecture est toujours bon, et il l'est.

Plusieurs types d'algorithmes et de boucles ont été testés (if imbriqués, if && if etc.), et rapportent toujours le même problème. Au final, des fonctions pour vérifier chaque acide aminé ainsi que le détail de chaque possibilité a été inscrit dans le code, ce qui le rend plus lourd, mais il semble correctement fonctionner plus souvent (l'évolution des versions est visible sur GitHub).

### Modules 6 & 7:

Pour les modules 6 et 7, il était conseillé de représenter les acides aminés à l'aide d'une structure afin qu'il soit plus facile de déterminer la polarité des acides aminés rencontrés. Nous avons trouvé un moyen de déterminer la polarité des acides aminés mais pas en utilisant une structure, ce qui aurait probablement permis une complexité moindre.

## Améliorations possibles

Une amélioration des fonctionnalités du module 6 a été tentée (affichage de la séquence consensus si demandé) mais sans succès : Les caractères ne semblent pas reconnus (on suspecte un problème de codage des caractères). Voici le code d'une procédure demandant si l'utilisateur souhaite l'affichage de la séquence (cette procédure serait à rajouter dans *module\_consensus.c* et rajouter sa déclaration dans *module\_consensus.h*) et l'affichage donné:

```

149
150 void visualisationConsensus(int n, char consensus[40]) {
151     int nbCaracteres=0;
152     char rep;
153     printf("\nSouhaitez-vous voir la séquence consensus ? (Y ou N)\n");
154     scanf("%s",&rep);
155
156     if (rep=='Y') {
157         FILE * seqCons=fopen(consensus,"r");
158         if (!seqCons) {
159             printf("\n\nLe fichier %s n'a pas pu être ouvert\n\n",consensus);
160         }
161         else {
162             printf("\nVoici la séquence : \n");
163             char c;
164             while(nbCaracteres<n){
165                 nbCaracteres+=1;
166                 c=fgetc(seqCons);
167                 printf("%c",c);
168             }
169         }
170     }
171     else if (rep!='N') {
172         printf("\nJe n'ai pas compris votre demande.\n");
173         visualisationConsensus(n,consensus);
174     }
175 }
176

```

Si on répond Y : (affichage incompréhensible)

```

nom du fichier: test.txt
Dans quel fichier vide voulez-vous enregistrer la séquence consensus ?
Nom du fichier: consensus.txt
La séquence consensus à été remplie dans le fichier consensus.txt avec succès.
Souhaitez-vous voir la séquence consensus ? (Y ou N)
Y
Voici la séquence :
*****
Merci d'avoir utilisé le module de séquence consensus, souhaitez-vous passer à un autre
module? 1 pour continuer, 2 pour quitter

```

Si on répond N : (le programme s'arrête comme prévu)

```

Souhaitez-vous voir la séquence consensus ? (Y ou N)
N
Merci d'avoir utilisé le module de séquence consensus, souhaitez-vous passer à un autre
module? 1 pour continuer, 2 pour quitter
1

```

Si on répond autre chose : (On redemande la réponse comme prévu)

```

La séquence consensus à été remplie dans le fichier consensus.txt avec succès.
Souhaitez-vous voir la séquence consensus ? (Y ou N)
T
Je n'ai pas compris votre demande.
Souhaitez-vous voir la séquence consensus ? (Y ou N)
J
Je n'ai pas compris votre demande.
Souhaitez-vous voir la séquence consensus ? (Y ou N)
Y
Voici la séquence :
*****
Merci d'avoir utilisé le module de séquence consensus, souhaitez-vous passer à un autre
module? 1 pour continuer, 2 pour quitter

```