# Présentation de projet

## analyse de données biologiques

Bouaoud Malik

Kenai Selma

Université de Lille/Informatique
UE Bio-informatique

12/02/2021

Université de Lille

# Introduction

- Analyse de données biologiques:
  - Récupération de séquence biologique depuis le NCBI.
  - Extraction d'information.
  - Traitement d'information.
  - Recherche des sites de facteurs de transcription.

Université de Lille

# Problème :

Rechercher les TF qui régulent un ensemble de gènes supposés co-régulés, grâce aux occurence de son TFBS.

Université de Lille

# Solution triviale:

Pour une première solution il suffit d'observer les occurences d'un TFBS pour un TF donné dans un ensemble de séquences d'ADN en amont du gène.
Et donc on devra procéder par **fenêtre glisssante** en récupérant la fenêtre avec le nombre d'ocurrence de notre TFBS, le score dans notre cas d'étude.

Université de Lille

# Solution proposée :

Au lien de compter simplement les occurences d'une fenêtre, nous proposons de calculer une moyenne pour chacunes des fenêtres Cette solution nous aie paru plus pertinente puisqu'ici nous prévilgions beaucoup plus le score de chaques séquences que son nombre d'occurence.

Université de Lille

# Étapes :

Cela se fait en 2 étapes principales :

- Traitement d’ensemble de séquences biologiques.
- Personalisation du calcul du meilleur score.

Université de Lille

# Traitement d'ensemble de séquences biologiques

1. Récupérer les séquences en amont du gène depuis le NCBI, grâce à **download promotors**
2. utiliser **scan seq** et **scan all sequences** pour récupérer les fenêtres depassant un certain seuil.
3. Stocker les resultats satisfaisaint notre contrainte dans une structure de donnée imbriquée, c'est à dire : **List[tuple(List,identifiant de la sequence)]**

Université de Lille

# Personalisation du calcul du meilleur score

Université de Lille

**Algorithm 1:** BestWindowPersonalized

**Result:** (startPosition,EndPosition,BestScore)
**Input:** (ListOFSequences,WindowsSize,seqSize)
startCoordinate $\leftarrow 0$
endCoordinate $\leftarrow windowSize$
List $\leftarrow []$
**For** i $\in List$
Moyenne $\leftarrow 0$
**if** $startCoordinate¡seqSize$ **and** $endCoordinate ¡= seqSize$ **then**
    Moyenne $\leftarrow scanSequence(i[0], startCoordinate, EndCoordinate)$
    List $\leftarrow$ **Append**$(StartCoordinate, EndCoordinate, Moyenne)$
    startCoordinate $\leftarrow EndCoordinate$
    EndCoordinate $\leftarrow EndCoordinate + windowSize$
**else**

**end**
//car la moyenne est le troisième paramètre de la liste MaxMoy
$\leftarrow Max(list[3])$
**return** (x,y,MaxMoy)

Université de Lille

# Des questions ?

Université de Lille