Université Paris Cité

Rapport projet court

UFR Sciences du vivant

Conception d'un programme d'alignement d'*embedding* par programmation dynamique

---

Mia Legras

$2^{ème}$ année, Master de Bioinformatique,

Université Paris Cité

2022 - 2023

Encadrants

Tatiana Galochkina

Jean-Christophe Gelly

Table des matières

1 Introduction

L'alignement de séquences protéiques est une méthode qui permet de visualiser régions similaires entre elles. Ils sont réalisés de façon à maximiser le nombre d'acides aminés équivalents entre les séquences par des programmes informatiques. Les insertions ou délétions sont alors représentés par ce qu'on appelle des gaps. Les séquences alignées sont obtenues grâce à un programme dynamique qui parcourt une matrice d'alignement qui peut être générée à partir de différents scores, comme l'embedding. Une matrice vectorielle dont chaque valeur correspond à une caractéristique physico-chimique, structurale, ou autre de la protéine. L'utilisation de cette valeur lors d'un alignement permet d'augmenter la précision de l'alignement.
Le but de ce projet consiste à implémenter un programme permettant de réaliser un alignement entre deux séquences protéiques, qui se basera sur un dot product entre l'embedding de ces séquences.

2 Matériels et Méthodes

2.1 Données

Ce projet nécessite deux séquences protéiques en entrée, qui ont chacune un fichier fasta et un embedding.
Le fichier fasta permet d'obtenir la séquence protéique, tandis que l'embedding Les séquences protéiques ont été encodé en embedding de longueur 1024, disponibles à l'adresse suivante www.dsimb.inserm.fr/ gelly/data/Emb.zip. Ils ont été obtenus par la méthode T5 ProtTrans (github.com/agemagician/ProtTrans), dans chacun de ces fichiers, une ligne de 1024 valeurs représente une position de la séquence protéique.

2.2 Méthode

Dans un premier temps, on calcule le dot product entre chaque position des séquences qui est stocké dans une matrice numpy, qui est utilisée pour calculer la matrice d'alignement. En fonction du type d'alignement voulu, cette matrice n'est pas implémentée de la même façon.

2.2.1 Alignement

La matrice d'alignement global et semi-global utilisent l'algorithme de Needleman Wunsch, qui consiste à mettre à la position mat[i, j] le maximum entre :

— $mat[i-1, j-1] + dot[i, j]$,

— $mat[i, j-1] + gap$,

— $mat[i-1, j] + gap$,

tel que *mat* correspond à la matrice d'alignement, *dot* à la matrice de dot product et le gap à la pénalité donnée lorsqu'il y a une brêche.

La matrice d'alignement local utilise l'algorithme de Smith Waterman, qui consiste à mettre à la position mat[i, j] le maximum entre :

— $mat[i-1, j-1] + dot[i, j]$,

— $mat[i, j-1] + gap$,

— $mat[i-1, j] + gap$,

— 0

Ce qui indique que quel que soit la valeur du gap ou de dot[i, j], la matrice d'alignement ne comportera pas de valeurs négatives.

2.2.2 Séquences d'alignement

Pour les trois types d'alignement proposés, on remonte dans la matrice en passant par le maximum entre :

— $mat[i-1, j-1]$

— $mat[i, j-1]$

— $mat[i-1, j]$

Ce qui ce différencie est la position d'initialisation et de terminaison. En effet, dans le cas d'un alignement global on commence par la dernière case en bas à droite et on remonte jusqu'à la position mat[0, 0].

Pour faire un alignement semi-global, on commence par la position avec le plus grand score dans la dernière colonne et termine à la première colonne.

Enfin, on part de la position avec le score maximum et fini au premier 0 rencontré lors d'un alignement local.

L'alignement local a été implémenté de façon à donner les différents alignements possibles dans le cas où il y a plusieurs valeurs maximum dans la matrice d'alignement.

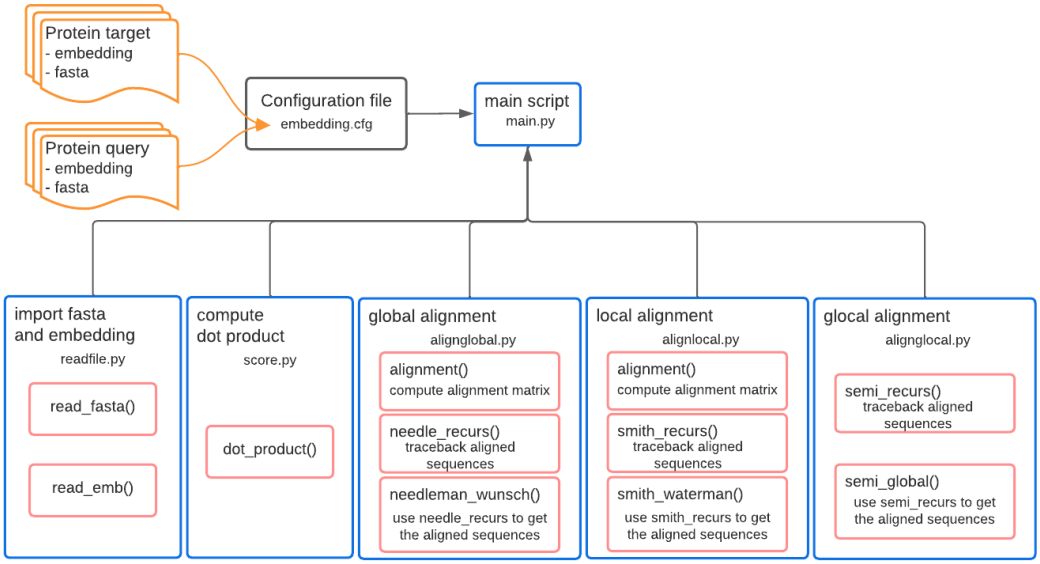

FIGURE 1 – Schéma de la structure du programme d'alignement à partir d'embedding de séquences protéiques

3 Résultats

On teste les alignements implémentés sur deux paires de séquences génomiques.

— 1RDS / 1DE3A $TM_{score} = 0.78741$ [1]

— 1BIF / 1GC7A $TM_{score} = 0.16355$

La pénalité du gap est fixé à 0. Les matrices d'alignement seront alors identiques entre les différents types d'alignement avec la dernière valeur comme score maximum, car la valeur obtenue ne sera jamais négative.

```
1 - ESC-E-YTCGS------T-C--Y-------WSS-DVS-AA-K-AKGY-SL-Y-E-S-G-DTIDDYPHEY-H-D-Y-E--GFD-F--P-----------------V----------S-G-TYYEYPIMSDY-D-V-
1 - AV-T-W-TCLNDQKNPKTN-KY-ETKRLLYNQNK-AES--N-S---HH-AP-L-S-D-GKTGSSYPHW-FTN-G-Y-DG-D-G-KLPKGRTPIKFGKSDCDRPP-KHSKDGNGKT-D-HYLLEFPTFPDGH-D-Y

2 - YTG------G-SPGADRVIFNG-D-D-ELAGVITHTGASG-DDFVACSS-S
2 - -KFDSKKPK-ENPGPARVIYTYP-N-KVFCGIIAHTKENQG-ELKLCS-H-
```

FIGURE 2 – Résultat de l'alignement global entre la protéine 1RDS et 1DE3A

$1^{ère}$ ligne : Séquence target 1RDS ; $2^{ème}$ ligne : Séquence query 1DE3A

```
1 - ESC-E-YTCGS------T-C--Y-------WSS-DVS-AA-K-AKGY-SL-Y-E-S-G-DTIDDYPHEY-H-D-Y-E--GFD-F--P-----------------V----------S-G-TYYEYPIMSDY-D-V-
1 - AV-T-W-TCLNDQKNPKTN-KY-ETKRLLYNQNK-AES--N-S---HH-AP-L-S-D-GKTGSSYPHW-FTN-G-Y-DG-D-G-KLPKGRTPIKFGKSDCDRPP-KHSKDGNGKT-D-HYLLEFPTFPDGH-D-Y

2 - YTG------G-SPGADRVIFNG-D-D-ELAGVITHTGASG-DDFVACSS-S
2 - -KFDSKKPK-ENPGPARVIYTYP-N-KVFCGIIAHTKENQG-ELKLCS-H-
```

FIGURE 3 – Résultat de l'alignement local entre la protéine 1RDS et 1DE3A

1. Template Modeling score

```
1 - ESC-E-YTCGS------T-C--Y-------WSS-DVS-AA-K-AKGY-SL-Y-E-S-G-DTIDDYPHEY-H-D-Y-E--GFD-F--P----------------V----------S-G-TYYEYPIMSDY-D-V-
1 - AV-T-W-TCLNDQKNPKTN-KY-ETKRLLYNQNK-AES--N-S---HH-AP-L-S-D-GKTGSSYPHW-FTN-G-Y-DG-D-G-KLPKGRTPIKFGKSDCDRPP-KHSKDGNGKT-D-HYLLEFPTFPDGH-D-Y

2 - YTG------G-SPGADRVIFNG-D-D-ELAGVITHTGASG-DDFVACSS-S
2 - -KFDSKKPK-ENPGPARVIYTYP-N-KVFCGIIAHTKENQG-ELKLCS-H-
```

FIGURE 4 – Résultat de l'alignement glocal entre la protéine 1RDS et 1DE3A

On a bien les mêmes alignements avec les trois méthodes.

J'ai ensuite comparé la dernière valeur de la matrice d'alignement avec le TMscore. La ressemblance entre les séquences protéiques est considérée comme significative lorsque le TMscore est supérieur à 0.5 .

Comme il est indiqué plus haut (3), 1RDS et 1DE3A ont une forte ressemblance tandis que le TMscore entre 1BIF et 1GC7A est faible, ce qui montre que ces deux séquences ne se ressemblent pas.

Le score d'alignement 1rds/1de3a (580.04) est bien supérieur celui de 1bif/1gc7a (2173.32). Les résultats de l'alignement sont donc cohérents avec les TMscores.

L'alignement a bien été implémenté.

4 Conclusion

Les alignements global, local, et semi-global à partir de l'embedding sont bien implémentés, et les résultats sont cohérents avec les TMscores. Pour obtenir un alignement plus précis, il reste encore à modifier le programme afin de gérer les gaps affines, en activant un flag lorsque le score ne vient pas de celui à la diagonal.

Références

[1] Bepler T, Berger B. Learning the protein language : Evolution, structure,and function. Cell Syst. 2021 Jun 16 ;12(6) :654-669.e3. doi :10.1016/j.cels.2021.05.017. PMID : 34139171 ; PMCID : PMC8238390.

[2] Smith, Temple F. Waterman, Michael S. (1981). "Identification of Common Molecular Subsequences" (PDF). Journal of Molecular Biology. 147 (1) : 195–197. CiteSeerX 10.1.1.63.2897. doi :10.1016/0022-2836(81)90087-5. PMID 7265238

[3] slide player, Rapid sequence alignment, visité le 11/09/22 https://slideplayer.com/slide/14446165/

5 Annexe

J'ai eu un peu de mal à récupérer les embeddings au départ et à créer la fonction récursive qui récupère la séquence d'alignement.

5.1 Exemple 1RDS / 1DE3A

Activer l'environnement conda : `conda activate embedding`

Se placer dans le répertoire `alignment_embedding/src/`.

Remplir le fichier de configuration comme ci-dessous, la partie [paths] n'est pas à modifier :

```
[paths]
    to_data = ../data/
    to_res = ../results/

[files]
    # protein target
    prot_int_emb = rnase_1rds.t5emb
    prot_int_fasta = RNASE_1RDS.fasta
    # protein query
    prot_comp_emb = RNase_U2_1de3a.t5emb
    prot_comp_fasta = RNASE_U2_1DE3A.fasta

    dot = dotprod_matrice.txt
    align = align_matrice.txt

[alignment]
    # True or False
    global = True
    local = True
    glocal = True
    # int
    gap = 0
```

FIGURE 5 – Contenu du fichier de configuration

Ensuite on exécute la commande `python main.py`

On obtient alors les fichiers tels que indiqué ici, avec les noms qu'on aura mis en argument dans le fichier de configuration.