Université
Paris Cité

Project :

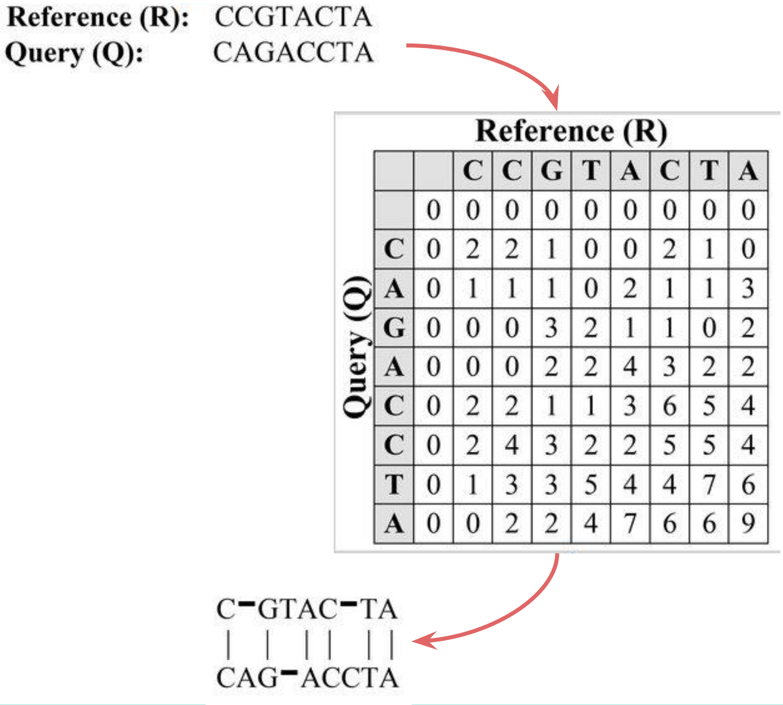
Design of an embedding alignment program by dynamic programming

Mia Legras

M2 BioInformatique
Années : 2022 - 2023

Aim : Implement an alignment using embeddings

Alignment



Reference (R): CCGTACTA
Query (Q): CAGACCTA

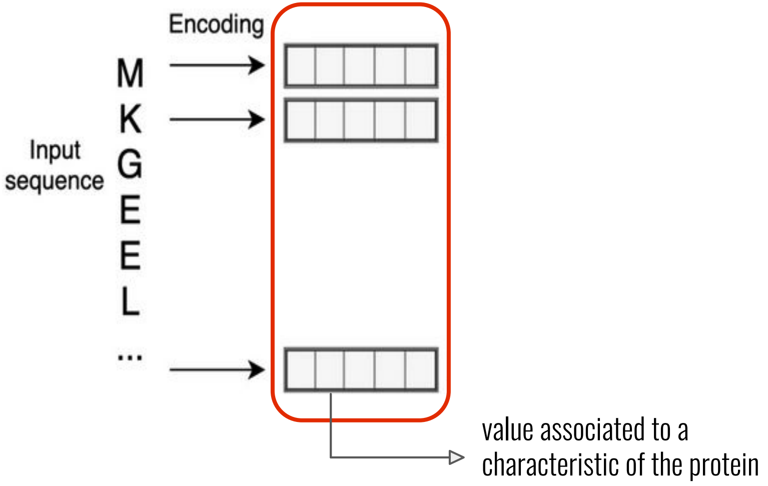
	Reference (R)								
Query (Q)		**C**	**C**	**G**	**T**	**A**	**C**	**T**	**A**
	0	0	0	0	0	0	0	0	0
C	0	2	2	1	0	0	2	1	0
A	0	1	1	1	0	2	1	1	3
G	0	0	0	3	2	1	1	0	2
A	0	0	0	2	2	4	3	2	2
C	0	2	2	1	1	3	6	5	4
C	0	2	4	3	2	2	5	5	4
T	0	1	3	3	5	4	4	7	6
A	0	0	2	2	4	7	6	6	9

```
C-GTAC-TA
| | || ||
CAG-ACCTA
```

Wang, Z., Combs, S.A., Brand, R. *et al.* LM-GVP: an extensible sequence and structure informed deep learning framework for protein property prediction

Embedding

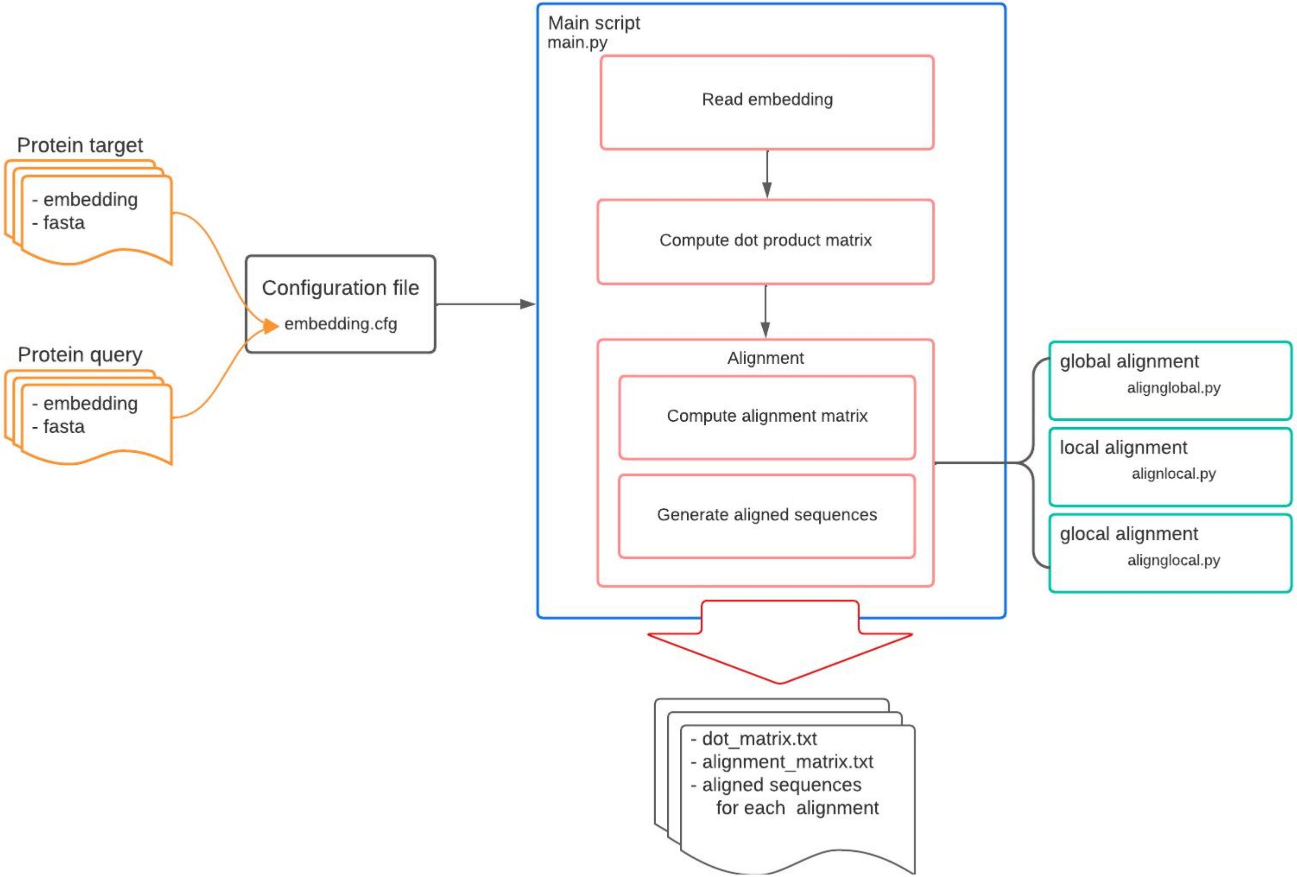
Vectorial matrix



value associated to a characteristic of the protein

Liao, Yi-Lun & Li, Yu-Cheng & Chen, Nae-Chyun & Lu, Yi-Chang. (2018). Adaptively Banded Smith-Waterman Algorithm for Long Reads and Its Hardware Accelerator

Workflow of the alignment program



Protein target

- embedding
- fasta

Protein query

- embedding
- fasta

Configuration file

embedding.cfg

Main script

main.py

Read embedding

Compute dot product matrix

Alignment

Compute alignment matrix

Generate aligned sequences

global alignment

alignglobal.py

local alignment

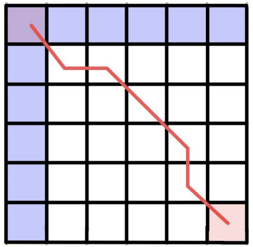
alignlocal.py

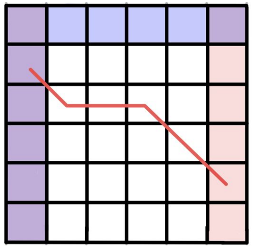
glocal alignment

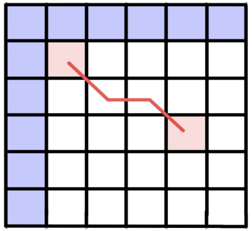
alignglocal.py

- dot_matrix.txt
- alignment_matrix.txt
- aligned sequences for each alignment

Differences between alignments



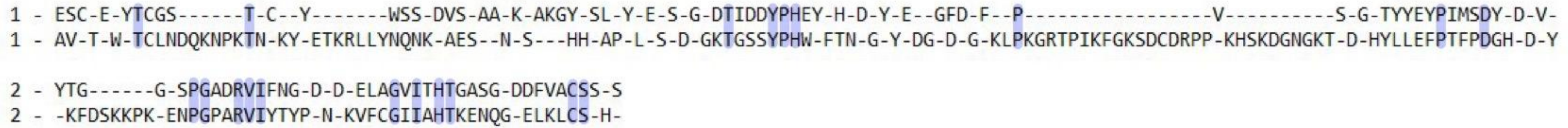




	Global Alignment	Glocal Alignment	Local Alignment
alignment matrix	Needleman & Wunsch algorithm	Needleman & Wunsch algorithm	Smith & Waterman algorithm
Traceback			
Initialization	Bottom right	Right column	Anywhere
Terminaison	Top left	Left column	First position with 0 encountered

Comparison of the 1RDS/1DE3A and 1BIF/1GC7A global alignments

Target : 1RDS Query : 1DE3A

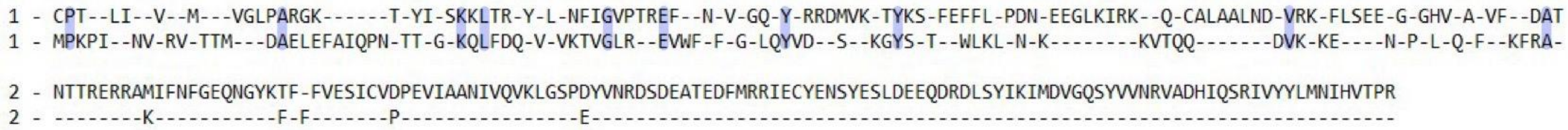


```
1 - ESC-E-YTCGS------T-C--Y-------WSS-DVS-AA-K-AKGY-SL-Y-E-S-G-DTIDDYPHEY-H-D-Y-E--GFD-F--P-----------------V----------S-G-TYYEYPIMSDY-D-V-
1 - AV-T-W-TCLNDQKNPKTN-KY-ETKRLLYNQNK-AES--N-S---HH-AP-L-S-D-GKTGSSYPHW-FTN-G-Y-DG-D-G-KLPKGRTPIKFGKSDCDRPP-KHSKDGNGKT-D-HYLLEFPTFPDGH-D-Y

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

Bottom right alignment matrix score : 2173.32 TMscore = 0.78741

Target : 1BIF Query : 1GC7A



```
1 - CPT--LI--V--M---VGLPARGK------T-YI-SKKLTR-Y-L-NFIGVPTREF--N-V-GQ-Y-RRDMVK-TYKS-FEFFL-PDN-EEGLKIRK--Q-CALAALND-VRK-FLSEE-G-GHV-A-VF--DAT
1 - MPKPI--NV-RV-TTM---DAELEFAIQPN-TT-G-KQLFDQ-V-VKTVGLR--EVWF-F-G-LQYVD--S--KGYS-T--WLKL-N-K--------KVTQQ-------DVK-KE----N-P-L-Q-F--KFRA-

2 - NTTRERRAMIFNFGEQNGYKTF-FVESICVDPEVIAANIVQVKLGSPDYVNRDSDEATEDFMRRIECYENSYESLDEEQDRDLSYIKIMDVGQSYVVNRVADHIQSRIVYYLMNIHVTPR
2 - --------K-----------F-F-------P----------------E------------------------------------------------------------------------
```

Bottom right alignment matrix score : 580.04 TMscore = 0.16355

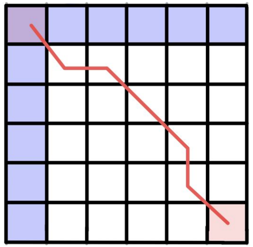
Conclusion & prospect

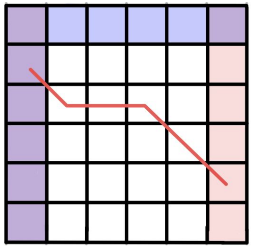
- Successful implementation of alignments using embeddings
- Consistent results with TMscores

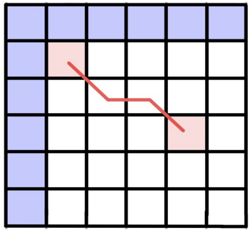
- Need to compute the right value for the gap penalty
- Implement affine gap penalty to compute alignment matrix

Thank you

Differences between alignments

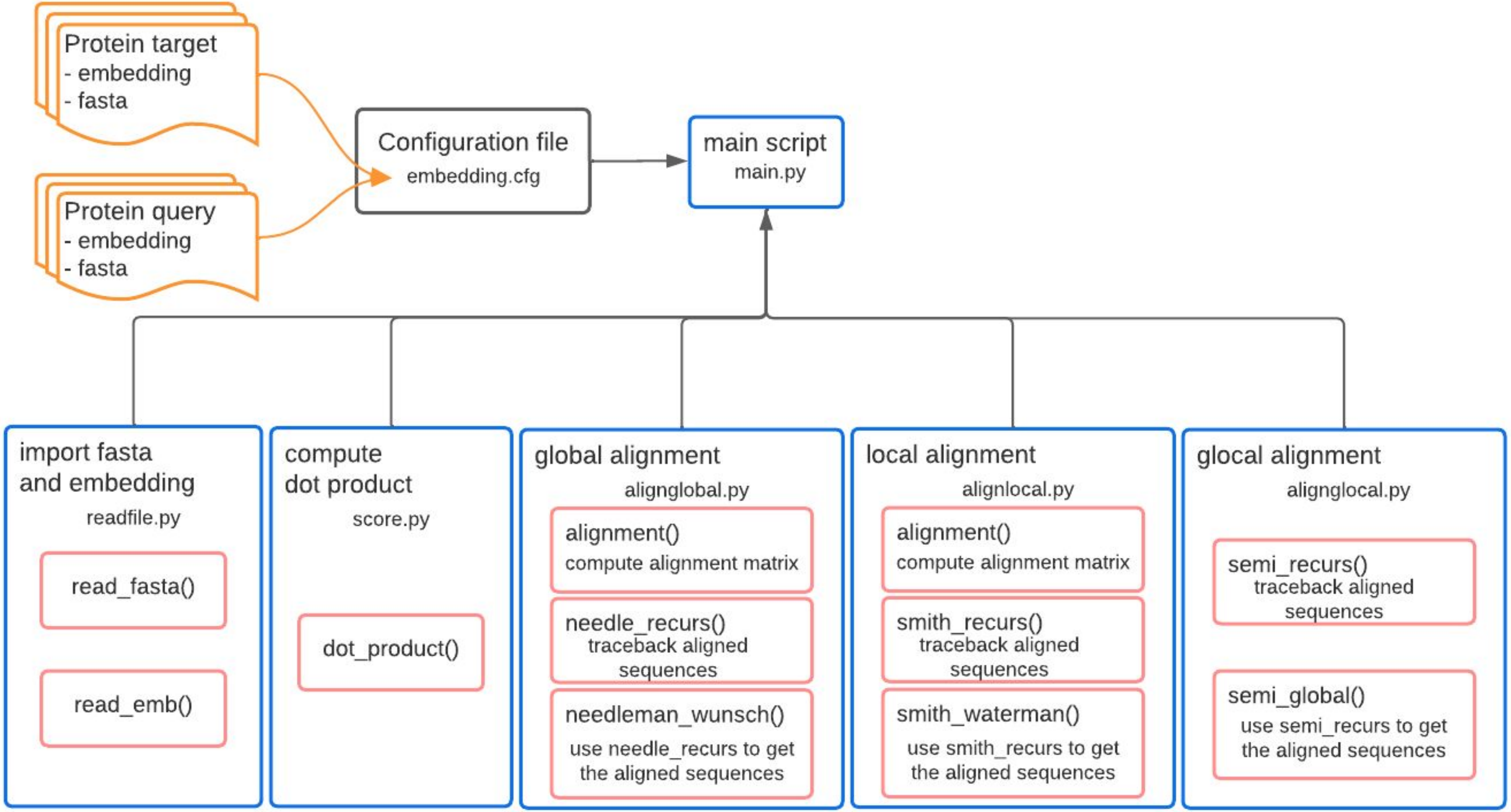




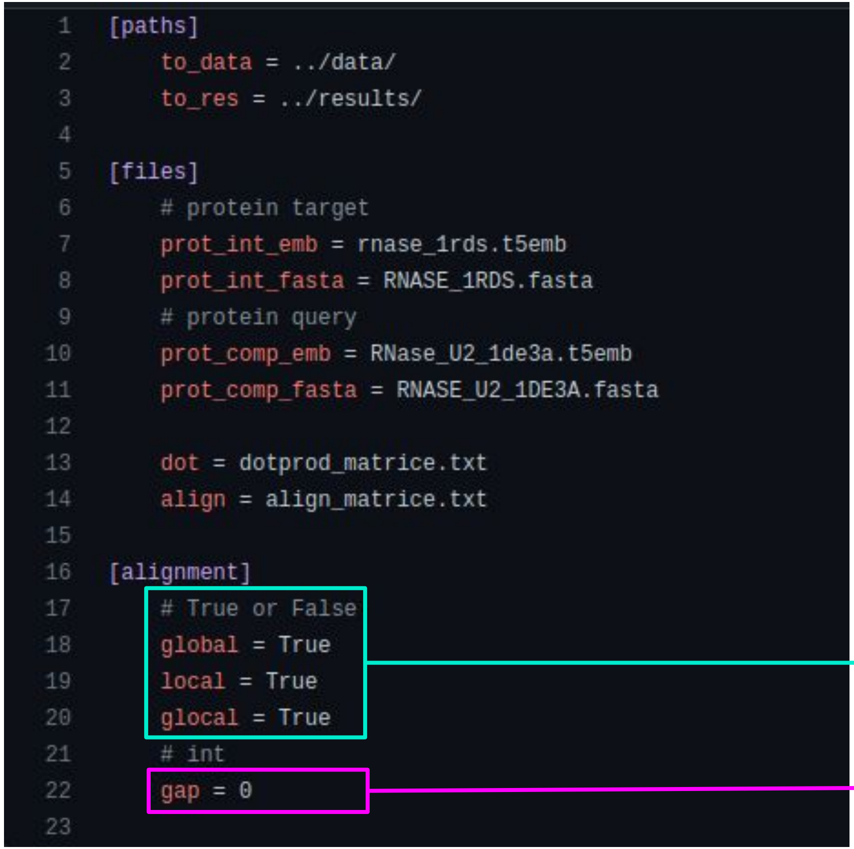


	Global Alignment	Glocal Alignment	Local Alignment
alignment matrix	$\max \left\vert \begin{array}{l} mat[i-1, j-1] + dot[i, j] \\ mat[i, j-1] + gap \\ mat[i-1, j] + gap \end{array} \right.$	$\max \left\vert \begin{array}{l} mat[i-1, j-1] + dot[i, j] \\ mat[i, j-1] + gap \\ mat[i-1, j] + gap \end{array} \right.$	$\max \left\vert \begin{array}{l} mat[i-1, j-1] + dot[i, j] \\ mat[i, j-1] + gap \\ mat[i-1, j] + gap \\ 0 \end{array} \right.$
Traceback			
Initialization	Bottom right	Right column	Anywhere
Terminaison	Top left	Left column	First position with 0 encountered

Structure of the alignment program



Configuration file



```
[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

```

if True : the alignment is generated
all alignments can be generated in one time

the value of the gap penalty