

Dellac Gregory
Mahtoub Moussa

Projet Informatique : Alignement de séquence

I) Introduction :

La création de modèles de protéines est un procédé fortement non linéaire, ce qui rend les modèles générés fortement influencés par les conditions de départ. La technique la plus couramment employée pour modéliser une protéine spécifique est la modélisation par homologie de séquence, qui se déroule généralement selon les étapes suivantes :

- Sélection d'une protéine similaire à la protéine cible, dont la structure 3D est déjà établie.
- Alignement de la séquence de la protéine cible sur cette structure connue.
- Application d'algorithmes avancés pour affiner ce modèle initial en une représentation plus réaliste de la structure de la protéine cible.

Les aspects fondamentaux du modèle, comme la forme, les dimensions, et l'organisation des structures secondaires et tertiaires, sont essentiellement définis lors des deux premières phases. Ces étapes s'appuient sur une méthode fondamentale en bioinformatique : l'alignement de séquences. Dans ce processus, les séquences sont alignées l'une par rapport à l'autre, avec l'insertion d'espaces dans l'une ou l'autre séquence pour aligner les résidus similaires et éviter la superposition de résidus différents. Chaque alignement possible reçoit un score, et la similarité entre deux séquences est déterminée par le score le plus élevé atteignable par leur alignement. Dans ce contexte, l'objectif est d'aligner la séquence d'une protéine dont la structure tridimensionnelle est connue avec celle d'une protéine dont la structure est à déterminer. Toute erreur dans cet alignement affecte directement le modèle final.

L'objectif du projet est de développer un logiciel capable d'aligner deux séquences de protéines.

II) Méthode :

Pour ce faire, nous allons nous appuyer sur la matrice BLOSUM62 (Blocks Substitution Matrix) et l'algorithme de Needleman-Wunsch.

BLOSUM62 est une des matrices de substitution les plus couramment utilisées pour l'alignement de séquences de protéines. Elle est conçue pour quantifier les substitutions acceptables entre différents acides aminés lors de l'évolution des protéines. La matrice est basée sur des observations empiriques de substitutions dans des blocs de séquences de protéines conservées au sein d'un groupe d'espèces. Chaque cellule de la matrice BLOSUM62 donne un score pour remplacer un acide aminé (indiqué par une ligne) par un autre (indiqué par une colonne). Un score positif indique une substitution favorable ou tolérée, tandis qu'un score négatif indique une substitution défavorable. L'utilisation de BLOSUM62 facilite

l'identification d'alignements de séquences biologiquement significatives en privilégiant les substitutions les plus probables basées sur l'évolution des protéines.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

L'algorithme de Needleman-Wunsch est une méthode algorithmique pour l'alignement global de séquences. Cet algorithme, publié en 1970 par Saul B. Needleman et Christian D. Wunsch, est conçu pour identifier l'alignement optimal entre deux séquences en maximisant un score d'alignement global. Il utilise une approche de programmation dynamique pour construire une matrice de scores, où chaque cellule représente le score optimal d'alignement jusqu'à ce point des séquences. L'algorithme prend en compte les scores de correspondance, de substitution (par exemple, à l'aide de la matrice BLOSUM62 pour les protéines), d'insertion et de suppression pour calculer le score.

Needleman-Wunsch

match = 1 mismatch = -1 gap = -1

		G	C	A	T	G	C	G	
		0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5	
A	-2	0	0	1	0	-1	-2	-3	
T	-3	-1	-1	0	2	1	0	-1	
T	-4	-2	-2	-1	1	1	0	-1	
A	-5	-3	-3	-1	0	0	0	-1	
C	-6	-4	-2	-2	-1	-1	1	0	
A	-7	-5	-3	-1	-2	-2	0	0	

The diagram illustrates the Needleman-Wunsch algorithm for global sequence alignment. The matrix shows scores for aligning two DNA sequences. The sequence 'GATTAAC' is aligned with 'GATTAAC'. The alignment path is highlighted with arrows: blue arrows indicate matches (G-G, A-A, T-T, A-A, C-C) and red arrows indicate mismatches (T-T). The final alignment is GATTAAC aligned with GATTAAC.

La figure ci-dessus illustre l'application de l'algorithme de Needleman-Wunsch pour l'alignement global de deux séquences d'ADN. L'algorithme utilise une matrice pour suivre les scores d'alignement, en attribuant des valeurs pour les matches, les mismatches et les gaps (insertions ou délétions).

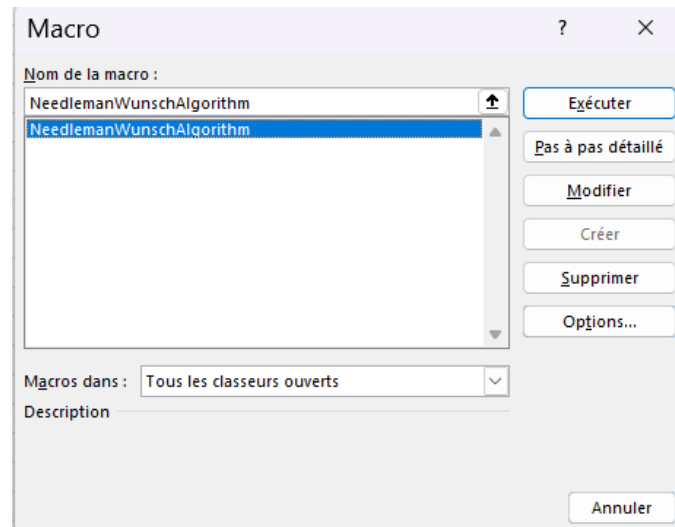
III) Exécution Dirigée :

Lancez Microsoft Excel et ouvrez le fichier "Alignement - M. Moussa et D. Gregory.xlsm". Lors de l'ouverture, une boîte de dialogue de sécurité apparaît vous demandant d'activer les macros. Cliquez sur "Activer le contenu" pour permettre l'exécution des macros VBA contenues dans le fichier.

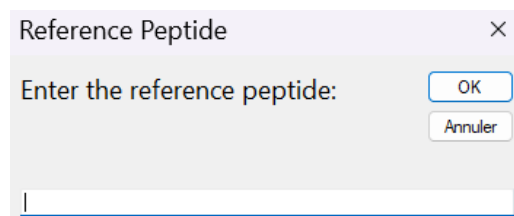
Cliquez sur l'onglet "Développeur" dans le ruban Excel. Si cet onglet n'est pas visible, vous devez l'activer en allant dans "Fichier" > "Options" > "Personnaliser le ruban" et cocher la case "Développeur".

Dans l'onglet "Développeur", cliquez sur "Macros" pour ouvrir la boîte de dialogue des macros.

Sélectionnez "NeedleWunschAlgorithm" dans la liste et cliquez sur "Exécuter".



Après avoir lancé la macro, une boîte de dialogue s'ouvre demandant de saisir un peptide de référence. Un peptide de référence est une séquence de nucléotides ou d'acides aminés utilisée comme modèle pour comparer et aligner d'autres séquences. Cette référence permet de comprendre les similarités structurales ou fonctionnelles.



Entrez les lettres A, C, G ou T (adénine, cytosine, guanine, thymine pour des séquences d'ADN, par exemple) correspondant aux peptides. Ces lettres peuvent être saisies en majuscules ou en minuscules et doivent être entrées à la suite sans espace.

Une seconde boîte de dialogue vous demandera ensuite de saisir le peptide de travail. Le peptide de travail est la séquence qui sera comparée au peptide de référence pour évaluer les similitudes et les différences. Cette analyse est nécessaire aux études de mutations, à l'évolution des espèces, ou à la recherche médicale.

Saisissez le peptide de travail en suivant les mêmes instructions que pour le peptide de référence.

Une fois les données saisies, la macro traite les informations et les résultats s'affichent automatiquement dans la première feuille du classeur Excel nommée "NeedleWunsch".

- Colonne A : Affiche le peptide de référence.
- Colonne B : Montre le peptide de travail.
- Colonne C : Détaille les alignements des deux peptides.
- Autres Colonnes : Présentent les résultats détaillés de l'analyse effectuée par l'algorithme de Needleman-Wunsch détaillé plus haut.

Ce processus permet de comparer et d'analyser des séquences biologiques directement depuis Excel, en exploitant la puissance du VBA pour automatiser et simplifier des tâches complexes en bioinformatique. Nous reviendrons sur le l'algorithme caché derrière la macro un peu plus tard.

IV) Algorithme Complet :

Voici l'algorithme caché derrière la macro :

```
Sub NeedlemanWunschAlgorithm()
    Dim refPeptide As String
    Dim workPeptide As String
    Dim scoreMatrix() As Integer
    Dim tracebackMatrix() As String
    Dim blosum62(20, 20) As Integer
    Dim aminoAcids As String
    Dim gapPenalty As Integer
    Dim i As Integer, j As Integer
    Dim scoreDiag, scoreUp, scoreLeft, maxScore As Integer
    Dim ws As Worksheet
    Dim alignA As String, alignB As String
    Dim temp As String
    ' Set gap penalty
    gapPenalty = -4
```

```

' Initialize amino acids order (according to BLOSUM62 matrix provided)
aminoAcids = "ARNDCQEGHILKMFPSTWYV"

' Load BLOSUM62 matrix from the worksheet
Set ws = ThisWorkbook.Sheets("blosum")
For i = 1 To 20
    For j = 1 To 20
        blosum62(i, j) = ws.Cells(i + 1, j + 1).Value
    Next j
Next i

' Ask for peptides
refPeptide = InputBox("Enter the reference peptide:", "Reference Peptide")
workPeptide = InputBox("Enter the work peptide:", "Work Peptide")

' Write peptides and labels in the NeedlemanWunsch worksheet
Set ws = ThisWorkbook.Sheets("NeedlemanWunsch")
ws.Cells.Clear
ws.Cells(1, 1).Value = "Ref Peptide" ' Added line for "Ref Peptide" label
ws.Cells(1, 2).Value = "Work Peptide" ' Added line for "Work Peptide" label

For i = 1 To Len(refPeptide)
    ws.Cells(i + 1, 1).Value = Mid(refPeptide, i, 1)
Next i

For i = 1 To Len(workPeptide)
    ws.Cells(i + 1, 2).Value = Mid(workPeptide, i, 1)
Next i

' Initialize matrices
ReDim scoreMatrix(Len(refPeptide) + 1, Len(workPeptide) + 1)
ReDim tracebackMatrix(Len(refPeptide) + 1, Len(workPeptide) + 1)

' Initialize first row and column of the score matrix
For i = 0 To Len(refPeptide)
    scoreMatrix(i, 0) = i * gapPenalty
    tracebackMatrix(i, 0) = "?"
Next i
For j = 0 To Len(workPeptide)
    scoreMatrix(0, j) = j * gapPenalty
    tracebackMatrix(0, j) = "?"
Next j
tracebackMatrix(0, 0) = " "

```

```

' Fill in the score and traceback matrices
For i = 1 To Len(refPeptide)
    For j = 1 To Len(workPeptide)
        scoreDiag = scoreMatrix(i - 1, j - 1) + Score(aminoAcids, blosum62,
Mid(refPeptide, i, 1), Mid(workPeptide, j, 1))
        scoreUp = scoreMatrix(i - 1, j) + gapPenalty
        scoreLeft = scoreMatrix(i, j - 1) + gapPenalty
        maxScore = Application.WorksheetFunction.Max(scoreDiag, scoreUp,
scoreLeft)
        scoreMatrix(i, j) = maxScore

' Determine traceback path
If maxScore = scoreDiag Then
    tracebackMatrix(i, j) = "?"
Elseif maxScore = scoreUp Then
    tracebackMatrix(i, j) = "?"
Else
    tracebackMatrix(i, j) = "?"
End If
    Next j
Next i

' Traceback and create alignment
i = Len(refPeptide)
j = Len(workPeptide)
alignA = ""
alignB = ""

While i > 0 And j > 0
    temp = tracebackMatrix(i, j)
    If temp = "?" Then
        alignA = Mid(refPeptide, i, 1) & alignA
        alignB = Mid(workPeptide, j, 1) & alignB
        i = i - 1
        j = j - 1
    Elseif temp = "-" Then
        alignA = Mid(refPeptide, i, 1) & alignA
        alignB = "-" & alignB
        i = i - 1
    Else
        alignA = "-" & alignA
        alignB = Mid(workPeptide, j, 1) & alignB
        j = j - 1
    End If
End While

```



```

        End If
    Wend

    ' Output alignment and score matrix
    ws.Cells(1, 3).Value = "Alignment:"
    ws.Cells(2, 3).Value = alignA
    ws.Cells(3, 3).Value = alignB

    For i = 0 To UBound(scoreMatrix, 1) - 1
        For j = 0 To UBound(scoreMatrix, 2) - 1
            ws.Cells(i + 1, j + 4).Value = scoreMatrix(i, j)
        Next j
    Next i
End Sub

Function Score(aminoAcids As String, matrix() As Integer, a As String, b As String)
    As Integer
    Dim posA As Integer
    Dim posB As Integer

    posA = InStr(aminoAcids, a)
    posB = InStr(aminoAcids, b)

    If posA > 0 And posB > 0 Then
        Score = matrix(posA, posB)
    Else
        Score = 0 ' In case of an error or unknown amino acid
    End If
End Function

```

V) Explication pas à pas de l'Algorithme :

Le programme VBA appelé "NeedlemanWunschAlgorithm" utilise l'algorithme de Needleman-Wunsch pour aligner des séquences de peptides, et il est structuré en plusieurs sections principales que l'on va expliquer en détail.

Déclaration et Initialisation des Variables :

Le programme commence par déclarer et initialiser des variables utilisées tout au long du processus :

```
Sub NeedlemanWunschAlgorithm()
```

```

Dim refPeptide As String
Dim workPeptide As String
Dim scoreMatrix() As Integer
Dim tracebackMatrix() As String
Dim blosum62(20, 20) As Integer
Dim aminoAcids As String
Dim gapPenalty As Integer
Dim i As Integer, j As Integer
Dim scoreDiag, scoreUp, scoreLeft, maxScore As Integer
Dim ws As Worksheet
Dim alignA As String, alignB As String
Dim temp As String
'''

```

Cette section de code prépare le terrain pour l'exécution de l'algorithme. Les variables comme `refPeptide` et `workPeptide` sont utilisées pour stocker les séquences de peptides entrées par l'utilisateur. `scoreMatrix` et `tracebackMatrix` sont des tableaux dynamiques qui stockeront respectivement les scores calculés et les directions pour le chemin de retour. `blosum62` est un tableau fixe qui contiendra les scores de substitution basés sur la matrice BLOSUM62, la matrice biologique standard utilisée pour l'alignement de séquences de protéines. `ws` représente la feuille de calcul dans Excel où les résultats seront écrits.

Configuration des Pénalités et Chargement de la Matrice BLOSUM62 :

```

' Set gap penalty
gapPenalty = -4

' Initialize amino acids order (according to BLOSUM62 matrix provided)
aminoAcids = "ARNDCQEGHILKMFPSTWYV"

' Load BLOSUM62 matrix from the worksheet
Set ws = ThisWorkbook.Sheets("blosum")
For i = 1 To 20
    For j = 1 To 20
        blosum62(i, j) = ws.Cells(i + 1, j + 1).Value
    Next j
Next i
'''

```

La pénalité de gap, définie ici à -4, est utilisée pour pénaliser les alignements avec des espaces, ce qui est courant dans l'alignement des séquences biologiques pour gérer les insertions et les suppressions. La chaîne `aminoAcids` contient les codes à une lettre des acides aminés, correspondant à l'ordre utilisé dans la matrice

BLOSUM62 pour déterminer les scores de substitution. La matrice BLOSUM62 est chargée à partir d'une feuille nommée "blosum" dans le même classeur. Cette matrice est essentielle pour calculer les scores entre chaque paire d'acides aminés lors de l'alignement.

Saisie des Peptides par l'Utilisateur et Initialisation des Feuilles :

```
' Ask for peptides
refPeptide = InputBox("Enter the reference peptide:", "Reference Peptide")
workPeptide = InputBox("Enter the work peptide:", "Work Peptide")

' Write peptides and labels in the NeedlemanWunsch worksheet
Set ws = ThisWorkbook.Sheets("NeedlemanWunsch")
ws.Cells.Clear
ws.Cells(1, 1).Value = "Ref Peptide" ' Added line for "Ref Peptide" label
ws.Cells(1, 2).Value = "Work Peptide" ' Added line for "Work Peptide" label

For i = 1 To Len(refPeptide)
    ws.Cells(i + 1, 1).Value = Mid(refPeptide, i, 1)
Next i

For i = 1 To Len(workPeptide)
    ws.Cells(i + 1, 2).Value = Mid(workPeptide, i, 1)
Next i
...
```

Ici, l'utilisateur est invité à entrer les séquences de peptides de référence et de travail via des boîtes de dialogue. Ces séquences sont ensuite écrites dans une feuille de calcul nommée "NeedlemanWunsch". Chaque caractère des peptides est écrit dans une cellule distincte sur la feuille pour préparer l'alignement.

Initialisation des Matrices de Scores et de Traçage :

La partie suivante du code initialise les matrices de scores et de traçage qui sont essentielles pour l'application de l'algorithme de Needleman-Wunsch :

```
' Initialize matrices
ReDim scoreMatrix(Len(refPeptide) + 1, Len(workPeptide) + 1)
ReDim tracebackMatrix(Len(refPeptide) + 1, Len(workPeptide) + 1)

' Initialize first row and column of the score matrix
For i = 0 To Len(refPeptide)
    scoreMatrix(i, 0) = i * gapPenalty
    tracebackMatrix(i, 0) = "?"
```

```

Next i
For j = 0 To Len(workPeptide)
    scoreMatrix(0, j) = j * gapPenalty
    tracebackMatrix(0, j) = "?"
Next j
tracebackMatrix(0, 0) = " "
'''

```

Dans cette section, les matrices `scoreMatrix` et `tracebackMatrix` sont redimensionnées en fonction de la longueur des peptides de référence et de travail. La première ligne et la première colonne de `scoreMatrix` sont initialisées pour tenir compte des pénalités de gap accumulées lorsqu'un peptide est aligné avec des gaps depuis le début. La matrice `tracebackMatrix` est utilisée pour enregistrer la direction du meilleur chemin lors du calcul de l'alignement ; ici, elle est initialisée avec des valeurs par défaut.

Calcul des Scores et Détermination du Chemin de Retour :

```

' Fill in the score and traceback matrices
For i = 1 To Len(refPeptide)
    For j = 1 To Len(workPeptide)
        scoreDiag = scoreMatrix(i - 1, j - 1) + Score(aminoAcids, blosum62,
Mid(refPeptide, i, 1), Mid(workPeptide, j, 1))
        scoreUp = scoreMatrix(i - 1, j) + gapPenalty
        scoreLeft = scoreMatrix(i, j - 1) + gapPenalty
        maxScore = Application.WorksheetFunction.Max(scoreDiag, scoreUp,
scoreLeft)
        scoreMatrix(i, j) = maxScore

        ' Determine traceback path
        If maxScore = scoreDiag Then
            tracebackMatrix(i, j) = "?"
        ElseIf maxScore = scoreUp Then
            tracebackMatrix(i, j) = "?"
        Else
            tracebackMatrix(i, j) = "?"
        End If
    Next j
Next i
'''

```

Ici, chaque cellule de la `scoreMatrix` est remplie en calculant les scores possibles pour un alignement diagonal, vers le haut, ou vers la gauche, en utilisant la matrice BLOSUM62 pour obtenir les scores de substitution entre les acides aminés. La

fonction `Score` calcule le score de substitution basé sur la position des acides aminés dans la chaîne `aminoAcids`. Le score maximum parmi les trois directions est sélectionné pour chaque position, et la direction correspondante est enregistrée dans `tracebackMatrix` pour tracer ultérieurement l'alignement optimal.

Traçage de l'Alignement Optimal et Affichage des Résultats :

```
' Traceback and create alignment
i = Len(refPeptide)
j = Len(workPeptide)
alignA = ""
alignB = ""

While i > 0 And j > 0
    temp = tracebackMatrix(i, j)
    If temp = "?" Then
        alignA = Mid(refPeptide, i, 1) & alignA
        alignB = Mid(workPeptide, j, 1) & alignB
        i = i - 1
        j = j - 1
    ElseIf temp = "-" Then
        alignA = Mid(refPeptide, i, 1) & alignA
        alignB = "-" & alignB
        i = i - 1
    Else
        alignA = "-" & alignA
        alignB = Mid(workPeptide, j, 1) & alignB
        j = j - 1
    End If
Wend

' Output alignment and score matrix
ws.Cells(1, 3).Value = "Alignment:"
ws.Cells(2, 3).Value = alignA
ws.Cells(3, 3).Value = alignB

For i = 0 To UBound(scoreMatrix, 1) - 1
    For j = 0 To UBound(scoreMatrix, 2) - 1
        ws.Cells(i + 1, j + 4).Value = scoreMatrix(i, j)
    Next j
Next i
End Sub
'''
```

Cette dernière partie effectue le traçage à rebours à partir de la fin des peptides pour déterminer l'alignement optimal, en utilisant les directions stockées dans `tracebackMatrix`. L'alignement résultant est construit en ajoutant progressivement des acides aminés ou des gaps. Une fois l'alignement complet, il est écrit dans la feuille Excel avec les scores calculés pour chaque position, permettant ainsi une visualisation directe du résultat.

VI) Notice :

Cette notice vous guide à travers les paramètres modifiables dans le programme VBA NeedlemanWunschAlgorithm, qui est utilisé pour aligner des séquences de peptides. Vous apprendrez à ajuster les conditions de calcul pour répondre à vos besoins spécifiques en matière d'analyse de séquences.

1. Gap Penalty (`gapPenalty`)

- **Définition** : La pénalité de gap est un score négatif appliqué pour chaque espace introduit dans l'alignement des séquences. Elle pénalise l'extension d'un gap dans l'une des séquences, ce qui aide à contrôler la fréquence et la longueur des gaps dans l'alignement final.
- **Valeur par défaut** : -4
- **Comment la modifier** : Ouvrez le module VBA dans Excel où le code est écrit. Localisez la ligne où `gapPenalty` est définie (`gapPenalty = -4`). Changez le chiffre -4 par la valeur souhaitée. Par exemple, pour rendre les gaps moins pénalisants, vous pourriez utiliser -2.
- **Impact de la modification** : Augmenter la valeur (moins négative) rend les gaps moins pénalisants, ce qui peut augmenter leur nombre dans l'alignement final. Diminuer la valeur (plus négative) rend les gaps plus pénalisants et peut conduire à des alignements avec moins de gaps. Choisir la bonne pénalité de gap dépend de l'importance relative des insertions et des suppressions dans votre étude particulière.

2. Matrice de substitution (`blosum62`)

- **Définition** : La matrice de substitution BLOSUM62 est utilisée pour scorer les substitutions entre différents acides aminés dans l'alignement. Chaque élément de la matrice fournit un score pour remplacer un acide aminé par un autre.
- **Comment la modifier** : Accédez à la feuille Excel nommée "blosum" dans votre classeur. Modifiez les valeurs directement dans cette feuille pour ajuster les scores de substitution entre les acides aminés. Chaque ligne et chaque colonne correspond à un acide aminé spécifique, en ordre avec la chaîne `aminoAcids`.

- **Impact de la modification** : Modifier les scores de substitution peut influencer significativement l'alignement en changeant la manière dont les acides aminés sont comparés. Par exemple, si une mutation particulière est courante ou biologiquement pertinente, vous pouvez ajuster les scores pour rendre cette substitution moins pénalisante.

3. Ordre des acides aminés (`aminoAcids`)

- **Définition** : Une chaîne qui représente l'ordre des acides aminés utilisés dans la matrice de substitution.
- **Comment la modifier** : Ouvrez le module VBA et localisez la ligne où `aminoAcids` est définie. Modifiez la chaîne pour refléter l'ordre désiré des acides aminés selon votre besoin.
- **Impact de la modification** : Changer l'ordre des acides aminés nécessite également d'ajuster la matrice de substitution `blosum62` pour que les scores correspondent correctement aux paires d'acides aminés. Cela est généralement nécessaire uniquement si vous travaillez avec une matrice personnalisée ou si les acides aminés standard ne conviennent pas à votre analyse.

En modifiant ces variables, vous pouvez finement contrôler comment les peptides sont alignés selon votre analyse biologique ou bioinformatique spécifique. Assurez-vous de comprendre les implications biologiques de chaque modification pour obtenir les résultats les plus informatifs.

VII) Retour d'expérience utilisateur :

L'utilisation de l'algorithme Needleman-Wunsch implémenté en VBA pour l'alignement des séquences de peptides a été testée afin de souligner les aspects positifs ainsi que les défis de l'outil.

La mise en place initiale s'est déroulée avec une relative facilité. Après avoir ouvert le fichier Excel contenant le code VBA et activé les macros, il est nécessaire d'avoir une certaine familiarité avec l'environnement Excel et le ruban de développeur pour naviguer dans le setup. Le processus commence par la saisie des séquences de peptides à comparer, facilitée par des boîtes de dialogue claires et simples.

Parmi les aspects positifs, l'interface utilisateur simplifiée et la visualisation directe des résultats dans le classeur Excel permettent une analyse rapide et efficace des alignements. La capacité de modifier la pénalité de gap et les scores de la matrice BLOSUM62 offre une personnalisation pour adapter l'analyse aux besoins spécifiques du projet.

Cependant, l'algorithme ne peut prendre en charge des peptides ayant des séquences de plus de 255 caractères dû aux limitations des boîtes de dialogues VBA, ce qui pourrait poser problème dans des analyses de grande envergure. Enfin, l'utilisation de l'environnement VBA peut être moins accessible pour ceux moins familiers avec Excel, comparativement à d'autres outils programmables comme Python.

En conclusion, l'expérience avec l'algorithme Needleman-Wunsch en VBA a été globalement positive, particulièrement grâce à l'intégration directe avec Excel qui simplifie la manipulation et l'analyse des données, elle montre tout de même des limites dans certains contextes.

VIII) Retour d'expérience programmeur :

L'implémentation de l'algorithme Needleman-Wunsch en VBA pour Excel présente des aspects à la fois bien exécutés et des opportunités d'amélioration.

Points Forts de l'Implémentation Actuelle :

L'une des plus grandes forces de cette implémentation est son intégration native avec Excel, un environnement familier pour de nombreuses personnes. Cela facilite la manipulation, l'analyse et la visualisation des données d'alignement sans nécessiter des logiciels spécialisés.

L'usage de boîtes de dialogue pour la saisie des séquences de peptides rend également le programme accessible à tous.

Enfin, la possibilité de modifier les valeurs de la pénalité de gap et d'ajuster les scores de substitution à l'aide de la matrice BLOSUM62 offre aux utilisateurs la flexibilité de personnaliser les calculs selon leurs besoins spécifiques de recherche.

Possibles Améliorations :

L'algorithme pourrait connaître une refonte pour améliorer sa vitesse de calcul en utilisant des structures de données plus efficaces ou l'implémentation en utilisant un langage de programmation plus rapide comme C++ ou Python.

Le programme actuel assume également que les entrées sont correctes et ne gère pas les erreurs liées aux entrées invalides. Ajouter des contrôles pour vérifier la validité des acides aminés saisis et gérer les erreurs de manière proactive améliorerait la robustesse de l'application.

IX) Conclusion :

En conclusion, l'étude de l'alignement de séquences à l'aide de l'algorithme de Needleman-Wunsch implémenté en VBA pour Excel a démontré l'importance de ces

outils en bioinformatique pour l'analyse comparative des séquences protéiques et nucléiques. L'approche utilisée a priorisé l'accès et la manipulation des données d'alignement, en tirant parti de l'environnement familier d'Excel, ce qui est particulièrement avantageux pour les chercheurs moins expérimentés en programmation.

Le programme VBA développé offre une bonne base pour l'alignement de séquences jusqu'à 255 caractères, avec la capacité d'ajuster les paramètres de scoring selon des besoins spécifiques grâce à la personnalisation de la pénalité de gap ou encore des scores de la matrice BLOSUM62.

L'outil actuel s'est alors avéré efficace pour des applications spécifiques et il pourrait évoluer vers des solutions plus robustes et rapides, en explorant d'autres langages de programmation ou en optimisant les structures de données. Cette adaptation permettrait de répondre aux exigences croissantes de la bioinformatique moderne et d'assurer une analyse plus précise et approfondie des données biologiques.