

Rapport de TP1-2 MCS

Member de groupe: MA Huan(travail tout seul)

Travail réalisé:

1, fonction [D, g, h]=CalculDistanceDTW(sequence1 , sequence2 , distance , r)

C'est la version modifiée pour résoudre les question 1.2 et 1.3.

le paramètre r est un entier qui décide la distance pour contrainte globale, et h est une matrice binaire pour stocker tous les chemins possibles.

La fonction AfficheChemins(g) utilise cette matrice binaire pour chercher le chemin optimal.

Une fonction recursive est utilisé dans AfficheChemins.

Pour question 1.1 et 1.4, le résultat sont testé ci-dessous.

2, En utilisant la fonction `CalculDistanceDTW` et `distance_vect`, je remplis la matrice de confusion avec les distances calculé de chacun référence de mot. Le prémier linges laisse vide pour mot l'orgine(ici, M01).

Pour trouver les fichiers reconnu, je declare un seuil de 10, donc tous les distances inférieur 10 sont reconnu, et puis diviser par le nombre de fichiers total pour calculer le taux de reconnaissance(mais je suis pas sûr est-ce que c'est correcte).

Jeux de test:

Tous les jeux de test sont réalisés dans Octave.

Les tests utilisent les différentes type de distance(ADN, numérique, son, pour la question 1.1) qui se satisfont des chacun exerce TD(pour la question 1.4).

Un test de graphique sortie de les fichiers vocales sont aussi fourni ci-dessous.

Pour question 2, un image copie-écran des résultat sont fourni.

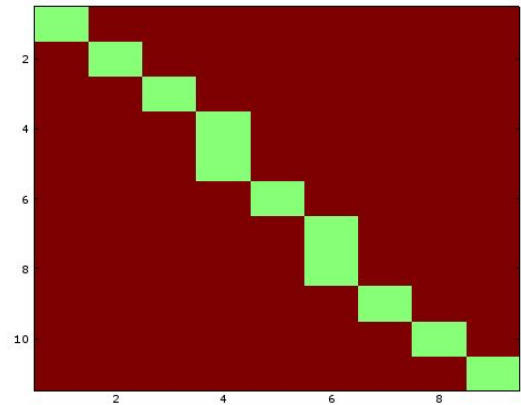
test de ADN et matrice binaire qui stocke les chemins possibles

```
>> s1
s1 = ATGGTACGTC
>> s2
s2 = AAGTAGGC
>> [D, g, k] = CalculDistanceDTW(s1, s2, 'distance_adn', 5)
D = 0.22222
g =
    0    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
    Inf    0    0    1    2    2    3    Inf    Inf
    Inf    1    1    2    1    2    3    4    Inf
    Inf    2    2    1    2    3    2    2    3
    Inf    3    3    1    2    3    2    2    3
    Inf    4    4    2    1    2    3    3    4
    Inf    4    4    3    2    1    2    3    4
    Inf    Inf    5    4    3    2    3    4    3
    Inf    Inf    Inf    4    4    3    2    3
    Inf    Inf    Inf    Inf    4    4    3    4
    Inf    Inf    Inf    Inf    Inf    5    4    3

k =
    1    0    0    0    0    0    0    0    0
    0    1    1    0    0    0    0    0    0
    0    0    1    0    0    0    0    0    0
    0    0    0    1    0    0    0    0    0
    0    0    0    1    0    0    0    0    0
    0    0    0    0    1    0    0    0    0
    0    0    0    0    0    1    0    0    0
    0    0    0    0    0    1    0    0    0
    0    0    0    0    0    0    1    1    0
    0    0    0    0    0    0    0    1    0
    0    0    0    0    0    0    0    0    0
```

une difference de contrainte globale et image de chemin optimal

```
>> [D, g,k] = CalculDistanceDTW(s1, s2, 'distance_adn',4)
D = Inf
g =
    0    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
   Inf     0     0     1     2     2    Inf    Inf    Inf
   Inf     1     1     2     1     2     3    Inf    Inf
   Inf     2     2     1     2     3     2     2    Inf
   Inf     3     3     1     2     3     2     2     3
   Inf     4     4     2     1     2     3     3     4
   Inf    Inf     4     3     2     1     2     3     4
   Inf    Inf    Inf     4     3     2     3     4     3
   Inf    Inf    Inf    Inf     4     3     2     2     3
   Inf    Inf    Inf    Inf    Inf     4     3     3     4
   Inf    Inf    Inf    Inf    Inf    Inf     4     4     3
```



test numerique et son chemin stockée

```
>> s3
s3 =
    -2    10   -10    15   -13    20    -5    14     2

>> s4
s4 =
     3   -13    14    -7     9    -2

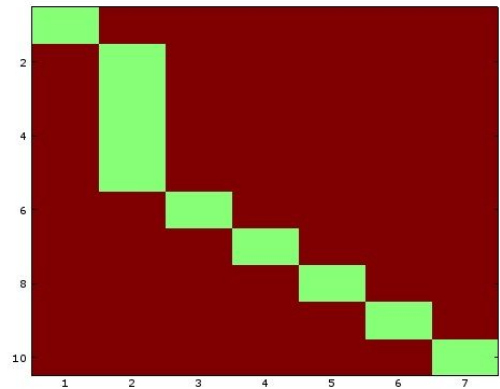
>> [D, g,h] = CalculDistanceDTW(s3, s4, 'distance_num',3)
D = 2.8000
g =
    0    Inf    Inf    Inf    Inf    Inf    Inf
   Inf    10    21    37    42    Inf    Inf
   Inf    17    40    29    46    44    Inf
   Inf    30    23    47    35    54    60
   Inf    42    51    25    47    47    64
   Inf    Inf    42    52    37    59    69
   Inf    Inf    Inf    54    64    59    81
   Inf    Inf    Inf    Inf    58    72    65
   Inf    Inf    Inf    Inf    Inf    68    81
   Inf    Inf    Inf    Inf    Inf    Inf    76

h =
     1     0     0     0     0     0     0
     0     1     0     0     0     0     0
     0     1     0     0     0     0     0
     0     1     0     0     0     0     0
     0     1     0     0     0     0     0
     0     0     1     0     0     0     0
     0     0     0     1     0     0     0
     0     0     0     0     1     0     0
     0     0     0     0     0     1     0
     0     0     0     0     0     0     1
```

difference de contrainte global et image de chemin

```
g =
    0    Inf    Inf    Inf    Inf    Inf    Inf
   Inf    10    21    37    Inf    Inf    Inf
   Inf    17    40    29    46    Inf    Inf
   Inf    30    23    47    35    54    Inf
   Inf    Inf    51    25    47    47    64
   Inf    Inf    Inf    52    37    59    69
   Inf    Inf    Inf    Inf    64    59    81
   Inf    Inf    Inf    Inf    Inf    73    65
   Inf    Inf    Inf    Inf    Inf    Inf    81
   Inf    Inf    Inf    Inf    Inf    Inf    Inf

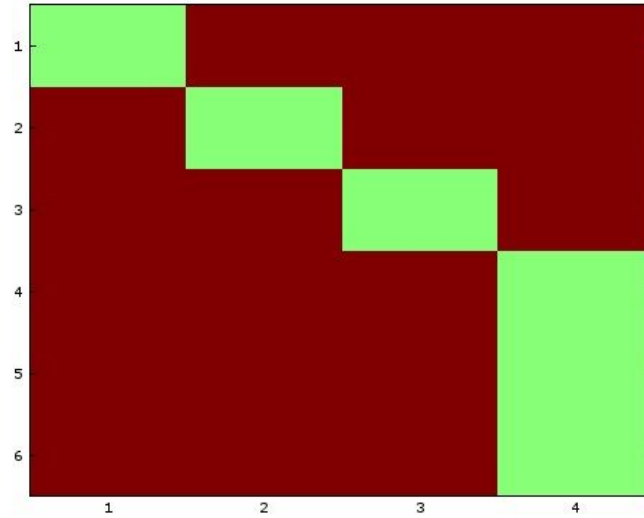
h =
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     1
```



test de son

```
<stanceDTW(s5, S6, 'distance_son',2)
D = 0
g =
    0    Inf    Inf    Inf
    Inf    0     2     4
    Inf     2     0     0
    Inf     4     0     0
    Inf    Inf     2     2
    Inf    Inf    Inf     4

h =
    1    0    0    0
    0    1    0    0
    0    0    1    1
    0    0    0    1
    0    0    0    1
    0    0    0    1
```



différence de contrainte:

```
<stanceDTIW(s5, S6, 'distance_son',1)
D = 0
g =
    0    Inf    Inf    Inf
    Inf    0     2    Inf
    Inf     2     0     0
    Inf    Inf     0     0
    Inf    Inf    Inf     2
    Inf    Inf    Inf    Inf

h =
    0     0     0     0
    0     0     0     0
    0     0     0     0
    0     0     0     0
    0     0     0     0
    0     0     0     1
```

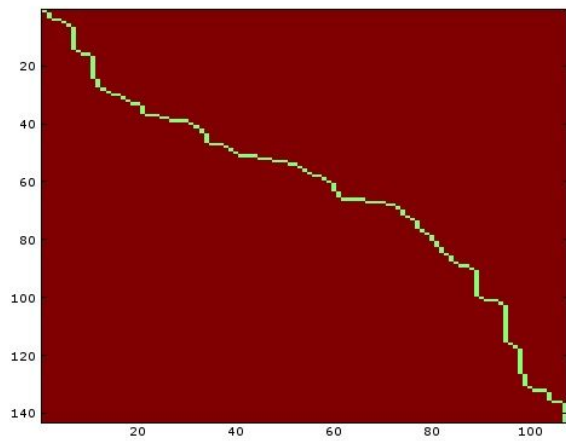


Image de test fichiers vocales(M01 et M02 'arretetoi' bruité):

test de reconnaissance vocales:

```
>> RecoVocale
  0.00000  0.00000  0.00000  0.00000  0.00000
55.47767  2.73769  2.71972  3.36725  8.02265
42.05458 10.61005 18.98825  3.09003  5.73873
11.93973 14.25246  6.48483  6.10941 30.76233
 9.85737 18.60909  1.52581  1.15930  5.23941
 7.59067 15.27781      Inf  3.41003 12.93852
 7.75616  8.58312  0.24280  3.43114  6.04300
11.77636 13.22614  8.77821  6.50363 17.07888
 8.17596  8.12462  2.80405  3.34209  6.65048
11.12520 17.91440 50.32478 19.19330 14.59230
 0.09214  9.92104  4.41920 20.16571  0.40668
17.20532 16.20541 10.71373  8.02206  8.24995
      Inf 15.05019  1.89761 29.79422 13.40971
Taux de reconnaissance : 57.051%
```

Analyse:

La contrainte globale ne change pas les résultats quand:

soit une matrice DTW de taille (m,n) , la contrainte globale est r , donc

$$r \geq \min(m,n)|2 + 1$$

par contre, si r inférieur $\min(m,n)|2 + 1$, le résultat de distance dtw devient infini, et il échec à trouver la matrice stocké les chemins.

Chercher le chemin optimal en utilisant la matrice binaire qui stocke les chemins possibles:

Cette matrice binaire stocke tous les cas minimum comme un chemin et noté ce cas avec 1, tous les restes rempli avec 0. Pour trouver un chemin optimal et sortir son graph, je crée un matrice pré-rempli avec infini et suis les chemins stocké dans la matrice binaire, en modifier les cases de chemins à valeur 0. Quand il y a un multi-choix, le position diagonale est toujours le premier choix si possible, considérant le distance euclidienne.

Comparaison entre M01 et M02(Ligne2 dans la matrice de confusion):

On peut voir qu'il y a qu'une valeur qui sont trop grand que les autres, donc dans colonne 1, il y a 55, et les autres sont inférieur que 10. Le mot de colonne1, 'arretetoi' est la seule erreur on cherche.