

# Modélisation en Biologie

Algorithmes de Smith-Waterman et de Gotoh pour  
l'alignement de séquences

Nicolas Thierry-Mieg

CNRS / TIMC-IMAG / BCM, Grenoble, France

Merci à Jacques Van Helden, Université d'Aix-Marseille, pour ses slides

ENSIMAG 2A 2016-2017

- Objectif: identifier les similarités entre deux séquences
  - Séquences d'ADN: ATGC + N
  - Séquences peptidiques == protéines: ACDEF...Y (20 acides aminés)

- Alignement global

- L'alignement final inclut obligatoirement les deux séquences complètes.

LQGPSKGTGKGS-SRSWDN

|----|--|||---|--|-

LN-ITKSAGKGAIMRLGDA

- Approprié, par exemple, pour les protéines homologues qui sont conservées sur toute leur longueur.
- Algorithme: ***Needleman-Wunsch*** (1970).

- Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48, 443-53.
- Smith, T. F. & Waterman, M. S. (1981). Identification of common molecular subsequences. J Mol Biol 147, 195-7.

## ■ Alignement global

- L'alignement final inclut obligatoirement les deux séquences complètes.

```
LQGPSKGTGKGS-SRSWDN
|----|--|||---|--|-
LN-ITKSAGKGAIMRLGDA
```

- Approprié, par exemple, pour les protéines homologues qui sont conservées sur toute leur longueur.
- Algorithme: **Needleman-Wunsch** (1970).

## ■ Alignement local

```
LQGPSKGTGKGS-SSRIWDN
|---|
LN-ITKAGKGAIMRLGDA
```

L'alignement final est restreint aux segments conservés.

```
KTGKG
|---|
KAGKG
```

- Approprié, par exemple, pour les protéines qui partagent un domaine commun, restreint à un segment de chaque séquence.
- Algorithme: **Smith-Waterman** (1981).

- Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48, 443-53.
- Smith, T. F. & Waterman, M. S. (1981). Identification of common molecular subsequences. J Mol Biol 147, 195-7.

- Objectif: identifier les similarités entre deux séquences
  - ▢ Séquences d'ADN: ATGC + N
  - ▢ Séquences peptidiques == protéines: ACDEF...Y (20 acides aminés)

GATTACA et TATA?

GATTAcA

| | | |

---TAtA

GATTACA et ATAC?

GATTACA

| | | |

-AT-AC-

- Objectif: identifier les similarités entre deux séquences
  - Séquences d'ADN: ATGC + N
  - Séquences peptidiques == protéines: ACDEF...Y (20 acides aminés)

GATTACA et TATA?

GATTAcA

| | | |

---TAtA

GATTACA et ATAC?

GATTACA

| | | |

-AT-AC-

- Score/coût d'une substitution, par exemple pour l'ADN: match +1, mismatch -1
- Insertions-délétions (indels), modèle linéaire de coût: -2 par '-'
- On cherche des alignements dont le score est maximal
- Alignement global: l'intégralité des deux séquences

- Objectif: identifier les similarités entre deux séquences
  - Séquences d'ADN: ATGC + N
  - Séquences peptidiques == protéines: ACDEF...Y (20 acides aminés)

GATTACA et TATA?

GATTAcA

| | | |

---TAtA

**Score?**

GATTACA et ATAC?

GATTACA

| | | |

-AT-AC-

**Score?**

- Score/coût d'une substitution, par exemple pour l'ADN: match +1, mismatch -1
- Insertions-délétions (indels), modèle linéaire de coût: -2 par '-'
- On cherche des alignements dont le score est maximal
- Alignement global: l'intégralité des deux séquences

- Objectif: identifier les similarités entre deux séquences
  - Séquences d'ADN: ATGC + N
  - Séquences peptidiques == protéines: ACDEF...Y (20 acides aminés)

GATTACA et TATA?

GATTAcA

| | | |

---TAtA

Score (global): -4

GATTACA et ATAC?

GATTACA

| | | |

-AT-AC-

Score (global): -2

- Score/coût d'une substitution, par exemple pour l'ADN: match +1, mismatch -1
- Insertions-délétions (indels), modèle linéaire de coût: -2 par '-'
- On cherche des alignements dont le score est maximal
- Alignement global: l'intégralité des deux séquences



- Objectif: identifier les similarités entre deux séquences
  - Séquences d'ADN: ATGC + N
  - Séquences peptidiques == protéines: ACDEF...Y (20 acides aminés)

GATTACA et TATA?

GATTAcA

| | | |

---TAtA

Score (global): -4

GATTACA et ATAC?

GATTACA

| | | |

-AT-AC-

Score (global): -2

- Score/coût d'une substitution, par exemple pour l'ADN: match +1, mismatch -1
- Insertions-délétions (indels), modèle linéaire de coût: -2 par '-'
- On cherche des alignements dont le score est maximal
- Alignement global: l'intégralité des deux séquences
- Alignement semi-global: ignorer les indels aux extrémités

- Objectif: identifier les similarités entre deux séquences
  - Séquences d'ADN: ATGC + N
  - Séquences peptidiques == protéines: ACDEF...Y (20 acides aminés)

GATTACA et TATA?

GATTAcA

| | | |

---TAtA

Score (global): -4

Score (semi-global): +2

GATTACA et ATAC?

GATTACA

| | | |

-AT-AC-

Score (global): -2

Score (semi-global): +2

- Score/coût d'une substitution, par exemple pour l'ADN: match +1, mismatch -1
- Insertions-délétions (indels), modèle linéaire de coût: -2 par '-'
- On cherche des alignements dont le score est maximal
- Alignement global: l'intégralité des deux séquences
- Alignement semi-global: ignorer les indels aux extrémités

- Objectif: identifier les similarités entre deux séquences
  - Séquences d'ADN: ATGC + N
  - Séquences peptidiques == protéines: ACDEF...Y (20 acides aminés)

GATTACA et TATA?

GATTAcA

| | | |

---TAtA

Score (global): -4

Score (semi-global): +2

GATTACA et ATAC?

GATTACA

| | | |

-AT-AC-

Score (global): -2

Score (semi-global): +2

**D'autres alignements semi-globaux aussi bons?**

- Score/coût d'une substitution, par exemple pour l'ADN: match +1, mismatch -1
- Insertions-délétions (indels), modèle linéaire de coût: -2 par '-'
- On cherche des alignements dont le score est maximal
- Alignement global: l'intégralité des deux séquences
- Alignement semi-global: ignorer les indels aux extrémités

- Objectif: identifier les similarités entre deux séquences
  - Séquences d'ADN: ATGC + N
  - Séquences peptidiques == protéines: ACDEF...Y (20 acides aminés)

GATTACA et TATA?

GATTAcA

| | | |

---TAtA

Score (global): -4

Score (semi-global): +2

GATTACA et ATAC?

GATTACA

| | | |

-AT-AC-

Score (global): -2

Score (semi-global): +2

**D'autres alignements semi-globaux aussi bons?**

GATTACA

| | | |

-A-TAC-

Score (global): -2

Score (semi-global): +2

GAtTACA

| | | |

--aTAC-

Score (global): -4

Score (semi-global): +2

- Objectif: identifier les similarités entre deux séquences
  - Séquences d'ADN: ATGC + N
  - Séquences peptidiques == protéines: ACDEF...Y (20 acides aminés)

GATTACA et TATA?

GATTAcA

| | | |

---TAtA

Score (global): -4

Score (semi-global): +2

GATTACA et ATAC?

GATTACA

| | | |

-AT-AC-

Score (global): -2

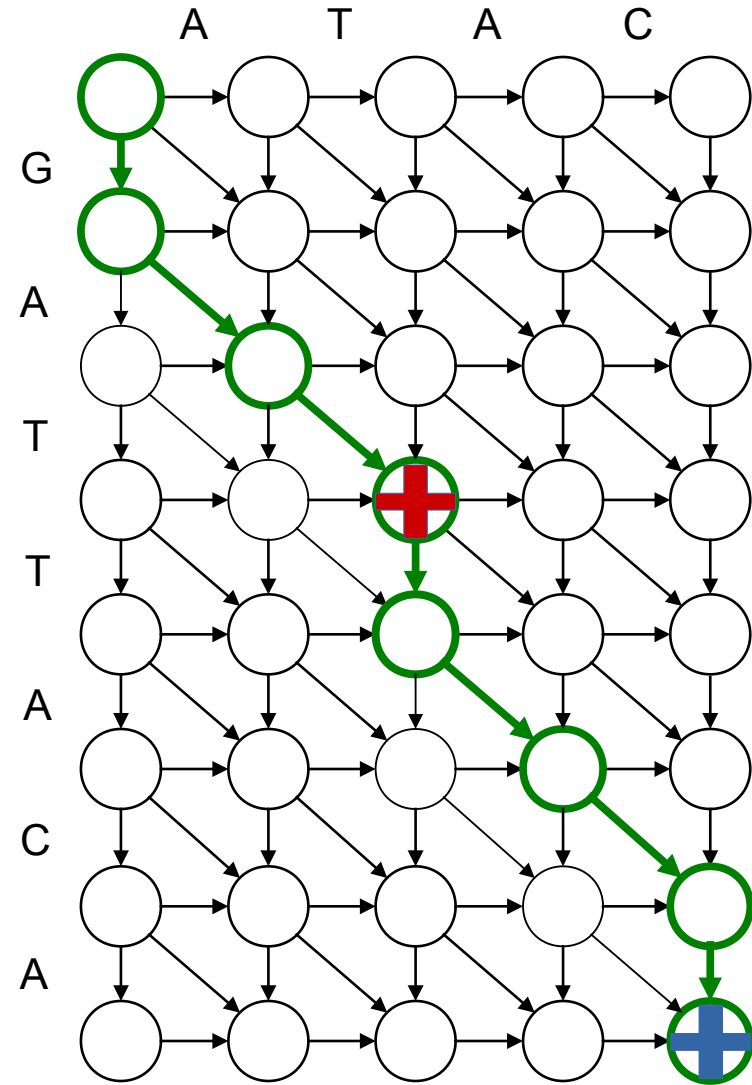
Score (semi-global): +2

- Score/coût d'une substitution, par exemple pour l'ADN: match +1, mismatch -1
- Insertions-délétions (indels), modèle linéaire de coût: -2 par '-'
- On cherche des alignements dont le score est maximal
- Alignement global: l'intégralité des deux séquences
- Alignement semi-global: ignorer les indels aux extrémités
- **Alignement local: sous-séquences**

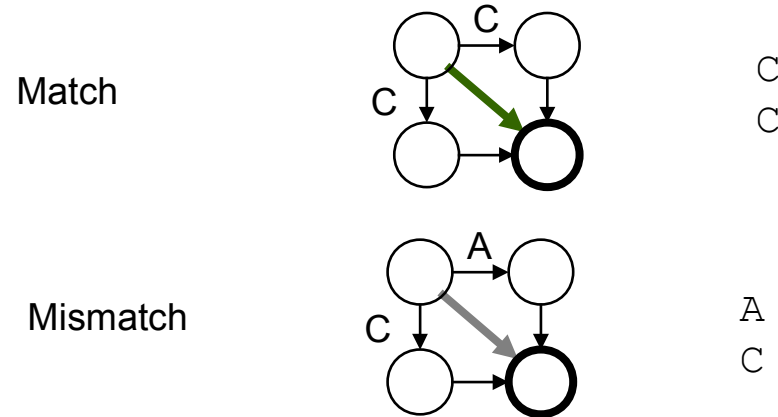
# Algorithme - programmation dynamique

Idée: remplir une matrice  $(n+1, p+1)$

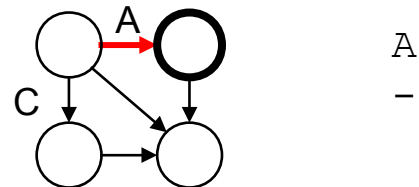
Un chemin de la case  $(0,0)$  à la case  $(i,j)$  représente un alignement des préfixes de longueurs  $i$  et  $j$  des deux séquences



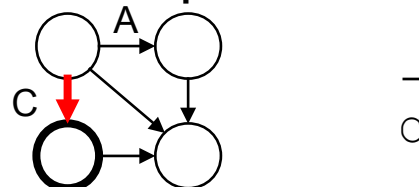
## Déplacement diagonal: substitution (match ou mismatch)



## Déplacement horizontal: insertion d'un '-' dans la séquence verticale



## Déplacement vertical: insertion d'un '-' dans la séquence horizontale



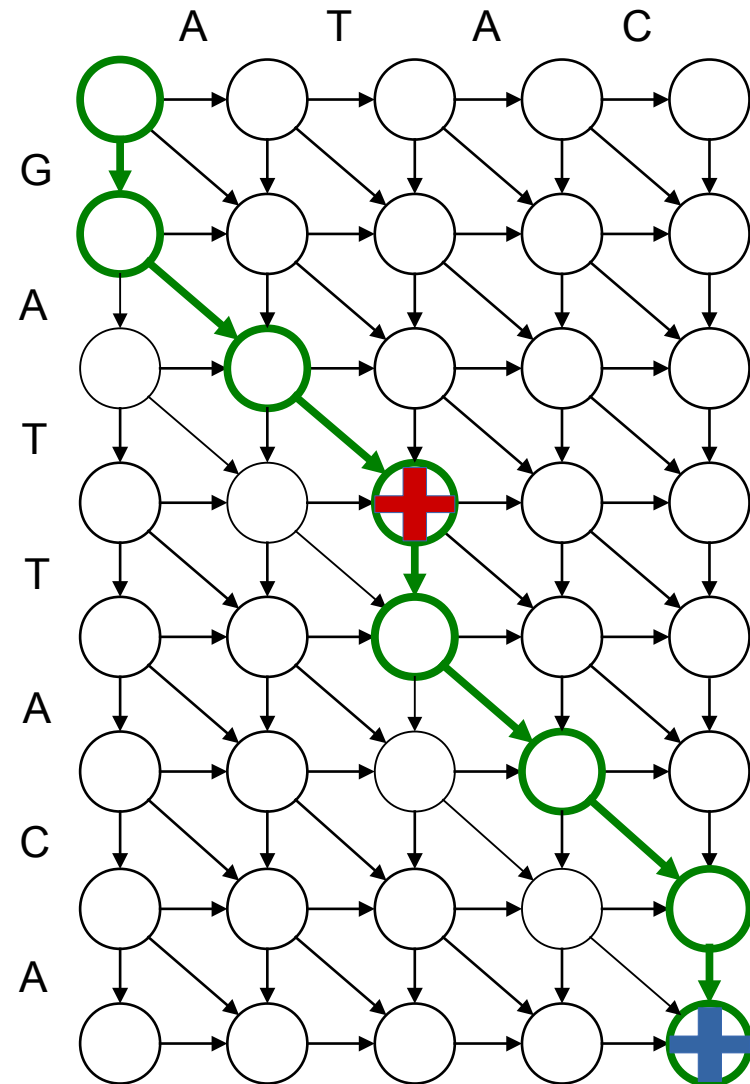
# Algorithme - programmation dynamique

Idée: remplir une matrice  $(n+1, p+1)$

Un chemin de la case  $(0,0)$  à la case  $(i,j)$  représente un alignement des préfixes de longueurs  $i$  et  $j$  des deux séquences

Exemple rouge+vert:

- A T  
  | |  
G A T





# Algorithme - programmation dynamique

Idée: remplir une matrice  $(n+1, p+1)$

Un chemin de la case  $(0,0)$  à la case  $(i,j)$  représente un alignement des préfixes de longueurs  $i$  et  $j$  des deux séquences

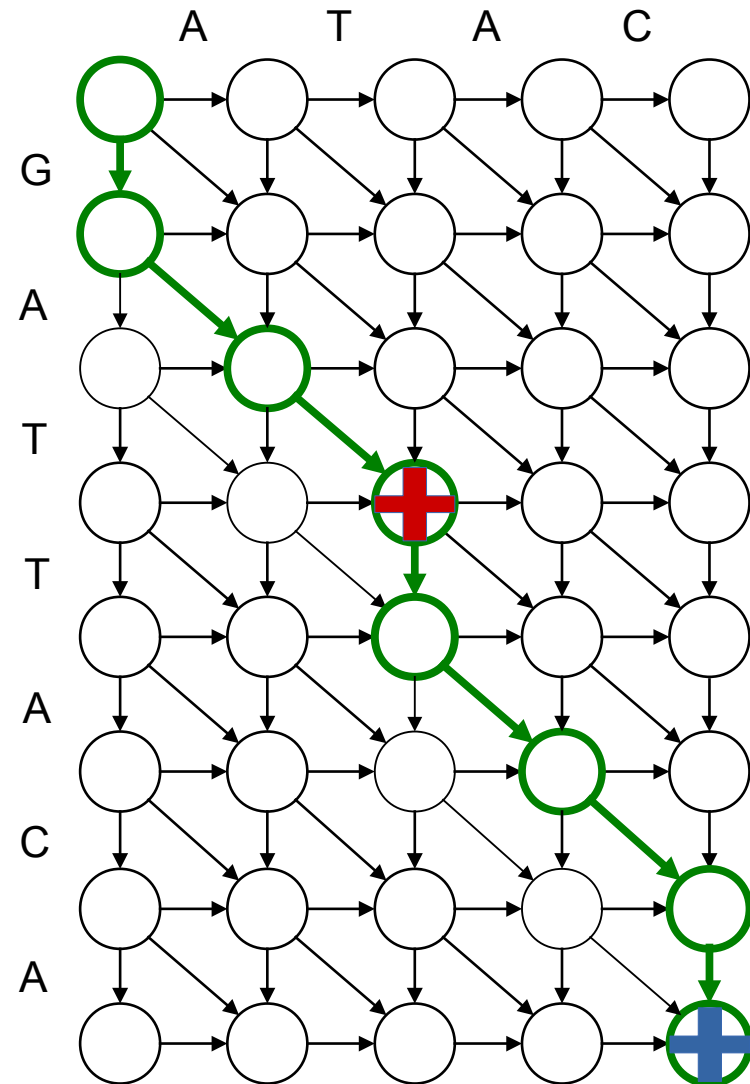
Exemple rouge+vert:

```
- A T
  | |
G A T
```

Un alignement global est un chemin du coin haut-gauche jusqu'au coin bas-droit

Exemple bleu+vert:

```
- A T - A C -
  | |   | |
G A T T A C A
```



# Algorithme - programmation dynamique

Idée: remplir une matrice  $(n+1, p+1)$

Un chemin de la case  $(0,0)$  à la case  $(i,j)$  représente un alignement des préfixes de longueurs  $i$  et  $j$  des deux séquences

Exemple rouge+vert:

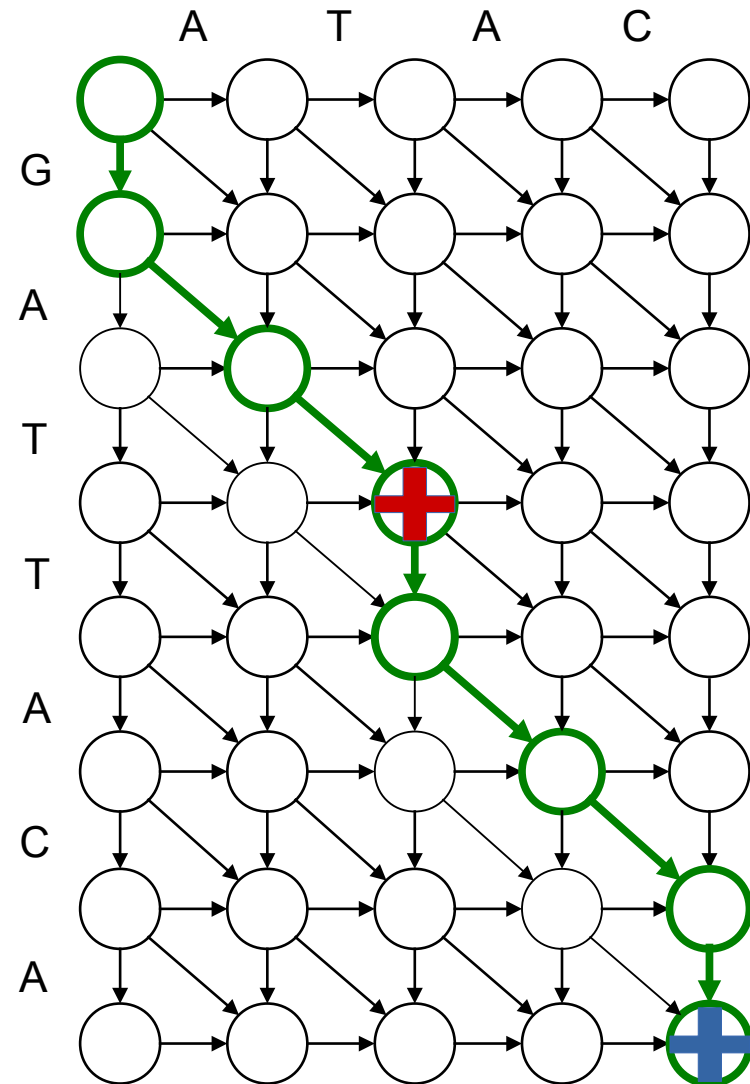
```
- A T
  | |
G A T
```

Un alignement global est un chemin du coin haut-gauche jusqu'au coin bas-droit

Exemple bleu+vert:

```
- A T - A C -
  | |   | |
G A T T A C A
```

Un alignement semi-global est...?



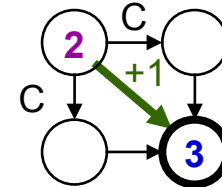
# Algorithme – scores

Chaque case contiendra le score du ou des meilleurs alignements de préfixes aboutissant à cette case

Exemple de modèle de coût:

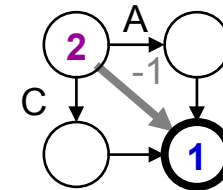
- match +1
- mismatch -1
- indel -2

Match

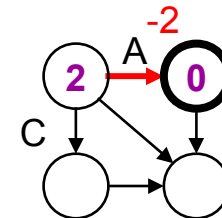


C  
C

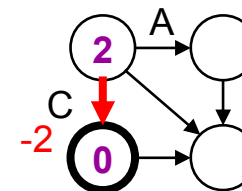
Mismatch



A  
C

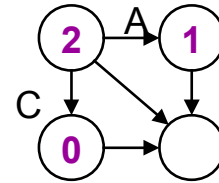
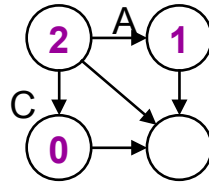
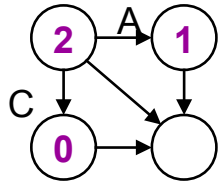


A  
-

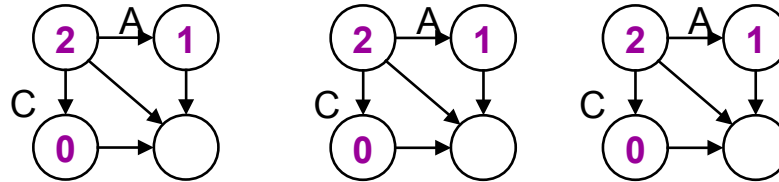


-  
C

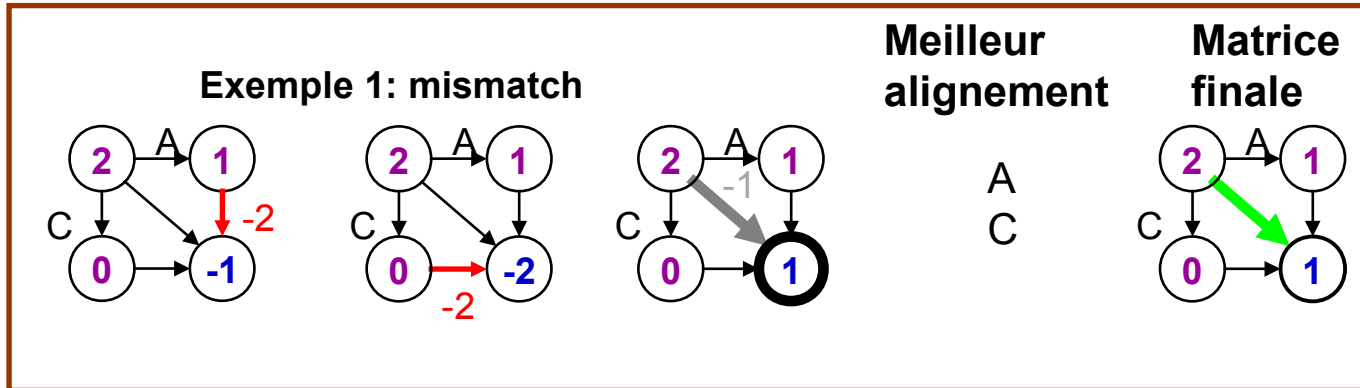
## Exemple 1



**Exemple 1**



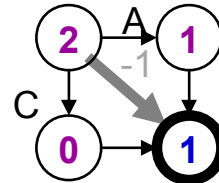
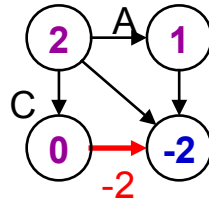
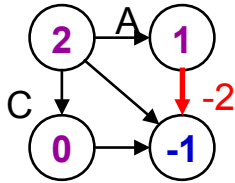
$$S(i,j) = \max \{ \begin{aligned} &S(i-1,j-1) + \text{subst}(s1[i-1],s2[j-1]) , \\ &S(i-1,j) + \text{indel} , \\ &S(i,j-1) + \text{indel} \} \end{aligned}$$



$$S(i,j) = \max \{ \begin{aligned} &S(i-1,j-1) + \text{subst}(s1[i-1],s2[j-1]) , \\ &S(i-1,j) + \text{indel} , \\ &S(i,j-1) + \text{indel} \} \end{aligned}$$

# Algorithme – scores & meilleur chemin

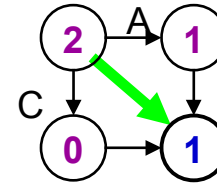
Exemple 1: mismatch



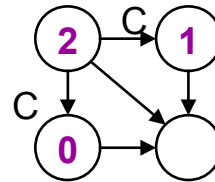
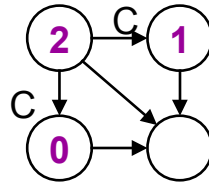
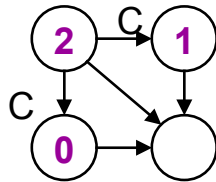
Meilleur  
alignement

A  
C

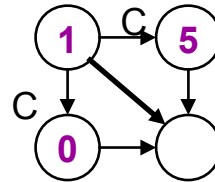
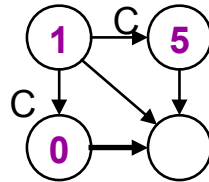
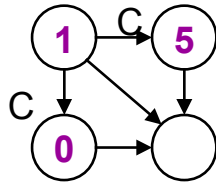
Matrice  
finale



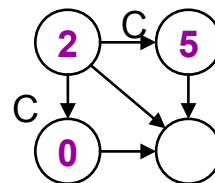
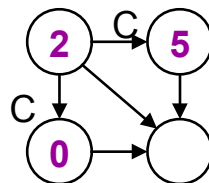
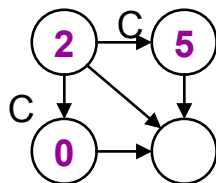
Exemple 2



Exemple 3

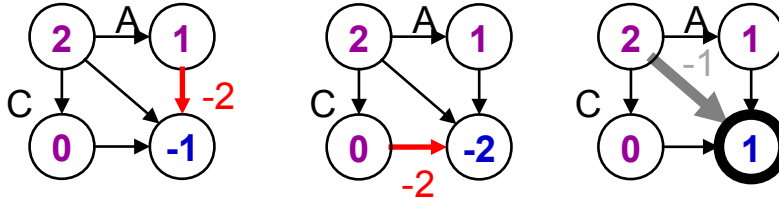


Exemple 4



# Algorithme – scores & meilleur chemin

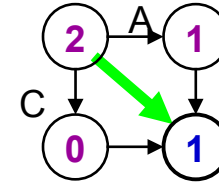
Exemple 1: mismatch



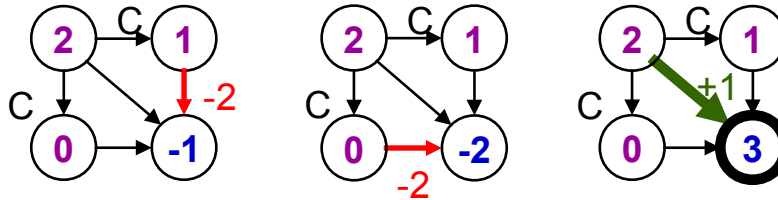
Meilleur  
alignement

A  
C

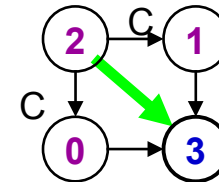
Matrice  
finale



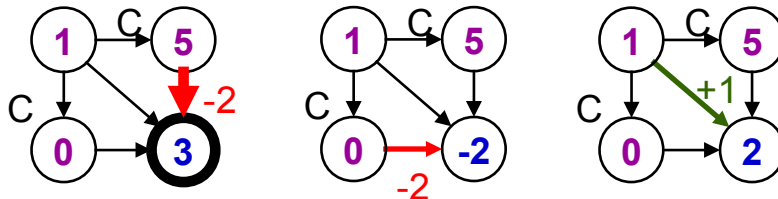
Exemple 2: match



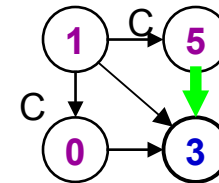
C  
C



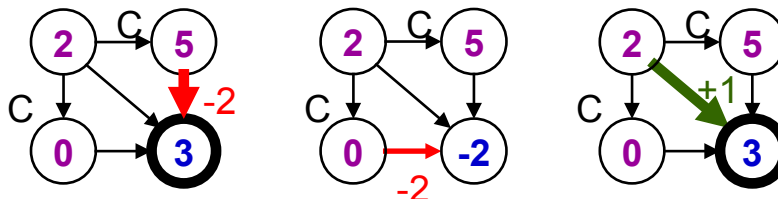
Exemple 3: indel



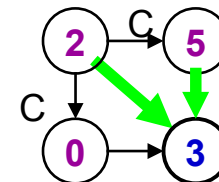
-  
C



Exemple 4: indel ou match



C  
C or -  
C

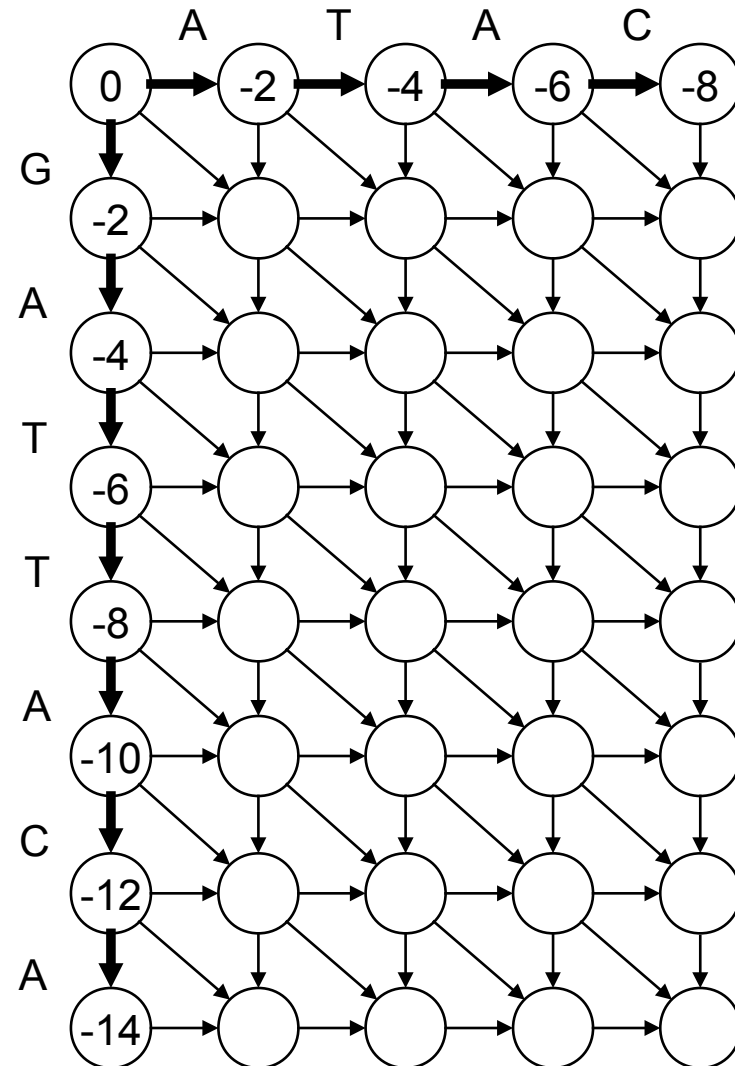




## Alignement global

Exemple de modèle de coût:

- match +1
- mismatch -1
- indel -2

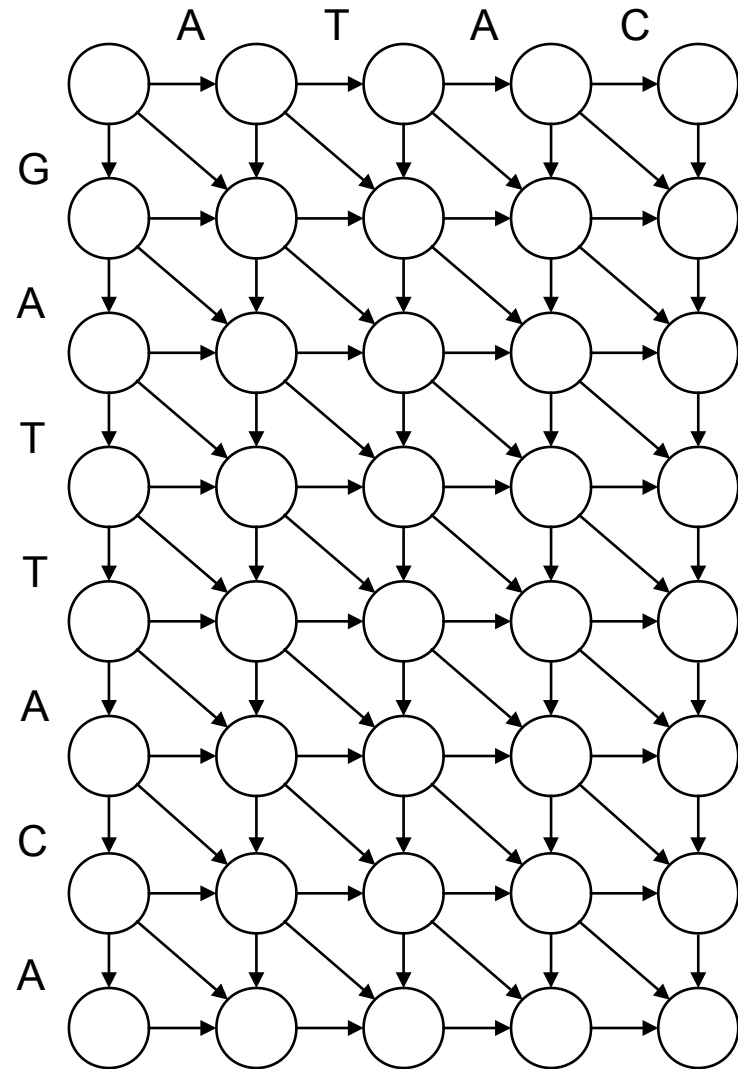


**Alignement global**

**Alignement semi-global?**

Exemple de modèle de coût:

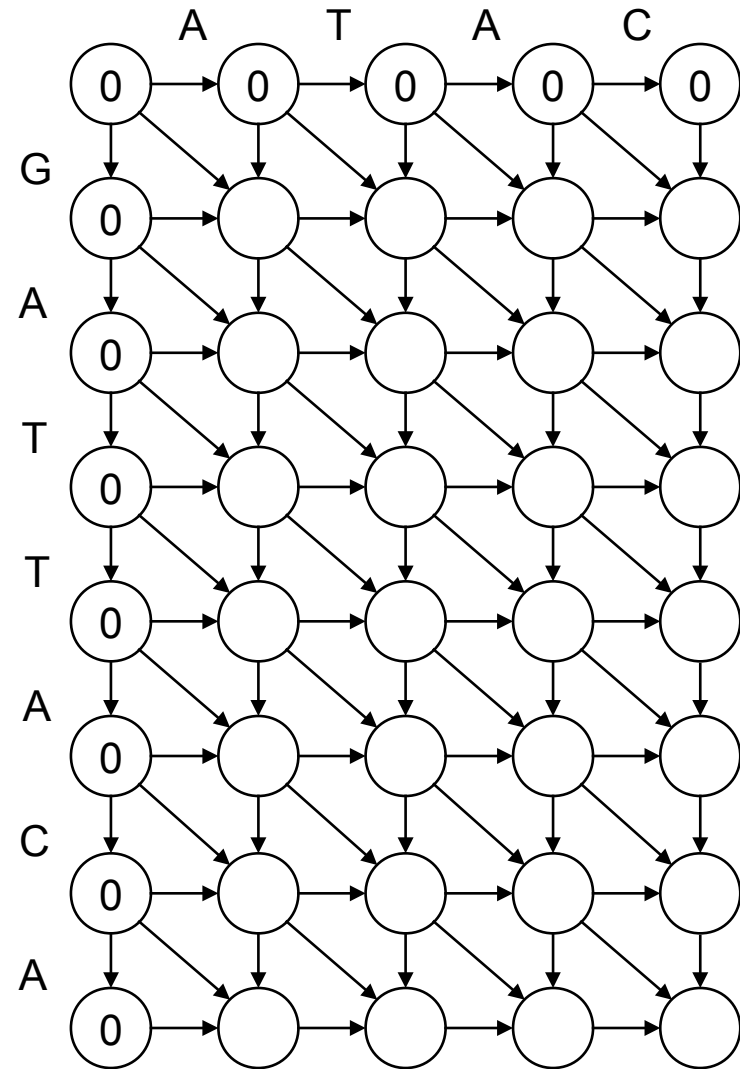
- match +1
- mismatch -1
- indel -2



## Alignement semi-global

Exemple de modèle de coût:

- match +1
- mismatch -1
- indel -2



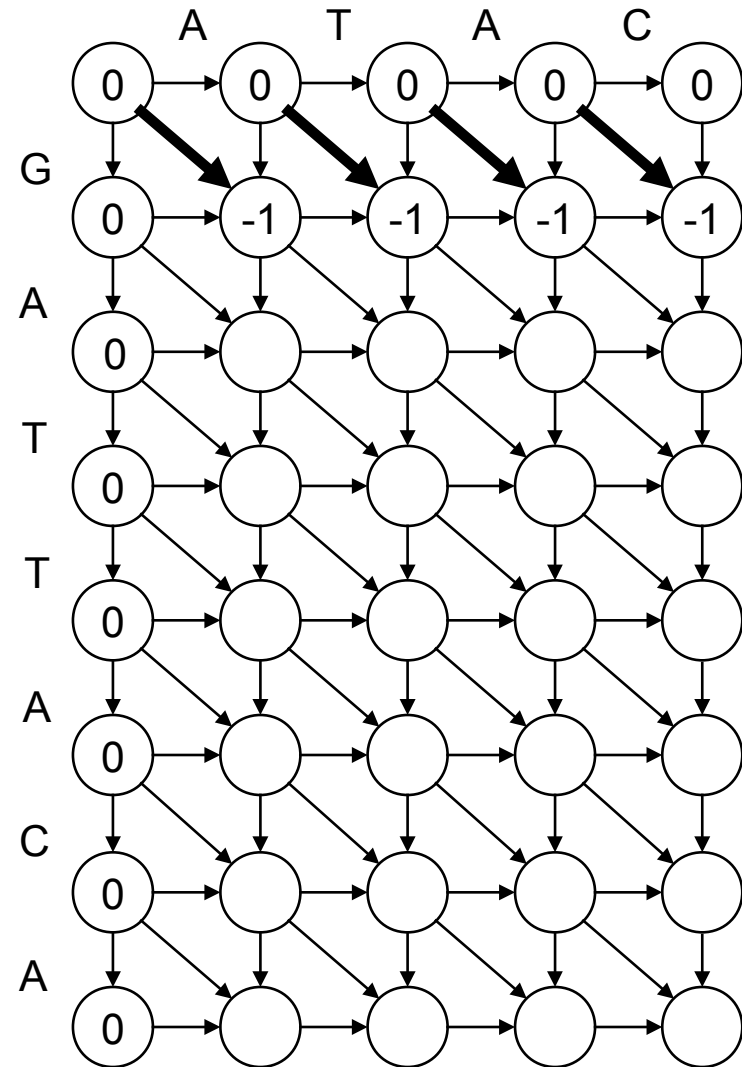
# Algorithme – remplissage de la matrice

1. Initialisation -> premières ligne et colonne

2. On remplit la seconde ligne de gauche à droite (ou la seconde colonne de haut en bas).

On retient:

- le meilleur score
- le ou les chemins qui l'ont produit



# Algorithme – remplissage de la matrice

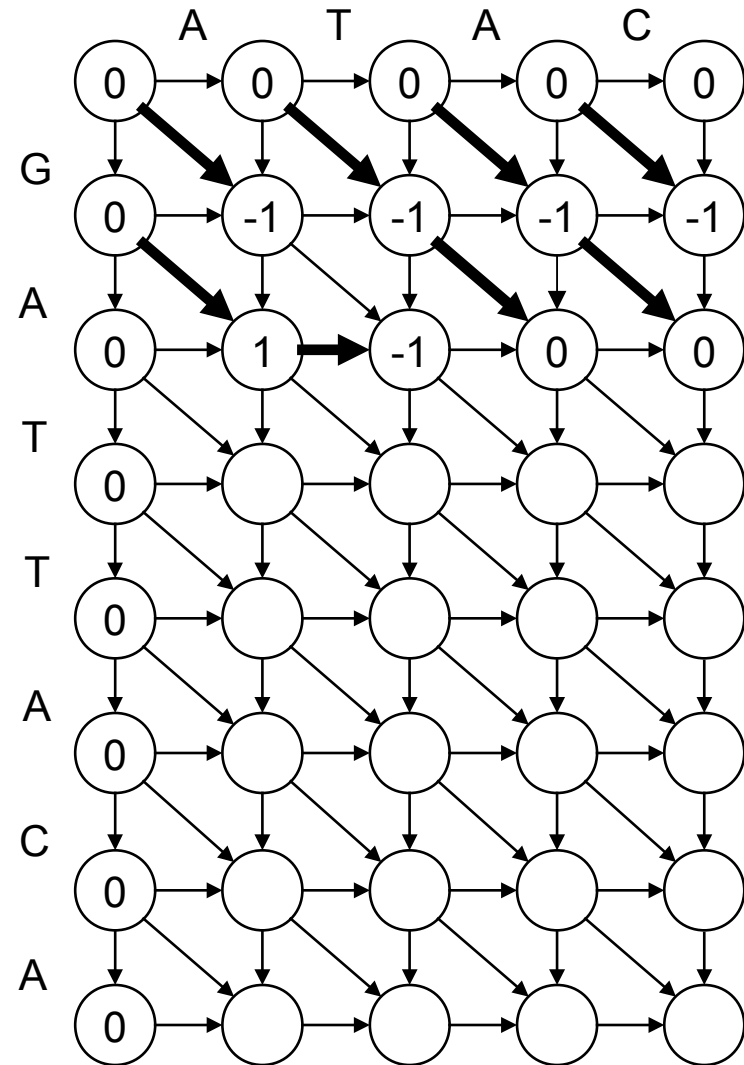
1. Initialisation -> premières ligne et colonne

2. On remplit la seconde ligne de gauche à droite (ou la seconde colonne de haut en bas).

On retient:

- le meilleur score
- le ou les chemins qui l'ont produit

3. Remplissage de la 3ème ligne: idem



# Algorithme – remplissage de la matrice

1. Initialisation -> premières ligne et colonne

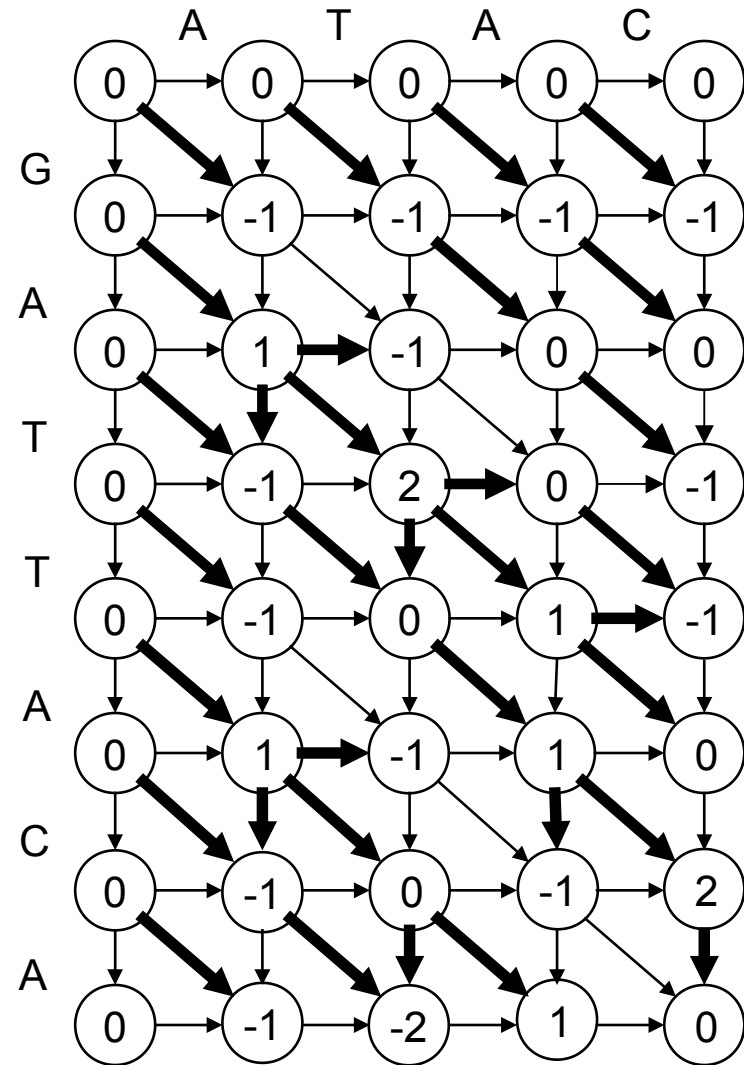
2. On remplit la seconde ligne de gauche à droite (ou la seconde colonne de haut en bas).

On retient:

- le meilleur score
- le ou les chemins qui l'ont produit

3. Remplissage de la 3ème ligne: idem

4. On itère... -> matrice remplie!



# Algorithme – remplissage de la matrice

1. Initialisation -> premières ligne et colonne

2. On remplit la seconde ligne de gauche à droite (ou la seconde colonne de haut en bas).

On retient:

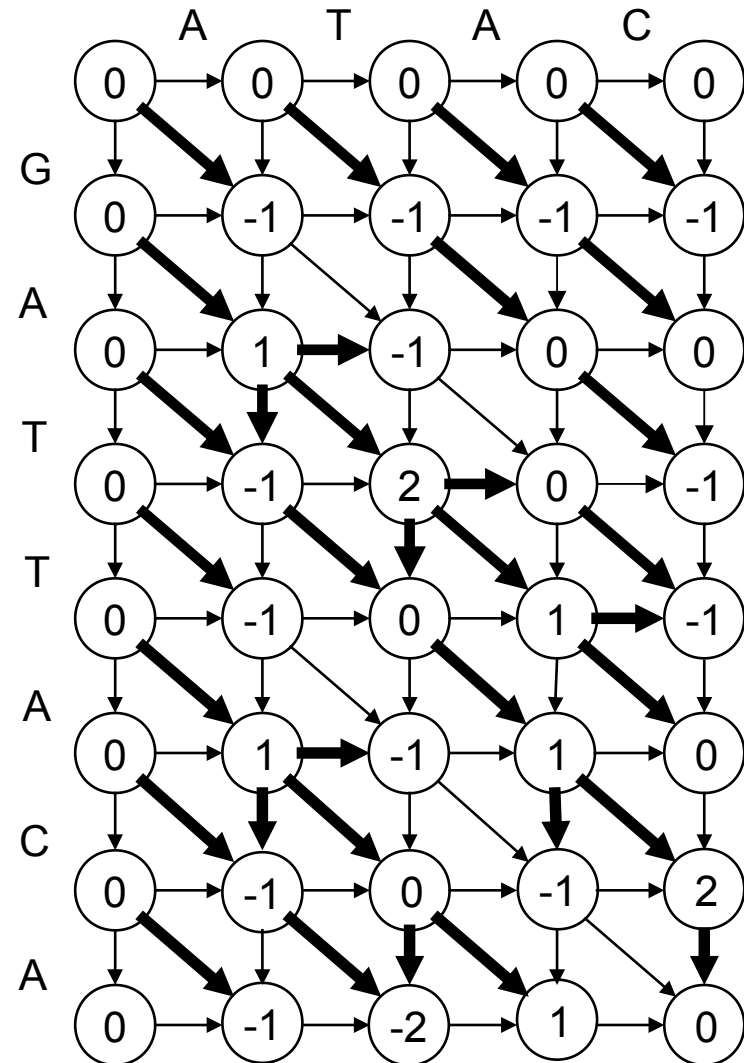
- le meilleur score
- le ou les chemins qui l'ont produit

3. Remplissage de la 3ème ligne: idem

4. On itère... -> matrice remplie!

5. Global: on part de la case bas-droite et on remonte en suivant les chemins optimaux jusqu'à atteindre la case haut-gauche

→ **les alignements optimaux**



# Algorithme – remplissage de la matrice

1. Initialisation -> premières ligne et colonne

2. On remplit la seconde ligne de gauche à droite (ou la seconde colonne de haut en bas).

On retient:

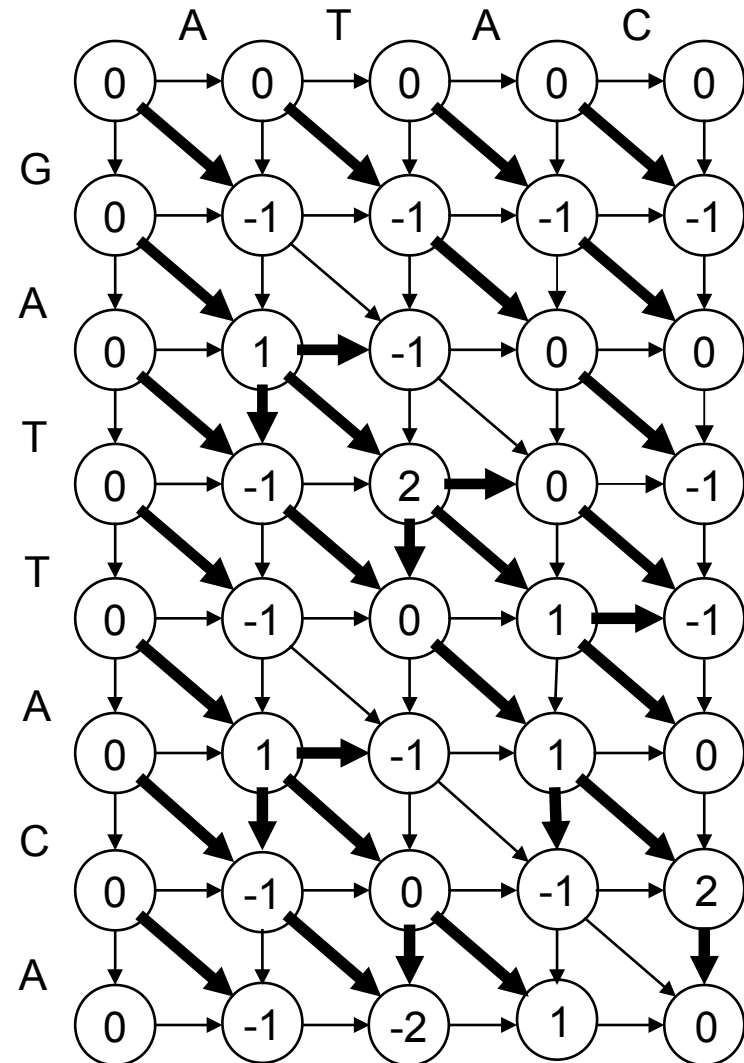
- le meilleur score
- le ou les chemins qui l'ont produit

3. Remplissage de la 3ème ligne: idem

4. On itère... -> matrice remplie!

5. Semi-global: on part de ??? et on remonte en suivant les chemins optimaux jusqu'à atteindre ???

→ les alignements optimaux





# Algorithme – remplissage de la matrice

1. Initialisation -> premières ligne et colonne

2. On remplit la seconde ligne de gauche à droite (ou la seconde colonne de haut en bas).

On retient:

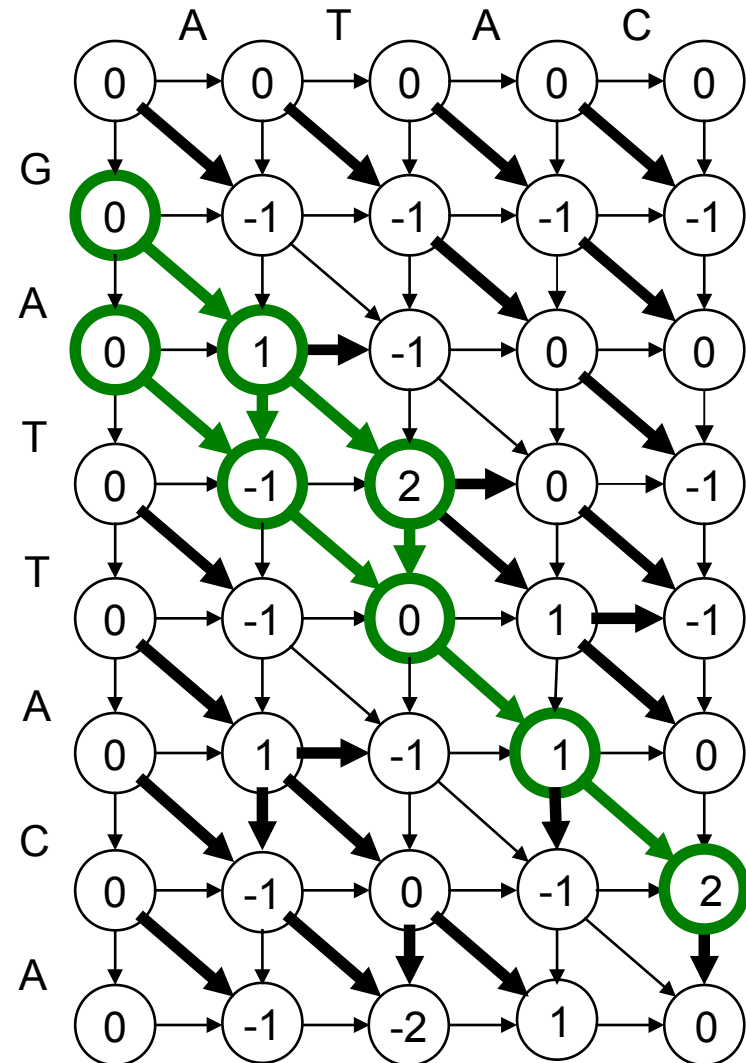
- le meilleur score
- le ou les chemins qui l'ont produit

3. Remplissage de la 3ème ligne: idem

4. On itère... -> matrice remplie!

5. Semi-global: on part de **la/les cases de score max dans la dernière ligne/colonne** et on remonte en suivant les chemins optimaux jusqu'à atteindre **une case de la première ligne/colonne**

→ **les alignements optimaux**



**GATTACA et ATAC?**

**Score (semi-global) : +2**

GATTACA

|| ||

-AT-AC-

GATTACA

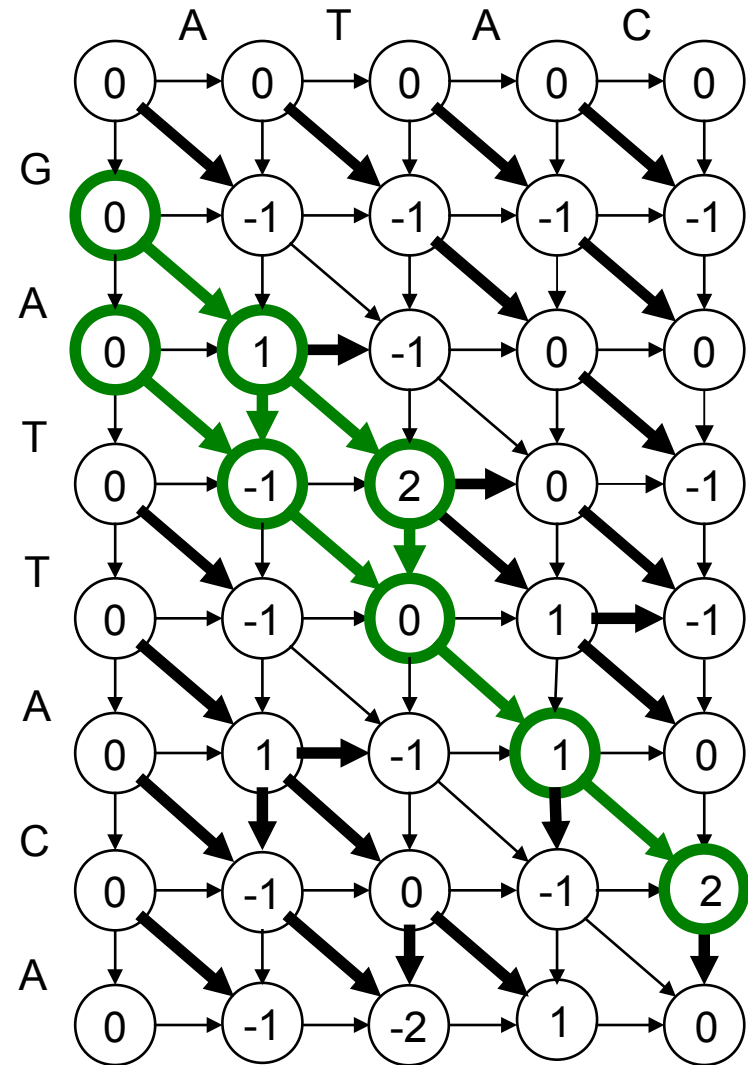
| |||

-A-TAC-

GAtTACA

||||

--aTAC-



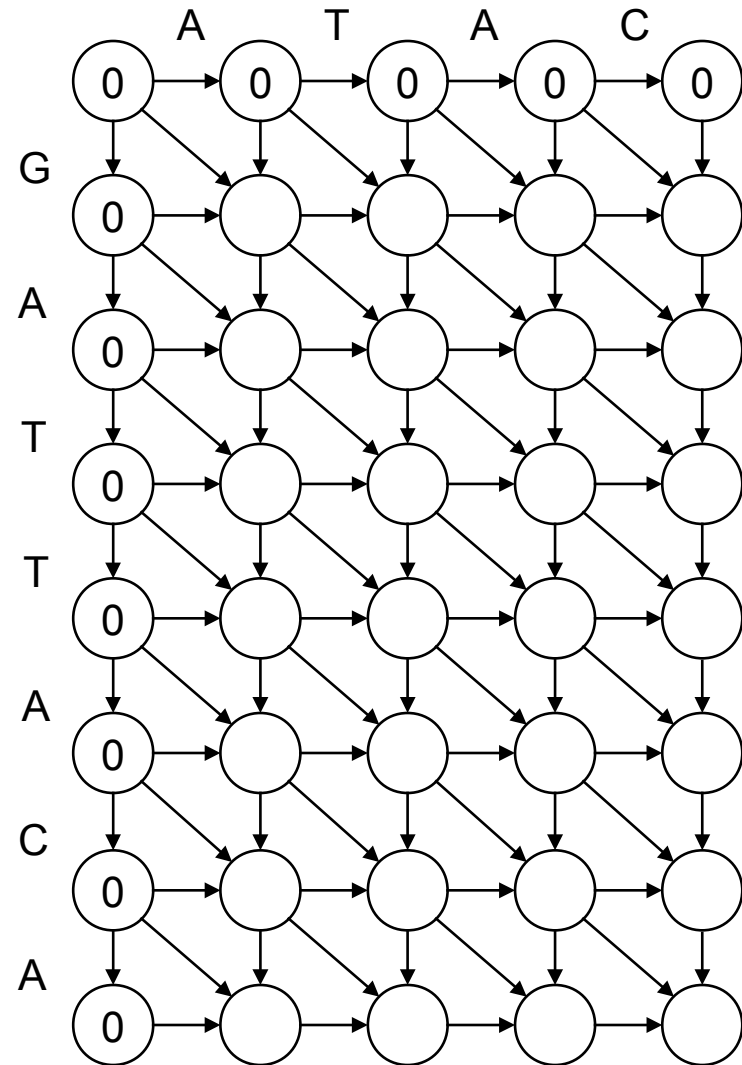
# Algorithme – remplissage de la matrice – alignement local

Alignement local: un alignement peut commencer n'importe où dans la matrice

→

Même initialisation que semi-global, et en plus tout score négatif est remplacé par zéro (sans prédécesseur: ce sera la début potentiel d'un alignement local optimal)

$$S(i,j) = \max \{ \begin{array}{l} S(i-1,j-1) + \text{subst}(s1[i-1],s2[j-1]) , \\ S(i-1,j) + \text{indel} , \\ S(i,j-1) + \text{indel} , \\ 0 \end{array} \}$$



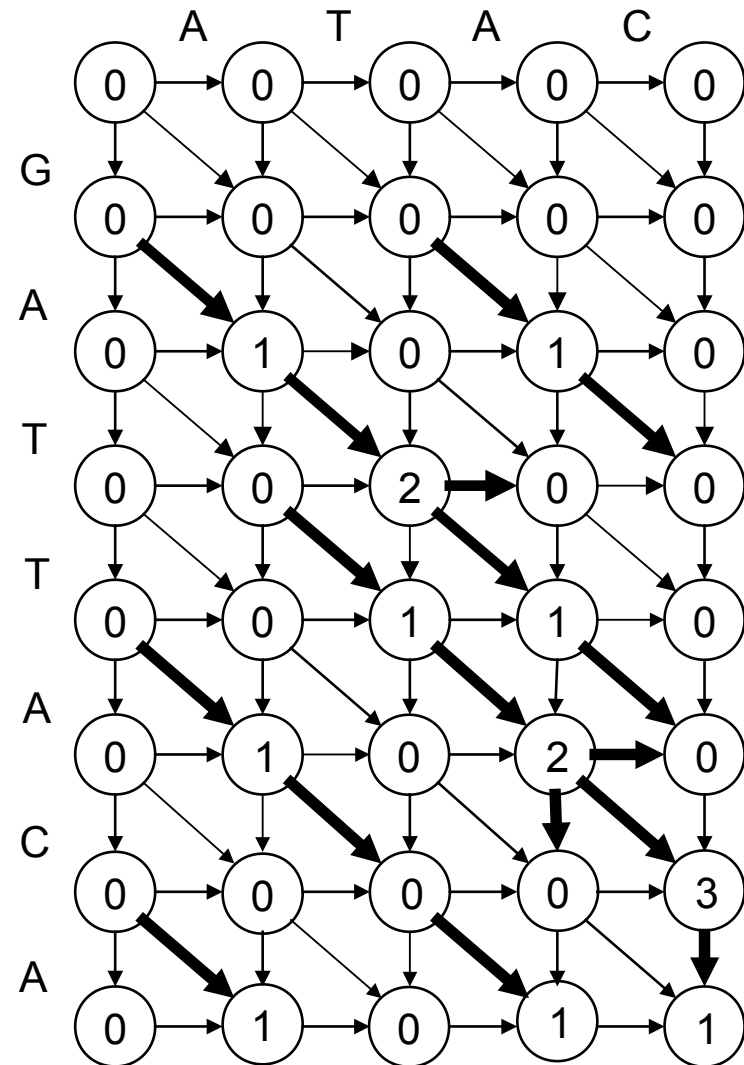
# Algorithme – remplissage de la matrice – alignement local

Alignement local: un alignement peut commencer n'importe où dans la matrice

→

Même initialisation que semi-global, et en plus tout score négatif est remplacé par zéro (sans prédécesseur: ce sera la début potentiel d'un alignement local optimal)

$$S(i,j) = \max \{ \begin{array}{l} S(i-1,j-1) + \text{subst}(s1[i-1],s2[j-1]) , \\ S(i-1,j) + \text{indel} , \\ S(i,j-1) + \text{indel} , \\ 0 \end{array} \}$$



# Algorithme – remplissage de la matrice – alignement local

Alignement local: un alignement peut commencer n'importe où dans la matrice

→

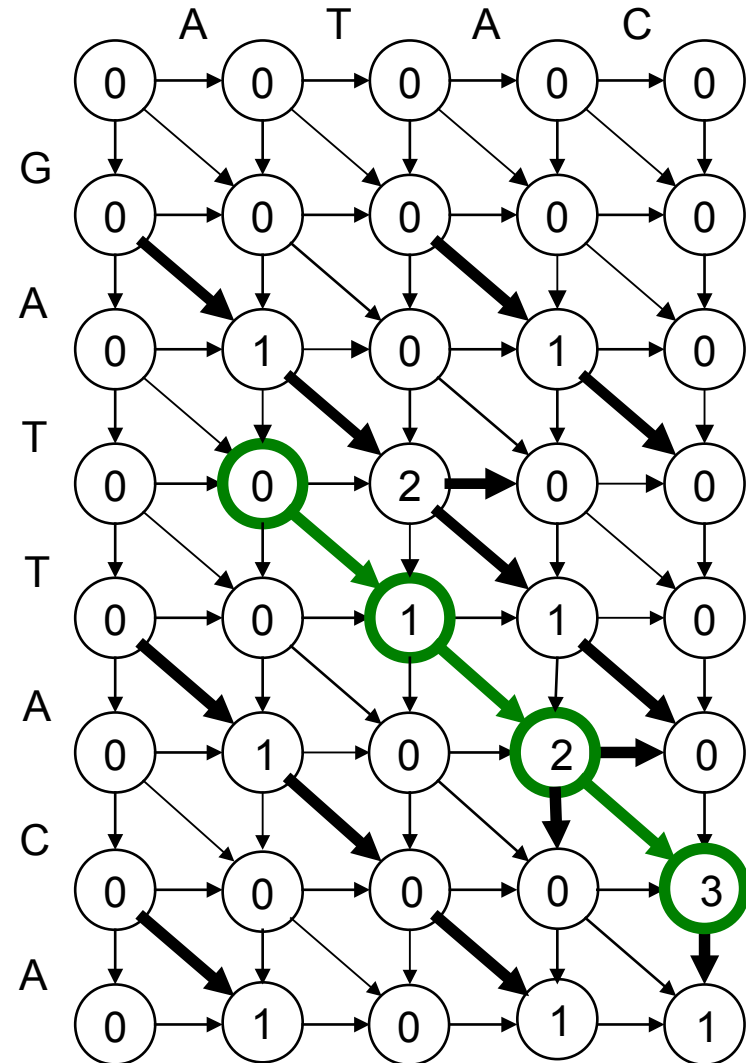
Même initialisation que semi-global, et en plus tout score négatif est remplacé par zéro (sans prédécesseur: ce sera la début potentiel d'un alignement local optimal)

Un alignement peut terminer n'importe où

→

on trouve les cases de score maximal dans **toute** la matrice: ce sont les fins des alignements optimaux

On remonte jusqu'à une case de score nul: début de l'alignement optimal considéré



# Algorithme – remplissage de la matrice – alignement local

Alignement local: un alignement peut commencer n'importe où dans la matrice

→

Même initialisation que semi-global, et en plus tout score négatif est remplacé par zéro (sans prédécesseur: ce sera la début potentiel d'un alignement local optimal)

Un alignement peut terminer n'importe où

→

on trouve les cases de score maximal dans **toute** la matrice: ce sont les fins des alignements optimaux

On remonte jusqu'à une case de score nul: début de l'alignement optimal considéré

Alignement local optimal (score 3) de l'exemple:

```
(gat) TAC (a)
      |||
      TAC
(a) TAC
```

