Bio-informatique : comparaison de séquences et calcul d'arbres phylogénétiques

Jean-Stéphane Varré

Université Lille 1 - CRIStAL - Inria Lille-Nord Europe

JEIA - 24 février 2016

Qu'est-ce que la bio-informatique ?

L'ordinateur est une extension de la paillasse pour mener des expériences en biologie moléculaire.
C'est de la biologie in silico.

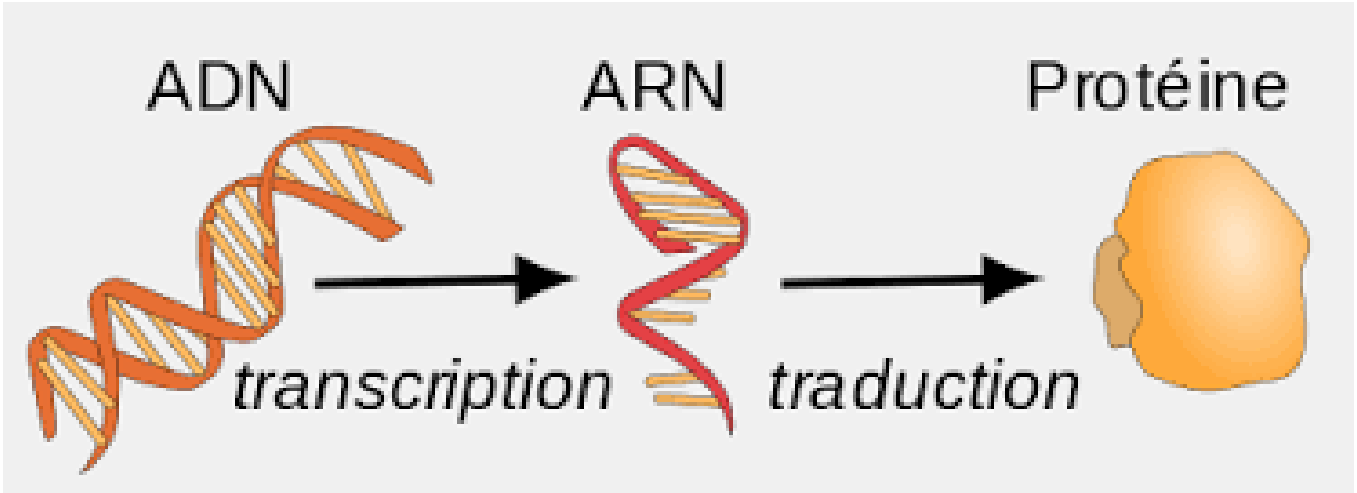
- acquisition et organisation des données biologiques
- conception de logiciels pour l'analyse, la comparaison et la modélisation des données
- analyse des résultats produits par les logiciels

Qu'est-ce que la bio-informatique ?

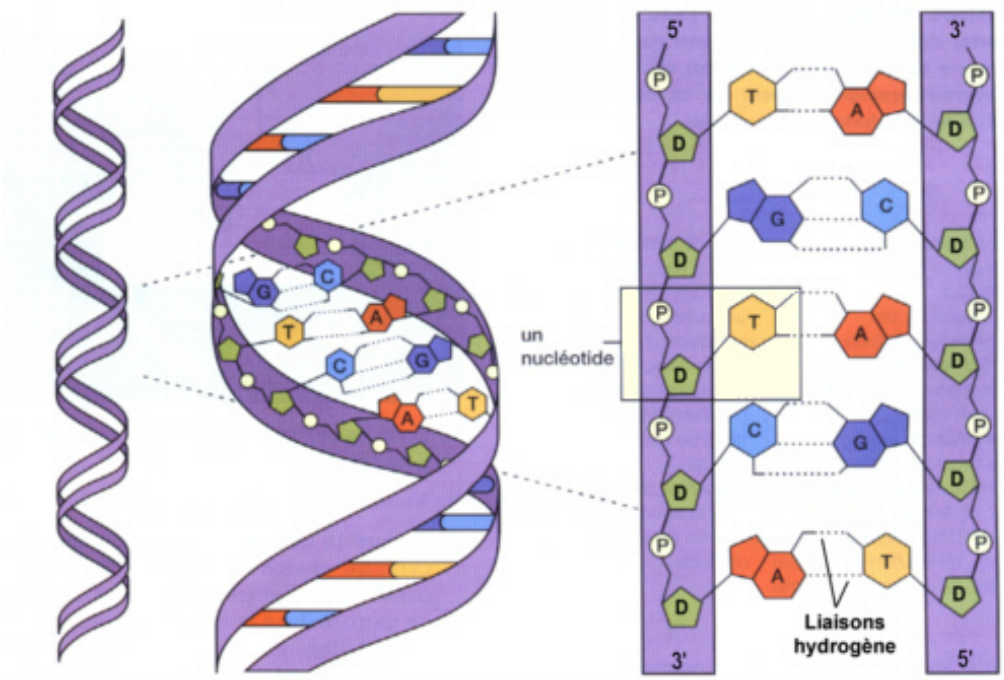
en anglais, distinction entre :

- « Bioinformatics » : applique des algorithmes, modèles statistiques dans l'objectif d'interpréter, classer et comprendre des données biologiques,
- « Computational Biology » : développer des modèles mathématiques et outils associés pour résoudre des problèmes biologiques.

Dogme central



L'ADN

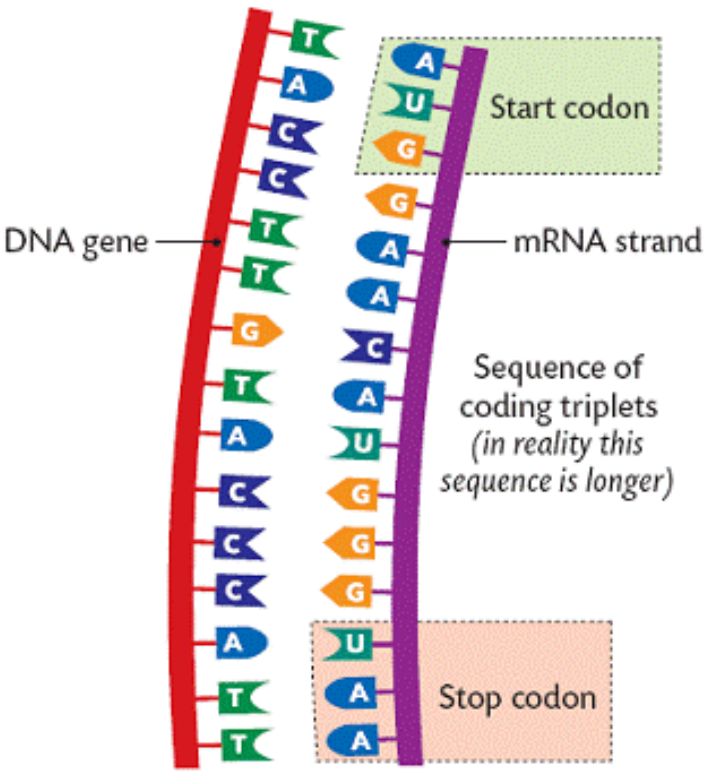


- c'est l'enchaînement de nucléotides le long d'une macromolécule d'ADN
- une séquence d'ADN est représentée par un texte écrit sur les 4 lettres A, T, C et G
- le brin complémentaire est la correspondance exacte de l'autre brin : A $\leftrightarrow$ T et C $\leftrightarrow$ G
- ayant un sens de lecture (5' vers 3')

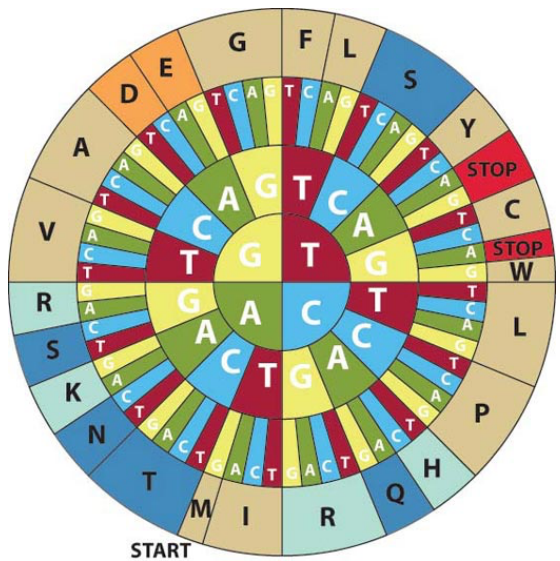
La séquence du gène ABO

```
>gi|58331215|ref|NM_020469.2| Homo sapiens ABO blood group (transferase A, alph
GGAGGCCGAGACCAGACGCGGAGCCATGGCCGAGGTGTTGCGGACGCTGGCCGGAAAACCAAAATGCCAC
GCACTTCGACCTATGATCCTTTTCCTAATAATGCTTGTCTTGGTCTTGTTTGGTTACGGGGTCCTAAGCC
CCAGAAGTCTAATGCCAGGAAGCCTGGAACGGGGGTTCTGCATGGCTGTTAGGGAACCTGACCATCTGCA
GCGCGTCTCGTTGCCAAGGATGGTCTACCCCCAGCCAAAGGTGCTGACACCGTGTAGGAAGGATGTCCTC
GTGGTGACCCCTTGGCTGGCTCCCATTGTCTGGGAGGGCACATTCAACATCGACATCCTCAACGAGCAGT
TCAGGCTCCAGAACACCACCATTGGGTTAACTGTGTTTGCCATCAAGAAATACGTGGCTTTCCTGAAGCT
GTTCCTGGAGACGGCGGAGAAGCACTTCATGGTGGGCCACCGTGTCCACTACTATGTCTTCACCGACCAG
CCGGCCGCGGTGCCCCGCGTGACGCTGGGGACCGGTCGGCAGCTGTCAGTGCTGGAGGTGCGCGCCTACA
AGCGCTGGCAGGACGTGTCCATGCGCCGCATGGAGATGATCAGTGACTTCTGCGAGCGGCGCTTCCTCAG
CGAGGTGGATTACCTGGTGTGCGTGGACGTGGACATGGAGTTCCGCGACCACGTGGGCGTGGAGATCCTG
ACTCCGCTGTTCGGCACCCTGCACCCCGGCTTCTACGGAAGCAGCCGGGAGGCCTTCACCTACGAGCGCC
GGCCCCAGTCCCAGGCCTACATCCCCAAGGACGAGGGCGATTTCTACTACCTGGGGGGGTTCTTCGGGGG
GTCGGTGCAAGAGGTGCAGCGGCTCACCAGGGCCTGCCACCAGGCCATGATGGTCGACCAGGCCAACGGC
ATCGAGGCCGTGTGGCACGACGAGAGCCACCTGAACAAGTACCTGCTGCGCCACAAACCCACCAAGGTGC
TCTCCCCCGAGTACTTGTGGGACCAGCAGCTGCTGGGCTGGCCCGCCGTCCTGAGGAAGCTGAGGTTCAC
TGCGGTGCCCAAGAACCACCAGGCGGTCCGGAACCCGTGAGCGGCTGCCAGGGGCTCTGGGAGGGCTGCC
GGCAGCCCCGTCCCCCTCCCGCCCTTGGTTTTAGCAGAACGGGTAAACTCTGTTTCCTTTGTCCGTCCTG
TTGTGAGTAACTGAAGCCTAGGCCCCGTCCCCACCTCAAATCACACACACCCCCTCCCCACCACAGAGAC
ACCATTACATACACAGACACACACAGAAAGACACACACAGACACAAAATCACACACACACCCTCCCCGCC
ACAGAGACACCATTACATACACAGACACACACAGAAAGACACAGACACAAAATCACACACACACCCTCCC
CGCCACAGAGACACACCATTACATACACAGACACGCAATCGCAGATACGCCCTTCCGGCCACAGAAACAC
ACCATTACACACACATACACAGAAAGACACACACAGACACACAATCACACGCAGCCCCTCCCCGCCACAG
AGACACACCATTACATACACAGACACACACAGAAAGACAC
```

Du gène à la protéine



- c'est la suite de nucléotides au sein du gène qui détermine la suite d'acides aminés qui compose une protéine
- décryptage grâce au code génétique



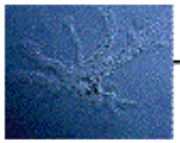
Exemple

133,275,214
GGAGGCCGAGACCAGACGCGGAGCCATGGCCGAGGTGTTGCGGACGCTGGCCG
gtgagtgcaggcctcggccccgggt.....tgatttttctactcctgttttccag
GAAAACCAAAATGCCACGCACTTCGACCTATGATCCTTTTCCTAATAATGCTTGTCTTGG
TCTTGTTTGG
gtaagacacatttgaccatcgaggc.....gtcttggcacacttcctttctgcag
TTACGGGGTCCTAAGCCCCAGAAGTCTAATGCCAGGAAGCCTGGAACGGGGGTTCTG
gtgagtgcagggaagagcaggtgga.....catctcctgtgttctcattctgcag
CATGGCTGTTAGGGAACCTGACCATCTGCAGCGCGTCTCGTTGCCAAG
gtataatgtcagtgcctcccttcag.....acgtggcggcgctttgcgcttgcag
GATGGTCTACCCCCAGCCAAAGGTGCTGACACCGTG
gtgagtaaagttactgacactgaaa.....accgcacgcctctctccatgtgcag
TAGGAAGGATGTCCTCGTG
gt
ACCCCTTGGCTGGCTCCCATTGTCTGGGAGGGCACATTCAACATCGACATCCTCAACGAG
CAGTTCAGGCTCCAGAACACCACCATTGGGTTAACTGTGTTTGCCATCAAGAA
gtaagtcagtgaggtggccgagggt.....cagcccccgtccgcctgccttgcag
ATACGTGGCTTTCCTGAAGCTGTTCCTGGAGACGGCGGAGAAGCACTTCATGGTGGGCCA
CCGTGTCCACTACTATGTCTTCACCGACCAGCCGGCCGCGGTGCCCCGCGTGACGCTGGG
GACCGGTCGGCAGCTGTCAGTGCTGGAGGTGCGCGCCTACAAGCGCTGGCAGGACGTGTC
CATGCGCCGCATGGAGATGATCAGTGACTTCTGCGAGCGGCGCTTCCTCAGCGAGGTGGA
TTACCTGGTGTGCGTGGACGTGGACATGGAGTTCCGCGACCACGTGGGCGTGGAGATCCT
GACTCCGCTGTTCGGCACCCTGCACCCCGGCTTCTACGGAAGCAGCCGGGAGGCCTTCAC
CTACGAGCGCCGGCCCCAGTCCCAGGCCTACATCCCCAAGGACGAGGGCGATTTCTACTA
CCTGGGGGGGTTCTTCGGGGGGTCGGTGCAAGAGGTGCAGCGGCTCACCAGGGCCTGCCA
CCAGGCCATGATGGTCGACCAGGCCAACGGCATCGAGGCCGTGTGGCACGACGAGAGCCA
CCTGAACAAGTACCTGCTGCGCCACAAACCCACCAAGGTGCTCTCCCCCGAGTACTTGTG
GGACCAGCAGCTGCTGGGCTGGCCCGCCGTCCTGAGGAAGCTGAGGTTCACTGCGGTGCC
CAAGAACCACCAGGCGGTCCGGAACCCGTGAGCGGCTGCCAGGGGCTCTGGGAGGGCTGC
CGGCAGCCCCGTCCCCCTCCCGCCCTTGGTTTTAGCAGAACGGGTAAACTCTGTTTCCTT
TGTCCGTCCTGTTGTGAGTAACTGAAGCCTAGGCCCCGTCCCCACCTCAAATCACACACA
CCCCCTCCCCACCACAGAGACACCATTACATACACAGACACACACAGAAAGACACACACA
GACACAAAATCACACACACACCCTCCCCGCCACAGAGACACCATTACATACACAGACACA
CACAGAAAGACACAGACACAAAATCACACACACACCCTCCCCGCCACAGAGACACACCAT
TACATACACAGACACGCAATCGCAGATACGCCCTTCCGGCCACAGAAACACACCATTACA
CACACATACACAGAAAGACACACACAGACACACAATCACACGCAGCCCCTCCCCGCCACA
GAGACACACCATTACATACACAGACACACACAGAAAGACAC
133,255,176

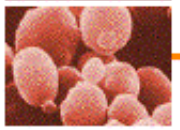
53 + 70 + 57 + 48
+ 36 + 19 + 113 + 1181 = 1580

354 acides aminés
maevlrtlag kpkchalrpm ilflimlvlv
lfgygvlspr slmpgslerg fcmavrepdh
lqrvslprmv ypqpkvltpc rkdvlvvtpw
lapivwegtf nidilneqfr lqnttigltv
faikkyvafl klfletaekh fmvghrvhyy
vftdqpaavp rvtlgtgrql svlevraykr
wqdvsmrrme misdfcerrf lsevdylvcv
dvdmefrdhv gveiltplfg tlhpgfygss
reaftyerrp qsqayipkde gdfyylggff
ggsvqevqrl trachqammv dqangieavw
hdeshlnkyl lrhkptkvls peylwdqqll
gwpavlrklr ftavpknhqa vrnp

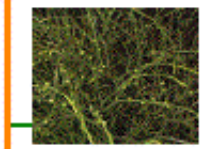
Taille des objets manipulés





	Taille du génome (nucléotides)	Nbre de gènes (*protein-coding*)
Amoeba dubia	~ 670 000 000 000	?
Psilotum nudum	~ 250 000 000 000	?
Fritillaria assyriaca	~ 100 000 000 000	?
Necturus lewisi	~100 000 000 000	?
Homo sapiens	2 900 000 000	23 000
Vitis vinifera	487 000 000	30 400
Drosophila melanogaster	160 000 000	14 000
Arabidopsis thaliana	115 000 000	28 000
Caenorhabditis elegans	98 000 000	19 400
Saccharomyces cerevisiae	12 500 000	5 800
Escherichia coli	4 600 000	4 300





B. Dujon (2008)

Activités

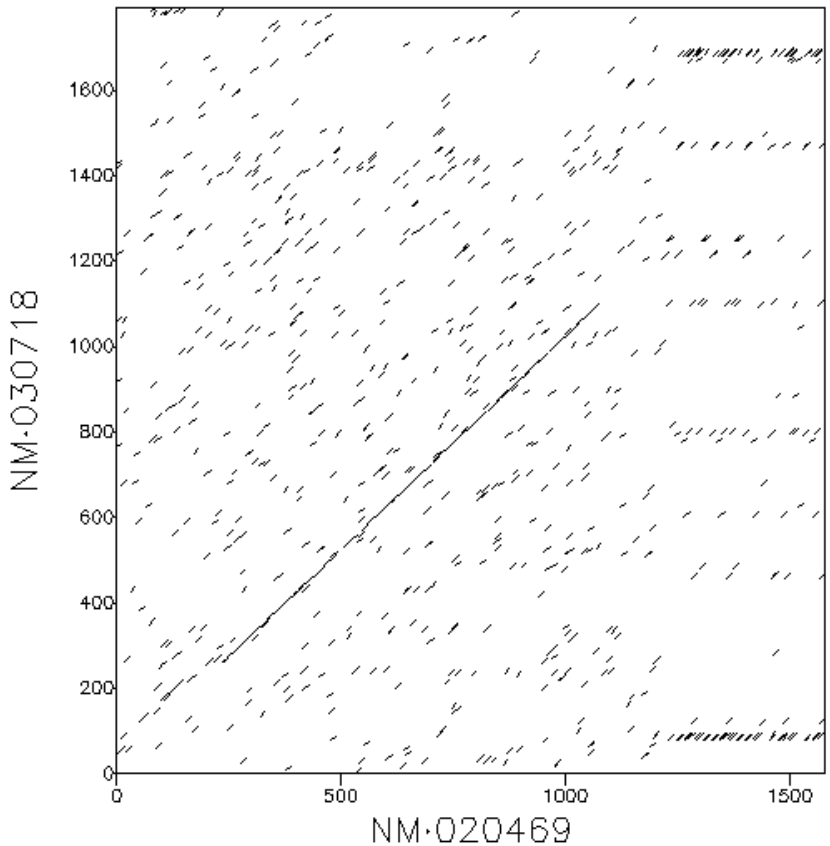
A Etant donné une séquence, calculer son taux de GC.

B Etant donné une séquence, calculer son complémentaire inversé

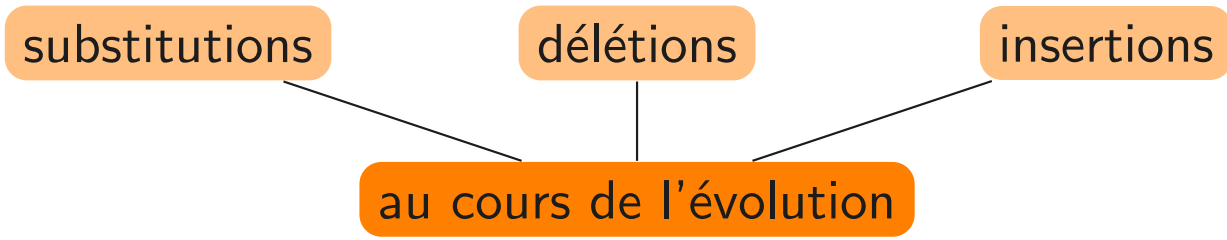
C Etant donné une séquence que l'on sait codant pour un gène, calculer la séquence protéique associée

D Etant donné une séquence, extraire toutes les ORF (Open Reading Frame) : toutes les sous-séquences débutant par un codon START, se terminant par un codon STOP, de taille supérieure à 3 codons, sur les deux brins (au premier STOP rencontré, la traduction s'arrête)

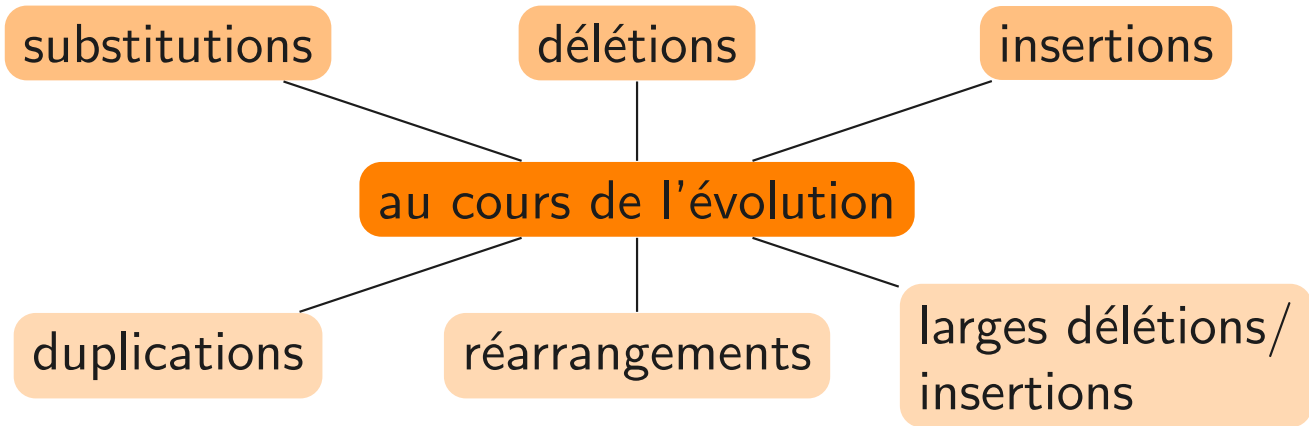
Comparaison de séquences



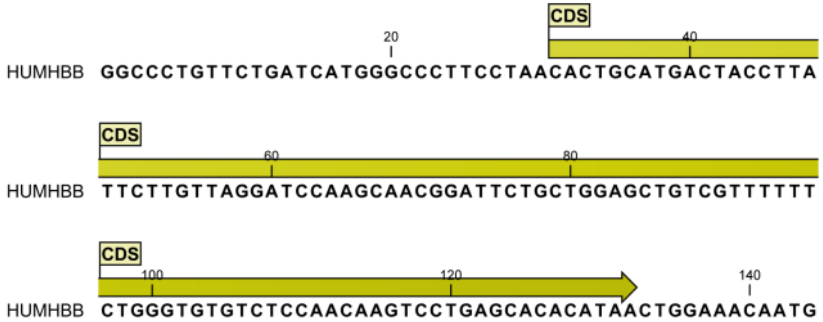
Pourquoi existe-t-il des séquences similaires ?

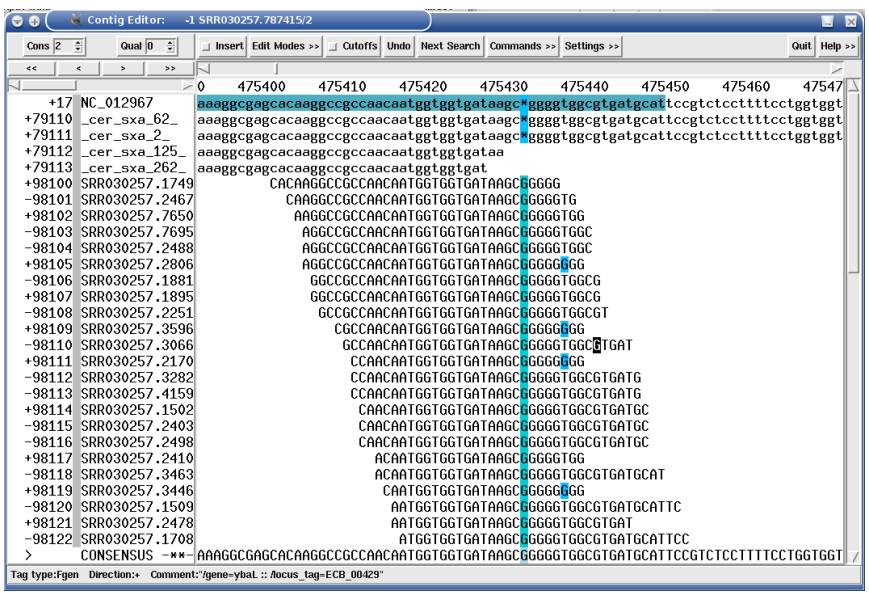


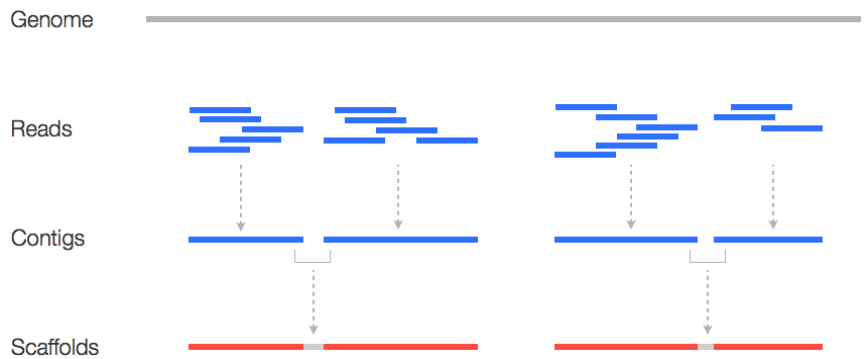
Pourquoi existe-t-il des séquences similaires ?

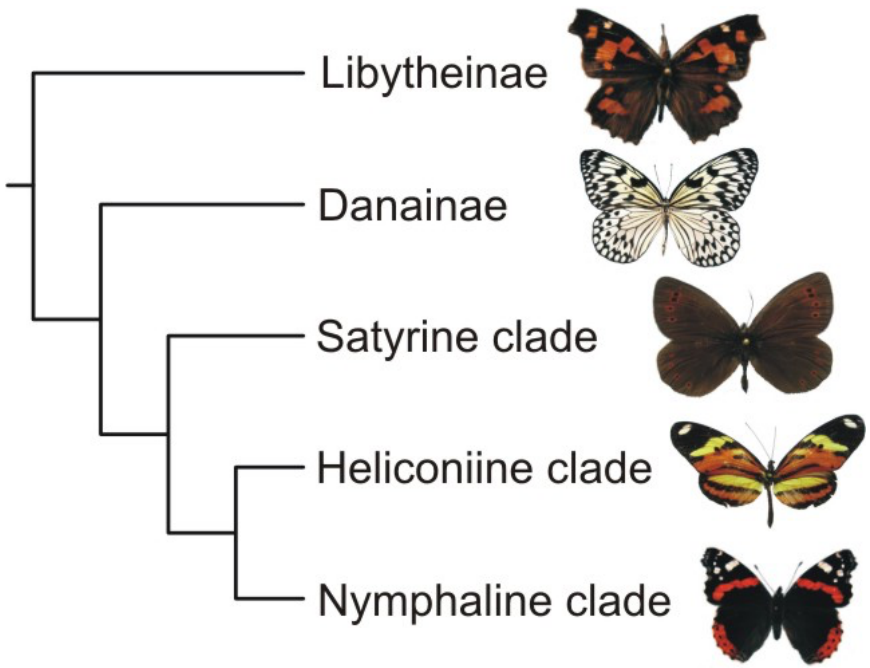


De l'utilité de savoir comparer









Exemples

```
G A C T C C G
|       | | |    3 substitutions
G C T A C C G
```

Exemples

```
G A C T C C G
|       | | |    3 substitutions
G C T A C C G

G A C T - C C G
|   | |   | | |  2 indels
G - C T A C C G
```

Exemples

```
G A C T C C G
|       | | |    3 substitutions
G C T A C C G

G A C T - C C G
|   | |   | | |  2 indels
G - C T A C C G
```

- calculer la similarité = aligner
- mesurer la similarité = obtenir l'alignement optimal
- nécessite de calculer un score

Exemples

```
G A C T C C G
|       | | |    3 substitutions
G C T A C C G

G A C T - C C G
|   | |   | | |  2 indels
G - C T A C C G
```

- calculer la similarité = aligner
- mesurer la similarité = obtenir l'alignement optimal
- nécessite de calculer un score

A rapprocher de . . .

distance de Hamming entre deux textes : nombre minimum de substitutions
distance de Levenshtein entre deux textes : nombre minimum de substitutions et d'indels

Comparaison du gène ABO homme/souris

```
NM_020469          0 --------------------------------------------------      0

NM_030718          1 gtgttcagagctgtgtattatctcccctgggggtgagttctcctgtgtgc     50

NM_020469          1 ------------------------ggaggccgagaccagacgcggagcca     26
                                            |...||.| .| |||||.|.||.|||
NM_030718         51 ctgagacctggcctgtgcctaacagcttgctg-ca-cagacactgaacca     98

NM_020469         27 tggcc-gaggtgt-tg-c-ggacgct-ggccggaaaaccaaaatgccac-     70
                     |  || | ||| | || | .||..|| .| .||||.|||.||||||.||
NM_030718         99 t--cctg-ggt-tctgacatgaatctcag-aggaagaccgaaatgcaact    143

NM_020469         71 gcacttcgacctatg--atccttttcc-taataatgcttgtcttggtctt    117
                     .| |||| |||| ||  |||| ||.|| |...|.||.|||||||.|||||
NM_030718        144 tc-cttc-acct-tggaatcc-ttcccttcgcagtgtttgtcttagtctt    189

NM_020469        118 gtttggttacggggtcctaagccccagaagtctaatgccaggaagcctgg    167
                     .|||||.|||..|.||||||||        | |     || |||||| .|
NM_030718        190 ctttggctacctgttcctaagc--------t-t-----ca-gaagcc-ag    223

NM_020469        168 aacgggggttctgcatggctgttagggaacctgaccatctgcagcgcgtc    217
                     |||     |  ||   || |         |   |||     ||| |.| |
NM_030718        224 aac-----t--tg---gg-t---------c---acc-----cag-gag-c    243

NM_020469        218 tcgttgccaagg-atggtctacccccagccaaaggtgctgacaccgtgta    266
                     | | ||.|.||| || |.|||.|..|||||||.||||||.|.|||...||
NM_030718        244 t-g-tgactaggaat-gcctatctgcagccaagggtgctaaaacccacta    290

NM_020469        267 ggaaggatgtcctcgtggtgacccccttggctggctcccattgtctgggag    316
                     ||||.|||||.||.||..||||.||||||||||.|||||..||||||||
NM_030718        291 ggaaagatgttcttgtcttgactccttggctggcgcccatcatctgggag    340

NM_020469        317 ggcacattcaacatcgacatcctcaacgagcagttcaggctccagaacac    366
                     ||.||.|||||||||||||..|.||.||||||||||||.|.|.|||.||
NM_030718        341 gggaccttcaacatcgacatattgaatgagcagttcaggattcggaatac    390

NM_020469        367 caccattgggttaactgtgtttgccatcaagaaa-tacgtggct-ttcct    414
                     .||.||||||..|.|||||.|||||.||||| ||| ||.|||| | |||||
NM_030718        391 tacgattggactgactgtatttgctatcaa-aaagtatgtgg-tgttcct    438
```

Alignement

- données :
  - une paire de séquences (ADN / protéine)
  - une schéma de score : comment compter ce qui se ressemble ?

Alignement

- données :
  - une paire de séquences (ADN / protéine)
  - une schéma de score : comment compter ce qui se ressemble ?
- but :
  - déterminer le degré de similarité (meilleur score)
  - montrer la similarité (meilleur alignement)

Alignement

- données :
  - une paire de séquences (ADN / protéine)
  - une schéma de score : comment compter ce qui se ressemble ?
- but :
  - déterminer le degré de similarité (meilleur score)
  - montrer la similarité (meilleur alignement)
- décrit la ressemblance grâce à 3 opérations (mutations ponctuelles)
  - insertion
  - délétion
  - identité/substitution

Alignement

- données :
  - une paire de séquences (ADN / protéine)
  - une schéma de score : comment compter ce qui se ressemble ?
- but :
  - déterminer le degré de similarité (meilleur score)
  - montrer la similarité (meilleur alignement)
- décrit la ressemblance grâce à 3 opérations (mutations ponctuelles)
  - insertion
  - délétion
  - identité/substitution
- mesure la ressemblance en donnant un poids à chaque opération
  - poids positif ("récompense") aux *bonnes parties* de l'alignement
    e.g appariement de deux lettres identiques ou proches
  - poids négatif ou nul ("pénalité") aux *mauvaises parties* de l'alignement
    e.g appariement de deux lettres non relatés, non-appariement

Composantes d'un schéma de scores

- score (ou poids) pour une identité/substitution : matrice s de similarité
  - exemple : $s(a, b) =$ score d'alignement des nucléotides a et b

$$\begin{pmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{pmatrix}$$

Composantes d'un schéma de scores

- score (ou poids) pour une identité/substitution : matrice s de similarité
  - exemple : $s(a, b) =$ score d'alignement des nucléotides a et b

$$\begin{pmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{pmatrix}$$

- score (ou poids) d'un *indel* (insertion/délétion)
  - exemple : score unitaire $= -2$ par *indel*

Composantes d'un schéma de scores

- score (ou poids) pour une identité/substitution : matrice s de similarité
  - exemple : $s(a, b) =$ score d'alignement des nucléotides a et b

$$\begin{pmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{pmatrix}$$

- score (ou poids) d'un *indel* (insertion/délétion)
  - exemple : score unitaire $= -2$ par *indel*
- **score de l'alignement = somme des scores des événements élémentaires**
- exemple :

```
A  A  C  G  T  A  C  G  A  T  A
|  |  |  |  |  |        |     |
A  A  C  G  T  A  -  A  A  G  A
-------------------------------
1  1  1  1  1  1 -2 -1  1 -1  1 = 4
```

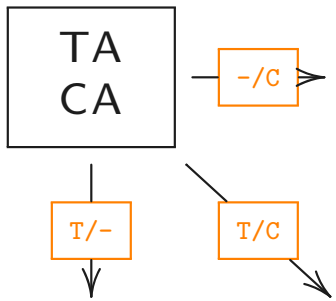
Comment calculer le meilleur alignement ?

- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`

TA
CA

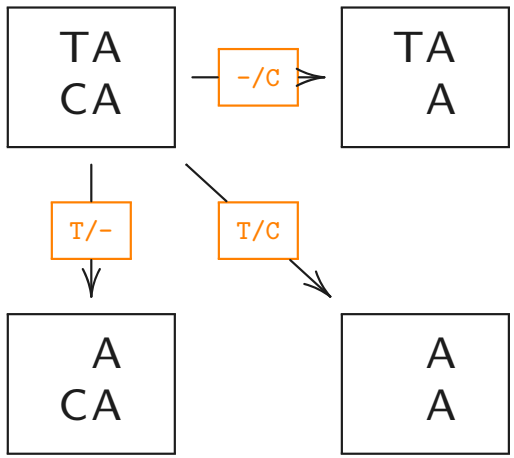
Comment calculer le meilleur alignement ?

- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`



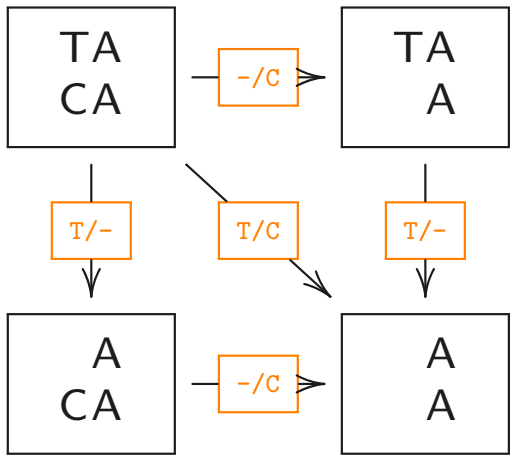
Comment calculer le meilleur alignement ?

- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`



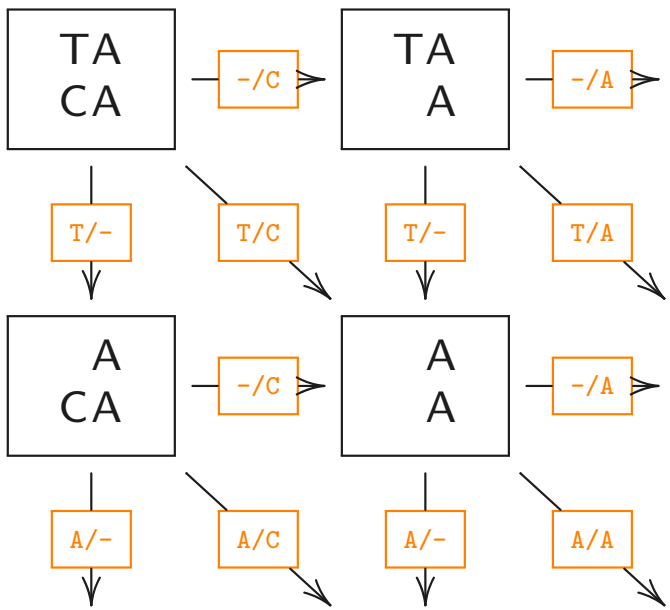
Comment calculer le meilleur alignement ?

- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`



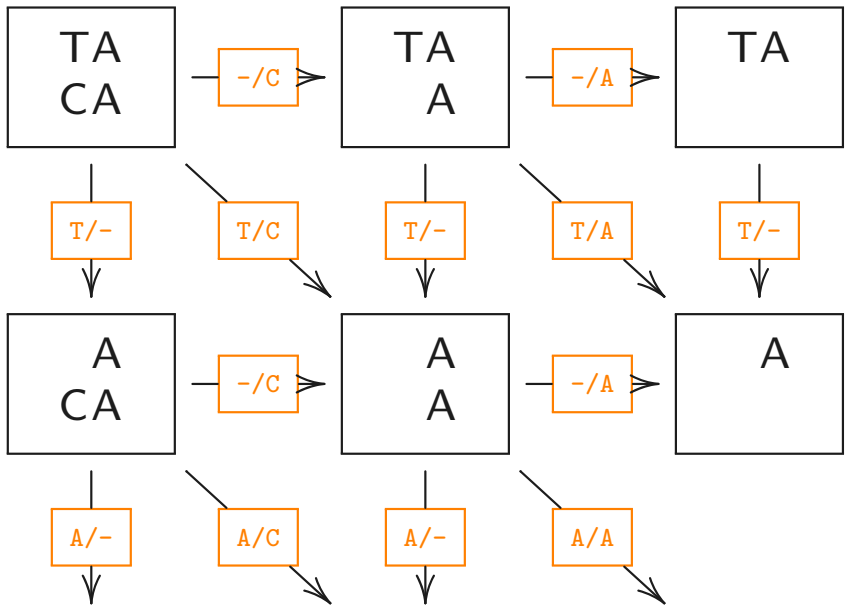
Comment calculer le meilleur alignement ?

- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`



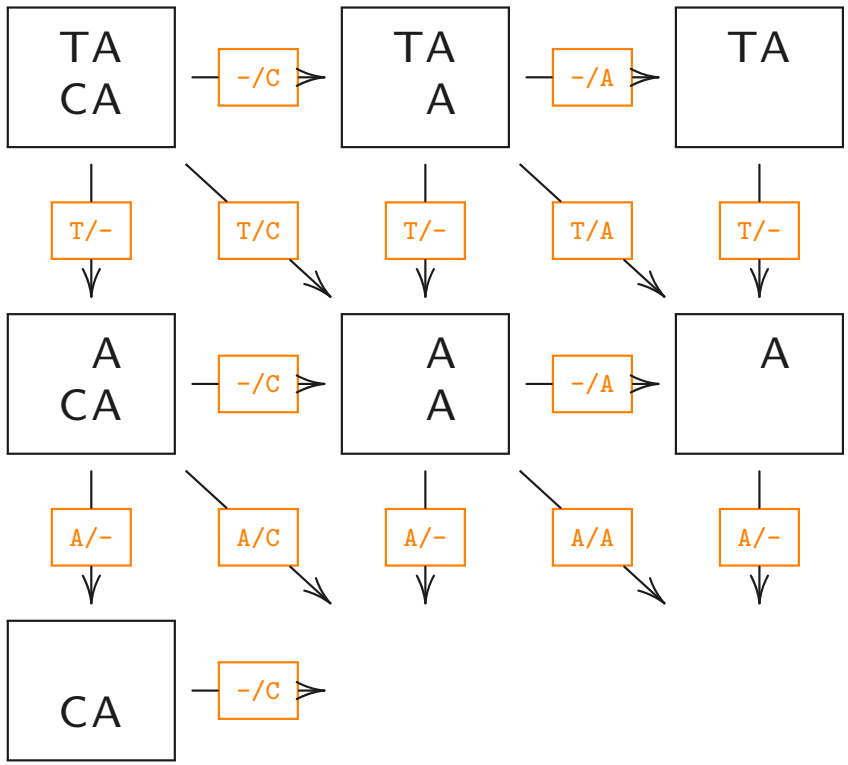
Comment calculer le meilleur alignement ?

- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`



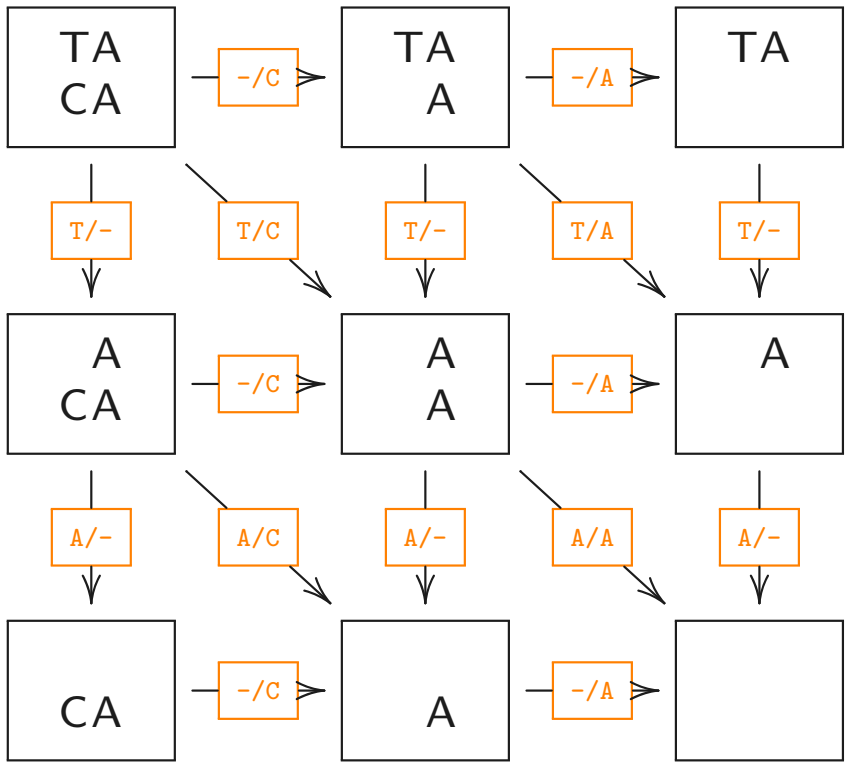
Comment calculer le meilleur alignement ?

- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`



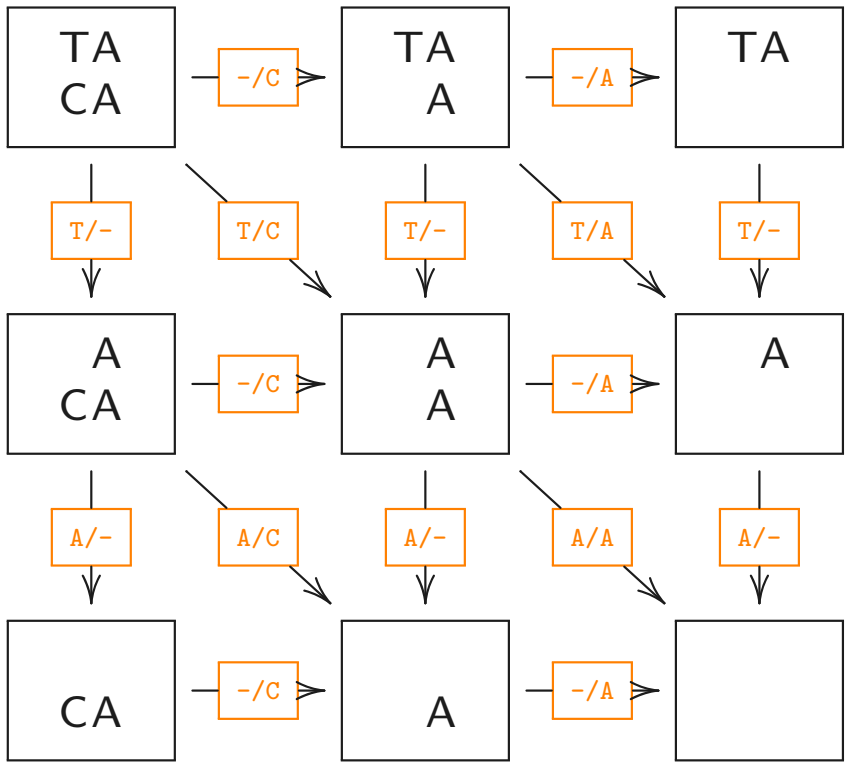
Comment calculer le meilleur alignement ?

- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`



Comment calculer le meilleur alignement ?

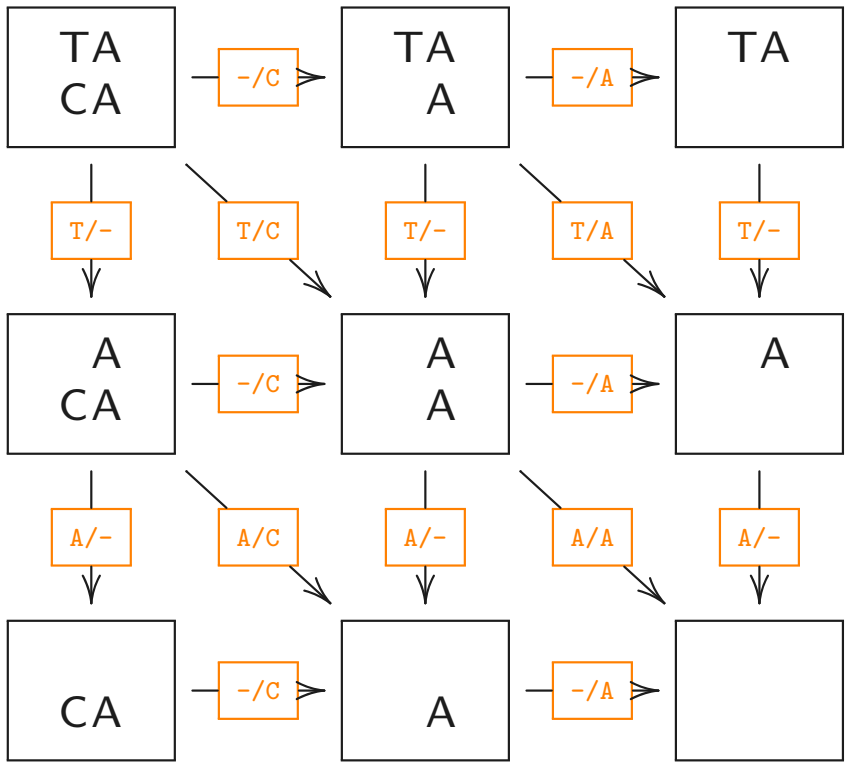
- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`



- une suite de couples de lettres en suivant les flèches donne un alignement

Comment calculer le meilleur alignement ?

- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`

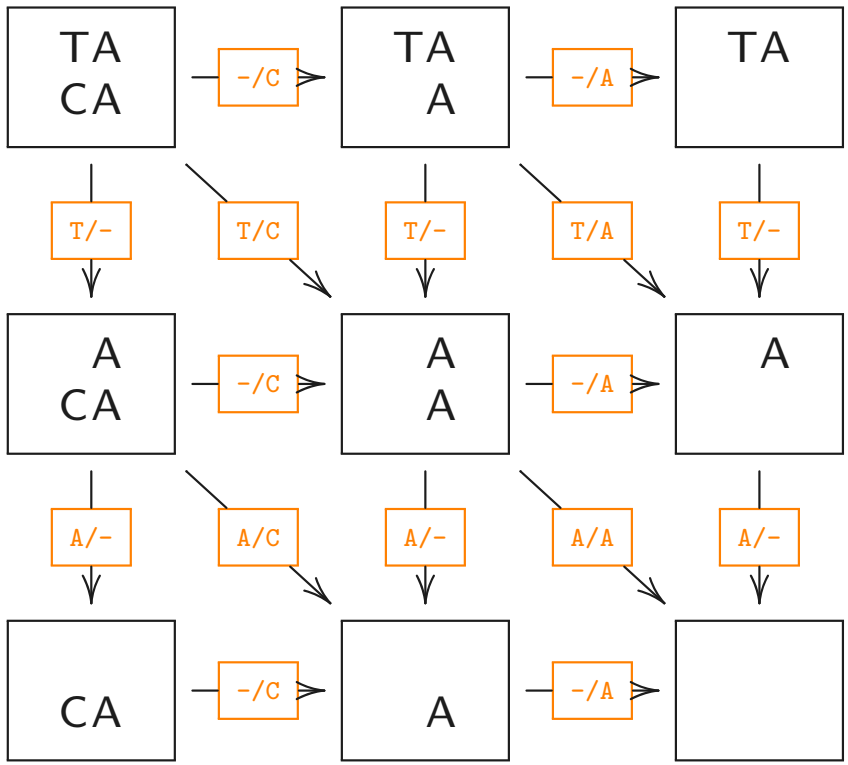


`TA`
`CA`

- une suite de couples de lettres en suivant les flèches donne un alignement

Comment calculer le meilleur alignement ?

- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`




```
TA
CA
```

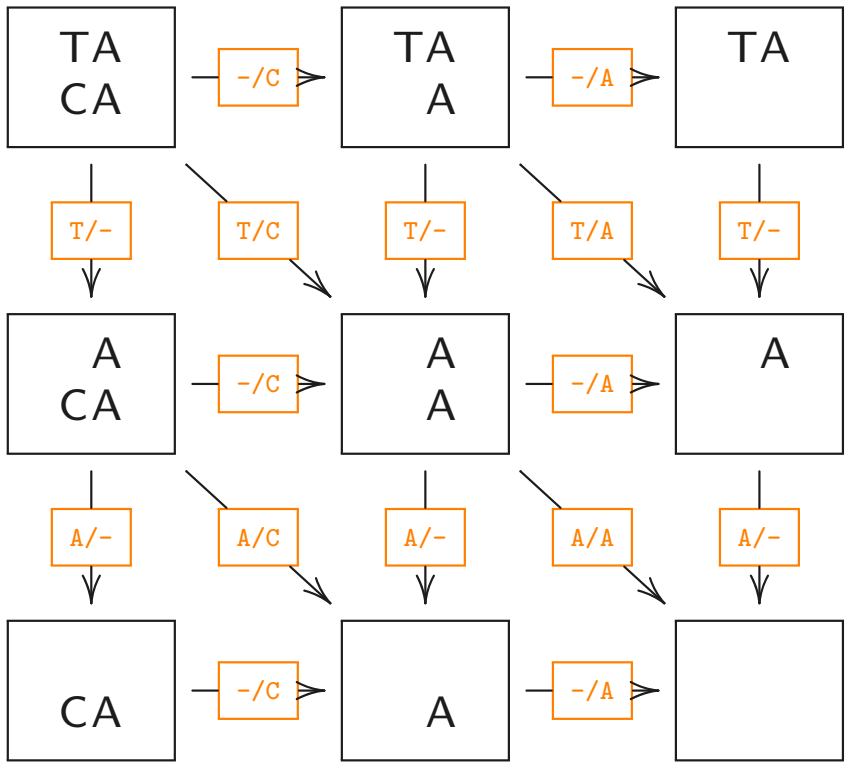
```
TA-     TA-     -TA
-CA     C-A     CA-
```

```
-TA     T-A     T-A
C-A     -CA     C-A
```

- une suite de couples de lettres en suivant les flèches donne un alignement

Comment calculer le meilleur alignement ?

- prenons 2 séquences de longueur n toutes les deux : alignement de longueur maximale $2n$
- exemple avec les séquences `TA` et `CA`




```
TA
CA
```

```
TA-    TA-    -TA
-CA    C-A    CA-
```

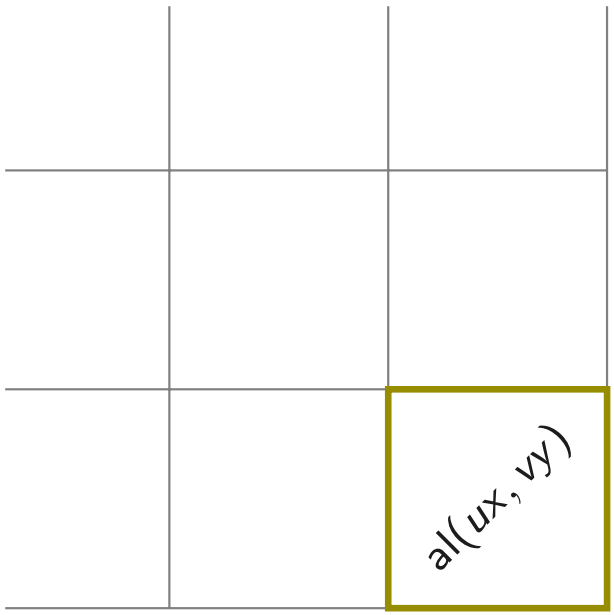
```
-TA    T-A    T-A
C-A    -CA    C-A
```

```
TA--   --TA
--CA   CA--
```

```
T-A-   -T-A   -TA-
-C-A   C-A-   C--A
```

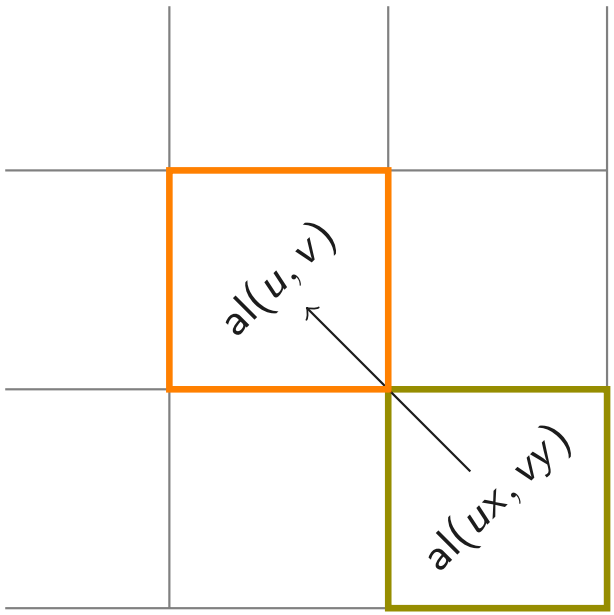
- une suite de couples de lettres en suivant les flèches donne un alignement

Une expression récursive du meilleur score



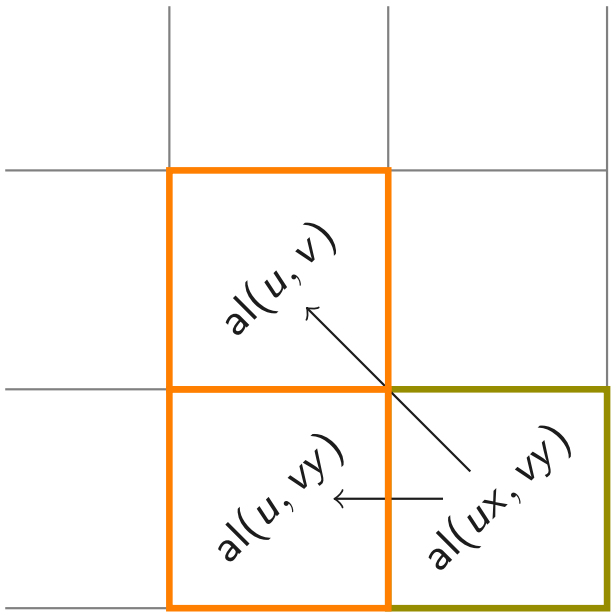
$$\mathrm{al}(ux, vy) = \max \left\{ \right.$$

Une expression récursive du meilleur score



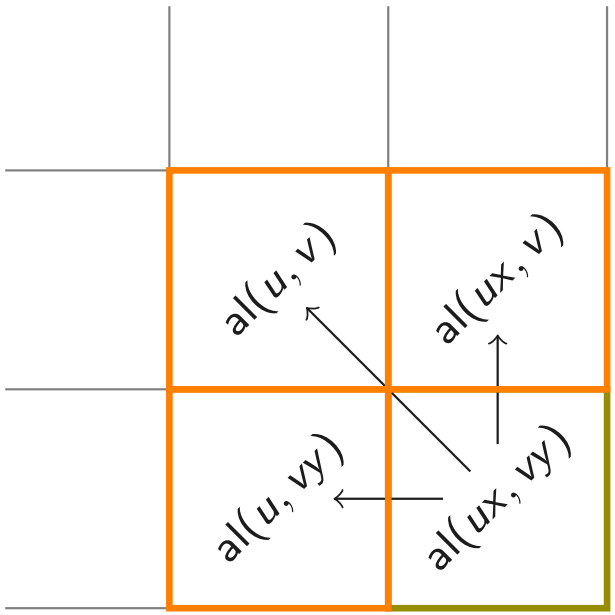
$$\mathrm{al}(ux, vy) = \max \begin{cases} \mathrm{al}(u, v) + \mathrm{score}(x, y) \\ \end{cases}$$

Une expression récursive du meilleur score



$$\mathrm{al}(ux, vy) = \max \begin{cases} \mathrm{al}(u, v) + \mathrm{score}(x, y) \\ \mathrm{al}(u, vy) + \mathrm{score}(x, -) \\ \end{cases}$$

Une expression récursive du meilleur score



$$\mathrm{al}(ux, vy) = \max \begin{cases} \mathrm{al}(u, v) + \mathrm{score}(x, y) \\ \mathrm{al}(u, vy) + \mathrm{score}(x, -) \\ \mathrm{al}(ux, v) + \mathrm{score}(-, y) \end{cases}$$

Programmation dynamique

Needleman-Wunsch

match = 1 mismatch = -1 gap = -1

		G	C	A	T	G	C	U
	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

source : Wikipedia

```
GCATG-CU
|  ||   |
G-ATTACA

GCA-TGCU
|  |  | |
G-ATTACA

GCA-TGCU
|  |  | |
G-ATTACA
```

Difficulté du problème

- nombre max d'alignements (séq. de lg n)

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \sim \frac{2^{2n}}{\sqrt{2\pi n}}$$

pour deux séquences de longueur 100 : 2.10^{57} *alignements*

- grâce à la représentation en tableau : complexité en temps et en espace $\mathcal{O}(n^2)$

(proportionnel au produit de la longueur des séquences)

pour deux séquences de longueur 100 : 10000 *opérations*

Un modèle plus fin pour les indels

- ne plus compter une succession d'indels comme plusieurs évènements mais comme un seul : un gap
- changement de la fonction de score :
  - coût associé à l'ouverture de gap
  - coût associé à l'extention de gap
- exemple (ouv = -2, ext = -1) :

```
A  T  C  G  G  C  A  T  G  C  C  G
|        |  |     |  |  |  |
A        G  G  A  A  T  G  C  -  G
1 -2 -1  1  1 -1  1  1  1  1 -2  1 = 2
```

Conséquence sur l'algorithme

La récurrence s'écrit alors :

$$\mathsf{al}(n, m) = \max \left\{ \right.$$

Conséquence sur l'algorithme

La récurrence s'écrit alors :

$$\mathrm{al}(n,m) = \max \begin{cases} \mathrm{al}(n-1,m-1) + \mathrm{score}(u[n], v[m]) \\ \\ \\ \end{cases}$$

Les conditions initiales (première ligne, première colonne) sont :

$$\mathrm{al}(0,i) = \mathrm{al}(i,0) = gap(i)$$

Conséquence sur l'algorithme

La récurrence s'écrit alors :

$$\mathrm{al}(n,m) = \max \begin{cases} \mathrm{al}(n-1,m-1) + \mathrm{score}(u[n], v[m]) \\ \max_i(\mathrm{al}(n-i,m) + \mathrm{gap}(i)) \\ \end{cases}$$

Les conditions initiales (première ligne, première colonne) sont :

$$\mathrm{al}(0,i) = \mathrm{al}(i,0) = gap(i)$$

Inconvénient : la complexité de l'algorithme augmente (devient en $O(n^3)$)
Mais on peut faire mieux ...

Conséquence sur l'algorithme

La récurrence s'écrit alors :

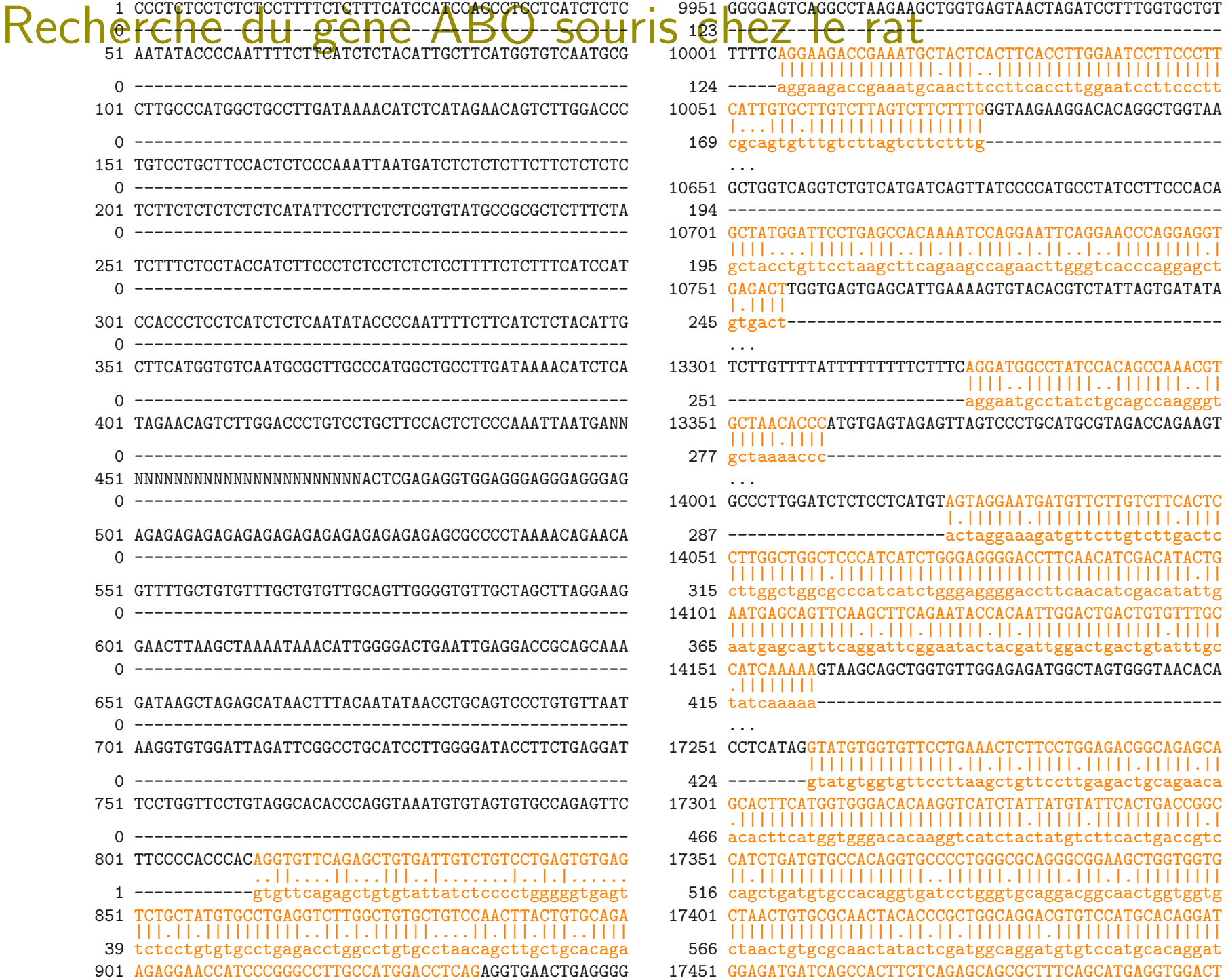
$$\mathrm{al}(n,m) = \max \begin{cases} \mathrm{al}(n-1,m-1) + \mathrm{score}(u[n], v[m]) \\ \max_i(\mathrm{al}(n-i,m) + \mathrm{gap}(i)) \\ \max_i(\mathrm{al}(n,m-i) + \mathrm{gap}(i)) \end{cases}$$

Les conditions initiales (première ligne, première colonne) sont :

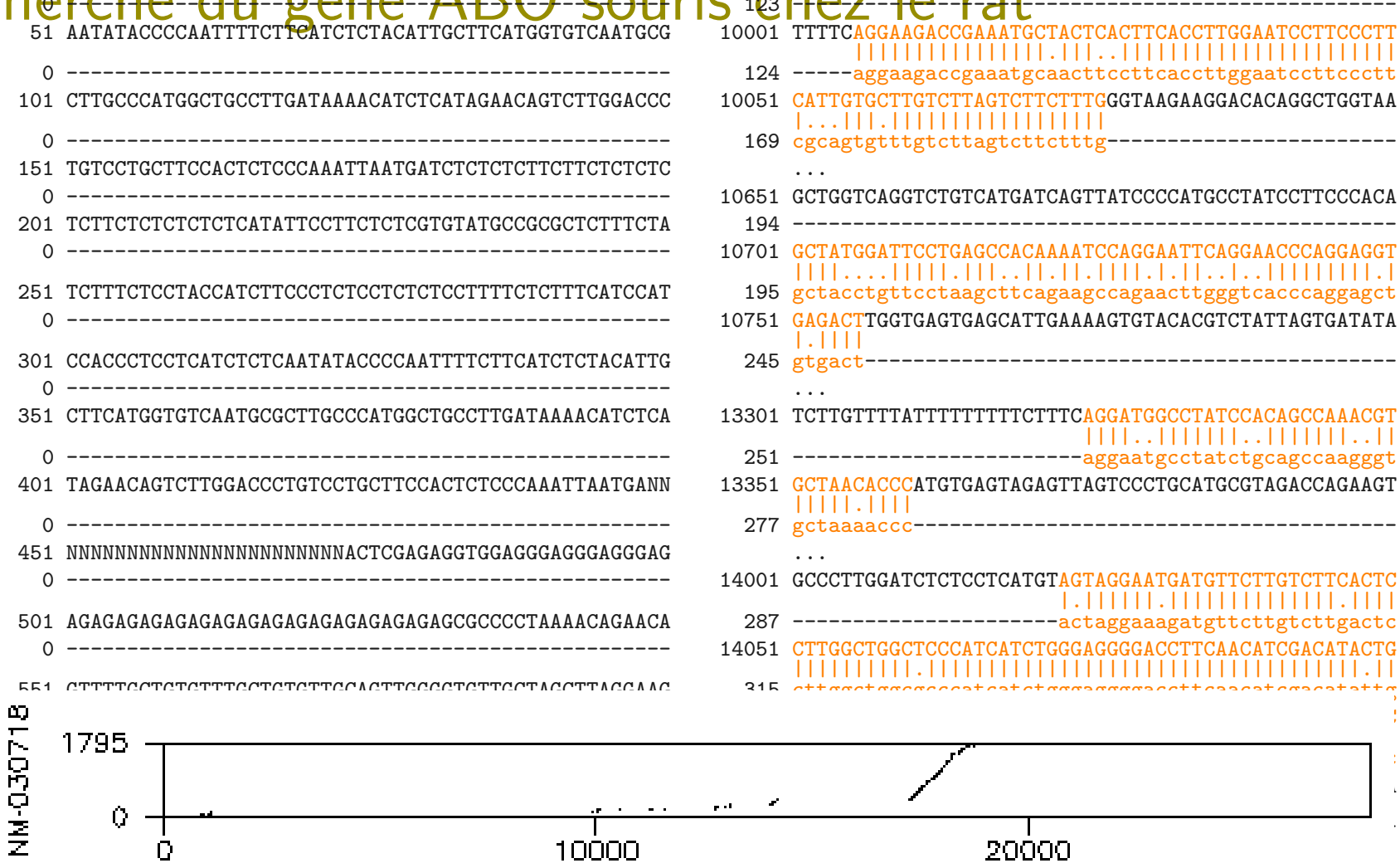
$$\mathrm{al}(0,i) = \mathrm{al}(i,0) = gap(i)$$

Inconvénient : la complexité de l'algorithme augmente (devient en $O(n^3)$)
Mais on peut faire mieux ...

Recherche du gene ABO souris chez le rat



Recherche du gene ABO souris chez le rat



Activités

A Calculer le score d'un alignement donné

B Modifier un alignement à la main pour obtenir un meilleur score

C Produire deux jeux de score « pertinents » tels que ce n'est pas le même meilleur alignement produit

D $\star$ Programmer le calcul du meilleur alignement

E $\star\star$ Produire un meilleur alignement

F $\star\star\star$ Produire les meilleurs alignements

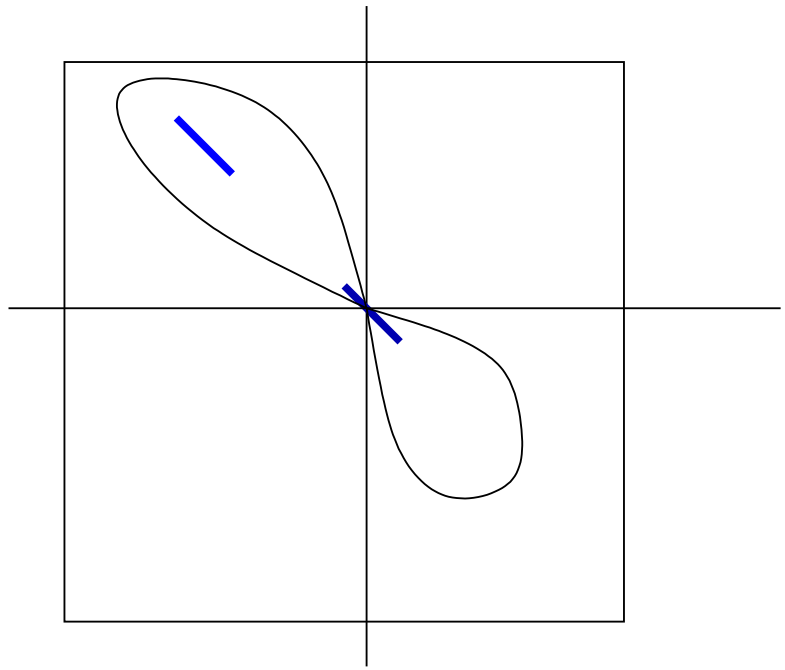
Histoire des algorithmes d'alignement

- Needleman & Wunsch, 1970 pour identifier une ressemblance globale de deux séquences
- Smith & Waterman, 1981 pour identifier une ressemblance locale de deux séquences (il suffit d'ajouter max avec 0 à l'éq. de rec.)
- variation pour ne pas compter les gaps en début/fin
- algorithme du K-band pour limiter la recherche aux alignements avec un petit nombre d'erreurs
- recherche des alignements co-optimaux
- recherche des n meilleurs alignements locaux

Et pour aller plus loin ?

- algorithmes inadaptés pour traiter de grands volumes de données
- mise en place d'heuristiques (telle que BLAST)

idée : indexer les séquences pour connaître toutes les positions de mots courts (de longueur 11 par exemple) qui formeront des candidats potentiels pour un alignement



Alignement multiple

- extension de l'alignement 2 à 2

  entrée k séquences

  ```
  C A T G C G A G T A G T A G
  C A T G G T A G T A G
  C C T G G A G T A C G T A G
  C A T G A G C G T A G
  ```

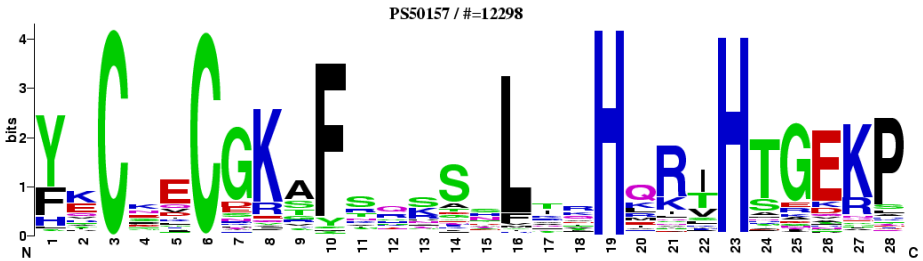
  sortie un tableau contenant les k séquences, avec des indels

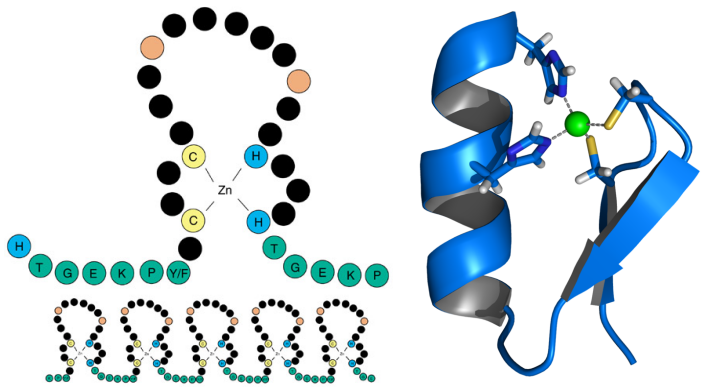
  ```
  C A T G C G A G T A - G T A G
  C A T G - - - G T A - G T A G
  C C T G - G A G T A C G T A G
  C A T G - - A G - - C G T A G
  ```

- algorithmiquement, on peut appliquer la même méthode
- biologiquement, à quelle conservation **syntaxique** correspond la famille de séquences portant la même fonction ?

Motif doigt de zinc (*C2H2-type*)

```
TYY1_HUMAN YVCPFDGCNKKFAQSTNLKSHILT--H
YKQ8_CAEEL YKCT--VCRKDISSSESLRTHMFKQHH
BASO_HUMAN FQCD--ICKKTFKNACSVKIHHKN-MH
ZG2-9_XENL FVCT--VCGKTYKYKHGLNTHLHS--H
P43_XENBO  LKCSVPGCKRSFRKKRALRIHVSE--H
IKAR_MOUSE FECN--MCGYHSQDRYEFSSHITRGEH
TRA1_CAEEL YKCEFADCEKAFSNASDRAKHQNR-TH
ZN10_HUMAN YKCN--QCGIIFSQNSPFIVHQIA--H
XFIN_XENLA FRCS--ECSRSFTHNSDLTAHMRK--H
TF3A_BUFAM CKCETENCNLAFTTASNMRLHFKR-AH
ZG58_XENLA FVCT--ECNLSFAGLANLRSHQHL--H
P43_XENBO  YRCSYEDCQTVSPTWTALQTHLKK--H
TSH_DROME  FRCV--WCKQSFPTLEALTTHMKDSKH
ZN76_HUMAN FRCGYKGCGRLYTTAHHLKVHERA--H
TF3A_BUFAM YRCPRENCDRTYTTKFNLKSHILT-FH
SUHW_DROAN YACK--ICGKDFTRSYHLKRHQKYSSC
ZN76_HUMAN YTCPEPHCGRGFTSATNYKNHVRI--H
SRYC_DROME FKCN--YCPRDFTNFPNWLKHTRR-RH
EVI1_HUMAN YRCK--YCDRSFSISSNLQRHVRN-IH
```





modélisation : motif Prosite

C-x(2,4)-C-x(3)-[LIVMFYWC]-x(8)-H-x(3,5)-H

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

Calculer un alignement multiple

avec un arbre guide

① `TACCATGA` ② `TACCATA` ③ `GACGACCA` ④ `GACCATCTCA`

1. calcul des meilleurs alignements 2 à 2 :

   scores (Mach = 1, Mismatch =-1, Indel = -1)

	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

1. calcul des meilleurs alignements 2 à 2 :

   scores (Mach = 1, Mismatch =-1, Indel = -1)

2. construction d'un arbre de clustering (et de l'alignement) :

	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

1. calcul des meilleurs alignements 2 à 2 :

   scores (Mach = 1, Mismatch =-1, Indel = -1)

2. construction d'un arbre de clustering (et de l'alignement) :

	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

Calculer un alignement multiple

avec un arbre guide

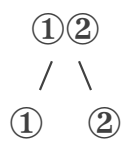
① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

1. calcul des meilleurs alignements 2 à 2 :

   scores (Mach = 1, Mismatch =-1, Indel = -1)

2. construction d'un arbre de clustering (et de l'alignement) :

	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.



Calculer un alignement multiple

avec un arbre guide

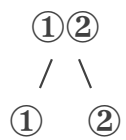
① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

1. calcul des meilleurs alignements 2 à 2 :

   scores (Mach = 1, Mismatch =-1, Indel = -1)

2. construction d'un arbre de clustering (et de l'alignement) :

	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.



① TACCATGA

② TACCAT-A

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

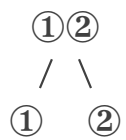
1. calcul des meilleurs alignements 2 à 2 :

   scores (Mach = 1, Mismatch =-1, Indel = -1)

2. construction d'un arbre de clustering (et de l'alignement) :

	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

	①②	③	④
①②	.	0	2.5
③	.	.	4
④	.	.	.



① TACCATGA
② TACCAT-A

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

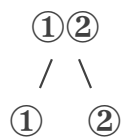
1. calcul des meilleurs alignements 2 à 2 :

   scores (Mach = 1, Mismatch =-1, Indel = -1)

2. construction d'un arbre de clustering (et de l'alignement) :

	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

	①②	③	④
①②	.	0	2.5
③	.	.	4
④	.	.	.



① TACCATGA
② TACCAT-A

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

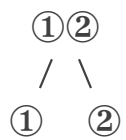
1. calcul des meilleurs alignements 2 à 2 :

   scores (Mach = 1, Mismatch =-1, Indel = -1)

2. construction d'un arbre de clustering (et de l'alignement) :

	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

	①②	③	④
①②	.	0	2.5
③	.	.	4
④	.	.	.



① TACCATGA

② TACCAT-A

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

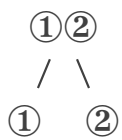
1. calcul des meilleurs alignements 2 à 2 :

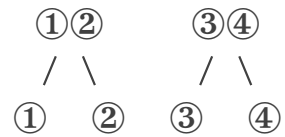
   scores (Mach = 1, Mismatch =-1, Indel = -1)

2. construction d'un arbre de clustering (et de l'alignement) :

	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

	①②	③	④
①②	.	0	2.5
③	.	.	4
④	.	.	.





① TACCATGA

② TACCAT-A

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

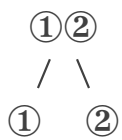
1. calcul des meilleurs alignements 2 à 2 :

   scores (Mach = 1, Mismatch =-1, Indel = -1)

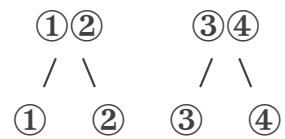
2. construction d'un arbre de clustering (et de l'alignement) :

	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

	①②	③	④
①②	.	0	2.5
③	.	.	4
④	.	.	.



① TACCATGA
② TACCAT-A



③ GACGA-C-CA
④ GACCATCTCA

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

1. calcul des meilleurs alignements 2 à 2 :

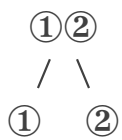
   scores (Mach = 1, Mismatch =-1, Indel = -1)

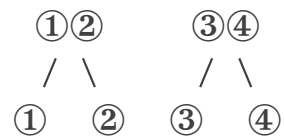
2. construction d'un arbre de clustering (et de l'alignement) :

	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

	①②	③	④
①②	.	0	2.5
③	.	.	4
④	.	.	.

	① ②	③④
①②	.	1.25
③④	.	.





① TACCATGA
② TACCAT-A

③ GACGA-C-CA
④ GACCATCTCA

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

1. calcul des meilleurs alignements 2 à 2 :

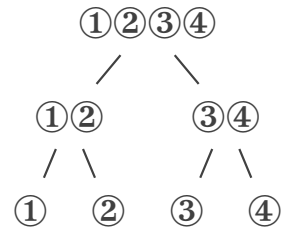
   scores (Mach = 1, Mismatch =-1, Indel = -1)

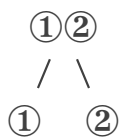
2. construction d'un arbre de clustering (et de l'alignement) :

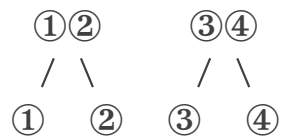
	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

	①②	③	④
①②	.	0	2.5
③	.	.	4
④	.	.	.

	① ②	③④
①②	.	1.25
③④	.	.







① TACCATGA
② TACCAT-A

③ GACGA-C-CA
④ GACCATCTCA

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

1. calcul des meilleurs alignements 2 à 2 :

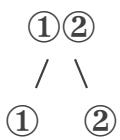
   scores (Mach = 1, Mismatch =-1, Indel = -1)

2. construction d'un arbre de clustering (et de l'alignement) :

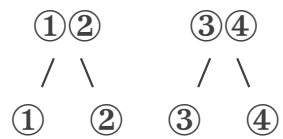
	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

	①②	③	④
①②	.	0	2.5
③	.	.	4
④	.	.	.

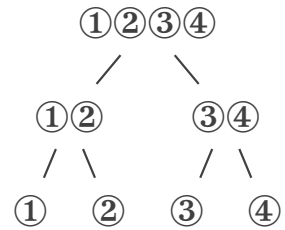
	① ②	③④
①②	.	1.25
③④	.	.



① TACCATGA
② TACCAT-A



③ GACGA-C-CA
④ GACCATCTCA



① TACCAT--GA
② TACCAT---A
③ GACGA-C-CA
④ GACCATCTCA

Calculer un alignement multiple

avec un arbre guide

① TACCATGA ② TACCATA ③ GACGACCA ④ GACCATCTCA

1. calcul des meilleurs alignements 2 à 2 :

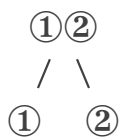
   scores (Mach = 1, Mismatch =-1, Indel = -1)

2. construction d'un arbre de clustering (et de l'alignement) :

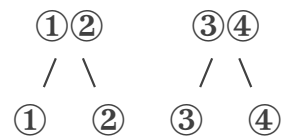
	①	②	③	④
①	.	6	0	2
②	.	.	0	3
③	.	.	.	4
④	.	.	.	.

	①②	③	④
①②	.	0	2.5
③	.	.	4
④	.	.	.

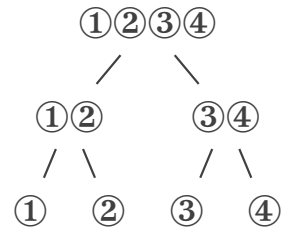
	① ②	③④
①②	.	1.25
③④	.	.



① TACCATGA
② TACCAT-A

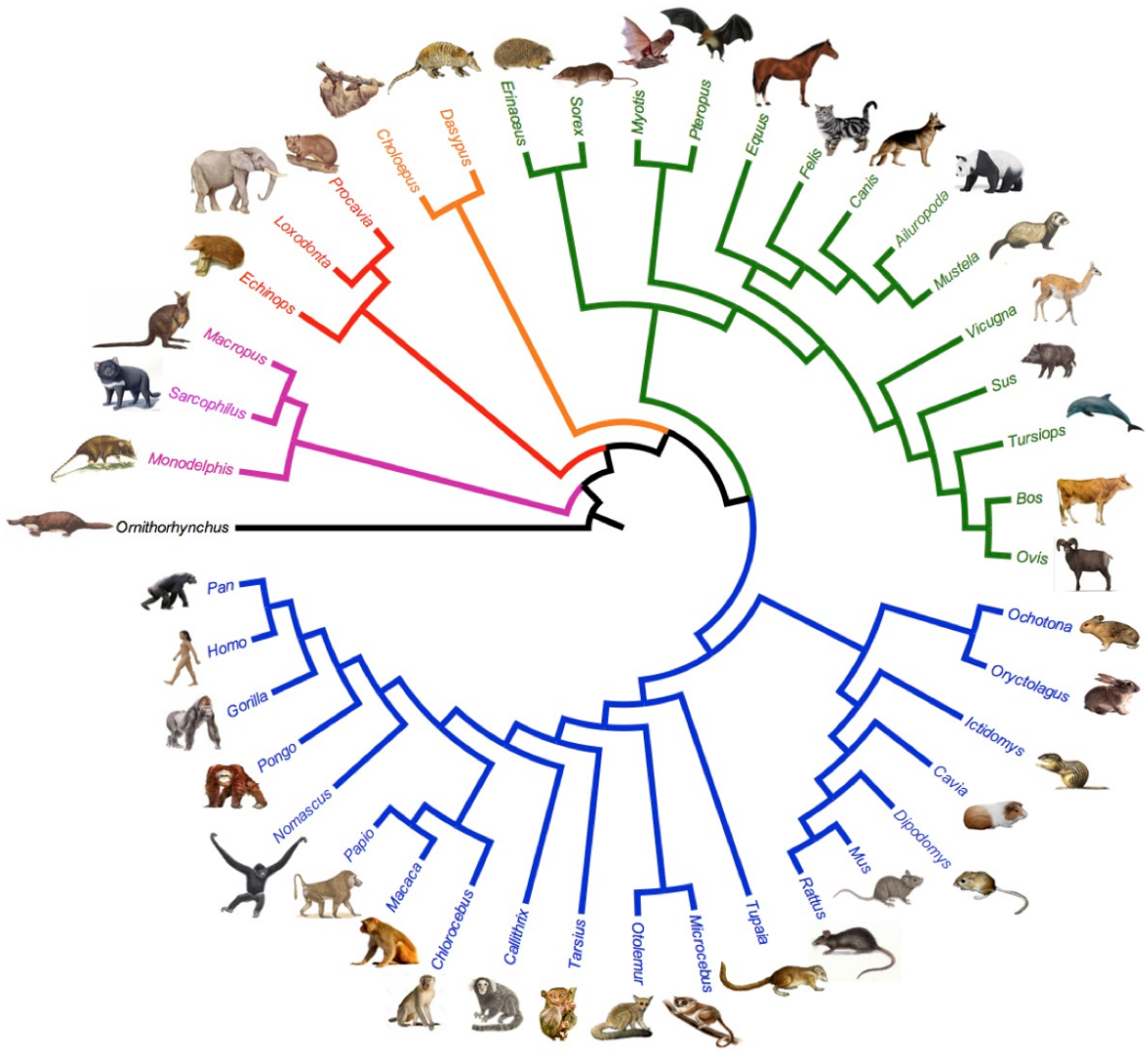


③ GACGA-C-CA
④ GACCATCTCA



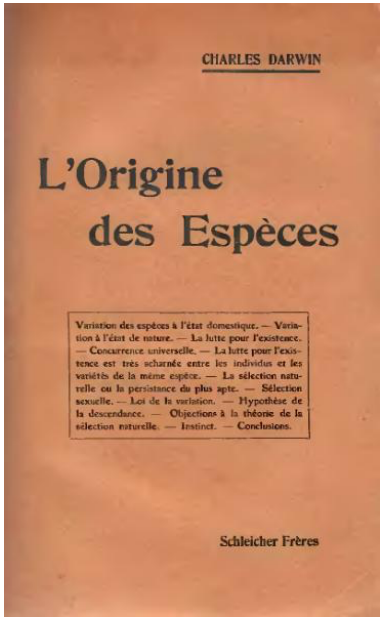
① TACCAT--GA
② TACCAT---A
③ GACGA-C-CA
④ GACCATCTCA

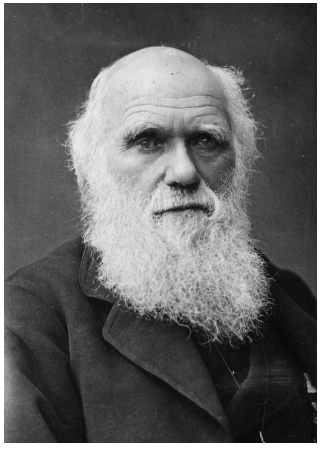
Reconstruction phylogénétique



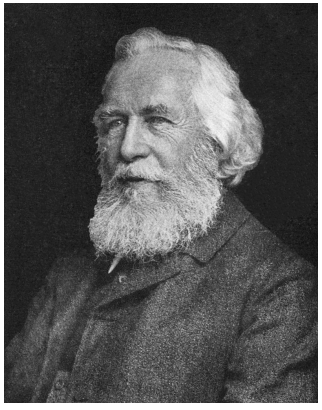
Reconstruction phylogénétique

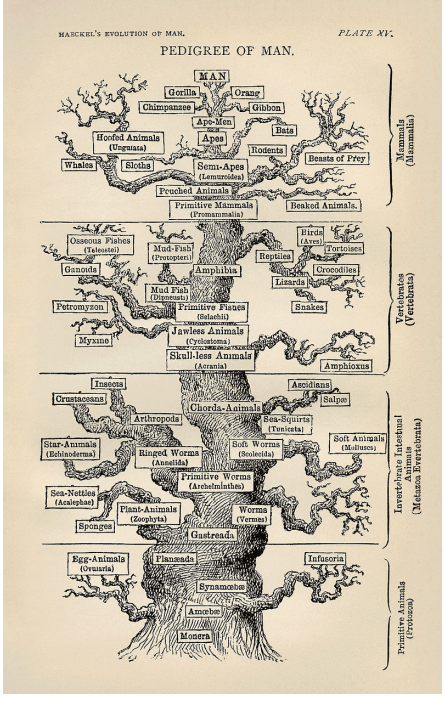
Darwin, 1859



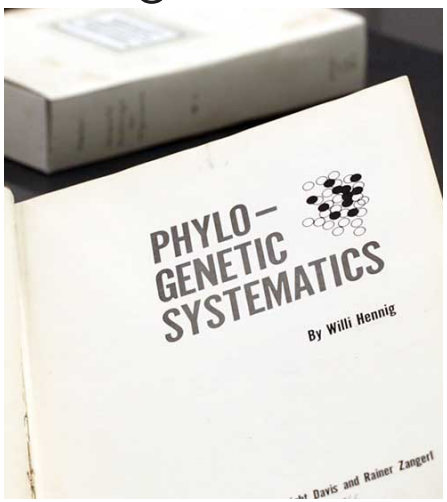


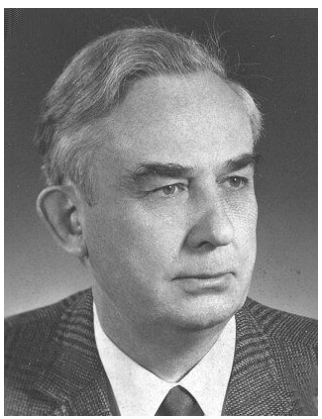
Haeckel, 1879





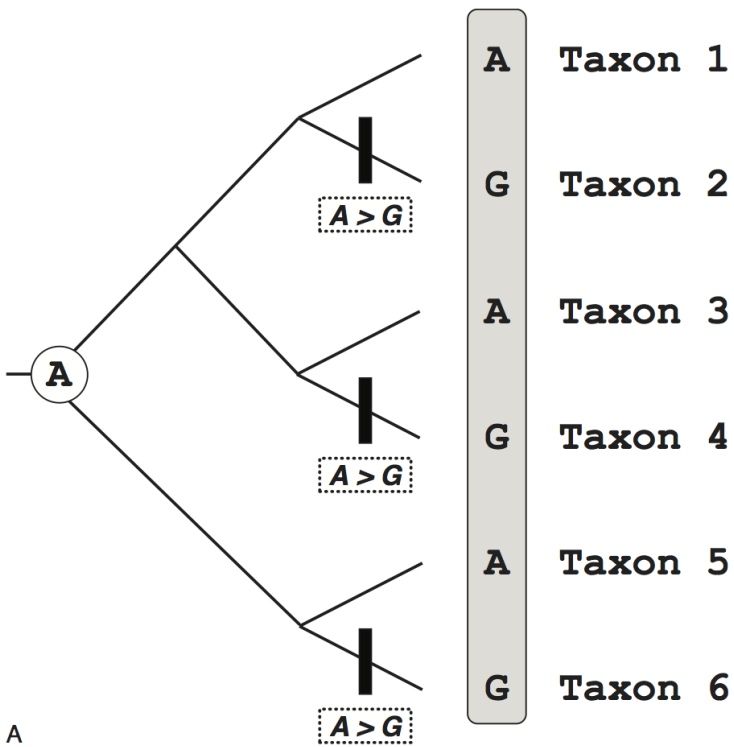
Hennig, 1950





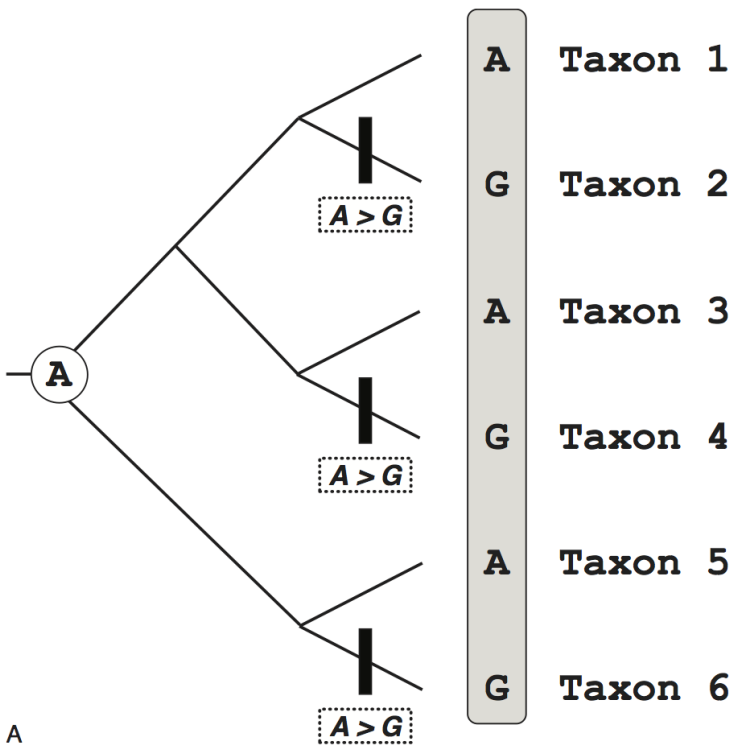
Sous l'éclairage de la phylogénie

« *Rien en biologie n'a de sens, si ce n'est à la lumière de l'évolution.* »
(Theodosius Dobzhansky)

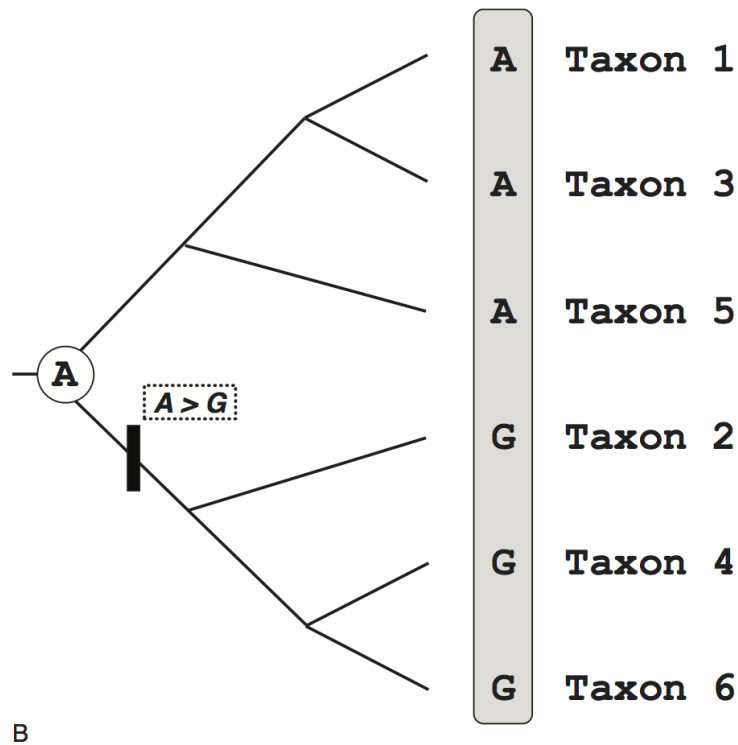


Sous l'éclairage de la phylogénie

« *Rien en biologie n'a de sens, si ce n'est à la lumière de l'évolution.* »
(Theodosius Dobzhansky)

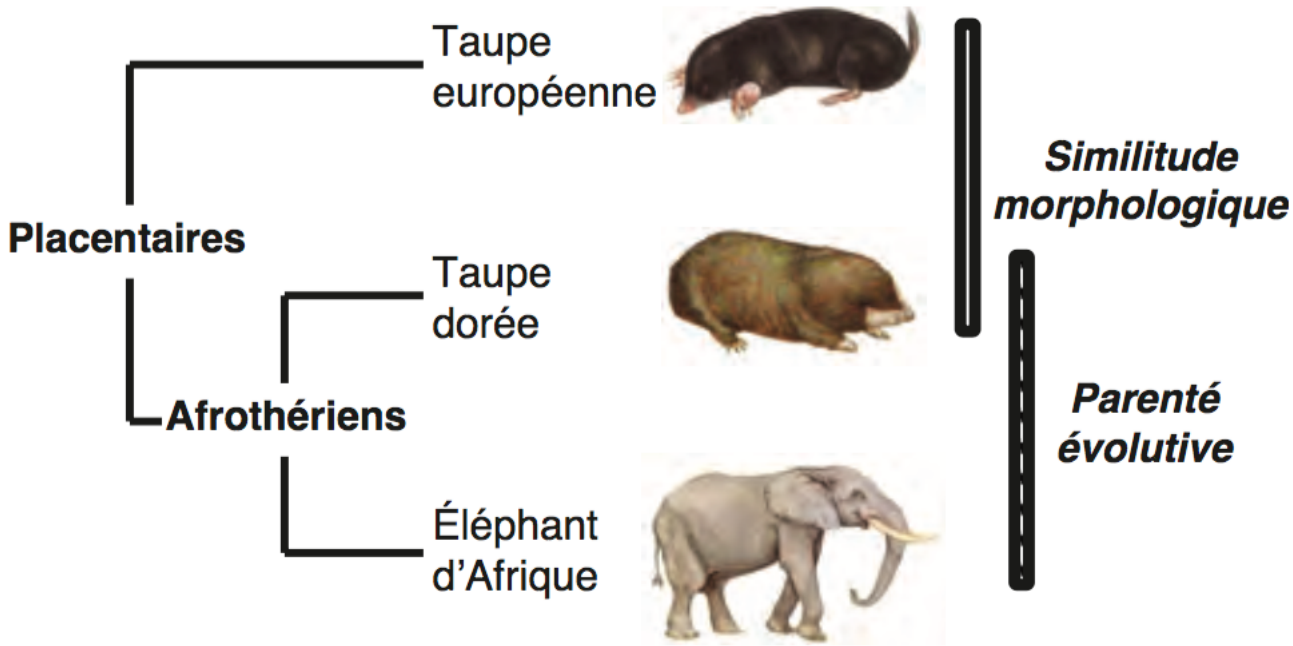


A



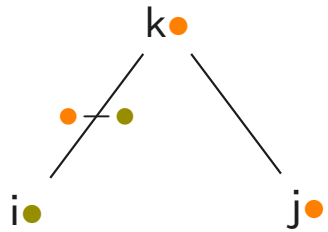
B

De la ressemblance à l'homologie



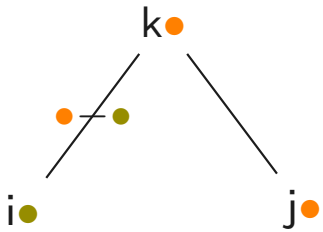
homologie = un caractère partagé par deux espèces ayant un ancêtre commun

L'évolution nous joue des tours

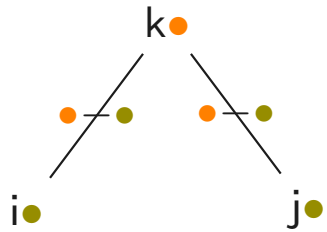


$d(i,j) = 1$
$p(i,j) = 1$

L'évolution nous joue des tours

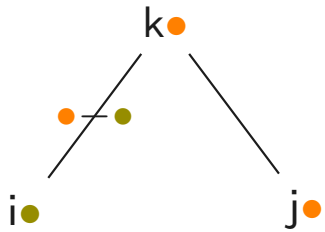


$d(i,j) = 1$
$p(i,j) = 1$

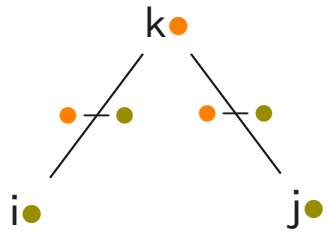


$d(i,j) = 0$
$p(i,j) = 2$
convergence

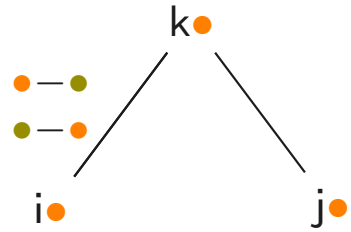
L'évolution nous joue des tours



$d(i,j) = 1$
$p(i,j) = 1$

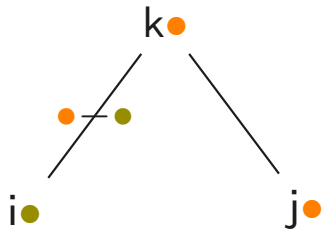


$d(i,j) = 0$
$p(i,j) = 2$
convergence

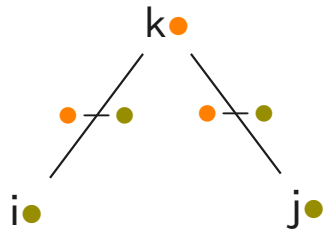


$d(i,j) = 0$
$p(i,j) = 2$
réversion

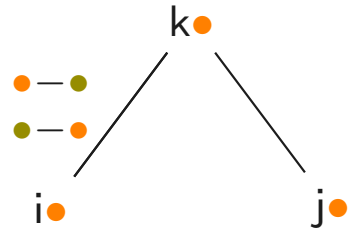
L'évolution nous joue des tours



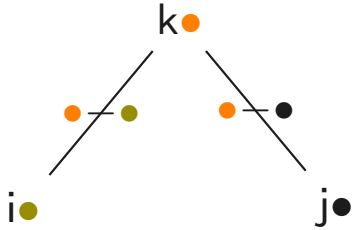
$d(i,j) = 1$
$p(i,j) = 1$



$d(i,j) = 0$
$p(i,j) = 2$
convergence

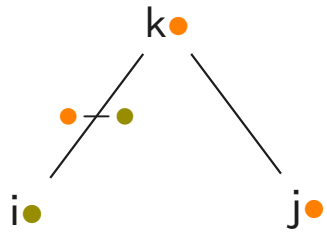


$d(i,j) = 0$
$p(i,j) = 2$
réversion

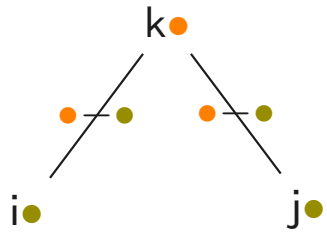


$d(i,j) = 1$
$p(i,j) = 2$
mutations mutiples

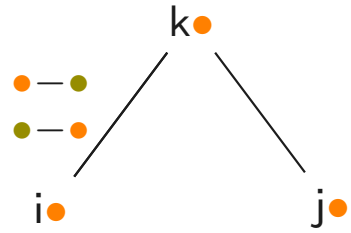
L'évolution nous joue des tours



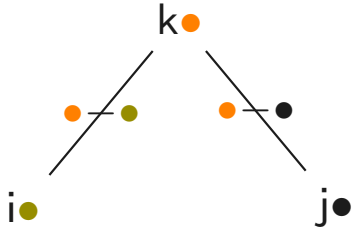
$d(i,j) = 1$
$p(i,j) = 1$



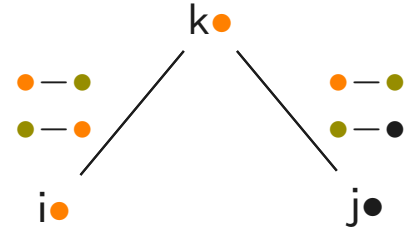
$d(i,j) = 0$
$p(i,j) = 2$
convergence



$d(i,j) = 0$
$p(i,j) = 2$
réversion



$d(i,j) = 1$
$p(i,j) = 2$
mutations mutiples



$d(i,j) = 1$
$p(i,j) = 4$
convergence + réversion + mutations mutiples

Méthode de distances

- l'alignement des séquences fournit un score
- sert d'estimateur de la distance évolutive entre deux séquences
- pour corriger les problèmes distance observée/distance réelle : on pose un modèle d'évolution permettant de corriger
- on procède à une classification hiérarchique

  On construit l'arbre à partir des feuilles en regroupant progressivement les noeuds 2 à 2 pour former des **clusters**.

Activité

A.1 Poser un modèle simple : chaque position évolue indépendamment, des substitutions surviennet aléatoirement selon un processus de Poisson
modèle de Jukes et Cantor, probabilité de substitution s et de conservation $1 - 3s$

A.2 Simuler l'évolution de séquences, conserver le vrai nombre de substitutions (i.e. la distance observée)

A.3 Calculer la distance observée puis la distance corrigée
$d = -\frac{3}{4} \ln\left(1 - \frac{4}{3}p\right)$

A.4 En faire une représentation graphique

Un exemple idéal

matrice de distances :

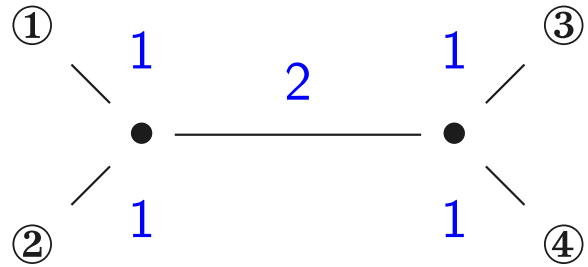
	①	②	③	④
①	0	...		
②	2	0	...	
③	4	4	0	...
④	4	4	2	0

Un exemple idéal

matrice de distances :

	①	②	③	④
①	0	...		
②	2	0	...	
③	4	4	0	...
④	4	4	2	0

arbre calculé :

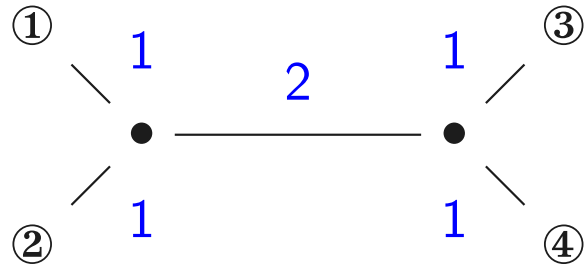


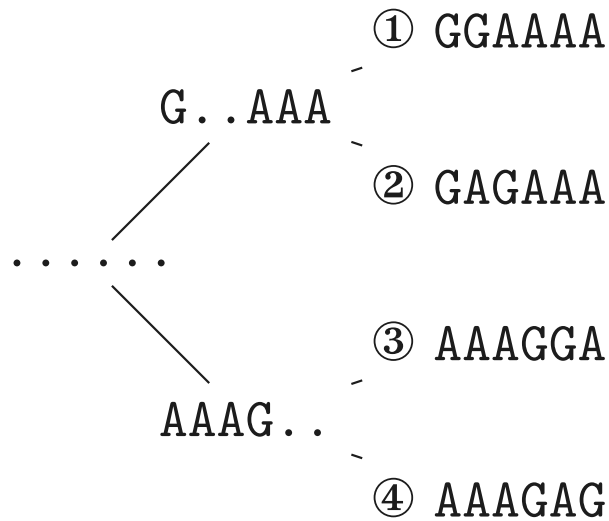
Un exemple idéal

matrice de distances :

	①	②	③	④
①	0	...		
②	2	0	...	
③	4	4	0	...
④	4	4	2	0

arbre calculé :



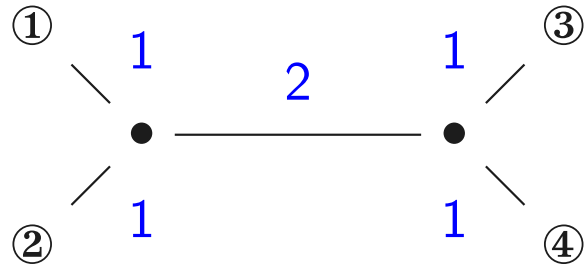


Un exemple idéal

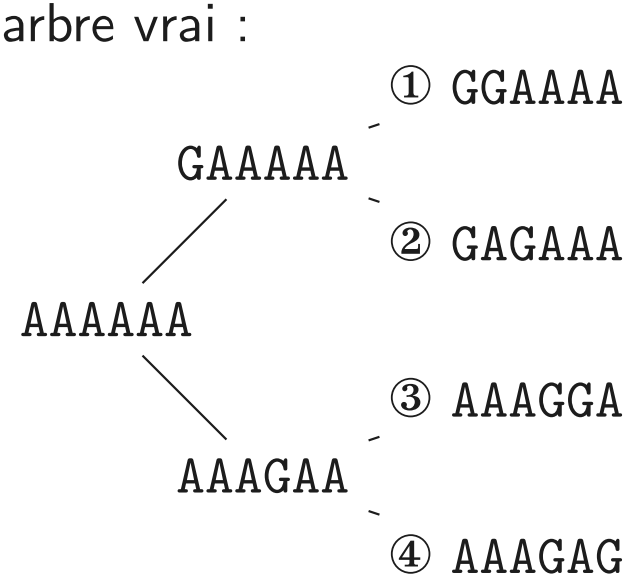
matrice de distances :

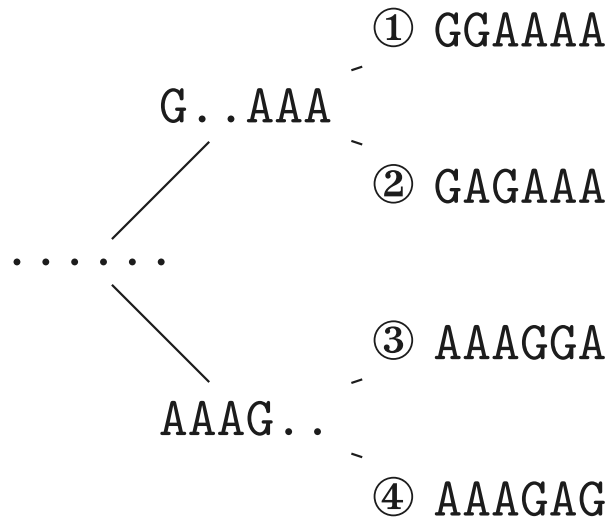
	①	②	③	④
①	0	...		
②	2	0	...	
③	4	4	0	...
④	4	4	2	0

arbre calculé :



arbre vrai :





Un exemple plus réaliste

matrice de distances :

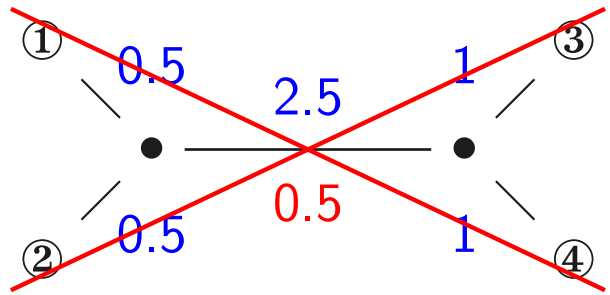
	①	②	③	④
①	0	...		
②	1	0	...	
③	4	3	0	...
④	2	3	2	0

Un exemple plus réaliste

matrice de distances :

	①	②	③	④
①	0	...		
②	1	0	...	
③	4	3	0	...
④	2	3	2	0

arbre calculé :

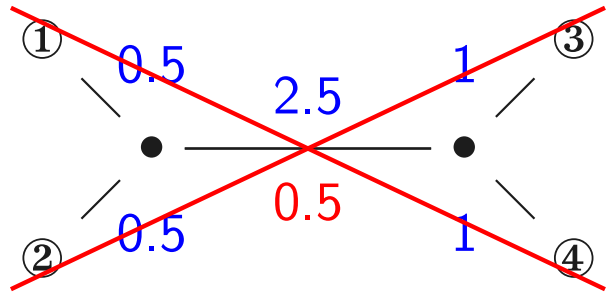


Un exemple plus réaliste

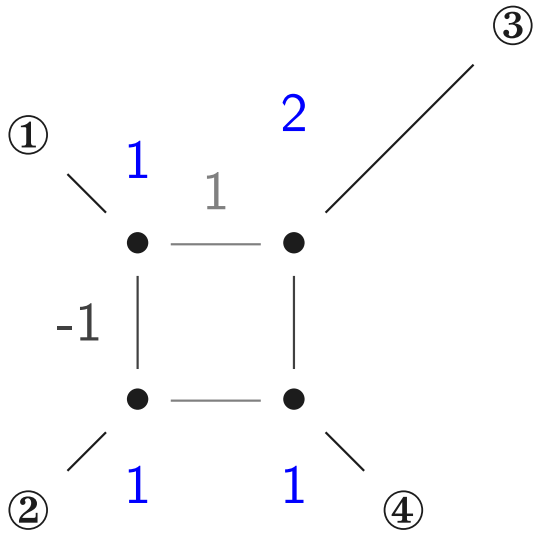
matrice de distances :

	①	②	③	④
①	0	...		
②	1	0	...	
③	4	3	0	...
④	2	3	2	0

arbre calculé :



arbre représentant la matrice :

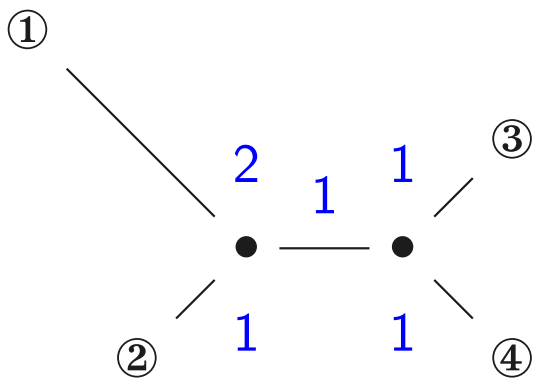


Un exemple plus réaliste

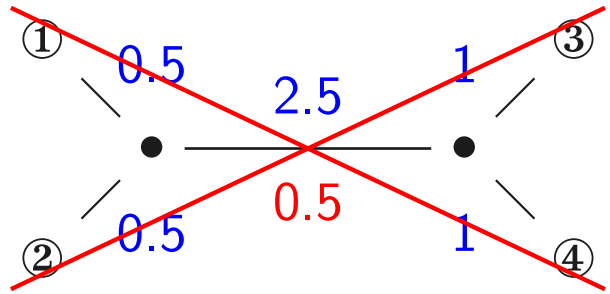
matrice de distances :

	①	②	③	④
①	0	...		
②	3	0	...	
③	4	3	0	...
④	4	3	2	0

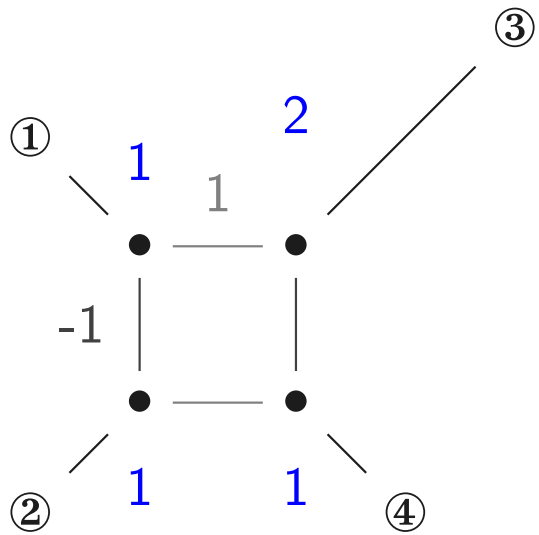
arbre vrai :



arbre calculé :



arbre représentant la matrice :



Neighbor-Joining

Satou et Nei, 1987

- autorise un taux de mutation différent sur les différentes branches,
- à partir des données initiales, calcule une matrice qui donne un arbre en étoile basé sur la divergence des taxons,
- la topologie de l'arbre est obtenu à partir de cette nouvelle matrice de distances,
- les longueurs des branches sont corrigées avec la divergence

Etapes du Neighbor-Joining

1. calcul des divergences
   $r_i = \sum_k d_{ik}$
2. calcul de la nouvelle matrice (représentation en étoile)
   $M_{ij} = d_{ij} - \frac{r_i + r_j}{n-2}$
3. choix des voisins les plus proches a et b, fusion en U, calcul des branches de l'arbre
   $l_{aU} = \frac{d_{ab}}{2} + \frac{r_a - r_b}{2n-4}, \qquad l_{bU} = d_{ab} - l_{aU}$
4. calcul des nouvelles distances pour les autres noeuds k
   $d_{kU} = \frac{d_{ak} + d_{bk} - d_{ab}}{2}$
5. recommencer à partir de cette nouvelle matrice

Neighbor-Joining

Exemple

d_{ij}	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8

Neighbor-Joining

Exemple

d_{ij}	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8

1.

$$r_i = \sum_k d_{ik}$$

i	A	B	C	D	E	F
r_i	30	42	32	38	34	44

Neighbor-Joining

Exemple

d_{ij}	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8

1.

$$r_i = \sum_k d_{ik}$$

i	A	B	C	D	E	F
r_i	30	42	32	38	34	44

2.

$$M_{ij} = d_{ij} - \frac{r_i + r_j}{n - 2}$$

M_{ij}	A	B	C	D	E	F
B	-13					
C	-11.5	-11.5				
D	-10	-10	-10.5			
E	-10	-10	-10.5	-13		
F	-10.5	-10	-11	-11.5	-11.5	

Neighbor-Joining

Exemple

d_{ij}	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8

1.

$$r_i = \sum_k d_{ik}$$

i	A	B	C	D	E	F
r_i	30	42	32	38	34	44

2.

$$M_{ij} = d_{ij} - \frac{r_i + r_j}{n - 2}$$

M_{ij}	A	B	C	D	E	F
B	-13					
C	-11.5	-11.5				
D	-10	-10	-10.5			
E	-10	-10	-10.5	-13		
F	-10.5	-10	-11	-11.5	-11.5	

Neighbor-Joining

Exemple

d_{ij}	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8

1.

$$r_i = \sum_k d_{ik}$$

i	A	B	C	D	E	F
r_i	30	42	32	38	34	44

2.

$$M_{ij} = d_{ij} - \frac{r_i + r_j}{n - 2}$$

M_{ij}	A	B	C	D	E	F
B	-13					
C	-11.5	-11.5				
D	-10	-10	-10.5			
E	-10	-10	-10.5	-13		
F	-10.5	-10	-11	-11.5	-11.5	

3.

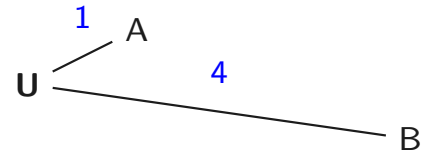
$$l_{A\mathbf{U}} = \frac{d_{AB}}{2} + \frac{r_A - r_B}{2(n-2)} = \frac{5}{2} + \frac{30 - 42}{2(6-2)} = 1$$

$$l_{B\mathbf{U}} = d_{AB} - l_{A\mathbf{U}} = 5 - 1 = 4$$

Neighbor-Joining

Exemple

d_{ij}	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8



1.

$$r_i = \sum_k d_{ik}$$

i	A	B	C	D	E	F
r_i	30	42	32	38	34	44

2.

$$M_{ij} = d_{ij} - \frac{r_i + r_j}{n - 2}$$

M_{ij}	A	B	C	D	E	F
B	-13					
C	-11.5	-11.5				
D	-10	-10	-10.5			
E	-10	-10	-10.5	-13		
F	-10.5	-10	-11	-11.5	-11.5	

3.

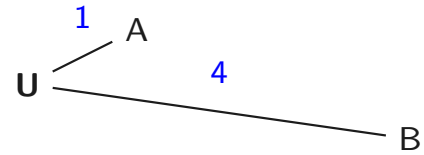
$$l_{A\mathbf{U}} = \frac{d_{AB}}{2} + \frac{r_A - r_B}{2(n-2)} = \frac{5}{2} + \frac{30 - 42}{2(6-2)} = 1$$

$$l_{B\mathbf{U}} = d_{AB} - l_{A\mathbf{U}} = 5 - 1 = 4$$

Neighbor-Joining

Exemple

d_{ij}	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8



4.

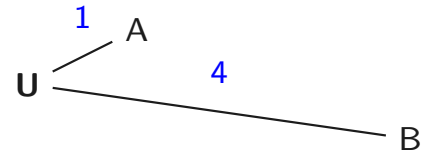
$$d'_{C\mathbf{U}} = \frac{d_{AC} + d_{BC} - d_{AB}}{2} = \frac{4 + 7 - 5}{2}$$

d'_{ij}	**U**	C	D	E	F
C	**3**				
D					
E					
F					

Neighbor-Joining

Exemple

d_{ij}	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8



4.

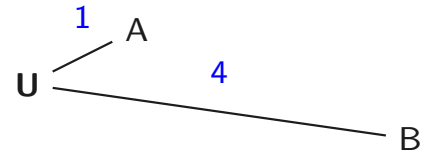
$$\forall k \quad d'_{k\mathbf{U}} = \frac{d_{Ak} + d_{Bk} - d_{AB}}{2}$$

d'_{ij}	**U**	C	D	E	F
C	**3**				
D	**6**	7			
E	**5**	6	5		
F	**7**	8	9	8	

Neighbor-Joining

Exemple

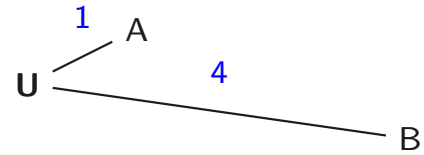
d_{ij}	**U**	C	D	E	F
C	3				
D	6	7			
E	5	6	5		
F	7	8	9	8	



Neighbor-Joining

Exemple

d_{ij}	**U**	C	D	E	F
C	3				
D	6	7			
E	5	6	5		
F	7	8	9	8	



1.

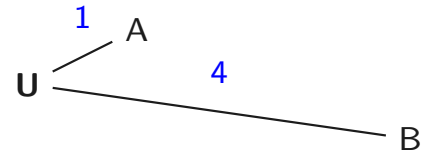
$$r_i = \sum_k d_{ik}$$

i	**U**	C	D	E	F
r_i	21	24	27	24	32

Neighbor-Joining

Exemple

d_{ij}	**U**	C	D	E	F
C	3				
D	6	7			
E	5	6	5		
F	7	8	9	8	



1.

$$r_i = \sum_k d_{ik}$$

i	**U**	C	D	E	F
r_i	21	24	27	24	32

2.

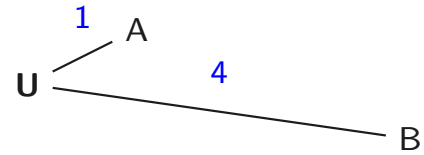
$$M_{ij} = d_{ij} - \frac{r_i + r_j}{n - 2}$$

M_{ij}	**U**	C	D	E	F
C	-12				
D	-10	-10			
E	-10	-10	-12		
F	-10.6	-10.6	-10.6	-10.6	

Neighbor-Joining

Exemple

d_{ij}	**U**	C	D	E	F
C	3				
D	6	7			
E	5	6	5		
F	7	8	9	8	



1.

$$r_i = \sum_k d_{ik}$$

i	**U**	C	D	E	F
r_i	21	24	27	24	32

2.

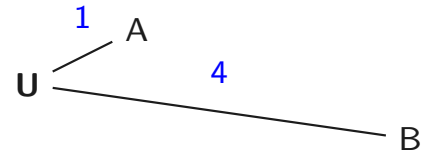
$$M_{ij} = d_{ij} - \frac{r_i + r_j}{n - 2}$$

M_{ij}	**U**	C	D	E	F
C	-12				
D	-10	-10			
E	-10	-10	-12		
F	-10.6	-10.6	-10.6	-10.6	

Neighbor-Joining

Exemple

d_{ij}	**U**	C	D	E	F
C	3				
D	6	7			
E	5	6	5		
F	7	8	9	8	



1.

$$r_i = \sum_k d_{ik}$$

i	**U**	C	D	E	F
r_i	21	24	27	24	32

2.

$$M_{ij} = d_{ij} - \frac{r_i + r_j}{n - 2}$$

M_{ij}	**U**	C	D	E	F
C	-12				
D	-10	-10			
E	-10	-10	-12		
F	-10.6	-10.6	-10.6	-10.6	

3.

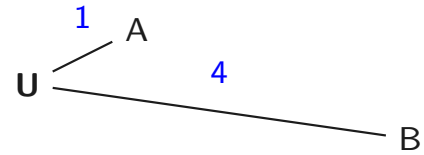
$$l_{D\mathbf{v}} = \frac{d_{DE}}{2} + \frac{r(D) - r(E)}{2(n-2)} = \frac{5}{2} + \frac{27 - 24}{2(5-2)} = 3$$

$$l_{E\mathbf{v}} = d_{DE} - l_{D\mathbf{v}} = 5 - 3 = 2$$

Neighbor-Joining

Exemple

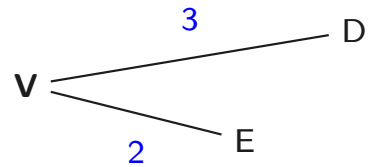
d_{ij}	**U**	C	D	E	F
C	3				
D	6	7			
E	5	6	5		
F	7	8	9	8	



1.

$$r_i = \sum_k d_{ik}$$

i	**U**	C	D	E	F
r_i	21	24	27	24	32



2.

$$M_{ij} = d_{ij} - \frac{r_i + r_j}{n - 2}$$

M_{ij}	**U**	C	D	E	F
C	-12				
D	-10	-10			
E	-10	-10	-12		
F	-10.6	-10.6	-10.6	-10.6	

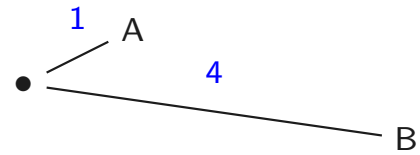
3.

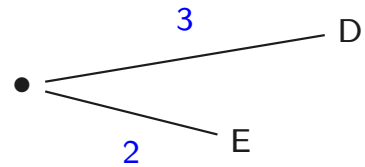
$$l_{D\mathbf{V}} = \frac{d_{DE}}{2} + \frac{r(D) - r(E)}{2(n-2)} = \frac{5}{2} + \frac{27 - 24}{2(5-2)} = 3$$

$$l_{E\mathbf{V}} = d_{DE} - l_{D\mathbf{V}} = 5 - 3 = 2$$

Neighbor-Joining

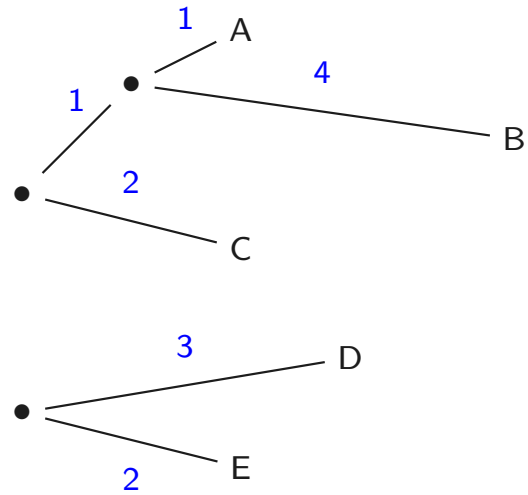
Exemple





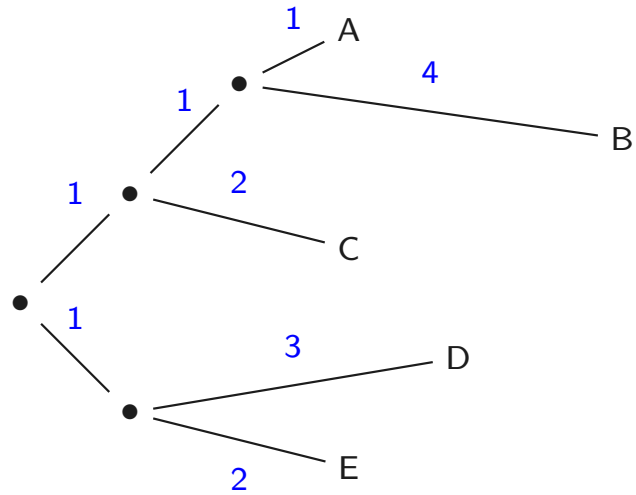
Neighbor-Joining

Exemple



Neighbor-Joining

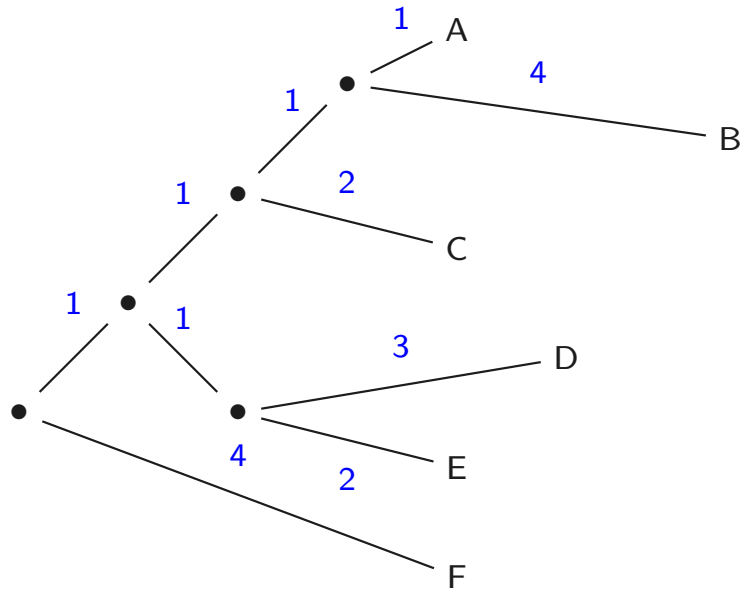
Exemple



Neighbor-Joining

Exemple

d_{ij}	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8



Il existe de nombreuses autres méthodes de RP

bien plus performantes

- les méthodes probabilistes
  - maximum de vraisemblance : quel est l'arbre le plus vraisemblable dans le modèle d'évolution

calcul de $p(\mathcal{D}|\mathcal{T})$, la probabilité des données $\mathcal{D}$ (i.e. de l'alignement multiple) sachant l'arbre $\mathcal{T}$

$$p(\mathcal{D}|\mathcal{T}) = \Pi_{i=1}^{N} p(\mathcal{D}_i|\mathcal{T})$$

  - inférence bayésienne :

calcul de $p(\mathcal{T}|\mathcal{D})$, la probabilité que l'arbre $\mathcal{T}$ soit le vrai sachant les données $\mathcal{D}$ (i.e. de l'alignement multiple)

$$p(\mathcal{T}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{T})p(\mathcal{T})}{\sum_T p(\mathcal{D}|\mathcal{T})p(\mathcal{T})}$$

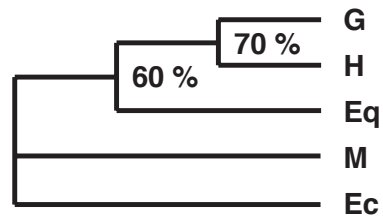
Puis-je croire en mon arbre ?

La méthode de bootstrap

Matrice originale

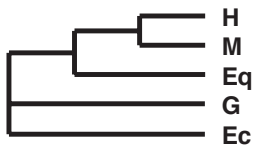
	0 7	0 9	1 0	1 1	4 5	6 5	7 5	8 6	9 6
Glaucomys	C	G	T	T	A	T	A	A	A
Mus	C	A	C	T	G	C	G	A	G
Homo	C	A	T	A	A	C	A	C	A
Equus	T	A	C	A	A	C	A	G	A
Echinops	T	G	C	A	G	T	G	G	G

Consensus de bootstrap



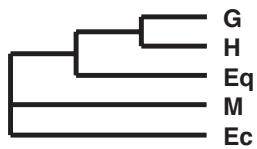
Réplication N°1

	000114669
	799015556
Glaucomys	CGGTTATTA
Mus	CAACTGCCG
Homo	CAATAACCA
Equus	TAACAACCA
Echinops	TGGCAGTTG



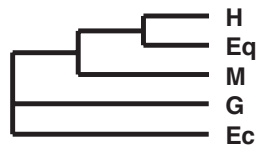
N°2

	001144477
	790055555
Glaucomys	CGTTAAAAA
Mus	CACCGGGGG
Homo	CATTAAAAA
Equus	TACCAAAAA
Echinops	TGCCGGGGG



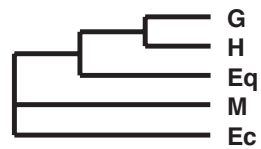
N°3

	000118999
	999116666
Glaucomys	GGGTTAAAA
Mus	AAATTAGGG
Homo	AAAAACAAA
Equus	AAAAAGAAA
Echinops	GGGAAGGGG



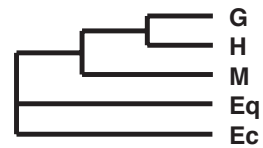
N°4

	111114448
	000115556
Glaucomys	TTTTTAAAA
Mus	CCCTTGGGA
Homo	TTTAAAAAC
Equus	CCCAAAAAG
Echinops	CCCAAGGGG



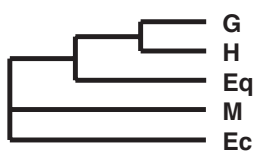
N°5

	001488889
	790566666
Glaucomys	CGTAAAAAA
Mus	CACGAAAAG
Homo	CATACCCCA
Equus	TACAGGGGA
Echinops	TGCGGGGGG



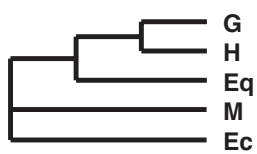
Réplication N°6

	014448999
	705556666
Glaucomys	CTAAAAAAA
Mus	CCGGGAGGG
Homo	CTAAACAAA
Equus	TCAAAGAAA
Echinops	TCGGGGGGG



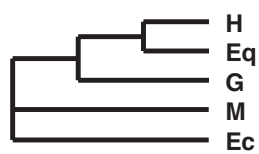
N°7

	001114699
	790015566
Glaucomys	CGTTTAAAA
Mus	CACCTGGGG
Homo	CATTAAAAA
Equus	TACCAAAAA
Echinops	TGCCAGGGG



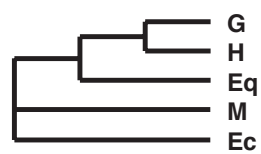
N°8

	014489999
	715566666
Glaucomys	CTAAAAAAA
Mus	CTGGAGGGG
Homo	CAAACAAAA
Equus	TAAAGAAAA
Echinops	TAGGGGGGG



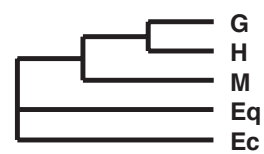
N°9

	011167899
	900155666
Glaucomys	GTTTTAAAA
Mus	ACCTCGAGG
Homo	ATTACACAA
Equus	ACCACAGAA
Echinops	GCCATGGGG

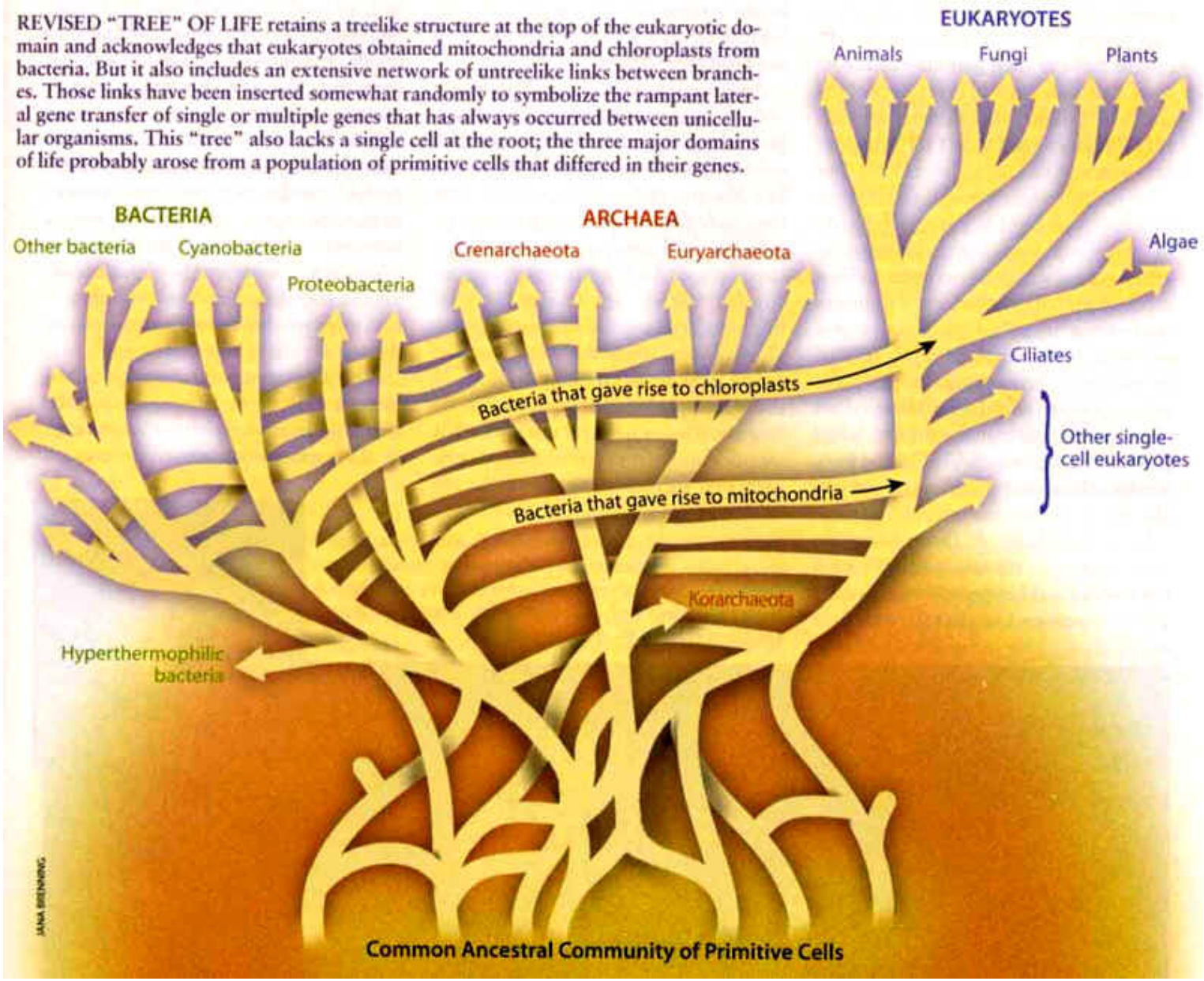


N°10

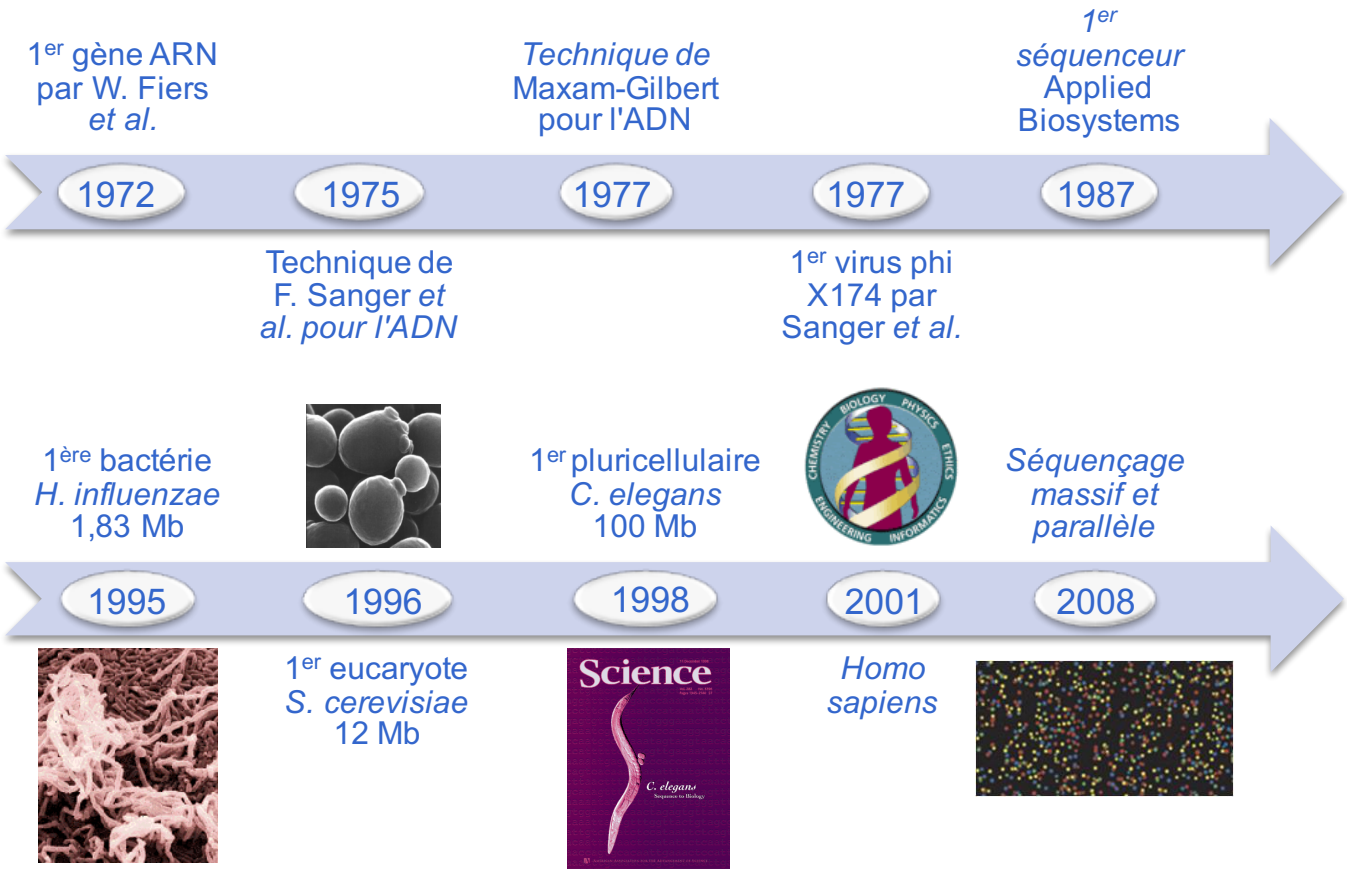
	001111689
	770001566
Glaucomys	CCTTTTTAA
Mus	CCCCCTCAG
Homo	CCTTTACCA
Equus	TTCCCACGA
Echinops	TTCCCATGG



Une histoire bien plus compelexe ...



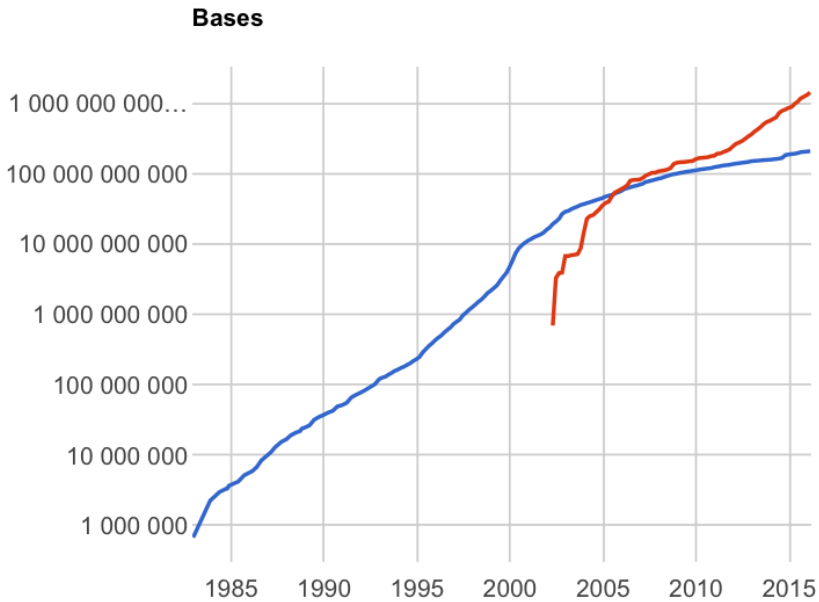
Petite histoire du séquençage

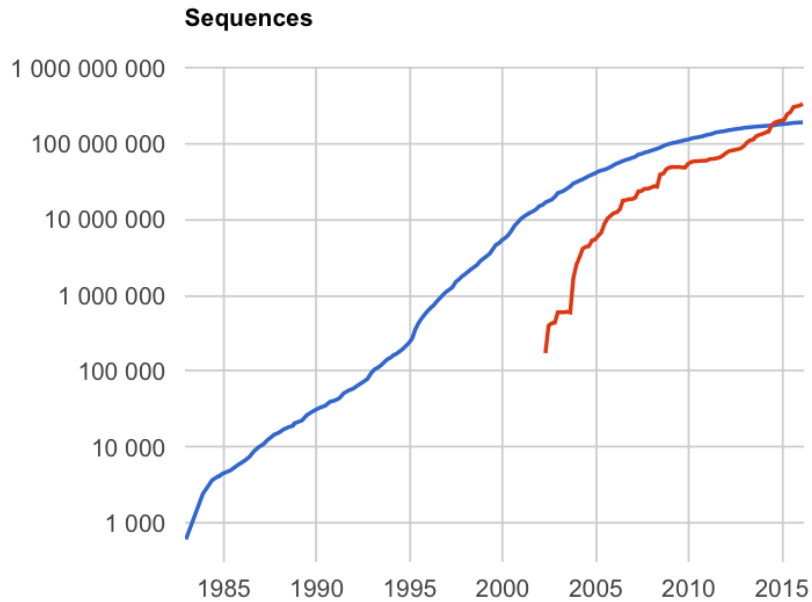


Genbank (release 212, février 2016)

GenBank		WGS	
Bases	Sequences	Bases	Sequences
207,018,196,067	190,250,235	1,399,865,495,608	333,012,760

Croissance de la taille des banques de données





WGS / Sequences

Banques de données

- EMBL (European Bioinformatics Institute, depuis 1992)
- GenBank (National Center for Biotechnology Information, depuis 1988)
- Ensembl (génomes eukaryotes)
- PubMed (banques de publications scientifiques)

Remerciements

Le matériel est issu principalement des cours de bio-informtique donnés à l'Université Lille 1 (auteurs : Sylvain Legrand, Laurent Noé, Hélène Touzet, Maude Pupin, Jean-Stéphane Varré).
Remerciements à Samuel Blanquart pour la nouvelle version du chapitre de livre ≪ Phylogénie Moléculaire ≫ du livre ≪ Biologie Evolutive ≫.