

# Calcul moléculaire

# Ordinateur moléculaire

Pourquoi ?

- vitesse
- coûts
- «car c'est là»
- médecin dans la cellule (Labo de Shapiro) :  
observe les chemins de signalisation, produit des signaux (c-à-d enzymes)  
lui-même, traitement de maladies au niveau cellulaire

# Implantation par biotech

Rappel : notre répertoire d'outils de biotechnologie

- hybridation
- PCR
- enzymes de restriction
- clonage
- electrophoresis
- séquençage

# Encodage

Encodage d'information par ADN : c'est presque trivial :)

Exemple : message secret (*microdot*)

A=CGA      K=AAG

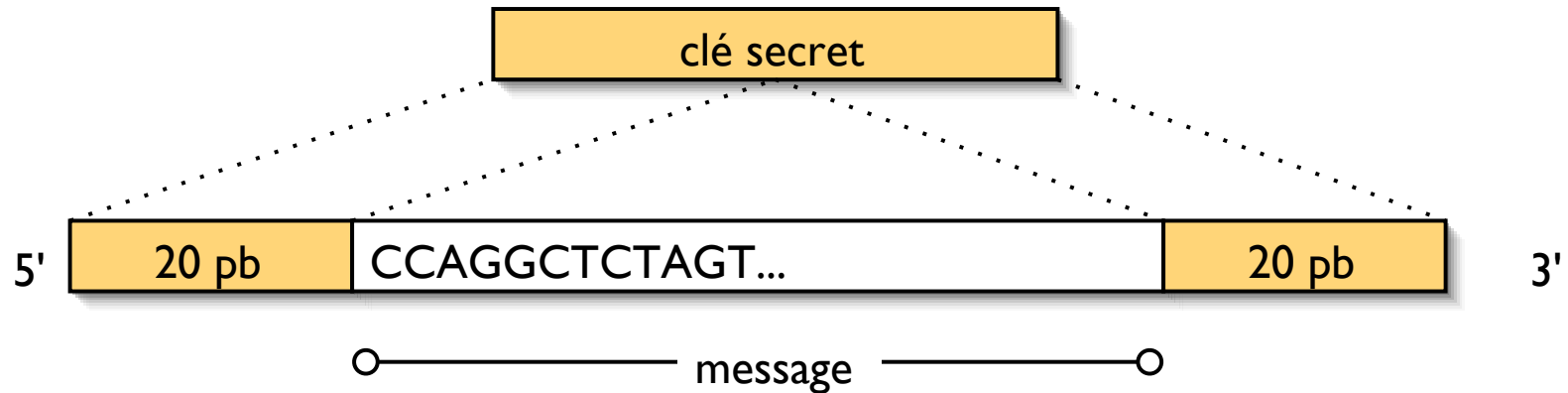
B=CCA      L=TGC

C=GTT      M=TCC

...

«Bonjour la tristesse» → CCAGGCTCTAGT...

# Message secret - 2



mélanger le message secret avec d'autres molécules ADN

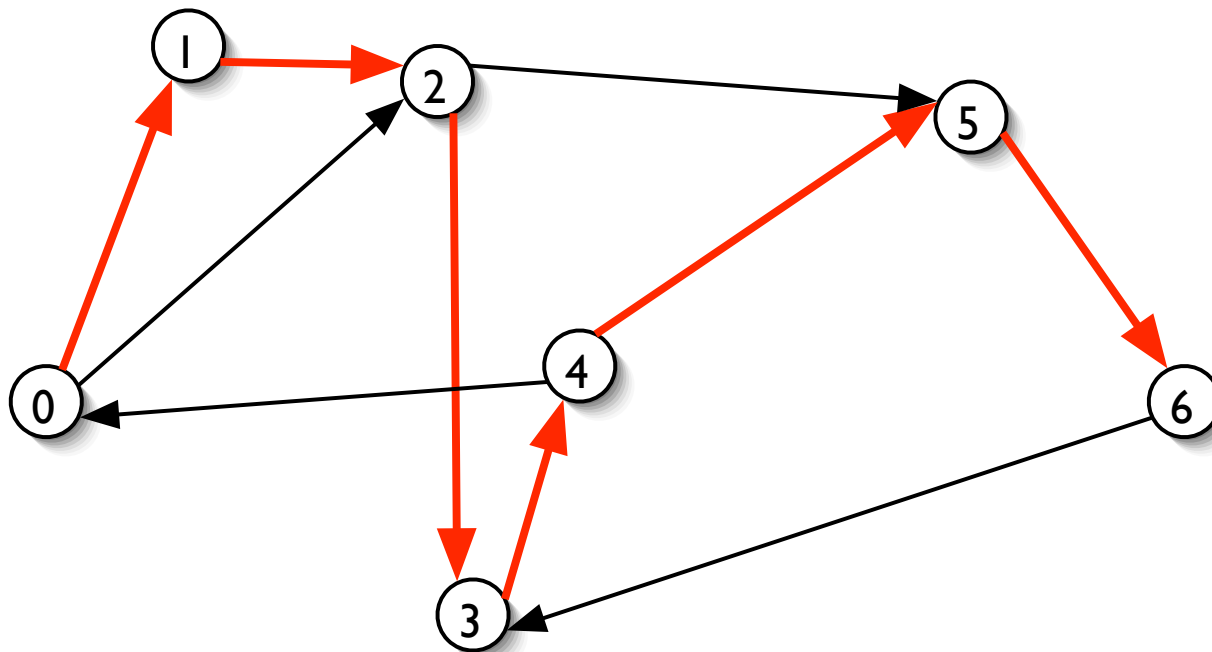
amplification du message par PCR — il faut savoir le clé secret (nombre de possibilités :  $4^{40}$ )

Clellan et al., Nature 399 :533



# Encodage de graphes

Expérience classique de Leonard Adleman (1994) : chemin Hamiltonien



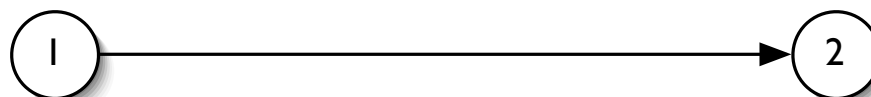
## Encodage de graphes - 2

Noeuds encodés par oligonucléotides  $O_i = O_i^{(g)}O_i^{(d)}$ , arêtes encodés par oligos  $O_{ij} = O_i^{(d)}O_j^{(g)}$

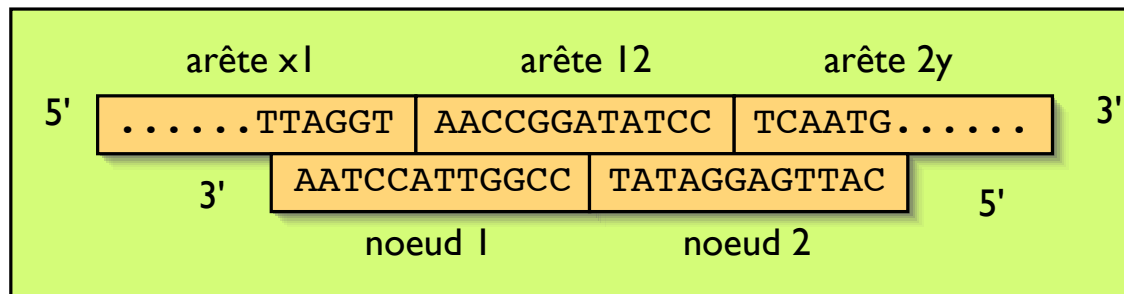
mélanger :  $\overline{O_i}$ , pour  $i = 1, \dots, 5$  (Watson-Crick complément);  $O_{ij}$



# Encodage de graphes - 3



$0_{12}$



chemin encodé

# Encodage de graphes - 4

chaque molécule formée correspond à un chemin :

1. amplification par PCR (filtrer les chemins  $0 \rightarrow 6$ )
2. mesurer la longueur
3. filtrer par chaque  $O_i$  pour vérifier sa présence

Problèmes : espace exponentiel (nbre exponentiel de chemins), modèle de calcul (formation de molécules en une dimension à partir d'un ensemble de pièces) est décidable  $\Rightarrow$  n'est pas aussi puissant que la machine de Turing

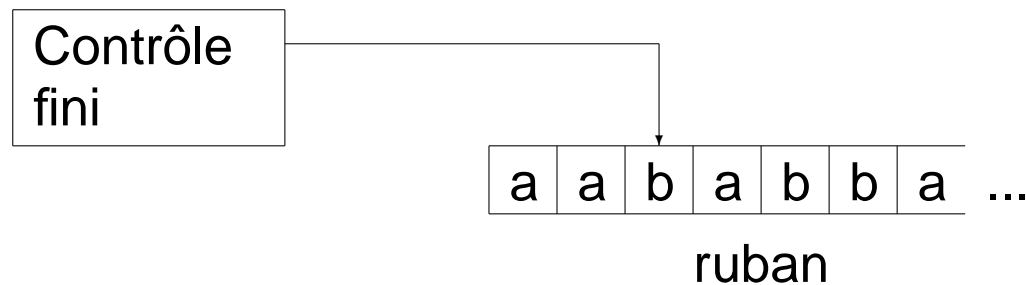
# Calcul moléculaire

logiciel, entrée, sortie, hardware

on veut implanter une machine de Turing universelle à l'aide d'ADN, enzymes, etc.

# Machine de Turing

**Modèle :**



**Caractéristiques :**

- Une machine de Turing peut lire ou écrire
- La tête de lecture peut bouger à gauche ou à droite
- Le ruban est infini vers la droite
- Quand on décide d'accepter ou de rejeter, c'est une décision finale.

# Machine de Turing




## DÉfinition 3.1 :

Une **machine de Turing (MT)** est un 7-tuplet  $(Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$  o

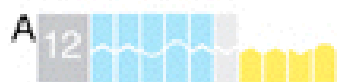
- $Q$  est l'ensemble fini d'**états**,
- $\Sigma$  est l'**alphabet**,
- $\Gamma$  est l'**alphabet de ruban**,  
 $\sqcup \in \Gamma$  et  $\Sigma \subseteq \Gamma$ ,
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$   
est la **fonction de transition**,
- $q_0$  est l'état **initial**,
- $q_a$  est l'état **acceptant**,
- $q_r$  est l'état **rejetant**.

# Calcul par automate fini

Automate fini à deux états :  $S_0$  et  $S_1$ , alphabet :  $\{a, b\}$

Symbol	a	b	terminator (t)
encodings & $\langle \text{state, symbol} \rangle$ sticky ends	$\langle S_1, a \rangle$  $\langle S_0, a \rangle$	$\langle S_1, b \rangle$  $\langle S_0, b \rangle$	$\langle S_1, t \rangle$  $\langle S_0, t \rangle$

# AF : transitions



T1:  $SO \xrightarrow{a} SO$



T5:  $S1 \xrightarrow{a} SO$



T2:  $SO \xrightarrow{a} S1$



T6:  $S1 \xrightarrow{a} S1$



T3:  $SO \xrightarrow{b} SO$



T7:  $S1 \xrightarrow{b} SO$

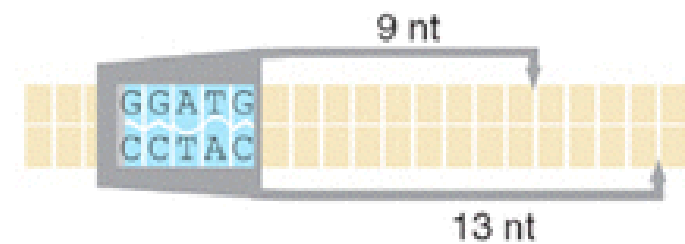
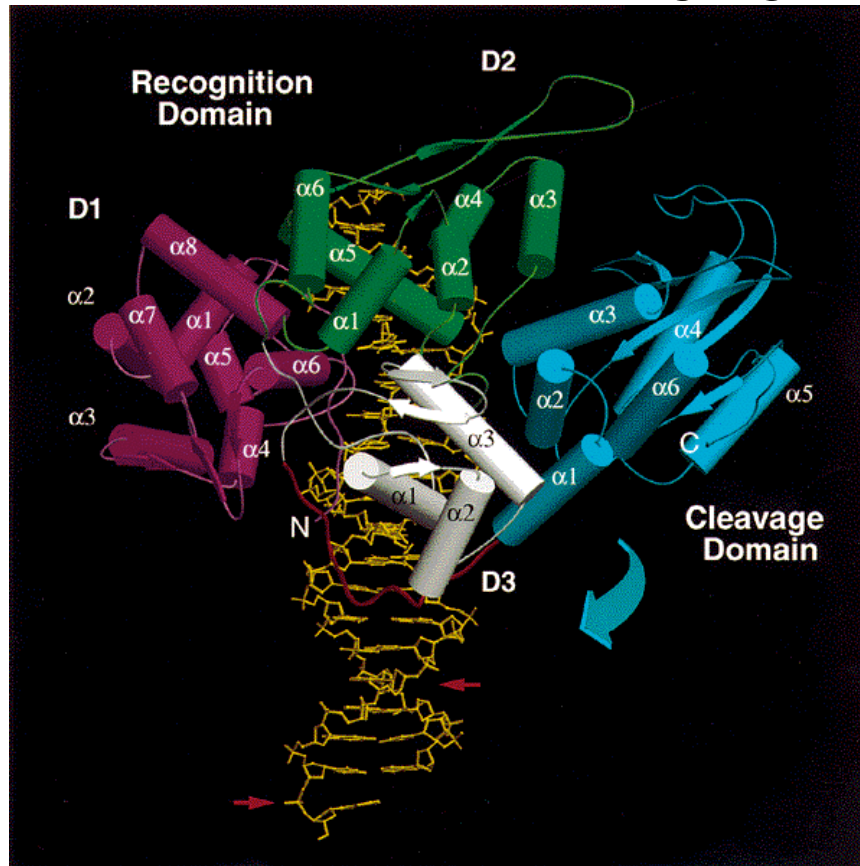


T4:  $SO \xrightarrow{b} S1$



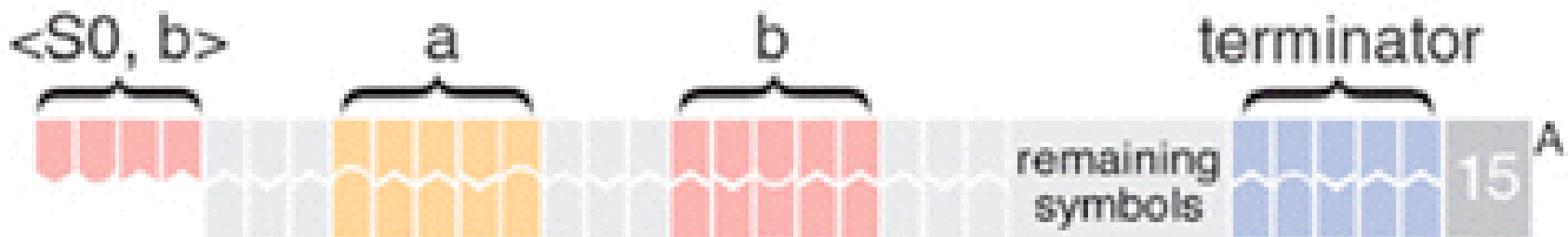
T8:  $S1 \xrightarrow{b} S1$

# AF : hardware — *FokI*





# AF : configurations



# AF : calcul

