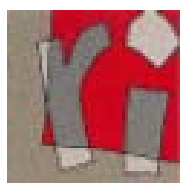
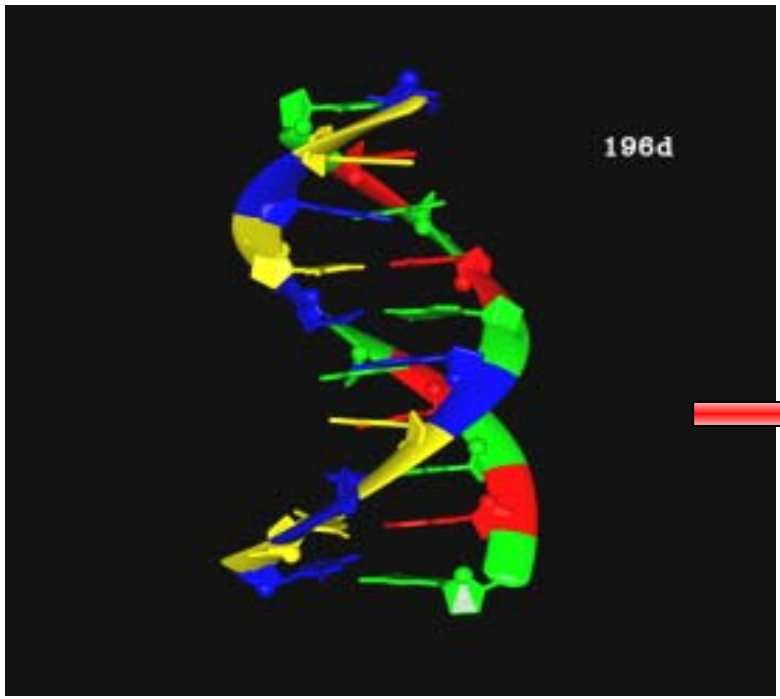
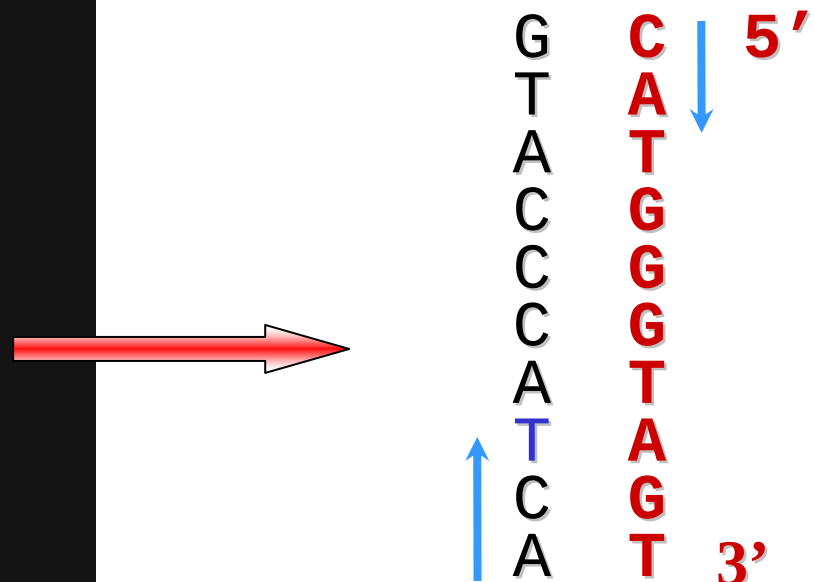
Séquences et structures aléatoires pour l'analyse des génomes


Alain Denise
Bioinformatique
LRI Orsay

UMR CNRS 8623
Université Paris-Sud 11

Structure de l'ADN





Analyse d'un génome

Gène protéique

Promoteur

Origine de réplication

transcription

ARNm

traduction

Protéine

ARNt
ARNr
Introns
...

- Faire l'inventaire du contenu génétique.
- Puis comprendre son organisation, les relations entre structure et fonction de l'information, les processus qui permettent son expression.

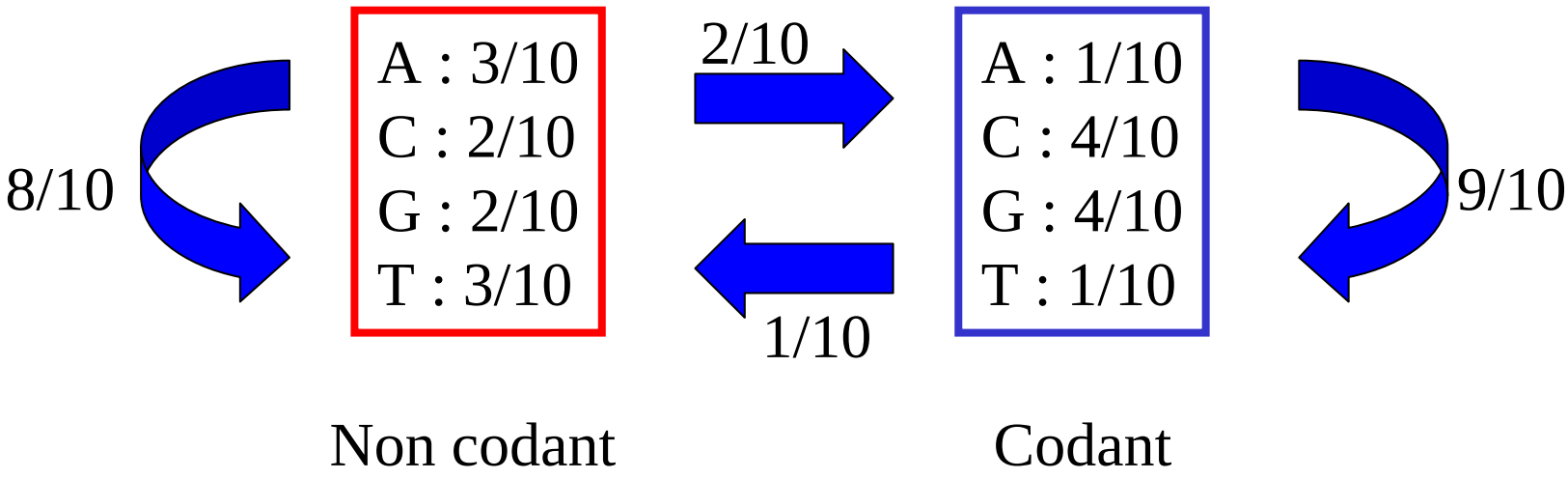
CACCACAATTGCAAAACTCCCAAGCCCGTCCACAAAAGAAGGACGGATTCTCACAGTTCATGCCATCTGCAACTACGAAGAACCCATATGCCCAGTAACT
CGACCGACTGGTTGTAATTTTACAAAAAGAGAGACAATTAAGAAAAGAAACAAGCGCCAGGCTTCCGTATCCCAGTTTTTCATCTCACTTTCTGGGCACG
ATTGTAATAATACTTCATGATAATAACTAAACTATATAAGTAGTGTCTCATCCGTAAATATACATTTAGACAGATTCTTGTATTTTCTCCGGGCAATTTT
TAACTTTTTTTCTGTTAGGGCACATGACACTTGCCTATTATGGACAGCCAGTAAAGATGTGCCCATATATTGCCCCCTTTACGCTCTCTGCCAGTATTAG
TGGGAAAAAAAAAAACTGAAAAAAAAAAATCGCAGGACTACTAATAATCACGTGATATTTCTTTTCACTCTCTTCATAAAGTTGCTAAAAACACACAATCG
AATGAGCCTCTGAGCAGTATAAATTGTACTTCAAAGCACTATGCATGAAAAACGCTTACATTAGTTCAGTTTGTCAAGGTTATGCTATTACTTGTACTTA
TTTCTTGCTATTGTTAGTGGCTCCCCACATTGACGTATTTTCACGTGATGCGCCTCACTGCGGAAGGCGCCACACATTGCCTGCAAAAAATTGTGGATGC
ACTCATTTGATAGTAAACTAAGTCATGTTAATCGTTTGGATTTGGCACACACCCACAAATATACACATTACATATATATATATATTCAAAATACAGCTGC
GTCCAATAGATGAGCTTCCGCTTCGTTGTACAACCTACCTGCTATCTTGTTCACGGATATTTCTTGCTTTTAATAAACAAAAGTAACTCTAGAACAGTCA
AGTCTTCGATAATTTTTTTAGTCACAGGGTCCGTCTAAAGTTTCTCTTTATTTGGAATAATAGAAAAGAAAGAAAAAAACGTAGTATAAAAGGAATGTCG
CATACTTTAAAATCGAAAACGCTCCAAGAGCTGGACATTGAGGAGATTAAGGAAACTAACCCATTGCTCAAACTAGTTCAAGGGCAGAGGATTGTTCAAG
TTCCGGAACTAGTGCTTGAGTCTGGCGTGGTCATAAATAATTTCCCTATTGCTTATAAGACGTGGGGTACACTGAATGAAGCTGGTGATAATGTTCTGGT
AATTTGTCATGCCTTGACTGGGTCCGCAGATGTTGCTGACTGGTGGGGCCCTCTTCTGGGTAACGACTTAGCATTCGACCCATCAAGGTTTTTTATCATA
TGTTTAAACTCTATGGGCTCTCCATATGGGTCTTTTTCGCCATTAACGATAAATGAGGAGACGGGCGTTAGATATGGACCCGAATTCCCATTATGTACTG
TGCGCGATGACGTTAGAGCTCACAGAATTGTTCTGGATTCTCTGGGAGTAAAGTCAATAGCCTGTGTTATTGGTGGCTCTATGGGGGGGATGCTGAGTTT
GGAATGGGCTGCCATGTATGGTAAGGAATATGTGAAGAATATGGTTGCTCTGGCGACATCAGCAAGACATTCTGCCTGGTGCATATCGTGGTCTGAGGCT
CAAAGACAATCGATTTACTCAGATCCCAACTACTTGGACGGGTACTATCCGGTAGAGGAGCAACCTGTGGCCGGACTATCGGCTGCACGTATGTCTGCAT
TGTTGACGTACAGGACAAGAAACAGTTTCGAGAACAAATTCTCCAGAAGATCTCCTTCAATAGCACAACAACAAAAAGCTCAAAGGGAGGAGACACGCAA
ACCATCTACTGTCAGCGAACACTCCCTACAAATCCACAATGATGGGTATAAAACAAAAGCCAGCACTGCCATCGCTGGCATTTCTGGGCAAAAAGGTCAA
AGCGTGGTGTCCACCGCATCTTCTTCGGATTCATTGAATTCTTCAACATCGATGACTTCGGTAAGTTCTGTAACGGGTGAAGTGAAGGACATAAAGCCTG
CGCAGACGTATTTTTCTGCACAAAGTTACTTGAGGTACCAGGGCACAAAGTTCATCAATAGGTTCGACGCCAATTGTTACATTGCCATCACACGTAAACT
GGATACGCACGATTTGGCAAGAGACAGAGTAGATGACATCACTGAGGTCCTTTCTACCATCCAACAACCATCCCTGATCATCGGTATCCAATCTGATGGA
CTGTTCACATATTCAGAACAAGAATTTTTGGCTGAGCACATACCGAAGTCGCAATTAGAAAAAATTGAATCTCCCGAAGCCACGATGCCTTCCTATTGGA
GTTTAAGCTGATAAACAAACTGATAGTACAATTTTTAAAAACCAACTGCAAGGCCATTACCGATGCCGCTCCAAGAGCTTGGGGAGGCGACGTTGGTAAC
GATGAAACGAAGACGTCTGTCTTTGGTGAGGCCGAAGAAGTTACCAACTGGTAGGGATAGATACCACACATACCTCAGGCATAACATAGATAAACCAGTA
CATGTATATCTATATCTATATTTATATATAGACAAACAGCATTAATTAACTATAACAAAGTTTCTAGTAACACTAACGGTAGTTAATTTCTCTTTTTTGT
CCTCGTTGTTGAAAAACGAAAGAAGAATGAAAAAAAAAAAAACAAAAGAGTAATAGCTAGTGTTTTAGAGCTTTTCCACATTCTGACCGCACTTGTAGAC
AGCCACTCTTTGCATTGCCACTCGACATTACATGAACGACTGTTCTTCTCCCTGTCGCCTTAGCTTACTTCTTTGAAAAAAGCAAATCGCCCTTTTATGT
AGGGACAAGTAACTTTTAGATC...

Phase d’inventaire

1. **Alignements.** Aligner sur la séquence
   - des ARN messagers de l’organisme en question
   - des séquences codantes d’autres organismes.

2. **Segmentation** (approche *« ab initio »*) : modèles de Markov cachés, …

Modèle de Markov caché : principes



Trouver la segmentation la plus probable d'une séquence :

Pr(ATTGAC) = 3/10 × 2/10 × 1/10 × 9/10 × 1/10 × …

Pr(ATTGAC) = 3/10 × 8/10 × 3/10 × 2/10 × 1/10 × …

Raffinements : fréquences d'oligonucléotides, phases du codant, caractères syntaxiques (Start, Stop, …)

Phase d'inventaire : problèmes

1. **Alignements**.
   - on ne détecte que des gènes déjà connus par ailleurs, ou des ARN fortement exprimés.
   - problèmes d'ordre technique : contamination par des ARN pré-messagers…
   - Imprécision des algorithmes d'alignement.

2. **Segmentation.** Dans *A. thaliana*, moins d'un gène sur deux est correctement reconnu ; deux gènes prédits sur trois sont faux. [Reese et al. 2000]

On prédit mal, et on ne prédit que ce que l'on connaît déjà.

Paradigme : comparaison biologie/aléatoire

Des différences observées entre séquences biologiques et séquences aléatoires, on peut déduire des faits biologiques.

Exemple : si un motif apparaît avec des fréquences très différentes dans une séquence réelle et dans une séquence aléatoire, alors il a probablement une fonctionnalité biologique.

Paradigm : biological vs. random sequences

Searching for overrepresented motifs

Biological sequence :

```
TTCATTATCTCCATTCGCTGGTGGGCAAGGACTTGAGCTATCGCCCTTTC...
GCATAAAGTTATTCATAAACTGTCAGGGGTTCGGTTGCCGCTGGTGGAAC...
AGGCTGGTGGACGCCTACGTTATTTTGCTGGTGGACTGGAAATCATCTAG...
TCCAACGAAATAGCTGGTGGTCTACACTCATATCGTTATTAACAAACGAA...
AGAAACTAATGGGTGTCACAGCTGGTGGGCTCGTATTTTGTAGGAGGTCA...
```

Random sequence :

```
ATATATATATTTATCTTGCAACTCGGAGAATTCTATTAATATATGAACGA...
ACGTAGATGACAACAATTAGCATGTGGATTTGTAAGGTAAGTTTCTTGTG...
CGTTGGTTGGTCATCGATGCAATGAATGAGTCGTTTAAAATAAGACTCGA...
TTGTCTCTCAAGTTTTTTTTGCATTACCATTCTAAGCTGGTGGATATAGG...
GTTTACAAGTTTTAACCTTTTGTCACTCGTCACCTTATGTGTGGCTTTAA...
```

→ *Chi* motif in *E. coli*.

Extraction de promoteurs

Régions en amont de 10 gènes de S. cerevisiae. [J. van Helden]

>MET1 MET1 upstream sequence, from -702 to -1, size 702
TTTTGACCCA……TCTCTTTCTAGAAATGCCATTATGCACGTGACATTACAAATTGTGGTGAAAAAAGG……TTCAAAAGA
>MET2 MET2 upstream sequence, from -800 to -1, size 800
GGGCACGATT……GACTACTAATAATCACGTGATAT……CCCCACATTGACGTATTTTCACGTGATGCGC……AGCGCCACA
>MET3 MET3 upstream sequence, from -800 to -1, size 800
AAGAGTACAA……AAAAAAGGTCACGTGACCAGAAAAGTCACGTGTAATTTTGTAACTCACCGCATTCT……ATAATTAAC
>MET6 MET6 upstream sequence, from -222 to -1, size 222
GGGAAGCTAGCTAGTTTTCCCAACTGCGAAAGAAAAAAAGGAAAGAAAAAAAAATTCTATATAAGTGA……TTCAATATT
>MET14 MET14 upstream sequence, from -800 to -1, size 800
TATTTTTTTA……AGACCGTGCCACTAATTTCACGTGATCAATATATTTACAAGCCACCTCAAAAAATG……AATTATTTC
>ZWF1 MET19 upstream sequence, from -558 to -1, size 558
GTAAGGTGTAGTTTTGCACCCGTGTACATAAGCGTGAAATCACCACAAACTGTGTGTATCAAGTACAT……TAAATAATA
>MET17 MET25 upstream sequence, from -800 to -1, size 800
TATACTAGAA……GCAAATGGCACGTGAAGCTGTCGATATTGGGGAACTGTGGTGGTTGGCAAATGACT……ATCCATACA
>MET30 MET30 upstream sequence, from -800 to -1, size 800
CCATTGCTGC……GTGTGTGGTACAATGTGTGTGTTTTAATGTAGAAATGAGGTTGTAGCACGTGATCG……GAGAAGGGC
>MUP3 MUP3 upstream sequence, from -61 to -1, size 61
TCTGTTTGTAGTCTAAGTTGCTGAGGGCAACGTAGACGTACAGTGCTCAAAATAAGTAAAA
>SAM1 SAM1 upstream sequence, from -548 to -1, size 548
AATATATATTTCTATTACTAAGTACTCGGATGGGTACCGAAAGTGGCAGATGGGCAGTGTTTACTCAA……CCTACTAGT

La probabilité d'une telle représentation de CACGTG dans des séquences aléatoires serait environ égale à 10^{-9}

Paradigm : biological vs. random sequences

Assessing significance of alignment scores

```
HBA_HUMAN  GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKL
HBB_HUMAN  GNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKL
```

Score
130

```
HBA_HUMAN  GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKL
RandomB1   QVGAKDLNALDGKVAHDMPAAVALGSAAHVDLSTNSKHHKL
RandomB2   VAHSDLDAVKGDMPNGSAKKVAAQAAHGLSLTNHAHKLLVD
…
RandomBK   HVDDMTNAGKKVPNAGSAQADAVADLHAHKLLVKGHLSALS


HBB_HUMAN  GNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKL
RandomA1   HLSEKVLGTNLKGTGKFSDGCDKLLKAHNPKVLAGAFALHD
RandomA2   KATEFATKVDGAFSDLSLLAHGKKVGGHLGNLPNLKHCDKL
…
RandomAK   GTKKHGFSELPKVAHGNLDNDGHCGLAFSADKLVLATLKLK
```

Average
score
25

Z-value and p-value

Alignments

$$Z(A) = \frac{E(X) - Score(A)}{\sqrt{V(X)}}$$

$$Pval(A) = \Pr(X \geq Score(A))$$

X = random variable, score of an alignment with a random sequence.

Motifs

$$Z(M) = \frac{E(X) - \#occ(M)}{\sqrt{V(X)}}$$

$$Pval(M) = \Pr(X \geq \#occ(M))$$

X = random variable, number of occurrences of M in a random sequence.

Choix du modèle de séquences aléatoires

Séquence biologique :

```
TTAATTATATAAATTAGCTGGTGGCAAACCAATTCACATATACAAATTTA...
CAATAAACTTATTAATAAAATCTAACCCCTTACCTTCAAGCTGGTGGAAA...
ACGCTGGTGGAACAATAACTTATTTTGCTGGTGGAATCCAAATAATATAC...
TAAAAACAAATAGCTGGTGGTCTACAATAATATACTTATTAAAAAAACAA...
ACAAAATAATCCCTTTAAAAGCTGGTGGCATACTATTTTCTACCACCTAA...
```

Etonnant !

Séquence biologique :

```
TTCGTTGTCTCCGTTCGCTGGTGGGCGGGGGCTTGGGCTGTCGCCCTTTC...
GCGTGGGGTTGTTCGTGGGCTGTCGGGGGTTCGGTTGCCGCTGGTGGGGC...
GGGCTGGTGGGCGCCTGCGTTGTTTTGCTGGTGGGCTGGGGGTCGTCTGG...
TCCGGCGGGGTGGCTGGTGGTCTGCGCTCGTGTCGTTGTTGGCGGGCGGG...
GGGGGCTGGTGGGTGTCGCGGCTGGTGGGCTCGTGTTTTGTGGGGGGTCG...
```

Moins étonnant !

Modèles classiques de séquences aléatoires [Fitch 83]

Séquence biologique :

AACGACGTGCCGTGCGCTCGACGT

Occurrences :

AACG : 1

Modèles classiques de séquences aléatoires [Fitch 83]

Séquence biologique :

AAACGACGTGCCGTGCGCTCGACGT

Occurrences :

AACG : 1
ACGA : 1

Modèles classiques de séquences aléatoires [Fitch 83]

Séquence biologique :

AACGACGTGCCGTGCGCTCGACGT

Occurrences :

AACG : 1
ACGA : 1
CGAC : 1

Modèles classiques de séquences aléatoires [Fitch 83]

Séquence biologique :

AACGACGTGCCGTGCGCTCGACGT

Occurrences :

AACG : 1
ACGA : 1
CGAC : 1
GACG : 1

Modèles classiques de séquences aléatoires [Fitch 83]

Séquence biologique :

AACGACGTGCCGTGCGCTCGACGT

Occurrences :

AACG : 1	CGTG : 2	CGCT : 1
ACGA : 1	GTGC : 2	GCTC : 1
CGAC : 2	TGCG : 2	CTCG : 1
GACG : 2	GCGT : 1	TCGA : 1
ACGT : 2	GCGC : 1	

Modèle exact (shuffling)

Séquences ayant **exactement** les mêmes nombres d'occurrences de nucléotides que la séquence de référence.

Occurrences :

AACG : 1
ACGA : 1
CGAC : 2
GACG : 2
ACGT : 2

CGTG : 2
GTGC : 2
TGCG : 2
GCGT : 1
GCGC : 1

CGCT : 1
GCTC : 1
CTCG : 1
TCGA : 1

Modèle markovien

Séquences ayant **en moyenne** les mêmes nombres d'occurrences de nucléotides que la séquence de référence.

$Pr(G|AAC) = 1$ $Pr(T|GCG) = 1/2$

Occurrences :

AACG : 1	CGTG : 2	CGCT : 1
ACGA : 1	GTGC : 2	GCTC : 1
CGAC : 2	TGCG : 2	CTCG : 1
GACG : 2	GCGT : 1	TCGA : 1
ACGT : 2	GCGC : 1	

Sur- (ou sous-) représentation de motifs

Données :

- un modèle de séquence,
- n : longueur de la séquence,
- un motif H.

Problème : établir la loi du motif dans les séquences aléatoires de longueur n

- Probabilité que H apparaisse r fois ? Ou au moins r fois ?
- Espérance ? Ecart-type ?

Deux types d'approches :

- Approches analytiques
- Approche expérimentale

Approches analytiques

Méthodes probabilistes

([Prum, Rodolphe, de Turkheim 95], [Schbath 97], [Apostolico, Bock, Xuyan 98], [Reinert, Schbath, Waterman 00]…)

Méthodes combinatoires : calcul de séries génératrices de probabilités.

- Approche « généraliste »

  ([Nicodème, Salvy, Flajolet 99], [Nicodème 01]…)

- Approche « spécialisée »

  ([Régnier, Szpankowski 98], [Robin, Daudin 99], [Régnier 00], …)

+ Grandes déviations

([Régnier, Szpankowski 98], [Reinert, Schbath, Waterman 00], …)

Série génératrice d'un langage

Mots qui contiennent au moins une fois le motif TT.

TT	ATT	AATT	AAATT	AAAATT	
	TTA	ATTA	AATTA	AAATTA	
	TTT	ATTT	AATTT	AAATTT	
		TATT	ATATT	AATATT	
		TTAA	ATTAA	AATTAA	
		...	...	...	
1	3	8	19	43	...

Série génératrice d'un langage

Mots qui contiennent au moins une fois le motif TT.

TT	ATT	AATT	AAATT	AAAATT
	TTA	ATTA	AATTA	AAATTA
	TTT	ATTT	AATTT	AAATTT
		TATT	ATATT	AATATT
		TTAA	ATTAA	AATTAA
		...	...	...

$$1z^2 + 3z^3 + 8z^4 + 19z^5 + 43z^6 + \cdots$$

$$= \sum_{n \geq 0} a_n z^n \qquad (a_n = \text{nbe de mots de longueur } n)$$

Fonction génératrice d'un langage

$\sum_{n\geq 0} a_n z^n = z^2 + 3z^3 + 8z^4 + 19z^5 + 43z^6 + \cdots$ est le développement de Taylor au voisinage de $z = 0$ de la fonction

$$f(z) = \frac{z^2}{(1-2z)(1-z-z^2)}.$$

Calcul de a_n :

fonction génératrice rationnelle $\rightarrow$
- Formule de récurrence
- Formule close
- Formule asymptotique : $a_n \sim \mathrm{P}(n)\,\alpha^n$

Occurrences d'un motif

Combien existe-t-il de mots de longueur n qui contiennent k fois le motif TT ?

$$f(z,u) = \sum_{n,k\geq 0} b_{n,k} z^n u^k$$

.	A	AA	TT	AAA	ATT	TTT
	T	AT		AAT	TTA	
		TA		ATA		
				TAA		
				TAT		

$$1 + 2z + (3 + u)z^2 + (5 + 2u + u^2)z^3 + \cdots$$

Version probabiliste

Quelle est la probabilité qu'une séquence aléatoire (p(A)=1/3, p(T)=2/3) de longueur n contienne k fois le motif TT ?

$$f(z,u) = \sum_{n,k\geq 0} p_{n,k} z^n u^k$$

.	A	AA	TT	AAA	ATT	TTT
	T	AT		AAT	TTA	
		TA		ATA		
				TAA		
				TAT		

$$1 + z + (\tfrac{5}{9} + \tfrac{4}{9}u)z^2 + (\tfrac{11}{27} + \tfrac{8}{27}u + \tfrac{8}{27}u^2)z^3 + \cdots$$

Grandeurs d'intérêt (pour un motif)

Fonction génératrice :

$$L(z,u) = \sum_{n,k \geq 0} Pr(X_n = k) z^n u^k$$

où X_n est la variable aléatoire qui compte le nombre d'occurrences de H dans une séquence de longueur n.

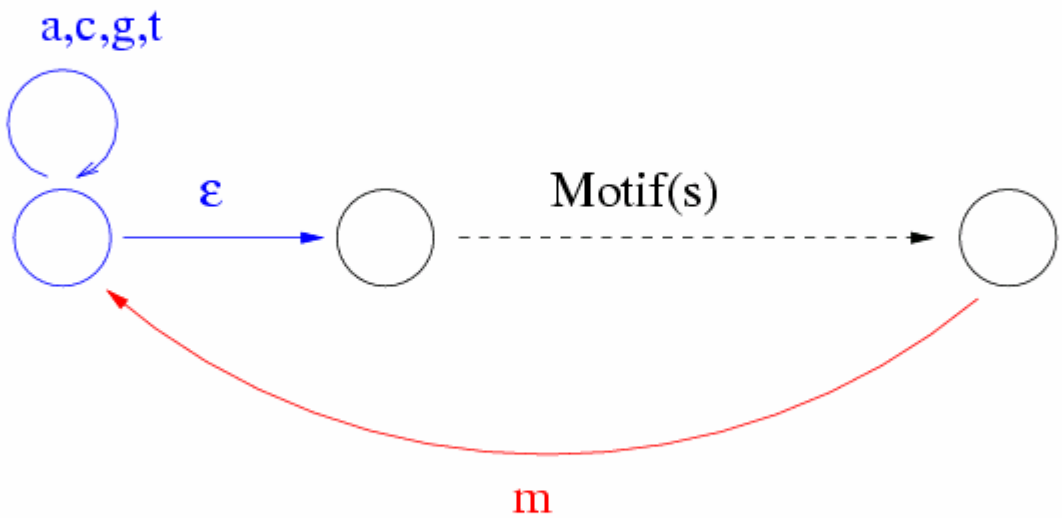
$$E(X_n) = \sum_{k \geq 0} k p_{n,k} = [z^n] \frac{\partial L}{\partial u}(z,1)$$

$$V(X_n) = [z^n] \left(\frac{\partial^2 L}{\partial u^2}(z,1) + \frac{\partial L}{\partial u}(z,1) - (\frac{\partial L}{\partial u}(z,1))^2 \right)$$

$$Pr(X_n \geq k) = 1 - [z^n u^{k-1}] \frac{1}{1-u} L(z,u).$$

Approche “généraliste” [Nicodème, Salvy, Flajolet]

Automate

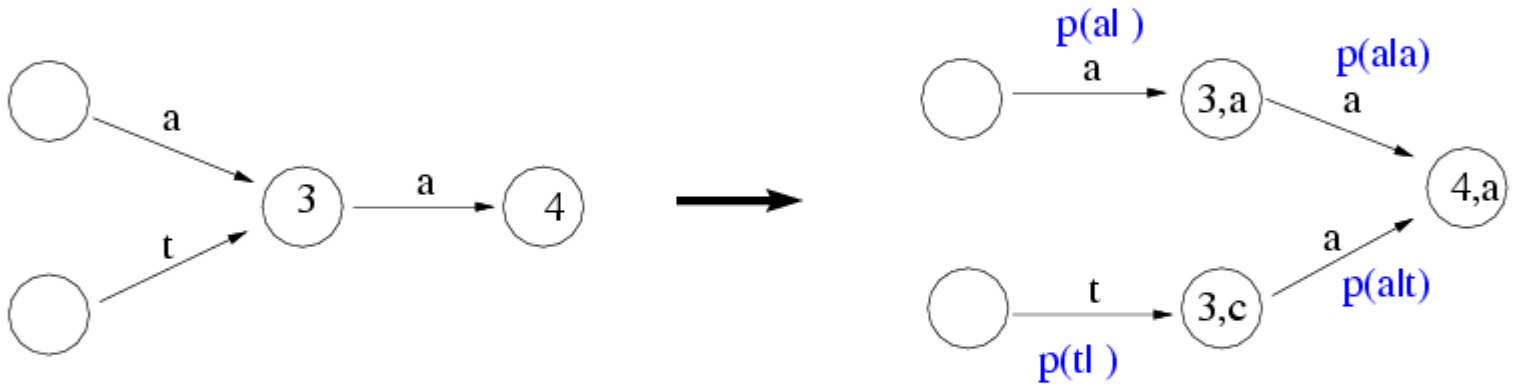


à déterminiser.

Approche “généraliste”

Automate × Chaîne de Markov = automate probabiliste

Ordre 1 :



→ Série génératrice de probabilité.

\+ : généralisable à tout ensemble de motifs décrit par une expression rationnelle.

\- : Pas de formules “closes”.

Approche "spécialisée" [Régnier, Szpankovski]

Décomposition du langage des séquences

$\mathcal{L} = \{$ mots qui contiennent au moins une fois le motif H $\}$

$\mathcal{R} = \{$ mots qui ont H pour suffixe et ne contiennent pas d'autre occurrence de H $\}$

$\mathcal{M} = \{$ mots w tels que Hw a H pour suffixe et ne contient pas d'autre occurrence de H $\}$

$\mathcal{U}_i = \{$ mots w tels qu'il y a une seule occurrence de H dans Hw $\}$

$$\underbrace{AATT}_{\in \mathcal{R}}\underbrace{ATT}_{\in \mathcal{M}}\underbrace{T}_{\in \mathcal{M}}\underbrace{ATATATT}_{\in \mathcal{M}}\underbrace{T}_{\in \mathcal{M}}\underbrace{ATAA}_{\in \mathcal{U}} \qquad \in \mathcal{L}$$

Fonction génératrice du langage des séquences

$$\underbrace{AATT}_{\in \mathcal{R}}\underbrace{ATT}_{\in \mathcal{M}}\underbrace{T}_{\in \mathcal{M}}\underbrace{ATATATT}_{\in \mathcal{M}}\underbrace{T}_{\in \mathcal{M}}\underbrace{ATAA}_{\in \mathcal{U}} \qquad \in \mathcal{L}$$

$$\begin{aligned} \mathcal{L} &= \mathcal{R} \,.\, (\varepsilon \cup \mathcal{M} \cup \mathcal{M}.\mathcal{M} \cup \mathcal{M}.\mathcal{M}.\mathcal{M} \cup \ldots) \,.\, \mathcal{U} \\ &(= \mathcal{R} \,.\, \mathcal{M}^* \,.\, \mathcal{U}) \end{aligned}$$

$$\begin{aligned} L(z,u) &= uR(z) \times (1 + uM(z) + (uM(z))^2 + (uM(z))^3 + \ldots) \times U(z) \\ &= uR(z) \times \frac{1}{1 - uM(z)} \times U(z) \end{aligned}$$

Equation pour la fonction génératrice de $\mathcal{M}$

$$\underbrace{ATT}_{\in\mathcal{M}}\ \underbrace{T}_{\in\mathcal{M}}\ \underbrace{ATATATT}_{\in\mathcal{M}}\ \underbrace{T}_{\in\mathcal{M}} \qquad \in \mathcal{M}^*$$

$$\mathcal{M}^* = \{\underbrace{1,\ T}_{\mathcal{A}},\ \underbrace{TT,\ ATT,\ TTT,\ AATT,\ \ldots}_{\mathcal{X}^*H}\}$$

$\mathcal{A}$ est l'ensemble d'autocorrélation du motif H.

$$\mathcal{M}^* = \mathcal{A} \cup \mathcal{X}^*H$$

$$\Rightarrow \quad \frac{1}{1-M(z)} \quad = \quad A(z) + \frac{1}{1-z} p(H)\, z^{\ell(H)}$$

(ici, $A(z) = 1 + zp(T)$.)

Expressions des fonctions génératrices

$$\underbrace{AATT}_{\in \mathcal{R}}\underbrace{ATT}_{\in \mathcal{M}}\underbrace{T}_{\in \mathcal{M}}\underbrace{ATATATT}_{\in \mathcal{M}}\underbrace{T}_{\in \mathcal{M}}\underbrace{ATAA}_{\in \mathcal{U}} \qquad \in \mathcal{L}$$

$$L(z,u) \quad = \quad \frac{uR(z)U(z)}{1-uM(z)}$$

où

$$M(z) = 1 + \frac{(z-1)}{D(z)} \quad , R(z) = \frac{p(H)z^{\ell(H)}}{D(z)} \quad , U(z) = \frac{1}{D(z)}$$

et

$$D(z) \quad = \quad (1-z)A(z) + p(H))z^{\ell(H)}.$$

Généralisation à un ensemble de motifs H

$\mathcal{L}_k = \{$ mots qui contiennent k occurrences valides de motifs de $\mathcal{H}$ $\}$

$\mathcal{R}_i = \{$ mots qui ont H_i pour suffixe et ne contiennent aucun autre motif de $\mathcal{H}$ $\}$

$\mathcal{M}_{i,j} = \{$ mots w tels que $H_i w$ admet H_j pour suffixe, et que H_i et H_j sont les seules occurrences valides dans $H_i w$ $\}$

$\mathcal{U}_i = \{$ mots w t.q. l'unique occurrence valide dans $H_i w$ soit H_i $\}$

$$\mathcal{H} = \{AAT, GATC, CTG\}$$

$$\underbrace{GTCAGTCAAT}_{\mathcal{R}_1}\underbrace{TGTGATC}_{\mathcal{M}_{1,2}}\underbrace{GTGTTTTTTAAT}_{\mathcal{M}_{2,1}}\underbrace{GCTG}_{\mathcal{M}_{1,3}}\underbrace{ATATATA}_{\mathcal{U}_3}$$

Décomposition du langage des séquences

Equations pour les langages $\mathcal{R}_i, \mathcal{M}_{i,j}, \mathcal{U}_i$ + Chaîne de Markov

$$\Downarrow$$

Formule générale pour la série génératrice de probabilités

$$L(z,u) = \sum p_{n,k} z^n u^k$$

$$\Downarrow$$

Formules pour l'évaluation de la surreprésentativité

Un résultat de grandes déviations

[Régnier, Szpankowsky 97]+[Régnier, AD 03

$$Pr(X_n \geq k) \approx \frac{1}{2\sigma_a\sqrt{n}} e^{-nI(a)}$$

où $a = k/n$,

$$\begin{aligned} I(a) &= a \ln \left(\frac{D_1(z_a)}{D_1(z_a) + z_a - 1} \right) + \ln z_a \ , \\ \sigma_a^2 &= a(a-1) - a^2 z_a \left(\frac{2D_1'(z_a)}{D_1(z_a)} - \frac{(1-z_a)D_1''(z_a)}{D_1(z_a) + (1-z_a)D_1'(z_a)} \right), \\ D_1(z) &= (1-z)A_1(z) + P(H_1)z^{|H_1|}, \end{aligned}$$

et z_a est la plus grande racine positive de l'équation

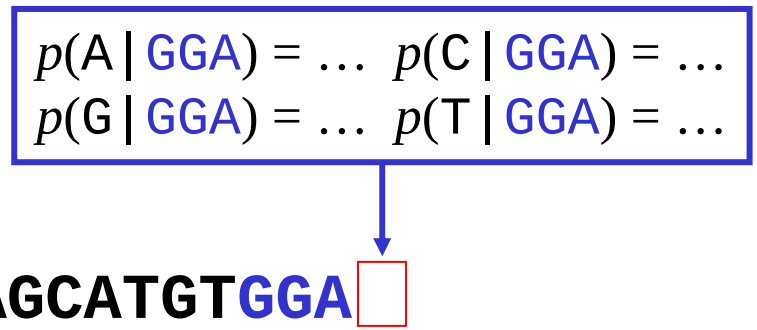
$$D_1(z)^2 - (1 + (a-1)z)D_1(z) - az(1-z)D_1'(z) = 0.$$

Approche expérimentale :

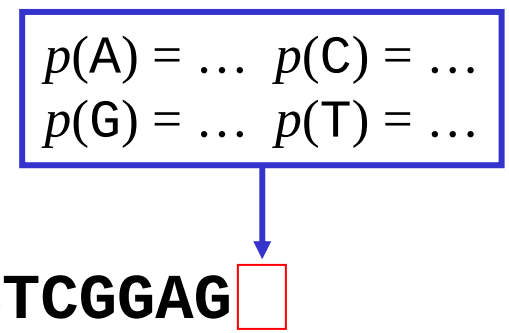
génération aléatoire de séquences génomiques

Modèle markovien

- **Chaîne de Markov :**



- **Loi de Bernoulli :**



Génération en fréquences moyennes (markovienne)

Pr(G|AAC) = 1
Pr(A|ACG) = 1/3
Pr(C|CGA) = 1
Pr(G|GAC) = 1
Pr(T|ACG) = 2/3

Pr(G|CGT) = 1
…

Occurrences :

AACG : 1
ACGA : 1
CGAC : 2
GACG : 2
ACGT : 2

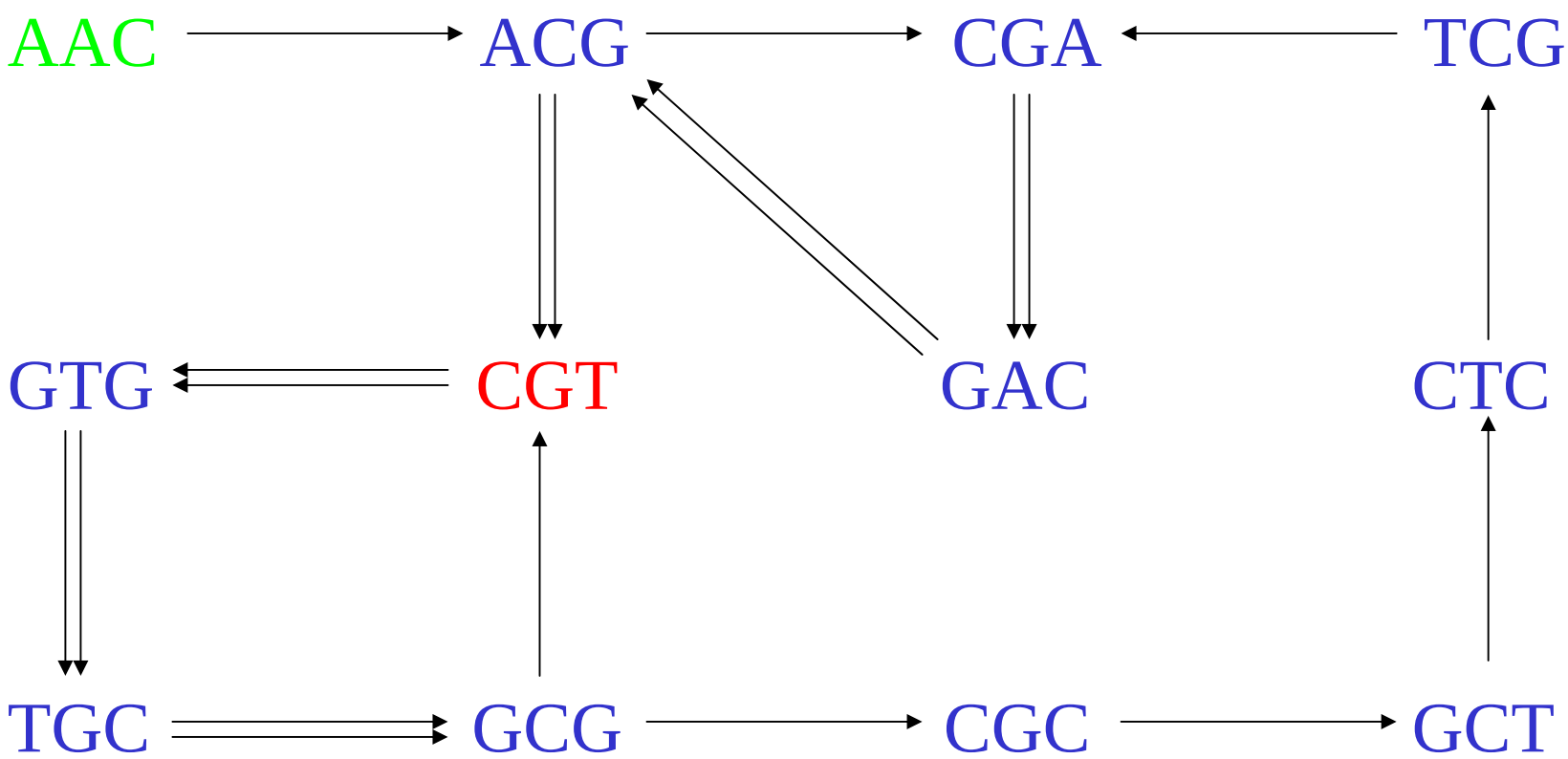
CGTG : 2
GTGC : 2
TGCG : 2
GCGT : 1
GCGC : 1

CGCT : 1
GCTC : 1
CTCG : 1
TCGA : 1

Génération en fréquences exactes

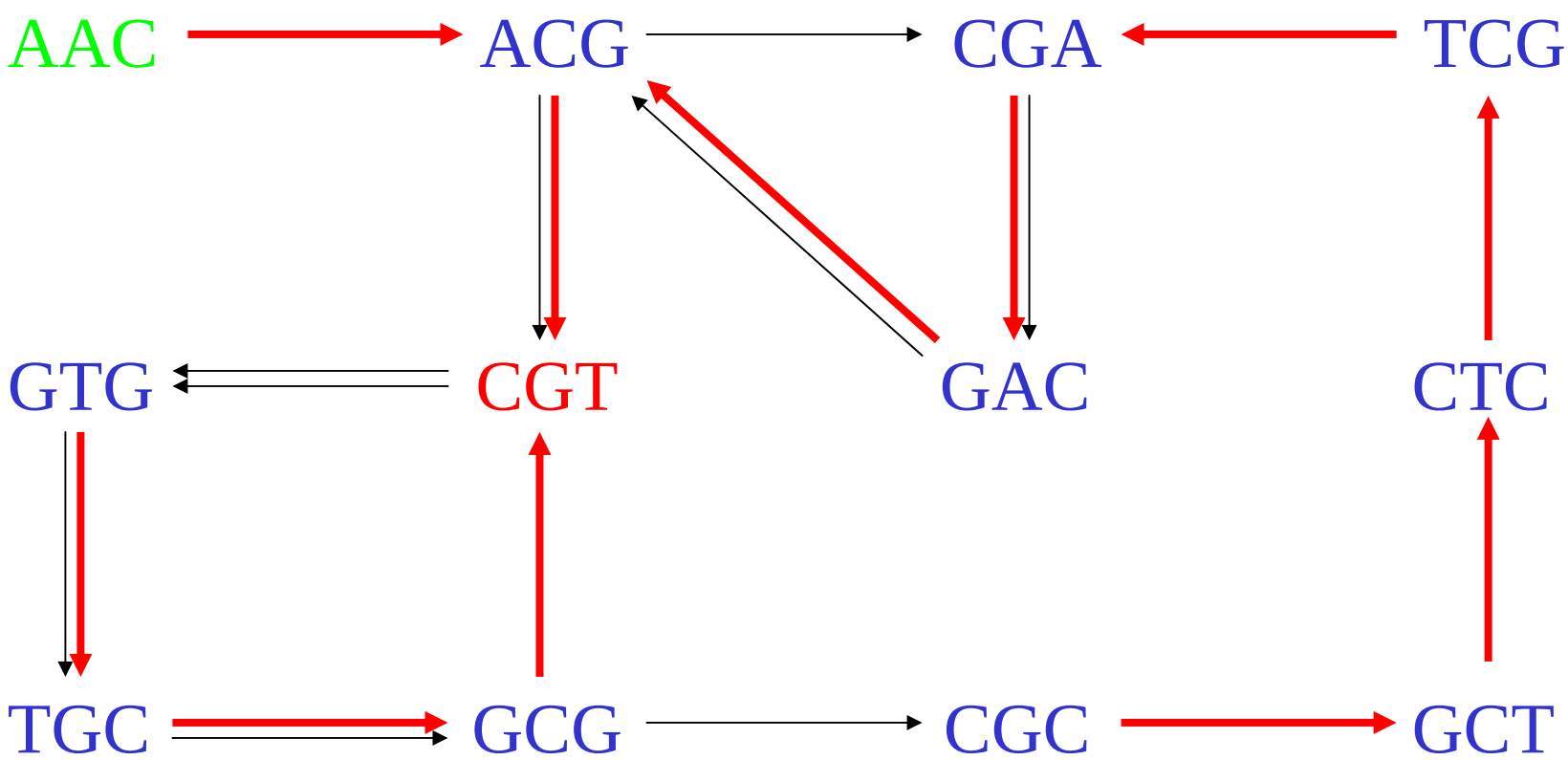
[Kandel, Matias, Unger, Winkler 96]

Chemin eulérien dans le graphe suivant :



AACG : 1	CGTG : 2	CGCT : 1
ACGA : 1	GTGC : 2	GCTC : 1
CGAC : 2	TGCG : 2	CTCG : 1
GACG : 2	GCGT : 1	TCGA : 1
ACGT : 2	GCGC : 1	

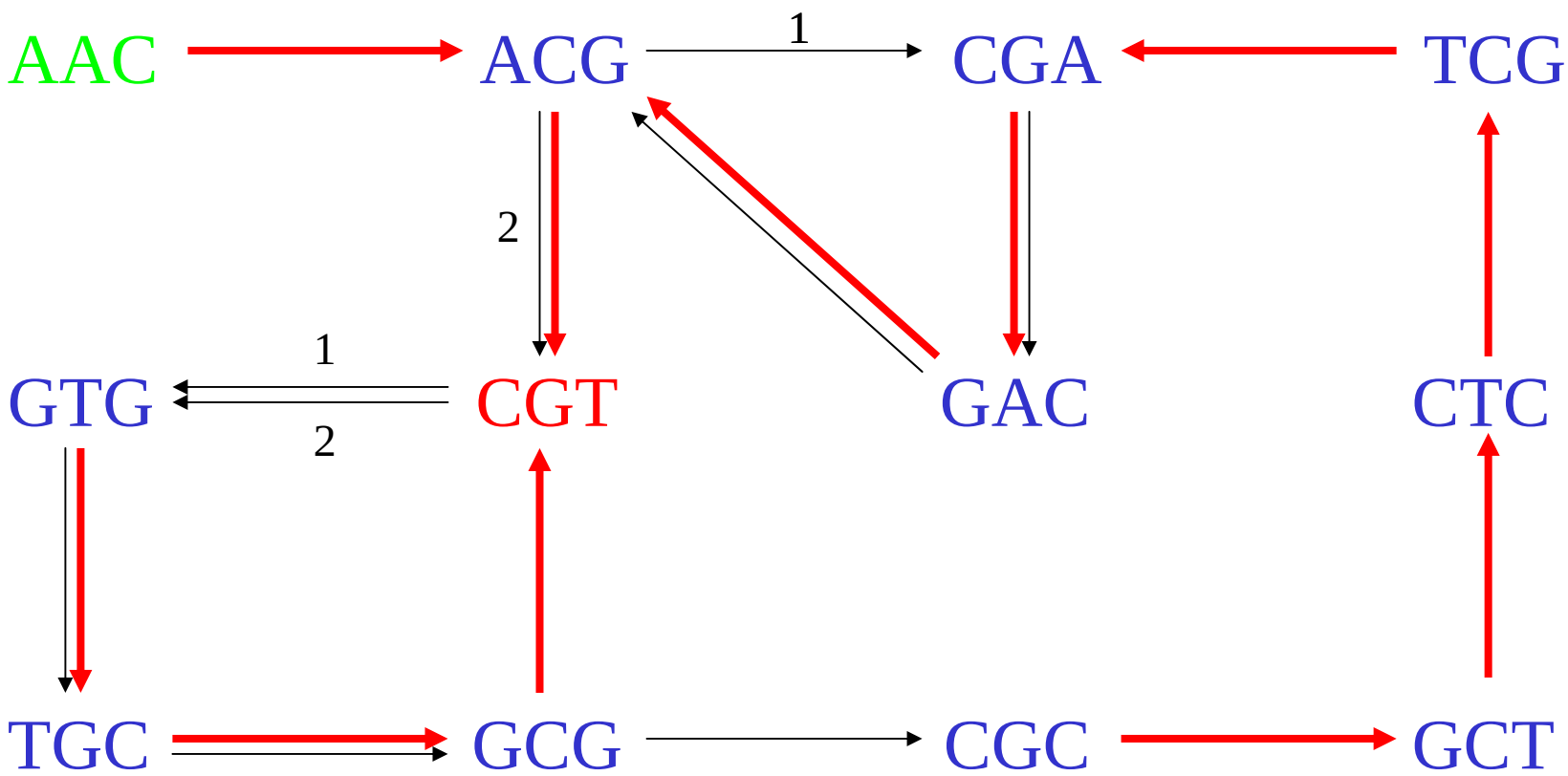
Génération en fréquences exactes



Chemin eulérien = arbre couvrant × ordre des arcs adjacents à un même sommet

[Aardenne-Ehrenfest, de Bruijn 51]

Génération en fréquences exactes

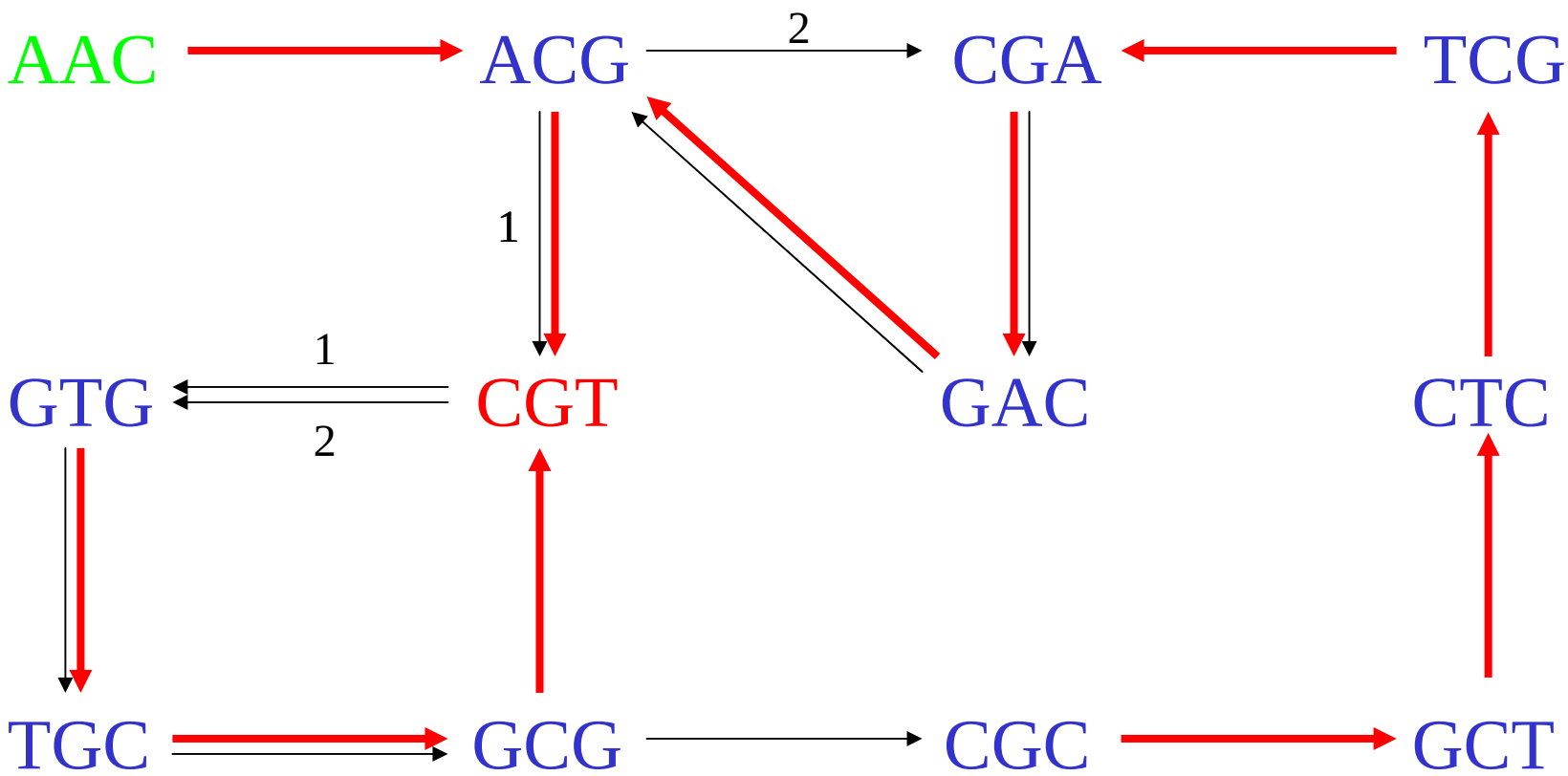


AACGACGTGCGCTCGACGTGCGT

Chemin eulérien = arbre couvrant × ordre des arcs adjacents à un même sommet

[Aardenne-Ehrenfest, de Bruijn 51]

Génération en fréquences exactes

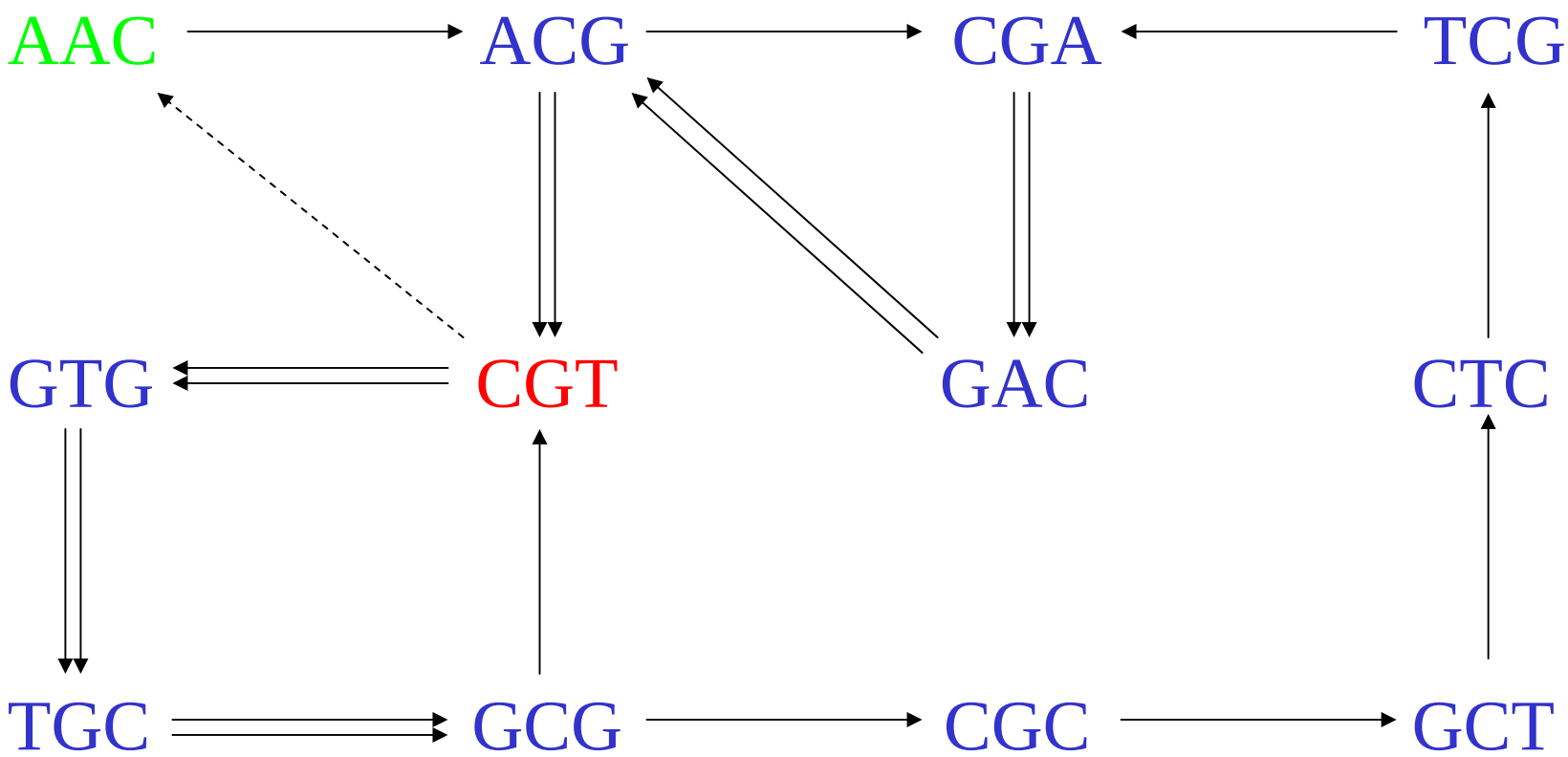


AACGTGCGCTCGACGACGTGCGT

Chemin eulérien = arbre couvrant × ordre des arcs adjacents à un même sommet

[Aardenne-Ehrenfest, de Bruijn 51]

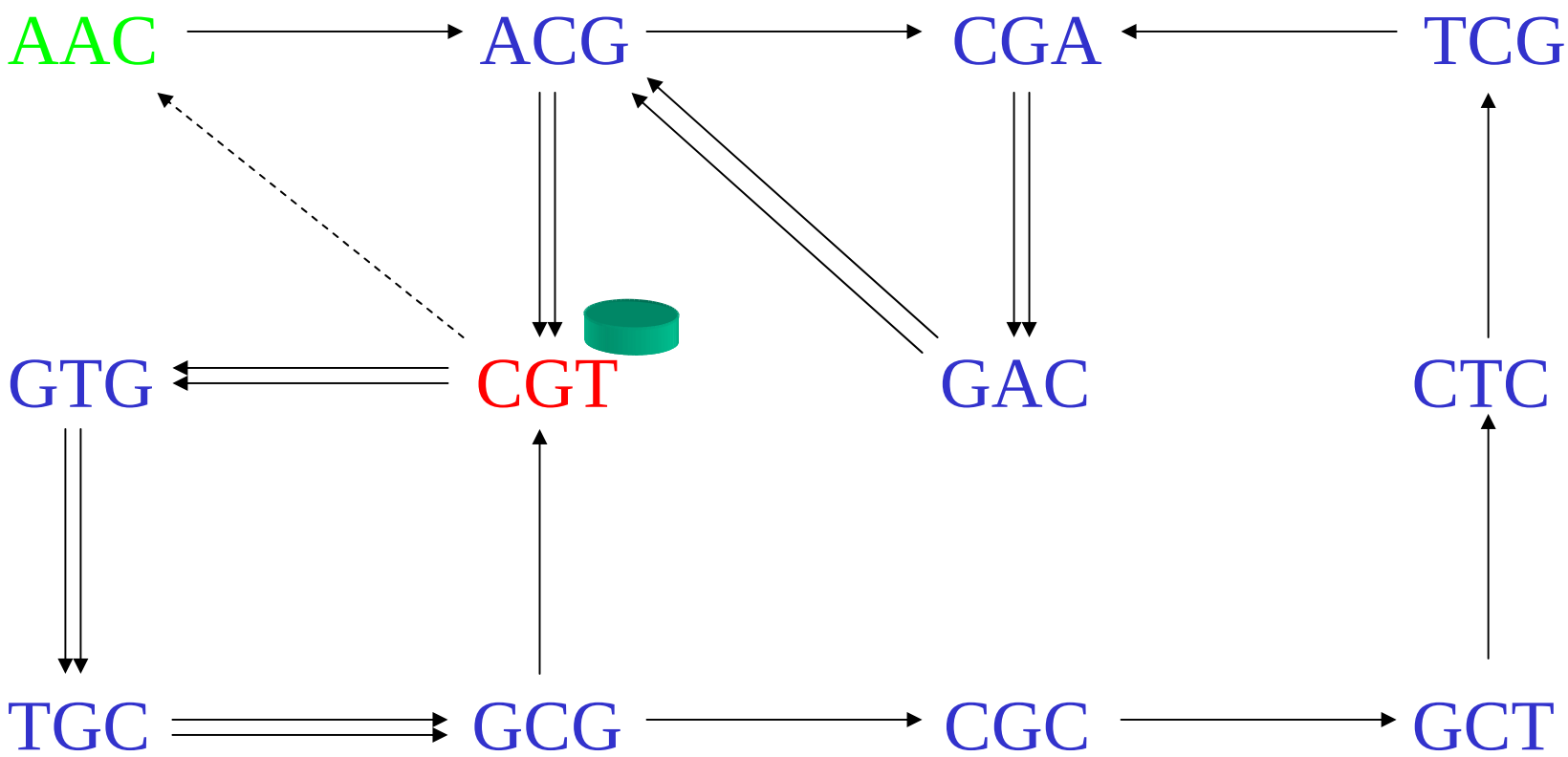
Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder]
[Wilson]

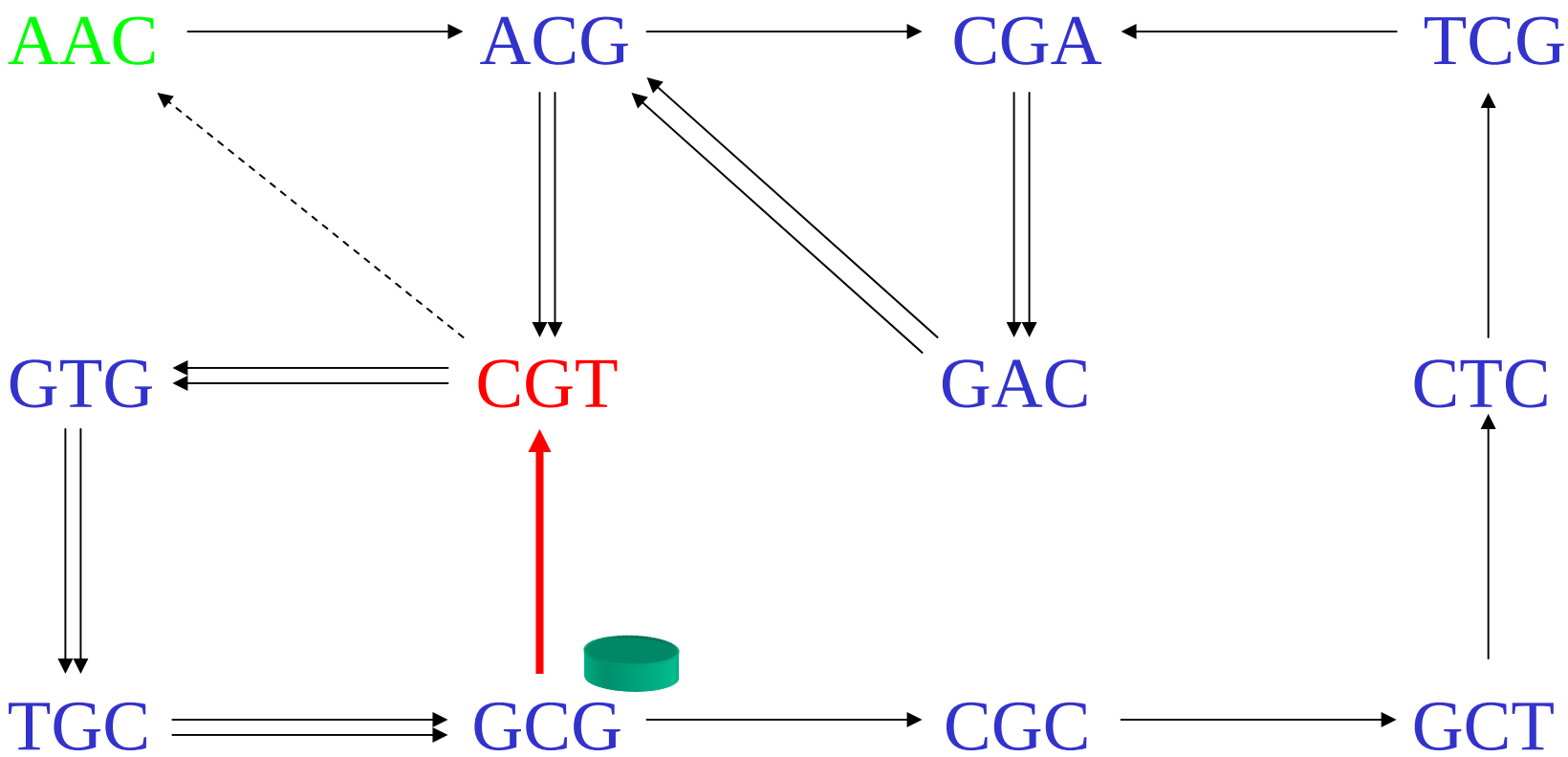
Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder]
[Wilson]

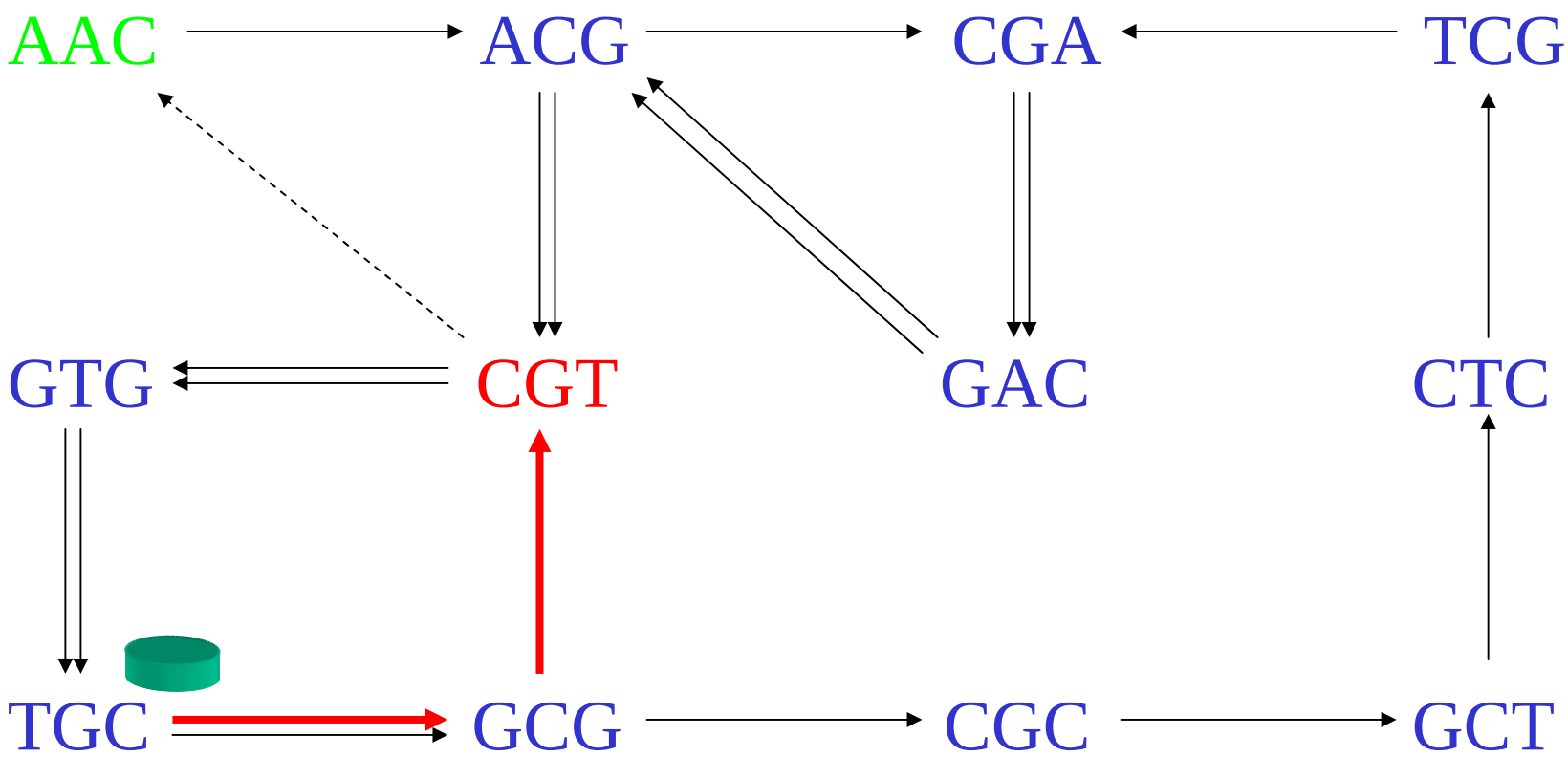
Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder]
[Wilson]

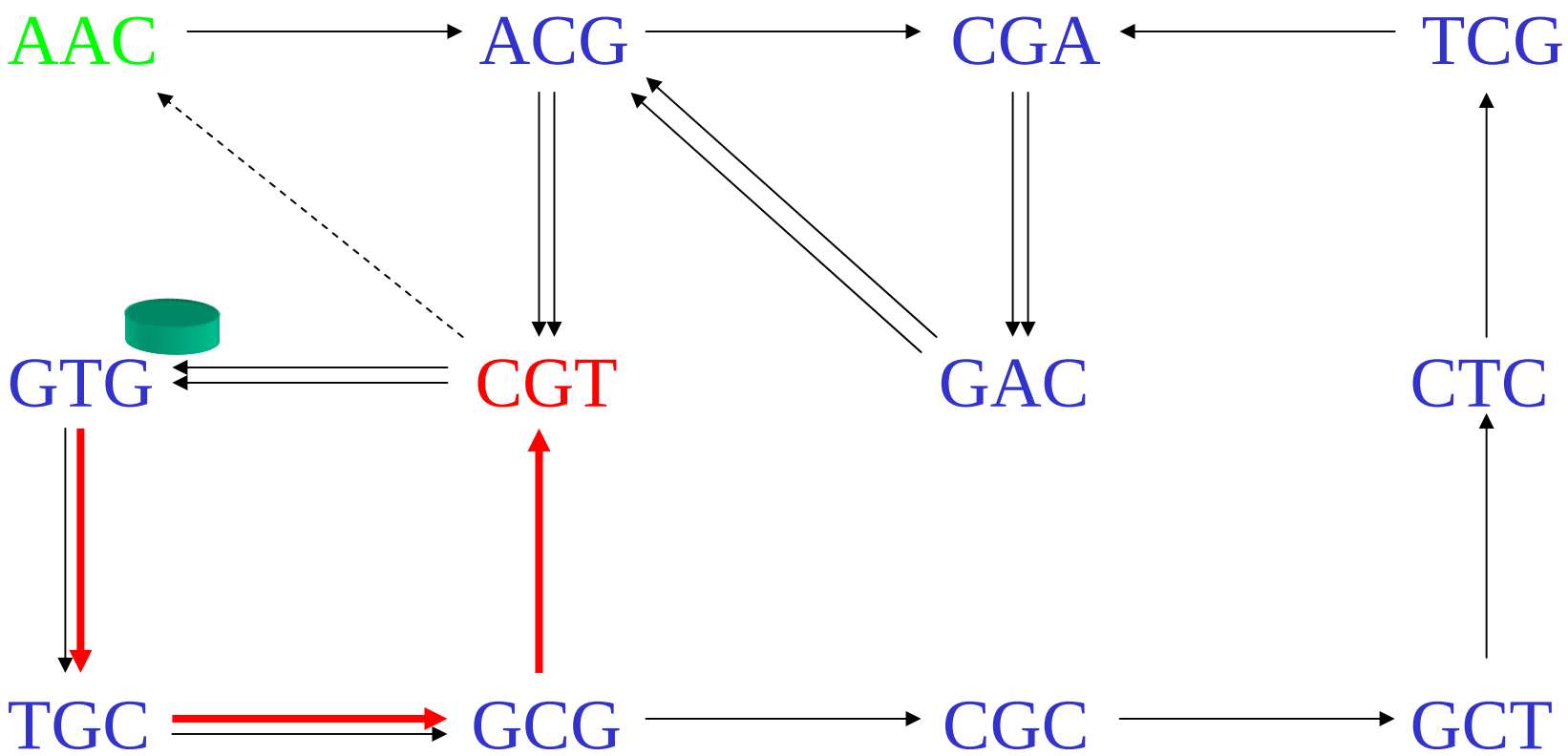
Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder]
[Wilson]

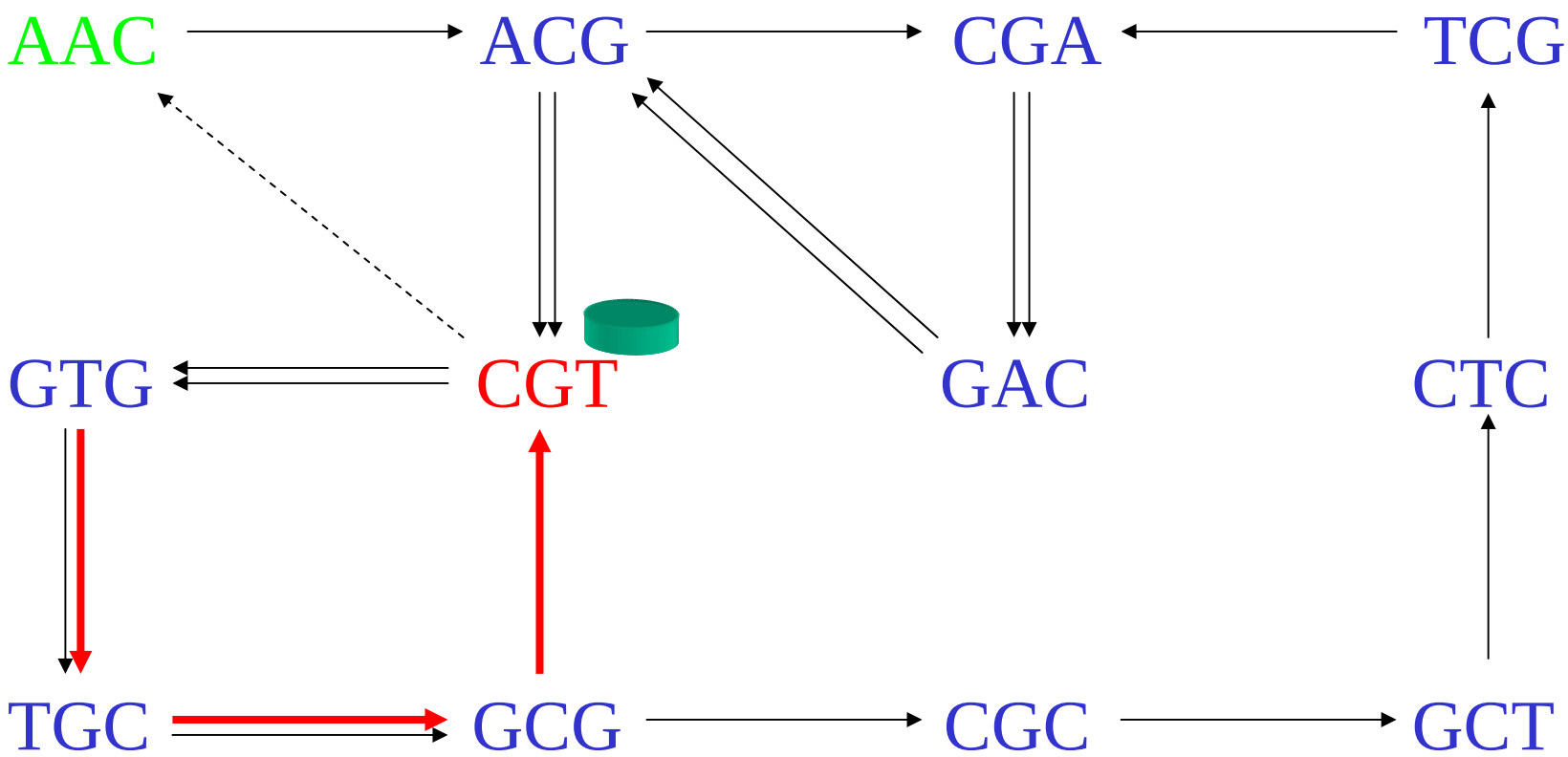
Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder 90]
[Wilson 97]

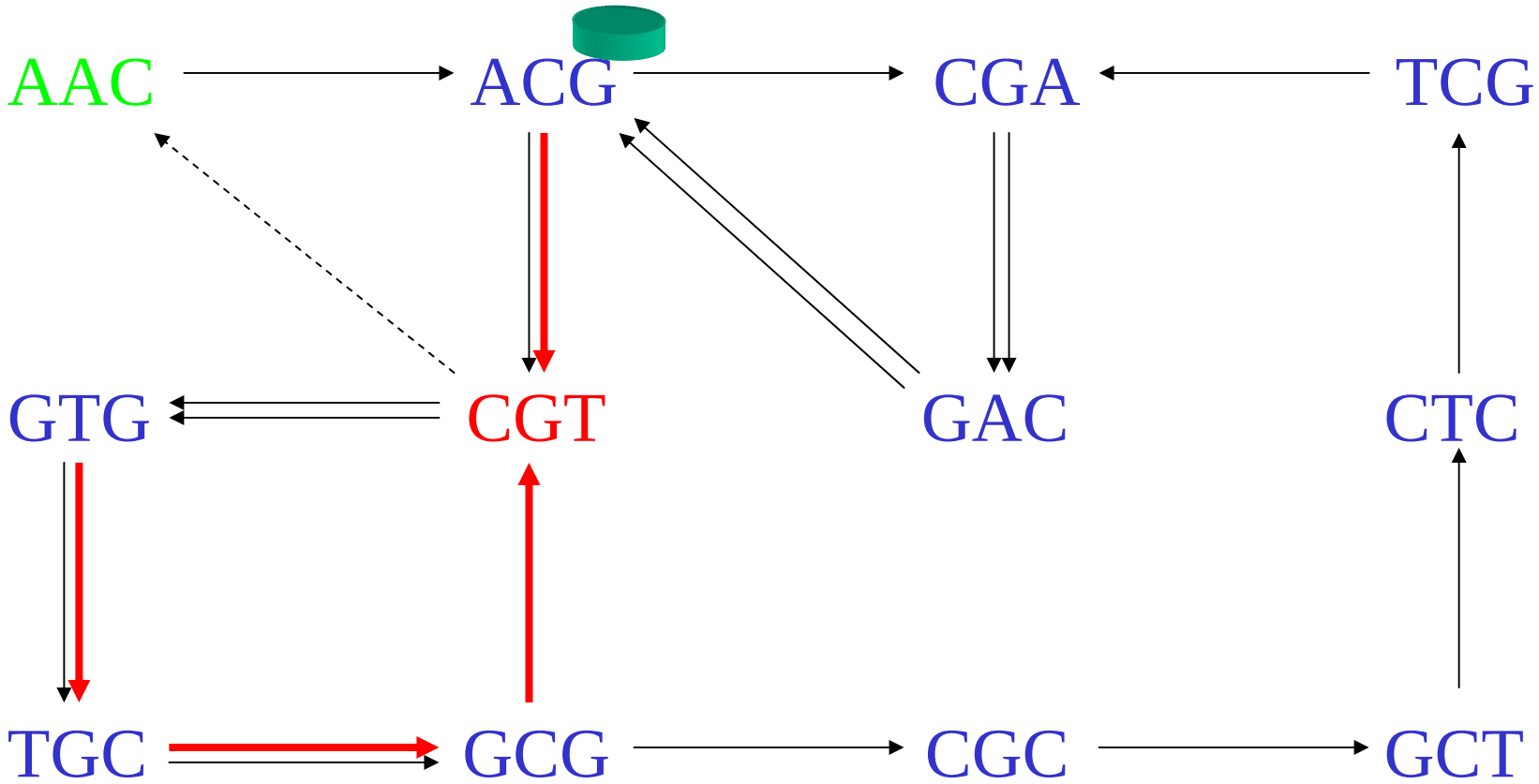
Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder 90]
[Wilson 97]

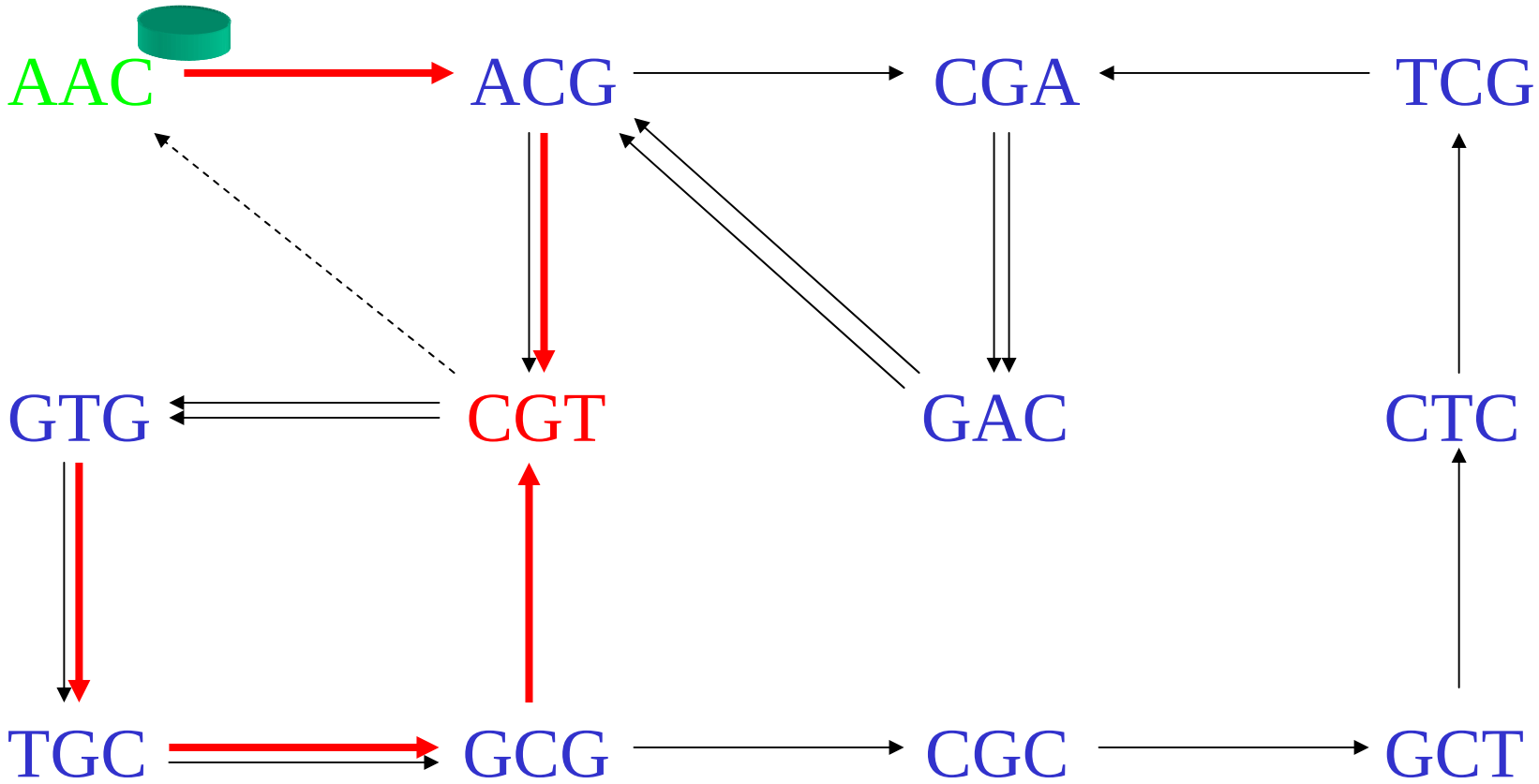
Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder 90]
[Wilson 97]

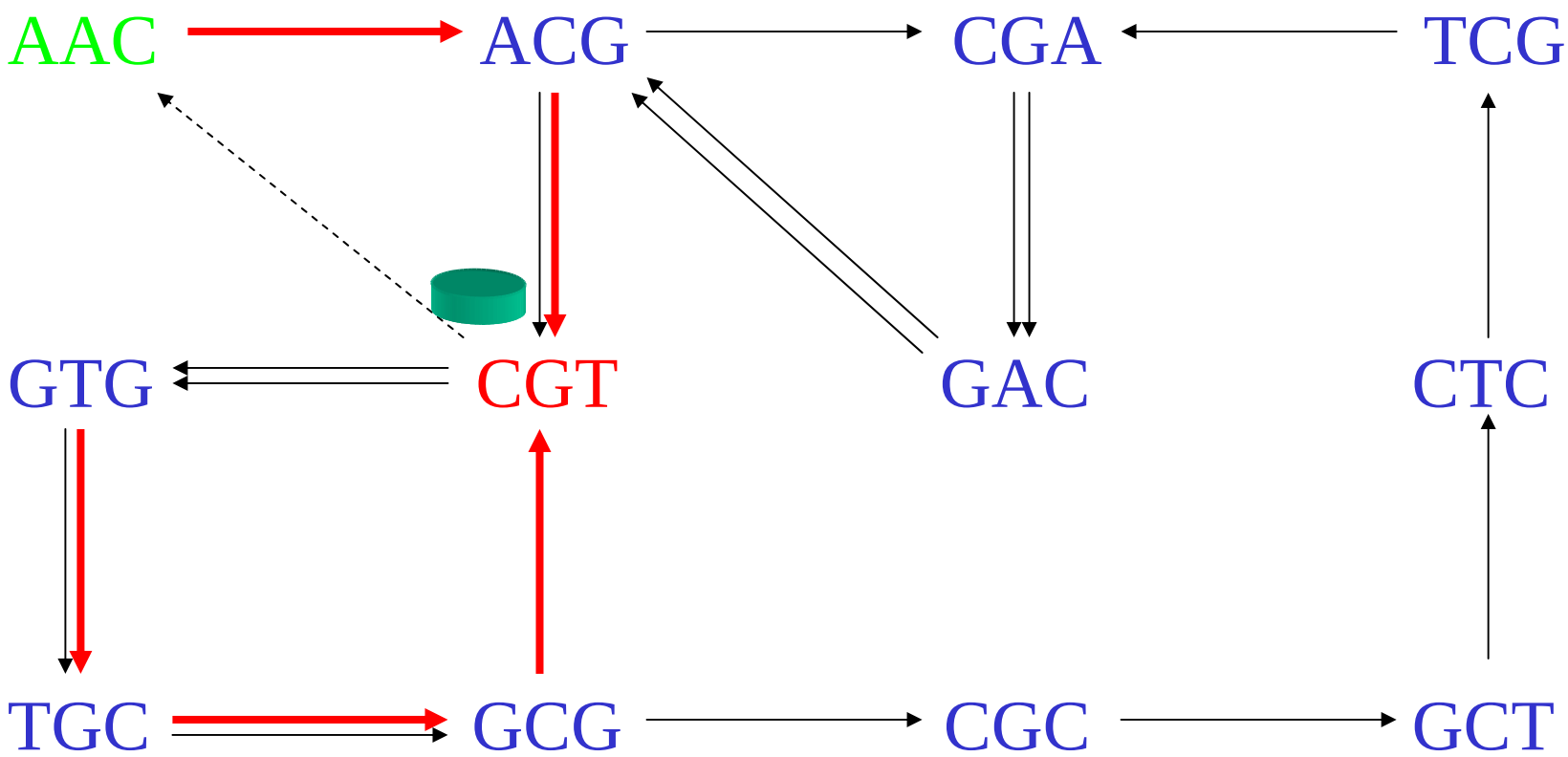
Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder 90]
[Wilson 97]

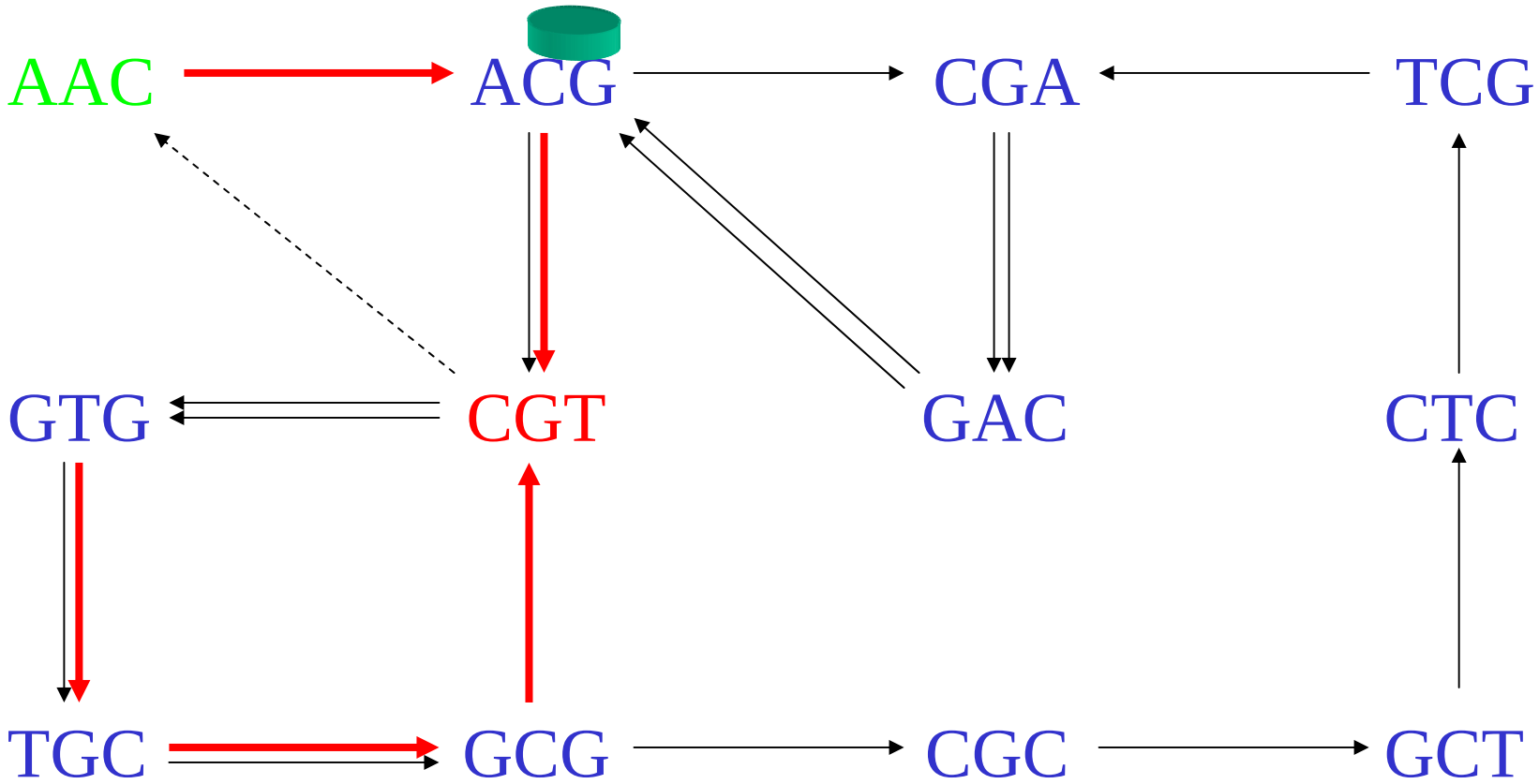
Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder 90]
[Wilson 97]

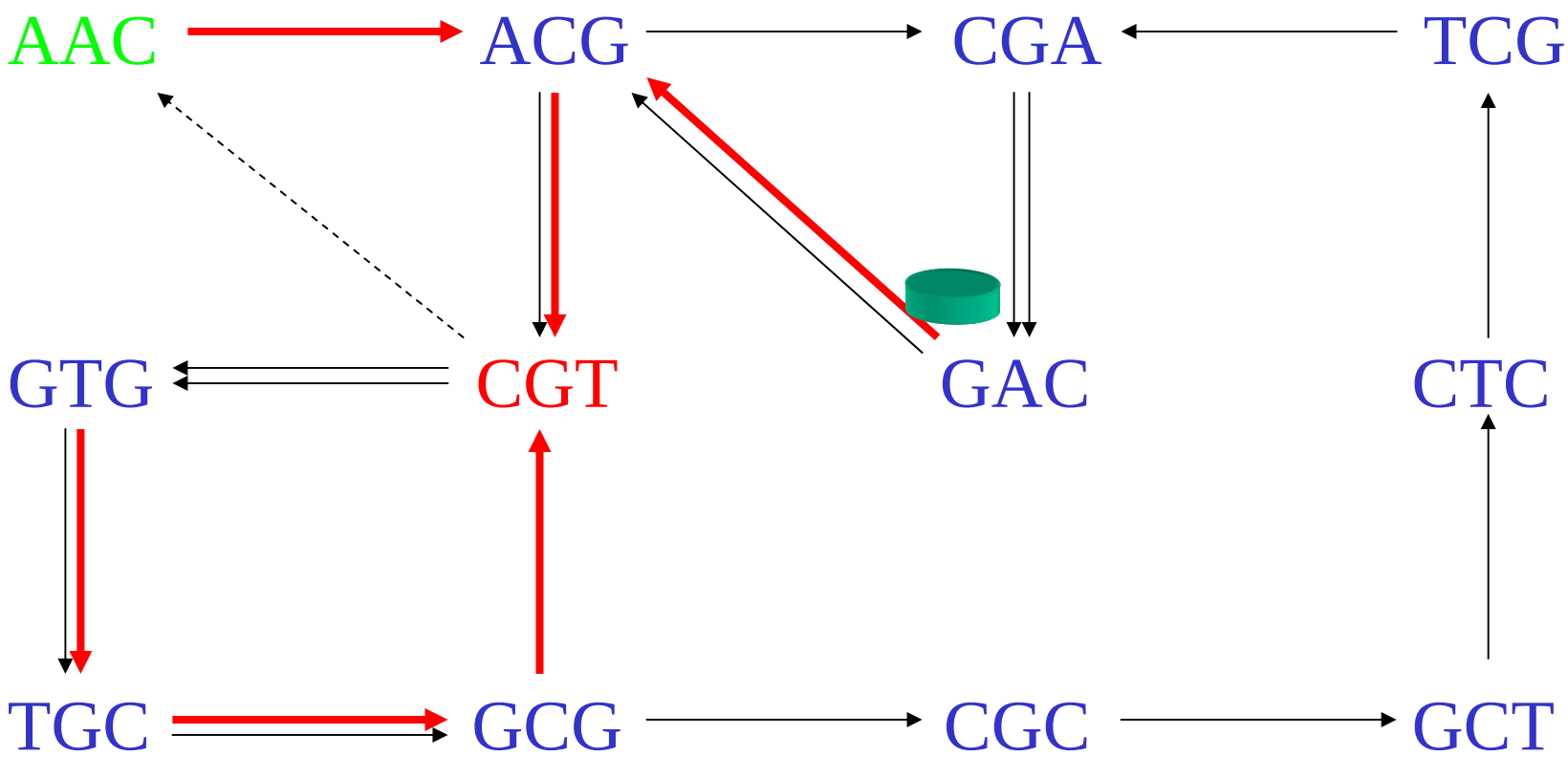
Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder 90]
[Wilson 97]

Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder 90]
[Wilson 97]

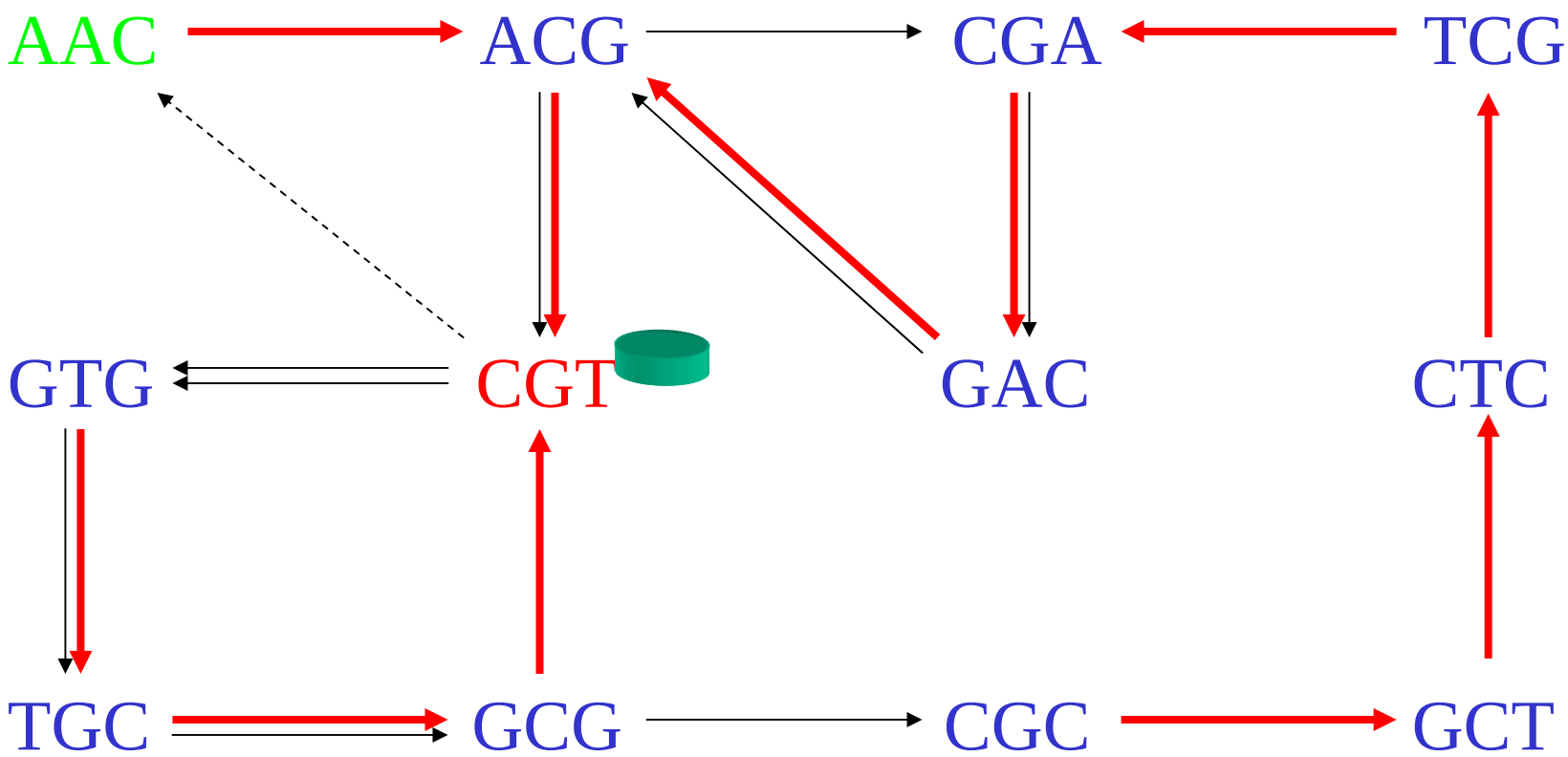
Génération en fréquences exactes

• • •

Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder 90]
[Wilson 97]

Génération en fréquences exactes



Engendrer un arbre couvrant aléatoire uniformément

[Aldous, Broder 90]
[Wilson 97]

Génération en fréquences exactes

Algorithme.

Partir du sommet final

Tant qu'on n'a pas un arbre couvrant :

Choisir uniformément un anti-arc adjacent a et le traverser.

Si a n'appartient pas à l'arbre et n'y ajoute pas de cycle,

l'ajouter à l'arbre.

Complexité moyenne : $O(\#\text{sommets}^2 \times \#\text{arcs})$.

Vers des modèles statistico-syntaxiques

Enrichir les modèles par plusieurs types de propriétés des séquences biologiques considérées, pour **affiner les résultats** de la comparaison biologique/aléatoire.

→ Ajouter aux paramètre statistiques classiques des **paramètres structurels** (syntaxiques).

Deux types d'approches :

- Approches analytiques
- Approche expérimentale

Variants des sites de polyadénylation des ARNm

UCCAACAUCACAGCCCAGCCCACCCACUGGGUAAUAAAAGUGGUUUGUGG
UAACUUCUUUUAAGUAGUUGAUGUGGAAAACAUUUUAAAGUGAAUUUGUC
GUUUUGUGUUUUAUCCAACUUUUGUGCAUAUAUAUAAAGUAUGUCAUGGC
CUUUUCCCUCCCUGGUGCUCAUUGGAAUCUGAGUAGAGUCUGGGGGAGGG
AUCACUGUAAUUAUUAUUAUUUUCUACAAUAAAUGGGACCUGUGCACAGG
GACCCAGAUGGGAUGUUCGGAUCGGUUUGUAAUUAAACCUGGGAAUGGCC
GGAUACACAAAUAAGUCAGUUAAAAUACAUAAAUAAAAACAUAAACCUGC
CAGGAGGGGAACGUGGUAAAACCCAAGACAUUAAAUCUGCCAUCUCAGGC
UUUUUUGUUUCAGUACCAGAGGCACUGACUUCAAUAAAGUUUAUUUAUAC
GAACUCUGCCCUGCCUGGGACUCUAUUUAUUCUGAUUAAAGGGGUUUUGC
GAAAAUAAUGAAUGAAAAUCAACAGAUGAAUAAAUGGUUCUUUAUAAGUG
AGGCCAGCCAGCUUGGGAGCAGCAGAGAAAUAAACAGCAUUUCUGAUGCC
ACCGGGGAAGCCGUCAGCUGCUGUGACAAUAAAACCUGCCCCGUGUCUGG
UUGAAAUUGAAAUGCUUUAUCUGUGUUUUCUGUAAAUAAAAGAGUGCAAU
GGUCUGCUCCCCACCCCUGCCUCGGAAGAAUAAAGAGAAUGUAGUUCCCU
UGUUAAGUAGUUGUUUUAAAAUACUUAAUAAAAUAAUUCUUUUCCUGUGG
GAAAUGUAAAGAUUACUUGAGGUGUUUAAAAUAAAUUUUUCAUUCAGACU
ACUACCAUCUCUCUCUUAAAACGAGAUCAGGUUAGCAAAUGAUGUAAAAG
UUUAACCGUAUGUAAACUUGGUUUCUAAUUAAAAAAAAUUUCUUUUUCC
GUUAUUUUGUACUUGUCUUAAUACACUAAGUGUAAUAAAAACGGCUUGAG
GCAAGAGUUCGAAUAGAAAGUUAUGUACCAAGUAACCAUUUCUCAGCUGC
AUCUAGUUUCCUAUGGAAAAGAAGAUGGCAGAUACAGGAGAGACGACAGA
ACUGUAAAGAGAAGUAAUUUUGCUCCUUGAUAAAGUAUUAUAUUAAUAAU
UCUGAGUACCCGCCGCUUCACAGGCUGAGUCCAGGCCUGUGUGCUUUGUA
UUGAUAAAUCUAGAAAAUGCAUUCAUACAAUUACAGAAUUCAAAUAUUGC
UUUUUUUCUUUUUUGCUACUGCAAACGAUGCUAUAAUAAAUGUCCUUAUC
UCUAUUUUUUCUCUCCUUUUCUUUUCUUCAAUAAAAAGAAUUAAAAACCC
GCUGGGGAGGGGGGAGGGGAACUUUGUUGGGAAUAAACUUCACUCUGUGG
UGCAUCUUCCAAAGCUAUUUCGAAAUAAACACGAAAAUUUACAGUUUGCC
AGAUUAUUUGUGAUCCCAUCCAUUCCCCAAUAAAGCAAGGCUUGUCCGAC
UUCUACUUGUUCUAAAACAAUCUGUCCACAAAUAUAAAACUAUAAGUAAU

(Base : étude de [Beaudoing *et al.* 2000])

Motif	Rang non conditionné	Rang conditionné
AAUAAA	**1**	**1**
AAAUAA	2	1300
AUAAAA	3	404
AUAAAG	4	34
CAAUAA	5	167
AUAAAU	6	4078
AAAAUA	7	420
UUUUUU	8	2
UAAAAU	9	211
AUUAAA	**10**	**3**

Conditionnement par un motif surreprésenté

Combien de fois attend-on un motif H_2 sachant qu'un autre motif H_1 est surreprésenté ?

Soit

$$T(z,u_1,u_2) = \sum_{n\geq 1, k_1+k_2\geq 0} Pr(X_{1,n}=k_1 \text{ et } X_{2,n}=k_2) z^n u_1^{k_1} u_2^{k_2}$$

Alors

$$\begin{aligned} E(X_{2,n}|X_{1,n}=k_1) &= \frac{\sum_{k_2\geq 0} k_2 Pr(X_{1,n}=k_2 \text{ and } X_{2,n}=k_2)}{Pr(X_{1,n}=k_1)} \\ &= \frac{[z^n u_1^{k_1}]\frac{\partial T}{\partial u_2}(z,u_1,1)}{Pr(X_{1,n}=k_1)}. \end{aligned}$$

Un autre résultat de grandes déviations

[Régnier, AD 03]

Si H_1 apparaît k fois dans une séquences de longueur n, avec $k > E(X_{1,n})$, alors

$$E(X_{2,n}/X_{1,n} = k) \approx k \frac{D_{1,2}(z_a) \times D_{2,1}(z_a)}{D_1(z_a)(D_1(z_a) + z_a - 1)}$$

où

$$D_{i,j}(z) = (1 - z)A_{i,j}(z) + P(H_j)z^{|H_j|},$$

z_a est la plus grande racine positive de l'équation

$$D_1(z)^2 - (1 + (a - 1)z)D_1(z) - az(1 - z)D_1'(z) = 0$$

et $a = k/n$.

Need for more constrained models (2)

Gène	aaacgt	aacgtg	aac.1.gtg	aactgt	aca.14.tgc	aca.15.gca	aca.6.gca	…	tgccaa
GDH3	0	0	0	**4**	0	0	0	…	**2**
YBR043C	**2**	0	0	0	0	0	0	…	0
APG14	0	0	0	**4**	0	0	0	…	**2**
AGP1	0	0	**2**	**2**	0	0	0	…	**2**
CHA1	0	0	**2**	**2**	0	0	**4**	…	**2**
UGA4	**4**	0	0	0	0	0	0	…	**2**
PRB1	0	0	**2**	0	0	0	0	…	**4**
CAN1	0	**2**	0	0	0	0	**2**	…	0
GAT1	0	0	0	0	**2**	0	0	…	0
UGA1	0	0	0	0	0	0	0	…	0
MEP1	**2**	0	0	0	0	**2**	0	…	0
YGR125W	0	0	**2**	**4**	0	0	**2**	…	0
DUR3	**2**	0	0	0	**4**	**2**	0	…	0
YHR029C	0	0	0	0	0	**6**	**16**	…	**2**
DAL1	**2**	0	0	0	**4**	**6**	0	…	0
DAL4	**2**	0	0	0	**4**	**6**	0	…	0
DAL2	0	0	0	0	0	**2**	0	…	**2**
…	…	…	…	…	…	…	…	…	…

95 genes, 44 motifs, 4 families

[van Helden 2004]

The constrained shuffling model

[Barth, Cohen, AD, **Rivière** 2004]

Biological sequence (reference):

S = TCACATCACGACATACACACG

#Occurrences of k-lets (k=4):

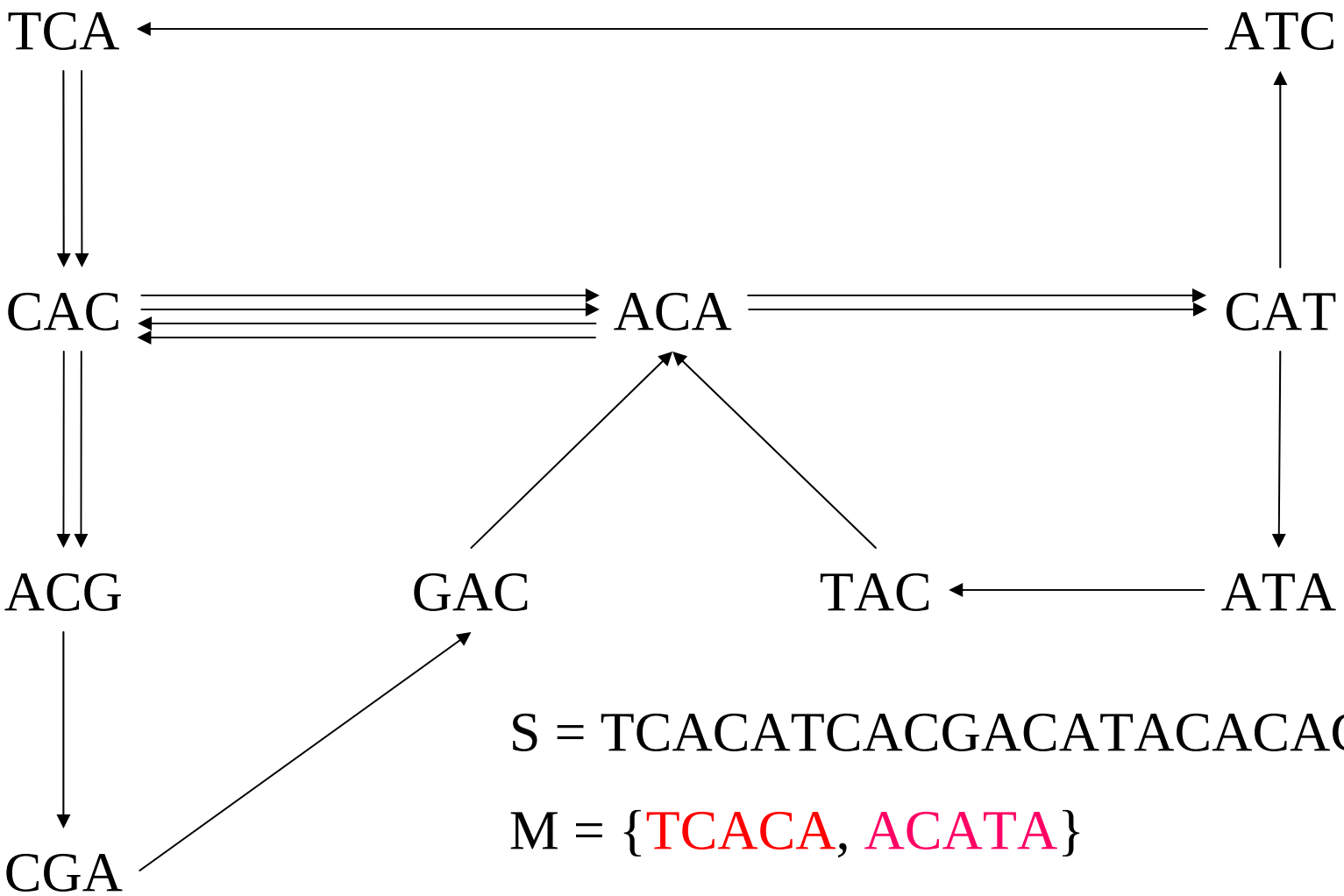
TCAC : 2	CACG : 2	ATAC : 1
CACA : 2	ACGA : 1	TACA : 1
ACAT : 2	CGAC : 1	ACAC : 2
CATC : 1	GACA : 1	
ATCA : 1	CATA : 1	

Multiset of motifs : M = {TCACA, ACATA}

Problem: generate uniformly random sequences

- which have exactly the same numbers of k-lets as S
- such that each motif of M appears at least as many times as its number of occurrences in M (without overlaps).

The constrained shuffling model

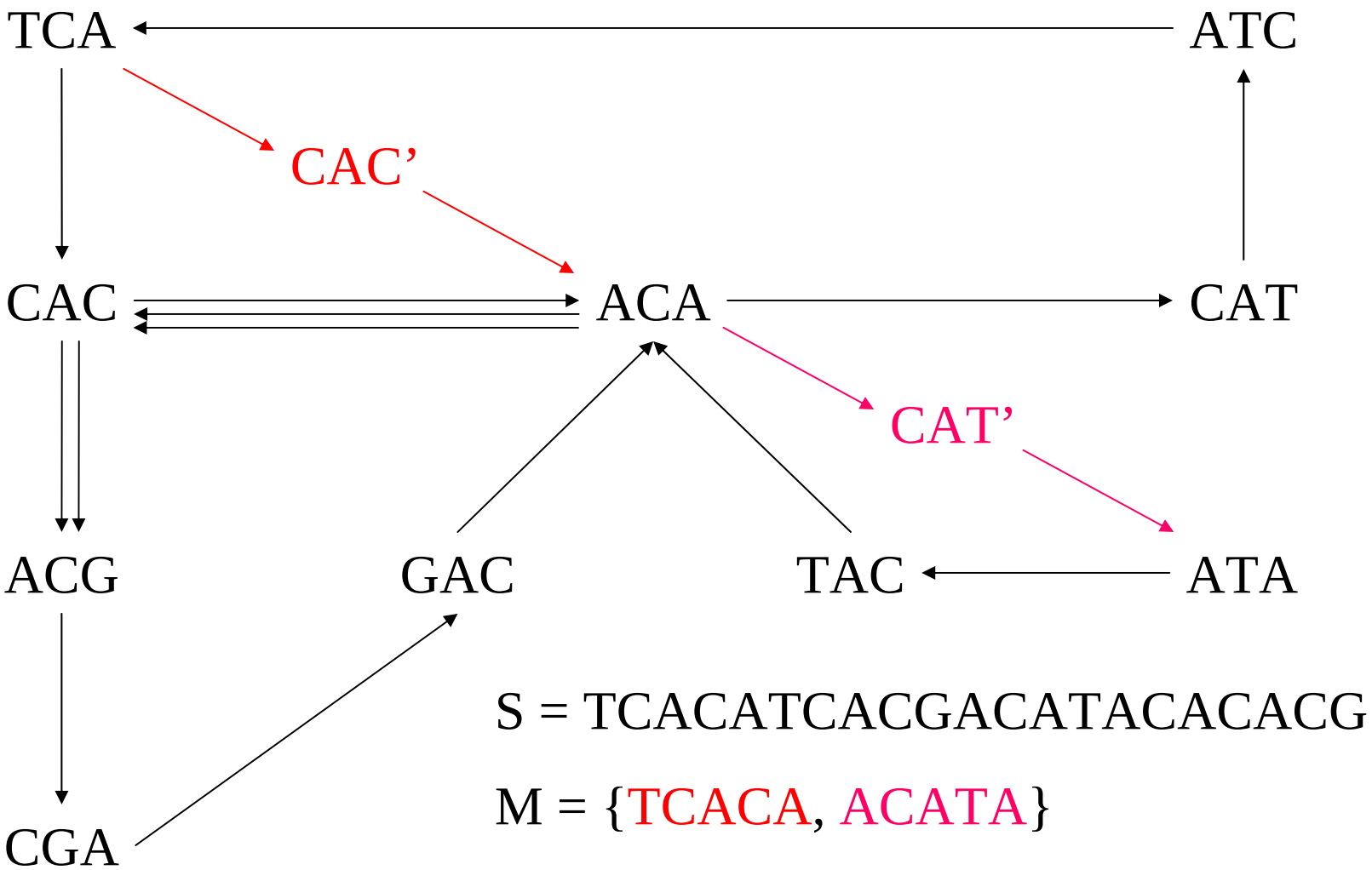


S = TCACATCACGACATACACACG

M = {TCACA, ACATA}

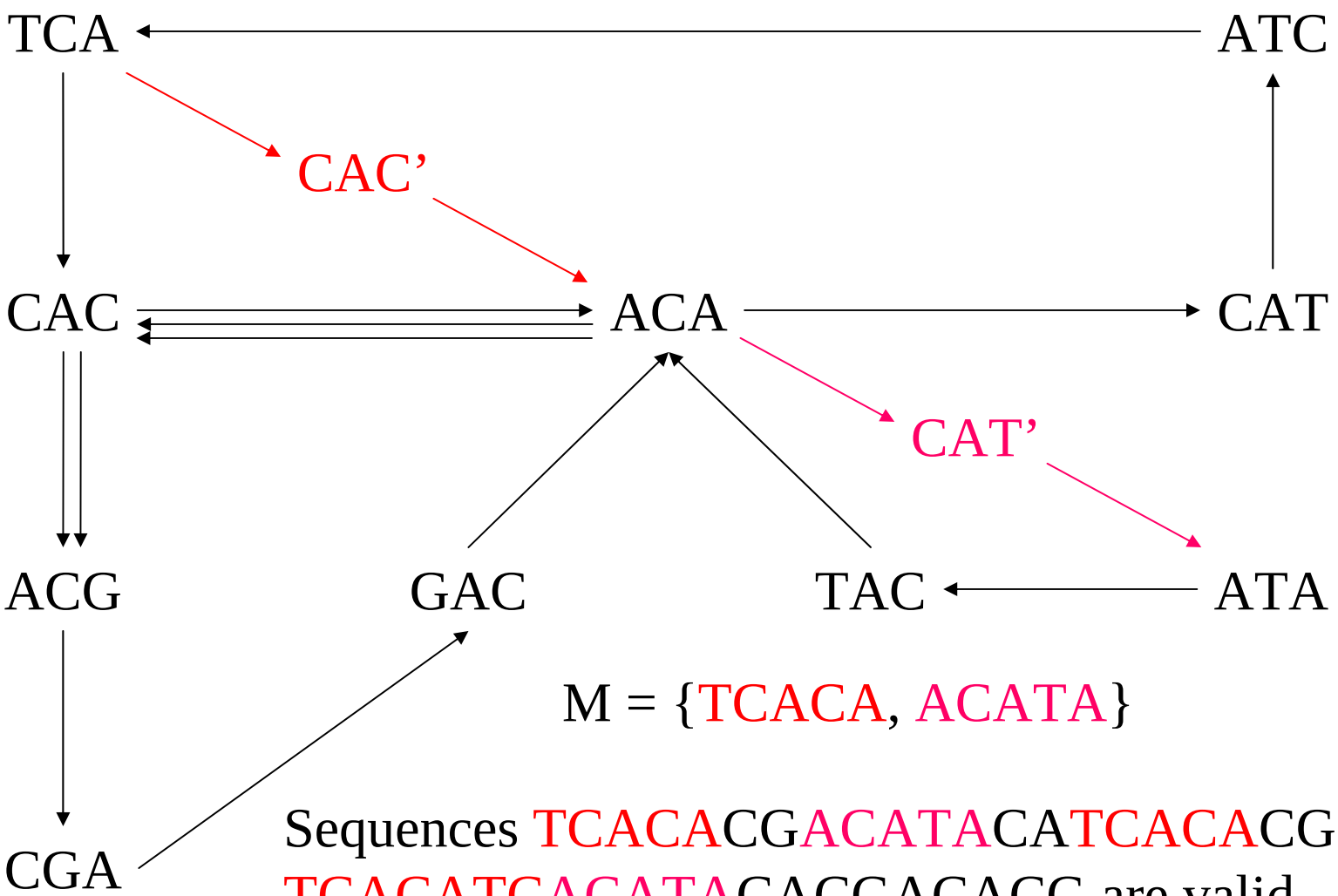
Sequence TCACGACACACATACATCACG is not valid.

The constrained shuffling model



Sequence TCACGACACACATACATCACG is not valid.

Problem 1 : Motifs overlaps



M = {TCACA, ACATA}

Sequences TCACACGACATACATCACACG and TCACATCACATACACGACACG are valid.

Sequence TCACGACATCACATACACACG is not valid, while it follows an Eulerian trail.

Problem 1 : Motifs overlaps

Problem :

Data :

- A multiset M of motifs on an alphabet X.
- A sequence S on X.

Question : Is S valid ?

Theorem : This problem is NP-complete.

Proof : reduction of 3DM.

Problem 1 : Motifs overlaps

Problem :

Data :

- A multiset M of motifs on an alphabet X.
- A sequence S on X, corresponding to an Eulerian trail in a constrained sequence graph with M as constraints.

Question : Is S valid ?

Theorem : This problem is NP-complete.

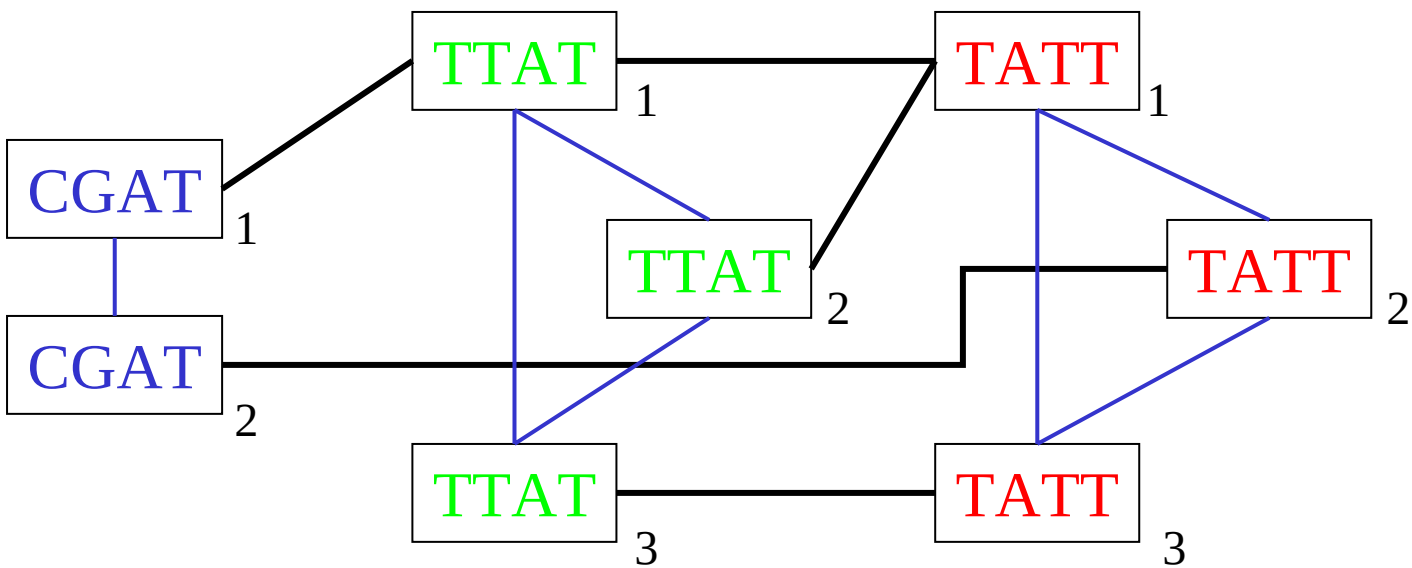
Proof : reduction of 3DM.

Problem 1 : Motifs overlaps

S = {ATTATCGATTATATTATCCGACGATATTCTATTAT}
M = {TATT, CGAT, TTAT}

Finding a maximal independent set in the following graph :

- Occurrences of same motifs form cliques (blue edges)
- Two occurrences are connected if they overlap (black edges)

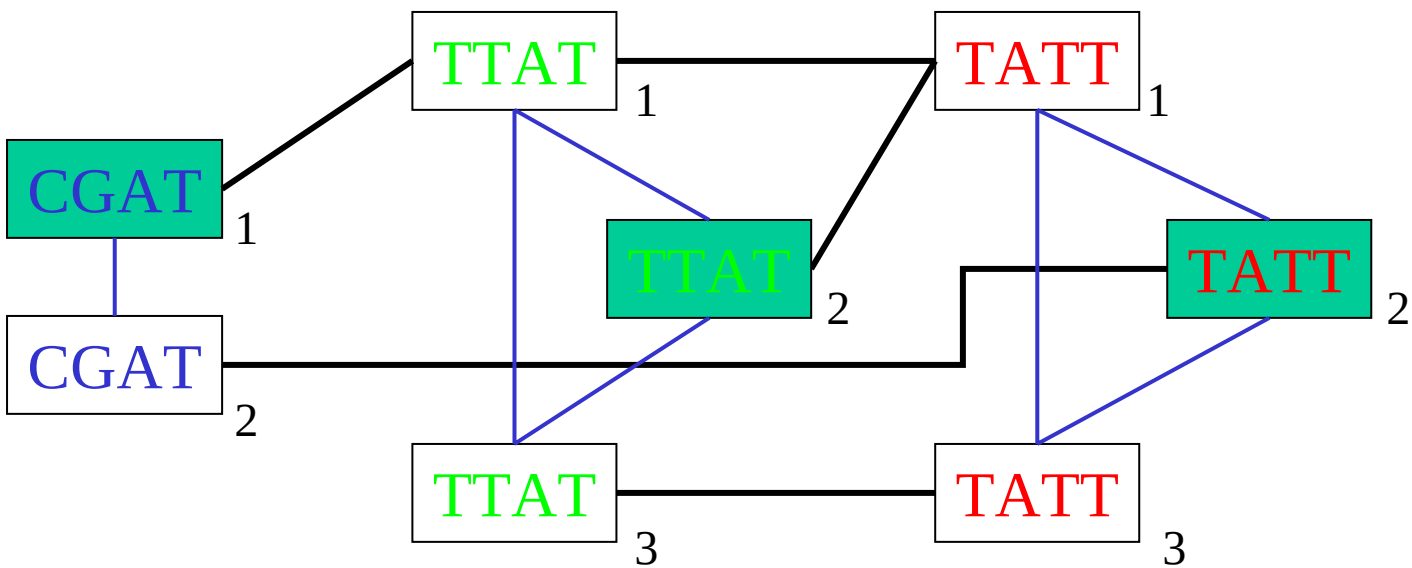


Problem 1 : Motifs overlaps

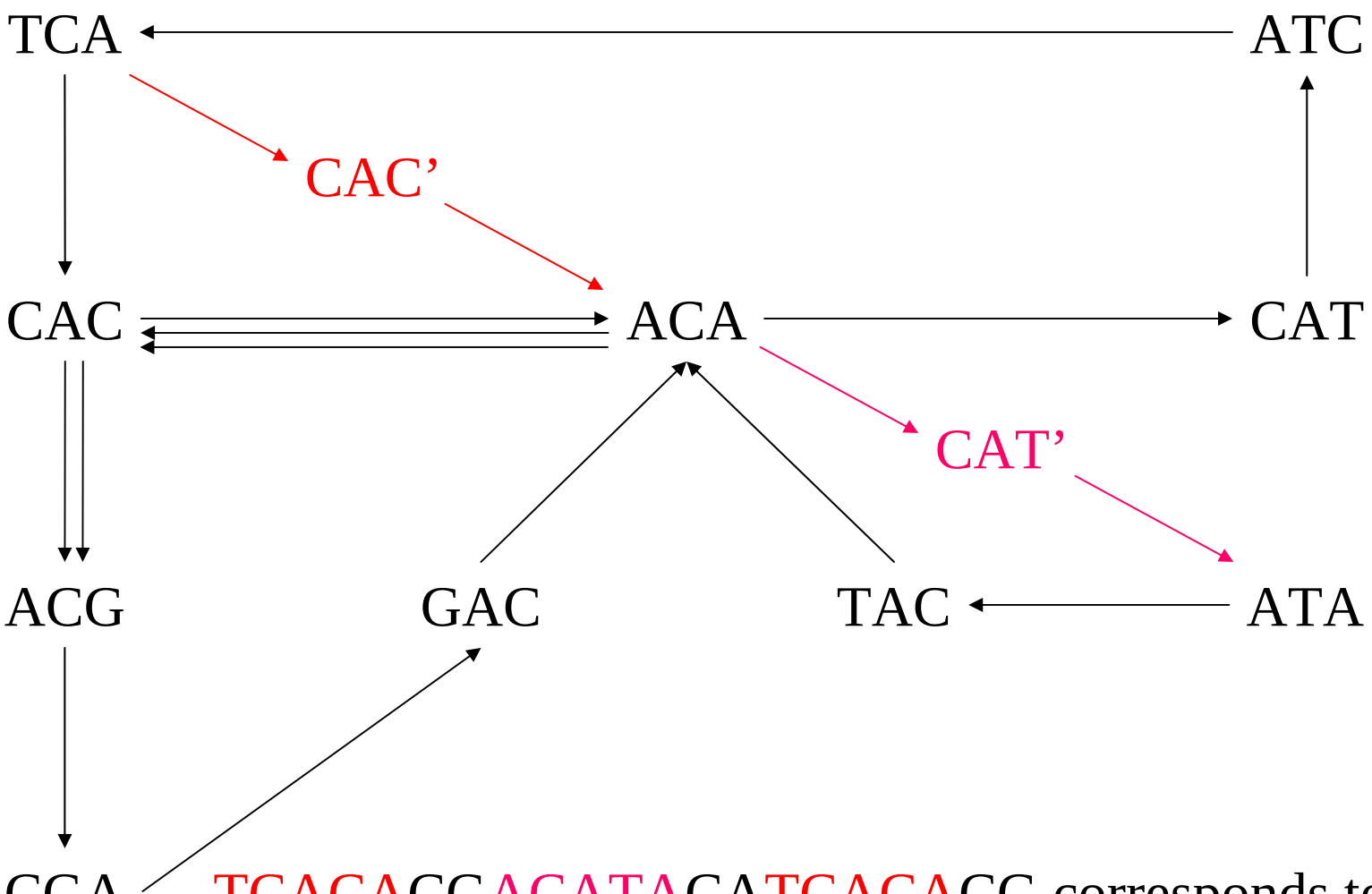
S = {ATTATCGATTATATTATCCGACGATATTCTATTAT}
M = {TATT, CGAT, TTAT}

Finding a maximal independent set in the following graph :

- Occurrences of same motifs form cliques (blue edges)
- Two occurrences are connected if they overlap (black edges)



Problem 2 : Non-uniformity



TCACACGACATACATCACACG corresponds to **two** Eulerian trails, while TCACACATCACGACATACACG corresponds to **only one**.

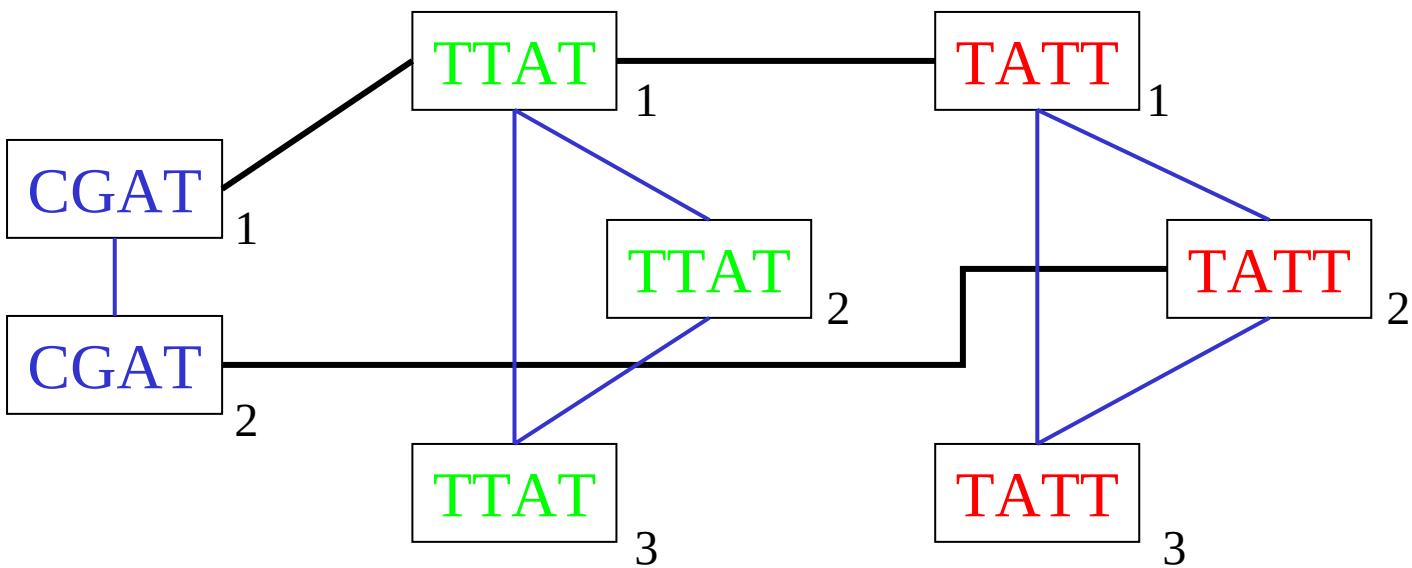
Problem 2 : Non-uniformity

S = {ATTATCGATTATATTATCCGACGATATTCTATTAT}
M = {TATT, CGAT, TTAT} **k=2**

Counting maximal independent sets in the following graph:

- Occurrences of same motifs form cliques (blue edges)
- Two occurrences are connected if they overlap

by at least k-1 letters (black edges)



Random generation algorithm (1)

1. Construct the constrained sequence graph G according to S, k and M.

2. Draw a random uniform Eulerian trail in G.

3. Verify if the corresponding sequence R is valid. If not, goto 2.

4. Count the number *t* of Eulerian trails in G which correspond to R.

5. Accept the sequence with probability 1/*t*, or reject it and goto 2.

Experimental results

Experimental results :

- Works well in practice [tested up to length(S)=50000 and card(M)=50] when motifs of M are unlikely to appear many times in the sequences.

- Step 3 (test for valid sequence) is very fast in almost all cases.

- Steps 4 and 5 (counting and rejection) constitute the actual bottleneck when motifs are likely to appear many times in the sequences.

Random generation algorithm (2)

0. Divide M in two subsets : M_1 the set of unlikely motifs
   M_2 the set of likely motifs

1. Construct the constrained sequence graph G according to S, k and M_1.

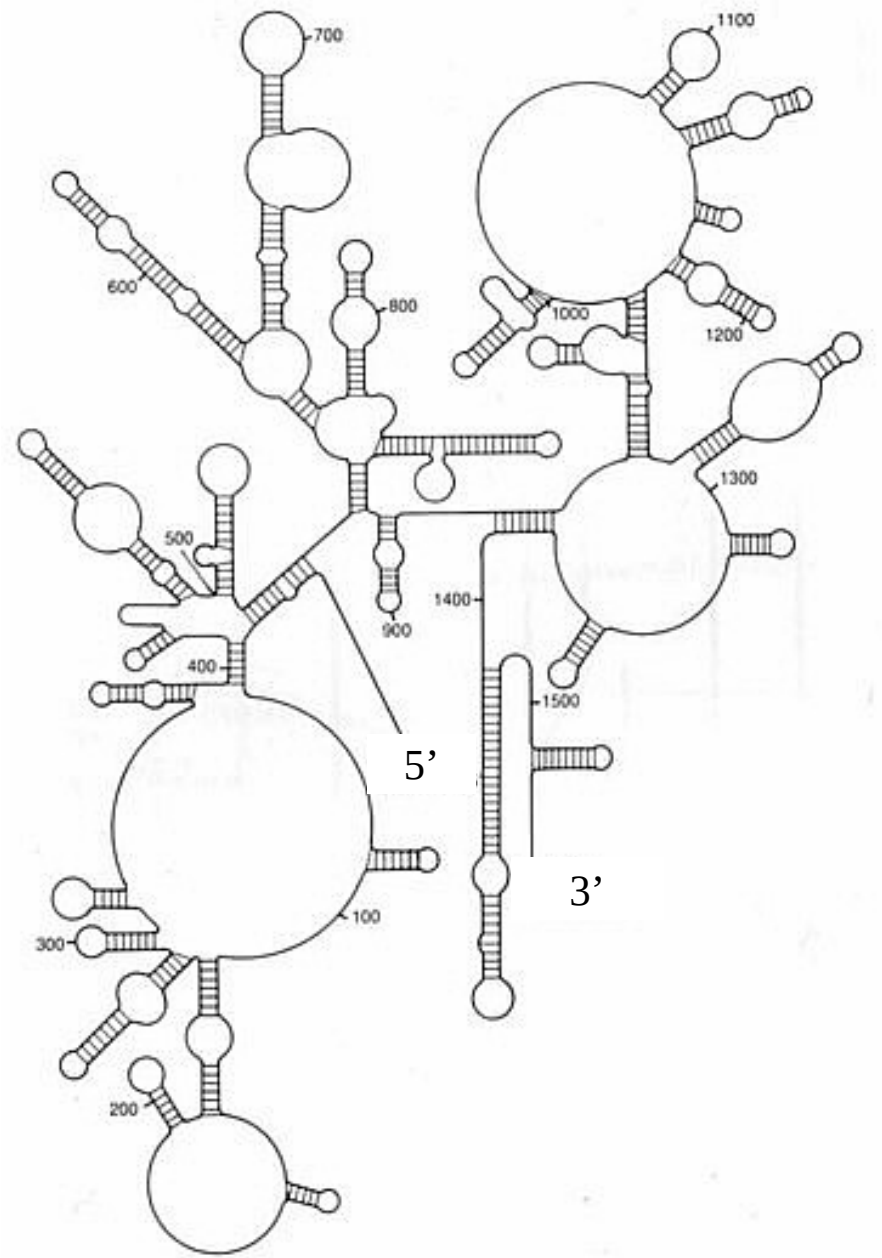
2. Draw a random uniform Eulerian trail in G.

3. Verify if the corresponding sequence R is valid over $M_1 \cup M_2$.
   If not, goto 2.

4. Count the number *t* of Eulerian trails in G which correspond to R.

5. Accept the sequence with probability 1/*t*,
   or reject it and goto 2.

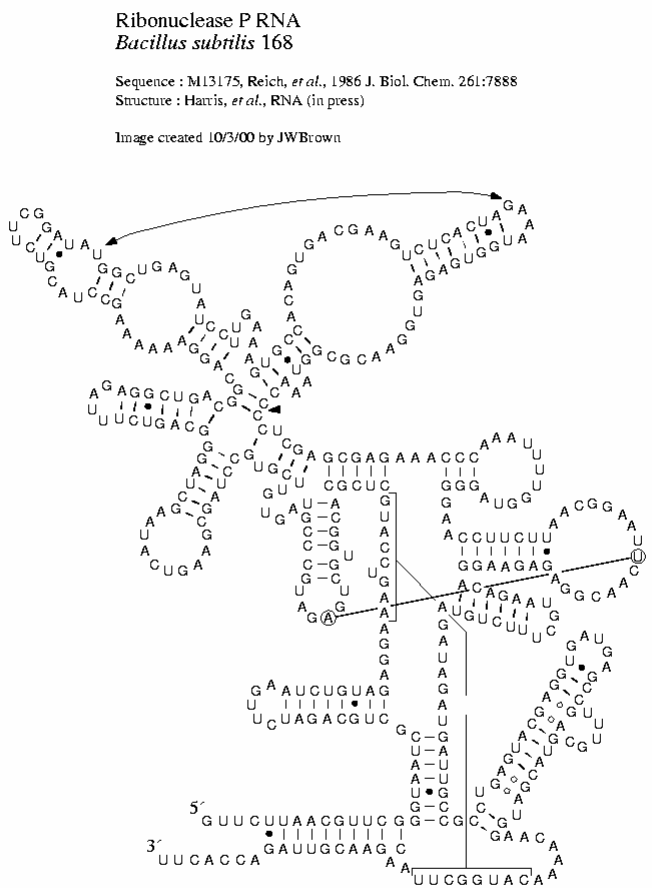
Contraintes structurelles : ARN

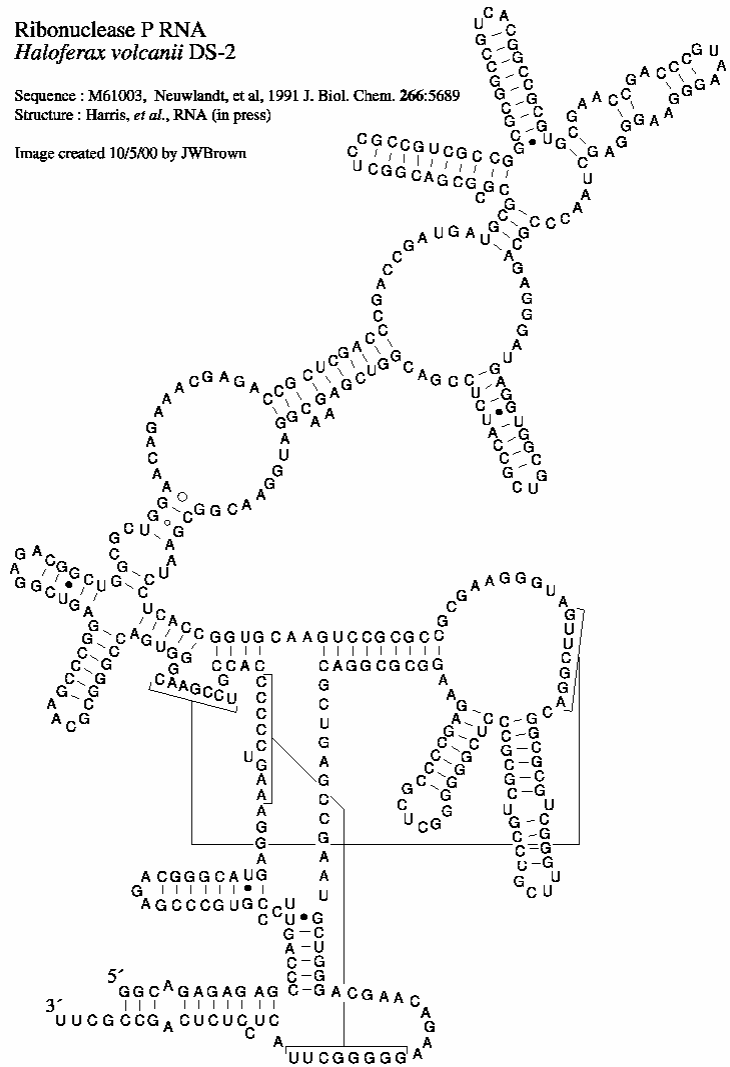


ARN 16S
E. coli

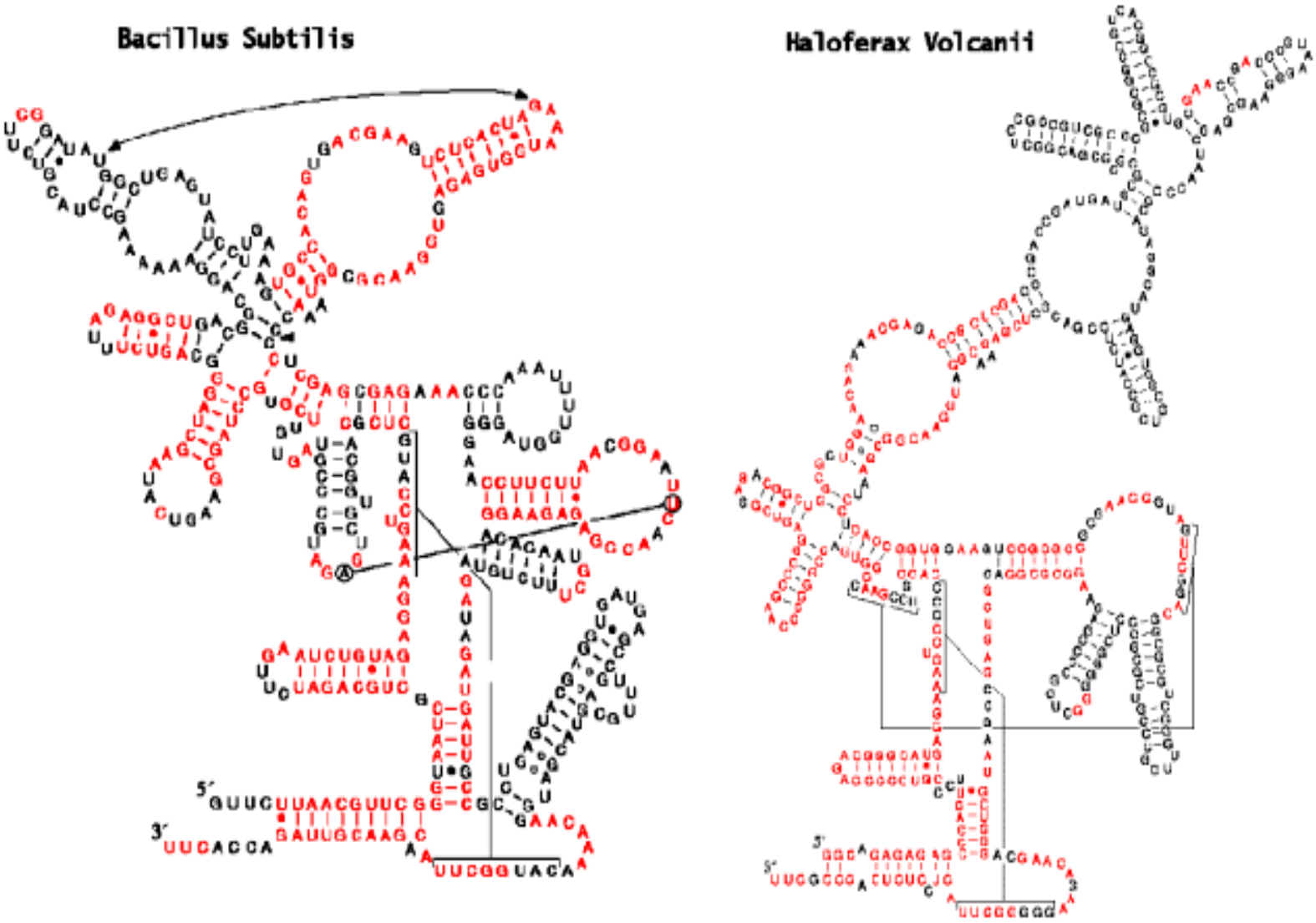
Comparaison de structures secondaires

[Dulucq, Tichit 2001]

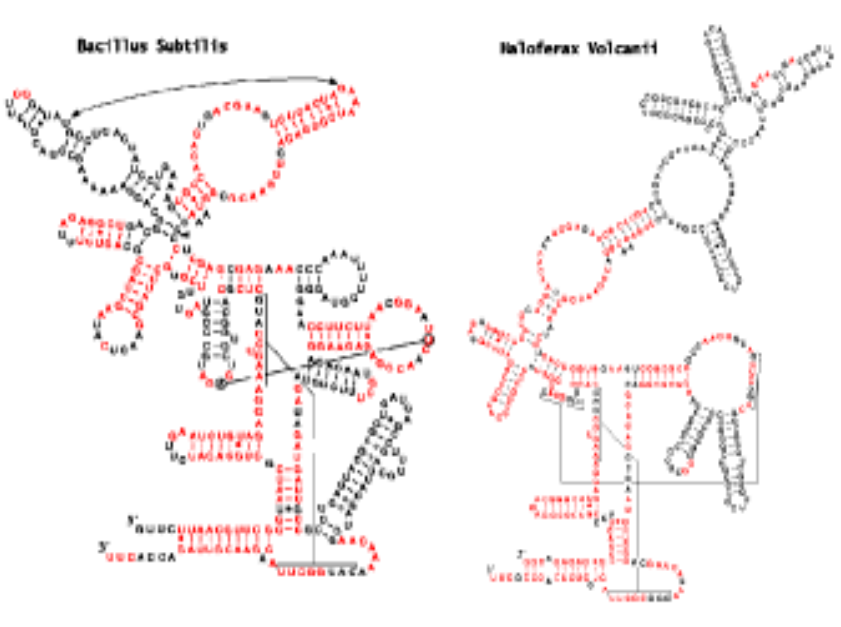




Comparaison de structures secondaires



Comparaison d'ARN



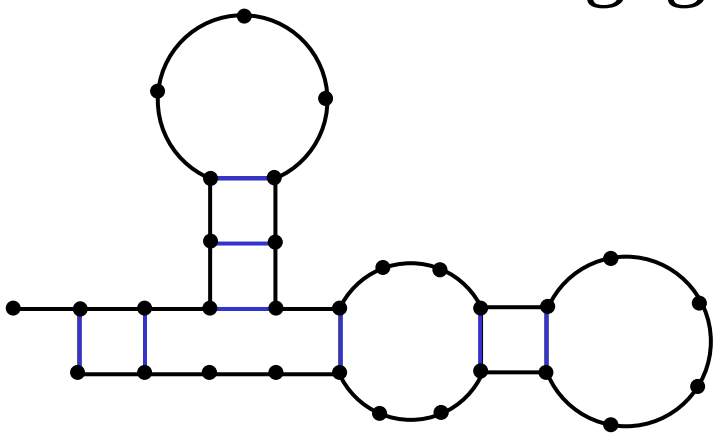
Objectifs : déterminer des paramètres biologiques pour la mesure de distance (matrices de substitution), étalonner et comparer les algorithmes, définir des seuils d'homologie.

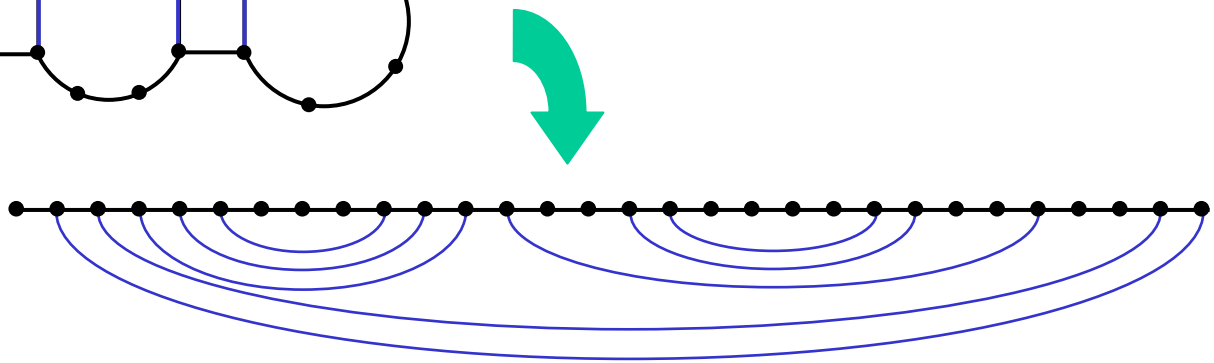
→Modèles combinatoires d'ARN et génération aléatoire.

Comment engendrer aléatoirement des structures d'ARN ayant des propriétés statistiques (nbe de boucles, longueurs des tiges…) similaires à celles d'une structure biologique de référence ?

Un langage pour les structures secondaires

[Vauchaussade, Viennot 85]





- (((((- - -))) (- - ((- - - -)) - -) - -))

c a a a a a c c c b b b a c c a a c c c c b b c c b c c b b

S → a S b S | c S | ε

Une grammaire pour les structures secondaires d'ARN

$$S \rightarrow a\,S\,b\,S \mid c\,S \mid \varepsilon$$

S → **c**S → c**a**S**b**S → cabS → cab**c**S → cabc

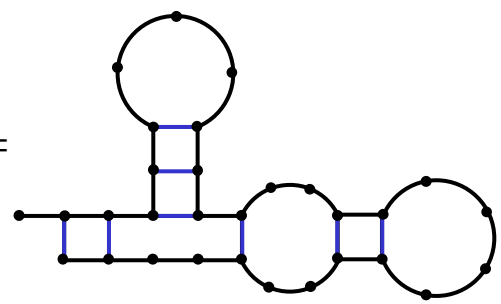
S → **c**S → c**a**S**b**S → ca**a**S**b**SbS → caa**a**S**b**SbSbS

→ caaa**a**S**b**SbSbSbS → caaa**a**S**b**SbSbSbS

→ caaaa**a**S**b**SbSbSbSbS → caaaaa**c**SbSbSbSbSbS

→ …

→ caaaaacccbbbaccaaccccbbccbccbb =



Génération en fréquences moyennes

Entrée :

- Alphabet $X = \{x_1, x_2, \ldots, x_k\}$,
- Langage L sur X.
- $n \in \mathbf{N}$, $\vec{v} = (v_1, v_2, \ldots, v_k)$. $\qquad (1 = v_1 + v_2 + \ldots + v_k)$

Sortie : un mot de L_n.

Contraintes :

- Tout mot de L_n peut être engendré.
- Les fréquences moyennes des lettres dans les mots engendrés respectent (asymptotiquement) le vecteur v :

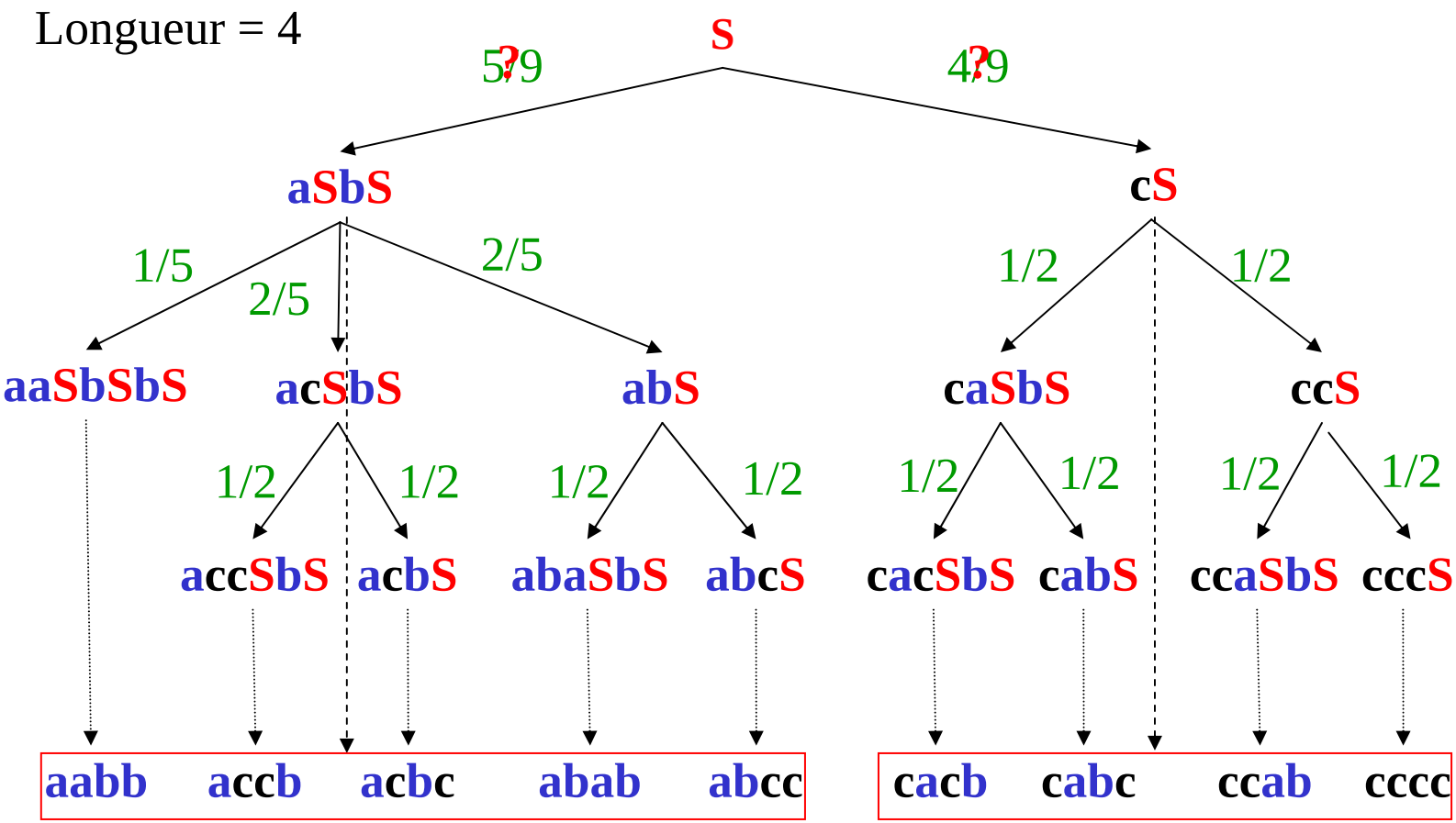
$$\frac{1}{n} \sum_{w \in L_n} |w|_{x_i} p(w) \ \sim \ v_i \quad \forall i \in \{1, 2, \ldots, k\}.$$

- Deux mots ayant la même distribution de lettres ont la même probabilité d'être engendrés.

Génération aléatoire uniforme

[Wilf 1977, Hickey, Cohen 1983, Flajolet, Zimmerman, Van Cutsem 1994]

S → a S b S | c S | ε



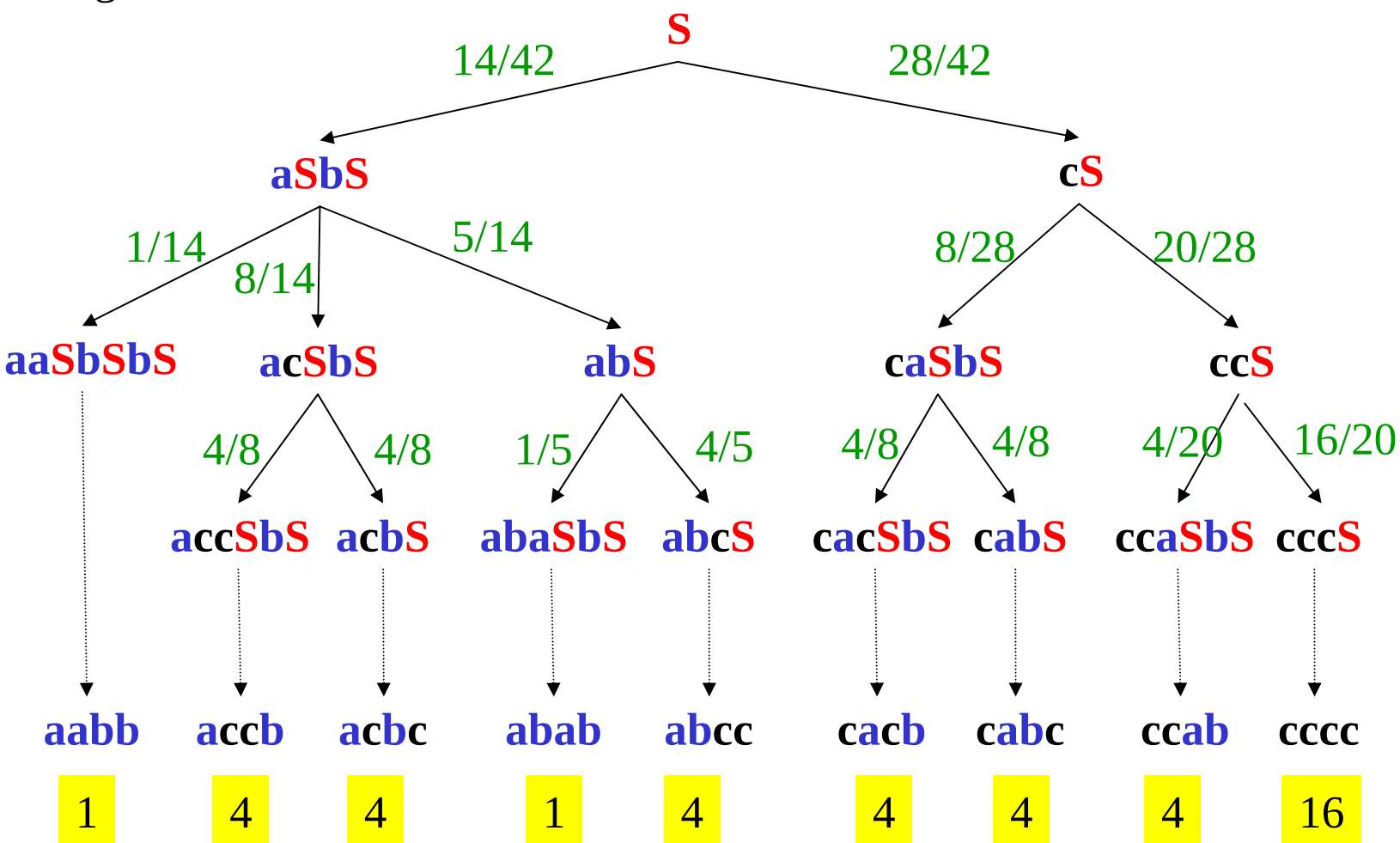
Génération aléatoire non uniforme contrôlée

[AD, O.Roques, M.Termier 2000]

S → a S b S | c S | ε

+ de nucléotides non appariés : Poids π(a) = π(b)=1
π(c) = **2**

Longueur = 4



Distribution des lettres

Séries génératrices :

$$L_\pi(t,\vec{x}) = \sum_{w\in L} \pi(w) t^{|w|} x_1^{|w|_{x_1}} \dots x_k^{|w|_{x_k}},$$

Distribution de la lettre x_i :

$$\mu_i(\pi) \quad = \quad \frac{\sum_{w\in L_n} |w|_{x_i}\pi(w)}{n\pi(L_n)} \quad = \quad \frac{[t^n]\Gamma_{\pi,x_i}(t)}{[t^n]\Gamma_\pi(t)}$$

avec

$$\Gamma_{\pi,x_i}(t) = \frac{\partial L_\pi(t,\vec{x})}{\partial x_i}(t,\vec{1}) \quad \text{et} \quad \Gamma_\pi(t) = t\frac{\partial L_\pi(t,\vec{x})}{\partial t}(t,\vec{1})$$

Calculs de fréquences et de pondérations

La pondération π étant donnée, quelle est la fréquence moyenne μ_i de la lettre x_i ?

Soit $f_\pi(t, x_1, x_2, \ldots, x_k) = \sum \pi(w) t^{|w|} x_1^{|w|_{x_1}} x_2^{|w|_{x_2}} \ldots x_k^{|w|_{x_k}}$

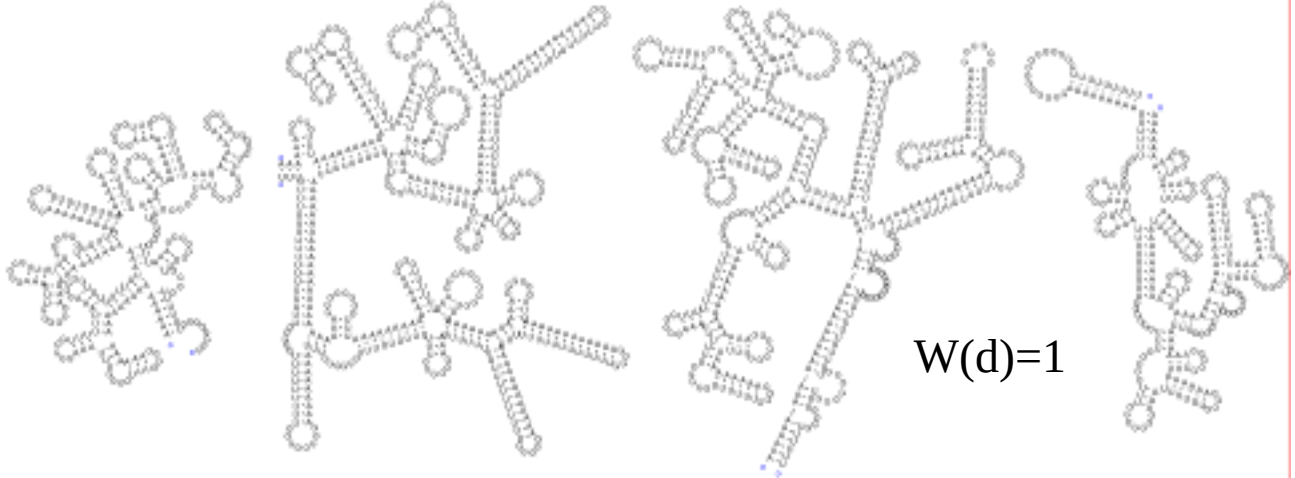
Alors $\boxed{\mu_i(\pi) = \dfrac{[t^n]\Gamma_{\pi,x_i}(t)}{[t^n]\Gamma_\pi(t)}}$ où $\Gamma_{\pi,x_i}(t) = \dfrac{\partial f_\pi(t,x_1,x_2,\ldots,x_k)}{\partial x_i}(t,1,1,\ldots,1)$

et $\Gamma_\pi(t) = t\dfrac{\partial f_\pi(t,x_1,x_2,\ldots,x_k)}{\partial t}(t,1,1,\ldots,1)$

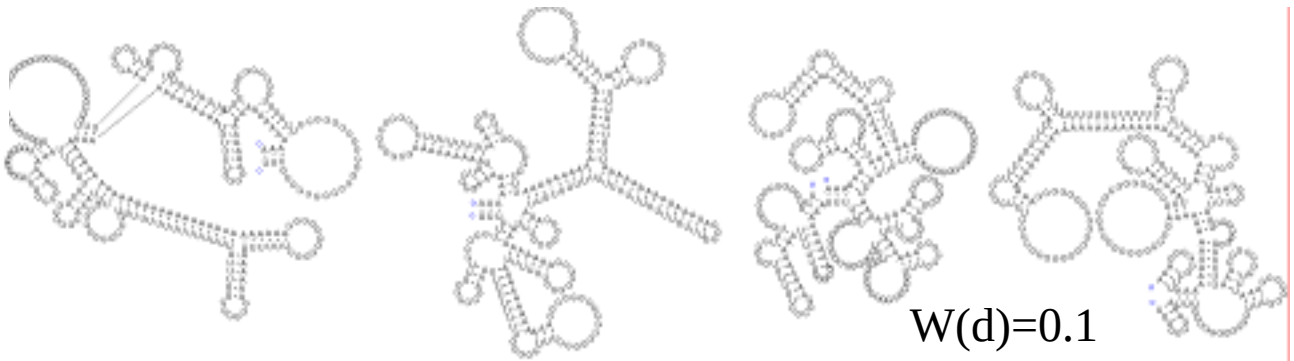
Les fréquences des lettres étant données, quelle doit être la pondération π ?

- Cas rationnel fortement connexe : résoudre un système d'équations algébriques.
- Cas algébrique : à traiter…

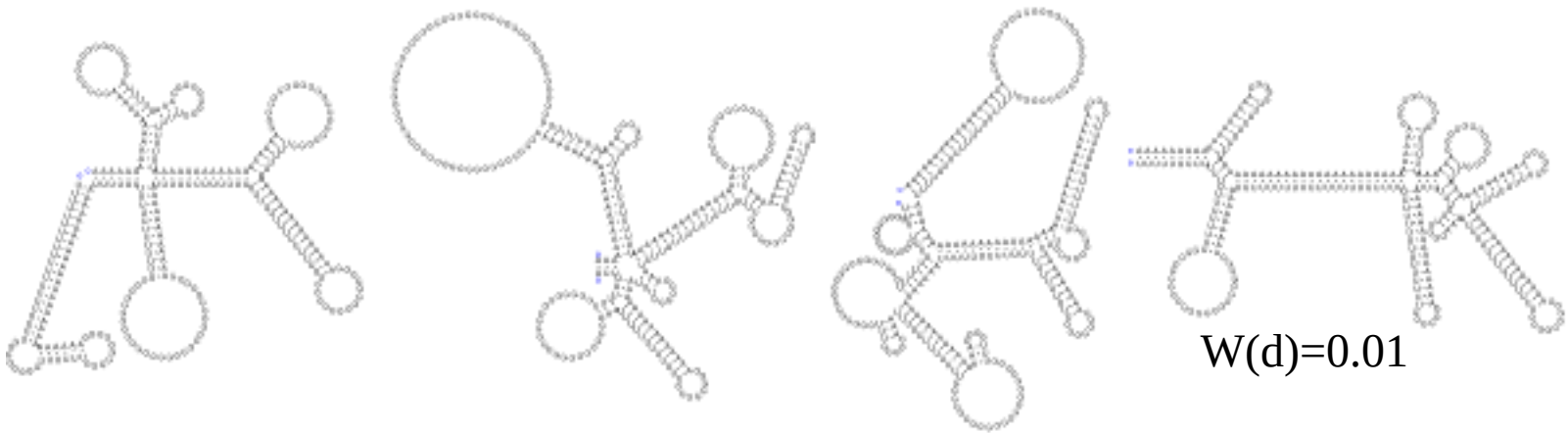
```
TYPE = GRAMMAR

SYMBOLS = LETTERS

RULES =

 S ::= A A A S U U U S;
 S ::= A A A S U U U;
 S ::= d T G;

 T ::= G T;
 T ::= G G G;

WEIGHTS =

  d 0.1
```



W(d)=1



W(d)=0.1



W(d)=0.01

Logiciel GenRGenS [Y.Ponty]

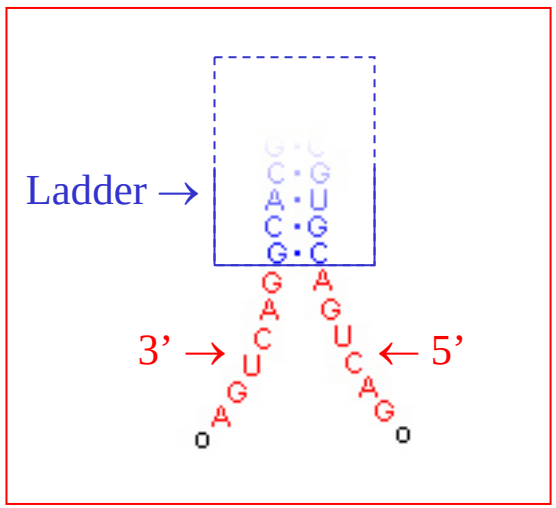
Une grammaire pour la structure secondaire d'ARN

(Inspirée de Waterman 78)

$$ARN \rightarrow 3' \text{ Ladder } 5'$$

$$3' \rightarrow t_3\ 3' \mid \varepsilon \qquad 5' \rightarrow t_5\ 5' \mid \varepsilon$$

ARN →

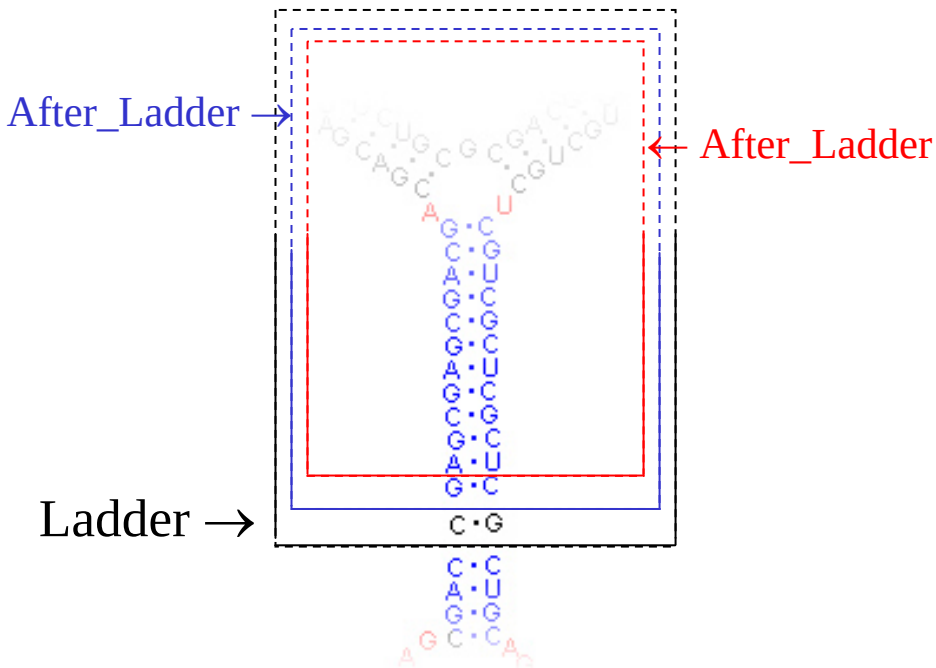


Une grammaire pour la structure secondaire d'ARN

ARN → 3' Ladder 5'

Ladder → a After_Ladder b

After_Ladder → a After_Ladder b



Une grammaire pour la structure secondaire d'ARN

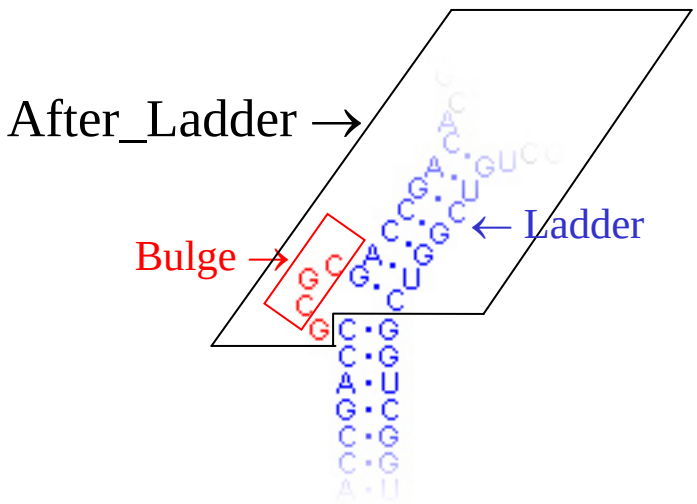
ARN → 3' Ladder 5'

Ladder → a After_Ladder b

After_Ladder → a After_Ladder b

| c Bulge Ladder

Bulge → c Bulge | ε



Une grammaire pour la structure secondaire d'ARN

ARN → 3' Ladder 5'

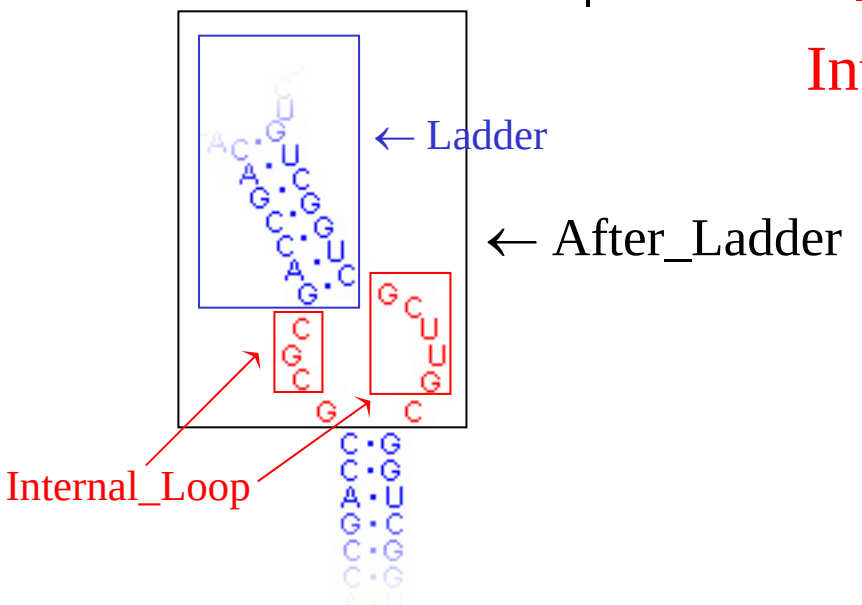
Ladder → a After_Ladder b

After_Ladder → a After_Ladder b

| c Bulge Ladder | Ladder c Bulge

| d Internal_Loop Ladder Internal_Loop d

Internal_Loop → d Internal_Loop | ε



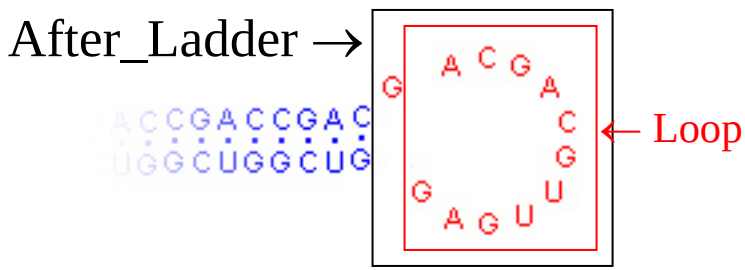
Une grammaire pour la structure secondaire d'ARN

ARN → 3' Ladder 5'

Ladder → a After_Ladder b

After_Ladder → a After_Ladder b
| c Bulge Ladder | Ladder c Bulge
| d Internal_Loop Ladder Internal_Loop d
| e Loop

Loop → e Loop | ε



Une grammaire pour la structure secondaire d'ARN

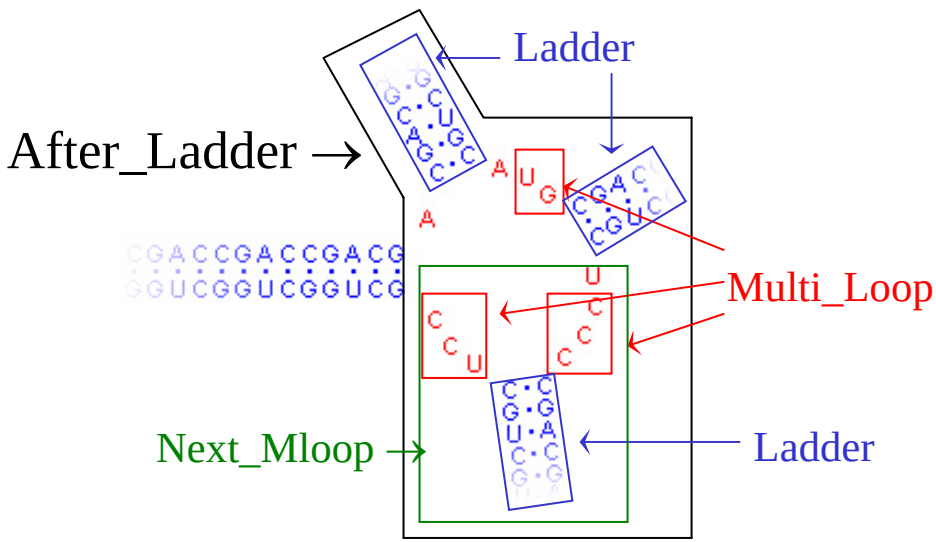
ARN → 3' Ladder 5'

Ladder → a After_Ladder b

After_Ladder → a After_Ladder b
| c Bulge Ladder | Ladder c Bulge
| d Internal_Loop Ladder Internal_Loop d
| e Loop
| f Multi_Loop Ladder f Multi_Loop Ladder Next_MLoop

Multi_Loop → f Multi_Loop | ε

Next_MLoop → Multi_Loop | f Multi_Loop Ladder Next_MLoop



Génération *équiprobable* de structures secondaires

GenRGenS

aaccaaaaccadddaadafaafaadaaeebbdbbffaddaebdbfffaebfbcbffaddadaeeeebdbddbffaaeebbbdbbddbbbbbbcccccccbc

aacaffffaebfaaaafafaeebffffaaffacccaccadaaeebbdbbbfaebffadaebdbfffbbfbfaebfbbbbfffaeeeebffbbcbccccc

caafaeebfacaafffacccaaaaebbbccbbfacacaeebbbbccccbbfffffaffafaaebbfffaddaebdbbfaaaaaebbbcccccbcbbffbbc

ccaafaacaebbcbfacafaeebffaafaaebcbfadacafadafaebfaaebbbdbfafaebfaaaebbbfbbbdbffaadaebddbcbbbfffbbbcb

caaffafaebfaebfafafafaccaadacaaebbbdbbbffaebbfaaeeebbbffafaffaaebbfffffffaeeebbfaeeebfbbffbfaaebbfbb

aaafaadddddddaaafaddaaaaccacafaddddddadaaebbddbddbffaebbbbbbccccbdbffaeeeeebfffbbbddddddbbfaaeebcbbbb

adafaacaaeeeeebcbbccbffadadafaffaffffffffaaccaaaebbccbbbffacccaeeeeebbffbfaeeeebfffbfaeeebfbdbddbbdb

cafaebfffaadddaccafaaccaccaaeebccbbbcccbfaebfffaaeebbfaebfffaaccaebbbbbdbbfaaadddddafaeebfaebbdbbbb

ccccccaaacaaaacafaffaebfacaddddafffacaaebccbbfaebbddddbbfaccaaaaebbcbbbffbffaaebcbfaeeebbbbcccbbbbcb

afacaccaebbbffaffaadaebddbcccbffaaffaaebbffaeeebbbffbfaaaddddaaaccafaeebfaacaeebbbbbbccccbddbbbfbc

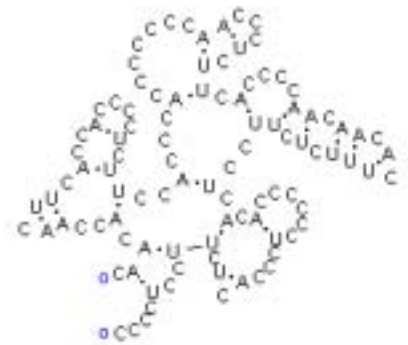
afaebfaafacaeeebbfffaebfaebffbccbfaaafffaebfacafaadddaebddbcbfaacccaaaacaeebbbbbbcbfffaebbbffbbccbbc

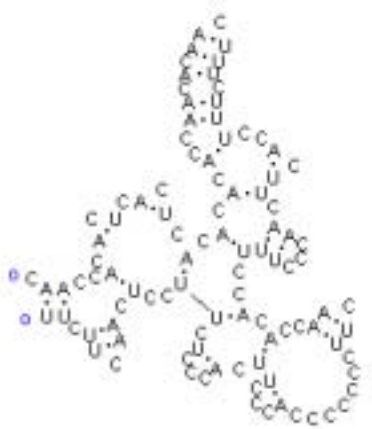
affafaaaeebbbfaafffaffaeebffacaffaacaeebbccccbfaeebbbffbfaebfaaebbfbccccccbbfffaccadacaeebbdbbbcccc

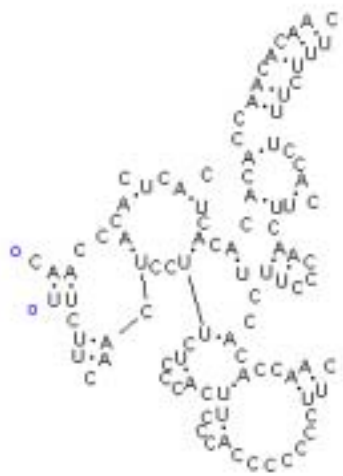
afffaaebcbfffaeeebfffafafaebffaeebfafaaccaebbbfaccaaccaebbcbbbffacaaaebcbbbbfadaaacaeebbccbbddbbffbc

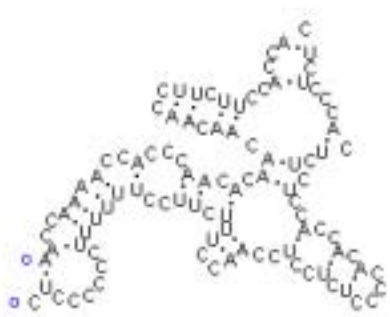
+ RNAViz

Génération *équiprobable* de structures secondaires

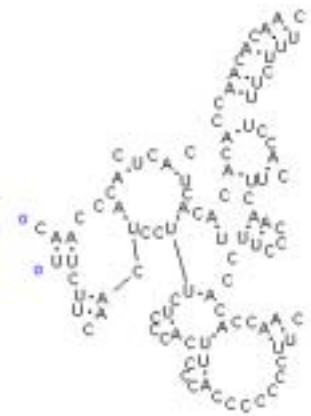


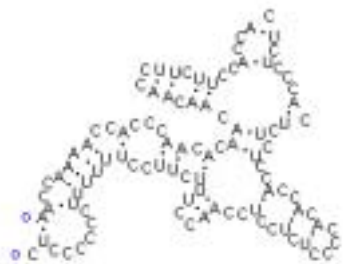






Génération *équiprobable* de structures secondaires





- Structures trop complexes
- Trop de bases non appariées
- Trop de Bulges
- Bulges trop gros
- Boucles pas assez grosses

$\Rightarrow$ Pondération des terminaux

$\Rightarrow$ On contraint ainsi les fréquences des terminaux.

On ne peut pas contraindre les fréquences des Bulges, Loop, … !!!

Introduction de marqueurs dans la grammaire

ARN → 3’ Ladder 5’

Ladder → $\mathbf{m_a}$ After_Ladder $\mathbf{m_b}$

After_Ladder → a After_Ladder b
| $\mathbf{m_c}$ Bulge Ladder | Ladder $\mathbf{m_c}$ Bulge
| $\mathbf{m_d}$ Internal_Loop Ladder Internal_Loop $\mathbf{m_d}$
| $\mathbf{m_e}$ Loop
| $\mathbf{m_f}$ Multi_Loop Ladder $\mathbf{m_f}$ Multi_Loop Ladder Next_MLoop

Next_MLoop → Multi_Loop | $\mathbf{m_f}$ Multi_Loop Ladder Next_MLoop

3’ → t_3 3’ | ε

5’ → t_5 5’ | ε

Bulge → c Bulge | ε

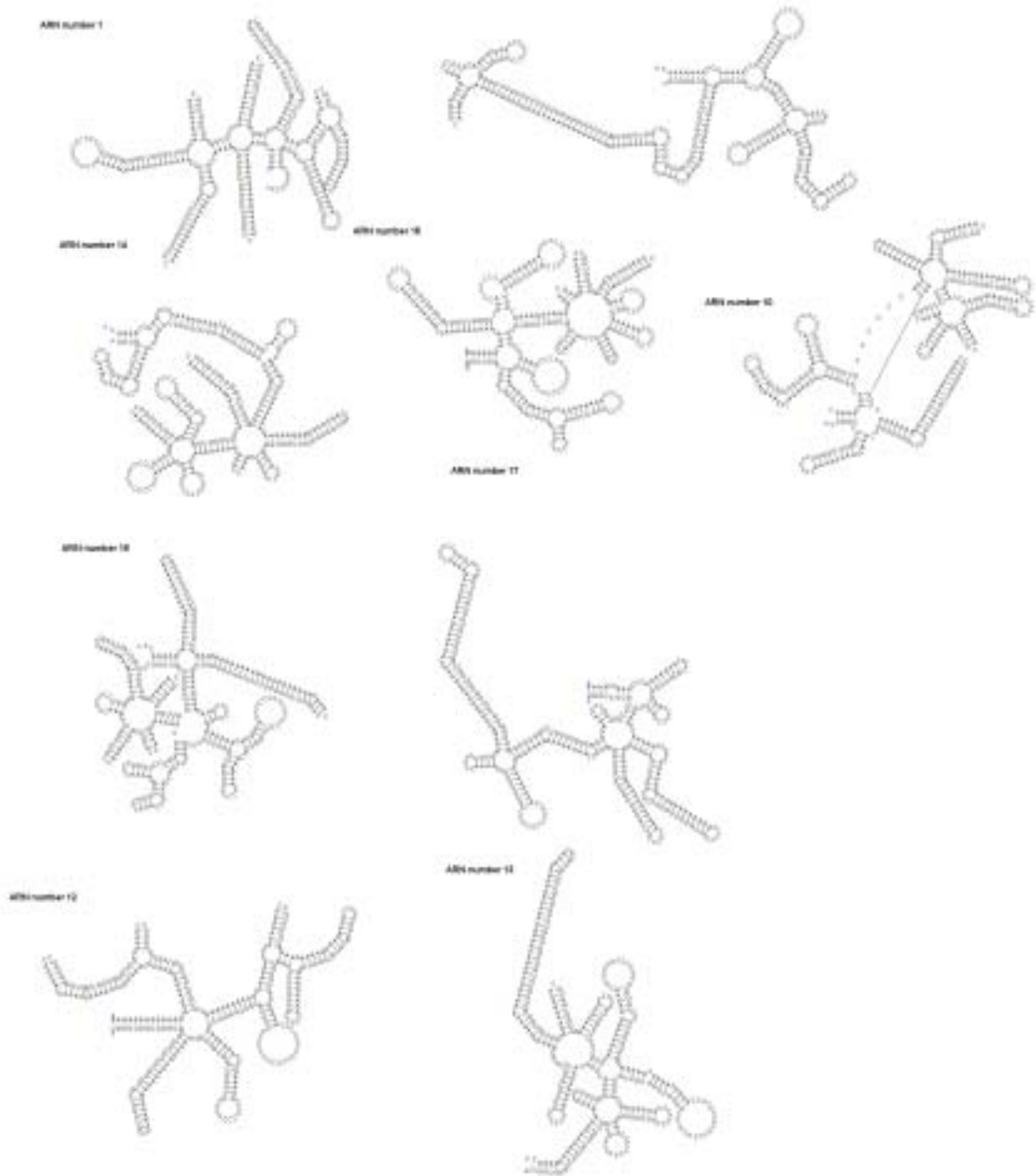
Internal_Loop → d Internal_Loop | ε

Loop → e Loop | ε

Multi_Loop → f Multi_Loop | ε

Pondérations

m_a	0.5
m_b	0.5
m_c	0.5
m_e	0.5
m_d	0.2
m_f	0.5
a	1.3
b	1.3
c	0.7
d	0.7
e	1.2
f	0.5



Pondérations

m_a	0.5
m_b	0.5
m_c	0.5
m_e	0.5
m_d	0.2
m_f	0.5
a	1.2
b	1.2
c	0.5
d	0.5
e	1.2
f	0.5