Un survol des méthodes probabilistes d’inférence phylogénétique

Un survol des méthodes probabilistes d'inférence phylogénétique

A Consistent Phylogenetic Backbone for the Fungi

Ingo Ebersberger,*[,1] Ricardo de Matos Simoes,[2] Anne Kupczok,[3] Matthias Gube,[4] Erika Kothe,[4] Kerstin Voigt,[4,5] and Arndt von Haeseler[1]

[1]Center for Integrative Bioinformatics Vienna, University of Vienna, Medical University of Vienna, University of Veterinary Medicine Vienna, Vienna, Austria

[2]Computational Biology and Machine Learning Lab, Center for Cancer Research and Cell Biology, Queens University Belfast, Belfast, United Kingdom

[3]Institute of Science and Technology Austria, Klosterneuburg, Austria

[4]Institute of Microbiology, Friedrich Schiller University, Jena, Germany

[5]Jena Microbial Resource Collection, Leibniz-Institute for Natural Product Research and Infection Biology, Jena, Germany

***Corresponding author:** E-mail: ingo.ebersberger@univie.ac.at.

Associate editor: Andrew Roger

Un survol des méthodes probabilistes d'inférence phylogénétique

Partie A: Méthodes par maximum-de-vraisemblance

Partie B: Méthodes bayésiennes

Plan pour l’avant-midi

Partie A

- Lancement d’un calcul pour plus tard…
- Introduction à l’inférence probabiliste par maximum de vraisemblance sur un exemple simple (pile-ou-face).
- Introduction à l’inférence phylogénétique par maximum de vraisemblance avec données nucléotidiques.
  - Fondements théoriques
  - Démo pratique en R
- Inférence phylogénétique avec données acides aminés.
  - Fondements théoriques
  - Retour sur calcul initial.
- Lancement d’un autre calcul pour plus tard…
- Pause

Plan pour l’avant-midi

Partie B

- Introduction à l’inférence phylogénétique bayésienne.
  - Fondements théoriques
  - Démo pratique avec le logiciel PhyloBayes
  - Retour sur calcul lancé avec la pause.
- Perspectives sur les développement récents
  - PhyloBayes MPI
  - Modèles de mutation-selection

Introduction à l’inférence phylogénétique par maximum de vraisemblence

Comprendre l’inférence ML

- Simuler un jeu de pile-ou-face.
- Inférer le parametre d’un jeu pile-ou-face.

Comprendre l’inférence ML

- Soit U un générateur de nombre aléatoire, qui retourne un nombre réel entre 0 et 1.

Comprendre l'inférence ML

- Soit U un générateur de nombre aléatoire, qui retourne un nombre réel entre 0 et 1.
- Proposez un algorithme qui simule une partie *juste* de pile-ou-face.

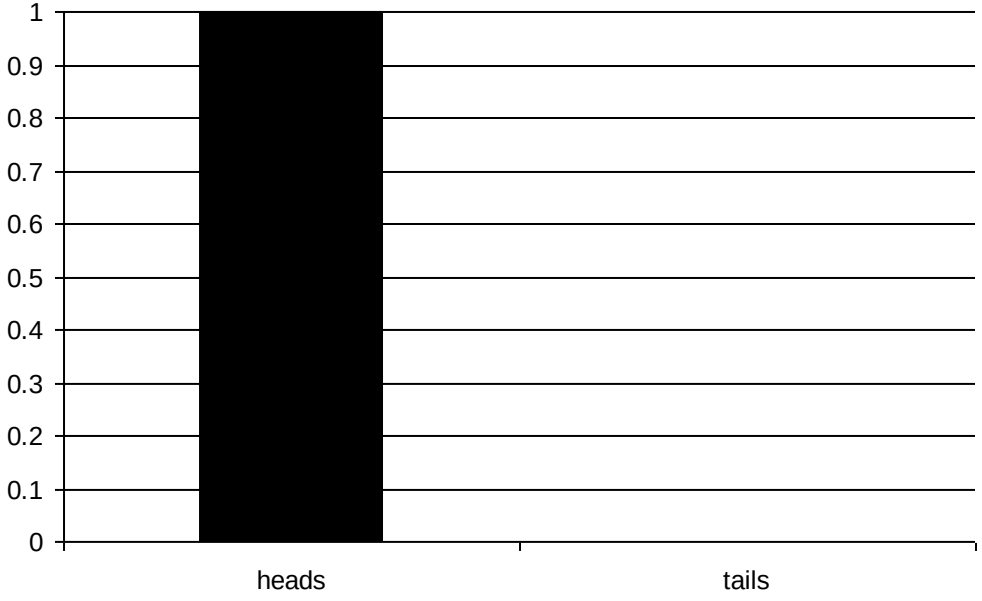
Comprendre l'inférence ML

- Soit U un générateur de nombre aléatoire, qui retourne un nombre réel entre 0 et 1.
- Proposez un algorithme qui simule une partie *juste* de pile-ou-face.
  - simulez $x \sim U$
  - si $x < 0.5$, déclarez le résultat "pile"
  - sinon, déclarez le résultat "face"

Comprendre l'inférence ML

- Soit *U* un générateur de nombre aléatoire, qui retourne un nombre réel entre 0 et 1.
- Proposez un algorithme qui simule une partie *juste* de pile-ou-face.

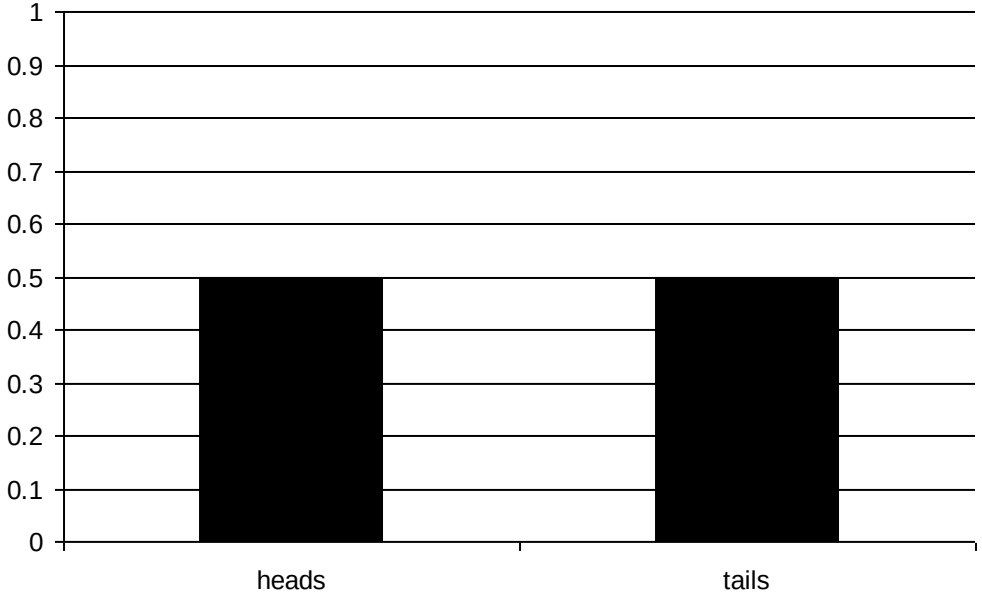
Après un coup:



Comprendre l'inférence ML

- Soit *U* un générateur de nombre aléatoire, qui retourne un nombre réel entre 0 et 1.
- Proposez un algorithme qui simule une partie *juste* de pile-ou-face.

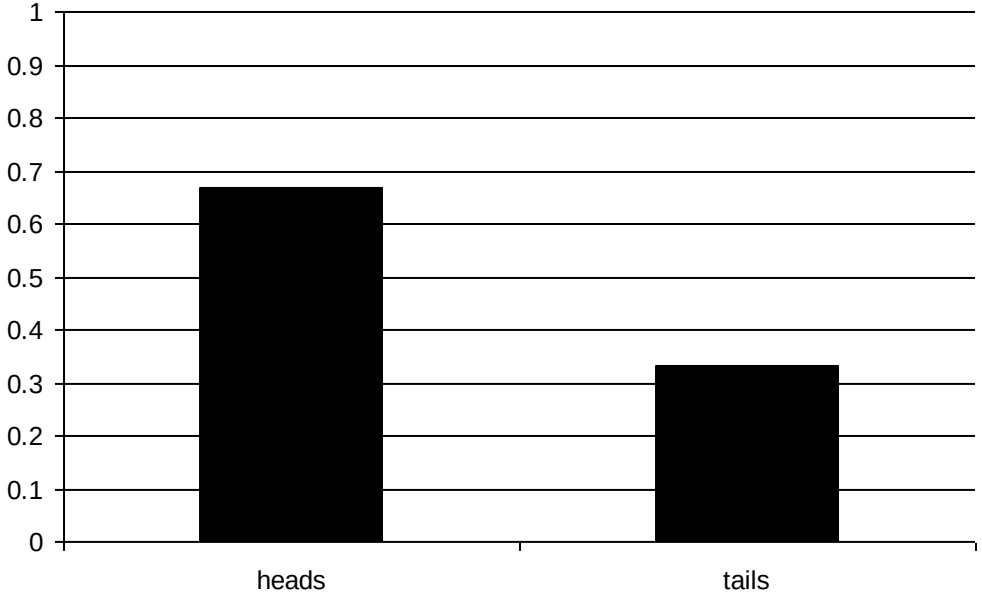
Après deux coups:



Comprendre l'inférence ML

- Soit *U* un générateur de nombre aléatoire, qui retourne un nombre réel entre 0 et 1.
- Proposez un algorithme qui simule une partie *juste* de pile-ou-face.

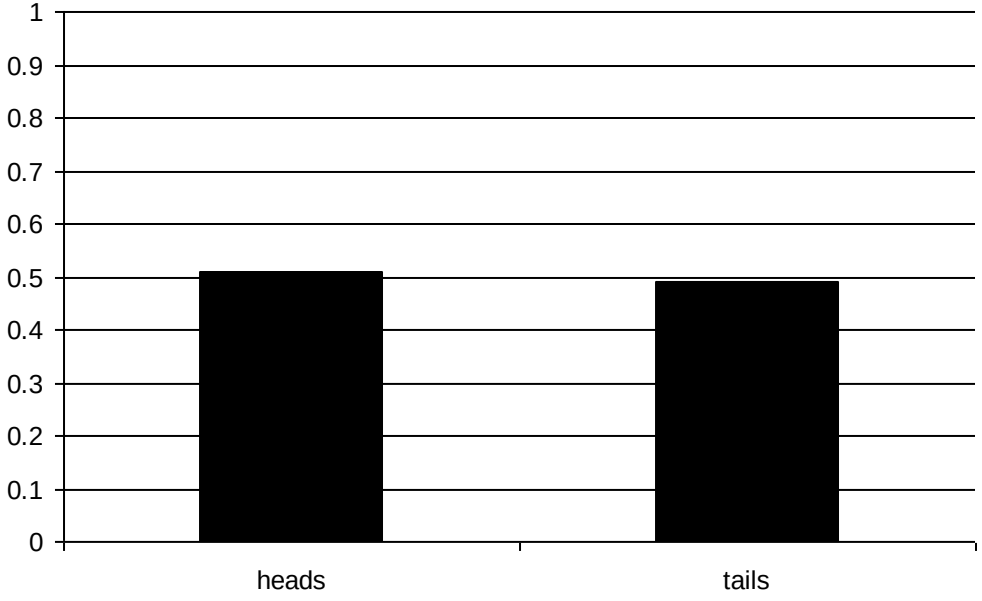
Après trois coups:



Comprendre l'inférence ML

- Soit *U* un générateur de nombre aléatoire, qui retourne un nombre réel entre 0 et 1.
- Proposez un algorithme qui simule une partie *juste* de pile-ou-face.

Après cent coups:



Comprendre l'inférence ML

- Soit U un générateur de nombre aléatoire, qui retourne un nombre réel entre 0 et 1.
- Proposez un algorithme qui simule une partie *juste* de pile-ou-face.
- Lorsque le nombre de parties tend vers l'infini, et si les parties sont bien justes, la proportion de "piles" (ou "faces") approchera 0.5.

Comprendre l’inférence ML

- Soit U un générateur de nombre aléatoire, qui retourne un nombre réel entre 0 et 1.
- Proposez un algorithme qui simule une partie ou la probabilité de “pile” est 0.45.

Comprendre l'inférence ML

- Soit U un générateur de nombre aléatoire, qui retourne un nombre réel entre 0 et 1.
- Proposez un algorithme qui simule une partie ou la probabilité de "pile" est 0.45.
  - simulez $x \sim U$
  - si $x < 0.45$, déclarez le résultat "pile"
  - sinon, déclarez le résultat "face"

Comprendre l’inférence ML

- Simuler un jeu de pile-ou-face.
- Inférer le parametre d’un jeu pile-ou-face.

Inférence ML

- “Pile” sera codé 1.
- “Face” sera codé par 0.
- Soit *h* la probabilité d’obtenir “pile”.
- Soit1 - *h* la probabilité de “face”.
- Soit *x* le résultat d’un lancé.

Inférence ML

- “Pile” sera codé 1.
- “Face” sera codé par 0.
- Soit *h* la probabilité d’obtenir “pile”.
- Soit1 - *h* la probabilité de “face”.
- Soit *x* le résultat d’un lancé.

$$p(x = 1) = h$$

Inférence ML

- “Pile” sera codé 1.
- “Face” sera codé par 0.
- Soit *h* la probabilité d’obtenir “pile”.
- Soit1 - *h* la probabilité de “face”.
- Soit *x* le résultat d’un lancé.

$$p(x = 1) = h$$

$$p(x = 0) = 1 - h$$

Inférence ML

- “Pile” sera codé 1.
- “Face” sera codé par 0.
- Soit *h* la probabilité d’obtenir “pile”.
- Soit1 - *h* la probabilité de “face”.
- Soit *x* le résultat d’un lancé.

$$p(x = 1) = h$$

$$p(x = 0) = 1 – h$$

$$p(x \mid h) = h^x(1 - h)^{1 - x}$$

Inférence ML

- “Pile” sera codé 1.
- “Face” sera codé par 0.
- Soit *h* la probabilité d’obtenir “pile”.
- Soit1 - *h* la probabilité de “face”.
- Soit *x* le résultat d’un lancé.

$$p(x = 1) = h$$

$$p(x = 0) = 1 – h$$

Si “pile” (x=1): $p(x \mid h) = h^x(1 - h)^{1 - x}$

Inférence ML

- “Pile” sera codé 1.
- “Face” sera codé par 0.
- Soit *h* la probabilité d’obtenir “pile”.
- Soit1 - *h* la probabilité de “face”.
- Soit *x* le résultat d’un lancé.

$$p(x = 1) = h$$

$$p(x = 0) = 1 – h$$

Si “pile” (x=1): $p(x \mid h) = h^x \cancel{(1 - h)^{1 - x}}$

Inférence ML

- “Pile” sera codé 1.
- “Face” sera codé par 0.
- Soit *h* la probabilité d’obtenir “pile”.
- Soit1 - *h* la probabilité de “face”.
- Soit *x* le résultat d’un lancé.

$$p(x = 1) = h$$

$$p(x = 0) = 1 - h$$

Si “face” (x=0) $\quad p(x \mid h) = h^x(1 - h)^{1 - x}$

Inférence ML

- “Pile” sera codé 1.
- “Face” sera codé par 0.
- Soit *h* la probabilité d’obtenir “pile”.
- Soit1 - *h* la probabilité de “face”.
- Soit *x* le résultat d’un lancé.

$$p(x = 1) = h$$

$$p(x = 0) = 1 - h$$

Si “face” (x=0) $\quad p(x \mid h) = \cancel{h^x}(1 - h)^{1 - x}$

Inférence ML

- “Pile” sera codé 1.
- “Face” sera codé par 0.
- Soit *h* la probabilité d’obtenir “pile”.
- Soit1 - *h* la probabilité de “face”.
- Soit *x* le résultat d’un lancé.

$$p(x = 1) = h$$

$$p(x = 0) = 1 – h$$

$$p(x \mid h) = h^x(1 - h)^{1 - x}$$

Inférence ML

- Soit $x = x_1, x_2, \ldots, x_n$ une liste de n résultats.
- $L(h \mid x) = p(x \mid h)$ est la fonction de vraisemblance.

$$L(h \mid x) = p(x \mid h) = \Pi_i \, h^{x_i}(1 - h)^{1 - x_i}$$

Inférence ML

- Soit $x = x_1, x_2, \ldots, x_n$ une liste de n résultats.
- $L(h \mid x) = p(x \mid h)$ est la fonction de vraisemblance.

$$L(h \mid x) = p(x \mid h) = \Pi_i \, h^{x_i}(1 - h)^{1 - x_i}$$

$$= h^{\Sigma_i x_i} \, (1 - h)^{\Sigma_i (1 - x_i)}$$

Inférence ML

- Soit $x = x_1, x_2, \ldots, x_n$ une liste de n résultats.
- $L(h \mid x) = p(x \mid h)$ est la fonction de vraisemblance.

$$L(h \mid x) = p(x \mid h) = \Pi_i \, h^{x_i}(1 - h)^{1 - x_i}$$

$$= h^{\Sigma_i x_i} \, (1 - h)^{\Sigma_i (1 - x_i)}$$

$$= h^{\Sigma_i x_i} \, (1 - h)^{n - \Sigma_i x_i}$$

Inférence ML

- L'inférence ML de *h* est la valeur qui maximise la fonction de vraisemblance.
- En pratique, nous travaillons souvent avec le logarithme de la fonction de vraisemblance.
- Vraisemblance:

$$L(h \mid x) = p(x \mid h) = h^{\Sigma_i x_i} (1 - h)^{n - \Sigma_i x_i}$$

- Log naturel de la vraisemblence:

$$\ln L = \Sigma_i x_i \ln h + (n - \Sigma_i x_i) \ln(1 - h)$$

Inférence ML

- Prennez la dérivée du log vraisemblance:

$$\ln L = \Sigma_i x_i \ln h + (n - \Sigma_i x_i)\ln(1 - h)$$

$$\partial \ln L / \partial h = h^{-1} \Sigma_i x_i - (1 - h)^{-1} (n - \Sigma_i x_i)$$

- Forcez l'égalité à zéro et isolez h:

$$h = 1/n \, \Sigma_i x_i$$

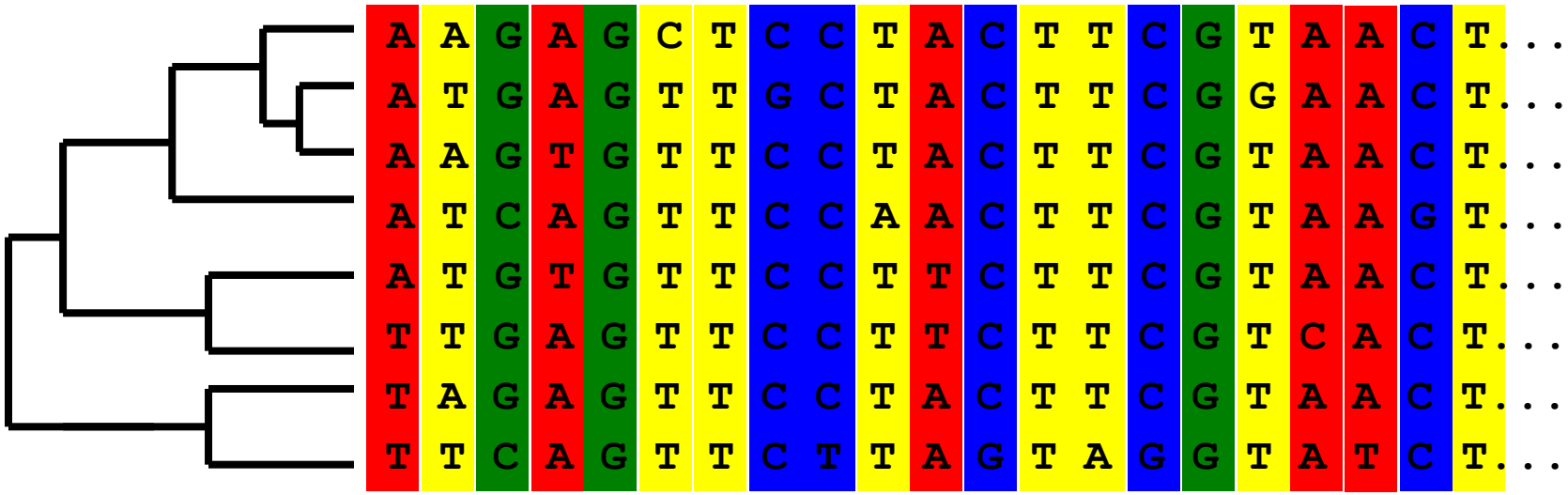
- Vérifez que la seconde dérivée est négative.

Comprendre l’inférence ML

- Simuler un jeu de pile-ou-face.
- Inférer le parametre d’un jeu pile-ou-face.

Modéliser l'évolution moléculaire

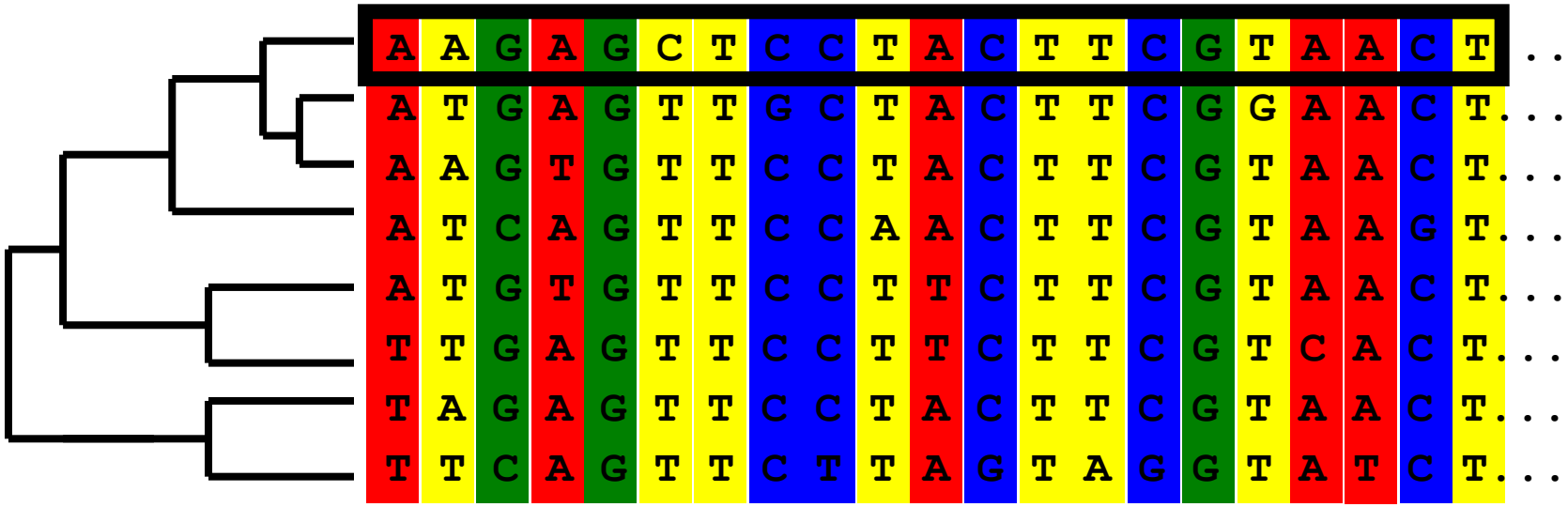
…par une chaîne de Markov en temps continu



…ayant les 4 nucléotides comme états.

Modéliser l'évolution moléculaire

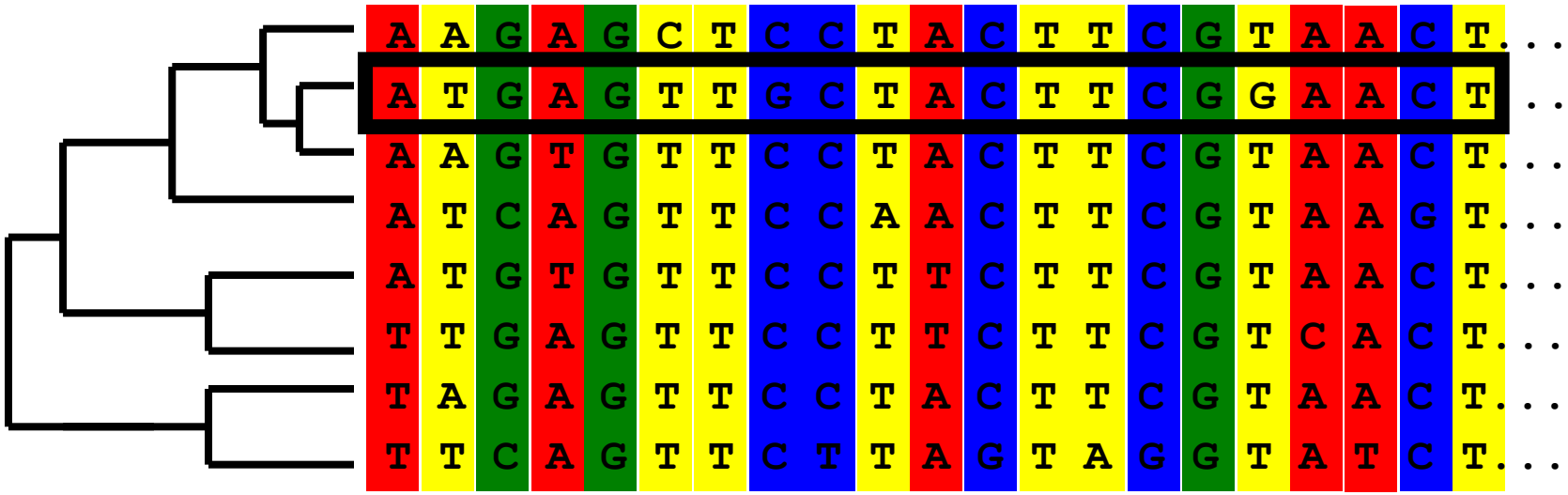
…par une chaîne de Markov en temps continu



…ayant les 4 nucléotides comme états.

Modéliser l'évolution moléculaire

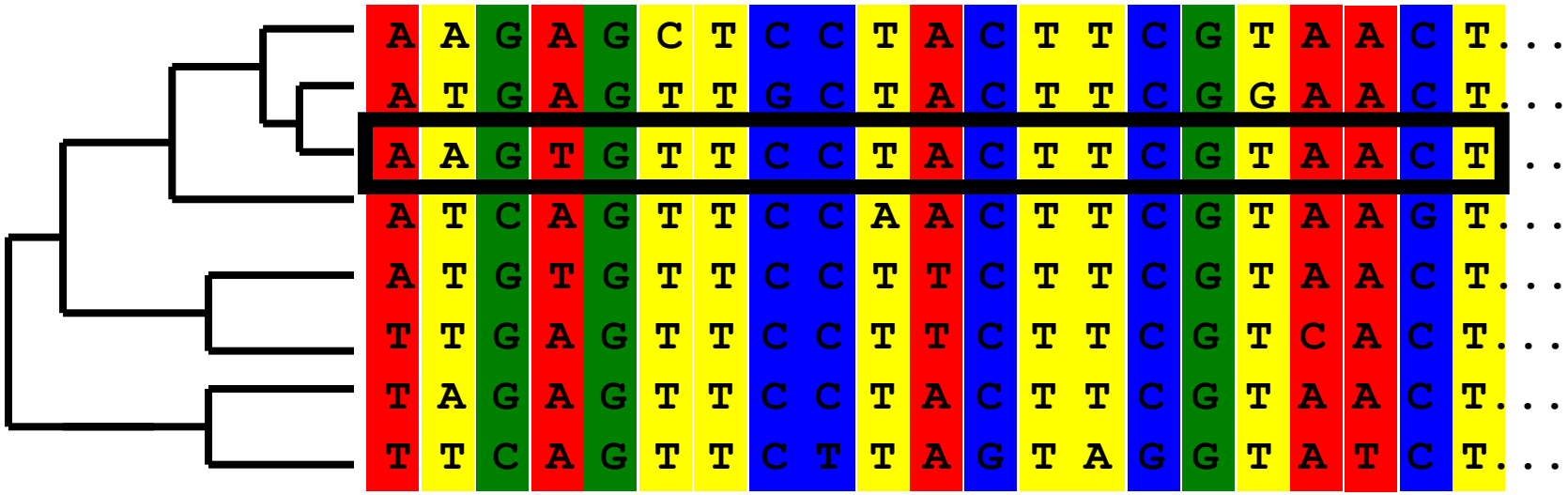
…par une chaîne de Markov en temps continu



…ayant les 4 nucléotides comme états.

Modéliser l'évolution moléculaire

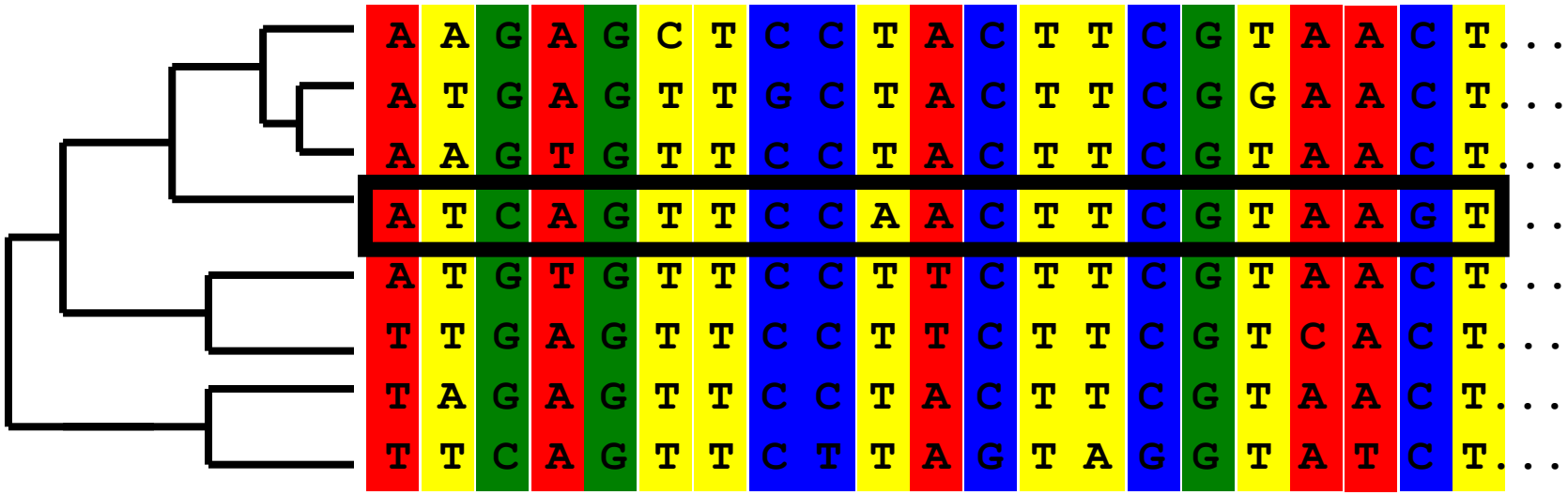
…par une chaîne de Markov en temps continu



…ayant les 4 nucléotides comme états.

Modéliser l'évolution moléculaire

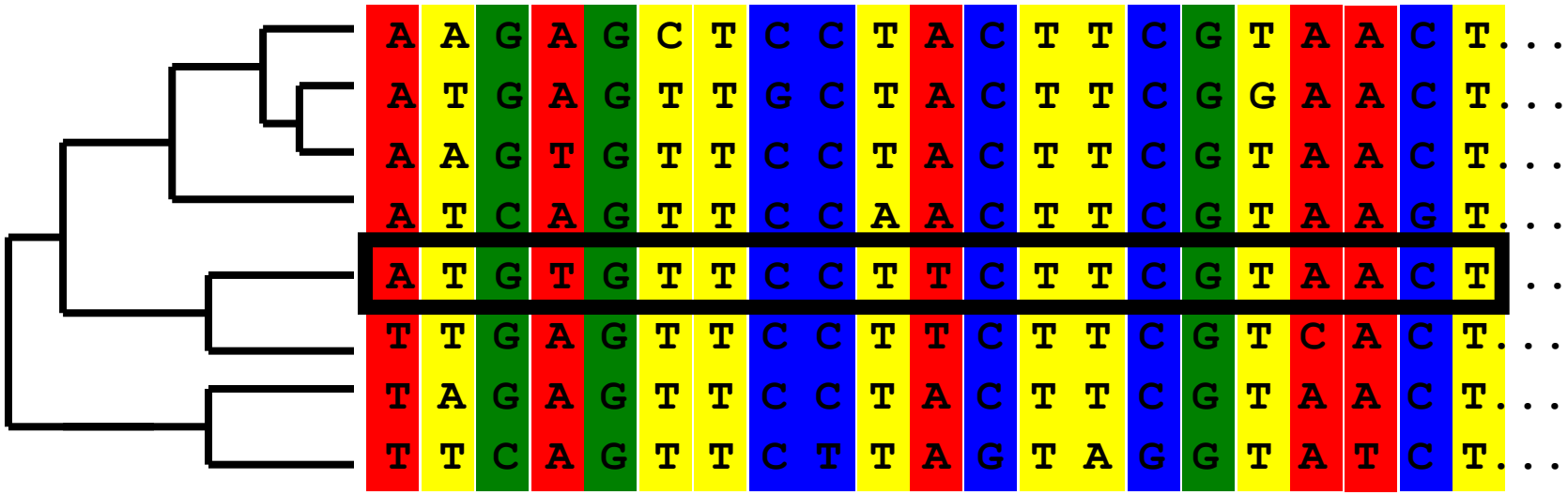
…par une chaîne de Markov en temps continu



…ayant les 4 nucléotides comme états.

Modéliser l'évolution moléculaire

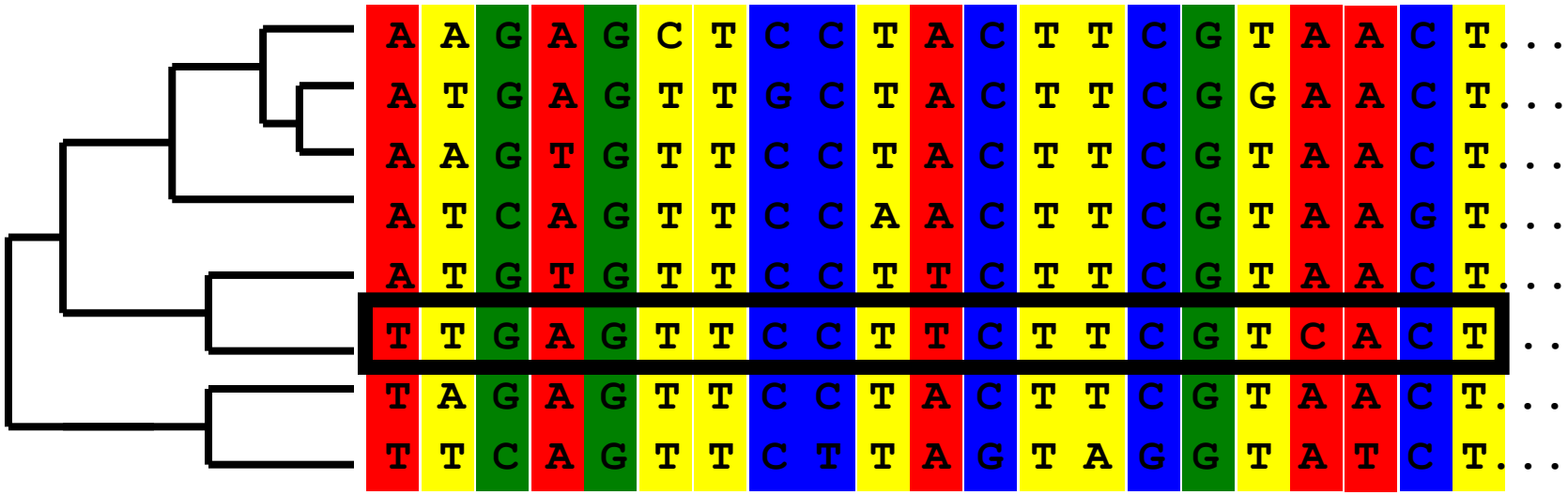
…par une chaîne de Markov en temps continu



…ayant les 4 nucléotides comme états.

Modéliser l'évolution moléculaire

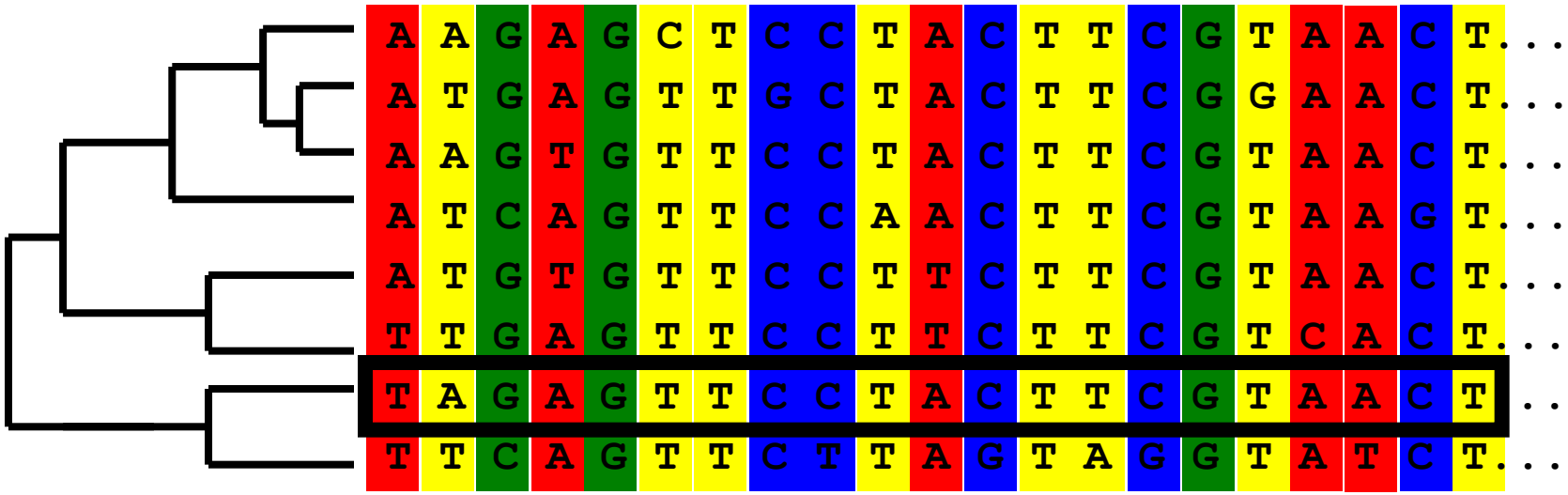
…par une chaîne de Markov en temps continu



…ayant les 4 nucléotides comme états.

Modéliser l'évolution moléculaire

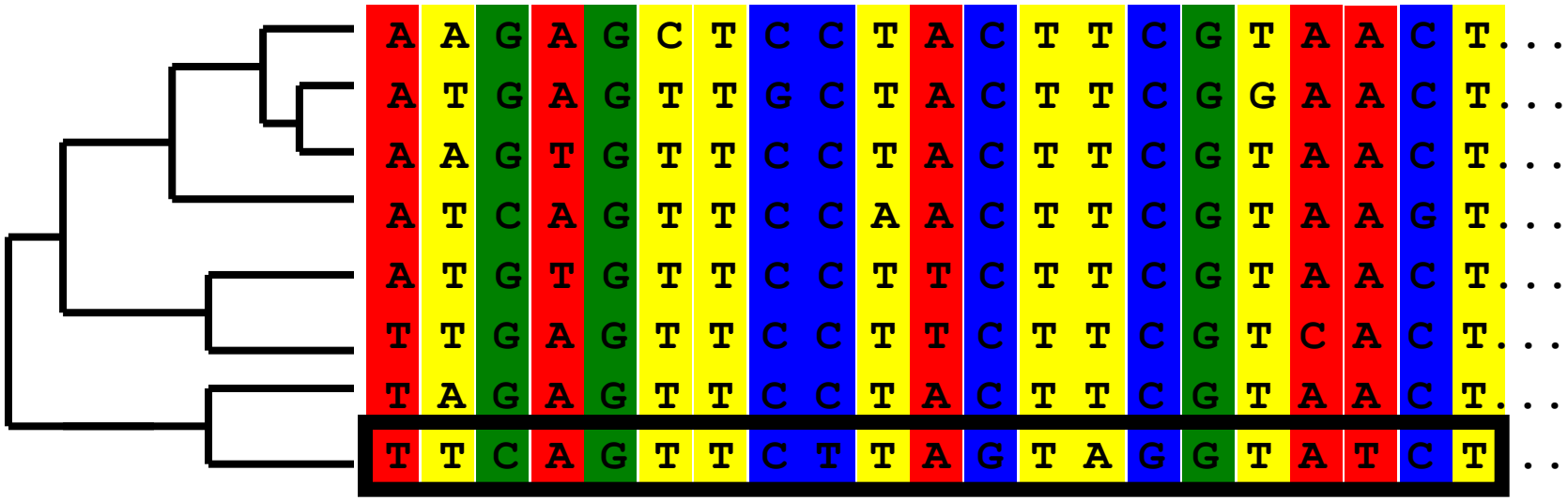
…par une chaîne de Markov en temps continu



…ayant les 4 nucléotides comme états.

Modéliser l'évolution moléculaire

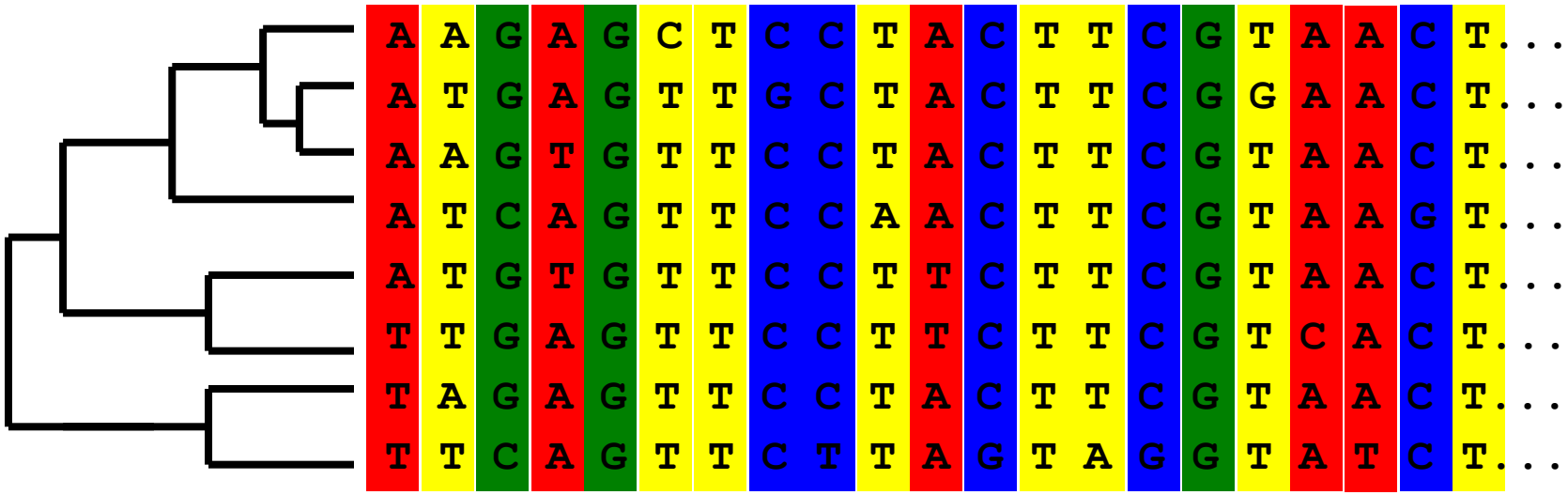
…par une chaîne de Markov en temps continu



…ayant les 4 nucléotides comme états.

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu



…ayant les 4 nucléotides comme états.

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

$$Q = [Q_{ab}]$$

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

$$Q = [Q_{ab}]$$

- Infinitesimal rate matrix
- Markov generator
- Infinitesimal generator
- Markov rate matrix
- …

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

$$Q_{ab} \propto \rho_{ab}\varphi_b, a \neq b$$

$$Q_{aa} = -\sum_{b \neq a} Q_{ab}$$

Modéliser l'évolution moléculaire

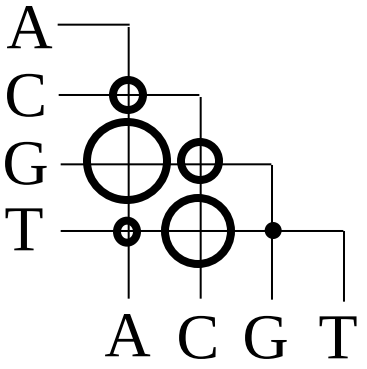
…par une chaîne de Markov en temps continu

$$Q_{ab} \propto \rho_{ab}\varphi_b, a \neq b$$

$$Q_{aa} = -\sum_{b \neq a} Q_{ab}$$

$$\rho = \left(\rho_{ab}\right)_{1 \leq a,b \leq 4}$$

L'interchangeabilité entre les nucléotides *a* et *b*.

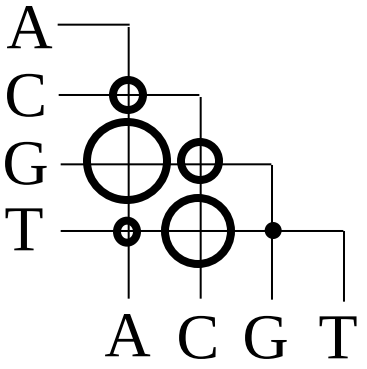


Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

$$Q_{ab} \propto \rho_{ab}\varphi_b, a \neq b$$

$$Q_{aa} = -\sum_{b \neq a} Q_{ab}$$



$\rho = (\rho_{ab})_{1 \leq a,b \leq 4}$ L'interchangeabilité entre les nucléotides *a* et *b*.

$\varphi = (\varphi_a)_{1 \leq a \leq 4}$ La fréquence relative du nucléotide *a*.

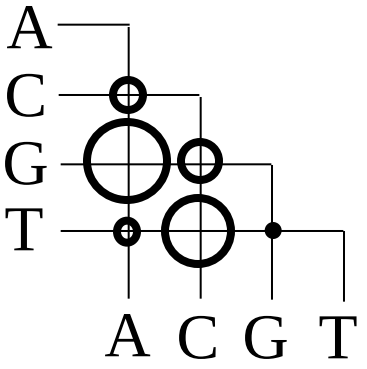
Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

GTR

$$Q_{ab} \propto \rho_{ab}\varphi_b, a \neq b$$

$$Q_{aa} = -\sum_{b \neq a} Q_{ab}$$



$\rho = (\rho_{ab})_{1 \leq a,b \leq 4}$ L'interchangeabilité entre les nucléotides *a* et *b*.

$\varphi = (\varphi_a)_{1 \leq a \leq 4}$ La fréquence relative du nucléotide *a*.

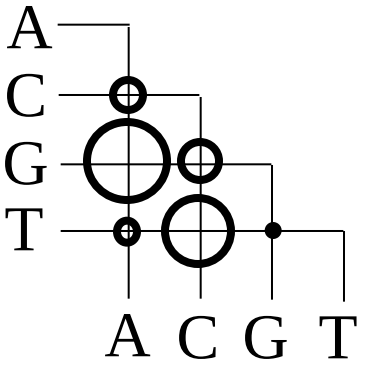
Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

GTR
HKY

$$Q_{ab} \propto \rho_{ab}\varphi_b, a \neq b$$

$$Q_{aa} = -\sum_{b \neq a} Q_{ab}$$



$\rho = (\rho_{ab})_{1 \leq a,b \leq 4}$ L'interchangeabilité entre les nucléotides *a* et *b*.

$\varphi = (\varphi_a)_{1 \leq a \leq 4}$ La fréquence relative du nucléotide *a*.

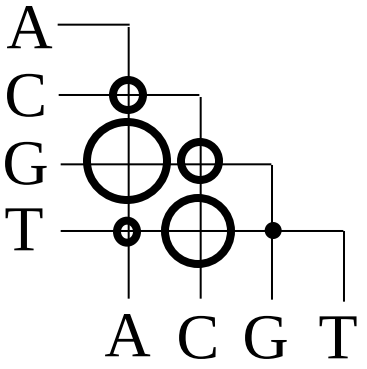
Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

GTR
HKY
F81

$$Q_{ab} \propto \rho_{ab}\varphi_b, a \neq b$$

$$Q_{aa} = -\sum_{b \neq a} Q_{ab}$$



$\rho = (\rho_{ab})_{1 \leq a,b \leq 4}$ L'interchangeabilité entre les nucléotides *a* et *b*.

$\varphi = (\varphi_a)_{1 \leq a \leq 4}$ La fréquence relative du nucléotide *a*.

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

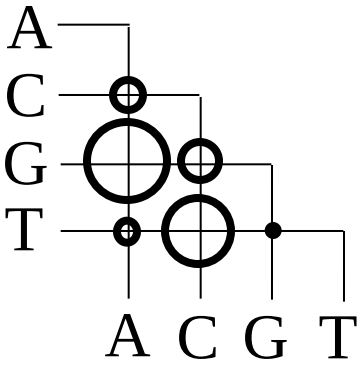
GTR
HKY
F81
JC

$$Q_{ab} \propto \rho_{ab}\varphi_b, a \neq b$$

$$Q_{aa} = -\sum_{b \neq a} Q_{ab}$$

$$\rho = (\rho_{ab})_{1 \leq a,b \leq 4}$$

L'interchangeabilité entre les nucléotides *a* et *b*.



$$\varphi = (\varphi_a)_{1 \leq a \leq 4}$$

La fréquence relative du nucléotide *a*.

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

Longueur de branche (λ)

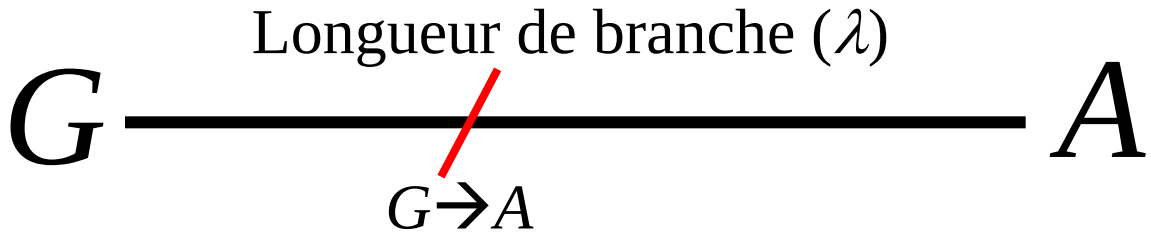
$$G \text{———————————} A$$

$$p(A \mid G, \theta) = \int_{\Phi} p(A, \phi \mid G, \theta) d\phi = \left[e^{\lambda Q} \right]_{GA}$$

Sachant l'état de départ (G) et les paramètres du processus de Markov (ρ, φ), la probabilité d'aboutir en un autre état (A) sur une distance évolutive donnée (λ, i.e., $\theta = \{\rho, \varphi, \lambda\}$) s'exprime par l'intégrale sur l'ensemble des histoires substitutionnelles (ϕ), et peut être calculée en exploitant des algorithmes de diagonalisation de matrices.

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

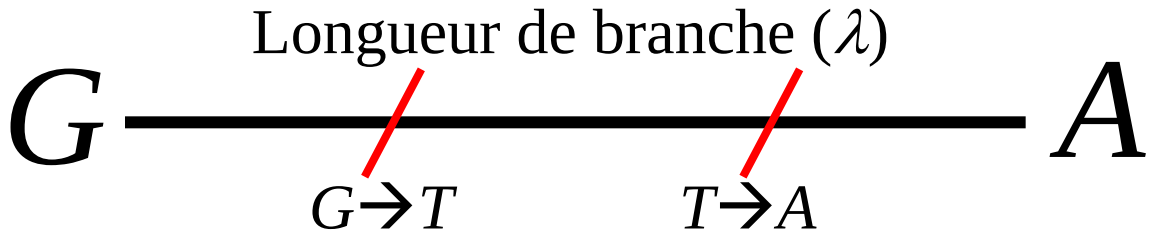


$$p(A \mid G, \theta) = \int_{\Phi} p(A, \phi \mid G, \theta) d\phi = \left[e^{\lambda Q} \right]_{GA}$$

Sachant l'état de départ (G) et les paramètres du processus de Markov (ρ, φ), la probabilité d'aboutir en un autre état (A) sur une distance évolutive donnée (λ, i.e., $\theta = \{\rho, \varphi, \lambda\}$) s'exprime par l'intégrale sur l'ensemble des histoires substitutionnelles (ϕ), et peut être calculée en exploitant des algorithmes de diagonalisation de matrices.

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

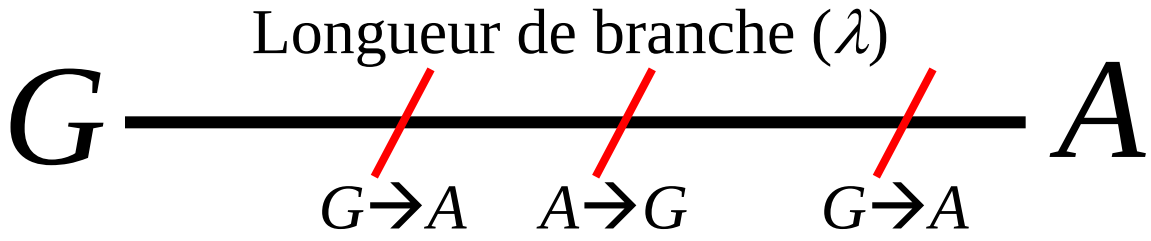


$$p(A \mid G, \theta) = \int_{\Phi} p(A, \phi \mid G, \theta) d\phi = \left[e^{\lambda Q} \right]_{GA}$$

Sachant l'état de départ (G) et les paramètres du processus de Markov (ρ, φ), la probabilité d'aboutir en un autre état (A) sur une distance évolutive donnée (λ, i.e., $\theta = \{\rho, \varphi, \lambda\}$) s'exprime par l'intégrale sur l'ensemble des histoires substitutionnelles (ϕ), et peut être calculée en exploitant des algorithmes de diagonalisation de matrices.

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu



$$p(A \mid G, \theta) = \int_{\Phi} p(A, \phi \mid G, \theta) d\phi = \left[e^{\lambda Q} \right]_{GA}$$

Sachant l'état de départ (G) et les paramètres du processus de Markov (ρ, φ), la probabilité d'aboutir en un autre état (A) sur une distance évolutive donnée (λ, i.e., $\theta = \{\rho, \varphi, \lambda\}$) s'exprime par l'intégrale sur l'ensemble des histoires substitutionnelles (ϕ), et peut être calculée en exploitant des algorithmes de diagonalisation de matrices.

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

Longueur de branche (λ)

$$G \text{———————————} A$$

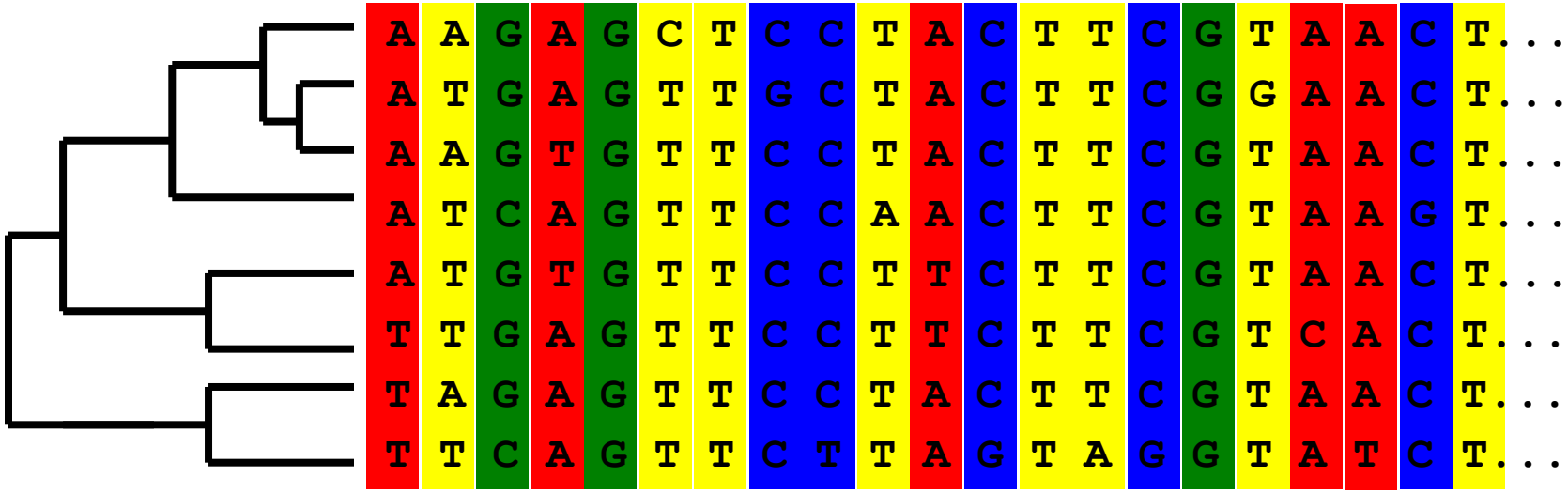
$$p(A \mid G, \theta) = \int_{\Phi} p(A, \phi \mid G, \theta) d\phi = \left[e^{\lambda Q} \right]_{GA}$$

Sachant l'état de départ (G) et les paramètres du processus de Markov (ρ, φ), la probabilité d'aboutir en un autre état (A) sur une distance évolutive donnée (λ, i.e., $\theta = \{\rho, \varphi, \lambda\}$) s'exprime par l'intégrale sur l'ensemble des histoires substitutionnelles (ϕ), et peut être calculée en exploitant des algorithmes de diagonalisation de matrices.

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

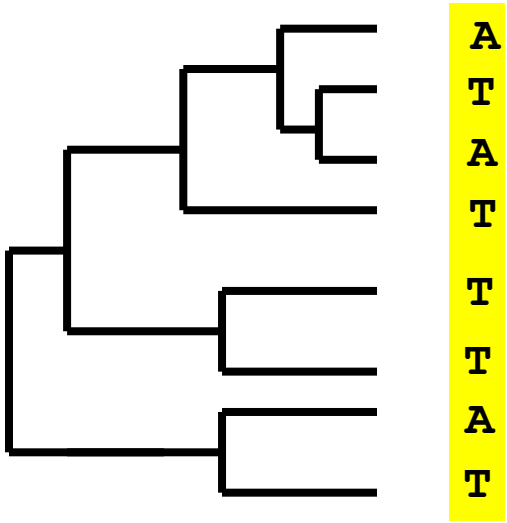
Felsenstein, *J. Mol. Evol.*, 17:368-376, 1981.



…ayant les 4 nucléotides comme états.

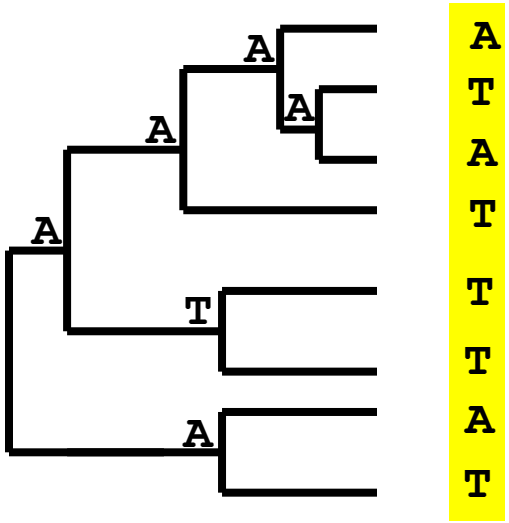
Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu



Modéliser l'évolution moléculaire

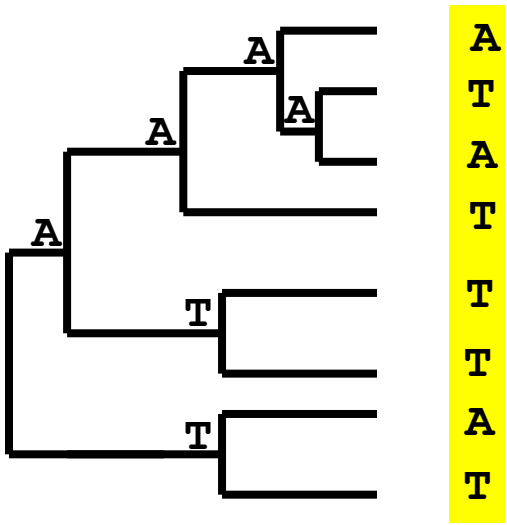
…par une chaîne de Markov en temps continu



Modéliser l'évolution moléculaire

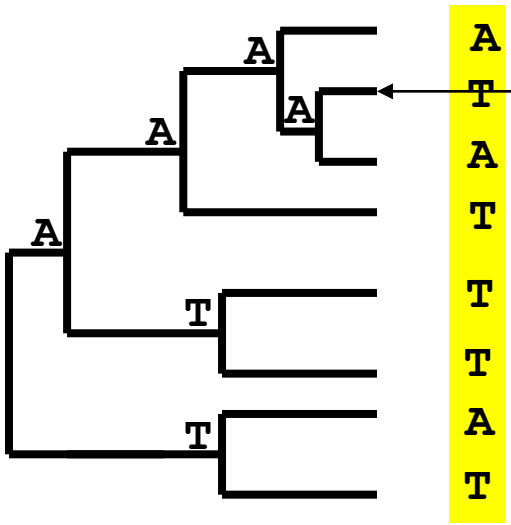
…par une chaîne de Markov en temps continu

$$p(A \mid A, \theta)$$



Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

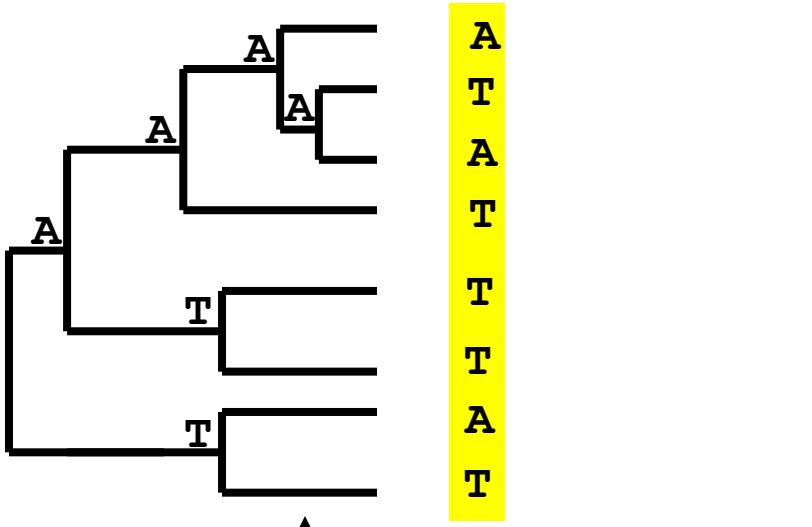


$$p(A \mid A, \theta) \times$$

$$p(T \mid A, \theta) \times$$

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu



$$p(A \mid A, \theta) \times$$

$$p(T \mid A, \theta) \times$$

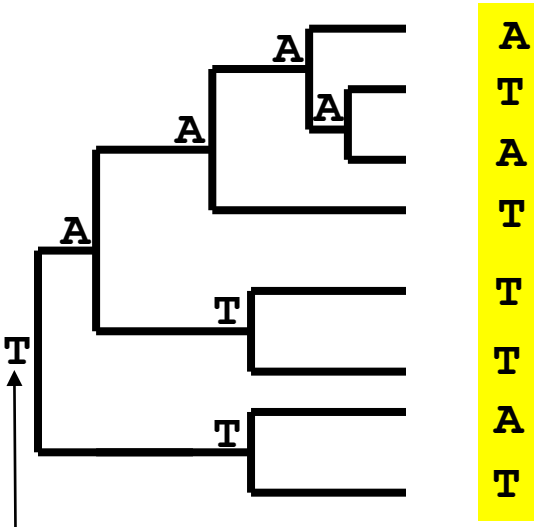
$$p(A \mid A, \theta) \times$$

$$\vdots$$

$$p(T \mid T, \theta) \times$$

Modéliser l'évolution moléculaire

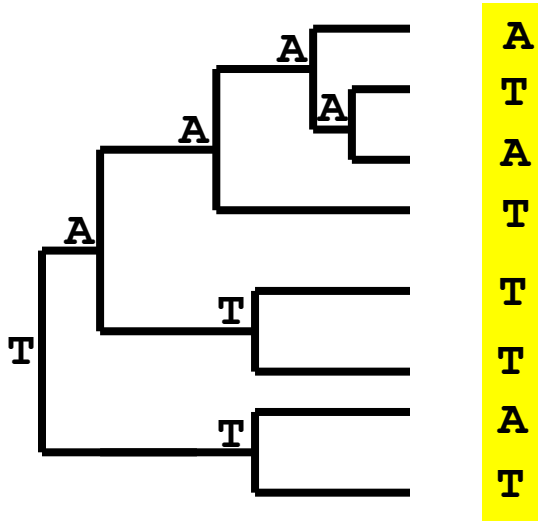
…par une chaîne de Markov en temps continu



$$p(A \mid A, \theta) \times$$
$$p(T \mid A, \theta) \times$$
$$p(A \mid A, \theta) \times$$
$$\vdots$$
$$p(T \mid T, \theta) \times$$
$$p(T \mid \theta)$$

Modéliser l'évolution moléculaire

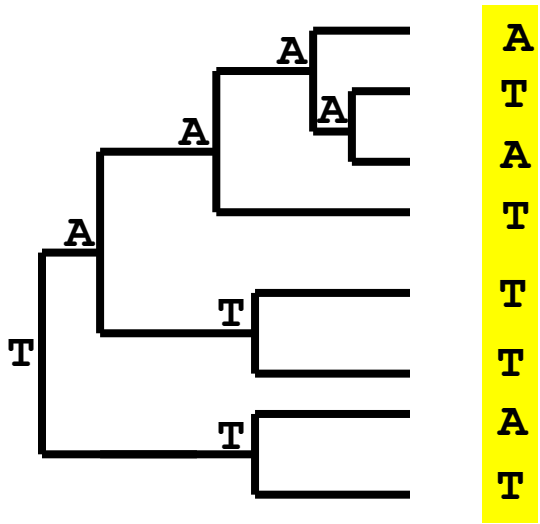
…par une chaîne de Markov en temps continu



$$p(A \mid A, \theta) \times$$
$$p(T \mid A, \theta) \times$$
$$p(A \mid A, \theta) \times$$
$$\vdots$$
$$p(T \mid T, \theta) \times$$
$$p(T \mid \theta)$$

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

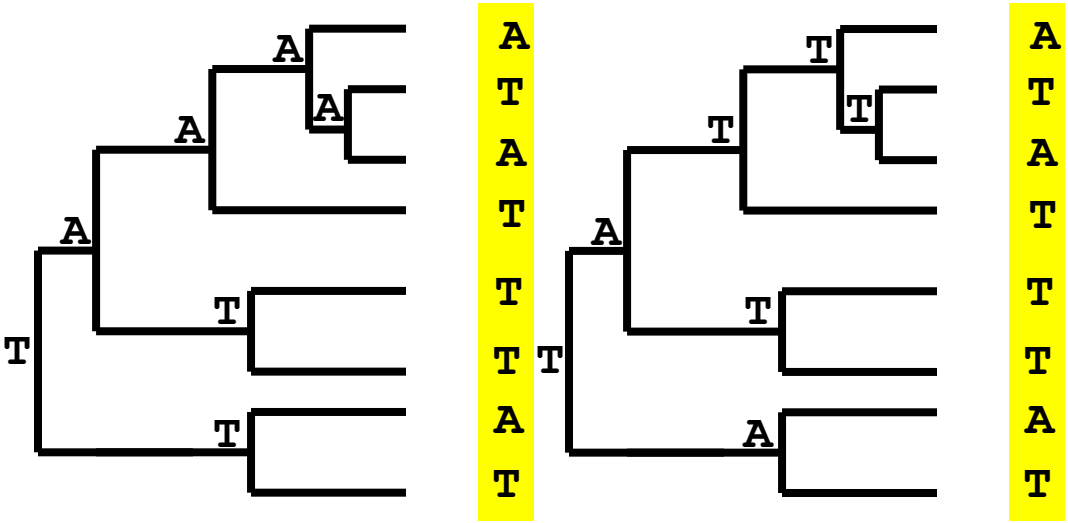


$$p(A \mid A, \theta) \times$$
$$p(T \mid A, \theta) \times$$
$$p(A \mid A, \theta) \times$$
$$\vdots$$
$$p(T \mid T, \theta) \times$$
$$p(T \mid \theta)$$

Il faut prendre la somme d'un tel calcul pour toutes les configurations d'états possibles aux nœuds internes de l'arbre.

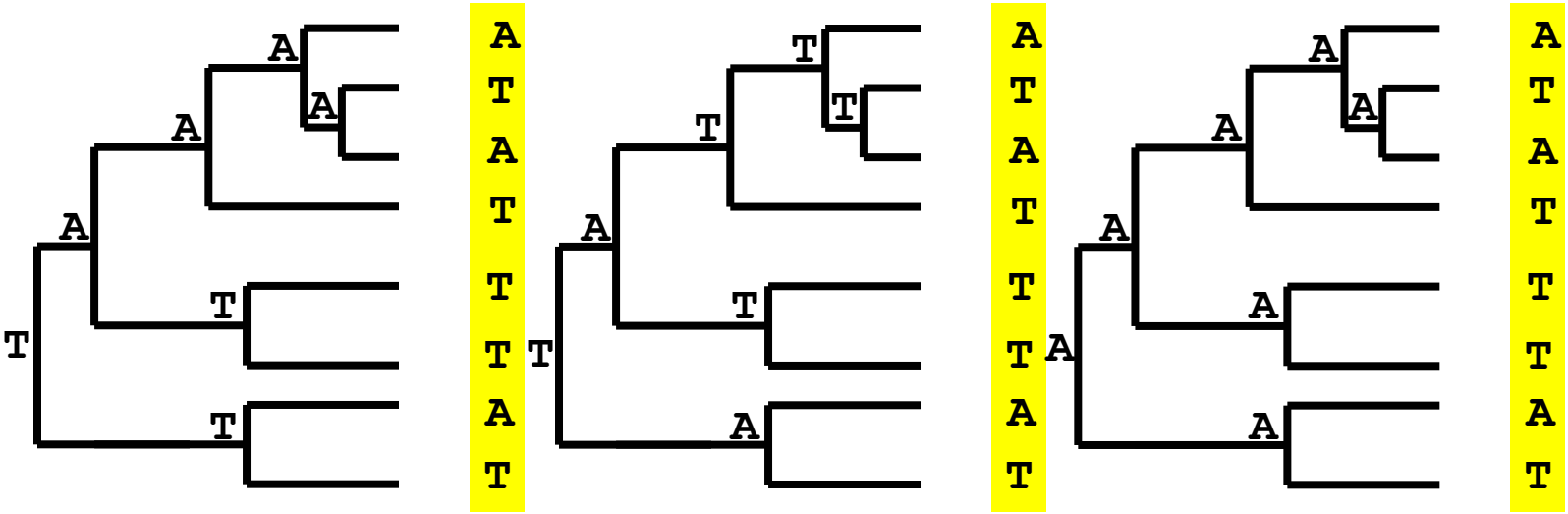
Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu



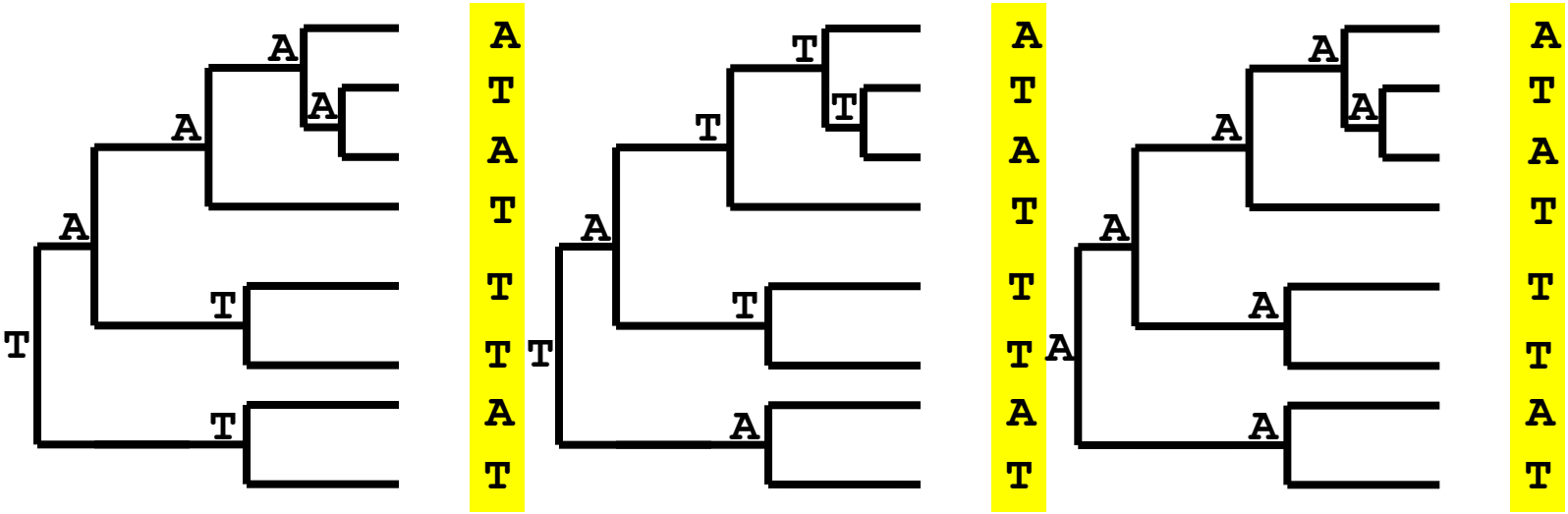
Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu



Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

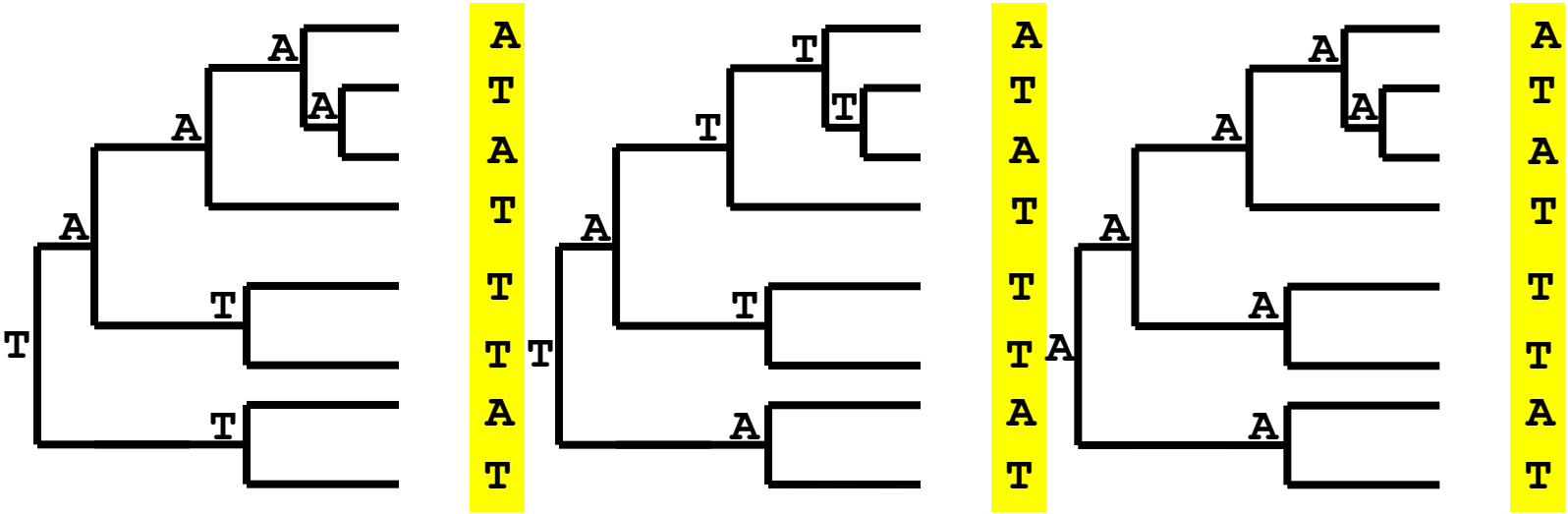


Il y a 4^{P-3} configurations d'états aux nœuds internes de l'arbre.

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

Felsenstein, J. Mol. Evol., 17:368-376, 1981.

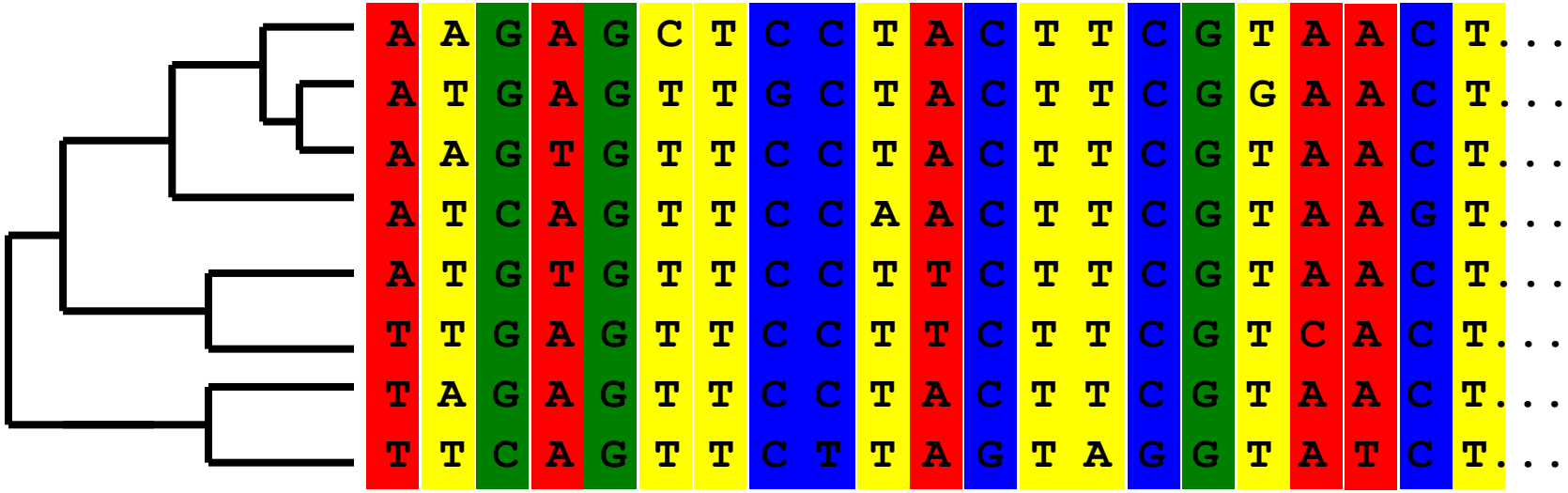


Il y a 4^{P-3} configurations d'états aux noeuds interns de l'arbre.

Modéliser l'évolution moléculaire

…par une chaîne de Markov en temps continu

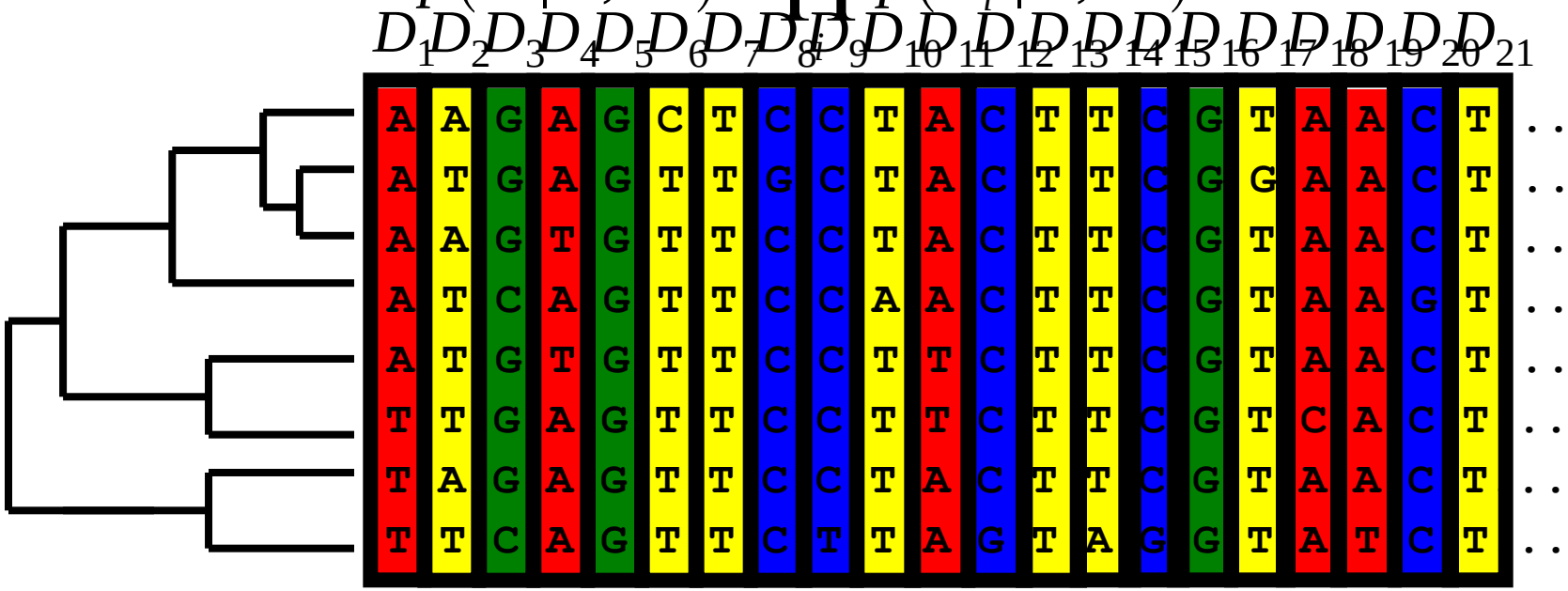
Felsenstein, J. Mol. Evol., 17:368-376, 1981.



Modéliser l'évolution moléculaire

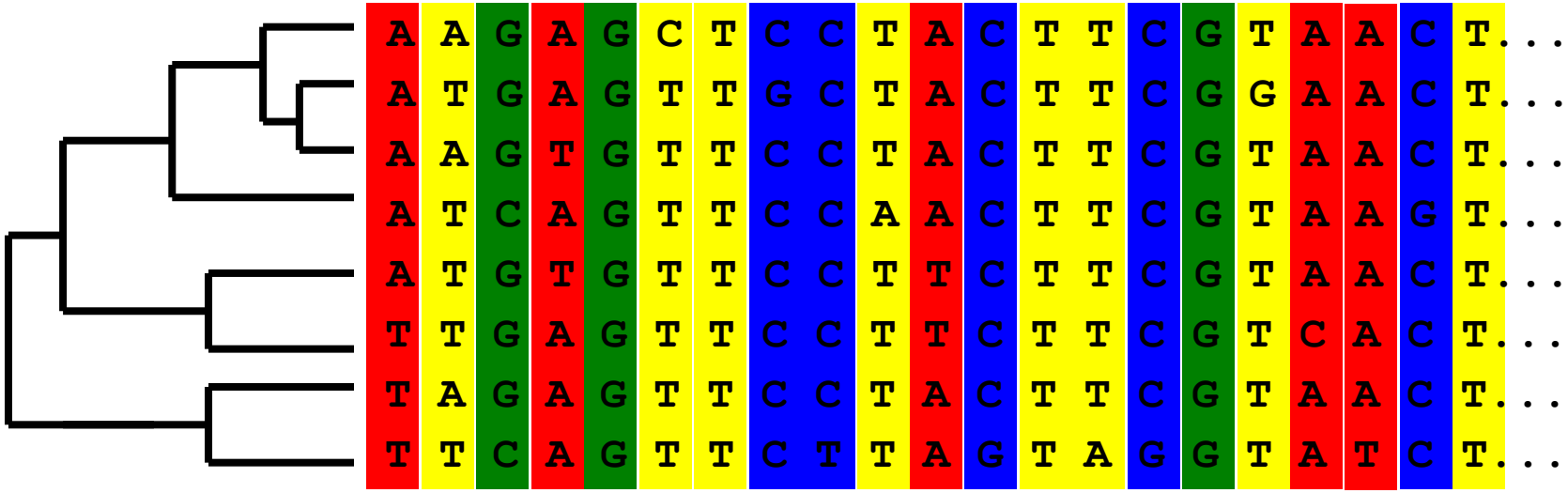
…par une chaîne de Markov en temps continu

$$p(D \mid \theta, M) = \prod_i p(D_i \mid \theta, M)$$

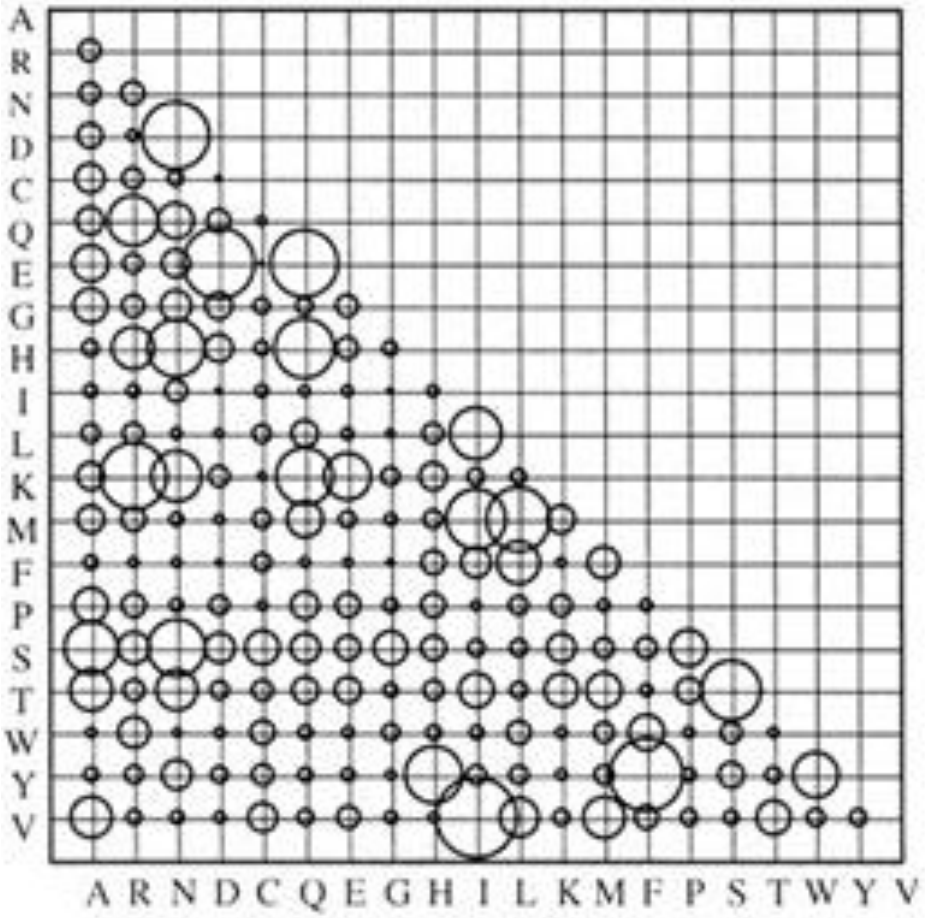


Sélection de modèle

Quel modèle de substitution devrions nous utiliser? Passons à la pratique…



Modéliser l'évolution moléculaire



Le paradigme Bayésien

$$p(\theta \mid D, M) = \frac{p(D \mid \theta, M)\, p(\theta \mid M)}{p(D \mid M)}$$

Le paradigme Bayésien

$$\underbrace{p(\theta \mid D, M)}_{\text{posterior}} = \frac{\overbrace{p(D \mid \theta, M)}^{\text{likelihood function}}\,\overbrace{p(\theta \mid M)}^{\text{prior}}}{\underbrace{p(D \mid M)}_{\text{marginal likelihood}}}$$

D	Data set
θ	Parameter vector
M	Model

Le paradigme Bayésien

$$\underbrace{p(\theta \mid D, M)}_{\text{postérieure}} = \frac{\overbrace{p(D \mid \theta, M)}^{\text{fonction de vraisemblance}}\,\overbrace{p(\theta \mid M)}^{\text{a priori}}}{\underbrace{p(D \mid M)}_{\text{vraisemblance marginale}}}$$

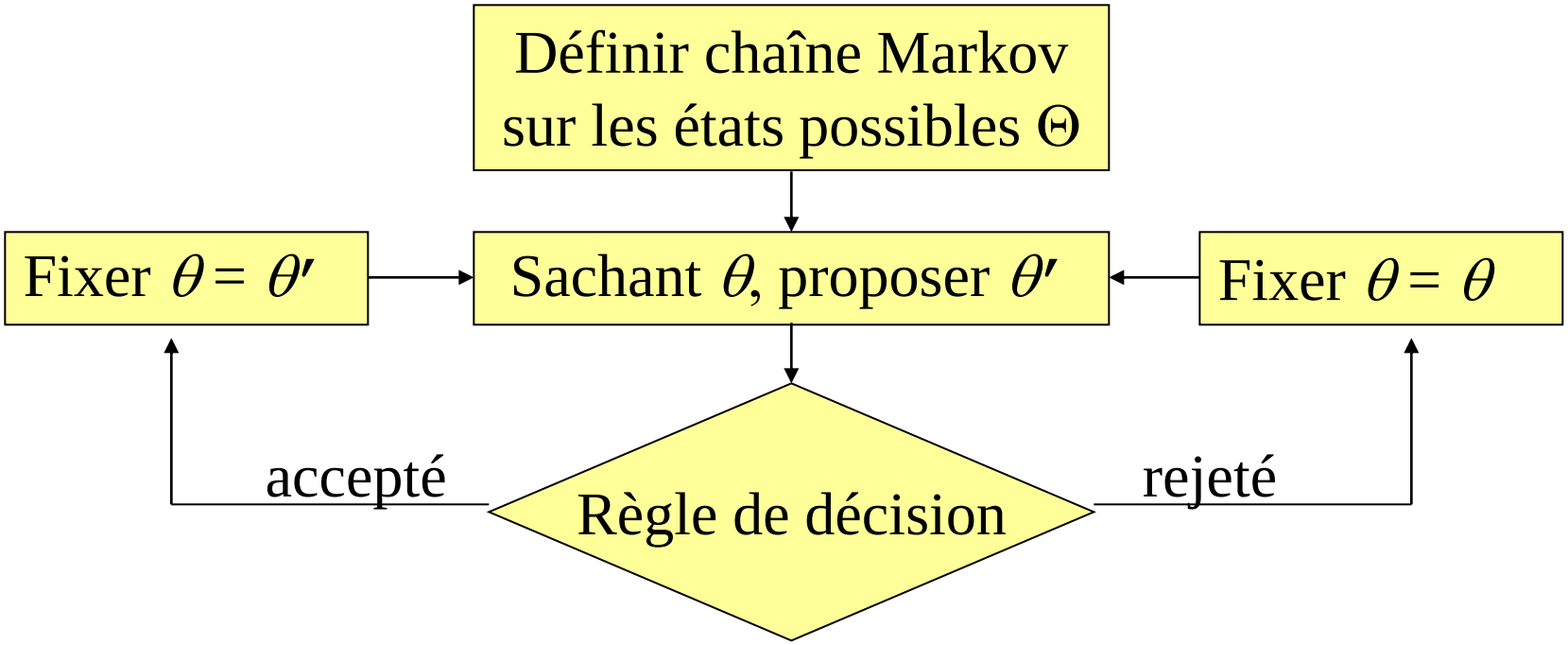
D	Les données
θ	Le vecteur de paramètres
M	Modèle

Le paradigme Bayésien

$$\underbrace{p(\theta \mid D, M)}_{\text{postérieure}} = \frac{\overbrace{p(D \mid \theta, M)}^{\text{fonction de vraisemblance}}\,\overbrace{p(\theta \mid M)}^{\text{a priori}}}{\underbrace{p(D \mid M)}_{\text{vraisemblance marginale}}}$$

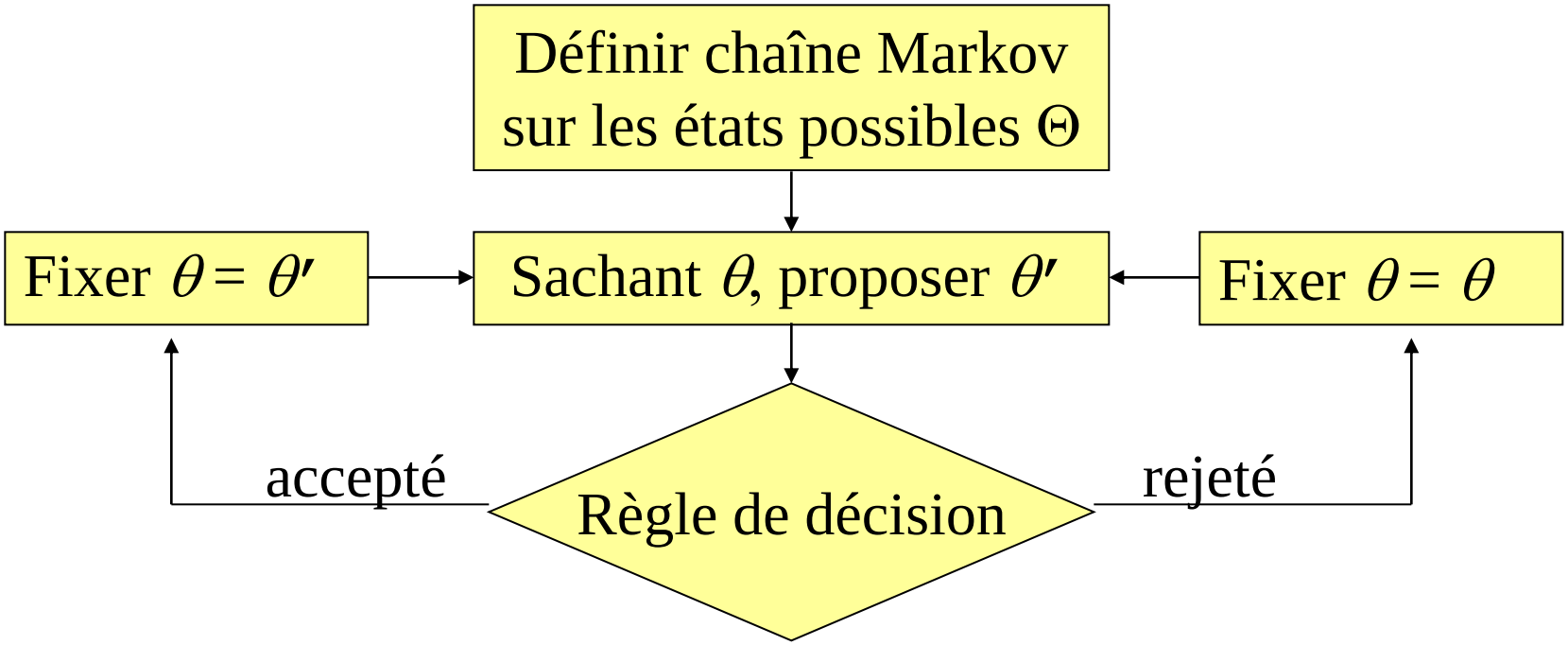
Obtenir un échantillon de la distribution postérieure en utilisant des algorithmes de chaînes de Markov Monte Carlo.

Le paradigme Bayésien

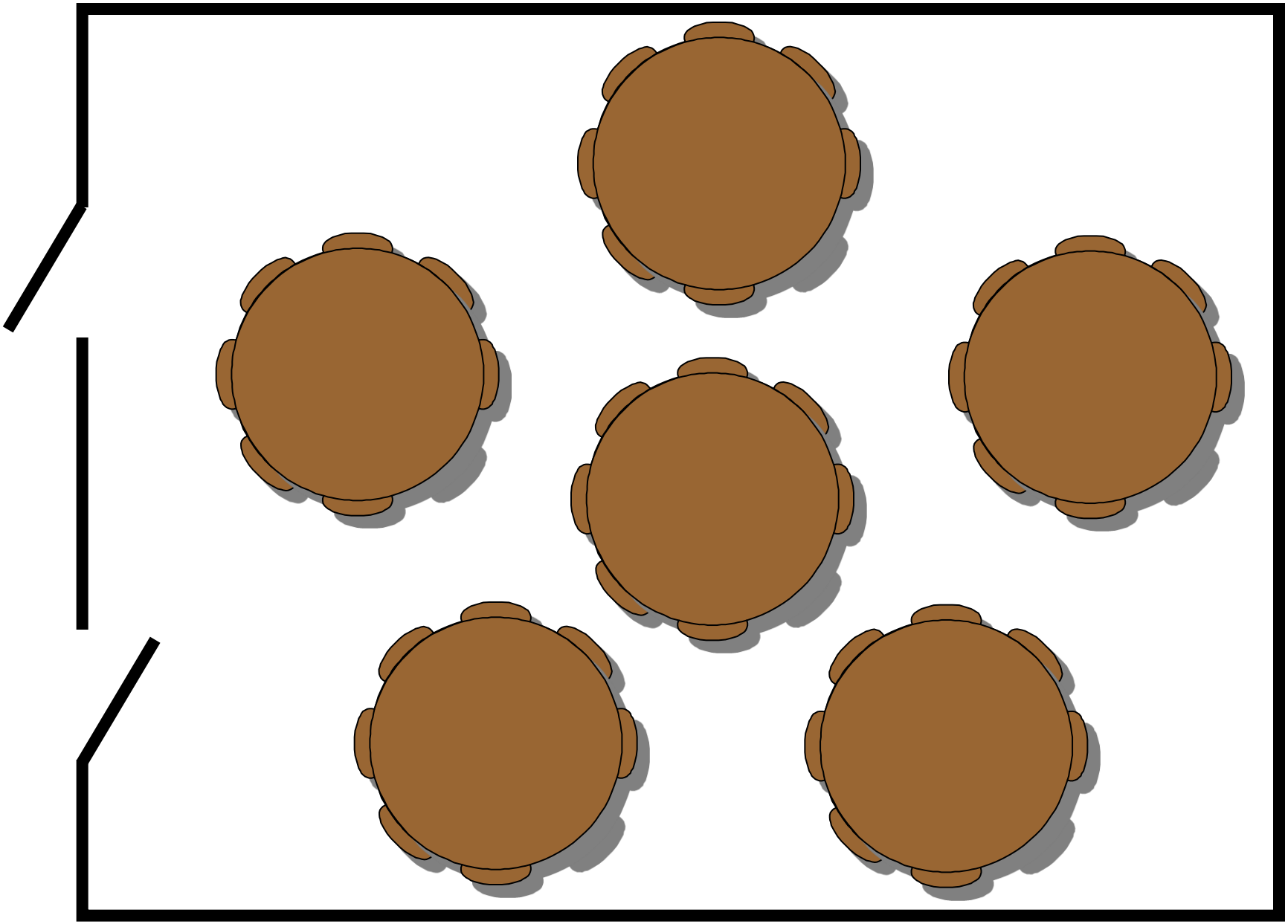


Accepté avec probabilité $a = \min\left\{1, \dfrac{p(\theta' \mid D, M)\, q(\theta, \theta')}{p(\theta \mid D, M)\, q(\theta', \theta)}\right\}$

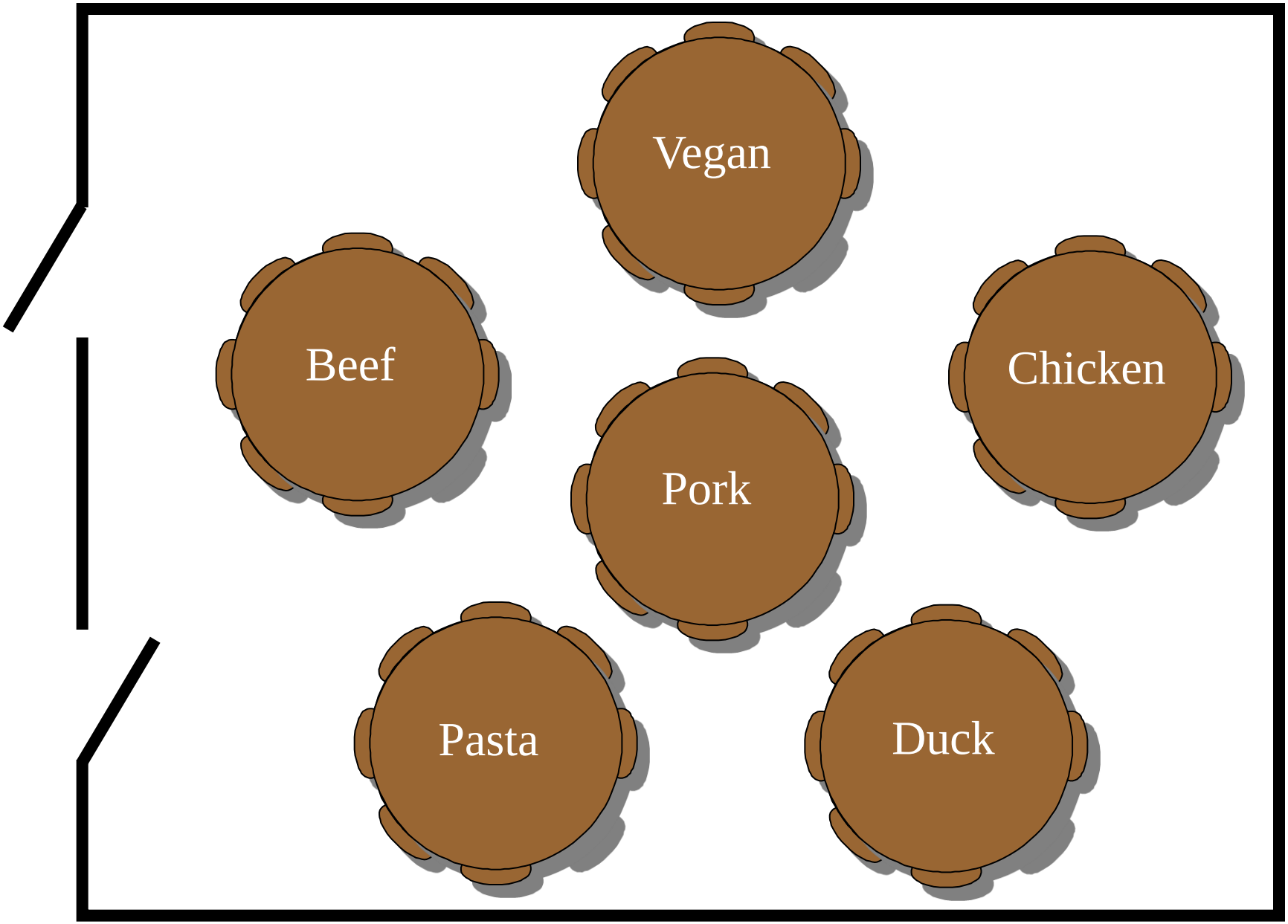
Le paradigme Bayésien



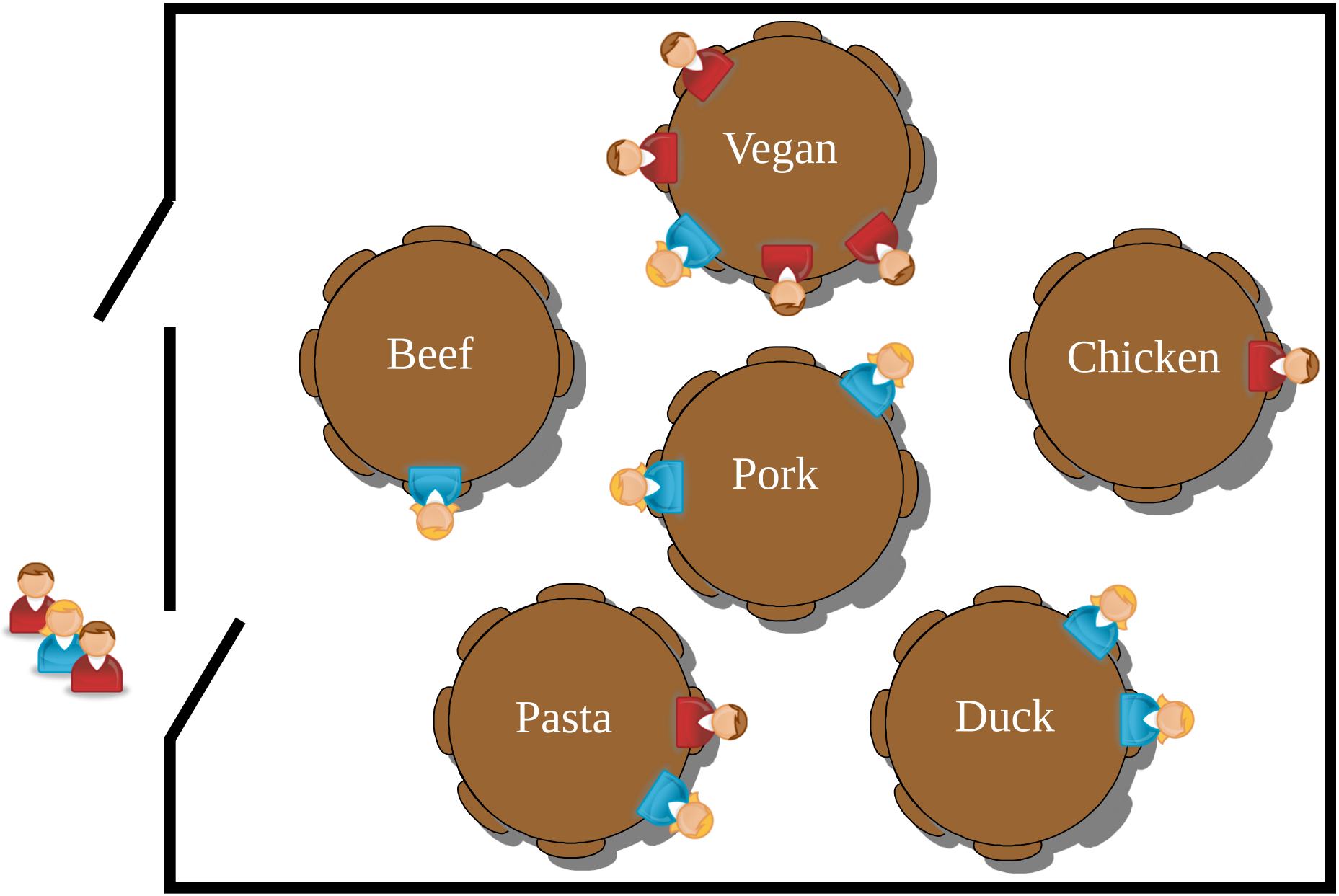
Le processus Dirichlet



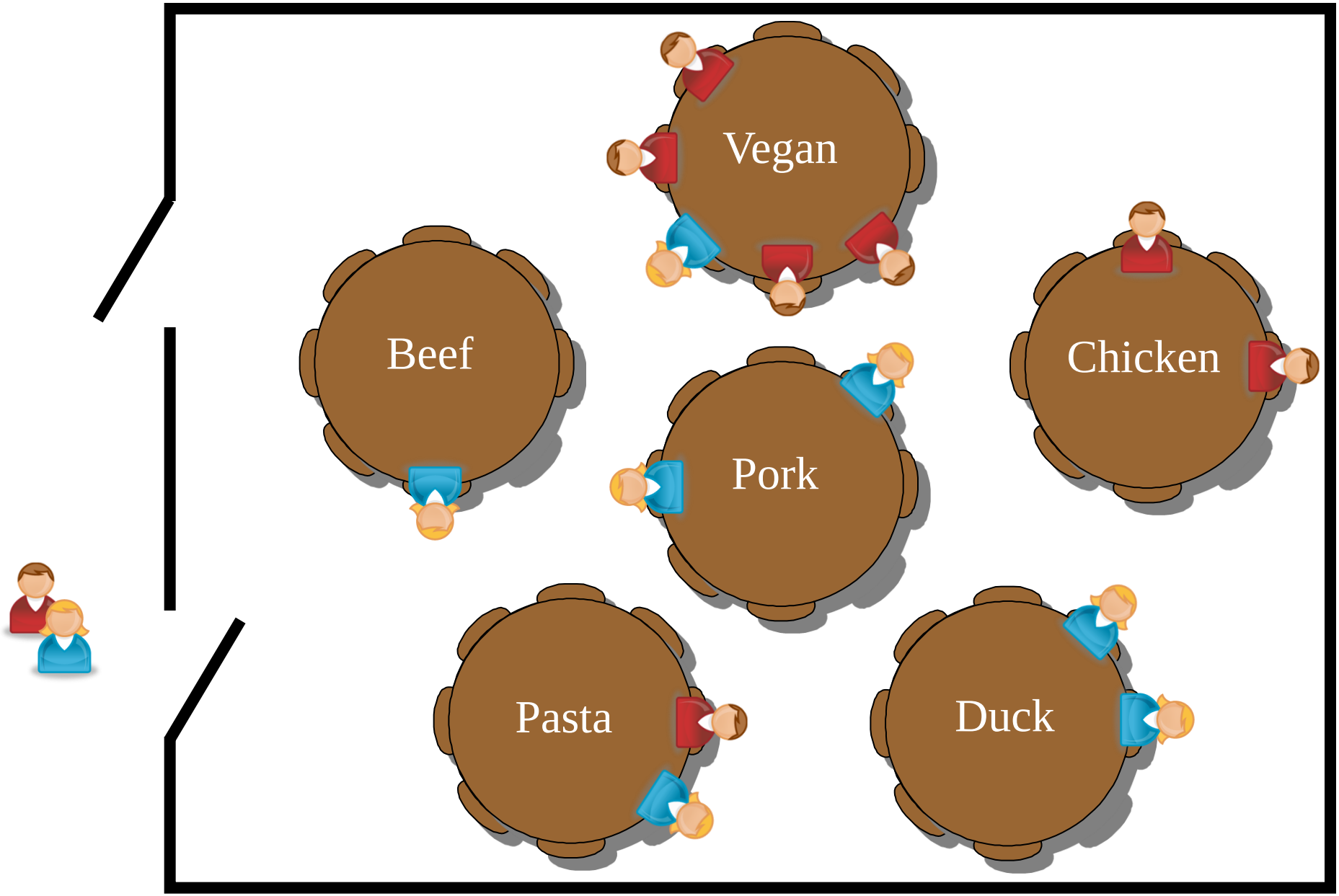
Le processus Dirichlet



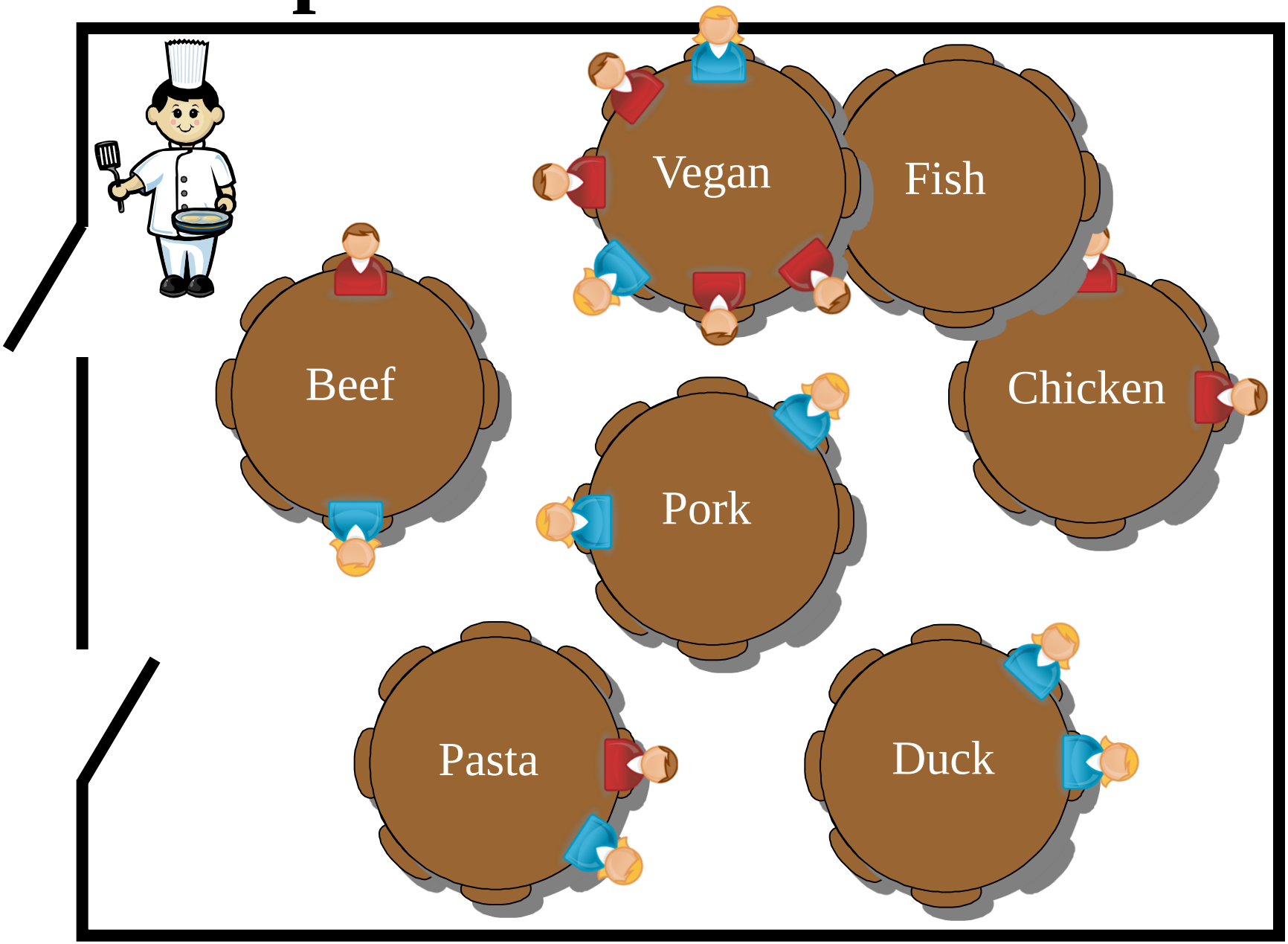
Le processus Dirichlet



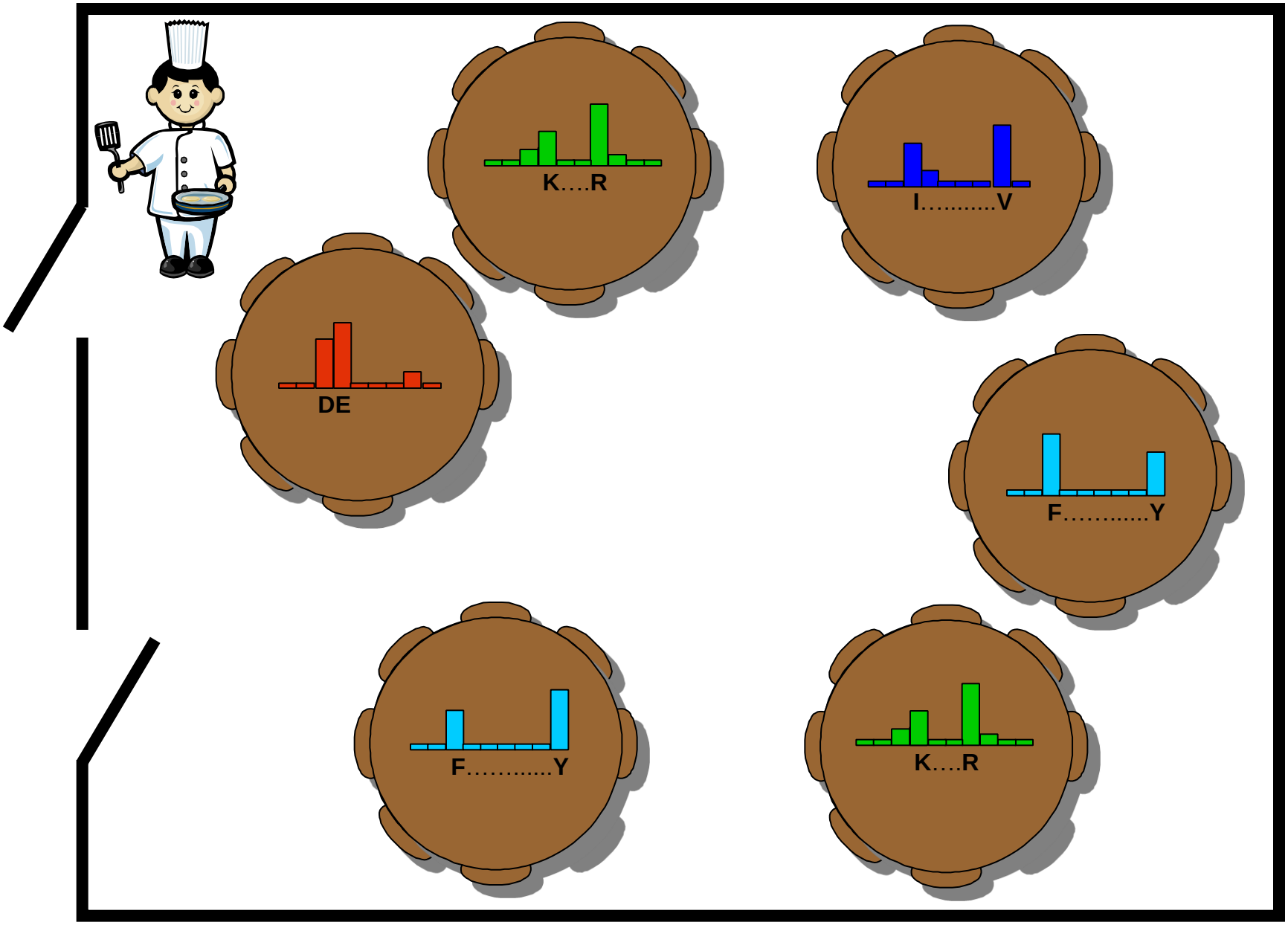
Le processus Dirichlet



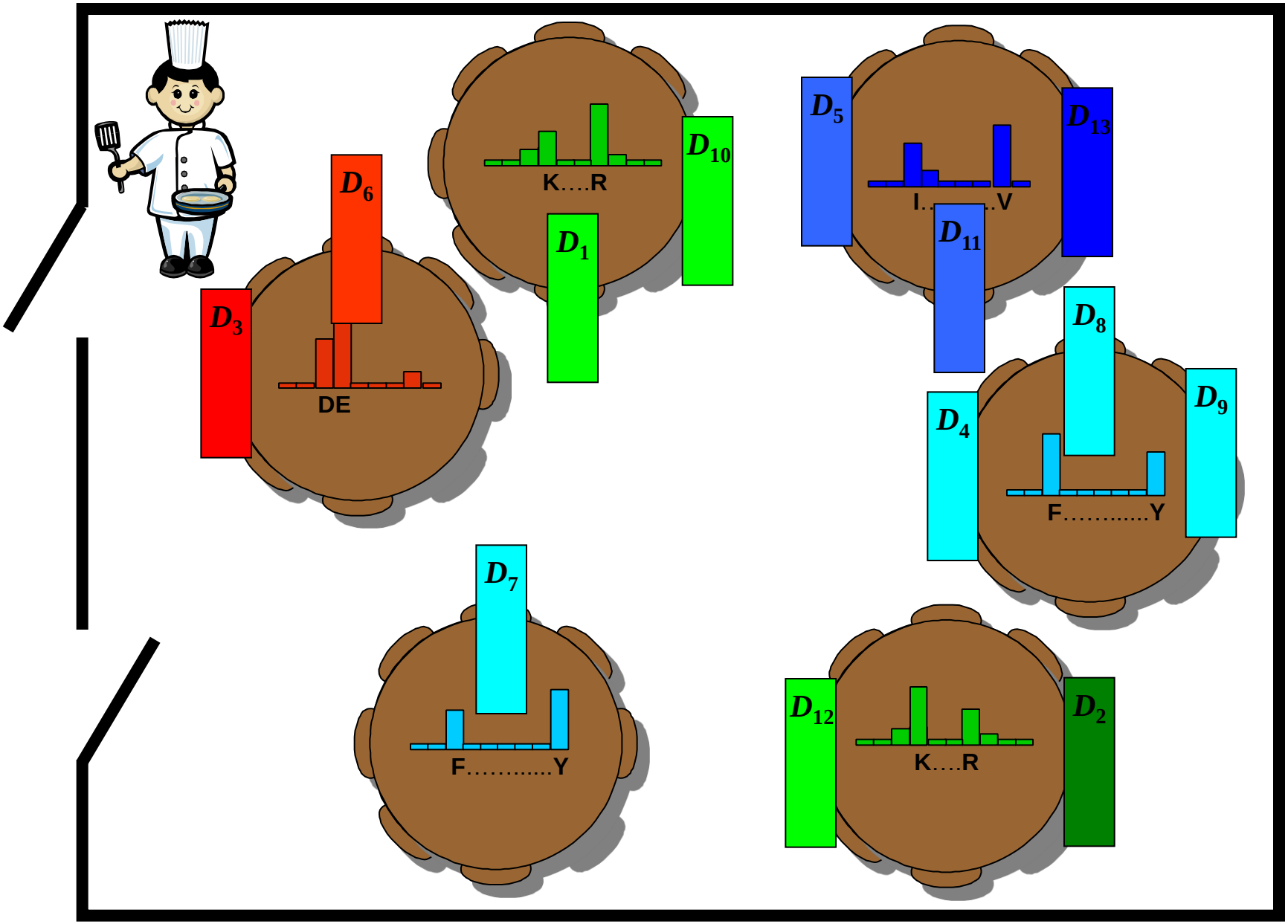
Le processus Dirichlet



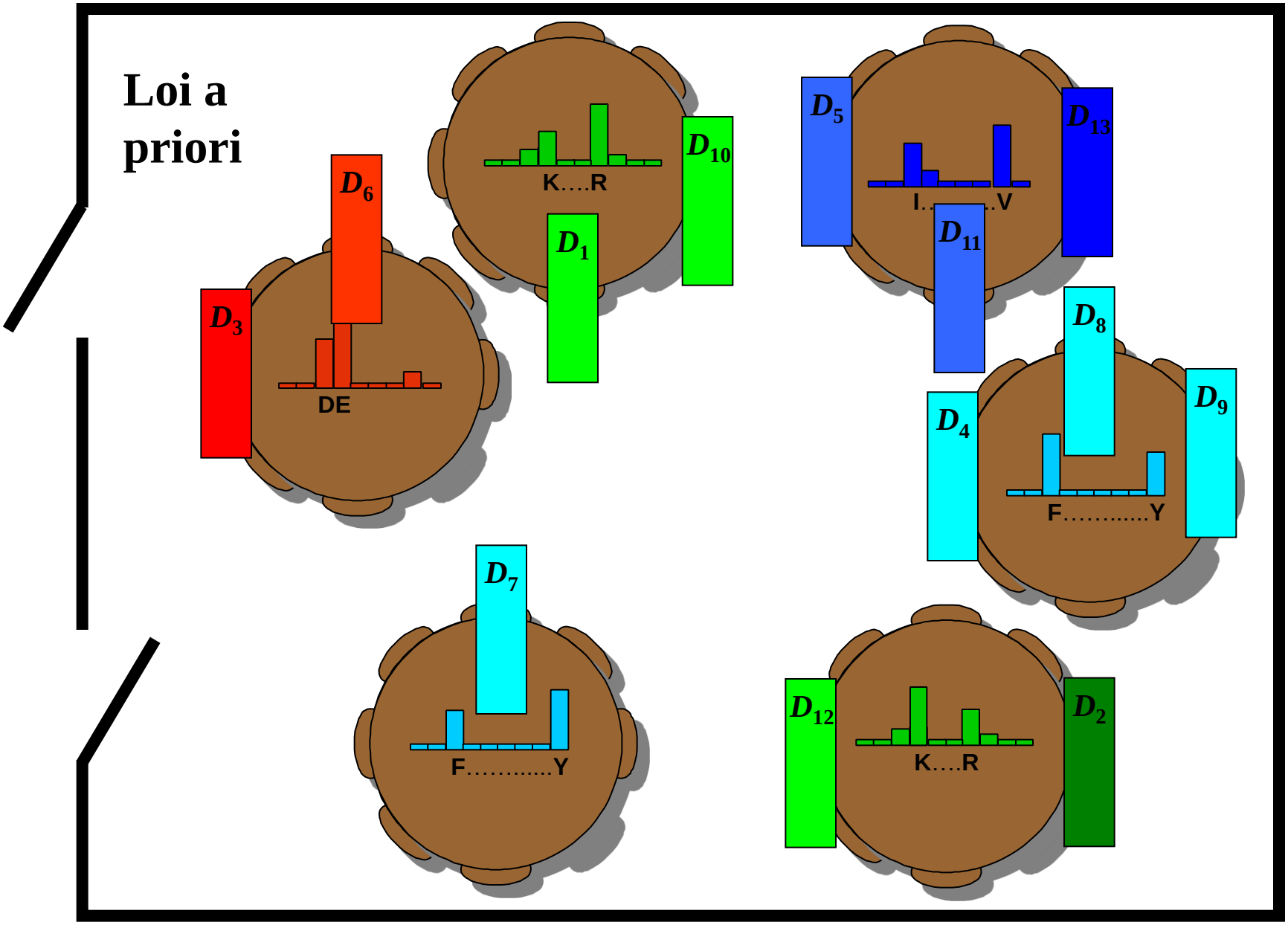
Le processus Dirichlet



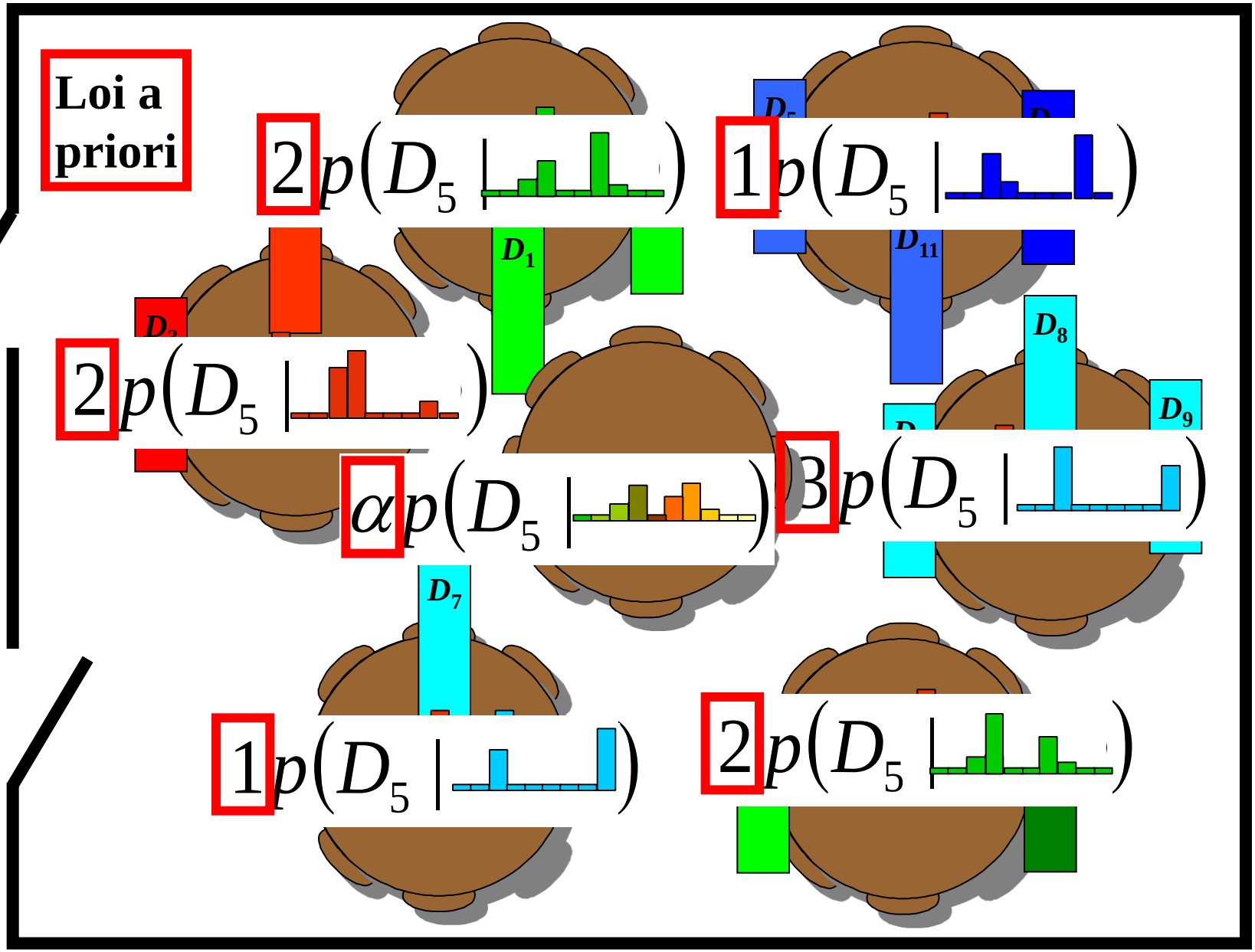
Le processus Dirichlet



Le processus Dirichlet



Le processus Dirichlet



Le processus Dirichlet

Passons à la pratique…