

SVM en génomique

Télécom Paris, 8 juin 2004

Jean-Philippe Vert

`Jean-Philippe.Vert@mines.org`

Plan

- Introduction à la génomique
- Espace des séquences
- Noyau spectral
- Noyau de sous-séquences
- Similarités par alignement
- Noyau de convolution
- Application: détection d'homologie lointaine
- Noyaux marginalisés et chaînes de Markov cachées

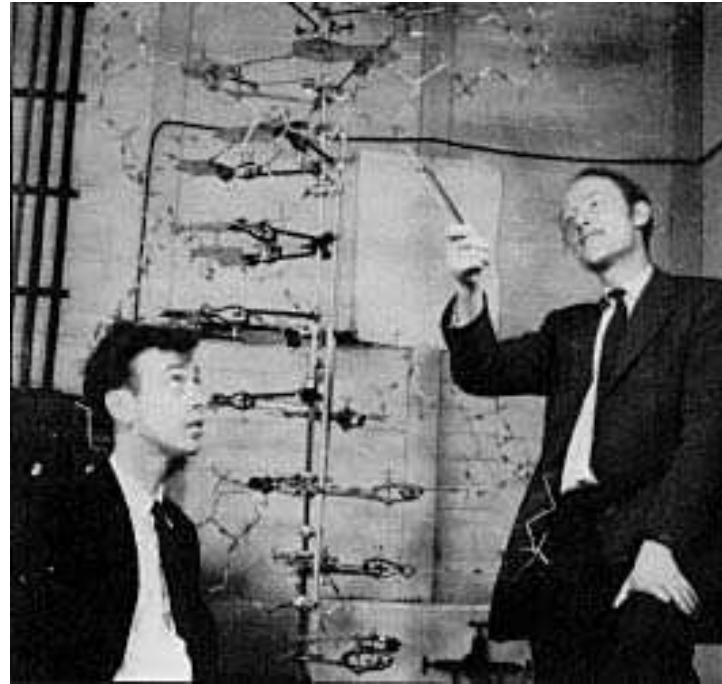
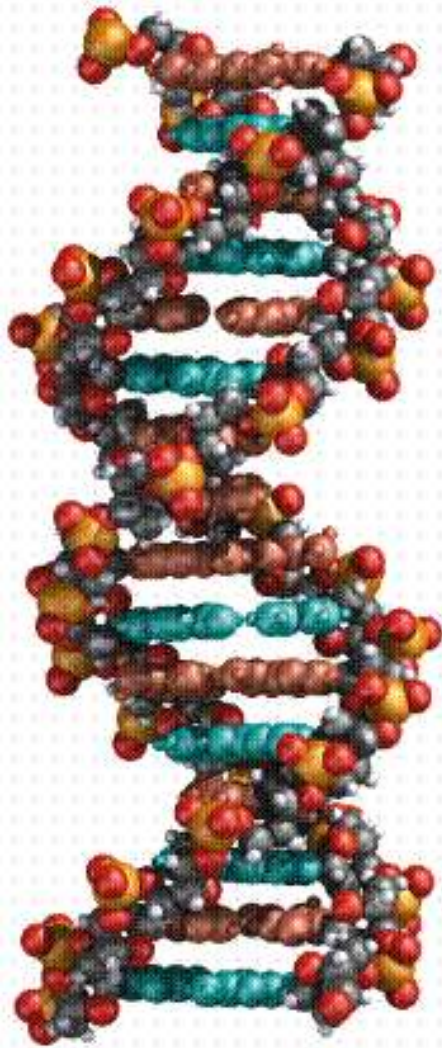
Introduction à la génomique

ADN

- L'*acide desoxyribonucléique* (ADN) est la molécule, présente dans toutes les cellules, qui contient l'information génétique transmise entre génération.
- L'ADN peut être en *simple brin* ou *double brin*.
- Un brin simple (aussi appelé polynucléotide) est un polymère linéaire composé de 4 *nucléotides*: adénosine (A), cytosine (C), guanine (G) et thymine (T)
- On représente un polynucléotide par une séquence orientée de lettre:

5' -A-T-T-C-A-G-G-C-A-T-T-A-G-C- 3'

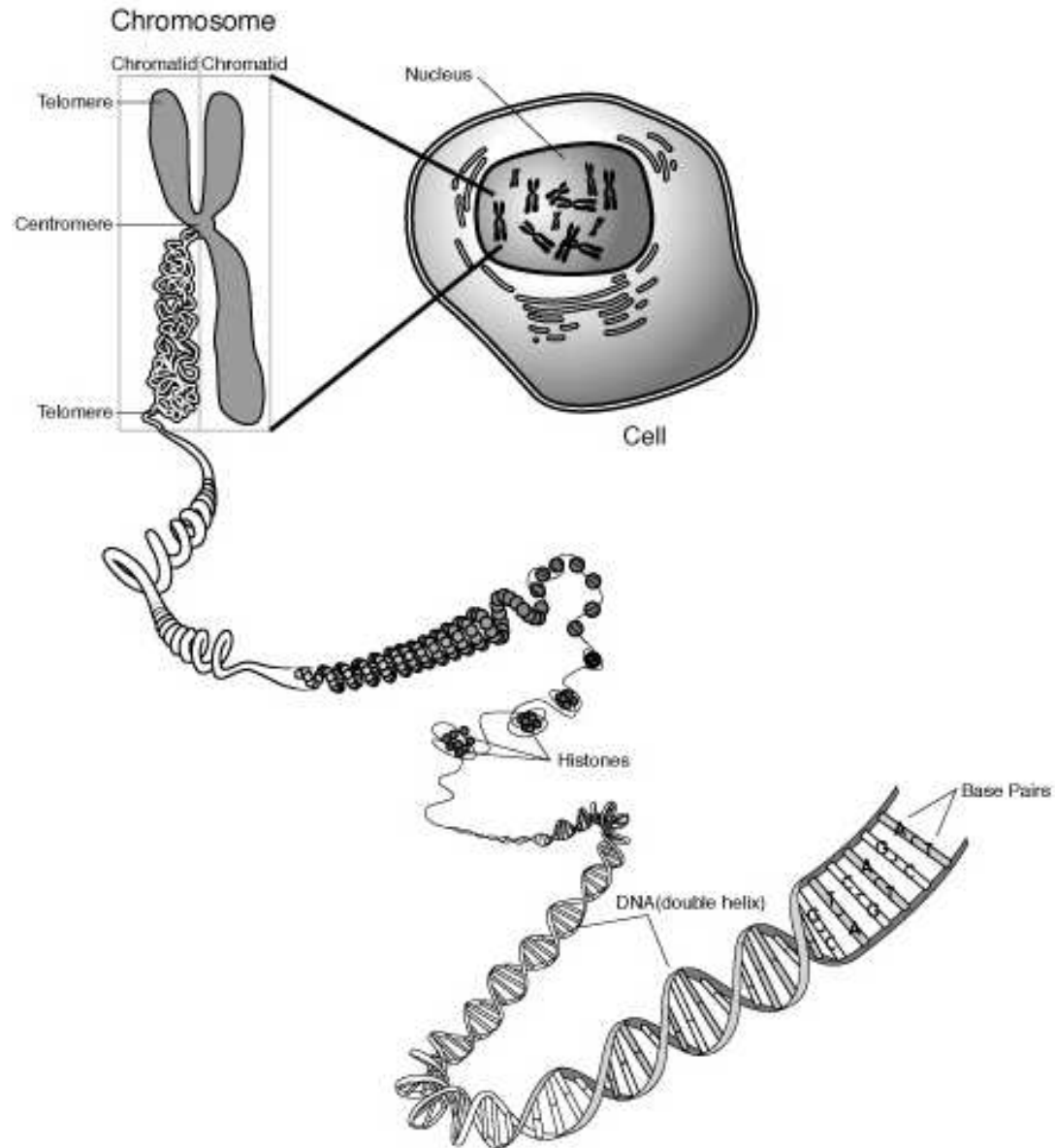
structure de l'ADN (Watson et Crick, 1953)



ADN et information

- La double hélice est stable, quelle que soit la séquence de nucléotides
- Parfait pour stocker 2 bits/base
- Distance entre 2 bases = 0.34nm, donc $6 \cdot 10^8$ bits/cm = *75GB/cm*

ADN et chromosomes

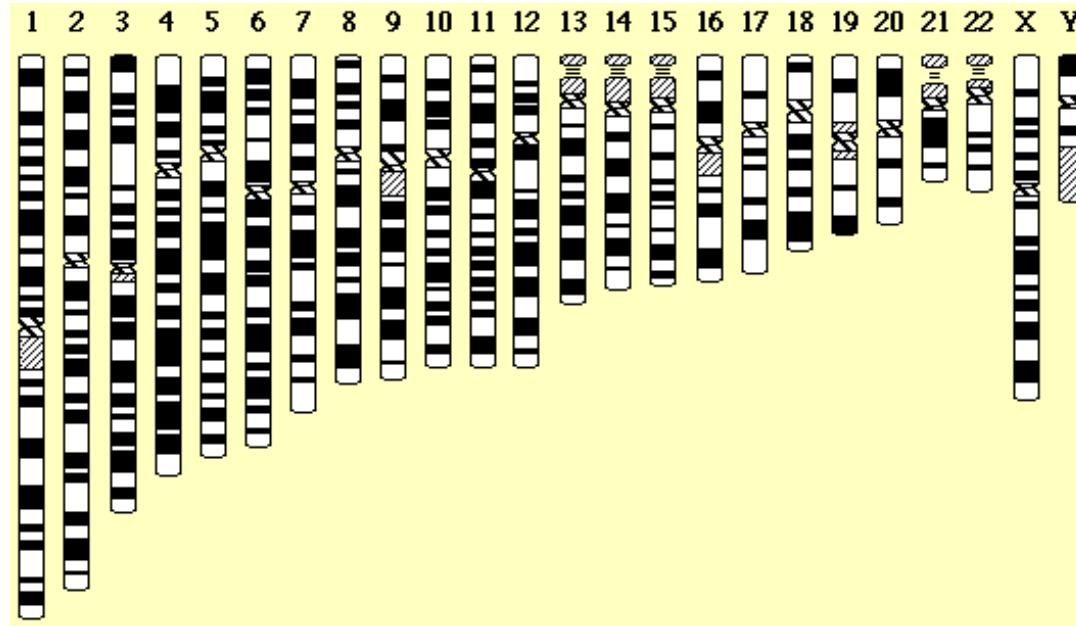


Génome

- Toutes les cellules d'un organisme ont (à peu près) le même ADN, appelé *génom*

Organisme	Chromosomes	Taille du génome (bp)
Bactéries	1	400,000 a 10,000,000
Levure	12	14,000,000
Mouche	4	300,000,000
Homme	46	6,000,000,000
Tulipe	24	60,000,000,000

Human genome



Séquencage

- *séquencer* = déterminer la séquence des lettres d'un ADN
- 1995: premier génome bactérien séquencé
- levure (1997), mouche (2000), homme (2003)...
- shotgun approche: les plus grands problèmes pour le séquencage des eukaryotes supérieurs sont informatique (assemblage)!

Sequencer



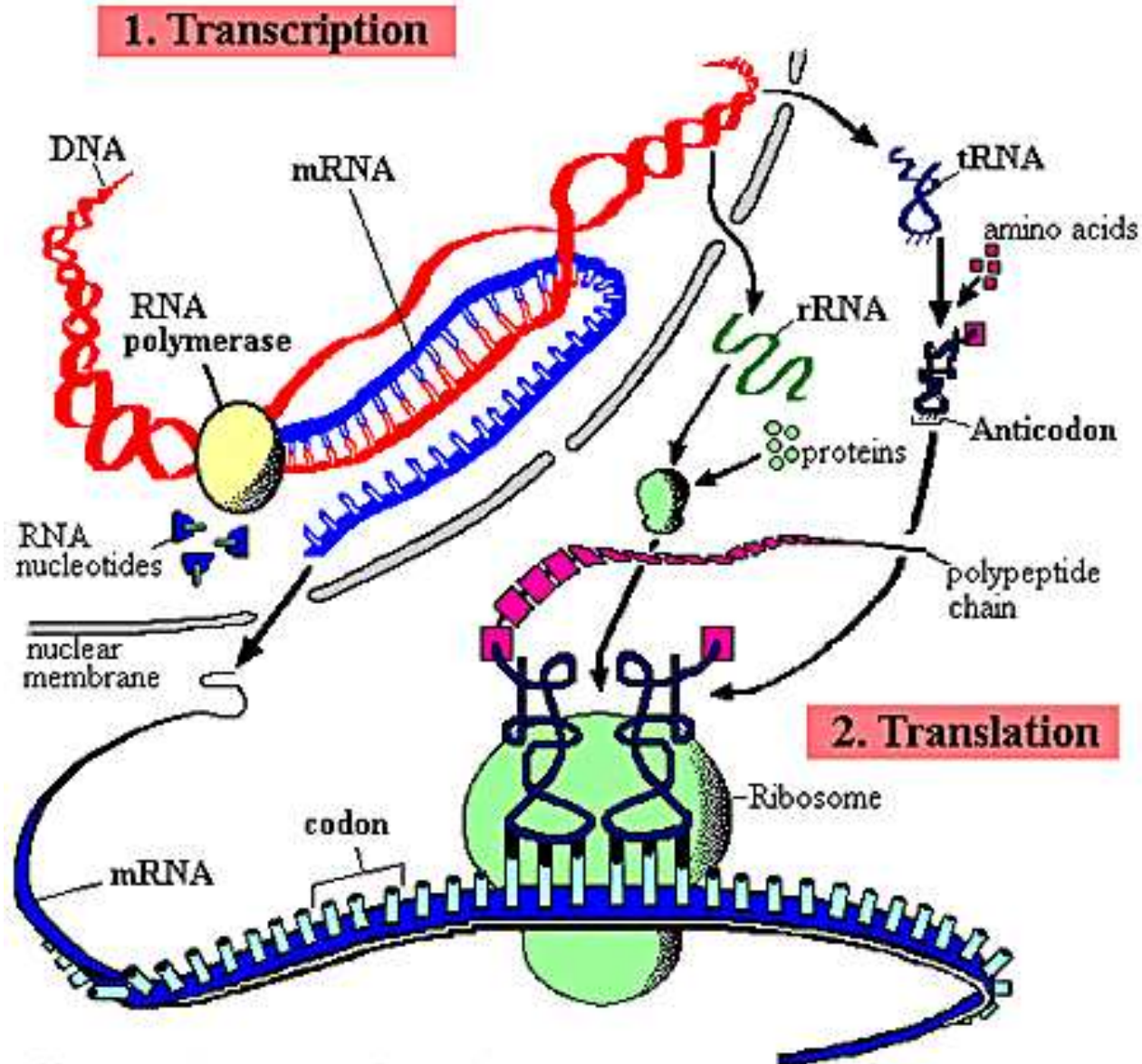
Gene

Une partie continue d'un brin d'ADN, a partir de laquelle une machinerie moleculaire complexe peut lire de l'information (encodée dans les lettres A,C,G,T) et créer une protéine particuliere

Dogme central

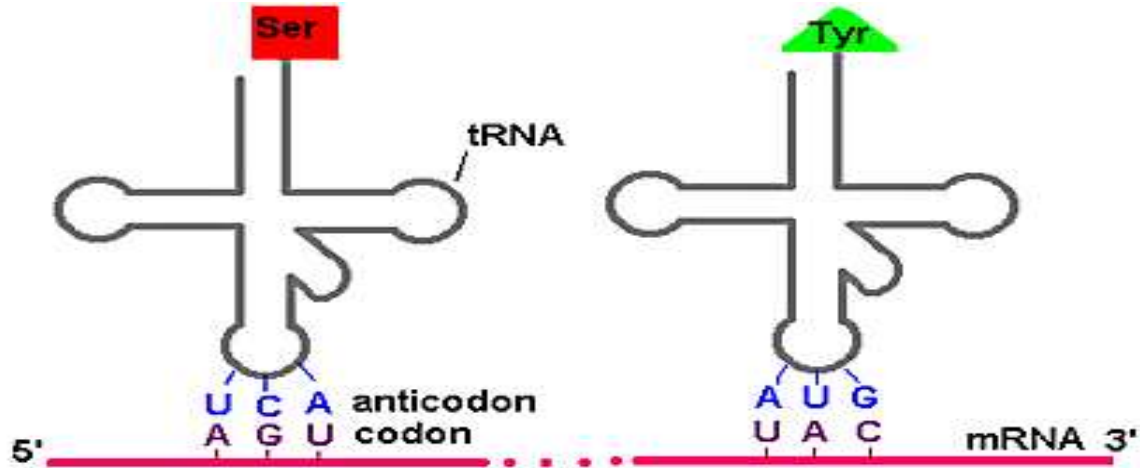


De l'ADN a la protéine



Protein synthesis

Code génétique



2nd base in codon

		U	C	A	G	
1st base in codon	U	Phe Phe Leu Leu	Ser Ser Ser Ser	Tyr Tyr STOP STOP	Cys Cys STOP Trp	U C A G
	C	Leu Leu Leu Leu	Pro Pro Pro Pro	His His Gln Gln	Arg Arg Arg Arg	U C A G
	A	Ile Ile Ile Met	Thr Thr Thr Thr	Asn Asn Lys Lys	Ser Ser Arg Arg	U C A G
	G	Val Val Val Val	Ala Ala Ala Ala	Asp Asp Glu Glu	Gly Gly Gly Gly	U C A G
						3rd base in codon

The Genetic Code

Espaces des séquences

Motivations

- *Croissance exponentielle* de la quantité de séquences disponible (effet direct des projets de séquençage): ADN, ARN, protéines.
- *Besoin d'algorithmes* pour comparer, classer, analyser ces séquences
- Applications: prédiction de structure/fonctions de protéines (*génomique fonctionnelle*), compréhension des signaux présents dans l'ADN, compréhension de processus tels l'épissage et la régulation génétique...

Séquences protéiques

- Nous nous concentrerons sur la structure primaire des protéines (dérivée de l'ADN codant grâce au *code génétique*)
- Ce sont des polymères linéaires d'*acides aminés*
- Il y a *20 acides aminés*. On peut donc représenter la structure chimique d'une protéine comme un texte sur un alphabet de 20 lettre.
- Exemple: l'insuline:
FVNQHLCGSHLVEALYLVCGERGFFYTPKA

Les 20 acides aminés

A : Alanine

F : Phenylalanine

E : Acide glutamique

T : Threonine

H : Histidine

I : Isoleucine

D : Acide aspartique

V : Valine

P : Proline

K : Lysine

C : Cysteine

V : Thyrosine

S : Sérine

G : Glycine

L : Leucine

M : Méthionine

R : Arginine

N : Asparagine

W : Tryptophane

Q : Glutamine

L'espace des protéines

- Soit \mathcal{A} l'ensemble des 20 acides aminés
- L'espace des séquences possibles est:

$$\mathcal{X} = \bigcup_{n=1}^{\infty} \mathcal{A}^n$$

- On note $|\mathbf{x}|$ la longueur d'une séquence $\mathbf{x} \in \mathcal{X}$.
- On compte actuellement environ 10^6 séquences protéiques observée 'dans la nature'
- ... a comparer aux 20^{300} séquences possibles de 300 lettres de \mathcal{A}

Evolution des séquences

- Au sein d'un organisme (croissance) et entre génération, les génomes sont dupliqués.
- Des *erreurs* de duplication ont parfois lieu, notamment:
 - *Substitution* d'une lettre par une autre
 - *Déletion* d'une ou plusieurs lettres
 - *Insertion* d'une ou plusieurs lettres
- Ces erreurs (aléatoires) sont à la base de *l'évolution*

Exemple

1. CGGSLIAMMWF'GV
2. CGGSLI**V**MMWF'GV
3. CGGSLIVMM**NRLM**WF'GV
4. CLIVMMNRLMWF'GV

que l'on écrit généralement:

1. CGGSLIAMM---WF'GV
2. CGGSLI**V**MM---WF'GV
3. CGGSLIVMM**NRLM**WF'GV
4. C--LIVMMNRLMWF'GV

Objectif

- Pendant l'évolution, les *structures et les fonctions* sont beaucoup plus conservées que les séquences
- Pour prédire structure et fonction, il faut donc 'reconnaitre' l'évolution à partir des séquences
- Objectif: faire un *noyau pour séquences protéiques qui reconnaisse l'évolution*: $K(x, x')$ est grand quand x et x' sont liées par l'évolution.

Noyau spectral

Motivations

- Nous connaissons plusieurs noyaux génériques pour des vecteurs: linéaire, polynomial, RBF Gaussien...
- Comment faire un noyau *simple* et *rapide a calculer* pour des *séquences de longueur variable*
- Référence: C. Leslie, E. Eskin, W.S. Noble, The spectrum kernel: a string kernel for SVM protein classification Proceedings of PSB 2002.

k -spectre d'une séquence

- Pour tout entier $k > 0$, le k -spectre d'une séquence de longueur finie est la *liste de k -mers (suites de k lettres) qu'elle contient.*
- Exemple: le 3-spectre de

$x = \text{CGGSLIAMMWF'GV}$

est:

(CGG , GGS , GSL , SLI , LIA , IAM , AMM , MMW , MWF , WF'G , F'GW

Noyau spectral

- Soit $k > 0$ fixé
- Pour tout $u \in \mathcal{A}^k$, soit $\Phi_u(\mathbf{x}) :=$ *le nombre de fois que le k -gram u apparaît dans \mathbf{x} .*
- On définit le *noyau spectral* par:

$$K(\mathbf{x}, \mathbf{x}') := \sum_{u \in \mathcal{A}^k} \Phi_u(\mathbf{x}) \Phi_u(\mathbf{x}')$$

- C'est un noyau *défini positif*!
- Deux séquences sont comparées à travers leurs k -spectres.

Implémentation

- C'est une somme sur 20^k termes...
- mais la plupart sont nuls. Le k -spectre d'une séquence \mathbf{x} a au plus $|\mathbf{x}| - k$ composantes non nulles.
- On peut donc *calculer le noyau en* $O(|\mathbf{x}| + |\mathbf{x}'|)$ (voir Vishwanathan et Smola, 2004)
- La *classification par SVM* d'une séquence \mathbf{x} prend $O(|\mathbf{x}|)$:

$$f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) = \sum_u w_u \Phi_u(\mathbf{x}) = \sum_{i=1}^{|\mathbf{x}|} w_{x_i \dots x_{i+k-1}}$$

Remarques

- Le noyau spectral est *indépendant du type de séquences*: il fonctionne aussi bien sur des séquences biologiques que sur du texte en langage naturel, des programmes en C, ...
- Des *variantes* ont été proposées, notamment pour comparer les spectres a quelques mutations pres (l'implémentation est alors moins rapide)
- Pas toujours tres performant, mais *simple et rapide*

Noyau de sous-séquences

Motivations

- Le noyau spectral ne peut pas prendre en compte les *insertions* et *deletions*.
- Il existe peut-être des features intéressantes qui sont des *suites non contigues* de lettres
- Références: H. Lodhi et al.,
Text classification using string kernels, Journal of Machine Learning Research (2), 419-444, 2002.

Sous-séquences

- Pour $1 \leq k \leq n \in \mathbb{N}$, on note $\mathcal{I}(k, n)$ l'ensemble des *suite d'indices* $\mathbf{i} = (i_1, \dots, i_k)$, avec $1 \leq i_1 < i_2 < \dots < i_k \leq n$.
- Pour une séquence $\mathbf{x} = x_1 \dots x_n \in \mathcal{X}$ de longueur n , et pour une suite d'indices $\mathbf{i} \in \mathcal{I}(k, n)$, on définit la *sous-séquence*:

$$\mathbf{x}(\mathbf{i}) := x_{i_1} x_{i_2} \dots x_{i_k}$$

- La *longueur* de la sous-séquence est:

$$l(\mathbf{i}) = i_k - i_1 + 1.$$

Exemple de sous-séquence

ABRACADABRA

- $\mathbf{i} = (3, 4, 7, 8, 10)$
- $\mathbf{x}(\mathbf{i}) = \text{RADAR}$
- $l(\mathbf{i}) = 10 - 3 + 1 = 8$

Noyau de sous-séquences

- Réciproquement, soit $k \in \mathbb{N}$ et $\lambda \in \mathbb{R}^+$ fixés. Pour tout $\mathbf{u} \in \mathcal{A}^k$, on définit $\Phi_{\mathbf{u}} : \mathcal{X} \rightarrow \mathbb{R}$ par:

$$\forall \mathbf{x} \in \mathcal{X}, \quad \Phi_{\mathbf{u}}(\mathbf{x}) = \sum_{\mathbf{i} \in \mathcal{I}(k, |\mathbf{x}|): \mathbf{x}(\mathbf{i}) = \mathbf{u}} \lambda^{l(\mathbf{i})}.$$

- Le *noyau de sous-séquences* est le n.d.p. défini par:

$$\forall (\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2, \quad K_{k,\lambda}(\mathbf{x}, \mathbf{x}') = \sum_{\mathbf{u} \in \mathcal{A}^k} \Phi_{\mathbf{u}}(\mathbf{x}) \Phi_{\mathbf{u}}(\mathbf{x}')$$

Exemple

u	ca	ct	at	ba	bt	cr	ar	br
$\Phi_u(\text{cat})$	λ^2	λ^3	λ^2	0	0	0	0	0
$\Phi_u(\text{car})$	λ^2	0	0	0	0	λ^3	λ^2	0
$\Phi_u(\text{bat})$	0	0	λ^2	λ^2	λ^3	0	0	0
$\Phi_u(\text{bar})$	0	0	0	λ^2	0	0	λ^2	λ^3

$$\begin{cases} K(\text{cat,cat}) = K(\text{car,car}) = 2\lambda^4 + \lambda^6 \\ K(\text{cat,car}) = \lambda^4 \\ K(\text{cat,bar}) = 0 \end{cases}$$

Implémentation

On désire calculer, pour $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, le noyau:

$$\begin{aligned} K_{n,\lambda}(\mathbf{x}, \mathbf{x}') &= \sum_{\mathbf{u} \in \mathcal{A}^n} \Phi_{\mathbf{u}}(\mathbf{x}) \Phi_{\mathbf{u}}(\mathbf{x}') \\ &= \sum_{\mathbf{u} \in \mathcal{A}^n} \sum_{\mathbf{i}: \mathbf{x}(\mathbf{i})=\mathbf{u}} \sum_{\mathbf{i}': \mathbf{x}'(\mathbf{i}')=\mathbf{u}} \lambda^{l(\mathbf{i})+l(\mathbf{i}')} \end{aligned}$$

L'énumération de toutes les sous-séquences est trop longue quand n augmente.

Implémentation

Pour $\mathbf{u} \in \mathcal{A}^n$ on a défini

$$\Phi_{\mathbf{u}}(\mathbf{x}) = \sum_{\mathbf{i}: \mathbf{x}(\mathbf{i}) = \mathbf{u}} \lambda^{i_n - i_1 + 1}.$$

Soit maintenant

$$\Psi_{\mathbf{u}}(\mathbf{x}) = \sum_{\mathbf{i}: \mathbf{x}(\mathbf{i}) = \mathbf{u}} \lambda^{|\mathbf{x}| - i_1 + 1}.$$

Implémentation (cont.)

En notant $\mathbf{x}(1, j) = x_1 \dots x_j$, une simple ré-écriture montre que, si on note $a \in \mathcal{A}$ la dernière lettre de \mathbf{u} ($\mathbf{u} = \mathbf{v}a$):

$$\Phi_{\mathbf{v}a}(\mathbf{x}) = \sum_{j \in [1, |\mathbf{x}|]: x_j = a} \Psi_{\mathbf{v}}(\mathbf{x}(1, j-1)) \lambda,$$

et

$$\Psi_{\mathbf{v}a}(\mathbf{x}) = \sum_{j \in [1, |\mathbf{x}|]: x_j = a} \Psi_{\mathbf{v}}(\mathbf{x}(1, j-1)) \lambda^{|\mathbf{x}| - j + 1}.$$

Implémentation (cont.)

On remarque également que, si on s'intéresse à une séquence de la forme $\mathbf{x}a$ (i.e., dont la dernière lettre est $a \in \mathcal{A}$), alors

- Si la dernière lettre de \mathbf{u} *n'est pas* a :

$$\begin{cases} \Phi_{\mathbf{u}}(\mathbf{x}a) &= \Phi_{\mathbf{u}}(\mathbf{x}), \\ \Psi_{\mathbf{u}}(\mathbf{x}a) &= \lambda \Psi_{\mathbf{u}}(\mathbf{x}). \end{cases}$$

- Si la dernière lettre de \mathbf{u} est égale à a (i.e., $\mathbf{u} = \mathbf{v}a$ avec $\mathbf{v} \in \mathcal{A}^{n-1}$):

$$\begin{cases} \Phi_{\mathbf{v}a}(\mathbf{x}a) &= \Phi_{\mathbf{v}a}(\mathbf{x}) + \lambda \Psi_{\mathbf{v}}(\mathbf{x}), \\ \Psi_{\mathbf{v}a}(\mathbf{x}a) &= \lambda \Psi_{\mathbf{v}a}(\mathbf{x}) + \lambda \Psi_{\mathbf{v}}(\mathbf{x}). \end{cases}$$

Implémentation (cont.)

Montrons comment

$$B_n(\mathbf{x}, \mathbf{x}') := \sum_{\mathbf{u} \in \mathcal{A}^n} \Psi_{\mathbf{u}}(\mathbf{x}) \Psi_{\mathbf{u}}(\mathbf{x}')$$

et le noyau qui nous intéresse:

$$K_n(\mathbf{x}, \mathbf{x}') := \sum_{\mathbf{u} \in \mathcal{A}^n} \Phi_{\mathbf{u}}(\mathbf{x}) \Phi_{\mathbf{u}}(\mathbf{x}')$$

peuvent se calculer de manière récursive. On a tout d'abord:

$$\begin{cases} B_0(\mathbf{x}, \mathbf{x}') = 1 & \text{pour tout } \mathbf{x}, \mathbf{x}' \\ B_k(\mathbf{x}, \mathbf{x}') = 0 & \text{si } \min(|\mathbf{x}|, |\mathbf{x}'|) < k \end{cases}$$

Calcul récurive de B_n

$$B_n(\mathbf{x}a, \mathbf{x}')$$

$$= \sum_{\mathbf{u} \in \mathcal{A}^n} \Psi_{\mathbf{u}}(\mathbf{x}a) \Psi_{\mathbf{u}}(\mathbf{x}')$$

$$= \lambda \sum_{\mathbf{u} \in \mathcal{A}^n} \Psi_{\mathbf{u}}(\mathbf{x}) \Psi_{\mathbf{u}}(\mathbf{x}') + \lambda \sum_{\mathbf{v} \in \mathcal{A}^{n-1}} \Psi_{\mathbf{v}}(\mathbf{x}) \Psi_{\mathbf{v}a}(\mathbf{x}')$$

$$= \lambda B_n(\mathbf{x}, \mathbf{x}') +$$

$$\lambda \sum_{\mathbf{v} \in \mathcal{A}^{n-1}} \Psi_{\mathbf{v}}(\mathbf{x}) \left(\sum_{j \in [1, |\mathbf{x}'|]: x'_j = a} \Psi_{\mathbf{v}}(\mathbf{x}'(1, j-1)) \lambda^{|\mathbf{x}'| - j + 1} \right)$$

$$= \lambda B_n(\mathbf{x}, \mathbf{x}') + \sum_{j \in [1, |\mathbf{x}'|]: x'_j = a} B_{n-1}(\mathbf{x}, \mathbf{x}'(1, j-1)) \lambda^{|\mathbf{x}'| - j + 2}$$

Calcul récurive de K_n

$$K_n(\mathbf{x}a, \mathbf{x}')$$

$$= \sum_{\mathbf{u} \in \mathcal{A}^n} \Phi_{\mathbf{u}}(\mathbf{x}a) \Phi_{\mathbf{u}}(\mathbf{x}')$$

$$= \sum_{\mathbf{u} \in \mathcal{A}^n} \Phi_{\mathbf{u}}(\mathbf{x}) \Phi_{\mathbf{u}}(\mathbf{x}') + \lambda \sum_{\mathbf{v} \in \mathcal{A}^{n-1}} \Psi_{\mathbf{v}}(\mathbf{x}) \Phi_{\mathbf{v}a}(\mathbf{x}')$$

$$= K_n(\mathbf{x}, \mathbf{x}') +$$

$$\lambda \sum_{\mathbf{v} \in \mathcal{A}^{n-1}} \Psi_{\mathbf{v}}(\mathbf{x}) \left(\sum_{j \in [1, |\mathbf{x}'|]: x'_j = a} \Psi_{\mathbf{v}}(\mathbf{x}'(1, j-1)) \lambda \right)$$

$$= \lambda K_n(\mathbf{x}, \mathbf{x}') + \lambda^2 \sum_{j \in [1, |\mathbf{x}'|]: x'_j = a} B_{n-1}(\mathbf{x}, \mathbf{x}'(1, j-1))$$

Remarques

- L'algorithme est en $O(n \|\mathbf{x}\| \|\mathbf{x}'\|^2)$. On peut réutiliser les calculs pour avoir une complexité $O(n \|\mathbf{x}\| \|\mathbf{x}'\|)$
- Beaucoup plus long que le noyau spectral.
- A fourni des résultats intéressants pour la classification de textes.

Similarité par alignement

Le probleme

Soient les deux séquences:

$$x_1 = \text{CGGSLIAMMWF'GV}$$
$$x_2 = \text{CLIVMMNRLMWF'GV}$$

Comment *quantifier leur similarité* au sens de l'évolution?

Alignement des séquences

Aligner = *mettre en correspondance les lettres* des deux séquences, quitte à ajouter des gaps dans l'une ou dans l'autre

```
CGGSLIAMM---WFGV
|. . . | | | | . . . . | | | |
C--LIVMMNRLMWFGV
```

Alignement (cont.)

Mathématiquement, un alignement π de $p \geq 0$ lettres entre deux séquences \mathbf{x} et \mathbf{x}' de longueurs m et n est une paire de p -uplets:

$$\pi = ((\pi_1(1), \dots, \pi_1(p)), (\pi_2(1), \dots, \pi_2(p))),$$

avec:

$$1 \leq \pi_1(1) < \pi_1(2) < \dots < \pi_1(p) \leq |\mathbf{x}|,$$

$$1 \leq \pi_2(1) < \pi_2(2) < \dots < \pi_2(p) \leq |\mathbf{y}|.$$

On note $\Pi(\mathbf{x}, \mathbf{x}')$ l'ensemble des alignements possibles entre les deux séquences \mathbf{x} et \mathbf{x}' .

Exemple

$$\pi = ((1, 5, 6, 7, 8, 9, 10, 11, 12, 13), (1, 2, 3, 4, 5, 6, 7, 12, 13, 14))$$

CGGSLIAMM---WFGV

|...|||...|||

C--LIVMMNRLMWF'GV

Scoring d'alignement

Un alignement est bon si:

- les lettres alignées 'se ressemblent';
- il y a peu de gaps.

On va construire une fonction (un score) qui est d'autant plus grand que les séquences se ressemblent

Scoring d'alignement (cont.)

Pour tout alignement $\pi \in \Pi(\mathbf{x}, \mathbf{y})$, on définit son score par:

$$s_{S,g}(\pi) := \sum_{i=1}^{|\pi|} S(x_{\pi_1(i)}, y_{\pi_2(i)})$$

$$- \sum_{i=1}^{|\pi|-1} [g(\pi_1(i+1) - \pi_1(i) - 1) + g(\pi_2(i+1) - \pi_2(i) - 1)],$$

ou:

- S est une matrice 20×20 qui définit la similarité entre les paires de lettres
- $g : \mathbb{N} \rightarrow \mathbb{R}$ est une fonction qui définit le coût d'un gap de longueur donné (avec $g(0) = 0$).

Exemple

$$\pi = ((1, 5, 6, 7, 8, 9, 10, 11, 12, 13), (1, 2, 3, 4, 5, 6, 7, 12, 13, 14))$$

CGGSLIAMM---WFGV

|...|||...|||

C--LIVMMNRLMWFGV

$$\begin{aligned} s_{S,g}(\pi) = & S(C, C) + S(L, L) + S(I, I) + S(A, V) + 2S(M, M) \\ & + S(W, W) + S(F, F) + S(G, G) + S(V, V) - g(3) - g(4) \end{aligned}$$

Score de Smith-Waterman

Definition 1 *On appelle score de Smith-Waterman, ou score d'alignement local entre deux séquences x et x' le score du meilleur alignement:*

$$SW_{S,g}(x, y) := \max_{\pi \in \Pi(x,y)} s_{S,g}(\pi).$$

Calcul de SW

- Soit le coût linéaire $g(n) = \gamma n$
- Pour tout $1 \leq a \leq b \leq |\mathbf{x}|$, soit $\mathbf{x}[a, b] := x_a \dots x_b$.
- Pour tout $1 \leq a \leq |\mathbf{x}|$ et $1 \leq a' \leq |\mathbf{x}'|$, on observe que:

$$SW_{S,g}(\mathbf{x}[1, a], \mathbf{x}'[1, a']) =$$

$$\max \begin{cases} 0 \\ SW_{S,g}(\mathbf{x}[1, a-1], \mathbf{x}'[1, a']) + \gamma \\ SW_{S,g}(\mathbf{x}[1, a], \mathbf{x}'[1, a'-1]) + \gamma \\ SW_{S,g}(\mathbf{x}[1, a-1], \mathbf{x}'[1, a'-1]) + S(x_a, x'_{a'}) \end{cases}$$

- Cela fournit une méthode de calcul en $O(|\mathbf{x}| |\mathbf{x}'|)$

Exercice

Trouver un algorithme en $O(|\mathbf{x}| |\mathbf{x}'|)$ pour calculer le score d'alignement local dans le cas d'un cout affine:

$$\begin{cases} g(0) & = 0, \\ g(n) & = d + e(n - 1) \text{ if } n \geq 1, \end{cases}$$

Noyau de convolution

Motivations

- Le score d'alignement local est censé être une bonne mesure de similarité entre séquences biologiques.
- Mais... ce n'est pas un noyau en général (dépend du choix des paramètres).
- Comment faire un noyau qui imite ce score?
- Ref: Haussler 1999 (Convolution kernels), Vert et al. 2004 (Local alignment kernels)

Lemme: Opérations de bases sur les n.d.p.

Lemme 2 *Si K_1 et K_2 sont des n.d.p., alors:*

$$K_1 + K_2,$$

$$K_1 K_2, \text{ et}$$

$$cK_1, \text{ pour } c \geq 0,$$

sont également des n.d.p. Si $(K_i)_{i \geq 1}$ est une famille de n.d.p. qui converge ponctuellement en une fonction K :

$$\forall (\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2, \quad K(\mathbf{x}, \mathbf{x}') = \lim_{n \rightarrow \infty} K_n(\mathbf{x}, \mathbf{x}'),$$

alors K est aussi un n.d.p.

Preuves

Soient A et B des matrices $n \times n$ semi-définies positives (s-d.p.). Alors, par diagonalisation, on peut écrire:

$$A_{i,j} = \sum_{p=1}^n f_p(i) f_p(j)$$

pour une famille de vecteurs f_1, \dots, f_n . Donc, pour $\alpha \in \mathbb{R}^n$ arbitraire:

$$\sum_{i,j=1}^n \alpha_i \alpha_j A_{i,j} B_{i,j} = \sum_{p=1}^n \sum_{i,j=1}^n \alpha_i f_p(i) \alpha_j f_p(j) B_{i,j} \geq 0.$$

Preuves (cont.)

La matrice $C_{i,j} = A_{i,j}B_{i,j}$ est donc s-d.p, donc K_1K_2 est un n.d.p. si K_1 et K_2 le sont. Les autres propriétés découlent directement de la définition. \square

Lemme: Noyaux produits

Lemme 3 Soit $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$. Soient K_1 un n.d.p. sur \mathcal{X}_1 , et K_2 un n.d.p. sur \mathcal{X}_2 . Alors les fonctions suivantes sont des n.d.p. sur \mathcal{X} :

- la somme directe,

$$K((\mathbf{x}_1, \mathbf{x}_2), (\mathbf{y}_1, \mathbf{y}_2)) = K_1(\mathbf{x}_1, \mathbf{y}_1) + K_2(\mathbf{x}_2, \mathbf{y}_2),$$

- le produit direct

$$K((\mathbf{x}_1, \mathbf{x}_2), (\mathbf{y}_1, \mathbf{y}_2)) = K_1(\mathbf{x}_1, \mathbf{y}_1) K_2(\mathbf{x}_2, \mathbf{y}_2).$$

Preuves

Si K_1 est un noyau, soit $\Phi_1 : \mathcal{X}_1 \mapsto \mathcal{H}$ tel que

$$K_1(\mathbf{x}_1, \mathbf{y}_1) = \langle \Phi_1(\mathbf{x}_1), \Phi_1(\mathbf{y}_1) \rangle_{\mathcal{H}}.$$

Soit $\Phi : \mathcal{X}_1 \times \mathcal{X}_2 \rightarrow \mathcal{H}$ défini par:

$$\Phi((\mathbf{x}_1, \mathbf{x}_2)) = \Phi_1(\mathbf{x}_1).$$

Alors pour $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ et $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2) \in \mathcal{X}$, on a

$$\langle \Phi((\mathbf{x}_1, \mathbf{x}_2)), \Phi((\mathbf{y}_1, \mathbf{y}_2)) \rangle_{\mathcal{H}} = K_1(\mathbf{x}_1, \mathbf{y}_1),$$

ce qui montre que $K(\mathbf{x}, \mathbf{y}) := K_1(\mathbf{x}_1, \mathbf{y}_1)$ est un n..p. sur $\mathcal{X}_1 \times \mathcal{X}_2$. Le lemme découle des opérations d'addition et

multiplication sur n.d.p. \square

Lemme: Noyaux pour parties

Lemme 4 Soit K un n.d.p. sur un espace \mathcal{X} , et soit $\mathcal{P}(\mathcal{X})$ l'ensemble des **parties finies** d'éléments de \mathcal{X} . Alors la fonction K_P sur $\mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X})$ définie par:

$$\forall A, B \in \mathcal{P}(\mathcal{X}), \quad K_P(A, B) := \sum_{\mathbf{x} \in A} \sum_{\mathbf{y} \in B} K(\mathbf{x}, \mathbf{y})$$

est un n.d.p. sur $\mathcal{P}(\mathcal{X})$.

Preuves

Soit $\Phi : \mathcal{X} \mapsto \mathcal{H}$ tel que

$$K(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle_{\mathcal{H}}.$$

Alors, pour $A, B \in \mathcal{P}(\mathcal{X})$, on a:

$$\begin{aligned} K_P(A, B) &= \sum_{\mathbf{x} \in A} \sum_{\mathbf{y} \in B} \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle_{\mathcal{H}} \\ &= \left\langle \sum_{\mathbf{x} \in A} \Phi(\mathbf{x}), \sum_{\mathbf{y} \in B} \Phi(\mathbf{y}) \right\rangle_{\mathcal{H}} \\ &= \langle \Phi_P(A), \Phi_P(B) \rangle_{\mathcal{H}}, \end{aligned}$$

avec $\Phi_P(A) := \sum_{\mathbf{x} \in A} \Phi(\mathbf{x})$. *qfd*

Convolution

Nous pouvons maintenant l'opération qui nous permettra de construire des noyaux pour séquences:

Definition 5 Soient K_1 and K_2 deux n.d.p pour séquences.. La **convolution** de K_1 et K_2 , denotée $K_1 \star K_2$, est la fonctions suivante, pour $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$:

$$K_1 \star K_2(\mathbf{x}, \mathbf{y}) := \sum_{\mathbf{x}_1 \mathbf{x}_2 = \mathbf{x}, \mathbf{y}_1 \mathbf{y}_2 = \mathbf{y}} K_1(\mathbf{x}_1, \mathbf{y}_1) K_2(\mathbf{x}_2, \mathbf{y}_2).$$

Theoreme 6 Si K_1 et K_2 sont des n.d.p., alors $K_1 \star K_2$ est aussi un n.d.p.

Preuve

Soit \mathcal{X} l'ensemble des séquences finies. Pour $\mathbf{x} \in \mathcal{X}$, soit

$$R(\mathbf{x}) = \{(\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{X} \times \mathcal{X} : \mathbf{x} = \mathbf{x}_1 \mathbf{x}_2\} \subset \mathcal{X} \times \mathcal{X}.$$

On peut alors écrire

$$K_1 \star K_2(\mathbf{x}, \mathbf{y}) = \sum_{(\mathbf{x}_1, \mathbf{x}_2) \in R(\mathbf{x})} \sum_{(\mathbf{y}_1, \mathbf{y}_2) \in R(\mathbf{y})} K_1(\mathbf{x}_1, \mathbf{y}_1) K_2(\mathbf{x}_2, \mathbf{y}_2)$$

qui est un n.d.p. par les lemmes 3 et 4. \square

3 noyaux de bases pour séquences

Pour toutes séquences $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, définissons:

- Un noyau trivial:

$$K_0(\mathbf{x}, \mathbf{y}) := 1.$$

- Un noyau pour lettres seules

$$K_a^{(\beta)}(\mathbf{x}, \mathbf{y}) := \begin{cases} 0 & \text{si } |\mathbf{x}| \neq 1 \text{ ou } |\mathbf{y}| \neq 1, \\ \exp(\beta S(\mathbf{x}, \mathbf{y})) & \text{sinon} \end{cases}$$

- Un noyau pour gaps:

$$K_g^{(\beta)}(\mathbf{x}, \mathbf{y}) = \exp[\beta (g(|\mathbf{x}|) + g(|\mathbf{y}|))]$$

Remarques

- $S : \mathcal{A}^2 \rightarrow \mathbb{R}$ est la fonction de similarité entre lettres utilisée dans le score d'alignement. $K_a^{(\beta)}$ est un n.d.p. seulement quand la matrice:

$$\left(\exp(\beta s(a, b)) \right)_{(a,b) \in \mathcal{A}^2}$$

est semi-définie positive.

- g est la fonction utilisée pour pénaliser les gaps dans le score d'alignement. Le noyau pour gap est toujours d.p. car il s'écrit simplement comme un produit scalaire:

$$K_g^{(\beta)}(\mathbf{x}, \mathbf{y}) = \exp(\beta g(|\mathbf{x}|)) \times \exp(\beta g(|\mathbf{y}|))$$

Convolution des noyaux

- Pour chaque $n \in \mathbb{N}$, on définit un noyau pour alignement de n lettres:

$$K_{(n)}^{(\beta)} = K_0 \star \left(K_a^{(\beta)} \star K_g^{(\beta)} \right)^{(n-1)} \star K_a^{(\beta)} \star K_0.$$

- Le *noyau d'alignement local* est alors défini par:

$$K_{LA}^{(\beta)} = \sum_{i=0}^{\infty} K_{(i)}^{(\beta)}.$$

Remarques

- $K_{(n)}^{(\beta)}$ est un n.d.p. en tant que convolué de n.d.p. (théoreme 6).
- $K_{LA}^{(\beta)}$ est un n.d.p. en tant que limite ponctuelle de n.d.p. (lemme 2)

Lien avec le score d'alignement

Theoreme 7 *Le n.d.p. $K_{LA}^{(\beta)}$ vérifie, pour tout $\mathbf{x}, \mathbf{y} \in \mathcal{X}$:*

$$K_{LA}^{(\beta)}(\mathbf{x}, \mathbf{y}) = \sum_{\pi \in \Pi(\mathbf{x}, \mathbf{y})} \exp(\beta s(\mathbf{x}, \mathbf{y}, \pi)),$$

et

$$\lim_{\beta \rightarrow +\infty} \frac{1}{\beta} \ln K_{LA}^{(\beta)}(\mathbf{x}, \mathbf{y}) = SW_{S,g}(\mathbf{x}, \mathbf{y}).$$

Remarques

- Le noyau d'alignement local est calculé à partir de scores d'alignements censés avoir un sens biologique
- Ce théorème montre pourquoi le score de SW n'est pas un n.d.p.:
 - il ne considère que le score du meilleur alignement, au lieu de moyenniser sur tous les alignements
 - il faudrait prendre son exponentielle pour se rapprocher du noyau d'alignement local.

Preuve du théorème 7

Pour deux séquences $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, soit:

$$\Pi_n^0(\mathbf{x}, \mathbf{y}) := \{\pi \in \Pi_n(\mathbf{x}, \mathbf{y}) : \pi_1(n) = |\mathbf{x}|, \pi_2(n) = |\mathbf{y}|\}.$$

On va alors utiliser le

Lemme 8 *Pour $n \geq 1$, on a:*

$$\sum_{\pi \in \Pi_n^0(\mathbf{x}, \mathbf{y})} \exp(\beta s(\mathbf{x}, \mathbf{y}, \pi)) = K_0 \star K_a \star (K_g \star K_a)^{n-1}(\mathbf{x}, \mathbf{y}),$$

et pour $n = 0$:

$$\sum_{\pi \in \Pi_0^0(\mathbf{x}, \mathbf{y})} \exp(\beta s(\mathbf{x}, \mathbf{y}, \pi)) = K_0(\mathbf{x}, \mathbf{y})$$

Preuve du théoreme 7

On déduit le théoreme 7 du lemme 8 en écrivant:

$$\begin{aligned} & \sum_{\pi \in \Pi_n(\mathbf{x}, \mathbf{y})} \exp(\beta s(\mathbf{x}, \mathbf{y}, \pi)) \\ &= \sum_{\mathbf{x}=\mathbf{x}_1 \mathbf{x}_2, \mathbf{y}=\mathbf{y}_1 \mathbf{y}_2} \sum_{\pi \in \Pi_n^0(\mathbf{x}_1, \mathbf{y}_1)} \exp(\beta s(\mathbf{x}_1, \mathbf{y}_1, \pi)) \\ &= \sum_{\mathbf{x}=\mathbf{x}_1 \mathbf{x}_2, \mathbf{y}=\mathbf{y}_1 \mathbf{y}_2} K_0 \star K_a \star (K_g \star K_a)^{n-1}(\mathbf{x}_1, \mathbf{y}_1) \\ &= K_0 \star K_a \star (K_g \star K_a)^{n-1} \star K_0(\mathbf{x}, \mathbf{y}) \quad \square \end{aligned}$$

Preuve du lemme 8

On prouve le lemme 8 par récurrence sur n .

- Il est trivial pour $n = 0$ (le score d'un alignement de 0 lettre étant nul).
- Pour $n = 1$, $\Pi_n^0(\mathbf{x}, \mathbf{y})$ est réduit au singleton $\{(|\mathbf{x}|, |\mathbf{y}|)\}$ (seules les dernières lettres sont alignées), qui a un score $S(x_{|\mathbf{x}|}, y_{|\mathbf{y}|})$. D'autre part,

$$\begin{aligned} K_0 \star K_a(\mathbf{x}, \mathbf{y}) &= \sum_{\mathbf{x}_1 \mathbf{x}_2 = \mathbf{x}, \mathbf{y}_1 \mathbf{y}_2 = \mathbf{y}} K_0(\mathbf{x}_1, \mathbf{y}_1) K_a(\mathbf{x}_2, \mathbf{y}_2) \\ &= \exp [\beta S(x_{|\mathbf{x}|}, y_{|\mathbf{y}|})] . \end{aligned}$$

Preuve du lemme 8

Supposons le lemme valide pour $n - 1$. Pour tout $\pi \in \Pi_n^0(\mathbf{x}, \mathbf{y})$, soit $f(\pi) \in \Pi_n(\mathbf{x}, \mathbf{y})$ l'alignement obtenu en retirant les dernières lettres alignées dans π . On a alors:

$$\begin{aligned} & \sum_{\pi \in \Pi_n^0(\mathbf{x}, \mathbf{y})} \exp(\beta s(\mathbf{x}, \mathbf{y}, \pi)) \\ &= \sum_{\mathbf{x}=\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3, \mathbf{y}=\mathbf{y}_1\mathbf{y}_2\mathbf{y}_3} \sum_{\pi' \in \Pi_{n-1}^0(\mathbf{x}_1, \mathbf{y}_1)} \exp(\beta s(\mathbf{x}_1, \mathbf{y}_1, f^{-1}(\pi'))), \end{aligned}$$

ou la somme porte porte sur toutes les décompositions telles que $|\mathbf{x}_3| = |\mathbf{y}_3| = 1$.

Preuve du lemme 8

Pour une telle décomposition $\mathbf{x} = \mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$, le score d'alignement est par définition:

$$s(\mathbf{x}, \mathbf{y}, \pi) = s(\mathbf{x}_1, \mathbf{y}_1, f(\pi)) + g(|\mathbf{x}_2|) + g(|\mathbf{y}_2|) + S(\mathbf{x}_3, \mathbf{y}_3),$$

donc:

$$\exp(\beta s(\mathbf{x}, \mathbf{y}, \pi)) = \exp(\beta s(\mathbf{x}_1, \mathbf{y}_1, f(\pi))) K_g(\mathbf{x}_2, \mathbf{y}_2) K_a(\mathbf{x}_3, \mathbf{y}_3).$$

Preuve du lemme 8

Le noyau K_a étant nul pour les séquences non réduites a une lettre, on en déduit que:

$$\begin{aligned} & \sum_{\pi \in \Pi_n^0(\mathbf{x}, \mathbf{y})} \exp(\beta s(\mathbf{x}, \mathbf{y}, \pi)) \\ &= \sum_{\mathbf{x}=\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3, \mathbf{y}=\mathbf{y}_1\mathbf{y}_2\mathbf{y}_3} \sum_{\pi \in \Pi_{n-1}^0(\mathbf{x}_1, \mathbf{y}_1)} \exp(\beta s(\mathbf{x}_1, \mathbf{y}_1, f(\pi))) \\ & \quad \times K_g(\mathbf{x}_2, \mathbf{y}_2) K_a(\mathbf{x}_3, \mathbf{y}_3), \end{aligned}$$

ou la somme porte sur toutes les décompositions possible de \mathbf{x} et \mathbf{y} en $\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$ et $\mathbf{y}_1\mathbf{y}_2\mathbf{y}_3$.

Preuve du lemme 8

En utilisant l'hypothese de récurrence on en déduit:

$$\begin{aligned} & \sum_{\pi \in \Pi_n^0(\mathbf{x}, \mathbf{y})} \exp(\beta s(\mathbf{x}, \mathbf{y}, \pi)) \\ &= \sum_{\mathbf{x}=\mathbf{x}_1 \mathbf{x}_2 \mathbf{x}_3, \mathbf{y}=\mathbf{y}_1 \mathbf{y}_2 \mathbf{y}_3} \sum_{\pi \in \Pi_{n-1}^0(\mathbf{x}_1, \mathbf{y}_1)} \exp(\beta s(\mathbf{x}_1, \mathbf{y}_1, f(\pi))) \\ & \quad \times K_g(\mathbf{x}_2, \mathbf{y}_2) K_a(\mathbf{x}_3, \mathbf{y}_3) \\ &= \sum_{\mathbf{x}=\mathbf{x}_1 \mathbf{x}_2 \mathbf{x}_3, \mathbf{y}=\mathbf{y}_1 \mathbf{y}_2 \mathbf{y}_3} K_0 \star K_a \star (K_g \star K_a)^{n-2}(\mathbf{x}_1, \mathbf{y}_1) \\ & \quad \times K_g(\mathbf{x}_2, \mathbf{y}_2) K_a(\mathbf{x}_3, \mathbf{y}_3) \\ &= K_0 \star K_a \star (K_g \star K_a)^{n-1}(\mathbf{x}, \mathbf{y}) \quad \square \end{aligned}$$

Implémentation

- Le score d'alignement peut se calculer en $O(|x||y|)$ par programmation dynamique.
- Le noyau d'alignement aussi!, avec de tres légere modifications.

On suppose que la fonction de gap est affine:

$$\begin{cases} g(0) & = 0, \\ g(n) & = d + e(n - 1) \text{ si } n \geq 1, \end{cases}$$

Implémentation

Le noyau peut se calculer par:

$$K_{LA}^{(\beta)}(\mathbf{x}, \mathbf{y}) = 1 + X_2(|\mathbf{x}|, |\mathbf{y}|) + Y_2(|\mathbf{x}|, |\mathbf{y}|) + M(|\mathbf{x}|, |\mathbf{y}|),$$

ou $M(i, j)$, $X(i, j)$, $Y(i, j)$, $X_2(i, j)$, et $Y_2(i, j)$ pour $0 \leq i \leq |\mathbf{x}|$,
et $0 \leq j \leq |\mathbf{y}|$ sont définis récursivement par:

$$\left\{ \begin{array}{l} M(i, 0) = M(0, j) = 0, \\ X(i, 0) = X(0, j) = 0, \\ Y(i, 0) = Y(0, j) = 0, \\ X_2(i, 0) = X_2(0, j) = 0, \\ Y_2(i, 0) = Y_2(0, j) = 0, \end{array} \right.$$

Implémentation

...et pour $i = 1, \dots, |\mathbf{x}|$ et $j = 1, \dots, |\mathbf{y}|$:

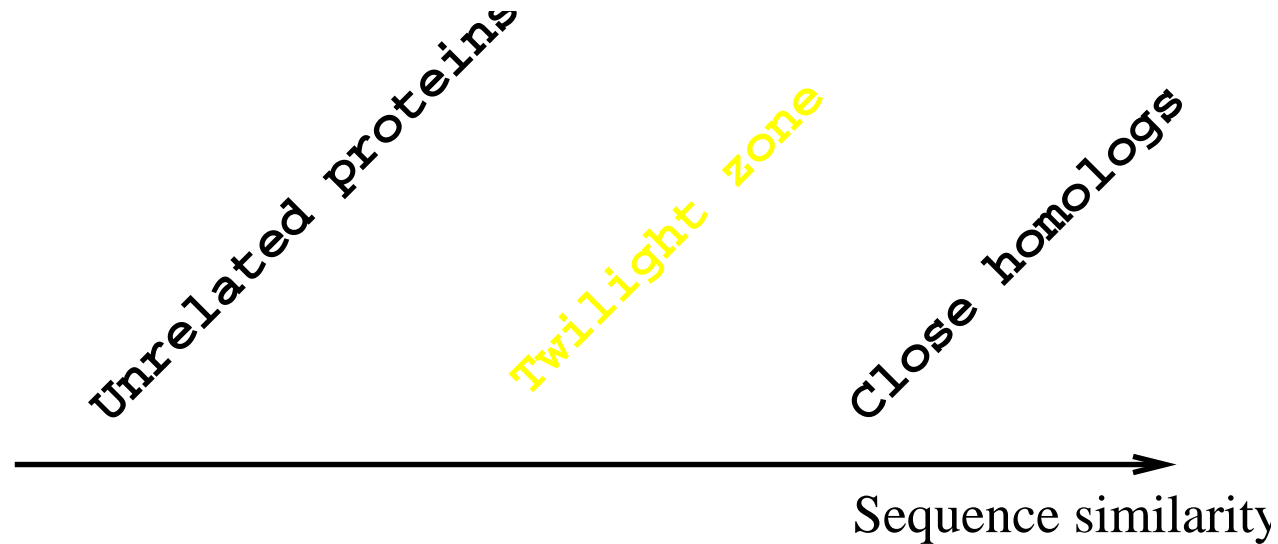
$$\left\{ \begin{array}{l} M(i, j) = \exp(\beta S(x_i, y_j)) \left[1 + X(i-1, j-1) \right. \\ \qquad \qquad \qquad \left. + Y(i-1, j-1) + M(i-1, j-1) \right], \\ X(i, j) = \exp(\beta d) M(i-1, j) + \exp(\beta e) X(i-1, j), \\ Y(i, j) = \exp(\beta d) [M(i, j-1) + X(i, j-1)] \\ \qquad \qquad \qquad + \exp(\beta e) Y(i, j-1), \\ X_2(i, j) = M(i-1, j) + X_2(i-1, j), \\ Y_2(i, j) = M(i, j-1) + X_2(i, j-1) + Y_2(i, j-1). \end{array} \right.$$

Implémentation: preuve

- La preuve de la validité de l'implémentation se fait par récurrence, voir Vert et al. (2004).
- Laissezée en exercice...

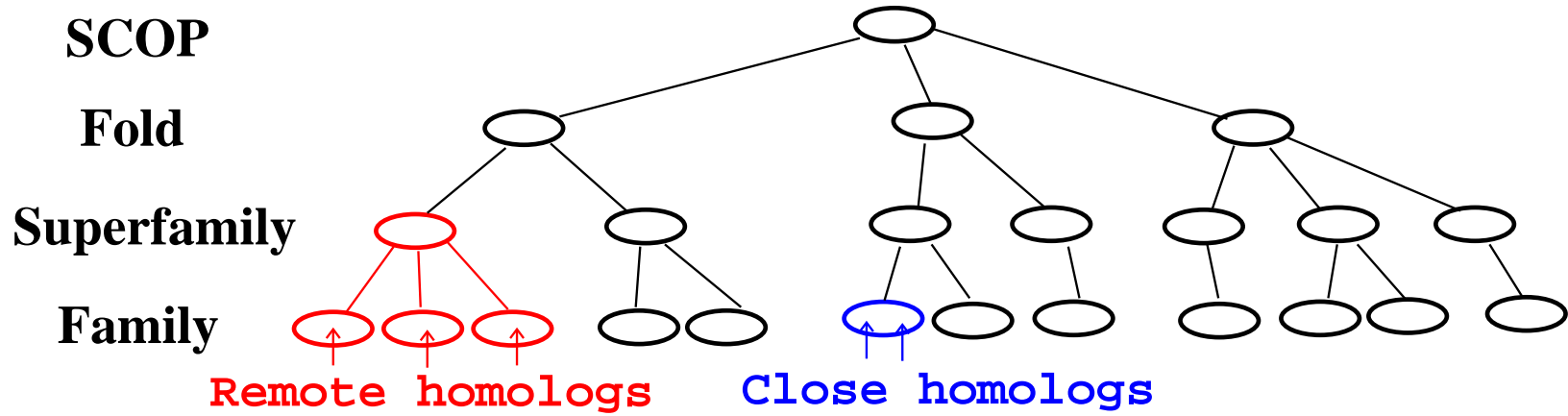
Application: détection d'homologie lointaine

Le probleme de l'homologie lointaine



- Same structure/function but sequence diverged
- *Remote homology can not be found by direct sequence similarity*

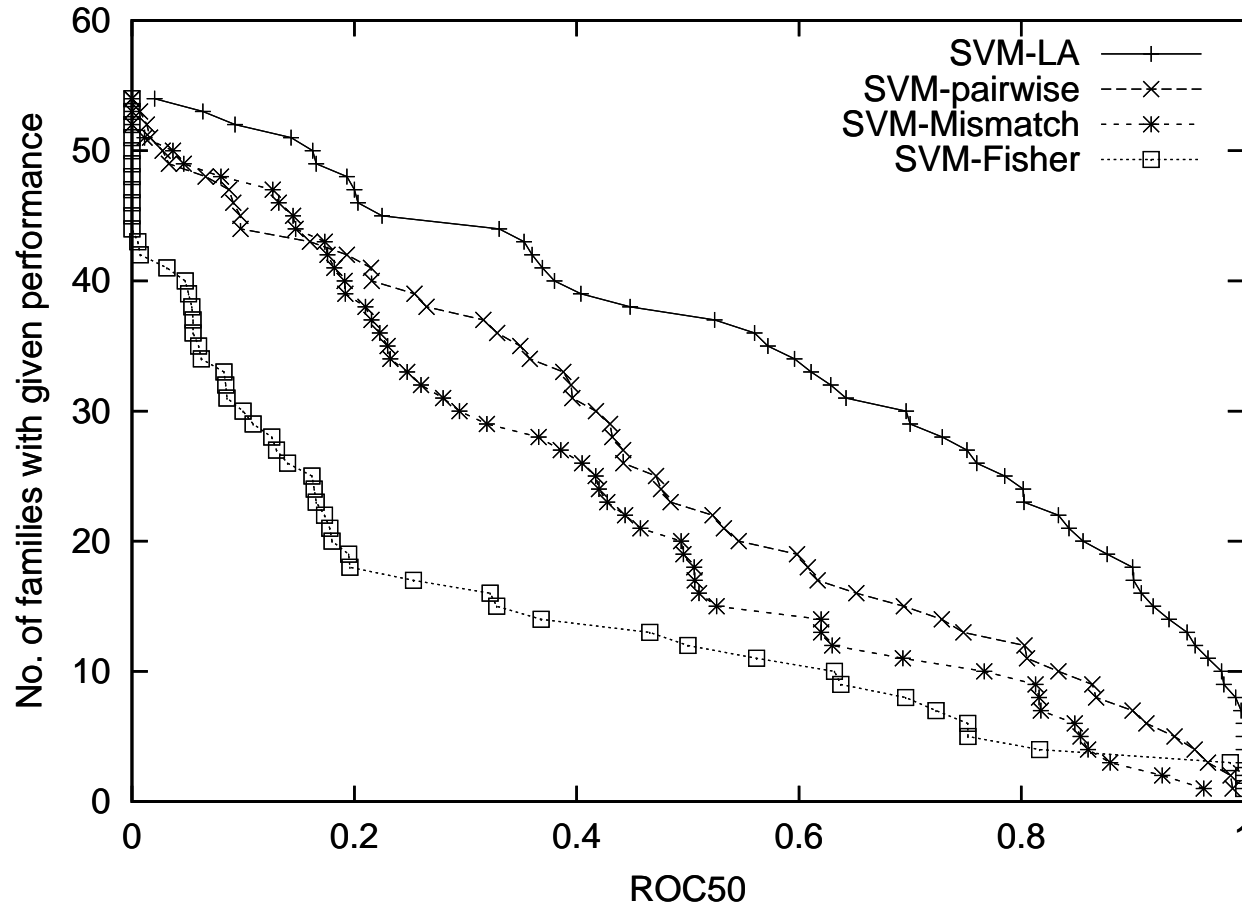
SCOP database



A benchmark experiment

- Can we predict the *superfamily* of a domain if we have not seen any member of its *family* before?
- During *learning*: remove a family and learn the difference between the superfamily and the rest
- Then, use the model to *test* each domain of the family removed

SCOP superfamily recognition benchmark



Remarque

- Les valeurs du noyau LA sont exponentiellement petite. En pratique on a du tricher et prendre leur logarithme. Le résultat n'est plus un n.d.p., mais on s'en est sorti avec des procédures ad hoc pour rendre la matrice de Gram symétrique définie positive.
- Le noyau d'alignement local est actuellement la méthode la plus performante pour cette expérience.

Noyaux marginalisés

Motivations

- On se place dans le cas où les données $x \in \mathcal{X}$ ne sont que la *partie "visible" d'objets plus complexes* $z = (x, y) \in \mathcal{Z} := \mathcal{X} \cup \mathcal{Y}$.
- $y \in \mathcal{Y}$ n'est *pas observé*, mais contient beaucoup d'*information pertinente* sur x (ex: x est une phrase, y sa structure grammaticale)
- On a un noyau naturel sur \mathcal{Z} (*les données complètes*).
- Comment en déduire un noyau sur \mathcal{X} (*les données observées*)?

Cadre probabiliste

- Etant donné une observation $\mathbf{x} \in \mathcal{X}$, la donnée cachée correspondante $y \in \mathcal{Y}$ ne peut pas être déduite de manière certaine.
- On se place donc dans un *cadre probabiliste*: $(\mathcal{Y}, \mathcal{B})$ est un espace mesurable, et on suppose que pour chaque $\mathbf{x} \in \mathcal{X}$, on a une probabilité "conditionnelle"
 $P_{\mathbf{x}} \in \mathcal{M}_1^+(\mathcal{Y}, \mathcal{B})$ (une mesure positive de masse 1).

Noyau marginalisé

Definition 9 Si $K_{\mathcal{Z}}$ est une fonction sur $\mathcal{Z}^2 = (\mathcal{X} \times \mathcal{Y})^2$, mesurable en tant que fonction de y pour tout x , l'opération de **marginalisation** consiste à définir la fonction $K_{\mathcal{X}}$ sur \mathcal{X}^2 par:

$$\begin{aligned} K_{\mathcal{X}}(\mathbf{x}, \mathbf{x}') &:= E_{P_{\mathbf{x}}(d\mathbf{y}) \times P_{\mathbf{x}'}(d\mathbf{y}')} K_{\mathcal{Z}}(\mathbf{z}, \mathbf{z}') \\ &= \int \int K_{\mathcal{Z}}((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) P_{\mathbf{x}}(d\mathbf{y}) P_{\mathbf{x}'}(d\mathbf{y}') \end{aligned}$$

Theoreme 10 Si $K_{\mathcal{Z}}$ est un noyau défini positif sur \mathcal{Z} , alors $K_{\mathcal{X}}$ est un noyau défini positif sur \mathcal{X} appelé **noyau marginalisé**.

Preuve du théorème 1

Si $K_{\mathcal{Z}}$ est un n.d.p. sur \mathcal{Z} , alors il existe un espace de Hilbert \mathcal{H} et $\Phi_{\mathcal{Z}} : \mathcal{Z} \rightarrow \mathcal{H}$ tel que

$$K_{\mathcal{Z}}(\mathbf{z}, \mathbf{z}') = \langle \Phi_{\mathcal{Z}}(\mathbf{z}), \Phi_{\mathcal{Z}}(\mathbf{z}') \rangle_{\mathcal{H}}.$$

La marginalisation donne donc:

$$\begin{aligned} K_{\mathcal{X}}(\mathbf{x}, \mathbf{x}') &= E_{P_{\mathbf{x}}(d\mathbf{y}) \times P_{\mathbf{x}'}(d\mathbf{y}')} K_{\mathcal{Z}}(\mathbf{z}, \mathbf{z}') \\ &= E_{P_{\mathbf{x}}(d\mathbf{y}) \times P_{\mathbf{x}'}(d\mathbf{y}')} \langle \Phi_{\mathcal{Z}}(\mathbf{z}), \Phi_{\mathcal{Z}}(\mathbf{z}') \rangle_{\mathcal{H}} \\ &= \langle E_{P_{\mathbf{x}}(d\mathbf{y})} \Phi_{\mathcal{Z}}(\mathbf{z}), E_{P_{\mathbf{x}'}(d\mathbf{y}')} \Phi_{\mathcal{Z}}(\mathbf{z}') \rangle_{\mathcal{H}} \end{aligned}$$

donc $K_{\mathcal{X}}$ est un n.d.p. sur \mathcal{X} . \square

Noyaux pour séquences par chaînes de Markov cachées

Motivations

- Les chaînes de Markov cachées (*HMM*: hidden Markov models) sont *extrêmement utilisées* pour modéliser des séquences biologiques
- Les états cachés (voir les slides suivants) ont en général un *sens biologique*
- La marginalisation va permettre de déduire un n.d.p. basé sur cette information biologique codée dans le modèle probabiliste
- Cette applications n'est pas restreinte aux séquences biologiques: les HMM sont très utilisées dans de nombreux autres domaines (reconnaissance vocale, traitements du signal,...).

Chaîne de Markov

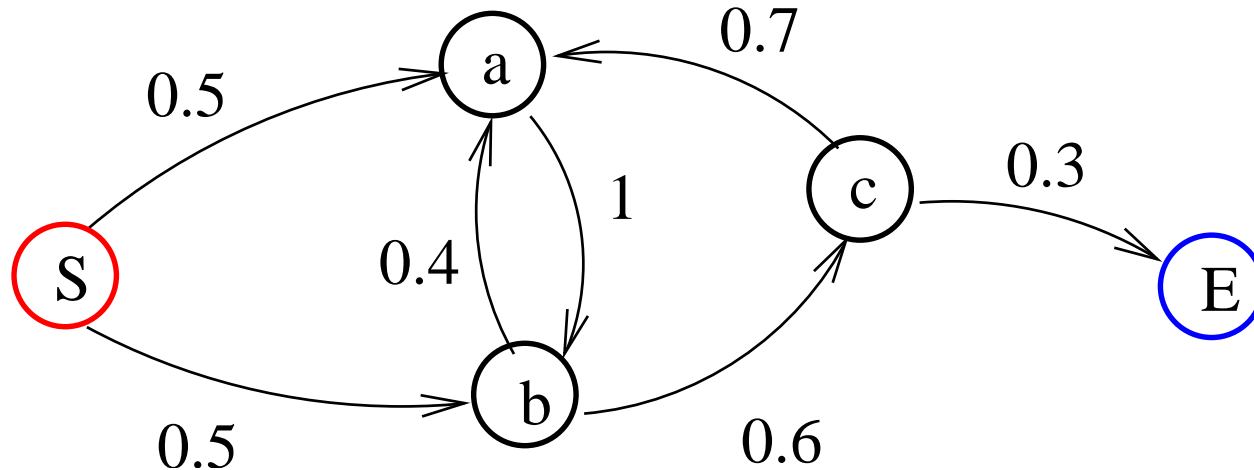
Soit \mathcal{S} un ensemble fini, comprenant un élément S (Start) et un élément E (End). Une chaîne de Markov est une distribution de probabilité P sur $\mathcal{Y} = \mathcal{S}^*$ définie par:

$$P(y_0 y_1 \dots y_n) = \begin{cases} 0 & \text{si } y_0 \neq S \text{ ou } y_n \neq E \text{ ou } y_i = E \text{ pour } i < n, \\ \prod_{i=1}^n p(y_i | y_{i-1}) & \text{sinon,} \end{cases}$$

avec pour tout $a, b \in \mathcal{Y}$

$$\begin{cases} p(a|b) \geq 0 \\ \sum_{c \in \mathcal{S}} p(c|a) = 1. \end{cases}$$

Exemple



Calcul de la probabilité d'une séquence sous ce modèle:

$$P(SababcE) = 0.5 \times 1 \times 0.4 \times 1 \times 0.6 \times 0.3 = 0.0036$$

Chaîne de Markov cachée (HMM)

On n'observe pas la suite des états, qui forment un chaîne de Markov. Par contre, on suppose que chaque état $y \in \mathcal{S}$ (sauf S et E) émet, indépendamment des autres, une variable $x \in \mathcal{A}$ (fini) que l'on observe.

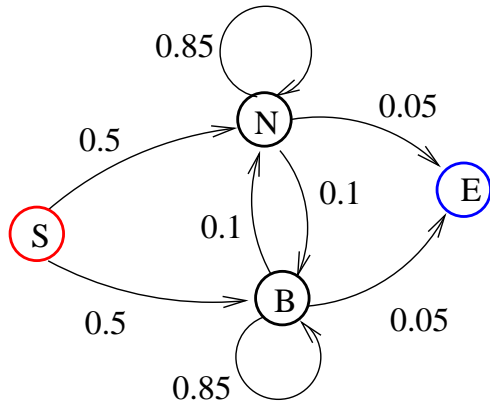
On obtient donc une probabilité sur les paires $(\mathbf{x}, \mathbf{y}) \in \mathcal{A}^* \times \mathcal{S}^*$ définie par:

$$P(x_1 \dots x_n, y_0 \dots y_{n+1}) = p(y_0 \dots y_{n+1}) \times \prod_{i=1}^n \pi(x_i | y_i),$$

ou π est la probabilité conditionnelle d'émission

$$(\sum_{x \in \mathcal{A}} \pi(x|y) = 1 \text{ pour } y \in \mathcal{S}).$$

Exemple: pile ou face biaisé



- Une pièce normale N , une biaisée B (non observé)

- On observe pile (0) ou face (1) avec probabilités:

$$\begin{cases} \pi(0|N) = 1 - \pi(1|N) = 0.5, \\ \pi(0|B) = 1 - \pi(1|B) = 0.8. \end{cases}$$

- Exemple de réalisation:

NNNNNBBBBBBBBBNNNNNNNNNNNBBBBBB
1001011101111010010111001111011

Noyau pour données observées

- Si on n'observe que $\mathbf{x} \in \mathcal{A}^*$, le n.d.p. le plus simple est le 1-spectral:

$$K(\mathbf{x}, \mathbf{x}') = \sum_{a \in \mathcal{A}} n_a(\mathbf{x}) n_a(\mathbf{x}'),$$

ou $n_a(\mathbf{x})$ est le nombre d'occurrences de a dans \mathbf{x} .

- Exemple:

$$\begin{aligned} \mathbf{x} &= 1001011101111010010111001111011, \\ \mathbf{x}' &= 0011010110011111011010111101100101, \end{aligned}$$

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= n_0(\mathbf{x}) n_0(\mathbf{x}') + n_1(\mathbf{x}) n_1(\mathbf{x}') \\ &= 11 \times 13 + 20 \times 21 = 563. \end{aligned}$$

Noyaux pour données totales

- Si on observait les données totales $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in \mathcal{A}^* \times \mathcal{S}^*$, on pourrait utiliser le noyau:

$$K_{\mathcal{Z}}(\mathbf{z}, \mathbf{z}') = \sum_{(a,s) \in \mathcal{A} \times \mathcal{S}} n_{a,s}(\mathbf{z}) n_{a,s}(\mathbf{z}'),$$

ou $n_{a,s}(\mathbf{x}, \mathbf{y})$ est le nombre d'occurrences de s dans \mathbf{y} qui émettent a dans \mathbf{x} .

- Exemple:

$$\begin{aligned} \mathbf{z} &= 1001011101111010010111001111011, \\ \mathbf{z}' &= 0011010110011111011010111101100101, \end{aligned}$$

$$\begin{aligned} k(\mathbf{z}, \mathbf{z}') &= n_0(\mathbf{z}) n_0(\mathbf{z}') + n_1(\mathbf{z}) n_1(\mathbf{z}') + n_2(\mathbf{z}) n_2(\mathbf{z}') + n_3(\mathbf{z}) n_3(\mathbf{z}') \\ &= 7 \times 15 + 9 \times 12 + 13 \times 6 + 2 \times 1 = 293. \end{aligned}$$

Noyau marginalisé

Le noyau marginalisé pour données observées est défini par:

$$\begin{aligned} K_{\mathcal{X}}(\mathbf{x}, \mathbf{x}') &= \sum_{\mathbf{y}, \mathbf{y}' \in \mathcal{S}^*} K_{\mathcal{Z}}((\mathbf{x}, \mathbf{y}), (\mathbf{x}, \mathbf{y}')) P(\mathbf{y}|\mathbf{x}) P(\mathbf{y}'|\mathbf{x}') \\ &= \sum_{(a,s) \in \mathcal{A} \times \mathcal{S}} \Phi_{a,s}(\mathbf{x}) \Phi_{a,s}(\mathbf{x}'), \end{aligned}$$

avec

$$\Phi_{a,s}(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{S}^*} P(\mathbf{y}|\mathbf{x}) n_{a,s}(\mathbf{x}, \mathbf{y})$$

Calcul du noyau marginalisé

Nous allons calculer explicitement $\Phi_{a,s}(\mathbf{x})$, pour tout $(a, s) \in \mathcal{A} \times \mathcal{S}$. Supposons

$$\mathbf{x} = x_1 \dots x_n \in \mathcal{A}^n.$$

Alors $P(\mathbf{y}|\mathbf{x}) > 0$ implique \mathbf{y} soit de la forme:

$$\mathbf{y} = Sy_1 \dots y_n E \in \mathcal{S}^{n+2}.$$

On a alors, en notant δ est le symbole de Kronecker ($\delta(u, v) = 1$ si $u = v$, 0 sinon):

$$n_{a,s}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n \delta(x_i, a) \delta(y_i, b),$$

Calcul du noyau marginalisé

On en déduit:

$$\begin{aligned}\Phi_{a,s}(\mathbf{x}) &= \sum_{\mathbf{y} \in \mathcal{S}^*} P(\mathbf{y}|\mathbf{x}) n_{a,s}(\mathbf{x}, \mathbf{y}) \\ &= \sum_{\mathbf{y} \in \mathcal{S}^*} P(\mathbf{y}|\mathbf{x}) \left\{ \sum_{i=1}^n \delta(x_i, a) \delta(y_i, s) \right\} \\ &= \sum_{i=1}^n \delta(x_i, a) \left\{ \sum_{\mathbf{y} \in \mathcal{S}^*} P(\mathbf{y}|\mathbf{x}) \delta(y_i, s) \right\} \\ &= \sum_{i=1}^n \delta(x_i, a) P(y_i = s|\mathbf{x}).\end{aligned}$$

Calcul de $P(y_i = s | \mathbf{x})$

Il existe une méthode classique pour calculer $P(y_i = s | \mathbf{x})$ de manière efficace: l'algorithme *Forward-Backward*. Soient:

$$\forall (s, i) \in \mathcal{S} \times [1, n], \quad \begin{cases} f_s(i) := P(x_1 \dots x_i, y_i = s), \\ b_s(i) := P(x_{i+1} \dots x_n | y_i = s) \end{cases}$$

Le calcul de $P(y_i = s | \mathbf{x})$ se ramène à celui de $f_s(i)$ et $b_s(i)$, car:

$$\begin{aligned} P(y_i = s | \mathbf{x}) &= \frac{P(x_1 \dots x_n, y_i = s)}{P(\mathbf{x})} \\ &= \frac{f_s(i)b_s(i)}{\sum_{s' \in \mathcal{S}} f_{s'}(i)b_{s'}(i)} \end{aligned}$$

Calcul de $f_s(i)$

$f_s(i)$ (forward) se calcule *récurivement* pour par:

$$\forall t \in \mathcal{S}, \quad f_t(1) = P(x_1, y_1 = t) = p(t|S)\pi(x_1|t),$$

et pour $t \in \mathcal{S}$ et $j = 1, \dots, i$:

$$f_t(j) = P(x_1 \dots x_j, y_j = t)$$

$$= \sum_{u \in \mathcal{S}} P(x_1 \dots x_j, y_{j-1} = u, y_j = t)$$

$$= \sum_{u \in \mathcal{S}} P(x_1 \dots x_{j-1}, y_{j-1} = u) P(x_j, y_j = t | y_{j-1} = u)$$

$$= \sum_{u \in \mathcal{S}} f_u(j-1) \times p(t|u)\pi(x_j|u).$$

Calcul de $b_s(i)$

De même $b_s(i)$ (backward) se calcule par:

$$\forall t \in \mathcal{S}, \quad b_t(n) = 1,$$

et pour $t \in \mathcal{S}$ et $j = n, n - 1, \dots, i$:

$$\begin{aligned} b_t(j) &= P(x_{j+1} \dots x_n | y_j = t) \\ &= \sum_{u \in \mathcal{S}} P(x_{j+1} \dots x_n, y_{j+1} = u | y_j = t) \\ &= \sum_{u \in \mathcal{S}} P(x_{j+1}, y_{j+1} = u | y_j = t) P(x_{j+2} \dots x_n | y_{j+1} = u) \\ &= \sum_{u \in \mathcal{S}} b_u(j+1) \times p(u|t) \pi(x_{j+1}|u). \end{aligned}$$

Remarque

Les algorithmes forward et backward sont les algorithmes de bases des HMM. On peut en particulier calculer la probabilité d'une séquence $\mathbf{x} = x_1 \dots x_n$ par:

$$P(\mathbf{x}) = \sum_{s \in \mathcal{S}} P(x_1 \dots x_n, y_n = s) = \sum_{s \in \mathcal{S}} f_s(n).$$

Extension

- Plutôt que le noyau 1-spectral, peut-on marginaliser des noyaux spectraux d'ordre supérieur?
- Par exemple:

$$K_{\mathcal{Z}}(\mathbf{z}, \mathbf{z}') = \sum_{(a,b,s,t) \in \mathcal{A}^2 \times \mathcal{S}^2} n_{a,b,s,t}(\mathbf{z}) n_{a,b,s,t}(\mathbf{z}'),$$

ou $n_{a,b,s,t}(\mathbf{z})$ est le nombre d'occurrences du 2-mer st dans y qui émettent ab .

- Intérêt: encoder plus d'information dans le noyau

Extension (exercice)

Le noyau 2-spectral peut se marginaliser en:

$$K_2(\mathbf{x}, \mathbf{x}') = \sum_{(a,b,s,t) \in \mathcal{A}^2 \times \mathcal{S}^2} \Phi_{a,b,s,t}(\mathbf{x}) \Phi_{a,b,s,t}(\mathbf{x}'),$$

avec:

$$\Phi_{a,b,s,t}(\mathbf{x}) = \sum_{i=1}^{n-1} \delta(x_i, a) \delta(x_{i+1}, b) P(y_i = s, y_{i+1} = t | \mathbf{x}),$$

calculable par:

$$P(y_i = s, y_{i+1} = t | \mathbf{x}) = \frac{p(t|s)\pi(x_{i+1}|t)f_s(i)b_t(i+1)}{P(\mathbf{x})}.$$