Modèles probabilistes
Alignement de séquences d'ARN par la structure

T. Schiex (INRA, tschiex@toulouse.inra.fr)
avec l'aide de M. Goulard (INRA)
et de C. Lawrence (cf. www.wadsworth.org/res&res/bioinfo)

décembre 1997

Plan général

- Modèles probabilistes, inférence statistique (bayésienne)

- Alignement par la structure de séquences d'ARN

- Conclusion

Idées générales (bayésien)

- construction d'un modèle probabiliste de toutes les variables :
  - observées (séquences)
  - non observées (alignement...);

  impliquant un certain nombre de paramètres.

- incorporation de connaissances "biologiques" (*a priori*)

- inférence : exploitation de la distribution de probabilité sur les non observées (ou les paramètres) à partir des données (observées).

Distribution de probabilités

Probabilités positives, $\sum(p) = 1$

Tirage d'une pièce: $P(f) = q, P(p) = 1 - q$

Si on note $f = 1, p = 0$: $\qquad P(x) = q^x.(1-q)^{(1-x)}$

N tirages indépendants :

$$P(x_1, \ldots, x_N) = \prod_i P(x_i) = q^{\sum_i x_i}(1-q)^{N-\sum_i x_i}$$

Probabilité de k faces parmi N :

$$P(\#f = k) = C_k^N.q^k.(1-q)^{N-k} \quad \Rightarrow \quad \widehat{q} = \frac{k}{N}$$

Probabilités conditionnelles

$$P(X|Y)$$

Probabilité de X sachant la valeur de Y.

Si deux pièces sont "couplées" :

$$P(x_1 = 0|x_2 = 0) \neq P(x_1 = 0|x_2 = 1)$$

Marginales et conditionnelles

Pièce 1 (columns); Pièce 2 (rows)

	Face	Pile	Total
Face	71	27	98
Pile	35	67	102
Total	106	94	200

$$\widehat{P}(f_1, f_2) = \frac{71}{200}, \widehat{P}(f_1|f_2) = \frac{71}{98}, \widehat{P}(f_2) = \frac{98}{200}$$

$$P(x_1, x_2) = P(x_1|x_2)P(x_2), \quad P(x_1, x_2) = P(x_2|x_1)P(x_1)$$

Modèles probabilistes en bioinformatique

– nombreuses variables

– les observées (données) notées D

– les manquantes (alignement, segmentation...) notées M

– les paramètres du modèle (notés θ)

Il est généralement simple d'exprimer la probabilité des données complètes (avec les manquants) étant donnés les paramètres.

$$\underbrace{P(D, M, \theta)}_{jointe} = \underbrace{P(D, M|\theta)}_{vraisemblance} \cdot \underbrace{P(\theta)}_{a\ priori}$$

Problèmes essentiels

– calcul de la vraisemblance complète (manquants supposés connus)

– calcul de la vraisemblance incomplète = probabilité des données étant donnés les paramètres :

$$P(D|\theta) = \sum_M P(D, M|\theta)$$

calcul d'une somme sur un espace souvent discret et de très grande dimension

Problèmes essentiels

- estimation des paramètres ou des manquants :
  - Maximum de vraisemblance (ML) :

$$\theta \text{ qui maximise } P(D|\theta) = \sum_M P(D, M|\theta)$$

  - Maximum a posteriori (MAP) :

$$\theta \text{ qui maximise}$$
$$P(\theta|D) = \sum_M P(\theta, M|D) \propto \sum_M P(\theta, M, D) = \sum_M P(D, M|\theta).P(\theta)$$

- explorer les distributions *a posteriori* : nombreuses sommations, attention à la nature des priors.

Exemple

Deux pièces différentes sont lancées en séquence N fois en tout.

$$\text{PPFPPFFPPPFPFFFPFFFPFFPFFF}$$

- données : séquence de N pile/face
- manquants : le point où la pièce a changé
- paramètres : q_1 et q_2 (probabilités associées aux pièces)

$$P(D, M|q_1, q_2) = q_1^{\#f<M}.(1-q_1)^{\#p<M}.q_2^{\#f\geq M}.(1-q_2)^{\#p\geq M}$$

ML : maximiser $P(D|q_1, q_2) = \sum_M P(D, M|q_1, q_2)$
MAP : maximiser $P(q_1, q_2|D) \propto \sum_M P(D, M|q_1, q_2).P(q1, q2)$

Explorer certaines distributions marginales a posteriori : $P(M|D)$

Algorithmes

Besoin systématique : calcul de marginale, i.e., de méthode de calcul de sommes de probabilités sur des espaces discrets multi-dimensionnels (problèmes aisément #P-complets).

- l'exploration exhaustive (addition sur tous les cas possibles) pour les cas simples ;

- la programmation dynamique pour les cas bien structurés ;

- les algorithmes type MCMC (Gibbs Sampling, Metropolis-Hasting...) pour l'exploration (simulation) des posterior marginales.

Les algorithmes EM (Expectation/Maximization) permettent la maximisation de la vraisemblance des paramètres avec données manquantes mais nécessitent un calcul d'espérance (i.e., encore une fois une sommation).

Outils d'expression de vraisemblance

Nombreux modèles stochastiques basés sur des structures de dépendance "graphiques" (chaque arête/sommet reçoit une probabilité)

- Chaînes de markov cachées (HMM) : structure linéaire.
- Grammaire stochastiques :
  - régulières : structure linéaire.
  - context free (acyclique) : structure arborescente.
- réseaux bayésiens : graphe acyclique dirigé.
- automates stochastiques, CSP probabilistes, graphes pondérés

HMM

Observées et manquants définissent un “chemin” unique dans ces structures.

Efficacité des algorithmes liée à la structure du graphe (PD).

La PD permet aussi bien d’optimiser une somme (plus court chemin) que de sommer des produits (fiabilité de graphe, calcul de marginales) : voir [Minoux 76].

HMM

D= séquences, θ = probabilités du modèles, M = séquence d'états.

- alignement = trouver la séquence d'états maximisant $P(M|D,\theta)$.

  Soluble par la procédure de Viterbi (programmation dynamique).

- recherche = calcul de la probabilité d'une séquence $P(D|\theta)$.

  Soluble par la programmation dynamique.

- estimation = trouver les θ maximisant $P(D|\theta)$.

  Algorithme de Baum-Welch (EM). Algorithme limité à l'optimisation de paramètres continus, atteinte d'un maximum local. Utilise la programmation dynamique.

ARN : approche de Eddy et Durbin

Une structure sans pseudo-nœuds (pu autre interaction complexe) se modélise sous forme d'arbre. Permet de conserver la programmation dynamique (voir [Bertele,Brioshi 72]).

Modèles de covariance

Extension des HMM aux arbres. Ne permet pas la prise en compte de pseudo-nœuds.

Proba. de transition $\mathcal{T}(y_{next}|y)$, Proba. d'émission $\mathcal{P}(x_i, x_j|y)$

Problèmes et algorithmes

- Alignement d'une séquence au modèle : extension de l'algorithme de Viterbi. En $O(n^2m)$ en espace et $O(n^3m)$ en temps. Adapté pour la recherche dans les banques.

- calcul de la probabilité d'une séquence : approximée par celle de son meilleur alignement.

- estimation des paramètres à partir d'une structure et de séquences via EM (approximation de Viterbi). Chaque itération nécessite l'alignement de toutes les séquences.

Alignement au modèle

Matrice $S_{i,j,y}$: alignement le plus probable de la sous-séquence $i \cdots j$ au sous-arbre enraciné à l'état y. On additionne des log-vraisemblance.

Relations de récurrence :

$$S_{i,j,y}(y = MATP) = \max_{y_{next}}[S_{i+1,j-1,y_{next}} + \log(\mathcal{T}(y_{next}|y)) + \log(\mathcal{P}(x_i, x_j|y))]$$

$$S_{i,j,y}(y = MATL, INSL) = \max_{y_{next}}[S_{i+1,j,y_{next}} + \log(\mathcal{T}(y_{next}|y)) + \log(\mathcal{P}(x_i|y))]$$

$$S_{i,j,y}(y = MATR, INSR) = \max_{y_{next}}[S_{i+1,j-1,y_{next}} + \log(\mathcal{T}(y_{next}|y)) + \log(\mathcal{P}(x_j|y))]$$

$$S_{i,j,y}(y = DEL) = \max_{y_{next}}[S_{i,j,y_{next}} + \log(\mathcal{T}(y_{next}|y))]$$

$$S_{i,j,y}(y = BIFURC) = \max_{i-1 \leq k \leq j}[S_{i,k,y_{left}} + S_{k+1,j,y_{right}}]$$

$S_{1,N}$ donne le log-vraisemblance du meilleur alignement.

EM

EM nécessite un calcul d'espérance (E) qui dans ce cas précis demanderait le calcul du nombre moyen de passage par chacune des arêtes/sommets du modèle pour tous les alignements possibles de toutes les séquences.

Hypothèse de Viterbi : on remplace cette espérance par le nombre de passage effectif des meilleurs alignements. Effets secondaires ?

Étape (M): il faut maximiser la vraisemblance sous l'hypothèse que les données manquantes sont disponibles (on utilise le résultat de la phase E):

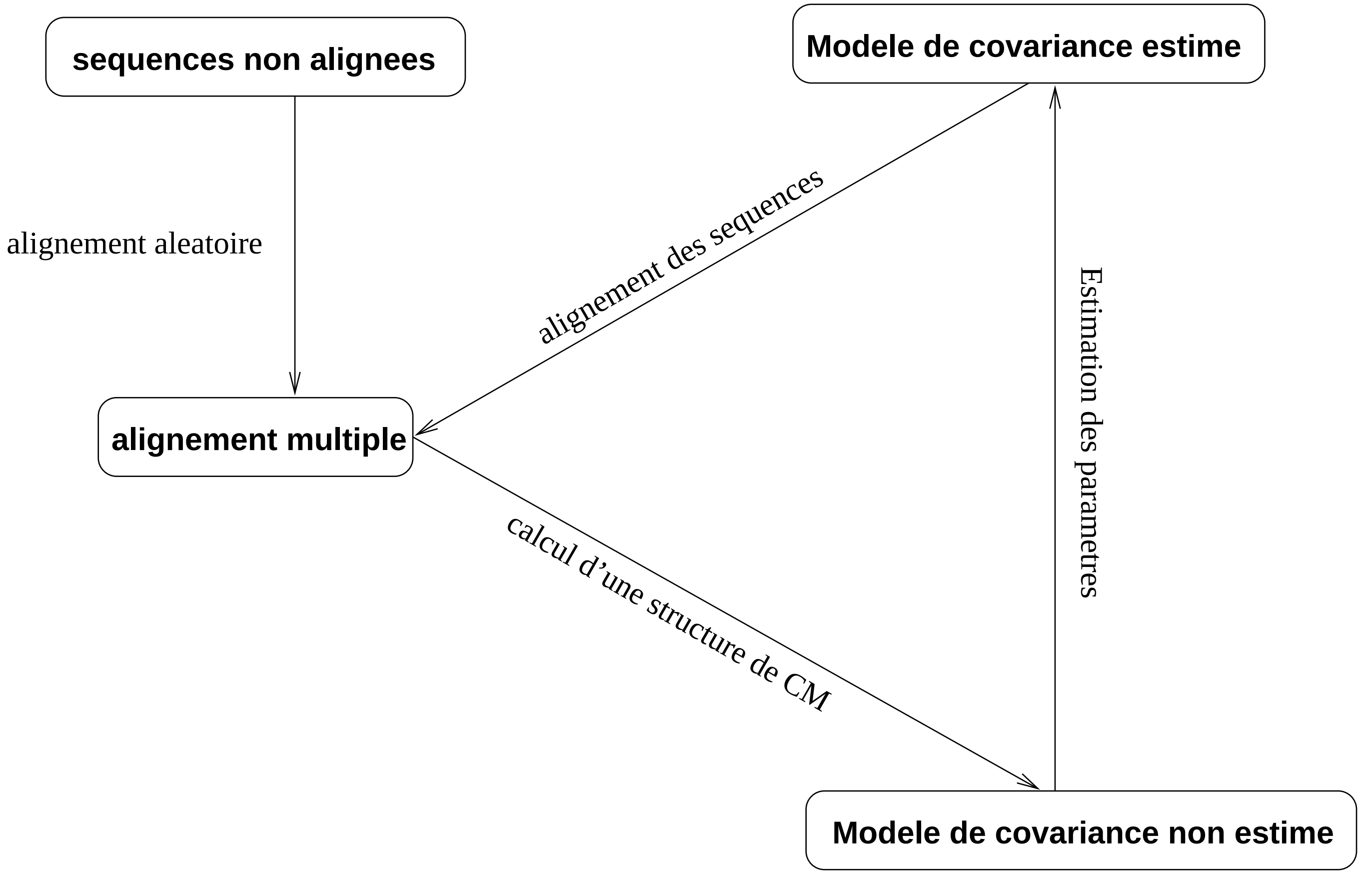
$$\mathcal{P}(x_i, x_j | y) = \frac{n(x|y) + \mathcal{R}(x|y)}{n(y) + \sum_{x' \in \{ACGU\}} \mathcal{R}(x'|y)}$$

$$\mathcal{T}(y_{next}|y) = \frac{n(y_{next}|y) + \mathcal{R}(y_{next}|y)}{n(y) + \sum_{y_{next}} \mathcal{R}(y_{next}|y)}$$

Mise à jour pour atteindre un MAP avec des priors donnés par les $\mathcal{R}$.

Prédiction de structure sans alignment

Par simulation de la procédure intuitive :



Problème : prédire la structure d'un CM à partir d'un alignement multiple.

Prédire la structure d'un CM

Exploitation des mutations compensées (les conservations complètes sont considérées comme non informatives).

$$M_{ij} = \sum_{x_i, x_j} f_{x_i,x_j}. \log_2 \frac{f_{x_i,x_j}}{f_{x_i}.f_{x_j}}$$

Exploitation de l'algorithme de Nussinov (adapté pour prendre en compte les M_{ij}). En $O(n^2)$ en espace et $O(n^3)$ en temps.

Matrice S_{ij} = somme de covariation maximale pour la sous-séquence $i \cdots j$.

Relation de récurrence :

$$S_{i,j} = \max[S_{i+1,j}, S_{i,j-1}, S_{i+1,j-1} + M_{i,j}, \max_{i<k<j} \{S_{i,k} + S_{k+1,j}\}]$$

Tests

Sur des tRNA

- modèle fin appris à partir des 1415 tRNA alignés (95% correct)

- modèle de bonne qualité appris à partir de 100 tRNA alignés (94% correct)

- modèle presque aussi bon à partir de 100 tRNA alignés (91% correct)

Tests complémentaires : temps de calcul assez long (typ. $> 10'$ sur un P133 pour la construction d'un modèle ab initio à partir de 100 tRNA). Sensibilité aux conditions de démarrage (max. locaux).

Tests complémentaires sur 26 seq. ARN selex : pas de résultat exploitable *a priori*. Convergence vers des modèles de vraisemblance variable suivant l'alignement aléatoire utilisé initialement.

Limitations

- temps CPU important. Traitement de longues séquences peu envisageables.

- nombre de séquences important nécessaire apparemment.

- optimisation de qualité moyenne (minimum locaux).

- pas de pseudo-nœuds.

- la prise en compte des M_{ij} seuls est insuffisante.

Autres approches

– SCFG : proche des CM. Initialement pas de prédiction de structure de la grammaire [Sakakibara et al. 94]. Algorithmes similaires (exploitant l'acyclicité).

  Maintenant, utilisation du Gibbs Sampling pour le placement et l'estimation d'hélices [Grate et al. 94] et extensions ad-hoc [Grate et al. 95] pour le traitement d'ARN plus grands.

  Prise en compte pseudo-nœuds par l'intersection de deux SCFG [Brown 96].

– Approches non probabiliste: [Stormo 95,97] permt la prise en compte de pseudo-nœuds (au prix d'une légère (?) sous-optimalité). Ces algorithmes pourraient probablement être adaptés à un modèle probabiliste rigoureux.

Avantages et inconvénients

- Modèle avec un sens précis : les critère optimisés sont bien définis à tout moment (alignement, apprentisage...) et ont un sens.

- souplesse des profils probabilistes (HMM) qui permettent des prendre en compte des variations sensibles autour d'un modèle.

- prise en compte d'a priori

- procédures d'apprentissages existantes (mais pas toujours idéales)

- lourd, difficile de s'éloigner de la programmation dynamique (Gibbs Sampling et autres MCMC (?), apparemment assez lourd).