

Réseaux de contraintes valuées et localisation de motifs structurés

Matthias Zytnicki

INRA BIA Toulouse

Contexte

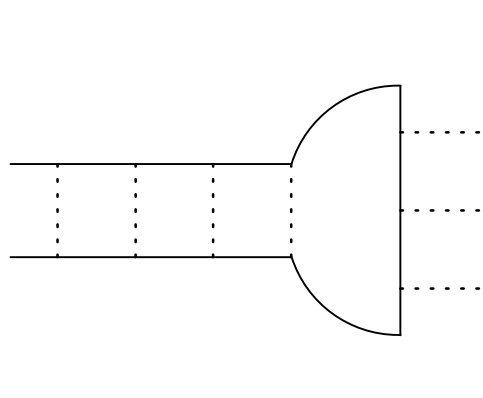
Objectif : rechercher les *meilleures* occurrences de motifs sous des contraintes données.

Contexte

Objectif : rechercher les *meilleures* occurrences de motifs sous des contraintes données.

Nous décrivons une famille d'ARN par sa structure secondaire, décomposable en quatre éléments de structures :

- le mot,
- l'espaceur,
- l'hélice,
- le duplex.



La contrainte duplex

Contexte

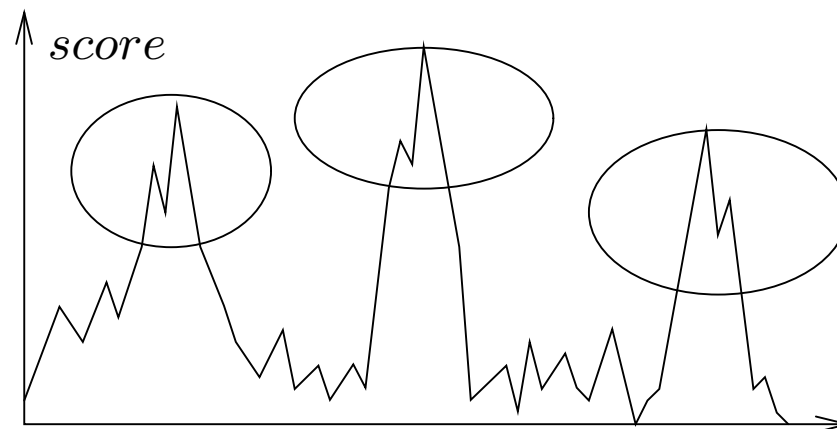
Objectif : rechercher les *meilleures* occurrences de motifs sous des contraintes données.

Nous décrivons une famille d'ARN par sa structure secondaire, décomposable en quatre éléments de structures :

le mot, l'espaceur, l'hélice, le duplex.

Le but est de trouver toutes les occurrences des séquences satisfaisant le motif structuré donné (problème NP-complet si l'on prend en compte les pseudo-nœud, triple hélice, *kissing hairpin* ...).

Problème : il faudrait classer ou trier les résultats donnés.



Contexte

Objectif : rechercher les *meilleures* occurrences de motifs sous des contraintes données.

Nous décrivons une famille d'ARN par sa structure secondaire, décomposable en quatre éléments de structures :

le mot, l'espaceur, l'hélice, le duplex.

Le but est de trouver toutes les occurrences des séquences satisfaisant le motif structuré donné (problème NP-complet si l'on prend en compte les pseudo-nœud, triple hélice, *kissing hairpin* ...).

On peut utiliser comme score une probabilité (comme dans tRNAscan-SE, cf. [DEKM98]) ou une énergie libre (comme dans Mfold, cf. [Zuk03]).

Les réseaux de contraintes valuées

Un réseau de contraintes est composé de :

- n variables (x_1, \dots, x_n) qui donnent l'emplacement des éléments de structure,
- leur domaine $(D(x_1), \dots, D(x_n))$ qui donnent l'ensemble des valeurs possibles pour chaque variable (au début, toute la séquence),
- e contraintes qui spécifient ces éléments de structure.

Les réseaux de contraintes valuées

Un réseau de contraintes est composé de :

- variables,
- domaines,
- contraintes.

Dans un réseau de contraintes valuées ([SFV95]), les contraintes sont des **fonctions de coût** dans E (qui peut être $\bar{\mathbb{N}}$ ou $\overline{\mathbb{R}^+}$), i.e. des éléments de : $D(x_{i_1}) \times \dots \times D(x_{i_r}) \rightarrow E$.

		x_1	
		a	b
		<hr/>	
x_2	a	1	0
	b	2	2

Un exemple de contrainte d'arité 2

Les réseaux de contraintes valuées

Un réseau de contraintes est composé de :

- variables,
- domaines,
- contraintes.

Dans un réseau de contraintes valuées ([SFV95]), les contraintes sont des **fonctions de coût** dans E (qui peut être $\bar{\mathbb{N}}$ ou $\overline{\mathbb{R}^+}$), i.e. des éléments de : $D(x_{i_1}) \times \dots \times D(x_{i_r}) \rightarrow E$.

Chaque contrainte peut être :

- satisfaite et donner un coût nul,
- violée et donner un coût-seuil (k),
- violée à un degré moindre et donner un coût compris entre 0 et k .

Les réseaux de contraintes valuées

Un réseau de contraintes est composé de :

- variables,
- domaines,
- contraintes.

Dans un réseau de contraintes valuées ([SFV95]), les contraintes sont des **fonctions de coût** dans E (qui peut être $\bar{\mathbb{N}}$ ou $\overline{\mathbb{R}^+}$), i.e. des éléments de : $D(x_{i_1}) \times \dots \times D(x_{i_r}) \rightarrow E$.

Les coûts des contraintes s'additionnent.

Le but est de trouver toutes les solutions dont la somme des coûts est inférieure à un seuil K :

$$\left\{ (v_1 \in D(x_1), \dots, v_n \in D(x_n)), \sum_i c_i(v_i) + \sum_{i \neq j} c_{ij}(v_i, v_j) + \dots < K \right\}$$

Algorithme de recherche

Exemple abstrait de recherche de solutions :

$$\begin{aligned} D(x_1) &= D(x_2) = D(x_3) = \{1, 2\} \\ c_{1,2} &= (x_1 - x_2)^+, c_{2,3} = (x_2 - x_3)^+, c_{1,3} = (x_1 - x_3)^+ \\ k_{1,2} &= k_{2,3} = k_{1,3} = 2, \quad K = 2 \end{aligned}$$

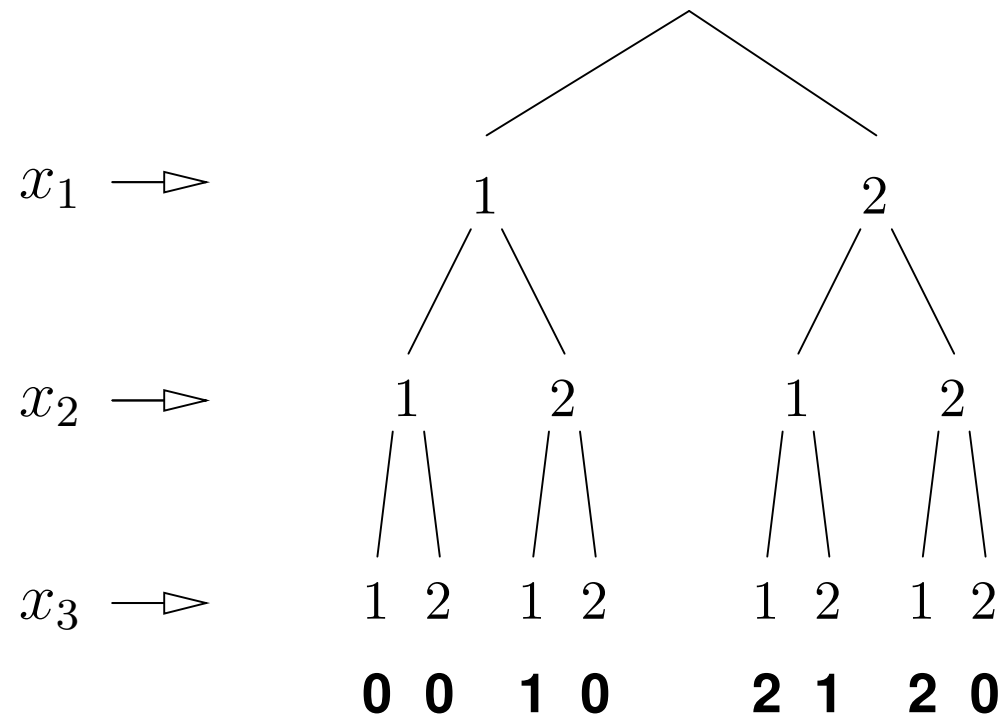
Le **sens** de la contrainte $c_{1,2}$ est :

x_1 doit être avant x_2 , ou le moins loin possible de x_2 .

Algorithme de recherche

Exemple abstrait de recherche de solutions :

$$D(x_1) = D(x_2) = D(x_3) = \{1, 2\}$$
$$c_{1,2} = (x_1 - x_2)^+, c_{2,3} = (x_2 - x_3)^+, c_{1,3} = (x_1 - x_3)^+$$
$$k_{1,2} = k_{2,3} = k_{1,3} = 2, \quad K = 2$$



Algorithme de recherche

Exemple abstrait de recherche de solutions :

$$\begin{aligned} D(x_1) &= D(x_2) = D(x_3) = \{1, 2\} \\ c_{1,2} &= (x_1 - x_2)^+, c_{2,3} = (x_2 - x_3)^+, c_{1,3} = (x_1 - x_3)^+ \\ k_{1,2} &= k_{2,3} = k_{1,3} = 2, \quad K = 2 \end{aligned}$$

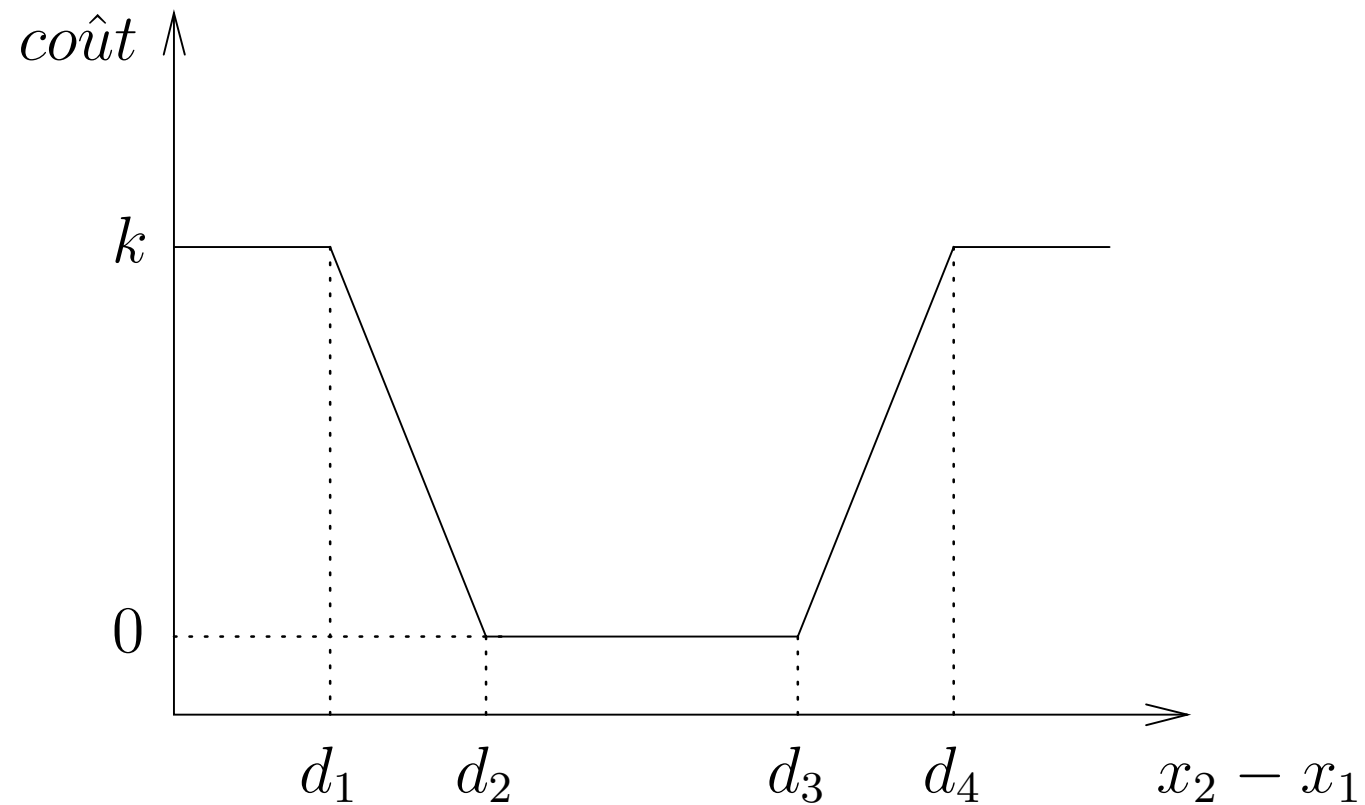
Par cette méthode naïve, on explore $\mathcal{O}(d^n)$ nœuds dans l'arbre.

(n représente le nombre de variables, d la taille du plus grand domaine)

Dans la pratique, on évalue un minorant *min* du coût de la solution courante à chaque nœud par **propagation** des coûts des contraintes. Si $min \geq k$, il n'y a pas de solution.

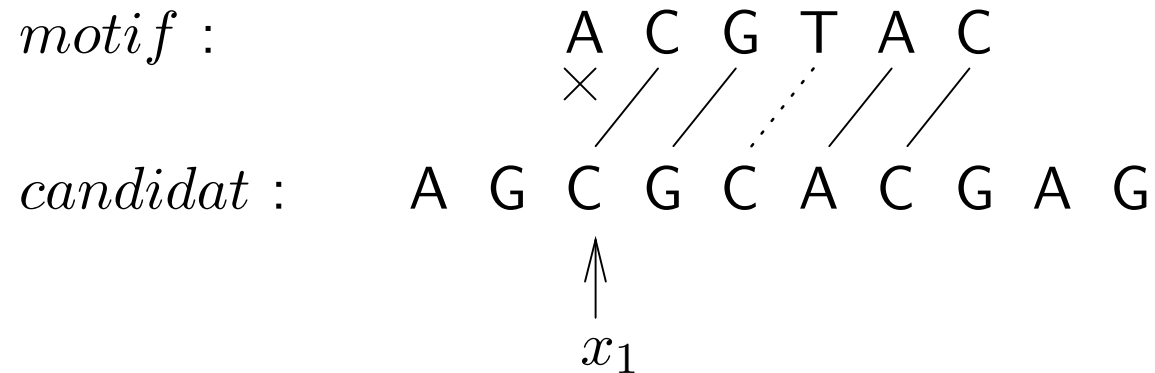
Exemples de contraintes

- l'espaceur,



Exemples de contraintes

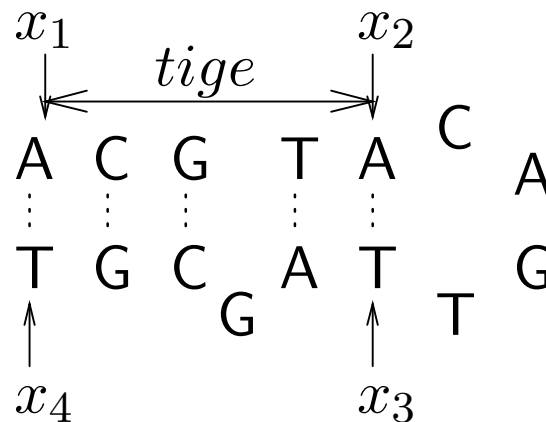
- l'espaceur,
- le mot,



Coût possible : $n_s \times c_s + n_{i/d} \times c_{i/d}$.
Soluble par automate (Wu-Manber).

Exemples de contraintes

- l'espaceur,
- le mot,
- l'hélice.

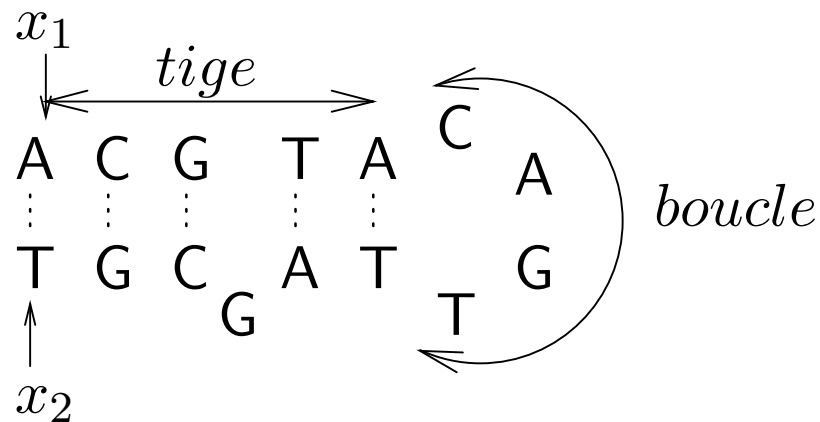


Coût possible : $n_s \times c_s + n_{i/d} \times c_{i/d} + |l_{mt} - l_t| \times c_{lt}$.

Soluble par programmation dynamique (Smith-Waterman).

Exemples de contraintes

- l'espaceur,
- le mot,
- l'hélice.



Coût possible : $n_s \times c_s + n_{i/d} \times c_{i/d} + |l_{mt} - l_t| \times c_{lt} + (l_b - 4) \times c_{lb}$.
Soluble par programmation dynamique (Nussinov).

Conclusion et perspectives

Conclusions :

- Nouvelle approche dans ce domaine
- Associe efficacement recherche et fonction de coût
- Particulièrement modulable
- Plus proche du modèle biologique
- Ouvre de nouvelles idées algorithmiques

Perspectives :

- Implémenter la contrainte duplex
- Faire de l'inférence de motif

Courte bibliographie

- [DEKM98] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.
- [SFV95] Thomas Schiex, Hélène Fargier, and Gérard Verfaillie. Valued constraint satisfaction problems: hard and easy problems. In *IJCAI-95*, 1995.
- [Zuk03] Michael Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13):3406–3415, 2003.