Amélioration des algorithmes d'alignement séquence/structure des méthodes de reconnaissance de repliements des protéines.

Jean-François Gibrat, Antoine Marin

Unité Mathématique, Informatique et Génome, INRA, Jouy-en-Josas

PaRI-STIC
21-23 novembre 2005

La biologie intégrative

- Révolution des techniques : séquençage, puces à ADN, gel 2D + spectrocopie de masse, etc.
- Changement de perspective en génétique on part du génome pour aller vers le phénotype.
- Démarche encyclopédique : on considère tous les gènes, toutes les protéines, l'ensemble des réseaux métaboliques, etc.
- Masse considérable de données très hétérogènes.
  - 1995 : séquençage du premier génome *H. influenzae*
  - 2000 : séquençage du génome humain
  - 2005 : 317 génomes publiés, 1351 en cours de séquençage

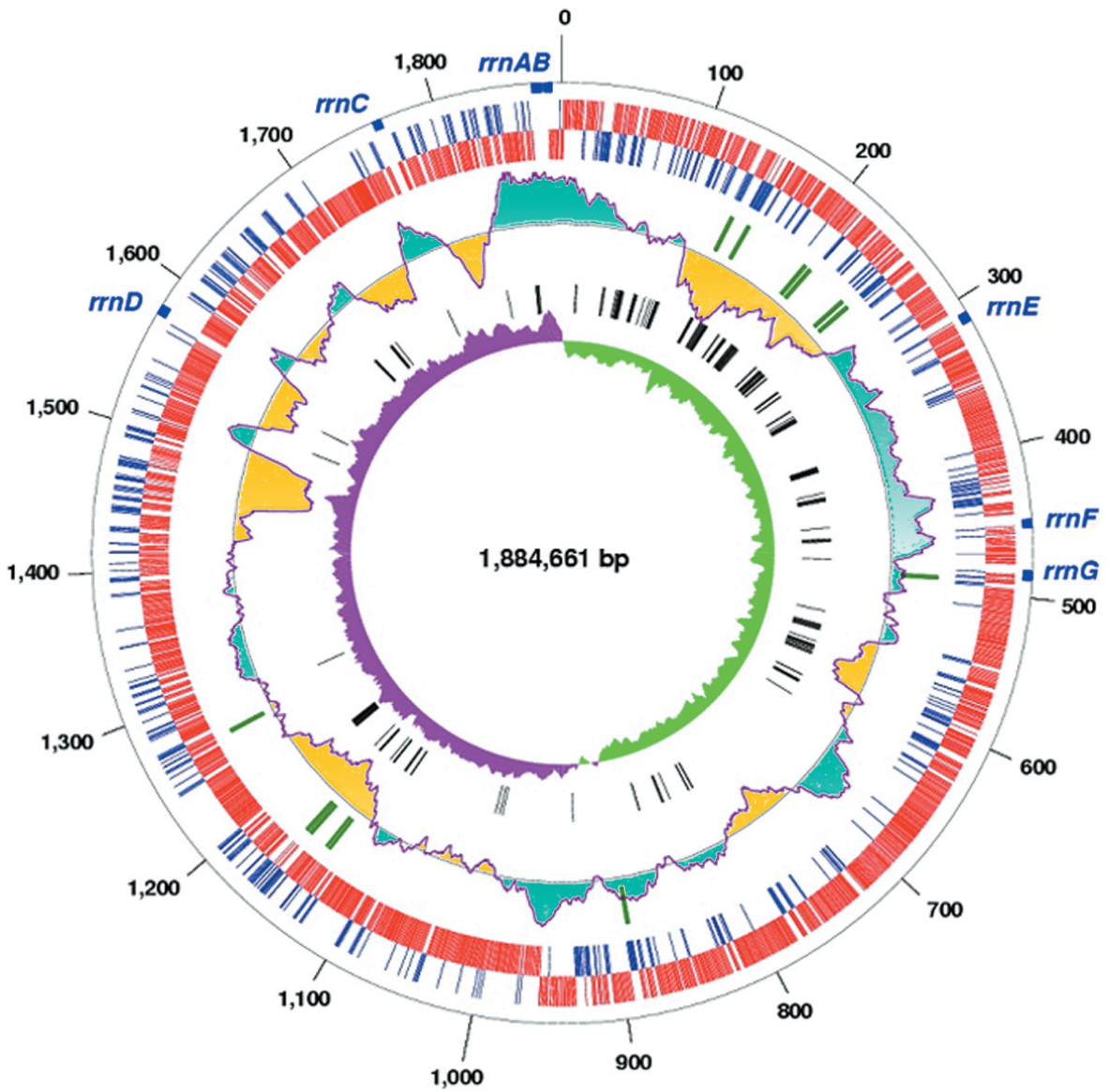
Du génome aux propriétés biologiques

- Comprendre comment le génome explique les propriétés biologiques d'un organisme.
- Analyse statique du génome
  - Au niveau des gènes : analyse structurelle (syntaxique)
  - Au niveau des protéines : analyse fonctionnelle
- Analyse dynamique du génome
  - Analyse des processus : interactions entre protéines

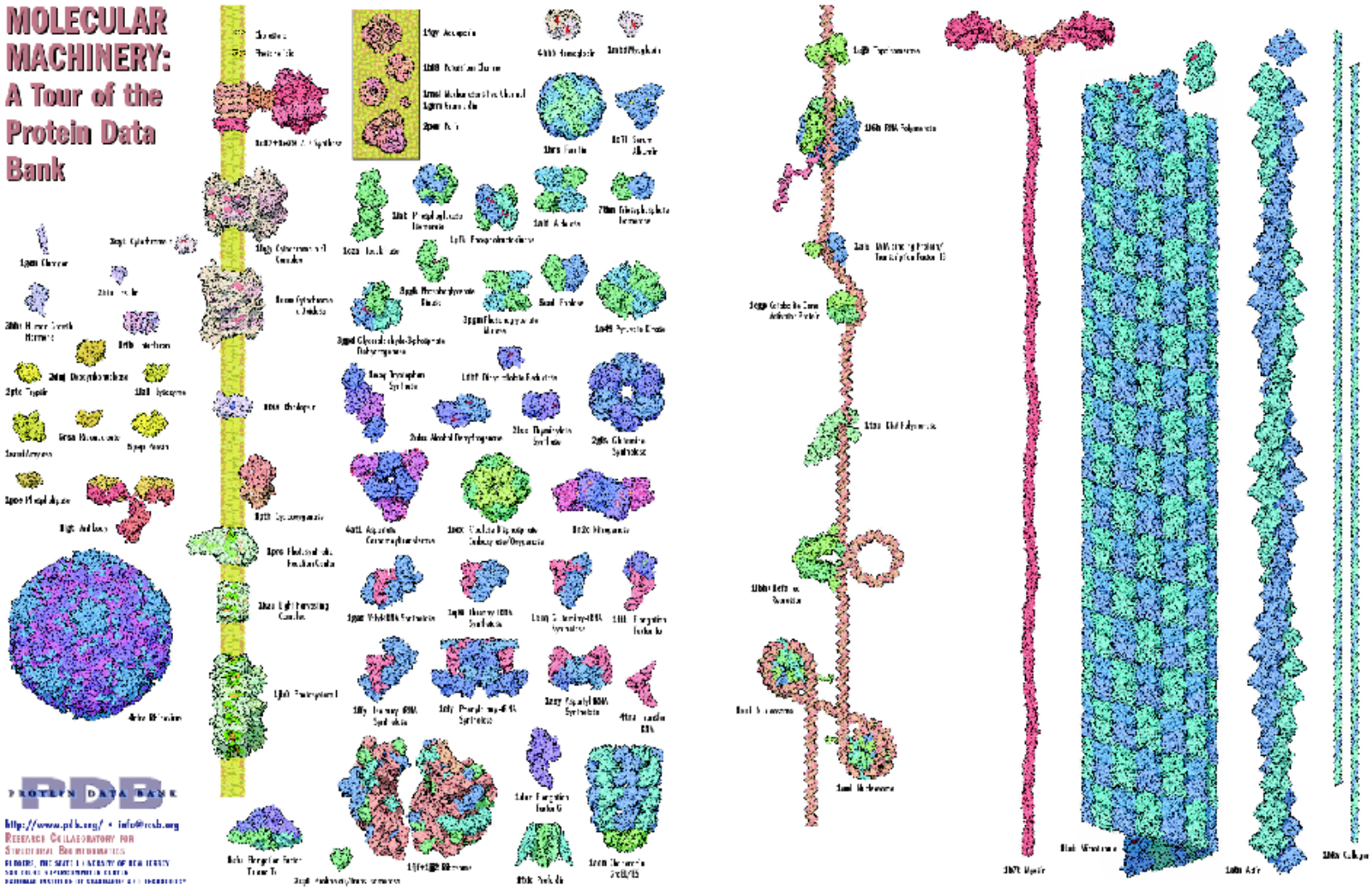
Du génome aux propriétés biologiques

- Comprendre comment le génome explique les propriétés biologiques d'un organisme.
- Analyse statique du génome
  - Au niveau des gènes : analyse structurelle (syntaxique)
  - Au niveau des protéines : analyse fonctionnelle
- Analyse dynamique du génome
  - Analyse des processus : interactions entre protéines

Génome de *L. sakei*



Rôle des protéines



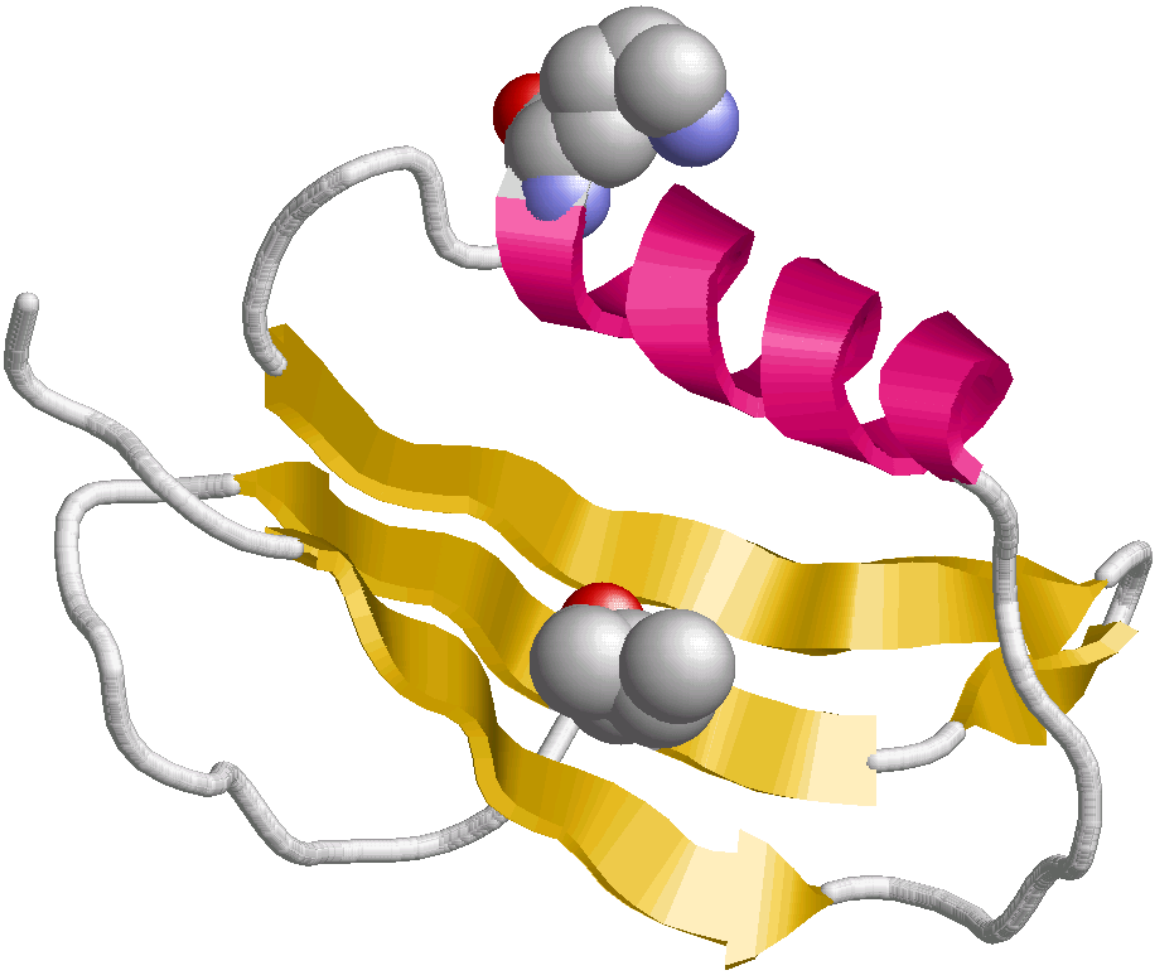
Propriétés des protéines

- Les protéines sont des hétéropolymères linéaires (20 « blocs », acides aminés ou résidus)
- La chaîne polypeptidique se replie en une structure unique : « native »
- La structure native ne dépend que de la séquence en acides aminés
- La fonction de la protéine dépend d'une *manière critique* de la structure 3D
- Il est très important de connaître la fonction d'une protéine et donc sa structure 3D

Propriétés des protéines

- Les protéines sont des hétéropolymères linéaires (20 « blocs », acides aminés ou résidus)
- La chaîne polypeptidique se replie en une structure unique : « native »
- La structure native ne dépend que de la séquence en acides aminés
- La fonction de la protéine dépend d'une *manière critique* de la structure 3D
- Il est très important de connaître la fonction d'une protéine et donc sa structure 3D

Description de la structure



Structures *vs* séquences

- Méthodes expérimentales pour résoudre la structure 3D des protéines
  - Diffraction des rayons X sur des cristaux
  - Spectroscopie RMN
- 40 000 structures 3D connues mais seulement 2900 familles différentes de protéines
- 2 200 000 séquences de protéines issues des projets de séquençage
- Il est important de pouvoir prédire la structure 3D à partir de la séquence.

Structures *vs* séquences

- Méthodes expérimentales pour résoudre la structure 3D des protéines
  - Diffraction des rayons X sur des cristaux
  - Spectroscopie RMN
- 40 000 structures 3D connues mais seulement 2900 familles différentes de protéines
- 2 200 000 séquences de protéines issues des projets de séquençage
- Il est important de pouvoir prédire la structure 3D à partir de la séquence.

De la biologie structurale à l'apprentissage automatique

- Projet GENOTO3D
- Laboratoire d'Informatique, de Robotique et de Microélectronique, **LIRMM**, Montpellier.
- Laboratoire d'Informatique Fondamentale, **LIF**, Marseille.
- Modèles informatique en biologie moléculaire, **MODBIO**, LORIA, Nancy.
- Systèmes et modèles biologiques, bioinformatique et séquences, **Symbiose**, IRISA, Rennes.
- Laboratoire de Bioinformatique et RMN structurale, **IBCP**, Lyon.
- Unité Mathématique, Informatique et Génome, **MIG**, Jouy-en-Josas.

Sous-problèmes étudiés... par différentes techniques

- Prédiction des ponts disulfures et des ponts salins (IBCP, IRISA, LIF, LORIA)
- Prédiction des structures secondaires (LIRMM, LORIA, MIG)
- Prédiction par homologie ou analogie (IBCP, IRISA, MIG)
- Prédiction *de novo* (MIG)

Sous-problèmes étudiés... par différentes techniques

- Prédiction des ponts disulfures et des ponts salins (IBCP, IRISA, LIF, LORIA)
- Prédiction des structures secondaires (LIRMM, LORIA, MIG)
- Prédiction par homologie ou analogie (IBCP, IRISA, MIG)
- Prédiction *de novo* (MIG)

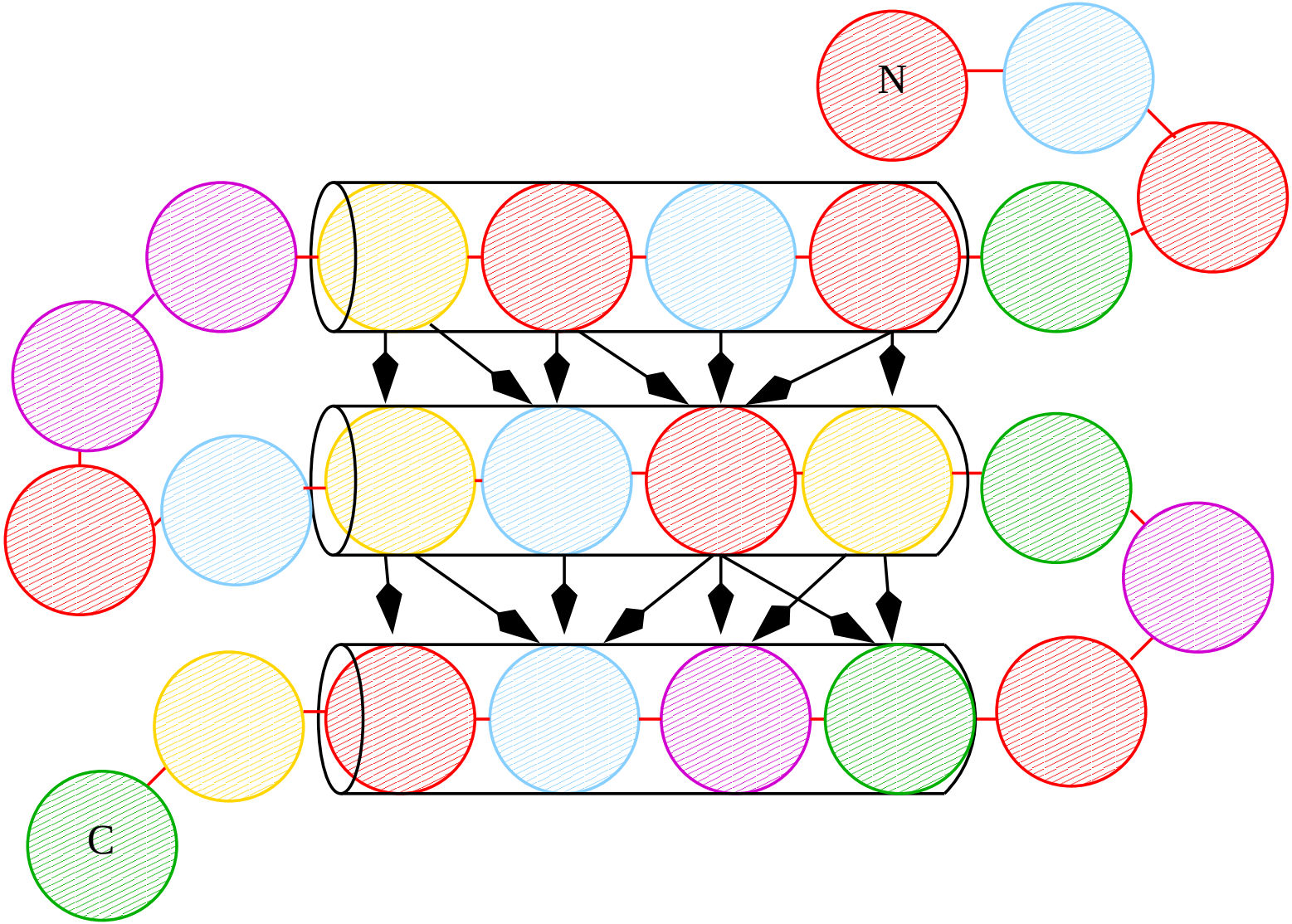
Méthode de reconnaissance de repliements

- Modélisation comparative, il faut disposer d'une structure 3D similaire dans les bases de données.
- Principe : aligner une séquence sur une structure 3D
- Méthode :
  - base de données de structures « cœurs »
  - fonction de score séquence/structure
  - algorithme d'alignement séquence/structure (« cœurs »)
  - analyse statistique de la significativité du score

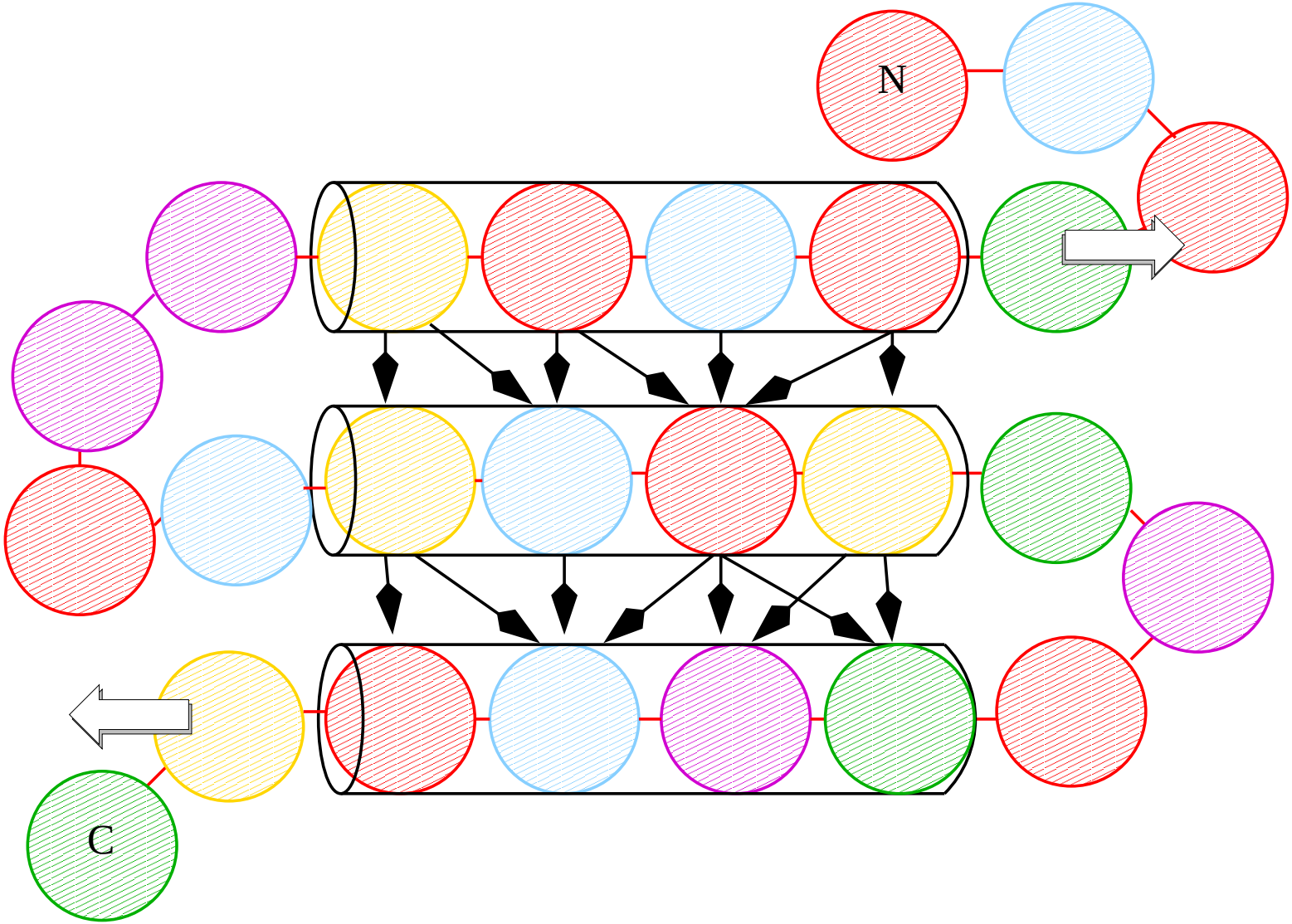
Méthode de reconnaissance de repliements

- Modélisation comparative, il faut disposer d'une structure 3D similaire dans les bases de données.
- Principe : aligner une séquence sur une structure 3D
- Méthode :
  - base de données de structures « cœurs »
  - fonction de score séquence/structure
  - algorithme d'alignement séquence/structure (« cœurs »)
  - analyse statistique de la significativité du score

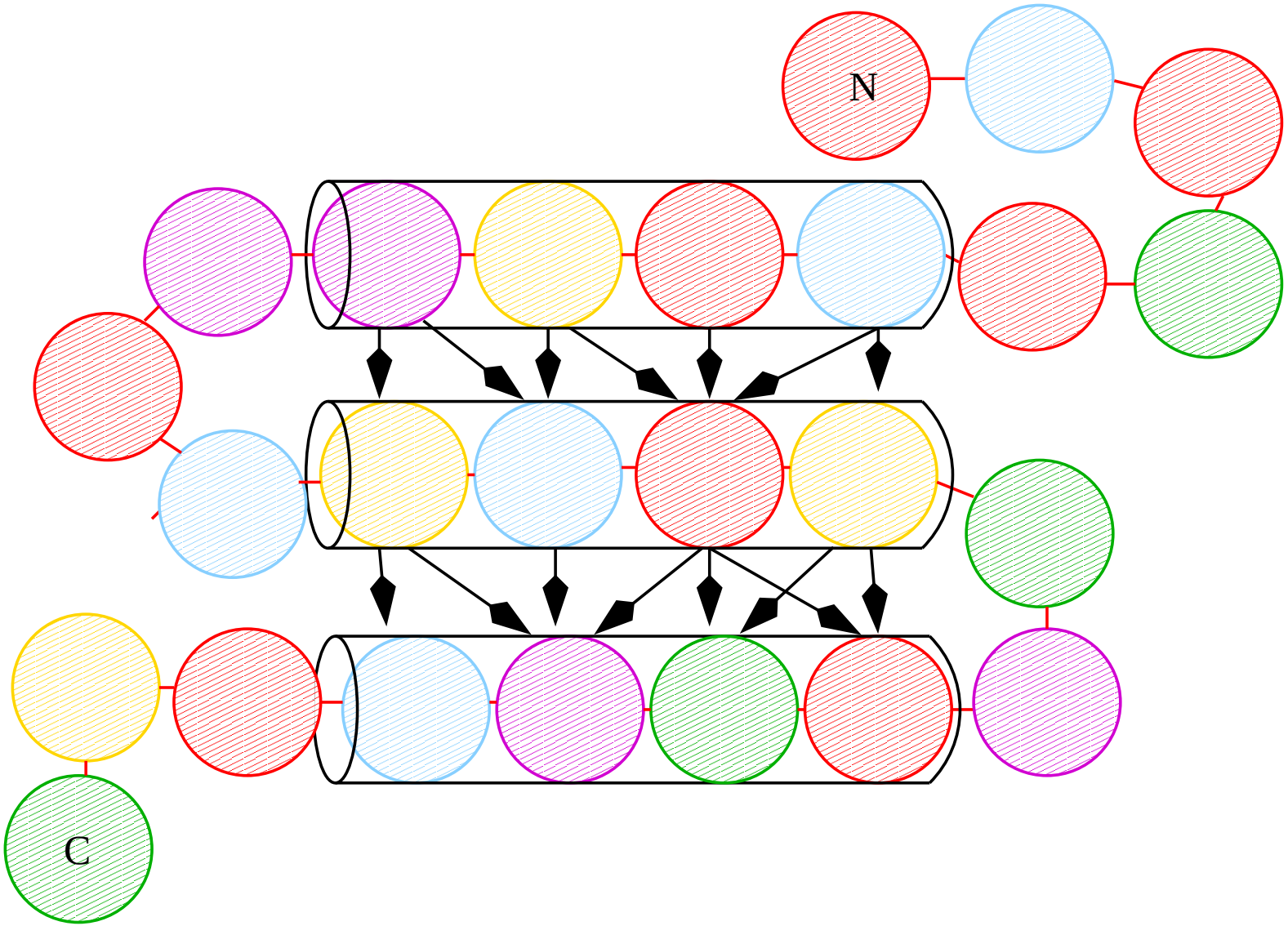
Threading : principe



Threading : principe



Threading : principe



Méthodes d'alignement de la séquence sur la structure

Problème qui a été démontré NP-difficile dans son cas le plus général.

- Méthodes heuristiques
  - méthodes *ad-hoc* : « double programmation dynamique », *frozen approximation*, etc.
  - méthodes stochastiques : Monte Carlo, échantillonnage de Gibbs.
- Méthodes exactes
  - méthodes de *branch & bound*
  - méthodes diviser pour régner
  - méthodes MIP (mixed integer programming)
    - R. Andonov IRISA, Rennes
    - S. Balev, U. du Havre
    - N. Yanev, U. Sofia (prof. invité à l'IRISA)

Méthodes d'alignement de la séquence sur la structure

Problème qui a été démontré NP-difficile dans son cas le plus général.

- Méthodes heuristiques
  - méthodes *ad-hoc* : « double programmation dynamique », *frozen approximation*, etc.
  - méthodes stochastiques : Monte Carlo, échantillonnage de Gibbs.
- Méthodes exactes
  - méthodes de *branch & bound*
  - méthodes diviser pour régner
  - méthodes MIP (mixed integer programming)
    - R. Andonov IRISA, Rennes
    - S. Balev, U. du Havre
    - N. Yanev, U. Sofia (prof. invité à l'IRISA)

Méthodes d'alignement de la séquence sur la structure

Problème qui a été démontré NP-difficile dans son cas le plus général.

- Méthodes heuristiques
  - méthodes *ad-hoc* : « double programmation dynamique », *frozen approximation*, etc.
  - méthodes stochastiques : Monte Carlo, échantillonnage de Gibbs.
- Méthodes exactes
  - méthodes de *branch & bound*
  - méthodes diviser pour régner
  - méthodes MIP (mixed integer programming)
    - R. Andonov IRISA, Rennes
    - S. Balev, U. du Havre
    - N. Yanev, U. Sofia (prof. invité à l'IRISA)

Méthodes d'alignement de la séquence sur la structure

Problème qui a été démontré NP-difficile dans son cas le plus général.

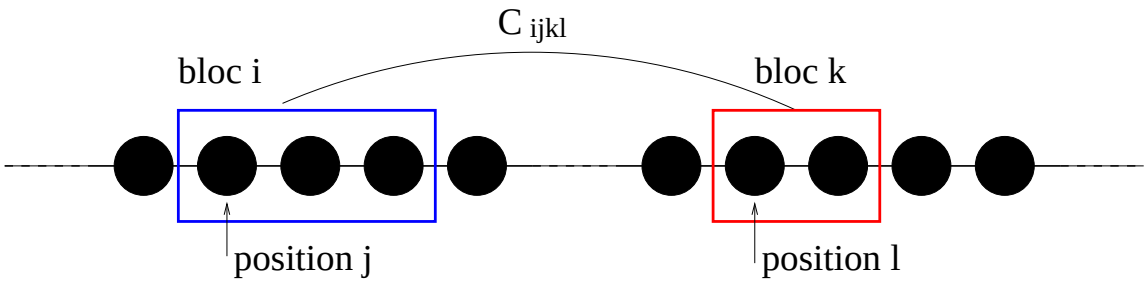
- Méthodes heuristiques
  - méthodes *ad-hoc* : « double programmation dynamique », *frozen approximation*, etc.
  - méthodes stochastiques : Monte Carlo, échantillonnage de Gibbs.
- Méthodes exactes
  - méthodes de *branch & bound*
  - méthodes diviser pour régner
  - méthodes MIP (mixed integer programming)
    - R. Andonov IRISA, Rennes
    - S. Balev, U. du Havre
    - N. Yanev, U. Sofia (prof. invité à l'IRISA)

Méthodes d'alignement de la séquence sur la structure

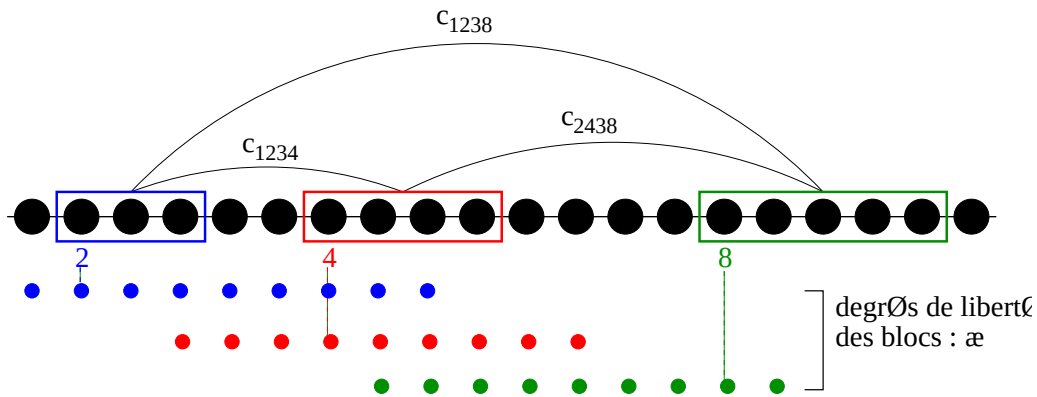
Problème qui a été démontré NP-difficile dans son cas le plus général.

- Méthodes heuristiques
  - méthodes *ad-hoc* : « double programmation dynamique », *frozen approximation*, etc.
  - méthodes stochastiques : Monte Carlo, échantillonnage de Gibbs.
- Méthodes exactes
  - méthodes de *branch & bound*
  - méthodes diviser pour régner
  - méthodes MIP (mixed integer programming)
    - R. Andonov IRISA, Rennes
    - S. Balev, U. du Havre
    - N. Yanev, U. Sofia (prof. invité à l'IRISA)

Score entre blocs

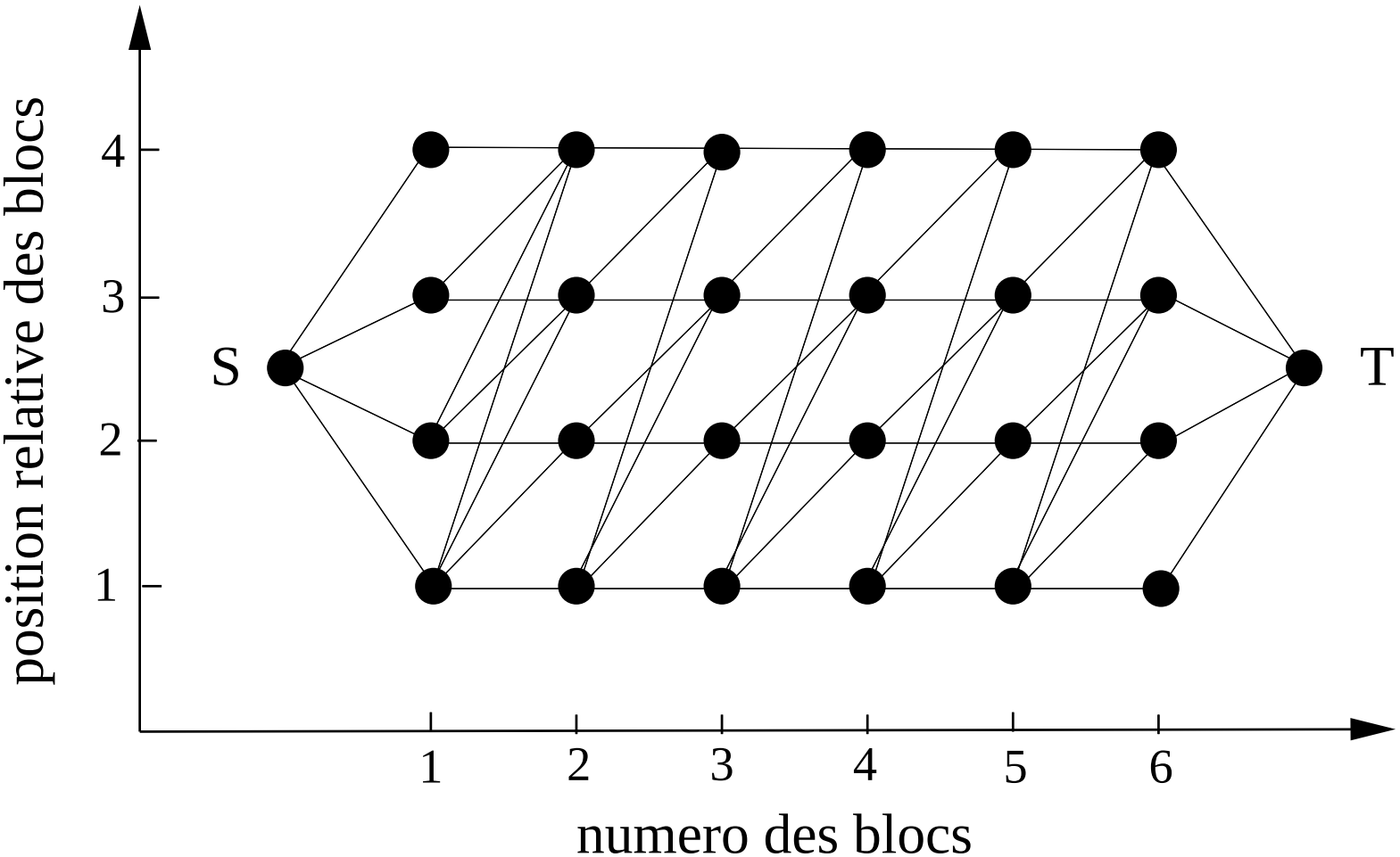


$$\phi(\pi) = \sum_{(i,k)\in E} c_{i\pi_i k\pi_k} \qquad 1 \leq \pi_i \leq \pi_k \leq \tilde{n}$$

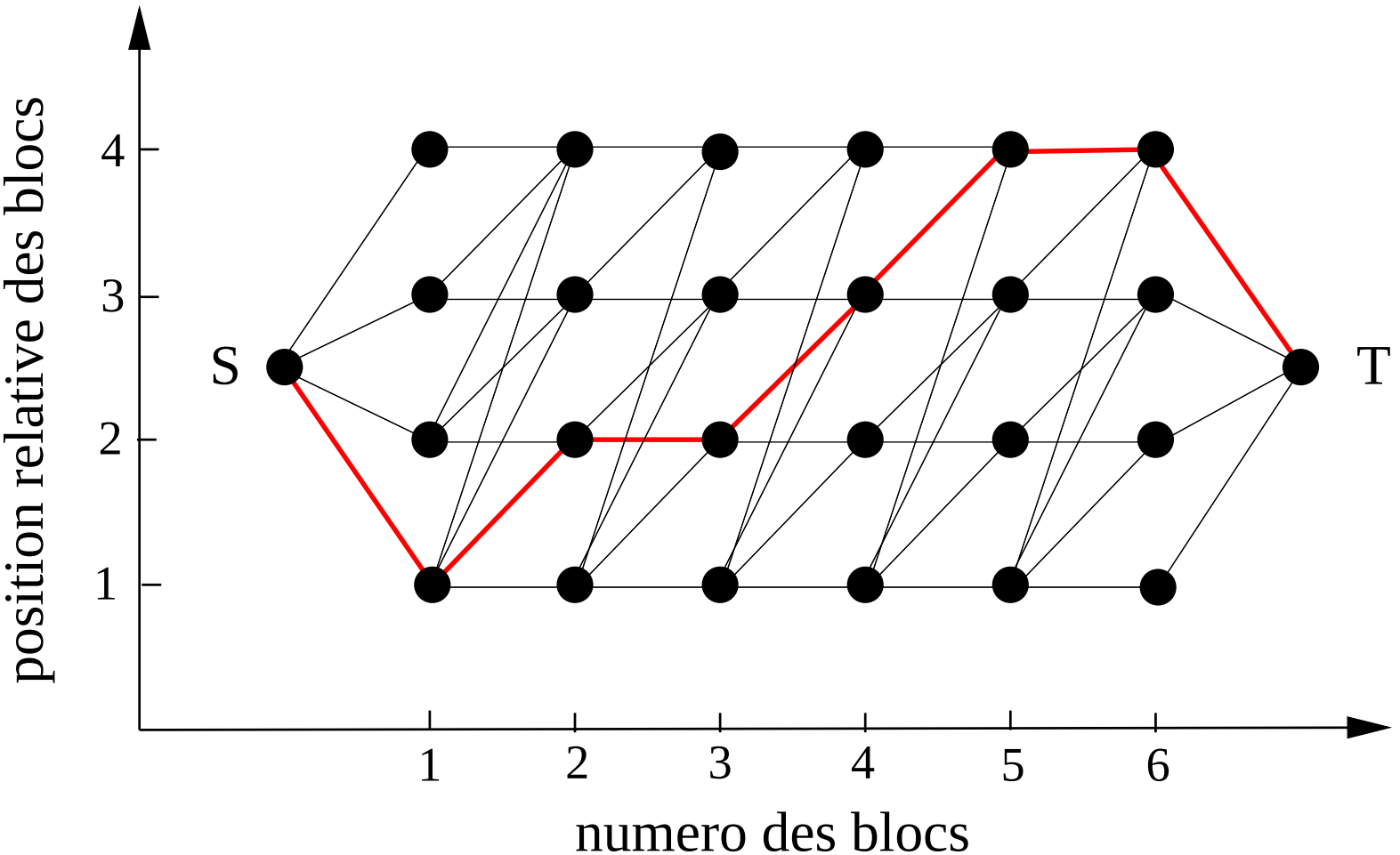


$$\phi(2,4,8) = c_{1224} + c_{1238} + c_{2438}$$

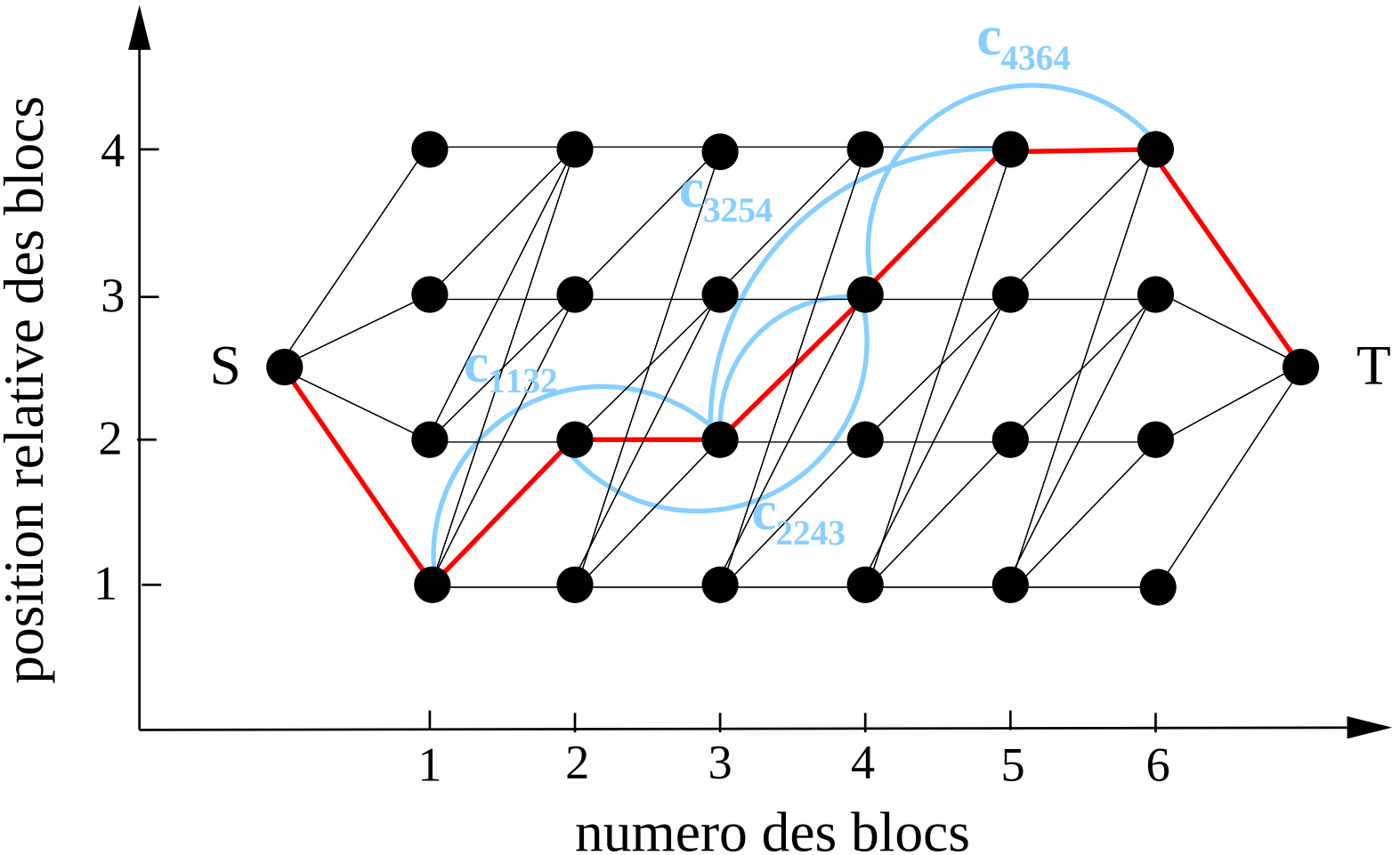
Représentation : diagramme de flot



Représentation : diagramme de flot



Représentation : diagramme de flot



Une formulation du problème

Variables :

$x_{ij} \in \{0,1\}, \quad i = 1,\ldots,m \; ; j = 1,\ldots,\tilde{n}$
$x_{ij} = 1 \quad \Longleftrightarrow$ bloc i en position j

Fonction objectif :

$$f(x) = \sum_{(i,k)\in E} \sum_{1\leq j\leq l\leq \tilde{n}} c_{ijkl}\, x_{ij}\, x_{kl}$$

Contraintes :

$$\sum_{j=1}^{\tilde{n}} x_{ij} = 1 \quad i = 1,\ldots,m$$

$$x_{i+1l} \leq \sum_{j=1}^{l} x_{ij} \quad i = 1,\ldots,m$$

Il existe d'autres façons (plus efficaces) de formuler le même problème...

Une formulation du problème

Variables :

$x_{ij} \in \{0,1\}, \quad i = 1,\ldots,m \; ; j = 1,\ldots,\tilde{n}$
$x_{ij} = 1 \quad \Longleftrightarrow$ bloc i en position j

Fonction objectif :

$$f(x) = \sum_{(i,k)\in E} \sum_{1\leq j\leq l\leq \tilde{n}} c_{ijkl}\, x_{ij}\, x_{kl}$$

Contraintes :

$$\sum_{j=1}^{\tilde{n}} x_{ij} = 1 \quad i = 1,\ldots,m$$

$$x_{i+1l} \leq \sum_{j=1}^{l} x_{ij} \quad i = 1,\ldots,m$$

Il existe d'autres façons (plus efficaces) de formuler le même problème...

Résoudre le problème en programmation entière

Problème initial
Programmation entière (PE)
$x \in \{0, 1\}$
$\mathcal{Z}_{PE} = min\ cx$
$x \in X$ – contraintes

Problème « relaxé » :
Programmation linéaire (PL)
$0 \leq x \leq 1$
$\mathcal{Z}_{PL} = min\ cx$
$x \in X$ – contraintes

- PL plus facile à résoudre que PE (méthode simplex)
- relaxation PL fournit une borne inférieure de la fonction objectif du problème PE (i.e., $\mathcal{Z}_{PL} \leq \mathcal{Z}_{PE}$)
- Utiliser cette borne dans un algo. « branch & bound » : solveur général PE/PL (MIP).
  - CPLEX, Ilog
  - GNU Linear Programming Kit

Résoudre le problème en programmation entière

Problème initial
Programmation entière (PE)
$x \in \{0, 1\}$
$\mathcal{Z}_{PE} = min\ cx$
$x \in X$ – contraintes

Problème « relaxé » :
Programmation linéaire (PL)
$0 \leq x \leq 1$
$\mathcal{Z}_{PL} = min\ cx$
$x \in X$ – contraintes

- PL plus facile à résoudre que PE (méthode simplex)
- relaxation PL fournit une borne inférieure de la fonction objectif du problème PE (i.e., $\mathcal{Z}_{PL} \leq \mathcal{Z}_{PE}$)
- Utiliser cette borne dans un algo. « branch & bound » : solveur général PE/PL (MIP).
  - CPLEX, Ilog
  - GNU Linear Programming Kit

Résoudre le problème en programmation entière

Problème initial
Programmation entière (PE)
$x \in \{0, 1\}$
$\mathcal{Z}_{PE} = min\ cx$
$x \in X$ – contraintes

Problème « relaxé » :
Programmation linéaire (PL)
$0 \leq x \leq 1$
$\mathcal{Z}_{PL} = min\ cx$
$x \in X$ – contraintes

- PL plus facile à résoudre que PE (méthode simplex)
- relaxation PL fournit une borne inférieure de la fonction objectif du problème PE (i.e., $\mathcal{Z}_{PL} \leq \mathcal{Z}_{PE}$)
- Utiliser cette borne dans un algo. « branch & bound » : solveur général PE/PL (MIP).
  - CPLEX, Ilog
  - GNU Linear Programming Kit

Résoudre le problème en programmation entière

Problème initial
Programmation entière (PE)
$x \in \{0, 1\}$
$\mathcal{Z}_{PE} = min\ cx$
$x \in X$ – contraintes

Problème « relaxé » :
Programmation linéaire (PL)
$0 \leq x \leq 1$
$\mathcal{Z}_{PL} = min\ cx$
$x \in X$ – contraintes

- PL plus facile à résoudre que PE (méthode simplex)
- relaxation PL fournit une borne inférieure de la fonction objectif du problème PE (i.e., $\mathcal{Z}_{PL} \leq \mathcal{Z}_{PE}$)
- Utiliser cette borne dans un algo. « branch & bound » : solveur général PE/PL (MIP).
  - CPLEX, Ilog
  - GNU Linear Programming Kit

Résoudre le problème en programmation entière

Problème initial
Programmation entière (PE)
$x \in \{0, 1\}$
$\mathcal{Z}_{PE} = min\ cx$
$x \in X$ – contraintes

Problème « relaxé » :
Programmation linéaire (PL)
$0 \leq x \leq 1$
$\mathcal{Z}_{PL} = min\ cx$
$x \in X$ – contraintes

- PL plus facile à résoudre que PE (méthode simplex)
- relaxation PL fournit une borne inférieure de la fonction objectif du problème PE (i.e., $\mathcal{Z}_{PL} \leq \mathcal{Z}_{PE}$)
- Utiliser cette borne dans un algo. « branch & bound » : solveur général PE/PL (MIP).
  - CPLEX, Ilog
  - GNU Linear Programming Kit

Résultats expérimentaux

- bien plus rapide que l'algorithme « branch & bound » original
- pour plus de 90% des instances réelles on trouve une solution entière directement ($\mathcal{Z}_{PL} = \mathcal{Z}_{PE}$).
- l'efficacité dépend de la façon dont le problème est formulé
- pour de grandes instances résoudre le problème linéaire est lent à cause du grand nombre de variables et de contraintes.
- temps de calcul : 10s pour des problèmes de taille 10^{20} et jusqu'à 2h pour des problèmes de taille 10^{40}

Relaxation Lagrangienne et dualité

Problème PE

$$\mathcal{Z}_{PE} = min\ cx$$
$$x \in X \ - \text{contraintes « faciles »}$$
$$Ax = b \ - \text{contraintes « compliquées »}$$

Idée (S. Balev) : supprimer une partie des contraintes et introduire dans la fonction objectif une pénalité en cas de violation de ces contraintes.

Relaxation Lagrangienne : $\mathcal{Z}_{RL}(\lambda) = min\ \{cx + \lambda(b - Ax) \mid x \in X\ \}$

- RL est un problème PE mais plus simple à résoudre
- RL est une relaxation de PE pour tous λ, i.e., $\mathcal{Z}_{RL}(\lambda) \leq \mathcal{Z}_{PE}$

Dual Lagrangien : $\mathcal{Z}_{DL}(\lambda) = max_{\lambda}\ \mathcal{Z}_{RL}(\lambda)$

- DL est une meilleure borne que PL : $\mathcal{Z}_{PL} \leq \mathcal{Z}_{DL} \leq \mathcal{Z}_{PE}$

Relaxation Lagrangienne et dualité

Problème PE

$$\mathcal{Z}_{PE} = min\ cx$$
$$x \in X \ - \text{contraintes « faciles »}$$
$$Ax = b \ - \text{contraintes « compliquées »}$$

Idée (S. Balev) : supprimer une partie des contraintes et introduire dans la fonction objectif une pénalité en cas de violation de ces contraintes.

Relaxation Lagrangienne : $\mathcal{Z}_{RL}(\lambda) = min\ \{cx + \lambda(b - Ax) \mid x \in X\ \}$

- RL est un problème PE mais plus simple à résoudre
- RL est une relaxation de PE pour tous λ, i.e., $\mathcal{Z}_{RL}(\lambda) \leq \mathcal{Z}_{PE}$

Dual Lagrangien : $\mathcal{Z}_{DL}(\lambda) = max_{\lambda}\ \mathcal{Z}_{RL}(\lambda)$

- DL est une meilleure borne que PL : $\mathcal{Z}_{PL} \leq \mathcal{Z}_{DL} \leq \mathcal{Z}_{PE}$

Relaxation Lagrangienne et dualité

Problème PE

$$\mathcal{Z}_{PE} = min\ cx$$
$$x \in X \ - \text{contraintes « faciles »}$$
$$Ax = b \ - \text{contraintes « compliquées »}$$

Idée (S. Balev) : supprimer une partie des contraintes et introduire dans la fonction objectif une pénalité en cas de violation de ces contraintes.

Relaxation Lagrangienne : $\mathcal{Z}_{RL}(\lambda) = min\ \{cx + \lambda(b - Ax) \mid x \in X\ \}$

- RL est un problème PE mais plus simple à résoudre
- RL est une relaxation de PE pour tous λ, i.e., $\mathcal{Z}_{RL}(\lambda) \leq \mathcal{Z}_{PE}$

Dual Lagrangien : $\mathcal{Z}_{DL}(\lambda) = max_{\lambda}\ \mathcal{Z}_{RL}(\lambda)$

- DL est une meilleure borne que PL : $\mathcal{Z}_{PL} \leq \mathcal{Z}_{DL} \leq \mathcal{Z}_{PE}$

Relaxation Lagrangienne et dualité

Problème PE

$$\mathcal{Z}_{PE} = min\ cx$$
$$x \in X \ - \text{contraintes « faciles »}$$
$$Ax = b \ - \text{contraintes « compliquées »}$$

Idée (S. Balev) : supprimer une partie des contraintes et introduire dans la fonction objectif une pénalité en cas de violation de ces contraintes.

Relaxation Lagrangienne : $\mathcal{Z}_{RL}(\lambda) = min\ \{cx + \lambda(b - Ax) \mid x \in X\ \}$

- RL est un problème PE mais plus simple à résoudre
- RL est une relaxation de PE pour tous λ, i.e., $\mathcal{Z}_{RL}(\lambda) \leq \mathcal{Z}_{PE}$

Dual Lagrangien : $\mathcal{Z}_{DL}(\lambda) = max_{\lambda}\ \mathcal{Z}_{RL}(\lambda)$

- DL est une meilleure borne que PL : $\mathcal{Z}_{PL} \leq \mathcal{Z}_{DL} \leq \mathcal{Z}_{PE}$

Résultats obtenus avec la relaxation Lagrangienne

	Nb ali	Nb seq	Min	Q_1	Med	Moy	Q_3	Max
1BGLA0	$1.2\ 10^{35}$	56	0.95	0.96	0.97	0.97	0.98	1.01
	$3.5\ 10^{58}$	192	35.6	39.9	42.2	45.2	50.0	73.2
	$1.3\ 10^{70}$	199	102.4	116.3	131.0	145.7	164.6	510.0
	$6.6\ 10^{77}$	150	203.8	229.7	252.6	291.7	327.5	797.4
1QBA_0	$8.3\ 10^{37}$	57	1.82	1.83	1.83	1.84	1.84	1.89
	$5.2\ 10^{57}$	197	27.1	30.2	32.5	36.3	39.8	76.6
	$2.8\ 10^{68}$	200	68.4	77.5	86.9	101.4	116.0	354.8
	$7.2\ 10^{75}$	200	130.1	154.7	178.3	207.0	239.8	789.8
1ALO_0	$6.0\ 10^{33}$	57	0.85	0.86	0.87	0.87	0.87	0.89
	$2.5\ 10^{57}$	190	25.8	29.3	36.1	40.8	46.7	135.2
	$1.6\ 10^{69}$	200	67.4	86.3	113.2	123.2	134.8	397.6
	$1.3\ 10^{77}$	200	139.9	175.7	231.0	262.2	303.4	735.0

Conclusion

- base de données de structures « cœurs »
- fonction de score séquence/structure
- algorithme d'alignement séquence/structure (« cœurs »)
- analyse statistique de la significativité du score

Conclusion

- base de données de structures « cœurs »
- fonction de score séquence/structure
- algorithme d'alignement séquence/structure (« cœurs »)
- analyse statistique de la significativité du score

Conclusion

- base de données de structures « cœurs »
- fonction de score séquence/structure
- algorithme d'alignement séquence/structure (« cœurs »)
- analyse statistique de la significativité du score

CASP7 (Critical Assessment of techniques for protein Structure Prediction)