

Analyse du contexte génétique d'un gène

Tristan COLOMBO¹, Yves QUENTIN²

¹ LIF – 39, rue Joliot Curie – 13453 Marseille Cedex 13 – tristan.colombo@cmi.univ-mrs.fr

² LMGM – 118, route de Narbonne – 31062 Toulouse Cedex 4 – quentin@ibcg.biotoul.fr

Les transporteurs ABC, de ATP Binding Cassette (Cassette qui fixe l'ATP), sont des systèmes d'export et d'import de molécules (le substrat) dans la cellule, présents à la fois chez les procaryotes et chez les eucaryotes. Ils peuvent transporter une grande variété de composés (des ions jusqu'aux protéines) et sont parfois également impliqués dans le rôle de pathogénicité de certaines bactéries ou dans leur résistance aux antibiotiques. Un système de transport est typiquement composé de quatre régions fonctionnelles (ou domaines), portés par quatre gènes. Dans ce cadre, l'analyse du contexte génétique d'un gène appartenant à un système dans un génome bactérien doit permettre :

- d'identifier de nouveaux partenaires de ce système (protéines impliquées dans le passage d'un composé du périplasma à l'extérieur de la bactérie dans le cas des bactéries Gram-),
- de préciser la nature des composés transportés en identifiant les enzymes impliquées dans le métabolisme de cette molécule,
- d'identifier les gènes impliqués dans la régulation de l'expression des gènes codant pour les partenaires du système.

D'après les observations faites sur les transporteurs ABC, les gènes codant pour les différents partenaires d'un système ne sont pas systématiquement organisés en opéron. Ils peuvent être dans la même orientation ou dans des orientations différentes et interrompus par des gènes ne codant pas pour des protéines liées fonctionnellement au transporteur. Plus rarement ces gènes peuvent être dispersés sur le chromosome.

Pour répondre à ce problème, nous avons développé une méthode permettant de détecter des groupes de gènes conservés au voisinage d'un gène – dit gène d'ancrage – dans de nombreux génomes et ceci quelle que soit leur orientation, leur ordre ou leur proximité locale (les gènes conservés ne seront pas forcément côte à côte). D'après la théorie de l'évolution, nous savons que les protéines codées par des gènes issus d'un gène ancestral commun possèdent des fonctions identiques ou similaires. C'est cette relation évolutive, appelée *orthologie* (Fitch, 1970), qui liera les gènes conservés entre génomes comme le montre la figure 1.

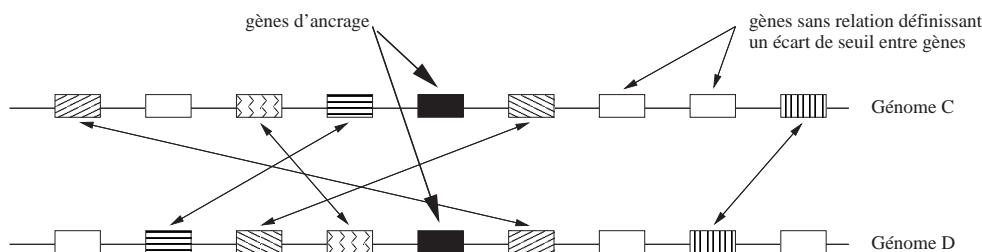


FIG. 1 – Représentation schématique du voisinage conservé autour de gènes d'ancrage (en noir) dans deux génomes C et D. Les gènes orthologues sont liés par une double flèche et possèdent le même motif. Les gènes en blanc sont des gènes sans relation.

Les hypothèses principales de cette étude sont que plus les gènes conservés seront loin du gène d’ancrage, plus leur lien fonctionnel avec le gène d’ancrage sera ténu ; plus le nombre de gènes sans relation entre deux gènes conservés sera grand, plus la cohérence de l’ensemble des gènes conservés sera faible ; plus les génomes étudiés seront taxonomiquement distants, plus les conservations détectées seront fortes. Pour estimer le lien fonctionnel entre ces gènes, nous posons deux contraintes :

- Un gène ne sera considéré comme conservé que jusqu’à une certaine distance du gène d’ancrage.
- Dans le voisinage du gène d’ancrage, on ne tolérera pas d’espacement entre gènes conservés supérieur à un seuil δ .

Dorénavant, lorsque nous parlerons d’*écart de seuil entre gènes*, il s’agira du nombre de gènes sans relation situés entre deux gènes possédant une relation génique dans la séquence comparée.

Historiquement, les méthodes développées pour répondre à des problèmes similaires sont celles de Snel *et col.* (2000), Morgat et Viari (2001) et Colombo *et col.* (2001) (Colombo *et col.*, 2002). Puis, Bergeron *et col.* (2003) ont développé la méthode GENETEAMS. Nous présenterons d’abord la méthode que nous avons développée, basée sur un formalisme dérivé des *CSPs* (Constraints Satisfaction Problems¹) (Montanari, 1974). Par la suite, nous présenterons l’implémentation de ce travail et les résultats obtenus, puis les différences avec d’autres méthodes de recherche de voisinage conservé.

1 Méthodologie

La méthode que nous présentons a pour but d’explorer le voisinage d’un gène donné – le gène d’ancrage – pour pouvoir inférer ses liens fonctionnels avec d’autres gènes. La recherche s’effectue en aval et en amont de la localisation chromosomique de ce gène. Nous prenons pour hypothèse que la conservation de gènes dans un même voisinage doit être le signe de liens fonctionnels entre leurs produits. Toutefois, nous relâchons la contrainte sur la distance intergénique, décomptée en nombre de gènes, où n’interviendra plus l’orientation des gènes. Dans un tel cadre, connaître la position d’un gène sur le chromosome constitue un élément fondamental puisque c’est à partir de cette position que nous pourrions déterminer l’écart de seuil entre gènes.

Définition 1.1 Soit Σ un ensemble de n gènes appartenant au chromosome C , et $P_C : \Sigma \rightarrow \mathbb{Z}$ la fonction qui à chaque gène $g \in \Sigma$ associe un entier $P_C(g)$ qui sera sa position.

Une fonction de ce type est très générale et permet de formaliser les différents types de distance intergénique. Elle induit une permutation sur un sous-ensemble S de Σ , ordonnant les gènes de S par position croissante.

Notation 1.2 La permutation correspondant à l’ensemble des gènes Σ du chromosome C sera notée π_C

Connaissant la position de deux gènes, nous définissons la distance qui les sépare.

Définition 1.3 Soient g et g' deux gènes de Σ , la fonction $\Delta_C : \Sigma \times \Sigma \rightarrow \mathbb{Z}$ définit la distance entre ces deux gènes sur le chromosome C : $\Delta_C(g, g') = |P_C(g') - P_C(g)|$.

Or dans notre cas, la distance entre un gène et le gène d’ancrage sera exprimée en nombre de gènes intercalés. Le gène d’ancrage occupe ici une position centrale par rapport à laquelle la position des autres gènes étudiés sera calculée. Nous devons donc étendre cette définition de la position.

Définition 1.4 Soient deux gènes g et A du chromosome C . A est le gène d’ancrage. Alors la position du gène $g \in \Sigma$, exprimée par rapport à A sur le chromosome C , sera donnée par la fonction

¹Problèmes de Satisfaction de Contraintes

$P_{C_A} : \Sigma \rightarrow \mathbb{Z}$ soit $P_{C_A}(g) = P_C(g) - P_C(\mathcal{A})$.

La position du gène d’ancrage est toujours 0, et les gènes situés en amont auront des positions négatives alors que les gènes situés en aval auront des positions positives. De plus, la distance séparant deux gènes est maintenant exprimée en nombre de gènes intercalés et elle est toujours calculée par rapport au gène d’ancrage. En nous basant sur la définition précédente :

Définition 1.5 Soient g et \mathcal{A} deux gènes de Σ où $g \neq \mathcal{A}$, la fonction $\Delta_{C_A} : \Sigma \rightarrow \mathbb{N}$ définit la distance de g au gène d’ancrage sur le chromosome C , exprimée en nombre de gènes intercalés : $\Delta_{C_A}(g) = |P_{C_A}(g)| - 1$.

Notre seconde hypothèse est que plus la distance entre gènes conservés² augmente, moins le lien fonctionnel est susceptible d’exister. Ainsi, un des paramètres essentiels de notre méthode est la taille maximale de l’écart de seuil entre gènes apparaissant entre un gène et le gène d’ancrage.

Notation 1.6 Nous noterons δ la taille maximale de l’écart de seuil entre gènes (défini par l’utilisateur).

Pour connaître la taille maximale de l’écart de seuil entre un gène et le gène d’ancrage, nous utilisons une nouvelle fonction de distance :

Notation 1.7 Soient g et \mathcal{A} deux gènes du chromosome C où \mathcal{A} est le gène d’ancrage ; soit \mathcal{A}' le gène du chromosome D lié par une relation génique³ à \mathcal{A} , $\Delta_{C_G} : \Sigma \rightarrow \mathbb{N}$ est la fonction indiquant le nombre maximum de gènes consécutifs entre g et \mathcal{A} qui sont sans relation génique avec un quelconque gène du voisinage de \mathcal{A}' .

Pour comparer la distribution des gènes entre deux fragments chromosomiques, nous devons être en mesure de dire si le gène g du chromosome C est le gène orthologue de g' sur le chromosome D (et réciproquement). Notons ici que la relation génique choisie peut être différente de l’orthologie mais doit être symétrique – ou bijective. Dans le cas d’une relation non bijective, un gène g du chromosome C pourrait être lié à g' du chromosome D , lui-même lié à g'' du chromosome C (figure 2). Nous ne saurions alors sur quel gène (g ou g'') porte la relation de g' . Avec ce genre de relation, une solution est de considérer que g et g'' représentent un seul et même gène qui a été scindé en deux parties au cours de l’évolution.

Notation 1.8 Soient g et g' deux gènes des chromosomes C et D respectivement, $p_{CD}(g, g')$ est une paire de gènes liés par une relation génique bijective. Utilisant l’orthologie, pour simplifier les notations, nous dirons que :

$$p_{CD} : \Sigma \times \Sigma \rightarrow \Sigma \cup \{\emptyset\} \text{ est telle que : } p_{CD}(g, g') = \begin{cases} g & \text{si } g \text{ et } g' \text{ sont orthologues} \\ \emptyset & \text{sinon} \end{cases}$$

Par cette fonction, deux gènes g et g' liés par une relation génique bijective auront donc la même dénomination ; ils seront identifiés par le même nom sur les différents chromosomes. Pour la description de la méthode, nous considérerons que les comparaisons sont effectuées entre fragments chromosomiques et que l’un d’entre eux est pris comme référence pour être comparé à tous les autres (la comparaison s’effectuant deux à deux). Le problème peut alors être exprimé comme suit : soit \mathcal{A}_C un gène d’ancrage issu du chromosome de référence C , et \mathcal{A}_D son orthologue sur le chromosome D ; notre objectif est de trouver la liste des gènes liés $p_{CD}(g, g')$ conservés dans le voisinage de la paire de gènes d’ancrage $p_{CD}(\mathcal{A}_C, \mathcal{A}_D)$

²Les gènes conservés sont ceux qui sont communs autour de \mathcal{A} dans plusieurs génomes.

³Relation pouvant être modifiée suivant les besoins.

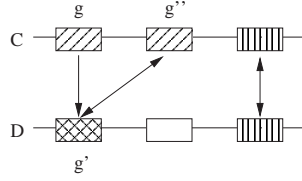


FIG. 2 – Illustration d’une relation non bijective sur deux chromosomes C et D . Les gènes g et g'' doivent être considérés comme un seul gène scindé au cours de l’évolution et en relation avec le gène g' .

avec un écart de seuil entre gènes tel que $\Delta_{C_G}(g) \leq \delta$ et $\Delta_{D_G}(g') \leq \delta$.

Propriété 1.9 *La liste des gènes conservés à un seuil $\delta - 1$ est incluse dans la liste des gènes conservés à un seuil δ .*

Pour une recherche exhaustive à tous les seuils inférieurs à δ , nous pourrions donc commencer par rechercher la liste des gènes conservés au seuil δ puis en déduire les listes des gènes conservés à des seuils inférieurs. Comme nous étudions le voisinage d’un gène particulier, nous devons définir ce voisinage pour nos calculs.

Notation 1.10 *On appellera fenêtre de taille w , le nombre de gènes étudiés en aval et en amont du gène d’ancrage (soit $2w + 1$ gènes au total).*

Nous avons commencé par développer un algorithme très simple, basé sur des intersections successives de listes de gènes.

1.1 Une première approche par intersections de listes

En phase initiale, deux listes contiennent l’ensemble des gènes g_i et g'_i appartenant respectivement à deux génomes différents C et D :

- qui sont à une distance du gène d’ancrage inférieure à la taille de la fenêtre w : $\Delta_{C_A}(g) \leq w$ et $\Delta_{D_A}(g') \leq w$,
- et qui appartiennent à des paires de gènes liés par une relation génique : $\exists g$ tel que $p_{CD}(g, g') \neq \emptyset$ et $\exists g'$ tel que $p_{CD}(g, g') \neq \emptyset$.

Ensuite, tant que l’écart de seuil maximal entre gènes est supérieur à δ , les gènes situés entre le dernier gène délimitant cet écart et la fin de la séquence sont supprimés de la liste (gène bordure de l’écart compris). Ce processus de suppression d’un (ou plusieurs) élément(s) d’une liste implique un processus d’intersection entre les deux listes pour propager les modifications effectuées sur une liste à la seconde liste. Or, le fait de supprimer un gène peut augmenter la taille d’un écart de seuil entre gènes et un processus de suppression des éléments trop éloignés devra être réappliqué. Ainsi, les procédures de suppression et d’intersection sont appliquées jusqu’à la convergence de la méthode. A la fin, on obtient donc deux listes, contenant un sous-ensemble des gènes initiaux, qui sont les gènes conservés pour une fenêtre fixée de taille w et un seuil δ .

Lors du calcul des voisinages conservés à tous les seuils possibles, on part du seuil maximal $\delta = w$ et on utilise la liste calculée pour $\delta = t$ pour inférer la liste des gènes conservés à $t - 1$ et ce jusqu’à $\delta = 0$. Le fait de ne pas fixer la valeur du δ permet de ne perdre aucune information : on a ainsi des gènes conservés avec une forte confiance – correspondant à un δ faible – et des gènes conservés avec une confiance moindre – correspondant à un δ élevé.

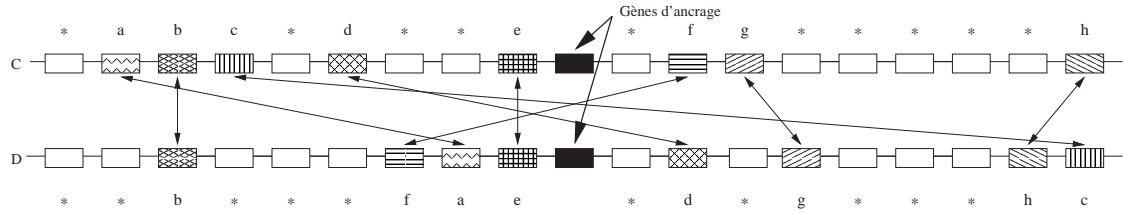


FIG. 3 – Conservation entre deux séquences de chromosomes C et D . Le "nom" de chaque gène est porté au-dessus de ce dernier dans le chromosome C (respectivement au-dessous dans le chromosome D). Le nom des gènes sur le chromosome D a été recodé à l'aide de la fonction p_{CD} . Ainsi, pour le gène a sur le chromosome C qui était en relation avec le gène x sur D , $p_{CD}(a, x) = a$ et le gène x a été renommé en a . Pour une meilleure lisibilité, les 0 renvoyés par la fonction en cas d'absence de relation entre gènes ont été remplacés par des "*".

Sur l'exemple présenté en figure 3, on étudie deux séquences chromosomiques en prenant une fenêtre de $w = 9$ gènes de part et d'autre des gènes d'ancrage. Des deux séquences, nous obtenons les listes :

$$\begin{aligned} L_C &= * a b c * d * * e \mathcal{A} * f g * * * * * h \\ L_D &= * * b * * * f a e \mathcal{A} * d * g * * * h c \end{aligned}$$

\mathcal{A} désignant le gène d'ancrage, l'écart de seuil entre gènes le plus important est de 5 gènes ; donc pour un δ variant de $w = 9$ à 5, tous les gènes seront conservés.

En considérant un $\delta = 4$: suppression du gène h dans L_C car trop éloigné en terme d'écart entre gènes, puis intersection avec L_D .

$$\begin{aligned} L_C &= a b c * d * * e \mathcal{A} * f g \\ L_D &= b * * * f a e \mathcal{A} * d * g * * * * c \end{aligned}$$

Pour un $\delta = 3$: suppression du gène c dans L_D et intersection avec L_C .

$$\begin{aligned} L_C &= a b * * d * * e \mathcal{A} * f g \\ L_D &= b * * * f a e \mathcal{A} * d * g \end{aligned}$$

Pour un $\delta = 2$: suppression du gène b dans L_D puis intersection avec L_C . Or le fait de supprimer b dans L_C va augmenter l'écart de seuil entre les gènes a et d . Cette distance est portée à 3 et dépasse le seuil δ . Il faut donc supprimer le gène a dans L_C et effectuer l'intersection avec L_D .

$$\begin{aligned} L_C &= d * * e \mathcal{A} * f g \\ L_D &= f * e \mathcal{A} * d * g \end{aligned}$$

Et ainsi de suite jusqu'à $\delta = 0$.

Cet algorithme est polynomial en temps. En posant n le nombre de gènes étudiés, pour un seuil δ donné, la phase d'initialisation (création des listes) est réalisée en n opérations et l'intersection entre deux listes de gènes peut être effectuée en temps linéaire. Mais dans le pire des cas, à chaque étape de l'algorithme, un seul gène sera supprimé (alternativement sur chaque génome) et cette modification devra être propagée en effectuant une nouvelle intersection. Il faudra donc effectuer n intersections : la complexité globale de cet algorithme est donc en $O(n^2)$.

Pour explorer des génomes complets il fallait un algorithme plus performant. Nous sommes en présence de contraintes, nous nous sommes donc tourné vers les CSPs. En effet, ce formalisme permet de représenter des problèmes basés sur des contraintes. Cette approche, à la fois originale et élégante, est également beaucoup plus performante dans le cas particulier qui nous intéresse.

1.2 Une approche inspirée des Problèmes de Satisfaction de Contraintes

Le formalisme CSP a été introduit par (Montanari, 1974). Pour définir des problèmes temporels, de nouveaux formalismes, dérivés des CSPs, ont été proposés : les TCSPs.

L'algorithme que nous avons développé s'inspire du modèle quantitatif (Dechter *et col.*, 1991), proposé par le formalisme TCSP (et plus précisément STP – Simple Temporal Problem – qui est une famille de problèmes plus simples, dont la complexité maximale en temps est en $O(n^3)$), et qui peut être transposé aux données spatiales sur une dimension, tels que des gènes ordonnés le long d'un chromosome. Dans un tel modèle, les variables seront les différents gènes présents dans la fenêtre définie par la taille w et centrée sur le gène d'ancrage \mathcal{A} . Nous aurons ici deux types de contraintes qui seront exprimées par des équations linéaires :

- C_{d_X} : contrainte de distance exprimant la distance entre un gène et le gène d'ancrage sur le chromosome X . Il s'agit de la distance $\Delta_{X\mathcal{A}}$, définie en p. 3, à la différence près que nous voulons conserver l'information sur la position du gène sur le chromosome par rapport au gène d'ancrage. Cette position $P_{X\mathcal{A}}(g)$ nous permet de définir une contrainte où la distance sera négative si le gène g se situe en aval du gène d'ancrage et positive sinon :

$$C_{d_X} : g - \mathcal{A} = \begin{cases} \Delta_{X\mathcal{A}}(g) & \text{si } P_{X\mathcal{A}} \geq 0 \\ -\Delta_{X\mathcal{A}}(g) & \text{sinon} \end{cases}$$

Cette contrainte ne sera calculée qu'une seule fois, la position des gènes sur le chromosome étant fixe.

- C_{g_X} : contrainte d'écart de seuil entre gènes exprimant la taille du plus grand écart existant entre un gène et le gène d'ancrage sur le chromosome X . Il s'agit de la distance Δ_{XG} définie en p. 3 :

$$C_{g_X} : g - \mathcal{A} = \Delta_{XG}(g)$$

Cette contrainte est susceptible d'être recalculée au cours du processus de résolution, la suppression d'un gène pouvant augmenter la taille d'un écart de seuil entre gènes.

Il existe également une contrainte sous-entendue qui est que lors de la recherche de la conservation à un seuil δ , on doit avoir $C_{g_X} \leq \delta$.

Le déroulement de l'algorithme sera alors le suivant : dans une première phase, dite *phase de révision*, on supprime tous les gènes qui violent la contrainte $C_{g_X} \leq \delta$. Pour chacun de ces gènes, si leur suppression implique un accroissement de l'écart de seuil entre gènes et oblige d'autres gènes à violer la contrainte, alors on *propage* l'information à ces gènes. Ce processus de révision/propagation est appliqué jusqu'à la convergence. Pour un calcul de la conservation à tous les seuils possibles, comme précédemment, nous utiliserons la propriété de la p. 4 : "(...) la liste des gènes conservés à un seuil $\delta - 1$ est incluse dans la liste des gènes conservés à un seuil δ ". Appliquons maintenant cet algorithme à l'exemple de la figure 3.

Si l'on recherche l'ensemble des gènes conservés dans le voisinage de \mathcal{A} (où la fenêtre est de $w = 9$ gènes de part et d'autre de \mathcal{A}) pour un seuil $\delta = 2$, nous obtiendrons :

- Suppression des gènes (phase de révision) :
 - b car $C_{g_D}(b) = 3 > \delta$
 - c car $C_{g_D}(c) = 3 > \delta$
 - h car $C_{g_C}(h) = 5 > \delta$ et $C_{g_D}(h) = 3 > \delta$
- Phase de propagation : la suppression des gènes b et c sur le chromosome C entraînent un accroissement de l'écart de seuil entre gènes. De plus, b et c se situaient entre le gène a et le gène d'ancrage \mathcal{A} . Donc $C_{g_C}(a)$ est mis à jour.
- Phase de révision : la nouvelle contrainte d'écart de seuil entre gènes sur a viole $C_{g_C}(a) \leq \delta$, donc le gène a est supprimé.

Ainsi, pour un seuil $\delta = 2$, les gènes conservés dans le voisinage de \mathcal{A} sont $\{d, e, f, g\}$.

Pour des séquences chromosomiques de n gènes, durant la phase d'initialisation, les contraintes C_{d_X} et

C_{g_x} sont calculées avec une complexité de l'ordre de $O(n)$. En considérant que les n gènes sont conservés à chaque itération, les phases de révision et de propagation sont effectuées en n étapes. En recherchant les voisinages conservés à tous les seuils possibles, il y a au plus $\delta_{max} = w$ itérations et les opérations précédentes sont donc effectuées en $\delta_{max} \times n$. Toutefois, dans le pire des cas, la complexité de notre algorithme est en $O(n)$. En effet, en pratique n décroît rapidement avec δ .

Destiné à une utilisation sur des génomes bactériens, le traitement des génomes circulaires a été pris en compte. Avec cet algorithme il ne s'agit en fait que d'une légère modification de la fonction P_{X_A} donnant la position d'un gène. On considère simplement que le gène qui précède le premier gène sur le chromosome considéré est en fait le dernier gène (et réciproquement). Pour ce qui est de la recherche de voisinage sur des génomes multiples, nous pouvons utiliser deux méthodes : (i) notre algorithme effectuant des comparaisons deux à deux en prenant pour référence les gènes de la première séquence chromosomique, nous pouvons effectuer les recherches de voisinage en considérant tour à tour chaque génome comme génome de référence, ou (ii) on peut également coder les contraintes s'exprimant sur tous les génomes et utiliser le même système de résolution ; seuls les gènes communs à tous les génomes seront alors pris en compte. La seconde méthode, bien que plus rapide, entraîne une perte d'information : les conservations de gènes doivent être observées dans absolument toutes les séquences pour être détectées. Or, avec la première méthode, nous pouvons détecter des gènes qui, par exemple, seraient conservés entre espèces très proches mais seraient conservés très rarement dans les autres espèces.

2 Interface

Pour permettre une utilisation simple ainsi qu'une analyse approfondie des résultats produits par l'algorithme basé sur les STP, j'ai développé une interface web.

Dans cette application, la relation génique employée est l'orthologie au sens de COG (Tatusov *et col.*, 1997). En effet, dans un premier temps nous avons calculé et utilisé des relations d'isorthologie mais les résultats n'étaient pas satisfaisants. L'isorthologie étant une relation très forte, seul un petit nombre de gènes était conservé, et dans le cas des transporteurs ABC, il ne s'agissait bien souvent que des transporteurs ABC eux-mêmes. Nous avons donc préféré utiliser les données de COG.

L'interface (<http://www.cmi.univ-mrs.fr/~tristan/bioinfo/GCTA>) a été travaillée de manière à être aussi simple et efficace que possible. Il faut tout d'abord sélectionner les génomes sur lesquels on désire travailler. En effet, suivant les cas, l'étude du voisinage d'un gène chez trois souches différentes d'une même bactérie ne sera pas informatif et ne contribuera qu'à densifier le volume de résultats. La seconde étape consiste à donner le nom du gène – ou le groupe COG – dont on souhaite explorer le voisinage. Il faut ensuite fixer la taille de la fenêtre puis la valeur maximale de l'écart de seuil entre gènes. Les données relatives aux gènes telles que l'orientation, la fonction, ou encore le groupe COG sont recherchées dans la base ABCDB (Quentin et Fichant, 2000). Après un laps de temps de quelques secondes⁴, le résultat de l'exploration du voisinage du gène soumis (ici un gène du groupe COG 3839) apparaît à l'écran (Figure 4). Les gènes sont représentés par des flèches indiquant l'orientation relative du gène. Deux gènes d'une même couleur sont des gènes orthologues au sens de COG (et peuvent donc être paralogues) et conservés dans au moins deux voisinages. Les gènes grisés sont les gènes appartenant au même groupe COG que le gène d'ancrage et ceux marqués d'un point en leur centre ne possèdent pas de groupe COG. On retrouve ici tous les partenaires du transporteurs ABC et deux groupes COG, correspondant à des enzymes, sont conservés à proximité. A partir de ces résultats, on peut prédire que ces systèmes transportent du sucre. En cliquant sur un gène on peut accéder directement à toutes ses informations sur la base ABCDB.

Deux fenêtres complémentaires indiquent la fonction des groupes COG conservés et leur nombre d'occurrences, c'est-à-dire le nombre de fois où ils apparaissent dans le voisinage du groupe COG correspondant

⁴Les comparaisons s'effectuant deux à deux et prenant tour à tour chaque génome comme génome de référence, pour n génomes nous effectuons $n \times (n - 1)$ appels à l'algorithme.

	Famille	1172	1129	1879	0747	1123	0601	1173	1653	1175	3839	3834	3833	0687	1131
COG1172	M_1	31	30	30	–	–	–	–	–	–	–	–	–	4	–
COG1129	N_1	30	31	32	–	–	–	–	2	2	2	–	–	4	–
COG1879	S_1	30	32	31	–	–	–	–	2	2	2	–	–	4	–
COG0747	S_2	–	–	–	7	2	4	4	4	4	4	–	–	–	–
COG1123	N_2	–	–	–	2	1	2	2	2	2	2	–	–	–	–
COG0601	M_2	–	–	–	4	2	3	4	2	2	2	–	–	–	–
COG1173	M_2	–	–	–	4	2	4	3	2	2	2	–	–	–	–
COG1653	S_5	–	2	2	4	2	2	2	23	24	10	–	–	–	2
COG1175	M_5	–	2	2	4	2	2	2	24	23	10	–	–	–	2
COG3839	N_5	–	2	2	4	2	2	2	10	10	9	–	–	–	–
COG3834	M_5	–	–	–	–	–	–	–	–	–	–	7	8	–	2
COG3833	M_5	–	–	–	–	–	–	–	–	–	–	8	7	–	2
COG0687	S_5	4	4	4	–	–	–	–	–	–	–	–	–	5	–
COG1131	N_7	–	–	–	–	–	–	–	2	2	–	2	2	–	3

TAB. 1 – Etude des transporteurs ABC conservés au voisinage du régulateur transcriptionnel *lacI* (COG1609). Chaque ligne indique les co-occurrences de conservation de groupes COG ainsi que le nombre d’occurrences. Pour chaque famille de transporteurs A_1, A_2, A_5 et A_7 , sont indiquées en gras les COG compatibles (les partenaires permettant de constituer un transporteur fonctionnel).

certain ordre à l’intérieur des génomes, ont été développées ((Mazumder *et col.*, 2001), (Tamames *et col.*, 2001), et (Suyama et Bork, 2001) dont l’objectif est surtout de montrer que la conservation des gènes n’est pas aléatoire).

STRING La méthodologie de STRING a été décrite dans (Snel *et col.*, 2000) et une mise à jour a été faite dans (von Mering *et col.*, 2003). L’utilisateur doit fournir un gène de requête qui sera utilisé comme gène d’ancrage – le *seed gene*. L’algorithme se déroule par itérations successives. Dans la première itération, STRING récupère et affiche les gènes qui apparaissent de manière répétée en co-occurrence avec le gène d’origine dans des groupes de gènes de multiples génomes. Les groupes de gènes sont ici définis en utilisant le concept de gènes en *série* d’Overbeek *et col.* (1999).

Dans les itérations suivantes, ce processus sera répété en utilisant successivement tous les nouveaux gènes, découverts lors de l’itération précédente, comme gène d’origine. Le processus général s’achève lorsque ce nombre est atteint ou lorsqu’aucun nouveau gène n’est découvert (convergence). Cette méthode bénéficie de plus dans sa version révisée d’une interface graphique aux informations multiples (textmining, ...) qui viennent enrichir les prédictions.

GeneTeams Cette méthode (Bergeron *et col.*, 2003) permet la recherche de δ -chaînes, c’est-à-dire de groupes de gènes d’orientation quelconque dans lesquels la distance entre deux gènes consécutifs n’est pas plus grande que le seuil δ . La limite de cette méthode est que les gènes conservés doivent être présents dans tous les génomes considérés.

Les paramètres de notre algorithme étant fixés ($w = 10, \delta = 3$), nous comparons les résultats obtenus grâce aux trois méthodes. L’étude de GENETEAMS (Luc *et col.*, 2003) ne portant que sur l’opéron tryptophane chez trois archaebactéries (*Archeoglobus fulgidus*, *Methanococcus thermoautotrophicum* et *Pyrococcus abyssi*), nous prendrons comme gène d’ancrage pour notre méthode et pour String le gène *trpA* d’*Archeoglobus fulgidus*⁵. Pour illustrer les différences de comportement, nous avons ajouté un génome supplémentaire où les gènes de l’opéron tryptophane ont été remaniés⁶ : *Aeropyrum pernix*. Pour en simplifier la lecture, les résultats obtenus dans le tableau 2 ne comportent pas les gènes sans relation qui sont intercalés. Le seuil choisi pour notre méthode et GENETEAMS est $\delta = 3$; une différence d’orientation des gènes est indiquée par un signe ‘-’.

Les résultats ne diffèrent que sur le premier génome. Tout d’abord l’algorithme STRING (en désactivant

⁵Il s’agit du premier gène de l’opéron. Les résultats sont identiques en considérant comme gène d’ancrage un autre gène de l’opéron.

⁶Nous sommes en présence de gènes dans les deux sens transcriptionnel.

GeneTeams							
<i>A. per.</i>	-trpG	-trpE	-trpD	<u>trpA</u>			
<i>A. ful.</i>	trpD	trpE	trpG	<u>trpF</u>	trpB	<u>trpA</u>	
<i>M. ther.</i>	trpE	trpG	trpC	trpF	trpB	<u>trpA</u>	trpD
<i>P. aby.</i>	trpC	trpD	trpE	trpG	trpF	trpB	<u>trpA</u>
String							
<i>A. per.</i>	<u>trpA</u>	trpB					
<i>A. ful.</i>	<u>trpC-D</u>	trpE	trpG	trpF	trpB	<u>trpA</u>	
<i>M. ther.</i>	trpE	trpG	trpC	trpF	trpB	<u>trpA</u>	trpD
<i>P. aby.</i>	trpC	trpD	trpE	trpG	trpF	trpB	<u>trpA</u>
STP							
<i>A. per.</i>	-trpG	- trpE	-trpD	<u>trpA</u>	trpB	trpC	
<i>A. ful.</i>	trpD	trpE	trpG	<u>trpF</u>	trpB	<u>trpA</u>	
<i>M. ther.</i>	trpE	trpG	trpC	trpF	trpB	<u>trpA</u>	trpD
<i>P. aby.</i>	trpC	trpD	trpE	trpG	trpF	trpB	<u>trpA</u>

TAB. 2 – Comparaison des résultats obtenus par les trois méthodes en prenant le gène *trpA* comme gène d’ancrage.

le "textmining") ne détecte que deux gènes conservés. En effet les trois gènes *trpG*, *trpE*, et *trpD* sont en orientation inverse. Les gènes *trpC* et *trpD* ne sont pas reconnus, contrairement à la méthode STP : le gène *trpD* est orienté en sens inverse et n’est donc pas détecté ; le gène *trpC* est trop éloigné (il rompt la série au sens d’Overbeek *et col.* (1999)).

D’autre part, STRING tient compte de la fusion des gènes (notamment pour le gène *trpC-D*). C’est la raison pour laquelle sur *Archeoglobus fulgidus* STRING est le seul à pouvoir détecter le domaine *trpC*.

Pour les différences de conservation entre les méthodes GENETEAMS et STP, la conservation de *trpB* et *trpC* dans le premier génome s’explique très simplement : ces gènes sont absents au moins une fois dans l’un des génomes considérés. Comme GENETEAMS recherche des groupes de gènes conservés dans tous les génomes, il considèrera forcément que ces gènes sont conservés sous la forme de gènes isolés et ne détectera donc pas la conservation générale.

Dans ce cas précis, on a une perte d’information sur *Aeropyrum pernix* en utilisant STRING ou GENETEAMS ; la méthode STP semble donc être plus adaptée à l’étude des transporteurs ABC.

5 Conclusion & Perspectives

Pour rechercher un voisinage conservé, on doit tout d’abord être capable d’identifier les gènes conservés entre génomes. Ceci est effectué en détectant les relations d’orthologie. Nous accordons beaucoup de poids à cette notion puisqu’on admet généralement que des gènes orthologues codent pour la même fonction dans des génomes différents. Mais peut-être peut-on se passer de cette relation lorsque l’on recherche une conservation de gènes. En effet, la co-occurrence de gènes homologues peut être suffisamment informative pour suspecter des liens fonctionnels. Une illustration de cette idée est donnée par l’identification de l’association fréquente de gènes codant pour des systèmes à deux composants avec des gènes codant pour des transporteurs ABC dans le groupe *Bacillus/Clostridium* ; des liens fonctionnels ont été découverts au moins au niveau transcriptionnel (Joseph *et col.*, 2002). Inversement, une relation plus "dure" telle que l’isorthologie ne donne pas de bons résultats car beaucoup trop discriminante. Il

faudrait donc pouvoir sélectionner une relation d'homologie parmi une liste lors de la requête. L'utilisation des données de la base HOBACGEN (Perrière *et col.*, 2000) pourrait être une piste intéressante. En effet, dans le cas des familles multigéniques (tel que *lacI*), il existe des sous-groupes apparaissant sur les arbres phylogéniques. L'utilisation de ces informations pourrait permettre de valider les prédictions de conservation entre régulateur et transporteur.

Il y a également le cas des gènes ayant fusionné. Pour l'instant, ils ne sont pas pris en compte par l'algorithme STP (bien que présents dans la base COG) et sont considérés comme un unique gène possédant donc un orthologue unique. Une modification très simple de l'algorithme permettrait d'intégrer de tels gènes et d'obtenir ainsi des résultats plus précis⁷. Cette modification entraînant également une reprogrammation de l'interface, elle n'a pas encore été effectuée.

On peut aussi utiliser des méthodes complémentaires qui vont apporter des informations différentes. Par exemple, une analyse par régressions linéaires simples a été réalisée pour la famille A_5 et son voisinage⁸ (Nicolas, 2003). Cette analyse, basée sur les grandes fonctions biologiques des groupes COG, a montré que la fonction des gènes représentés dans le voisinage des importeurs de la famille A_5 serait liée au métabolisme dans lequel est impliqué le substrat importé.

Cet algorithme rapide peut traiter de nombreux génomes et son interface permet une visualisation précise des résultats. Toutefois, une amélioration possible serait la détection automatique de liens fonctionnels. Cette étape ferait appel à un module de fouille de textes pour déterminer si la fonction COG du gène conservé est compatible ou non. Il devrait également être intégré dans la base de connaissance ABCkb (Capponi *et col.*, 2001) et ainsi l'enrichir automatiquement avec les informations prédites.

Références

- Bergeron, A., Corteel, S. et Raffinot, M. (2003) The algorithmic of gene teams. In Guigo, R. et Gusfield, D. (eds.), *Workshop on Algorithms in Bioinformatics*, pp. 464–476. LNCS.
- Capponi, C., Chabalier, J., Quentin, Y. et Fichant, G. (2001) A knowledge base for biological integrated systems. *IEEE Intelligent Systems, Special Issue : Intelligent Systems in Biology*, **16**, 52–60.
- Colombo, T., Benhamou, B., Capponi, C., Fichant, G. et Quentin, Y. (2001) Inférence fonctionnelle par l'analyse du contexte génétique - une application aux transporteurs ABC. In *Entretiens J. Cartier on Comparative Genomes et Poster JOBIM (2002)*. Lyon. Communication orale.
- Colombo, T., Guénoche, A. et Quentin, Y. (2002) Inférence fonctionnelle par l'analyse du contexte génétique - une application aux transporteurs ABC. In *Journées ALBIO*. Montpellier. Communication orale.
- Dechter, R., Meiri, I. et Pearl, J. (1991) Temporal constraint satisfaction problems. *Artificial Intelligence*, **49**, 61–95.
- Fitch, W. M. (1970) Distinguishing homologous from analogous proteins. *Syst. Zool.*, **19**, 99–113.
- Joseph, P., Fichant, G., Quentin, Y. et Denizot, F. (2002) Genetic link between ABC permease and regulatory systems in the bacillus/clostridium group suggests an involvement in a common physiological process. *J. Mol. Microbiol. Biotechnol.*, **4**, 503–513.
- Luc, N., Risler, J.-L., Bergeron, A. et Raffinot, M. (2003) Gene teams : a new formalization of gene clusters for comparative genomics. *Comput. Biol. and Chem.*, **27**, 59–67.

⁷Cette modification a d'ailleurs été apportée très récemment à l'algorithme GENEteams (Pasek *et col.*, 2004)

⁸Fréquence des groupes COG dans le voisinage de la famille A_5 en fonction de leur fréquence dans les génomes.

- Mazumder, R., Kolaskar, A. et Seto, D. (2001) Geneorder : comparing the order of genes in small genomes. *Bioinformatics*, **17**, 162–166.
- Montanari, U. (1974) Networks of constraints : fundamental properties and application to picture processing. *Information Sciences*, **7**, 95–132.
- Morgat, A. et Viari, A. (2001) Synténies bactériennes. In *Entretiens J. Cartier on Comparative Genomes*. Lyon. Communication orale.
- Nicolas, F. (2003) *Analyse comparative et évolutive des répertoires de transporteurs ABC dans les génomes bactériens séquencés*. Mémoire de DEA, Université d’Aix-Marseille II.
- Overbeek, R., Fonstein, M., D’Souza, M., Pusch, G. D. et Maltsev, N. (1999) The use of gene clusters to infer fonctional coupling. In *Proc. Nat. Acad. Sci. USA*, volume 96, pp. 2896–2901.
- Pasek, S., Bergeron, A., Risler, J.-L., Louis, A., Ollivier, E. et Raffinot, M. (2004) Identification of genomic features using domain teams. *Proceedings of JOBIM*.
- Perrière, G., Duret, L. et Gouy, M. (2000) HOBACGEN : database system for comparative genomics in bacteria. *Genome Res.*, **10**, 379 – 385.
- Quentin, Y. et Fichant, G. (2000) ABCDB : an ABC transporter database. Assembly and analysis of ABC transport systems in complete genomes. *J. Mol. Microbiol. Biotechnol.*, **2**, 501–504.
- Snel, B., Lehmann, G., Bork, P. et Huynen, M. A. (2000) STRING : a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. *Nucleic Ac. Res.*, **28**, 3442–3444.
- Suyama, M. et Bork, P. (2001) Evolution of prokaryotic gene order : genome rearrangements in closely related species. *Trends in Genetics*, **17**, 10–13.
- Tamames, J., Gonzalez-Moreno, M., Mingorance, J., Valencia, A. et Vicente, M. (2001) Bringing gene order into bacterial shape. *Trends in Genetics*, **17**, 124–126.
- Tatusov, R. L., Koonin, E. V. et Lipman, D. J. (1997) A genomic perspective on protein families. *Science*, **278**, 631–637.
- von Mering, C., Huynen, M., Jaeggi, D., Schmidt, S., Bork, P. et Snel, B. (2003) STRING : a database of predicted functional associations between proteins. *Nucleic Acids Res.*, **31**, 258–261.