Action ATGC

Action Transversale Génome et Calcul

Localisation : *Rocquencourt*

Mots-clés : algorithme, analyse de données, base de données, génome, modèle markovien, modélisation en biologie, ordonnancement, parallélisme, traitement du signal.

1 Composition de l'équipe

Responsables scientifiques

Jean-Jacques Codani, Ingénieur de Recherche Inria
Jacques Henry, Directeur de Recherche Inria

Secrétaire

Josy Baron, en commun avec LOCO et ATOLL

Conseiller scientifique

Michel Scholl, Professeur CNAM

Chercheurs associés Inria

Stéphane Grumbach, Chargé de Recherche Inria
Mireille Régnier, Directeur de Recherche Inria

Chercheur extérieur

René Devilliers, Ingénieur de Recherche Inserm

Ingénieurs experts

Jean–Marie Geffroy
Andrzej Wozniak, en commun avec PRAXITELE

Chercheur post-doctorant

Marjorie Moulet, jusqu'en juin 1996

Chercheurs doctorants

Jean-Christophe Aude, Boursier Inria, Université Paris Dauphine
Jean-Paul Comet, Boursier Inria, Université de Technologie de Compiègne
Eric Glémet, Boursier Inria, Université Versailles Saint-Quentin
Pierre Nicodème, Convention avec l'Université Paris 7
Fariza Tahi, ATER Cnam

2 Présentation du projet

La biologie moléculaire est entrée dans une nouvelle ère, celle conduisant à moyen terme à la production de nouveaux types de thérapeutiques et de diagnostics. Elle occupe désormais une place dominante dans les sciences biologiques et leurs applications biotechnologiques. Le séquençage de l'ADN des organismes vivants est devenu une technique de routine, automatisée et réalisée par des milliers de laboratoires. La place de l'informatique est grandissante en biologie moléculaire, et cette évolution n'est pas terminée. Ainsi, tout travail de recherche dans le domaine de la biologie fait appel à des des bases de données, qu'elles soient bibliographiques, de séquences d'ADN ou de protéines. Le nombre et la taille de ces bases de données grandissent sans cesse. La recherche d'une information pertinente sur une séquence nécessite désormais des méthodes plus sensibles et des moyens de calcul performants.

Les actions de Recherche menées au sein de l'"Action Transversale Génome et Calcul" (ATGC), à l'Unité de Recherche de Rocquencourt, se structurent autour de la réalisation du logiciel de comparaison de séquences LASSAP (Large Scale Sequence compArisons Package). LASSAP est un système programmable qui lève de nombreuses limitations des programmes actuels. Il est conc cu pour répondre aux besoins des projets d'analyse à grande échelle. LASSAP offre au programmeur un "cadre" permettant d'intégrer de nouveaux algorithmes et de les combiner. Tout algorithme bénéficie automatiquement de valeurs ajoutées sur les requêtes, les résultats et les performances. LASSAP est conçu pour traiter les problèmes de recherche de similarités et d'alignements à grande échelle. Les travaux menés cette année portent en particulier sur les stratégies de parallélisation. En outre, Jean-Marie Geffroy a rejoint l'action ATGC afin de produire une interface graphique pour l'interaction et la visualisation des résultats. Ce travail est soutenu par l'ANVAR. Andrzej Wozniak a développé des optimisations poussées des algorithmes de programmation dynamique, qui ont été intégrées dans LASSAP.

Autour de la réalisation de ce logiciel, des recherches méthodologiques sont menées. Elles concernent:

- l'analyse de la significativité des scores d'alignements;
- la combinaison de méthodes de classification;
- le développement de nouveaux algorithmes de comparaison et de recherche de motifs;
- des recherches sur la méthodologie de la programmation dynamique.

Ces travaux se font en collaboration avec des équipes de biologistes moléculaires comme le CGM/UVSQ (étude du génome de la levure) et des centres de services tels que INFOBIOGEN, le CNUSC et l'EBI.

3 Actions de recherche

3.1 LASSAP

Participants : Eric Glémet, Jean-Jacques Codani

Nous avons poursuivi le développement de LASSAP, et notamment affiné les stratégies de répartition de charge. Dans un premier temps, les complexités des algorithmes présents dans LASSAP ont été étudiées de manière théorique et/ou statistique. Ces complexités varient fortement en fonction de l'algorithme (de Boyer-Moore à la programmation dynamique). De surcroît, pour un algorithme donné, les complexités d'initialisations (automates, ...) peuvent jouer un rôle non négligeable suivant le type de calcul (par exemple, comparer une séquence contre une banque n'est pas équivalent, en terme de temps de calcul, à comparer une banque contre une séquence). Ceci influe fortement sur la stratégie de découpage. Sur une machine parallèle donnée, le temps de calcul dépend de plusieurs paramètres:

- de la complexité de l'algorithme;

- du type de calcul: banque contre banque, banque contre elle-même, séquence contre banque;
- de la complexité induite par les données: nombre de séquences et distribution de leurs longueurs dans les banques;
- de l'architecture de la machine: nombre de processeurs, type d'architecture, ressources disponibles (mémoire, entrées-sorties, ...).

Une stratégie de découpage et de répartition qui minimise le temps de calcul, quels que soient les paramètres, est théoriquement réalisable moyennant certaines approximations, en particulier en faisant abstraction de la localité des données et de la charge des processeurs. Une étude des différents cas de figures entre découpage (statique ou dynamique) et répartition (statique ou dynamique) montre qu'un découpage statique associé à une répartition dynamique est la solution qui permet la variation des paramètres tout en garantissant une efficacité de calcul. A titre d'exemple, si un découpage figé (découpage statique, répartition statique, banques locales) des banques sur différents processeurs – dans le cas d'architectures à mémoire distribuée – semble intuitivement être une solution efficace, il est mis en défaut par le mécanisme de sélection (calcul d'une sous-banque) de LASSAP. On se reportera à la thèse d'Eric Glémet pour une discussion détaillée.

Ceci nous a donc conduit à implanter des stratégies de découpage et de répartition de la charge en fonction de l'algorithme, du nombre de processeurs, et de la taille des données. Cette implantation sera utilisée entre le CNUSC et INFOBIOGEN, dans un modèle client serveur afin de servir les besoins de la communauté biologiste en comparaison de séquences.

Pour ce faire, trois développements ont été menés:

- Mise au point d'un évaluateur temps/complexité de calcul en fonction de la requête soumise. Il permet de piloter un allocateur de processeurs.
- En collaboration avec INFOBIOGEN, mise au point d'un protocole client/serveur. LASSAP intègre désormais la notion de fichier local et de fichier distant.
- Mise au point de filtres pour l'analyse post-opératoire des résultats structurés produits par LASSAP. Les filtres permettent la relecture des résultats ainsi que de multiples sélections sans avoir à relancer le calcul.

3.2 Optimisations de l'algorithme de Smith/Waterman

Participant : Andrzej Wozniak

L'amélioration de la vitesse d'exécution des algorithmes de programmation dynamique répond à un besoin important en comparaison de séquences. En témoignent les calculs menés sur des génomes entiers et les solutions matérielles spécialisées.

Nous avons conçu et implanté une version optimisée de l'algorithme de Smith/Waterman (SW). Le principe est de vectoriser les opérations élémentaires nécessaires à la construction des cellules de la matrice calculée. En effet, les cellules composant une diagonale de la matrice peuvent être calculées au même instant. Notre implantation tire donc profit d'un parallélisme au niveau du jeu d'instruction des processeurs. L'algorithme de SW a donc été modifié pour en produire une version "vectorisable". Cette version a ensuite été instanciée sur le processeur UltraSparc (Sun), en utilisant le jeu d'instruction VIS (Video Instruction Set). VIS permet de décomposer un mot de 64 bits en plusieurs sous-mots de 8, 16 ou 32 bits, et d'appliquer un opérateur sur chaque sous-mot dans le même cycle d'horloge. Il s'agit d'un micro-parallélisme SIMD. Nous avons considéré des sous-mots de 16 bits, ce qui permet de calculer des scores qui couvrent la quasi-totalité des cas rencontrés. Plus important, l'implémentation assure l'intégrité des calculs en cas de saturation, c'est à dire qu'elle garantit qu'aucun dépassement de capacité ne perturbe le résultat. Cette propriété a été prouvée par J.P. Comet.

Un facteur deux a été gagné sur un UltraSparc à 167 MHZ. La performance atteinte est de 18 millions de cellules de matrices calculées par seconde (MMCS), ce qui est, à notre connaissance, l'implantation

la plus rapide de SW sur station de travail. De plus, elle la propriété d'être basée sur la comparaison de deux séquences. Elle n'est donc pas liée au cas particulier de "database scanning", mais garde au contraire son efficacité dans tous les cas de figures (banque contre banque, Z-score, ...). Elle a donc été intégrée dans LASSAP. Elle tire ainsi parti d'un parallélisme macroscopique. Des mesures de performances réalisées chez Sun France montrent que 200 MMCS sont atteints sur un serveur Enterprise 6000 à 12 processeurs/167 MHZ. Une séquence de 300 acides aminés est comparée à SWISSPROT (release 33) en 30 secondes. Ces performances sont du même ordre de grandeur que celles du matériel spécialisé.

3.3 Vlassap : interface graphique pour LASSAP

Participant : Jean–Marie Geffroy

Dans l'optique d'une interface conviviale pour LASSAP, la solution la plus simple et légère aussi bien du point de vue du temps de développement que de la simplicité d'utilisation consistait naturellement à fournir une interface permettant d'utiliser Lassap sur le Web. L'objectif à court terme est double:

- fournir une interface d'interaction,
- fournir une interface de visualisation.

Pour des raisons de portabibilité et pour satisfaire notamment au second point, le langage Java a été choisi.

Le premier objectif est atteint et une première version va prochainement être mise en service dans le cadre du projet INRIA/INFOBIOGEN/CNUSC. Le second objectif est en cours de réalisation.

La prochaine étape va consister à:

- Enrichir cette interface de manipulation de Lassap de manière à fournir aux biologistes des outils élaborés pour leur permettre de définir des protocoles d'analyse de séquence de façon simple (Workflow). Le public visé n'étant pas un public de programmeurs, nous proposerons un petit système aussi visuel que possible.
- Ajouter des procédures de visualisation d'objets complexes, telles que les pyramides générées par PLATO.

3.4 PROTO2: une base de données de séquences sous O_2

Participants : Marjorie Moulet, Jean-Jacques Codani

Dans le cadre d'un projet commun avec l'INRA Toulouse, financé par le GREG (Groupement de Recherches et d'Etudes sur les Génomes), nous nous sommes intéressés à la réalisation d'une base de données de séquences sous le SGBD-OO O_2, permettant de centraliser l'ensemble des informations contenues jusqu'à présent dans des banques indépendantes. Nous avons implémenté sous O_2 une base de données unique regroupant les quatre banques: SWISSPROT, PRODOM, PROSITE et PDBshort. Ceci permet de manipuler simultanément des informations sur les séquences, les motifs, les domaines et les structures des protéines.

L'implémentation courante est faite sur un server SuperSparc doté de 128 Moctets de mémoire.

Les quatres banques SWISSPROT, PRODOM, PDBSHORT et PROSITE contiennent respectivement 77 Mo, 20 Mo, 31,5 Mo et 1Mo de données, soit au total 132 Mo. Ceci produit une base O_2 de 270 Mo, incluant les références croisées et les index. Le chargement des données s'effectue sous le mode NC_A_NR pour de ne pas surchager le volume "shadow" afin d'améliorer le temps de chargement. Le chargement de l'ensemble s'effectue en 3H30, ce qui est très raisonnable.

Les fichiers d'index sont crées sous le mode le moins sécurisé: NC_NA_NR. Ainsi, la création des index de SWISSPROT est réalisée en moins de cinq minutes. Une fois l'ensemble des index crées,

la résolution des références croisées (avant, arrière, directes et indirectes) s'effectue en une heure de temps.

Une requête simple sur des attributs indexés est quasi-immédiate. *A contrario*, une requête sur des attributs non indexés peut durer plusieurs minutes. Les requêtes complexes ont des temps de réponse variables, suivant la manière dont est écrite la requête. Une requête écrite en O2C est ainsi beaucoup plus rapide, puisque le système peut être guidé.

Grâce au caractère objet de la base, des méthodes simples de pattern matching ont été intégrées facilement. Elles permettent des requêtes calculatoires et donc une grande puissance dans l'expression des requêtes.

Enfin, un serveur W3 a été développé à l'aide d'O2Web et est opérationnel. Il se heurte bien sûr à l'inadéquation entre un système programmable, donc puissant, et les limitations induites par les faibles possibilités de l'interaction graphique.

Bien qu'un tel système permette des requêtes très élaborées, sa mise en service ou sa diffusion reste cependant délicate dans la mesure oèu seuls les producteurs des banques sont réellement à même d'imposer un système propriétaire.

3.5 Significativité des alignements : Z-score

Participants : Jean-Christophe Aude, Jean-Paul Comet, Jean-Jacques Codani

L'algorithme de Smith & Waterman ne fournit pas de probabilités sur les alignements et ne permet donc pas de générer un indice de dissimilarité nécessaire pour réaliser une classification des séquences. Nous avons donc adapté cet algorithme pour approcher la probabilité qu'un tel alignement soit dû ou non au hasard.

L'évaluation de la significativité d'un score est obtenu en calculant un *Z-score*. Pour deux séquences i et j ayant un alignement de score $S(i,j)$, on compare ce score avec les scores obtenus à partir de i et d'un ensemble de séquences aléatoires. Cet ensemble respecte les critères suivants :

- Les séquences sont de longueurs identiques à celle de j.
- Les compositions en acides aminés sont identiques à celle de j.

Le calcul exact de la moyenne et de l'écart-type de cet échantillon n'étant pas possible, on utilise la méthode de *Monte-Carlo* pour estimer ces deux valeurs.

Cependant plusieurs problèmes subsistent: *(i)* existe-t-il un moyen de réduire le temps de calcul ? *(ii)* à partir de quelle valeur (cut-off) un Z-score est-il significatif ? *(iii)* existe-t-il une probabilité associée à ce Z-score?

3.5.1 Analyse de la variance du Z-score

L'étude de la variance du Z-score a permis d'établir une relation entre le nombre N de séquences aléatoires à calculer, la variance estimée du Z-score et le Z-score :

$$\tilde{}\sigma^2(Z(i,j)) = (A \cdot N^B) \times \tilde{}Z(i,j) + \frac{1}{A' \cdot N + B'},$$

oèu $\tilde{}Z(i,j)$ est le Z-score non-approximé (*i.e.* sans utiliser Monte-Carlo) entre les séquences i et j, A et B les paramètres de la régression puissance, A' et B' ceux de la régression polynomiale inverse. L'utilisation de cette équation, en calculant itérativement le Z-score jusqu'à obtenir une variance satisfaisante, permet une réduction importante du temps de calcul et permet d'améliorer la qualité des résultats produits.

3.5.2 Détermination d'une valeur de *cut-off*

Ce que l'on appelle communément le "cut-off" est la valeur du Z-score au-delà de laquelle les alignements sont susceptibles de présenter un intérêt biologique. Jusqu'à présent, cette valeur était déterminée de manière expérimentale par les biologistes. Nous avons établi un modèle permettant de déterminer statistiquement ce "cut-off".

Pour ce faire, on a constitué un modèle de *Bernouilli* basé sur un échantillon de séquences réelles et un second échantillon de séquences aléatoires ayant une distribution en acides aminés réelle (afin de ne pas générer des séquences *absurdes* au sens biologique). On a pu ainsi déterminer une valeur du Z-score qui sépare l'échantillon aléatoire de l'échantillon réel ; cette valeur (au voisinage de 8) est le "cut-off" recherché.

3.5.3 Détermination du modèle probabiliste sous-jacent

On cherche à associer au Z-score une loi de probabilité mesurant l'évènement : "*cet alignement est-il dû au hasard ?*" Le modèle pressenti dans le cas du Z-score est basé sur la loi de valeurs extrèmes type I (loi de Gumble). Nous avons montré que ce modèle n'était pas satisfaisant car il surestimait trop les Z-score supérieurs au "cut-off". Aussi avons-nous déterminé un nouveau modèle à partir de cette distribution et d'une fonction puissance. Ce nouveau modèle corrige la surestimation exposée précédemment.

Nous disposons désormais d'une probabilité. En utilisant la mesure de l'information de Wiener, nous pouvons en déduire un indice de similarité et *a fortiori* un indice de dissimilarité.

3.6 Analyse de données

Participant : Jean-Christophe Aude

3.6.1 Analyse du génome : les familles de séquences

L'analyse de génomes consiste à déterminer les familles de séquences protéiques ayant des fonctions similaires au sein d'un ou plusieurs organismes. Une collaboration étroite est menée avec le CGM sur le génome de la levure. Il s'agit du premier organisme eukaryote séquencé en totalité.

Le nombre très important de résultats produits dans ce cas nous a contraint à adapter les méthodes classiques d'analyse de données. En effet, il faut noter que :

- pour utiliser les techniques de clustering (nuées dynamiques, Analyse Factorielle des Correspondances, ...), on doit disposer d'une distance entre les individus, ce qui n'est pas notre cas.
- Le nombre important d'individus ne permet pas d'analyser de manière fine les classes avec l'analyse hiérarchique ou pyramidale (aucun logiciel ne peut traiter plus de 300 individus).

Ainsi, nous avons du combiner plusieurs méthodes pour trouver les familles de séquences. Les méthodes utilisées sont le clustering au sens de la théorie des graphes, la classification hiérarchique des clusters et la classification pyramidale pour chaque cluster.

3.6.2 Représentation des résultats : la théorie de graphes

Les résultats peuvent être représentés sous la forme d'un graphe $G(S, A)$, oèu S est l'ensemble des séquences (les noeuds) et A l'ensemble des alignements (les arêtes). Il s'agit d'un graphe non-orienté, valué. L'utilisation d'un graphe a permis de construire des composantes connexes, (par augmentation progressive du Z-score). Elles constituent un premier regroupement en famille des séquences. On obtient ainsi une partition de l'ensemble des séquences. Cependant, on a perdu les relations entre les composantes connexes; aussi doit-on procéder à une analyse hiérarchique des composantes connexes pour retrouver les liens existant entre ces diverses composantes.

3.6.3 Analyse hiérarchique

L'analyse hiérarchique est basée sur la distance entre les composantes connexes. On procède alors par agrégation successive des individus pour créer un arbre de proximité de composantes. Le critère choisi est le *min*, c.à.d deux composantes sont agrégées ensemble en fonction de la proximité de ses deux plus proches individus. La détermination des familles de séquences est finalement réalisée en utilisant les pyramides.

3.6.4 Analyse pyramidale

Nous disposons maintenant d'une hiérarchie sur les composantes. Ceci ne nous permet toujours pas d'extraire la décomposition en famille du génome étudié. Les séquences protéiques sont composées de plusieurs fragments ayant des fonctions différentes (*e.g.* une fonction d'activation et une fonction de répression) qui contribuent à une "macro-fonction" plus complexe. Partitionner les composantes avec une hiérarchie nous amenerait alors à affecter arbitrairement certaines de ces séquences à une seule famille, ce qui n'est pas satisfaisant. Il en est de même pour les méthodes d'alignements multiples basées sur des dendogrames.

Aussi devons nous déterminer une partition *recouvrante* permettant ainsi à certaines séquences d'appartenir à plusieurs familles. Les pyramides offrent à cet égard un avantage double, elles fournissent : *(i)* des partitions recouvrantes (un individu peut appartenir au plus à deux classes), *(ii)* un indice de dissimilarité induit par la classification. Ce dernier point nous permet d'estimer la robustesse des familles. Les premiers tests effectués montrent que cette méthode fournit des résultats plus fins que les méthodes classiques d'alignements multiples.

L'ensemble de la procédure décrite précedemment fait l'objet d'une implantation. Il s'agit de la bibliothèque PLATO (Post Lassap Analysis TOolkit).

3.7 Introduction de modèles probabilistes dans l'algorithme de programmation dynamique

Participants : Jean-Paul Comet, Jacques Henry

Le but est ici de prendre en considération dans l'algorithme de programmation dynamique, l'information qu'on peut avoir en comparant la séquence à un modèle markovien. La méthode consiste tout d'abord à apprendre un modèle probabiliste sur un ensemble de données (une famille homogène, des parties de séquences à privilégier, des motifs, ...). Ce modèle peut être une chaîne de Markov, ou bien un modèle de Markov caché. On pondère alors l'algorithme de la programmation dynamique suivant l'erreur de prédiction du modèle. Le coeur de l'algorithme n'est pas modifié, mais le choix de la fonction de pondération est délicat.

La première extension de cette méthode est de considérer comme domaine d'apprentissage, non pas un ensemble de séquences entières, mais le résultat d'une première passe de la programmation dynamique. Ainsi, on considère comme domaine d'apprentissage l'ensemble des k meilleurs alignements donnés par l'algorithme SIM de Xiaoqiu Huang et Webb Miller. Les résultats intermédiaires conjugués avec ceux de l'étude préliminaire sont encourageants : nous avons pris pour tests les tRNA-synthétases qui comportent dans les alignements de très longs trous (gaps en anglais). La chaîne de Markov apprise sur les alignements suboptimaux reconnaît une partie des zones à aligner.

Une autre approche utilise la modélisation des séquences biologiques par des chaînes de Markov cachées. Chaque zone bien conservée ou domaine d'une famille de séquences permet la construction d'une chaîne de Markov cachée. On crée ensuite un modèle complet en reliant chacun de ces modèles. L'algorithme de Viterbi permet, pour une séquence donnée, de déterminer la suite des états cachés et d'en déduire les domaines. On va pouvoir en fonction de la zone dans laquelle on se trouve, pondérer la programmation dynamique, et ainsi aligner la séquence inconnue avec la famille dont les domaines sont déjà connus, et cela indépendamment des longueurs de gaps.

3.8 Un algorithme de comparaison par programmation dynamique prenant en compte les motifs

Participants : Jean-Paul Comet, Jacques Henry

La méthode proposée vise à simuler la démarche du biologiste qui, pour aligner deux séquences va mettre en correspondance les motifs qu'il sait être pertinents pour les séquences considérées. On pense ici typiquement à des séquences protéiques et à l'utilisation de motifs décrits dans la base PROSITE. La méthode ne vise pas à *imposer* l'alignement de ces motifs car dans le cas oèu ces motifs apparaîtraient dans un ordre différent sur les deux séquences, on aboutirait à une impossibilité. La méthode consiste donc à réaliser un alignement par programmation dynamique lettre à lettre au sens de Needleman et Wunsch, en attribuant un bonus B supplémentaire à tout chemin mettant en correspondance le même motif sur les deux séquences.

La première phase consiste donc à repérer les occurrences des motifs de la base sur les deux séquences à l'aide d'un algorithme de "pattern matching".

On suppose ici pour simplifier que le motif apparaît avec la même longueur sur les deux séquences. Si $S_{i,j}$ désigne le score optimal à la position (i, j), $d(a_i, b_j)$ le coût de la substitution de a_i par b_j, r le coût d'insertion, on a la récurrence :

$$S_{i,j} = \max(S_{i-1,j-1} + d(a_i, b_j), S_{i-1,j} + r, S_{i,j-1} + r)$$

en tout point (i, j) sauf ceux correspondant à la fin d'un motif détecté sur les deux séquences. En un tel point, on a le quadruple choix suivant, en supposant le motif de longueur l :

$$\begin{aligned} S_{i,j} = \quad & \max(S_{i-1,j-1} + d(a_i, b_j), S_{i-1,j} + r, S_{i,j-1} + r, \\ & S_{i-l+1,j-l+1} + \textstyle\sum_{k=0}^{k=l-1} d(a_{i-k}, b_{j-k}) + B) \end{aligned}$$

Dans le cas d'un motif à longueur variable, la difficulté vient de l'alignement de deux occurrences du même motif. Il ne suffit pas de connaître l'occurrence du motif et sa longueur, mais il faut aussi connaître la structure du motif, pour pouvoir positionner les insertions aux bons endroits.

La méthode a été testée pour un petit nombre de cas et semble fournir des résultats prometteurs. Cependant la difficulté principale vient de la détermination des bonus à attribuer à chaque motif. Pour la lever, il faut envisager une étude à grande échelle. Chaque motif se verrait affecter d'une valeur différente prenant en considération, sa longueur mais aussi sa signification biologique.

3.9 BlastMultAl

Participant : Pierre Nicodème

Pour ce qui concerne les algorithmes de type *Blast*, Pierre Nicodème constitue dans le cadre de sa thèse un document regroupant les développements mathématiques permettant d'obtenir les formules des valeurs extrêmes (Karlin-Iglehart) donnant la pertinence probabiliste du score de similarité de deux séquences. D'autre part, le logiciel *BlastMultAl* (anciennement *BlastProDom*), qui permet d'effectuer des recherches de similarité entre une séquence et une famille de séquences, est en cours d'adaptation, de façon à ce qu'il puisse traiter les familles contenant des gaps, telles qu'on en trouve dans la dernière version (version 33) de la base de domaines protéiques *ProDom*; ce travail est effectué en collaboration avec l'INRA-Toulouse (Daniel Kahn et Florence Corpet en particulier).

3.10 Méthodes formelles d'analyse du génome

Participants : Mireille Régnier, Fariza Tahi

Nous nous intéressons à deux problèmes différents d'analyse des séquences génétiques, et plus précisément des séquences d'ARN. Il s'agit du problème de prédiction de structures secondaires d'ARN, et

du problème d'énumération de ces mêmes structures. L'approche que nous avons utilisée pour la résolution du premier problème relève de l'algorithmique, tandis que celle utilisée pour le second problème relève de l'analyse combinatoire.

Nous avons développé un algorithme de prédiction de structures secondaires basé sur l'approche comparative. Celle-ci consiste à considérer un ensemble de séquences d'un même ARN appartenant à des espèces vivantes de même famille, pour lesquelles est recherchée la structure secondaire commune. Cette approche a été utilisé *manuellement* par des biologistes pour la détermination des structures secondaires de certains ARN ribosomaux.

Notre algorithme réalise la recherche des hélices communes en trois étapes : l'étape de recherche des palindromes dans l'une des séquences considérées, l'étape de *comparaison* avec les autres séquences permettant de déduire les palindromes *conservés*, puis l'étape de sélection des palindromes *structurants*, c'est-à-dire des palindromes susceptibles de définir des hélices.

La sélection des palindromes structurants est basée sur des critères de longueur et de nombre de mutations compensatoires. La mise en œuvre de la recherche des palindromes structurants est basée sur l'approche "diviser pour régner". Un palindrome trouvé permet en effet de diviser la séquence initiale en deux sous-séquences indépendantes, celle qui lui est interne, et celle résultant de la jonction des deux sous-séquences qui lui sont externes. Nous recherchons ainsi des *points d'ancrage*, c'est-à-dire des palindromes satisfaisant des critères de sélection prédéfinis, qui permettent de subdiviser la séquence en sous-séquences de plus petites tailles oèu sont recherchés d'autres points d'ancrage vérifiant des critères de sélection plus faibles.

Nous avons testé et validé notre algorithme sur un ensemble de séquences d'ARN ribosomal 16S et un ensemble de séquences d'ARN ribosomal 23S, oèu nous avons réussi à isoler la majorité des hélices des structures secondaires communes associées à chacun de ces deux ensembles. Les résultats que nous obtenons sur ces deux ARNs sont donc très satisfaisants, et nous comptons tester notre algorithme sur d'autres ARNs ribosomaux.

La seconde partie de notre travail concerne l'énumération de certaines structures apparaissant dans une séquence de nucléotides, et plus particulièrement de structures secondaires. Nous généralisons les résultats de Waterman, qui s'était initialement intéressé à ce problème. Waterman a utilisé une approche basée sur la récurrence, qui présente l'inconvénient d'être limitée. Les généralisations dans la forme des structures considérées rendent très vite les expressions récurrentes associées difficiles à traiter. Nous proposons l'utilisation d'un outil mathématique très intéressant, les séries génératrices, grâce auquel cette généralisation devient implicite [5]. Nous montrons ainsi l'apport des séries génératrices dans la résolution de ce type de problèmes. (voir rapport d'activité du projet "Algorithmes").

3.11 Programmation dynamique et factorisation d'opérateurs elliptiques

Participant : Jacques Henry

La méthode de base de comparaison de séquences biologiques qui repose sur la maximisation des scores des opérations d'édition élémentaires, est fondée sur le principe de la programmation dynamique. Une réflexion a donc été menée sur les méthodes de programmation dynamique et de plongement invariant, qui ont des applications dans des domaines très variés. En particulier, les travaux menés précédemment au sein du projet SOSSO sur la factorisation d'opérateurs elliptiques du second ordre ont été poursuivis. On a ainsi mis en évidence par la méthode du plongement invariant dans une famille de problèmes définis sur des domaines variables, qu'un problème elliptique du second ordre peut se factoriser en un produit de deux problèmes du premier ordre non couplés et qui font intervenir un opérateur P vérifiant une équation de Riccati. Cet opérateur relie les conditions de Dirichlet et de Neumann sur la frontière variable. Sur le problème discrétisé, cette factorisation s'interprète exactement comme la factorisation de Gauss par blocs de la matrice du système linéaire. Ces travaux menés initialement sur des ouverts cylindriques ont été généralisés à d'autres géométries. Enfin ce découplage a été interprété comme celui d'un problème de contrôle qui a été explicité ce qui permet d'envisager une généralisation aux problèmes non linéaires. Ces travaux ont donné lieu à la thèse d'A. Ramos soutenue à l'université de

Madrid. Ces travaux conduisent aussi à l'interprétation de la factorisation de Gauss comme méthode de programmation dynamique. Ils permettent de faire un lien entre la programmation dynamique utilisée pour la comparaison de séquences, pour le contrôle optimal et pour l'estimation des états cachés d'un modèle de Markov caché par l'algorithme de Viterbi.

3.12 Etude fonctionnelle de la boucle V3 de la protéine d'enveloppe du VIH1 (SIDA)

Participant : René Devilliers

Nous nous attachons a' la définition d'une base de données décrivant les paramètres de calcul conformationel de tous les acides aminés chimiquement modifiables devant conduire à la synthèse de peptides conformationnellement contraints dans la forme 3D (déjà déterminée) d'un modèle antigénique ou pharmacologique. Des applications sont prévues pour les maladies liées aux PRIONS et au VIH.

Quand la détermination de modèles structuraux et fonctionnels de la boucle hypervariable V3 appartenant à la séquence de la protéine d'enveloppe gp120 du VIH sera réalisée, une nouvelle étape sera nécessaire. Elle consiste en la synthèse et en des tests biologiques de peptides choisis pour être conformationnellement contraints dans la forme fonctionnelle et antigénique du modèle.

Pour augmenter les chances de sélectionner des formules chimiques pertinentes pour ces molécules, on pourra remplacer en chaque position de séquence du peptide modèle in vivo (en fait la boucle V3) un acide aminé normal par un acide aminé modifié possédant localement les mêmes fonctions d'interaction et de reconnaissance moléculaires.

Cette dernière réflexion a conduit à envisager toutes les modifications chimiques possibles des 20 acides aminés standards. Ceci a permis de définir environ 3500 formes distinctes et communes de squelette d'acides aminés auxquelles s'ajoutent un certain nombre de modifications spécifiques de leur chaines latérales. En théorie et à partir de la connaissance d'un modèle, on pourra sélectionner une solution parmi 3500x100 combinaisons pour chaque position d'acide aminé dans une séquence quelconque. Pour un peptide de 35 AA (comme la boucle V3) on pourrait choisir une solution parmi au moins $10^{1}75$ possibilités en utilisant un algorithme de comparaison de structures fonctionnelles. Ceci nécessite le calcul préalable des conformations les plus stables de (3500+100) fragments moléculaires qui représentent le squelette et les chaînes latérales des nouveaux acides aminés modifiés.

La même base de données pourra être utilisée pour les problèmes de détermination de la structure chimique de peptides devant correspondre à des modèles pharmacologiques d'interaction connus. De plus, pour cette problématique, la connaissance a priori, des conformations moléculaires d'acides aminés modifiés, lorsqu'elles sont peu nombreuses, permet de concevoir et valider (sur le plan expérimental) des modèles physiques d'interaction.

D'autre part, cette méthode pourra aussi être appliquée à la détermination de la structure chimique de peptides antigéniques analogues, sur le plan de la reconnaissance moléculaire, à des fragments (á préciser)! des protéines PRIONS transformées.

Ce travail qui correspond en fait à la deuxième partie du projet de définition d'antigènes conformationels de la boucle V3 à large pouvoir de neutralisation croisée (qui a été déjà présenté), permet d'intégrer les acides aminés modifiés dans les calculs énergétiques de peptides et, contribue ainsi à l'unification des méthodes de calcul conformationel et fonctionnel.

4 Actions industrielles

Une action industrielle est menée pour le logiciel LASSAP. En effet, celui-ci répond aux besoins:

1. des projets scientifiques en biologie moléculaire qui requièrent du calcul intensif (construction de familles, ...)
2. des centres de séquençage à haut débit (plusieurs centaines de milliers de paires de bases par jour),

3. des centres de services désirant offrir des temps de réponse de qualité,
4. des centres de bioinformatique en charge de la construction et de la maintenance des banques de séquences.

LASSAP peut ainsi répondre aux besoins des sociétés de biotechnologie qui ont entrepris du séquençage systématique. Il ne s'agit pas ici d'un marché de masse, mais d'une niche à haute valeur ajoutée. A plus long terme, l'industrie pharmaceutique et agro-alimentaire est visée. Il faut également noter qu'une version de LASSAP sur PC/Macintosh est envisageable et comblerait un vide important. Enfin, la compacité du code permet d'envisager des solutions embarquées utilisables par l'industrie du diagnostic (diagnostic moléculaire). Pour ce faire, un dossier d'aide au transfert a été soumis et accepté par l'ANVAR. Parallèlement, diverses actions sont menées auprès de constructeurs d'ordinateurs désireux de proposer des solutions performantes sur leurs architectures. Ces acteurs constituent de plus des vecteurs potentiels de diffusion auprès de l'industrie pharmaceutique. Ainsi:

– LASSAP est référencé dans la brochure "IBM Solutions for Computational Biology and Bioinformatics".
– Un "SUN ACADEMIC EQUIPMENT GRANT (AEG) PROGRAM" est en cours de négociation avec Sun Microsystems. Il concerne les optimisations de LASSAP en VIS effectuées par Andrzej Wozniak.

5 Actions nationales et internationales

L'action a bénéficié de deux contrats du GREG. Une subvention du Ministère de la Recherche par l'ACC-SV13 est en cours. Une subvention de l'Union Européenne vient d'être accordée pour l'utilisation de LASSAP par l'EBI et l'Université de Genève (A. Bairoch).

J.J. Codani est membre de l'OFTA (Observatoire Français des Techniques Avancées), groupe "ordinateurs massivement parallèles".

J.J. Codani, J. Henry et M. Scholl sont experts pour l'ACC-SV13 du Ministère de la Recherche.

M. Scholl et J.J. Codani sont membres du comité de pilotage du GIS INFOBIOGEN.

6 Diffusion des résultats

6.1 Actions d'enseignement

– J. Henry assure un enseignement spécialisé de calcul scientifique à l'Ecole des Mines de Paris.
– J.P. Comet assure des TD de statistique à l'Université de Nanterre.
– E. Glémet assure l'enseignement d'algorithmique de la Miage de l'Université de Créteil.
– J.C. Aude assure des TD de bioinformatique à l'Université Paris 6.

6.2 Participation à des conférences et colloques

Des membres de l'équipe ont participé à des conférences et *workshops* ; on se reportera à la bibliographie pour en avoir la liste.

J.J. Codani et E. Glémet ont été hébergés par IBM et Compugen Ltd. afin d'effectuer des démonstrations de LASSAP lors de la "Eighth International Genome Sequencing and Analysis Conference", Hilton head, SC, october 1996.

6.3 Diffusion de produits

Le logiciel de multiplexage de PCR (Réactions de Polymérisation en Chaîne) *Multipcr*, écrit par Pierre Nicodème a été mis en service sur le site WEB d'Infobiogen.

LASSAP a été installé au CNUSC (Montpellier) doté d'un IBM SP2 80 processeurs, à INFOBIOGEN et à l'INRA Toulouse.

7 Publications

Articles et chapitres de livre

[1] E. GLÉMET, J. CODANI, «LASSAP : a large scale sequence comparisons package», *Comp. Appl. BioSci.*, 1996, in press.

[2] A. WOZNIAK, «Using Video Oriented Instructions to Speed-Up Sequence Comparison», *Comp. Appl. BioSci.*, 1996, in press.

Communications à des congrès, colloques, etc.

[3] J. AUDE, J. COMET, «Identification of Protein families», *in : SwissProt*, november 1996. Jerusalem.

[4] E. GLÉMET, J. CODANI, «LASSAP : a large scale sequence comparisons package», *in : International Symposium on Theoretical and Computational Genome Research*, March 1996. Heidelberg.

[5] M. RÉGNIER, F. TAHI, «Enumeration and Asymptotics in Computational Biology», *in : Workshop "Mathematical Analysis of Biological Sequences"*, Trondeim, Norvège, 4-6 August 1996.

8 Abstract

One of the main goal of the Action ATGC is to develop the LASSAP software. LASSAP is a software package for sequence comparison. It is a programmable, high performance system. It provides an API (Application Programming Interface) allowing the integration of any generic pairwise-based algorithm. This API allows to program the large number of special purpose tasks that involve pairwise sequence comparisons without having to handle input/output or flow control tasks.

LASSAP is targeted to:

- Genomic projects which require intensive comparisons (clustering of databanks, ...). LASSAP is used in the building of PRODOM, and the exhaustive comparisons of yeast sequences.
- Sequencing centers with high throughput.
- Centers in charge of the production of international databases (redundancy, ...). LASSAP is used in the sub-fragment matching problem of TREMBL (EBI supplement of SWISS-PROT).
- Centers of services willing to provide high-end performance.

Around LASSAP, methodological researches are led. They focus on:

- significance of alignment scores;
- combination of classification methods;
- new sequence comparison and pattern matching algorithms;
- prediction of secondary structures of RNA;