Interrogation de sources biomédicales : prise en compte des préférences de l'utilisateur

Sarah Cohen Boulakia*, Christine Froidevaux*, Séverine Lair**

*Laboratoire de Recherche en Informatique, CNRS UMR 8623
Université Paris-Sud, 91405 Orsay Cedex
{cohen, chris}@lri.fr
** Institut Curie, CNRS UMR 144
26 rue d'Ulm, 75248 Paris Cedex 05
severine.lair@curie.fr

Résumé. Nous nous plaçons dans le cadre d'un projet de constitution d'une plate-forme intégrative de données biomédicales pour l'étude génomique des cancers. La plate-forme comporte, entre autres, un certain nombre de scénarios d'analyse qui sont proposés à l'utilisateur. A chaque étape d'un scénario qu'il a choisi de réaliser pour les besoins de son étude, l'utilisateur peut être amené à poser une requête nécessitant d'accéder à différentes sources et il doit alors choisir les sources pertinentes. Nous proposons un guide à l'utilisateur sous forme d'un algorithme de sélection de sources adapté à sa requête et à ses préférences. Pour cela, nous explorons quelques spécificités des banques de données biomédicales et définissons différents critères de préférence utiles pour les biologistes. Nous illustrons notre démarche avec un exemple de requête biomédicale.

1. Introduction

Nous nous plaçons dans le cadre du projet européen HKIS qui vise à la constitution d'une plateforme intégrative de données biologiques et biomédicales pour l'étude génomique des cancers. L'objectif de la plate-forme est de permettre à ses utilisateurs d'analyser leurs résultats d'expérience en les combinant avec d'autres données présentes dans des sources accessibles par le Web. La plateforme offre à chaque utilisateur la possibilité de rapatrier sur son espace de travail un ensemble de banques publiques, le temps d'un traitement. Le besoin de travailler en local répond à une double demande : faire tourner des traitements complexes sur un important volume de données et assurer une confidentialité totale sur les données et les outils manipulés durant les traitements. Cet aspect confidentialité est tout à fait crucial pour les biologistes travaillant sur des données sensibles comme les gènes impliqués dans le cancer du sein par exemple. La plateforme a aussi pour but d'offrir la possibilité d'exporter les résultats obtenus vers les banques privées des utilisateurs. Elle comporte en outre un certain nombre de *scénarios d'analyse,* décrivant différentes méthodologies d'analyse des données, qui sont proposés à l'utilisateur. A chaque étape d'un scénario qu'il a choisi, l'utilisateur peut être amené à poser une requête nécessitant d'accéder à différentes sources. Se pose alors le problème du choix de ces sources.

Dans cet article, nous ne détaillons pas l'architecture de la plate-forme ni ne précisons comment les scénarios d'analyse sont recueillis, puis représentés et implémentés dans celle-ci. Nous nous intéressons ici à la phase d'interrogation et visons à améliorer sa qualité en

guidant l'utilisateur dans son choix des sources. Pour cela, nous proposons à l'utilisateur un algorithme de sélection de sources qui prend en compte à la fois sa requête et ses préférences. Dans un premier temps, nous explorons quelques spécificités des banques de données biomédicales (section 2) et introduisons un exemple biologique motivant notre démarche (section 3). Nous décrivons dans la section 4 un algorithme qui calcule les diverses combinaisons de sources permettant de répondre à une requête. Nous définissons ensuite dans la section 5 différents critères de préférence utiles pour les biologistes et montrons comment ils peuvent être utilisés pour sélectionner et ordonner les diverses combinaisons de sources. Nous illustrons notre démarche avec un exemple de requête biomédicale (section 6).

2. Exploration des spécificités des sources biomédicales

2.1. Des sources de contenu volumineux et varié

Les sources biomédicales sont extrêmement variées au niveau de leur contenu. Certaines, qui contiennent des informations de base (sur les gènes, les protéines et les séquences), sont massivement utilisées, comme GenBank qui est la banque de référence. Néanmoins, ces sources sont de taille très importante et leurs données n'ont pas toujours été **validées,** c'est-à-dire, vérifiées par un ensemble d'experts. Le biologiste doit alors interroger d'autres banques plus spécialisées ou qu'il juge plus sûres, pour obtenir des informations plus précises ou pour pouvoir confronter entre elles les informations récupérées des diverses sources. En effet, d'une part, chaque source est agencée autour d'une entité biologique - que l'on appellera son **focus** - et offre ainsi un point de vue différent sur les données. D'autre part, chaque biologiste a un groupe de bases de prédilection, qu'il utilise fréquemment et en lesquelles il a confiance. Cette confiance en une banque est d'autant plus grande que l'utilisateur pense que les données qu'elle fournit ont été soigneusement validées. Dans le cas de réponses divergentes (car si les banques fournissent des informations essentiellement complémentaires voire redondantes, elles sont parfois aussi divergentes), le biologiste sera amené à privilégier les réponses obtenues à partir des bases en lesquelles il a le plus confiance.

2.2. Des sources autonomes mais reliées

Chaque source de données biomédicale est conçue et construite indépendamment des autres. Néanmoins, un effort très important est fait pour relier les données entre banques par des **références croisées**. C'est l'une des particularités majeures des sources biomédicales. Notre objectif dans cet article est d'offrir un guide à l'utilisateur dans le choix des sources susceptibles de lui fournir des données pertinentes en réponse à sa requête. On se place donc en amont des données, au niveau du type des données contenues dans ces sources. Dans ce cadre, on exploitera les liens entre les données à un niveau méta en considérant que les banques se référencent. Nous avons vu que les sources pouvaient être plus ou moins validées : il en est de même pour les liens de références croisées, selon qu'ils ont été ajoutés manuellement (liens sûrs) ou générés automatiquement (liens non sûrs).

Dans le cadre du projet HKIS une trentaine de banques a été sélectionnée, comme étant utilisées fréquemment par les partenaires de la communauté biomédicale du projet (*Institut Curie, Paris (France), Université et Faculté de Médecine d'Ulm (Allemagne), Institut*

Européen d'Oncologie de Milan (Italie)). Ces banques constituent les nœuds d'un graphe dont les arcs sont les liens de références croisées. Nous n'en présentons ci-dessous qu'un sous-graphe (FIG. 1). On a indiqué sur la figure, pour chaque banque, le niveau de validité établi par les partenaires (nombre compris entre 1 et 10), la liste des entités biologiques sur lesquelles elle fournit des informations, en commençant par l'entité focus (en gras), et pour chaque lien s'il est sûr ou non (lien non sûr en pointillés).

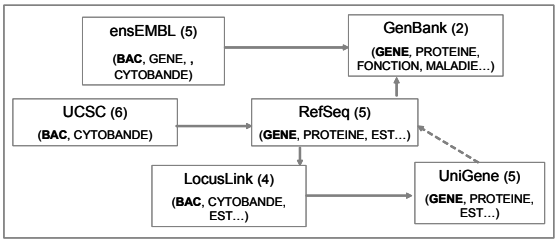

FIG. 1 – Graphe des banques

3. Interrogation de données biologiques : un exemple

Certains mécanismes du cancer sont aujourd'hui connus [Veale *et al.*, 1987] : les cellules tumorales ont des caractéristiques génomiques, *i.e.* portant sur les chromosomes, différentes de celles des cellules de tissu sain. Lors de l'étude des mécanismes d'un cancer les biologistes cherchent à corréler les données biologiques (génome, transcriptome et protéome) et les données médicales (tumeur) de patients, le plus souvent, en utilisant la technologie des biopuces. L'analyse des résultats obtenus avec ces puces est complexe et comporte de nombreuses étapes. Parmi elles, on trouve l'utilisation d'outils statistiques pour normaliser et analyser les données mais aussi la recherche, à travers les banques publiques, d'informations précises sur les morceaux de chromosomes (BAC[1]) qui semblent avoir été modifiés dans les cellules cancéreuses. Cette recherche est notamment présente dans le « Bac Augmentation Scenario » (mis au point dans le projet HKIS) qui comporte une étape visant *à partir d'un numéro de BAC, à retrouver des informations sur celui-ci et sur les gènes qu'il peut contenir*. C'est sur ce type de requêtes que nous nous focalisons dans cet article.

Les partenaires du projet HKIS ont dégagé l'ensemble des entités biologiques qui doivent pouvoir être manipulées lors de l'étude des cancers. Ces entités ne sont pas indépendantes les unes des autres et forment un graphe dont un sous-graphe est donné en FIG. 2. Ce graphe est un réseau sémantique dont les nœuds sont des entités biologiques et les arêtes les relations entre ces entités. Ce réseau constitue le méta-modèle de la plateforme. Il va permettre à l'utilisateur de mettre en évidence et de sélectionner les entités biologiques sous-jacentes à ses requêtes. Dans l'exemple précédent, BAC et GENE sont les **entités sous-jacentes** (en foncé sur le graphe) **à la requête** « *Retrouver des informations sur un BAC ainsi que sur les gènes qu'il peut contenir à partir du numéro du BAC* ». A partir de ces entités, nous avons élaboré un algorithme de construction de chemins où chaque chemin indique à l'utilisateur dans quelle banque trouver chacune des entités sous-jacentes à sa requête et dans quel ordre consulter les banques. Cet algorithme est présenté dans la section suivante.

[1] Bacterial Artificial Chromosome

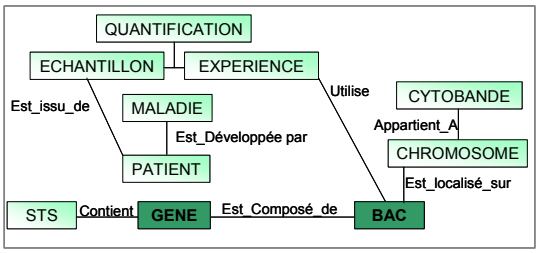

FIG. 2 – Graphe des entités biologiques

4. Construction de chemins dans un graphe

4.1. Objectif : suivre la démarche du biologiste

L'algorithme dont nous donnons la spécification et la description dans les sous-sections suivantes a été construit en suivant la démarche des biologistes lors de la recherche d'informations dans les sources. Lorsqu'un biologiste a mis en évidence les entités qu'il recherche, il essaie de trouver un ensemble de banques lui fournissant des informations sur celles-ci en suivant des liens de références croisées. Il est tout à fait probable que chaque banque ne pourra lui offrir des informations que sur un sous-ensemble d'entités mais l'ensemble des sources interrogées devra lui permettre d'avoir *in fine* des informations sur l'ensemble de ces entités (*). Sa démarche consiste à prendre en compte les entités l'une après l'autre et à rechercher des informations sur chacune des entités. Il peut être amené à suivre des liens de références croisées entre banques pour des informations concernant une même entité. Il collecte alors un maximum d'information sur chacune des entités avant de passer à la suivante (**). Notons qu'il peut considérer plusieurs fois une même banque si elle lui offre des informations sur des entités différentes (***). Enfin, pour être exhaustif, il prend en compte tous les ordres possibles entre celles-ci.

4.2. Spécification de l'algorithme

Pour plus de lisibilité nous utiliserons les notations suivantes : $E=\{e_1,\ldots,e_n\}$ est l'ensemble des n noeuds du graphe des entités biologiques, $E_R=\{e_{q1},\ldots,e_{qnr}\}$ est l'ensemble des nr entités sous-jacentes à la requête R de l'utilisateur ($E_R \subseteq E$) et $B=\{b_1,\ldots,b_m\}$ est l'ensemble des m banques du graphe des banques. On appellera **chemin banques_entités** toute suite de couples (b,e) de B x E telle que l'entité e est contenue dans la banque b et telle que si deux couples (b_{i1},e_{i1}), (b_{i2},e_{i2}) sont consécutifs, alors $b_{i1}=b_{i2}$ ou il y a un arc de b_{i1} vers b_{i2} dans le graphe des banques (lien de référence croisée). Chaque couple (b,e) d'un tel chemin renvoyé par l'algorithme signifie que l'utilisateur peut utiliser une vue de la banque b sur l'entité e pour obtenir des instances de l'entité e. De plus, l'ordre des couples du chemin indique l'ordre dans lequel les combinaisons entre les données de ces banques doivent être réalisées.

Plus précisément, l'algorithme retourne l'ensemble $L=\{chemin_1,\ldots,chemin_k,\ldots chemin_t\}$ de tous les ***chemins banques_entités_complets*** qui sont les chemins banques_entités qui vérifient les 3 propriétés ci-dessous :

(1) Les chemins de L contiennent toutes les entités sous-jacentes à la requête de l'utilisateur : pour tout $chemin_k$ de L, $1 \leq k \leq t$, pour toute entité e sous-jacente à la requête, il existe (au moins) un couple (b,e) dans $chemin_k$;

(2) Les chemins de L regroupent les banques autour d'une même entité en considérant chaque entité une et une seule fois : entre deux couples relatifs à une même entité e dans un chemin donné, il n'existe pas de couple relatif à une entité e', avec $e \neq e'$;

(3) Un couple (b,e) apparaît au plus une fois dans un chemin de L.

La propriété (1) répond au besoin (*) vu précédemment, (2) à (**) et (3) à (***).

4.3. Algorithme de construction de chemins

L'algorithme de construction de chemins banques_entités_complets se compose d'une procédure principale *Chem_Banque_Entite_Complet* qui construit l'ensemble des chemins caractérisés dans la sous-section précédente. Il est important de noter que les chemins ne sont pas construits en naviguant dans le graphe des entités puisque l'on considère les entités sous-jacentes à la requête l'une après l'autre indépendamment des liens sémantiques du réseau. On ne se ramène pas non plus à un simple parcours en profondeur du graphe des banques car notre algorithme, respectant la démarche du biologiste, est centré sur les entités. Il comporte, en effet, deux étapes. D'abord, la procédure *Chem_Centrés_Entité* construit l'ensemble des **chemins centrés sur une entité**, c'est-à-dire, des chemins banques_entités dont tous les couples concernent une même entité (cf. (2)). Ensuite, la procédure *Construction_Rec* construit récursivement les chemins banques_entités_complets qui sont des combinaisons de chemins centrés sur une entité (cf. (1)). La procédure principale est présentée ci-dessous et les deux procédures qu'elle appelle sont détaillées en Annexe 1.

Procédure Chem_Banque_Entite_Complet
Entrée : E_R : Ensemble des entités sous-jacentes à la requête R
Sortie : CHEM_COMPLET : Ensemble de tous les chemins banques_entités_complets relatifs aux entités de E_R
Début
  // Pour chaque entité e, on calcule tous les chemins possibles entre les banques fournissant cette
  // entité Pour toute entité e ∈ E_R faire
    B ← *Banq_Contenant* (e) ; // B : ensemble des banques contenant des données sur e
    Chem_Centrés_Entité (e, B, TAB_CHEM_CENTRE (e)) // On stocke tous les chemins
    // entre banques, relatifs à l'entité e, à l'indice e d'un tableau d'ensembles de chemins
  // On calcule toutes les combinaisons entre chemins centré-entité
  Construction_Rec (E_R, TAB_CHEM_CENTRE, CHEM_COMPLET)
Fin Chem_Banque_Entite_Complet

4.4. Illustration sur l'exemple

Nous illustrons ici notre algorithme sur la requête (introduite en section 3), qui a pour entités sous-jacentes BAC et GENE. On considère l'ensemble B des banques de FIG.1 ainsi que les entités qu'elles contiennent comme indiqué sur la figure. On désignera par la suite les

entités BAC et GENE respectivement par b et g et les banques UCSCGenome, LocusLink, ensEMBL, GenBank, RefSeq et UniGene respectivement par *UCSC*, *LL*, *ENS*, *GB*, *RS* et *UG*.

La première étape de l'algorithme *Chem_Banque_Entite_Complet* consiste à construire les ensembles CHEM_CENTRE(b) et CHEM_CENTRE(g) que l'on désignera dans la suite par CC(b) ={ch, ch', ch''} et CC(g) = {ch1,ch2,ch3,ch4,ch5,ch6,ch7,ch8}

avec : ch=[(*ENS*, b)], ch'= [(*LL*, b)], ch''=[(*UCSC*, b)] et ch1=[(*GB*, g)], ch2=[(*UG*,g)], ch3=[(*RS*,g)], ch4=[(*ENS*, g)], ch5=[(*UG*, g),(*RS*, g)] , ch6=[(*ENS*,g), (*GB*, g)], ch7=[(*RS*, g), (*GB*, g)], ch8=[(*UG*, g),(*RS*, g), (*GB*, g)].

Ainsi, pour avoir des informations sur les BACs, les banques ensEMBL, LocusLink, et UCSCGenome peuvent être consultées seules. Pour obtenir des informations sur les gènes, deux types de chemins sont possibles : soit des chemins où une seule banque est consultée (GenBank, UniGene, RefSeq ou ensEMBL), soit des chemins où des références croisées entre les banques peuvent être suivies (par exemple, en suivant le lien de UniGene vers RefSeq).

Dans la seconde étape de l'algorithme, les chemins de CHEM_COMPLET (ensemble des réponses) sont construits à partir de CC(b) et CC(g) en concaténant, lorsqu'une référence croisée le permet ou lorsque l'on reste dans une même banque, les chemins de CC(b) avec ceux de CC(g), puis on fait de même en considérant CC(g) puis CC(b). Ainsi, CHEM_COMPLET se compose des chemins suivants que nous avons numérotés de 1 à 11: 1=[ch,ch1], 2=[ch,ch4], 3=[ch,ch6], 4=[ch',ch2], 5=[ch',ch5], 6=[ch',ch8], 7=[ch'',ch3], 8=[ch'',ch7] de CC(b)xCC(g) et 9=[ch3,ch'], 10=[ch4,ch], 11=[ch5,ch'] de CC(g) x CC(b).

En définissant la **taille** d'un chemin comme le nombre de références croisées suivies, on peut noter que l'algorithme renvoie des chemins de taille 0 comme le chemin 2=[(*ENS*,b),(*ENS*,g)], de taille 1 comme le chemin 7=[(*UCSC*,b),(*RS*,g)] et de taille 3 comme le chemin 6=[(*LL*,b), (*UG*,g), (*RS*,g), (*GB*,g)].

4.5. Complexité et améliorations possibles

4.5.1 Complexité

Etudier la complexité de *Chem_Banque_Entite_Complet* nécessite de compter le nombre de chemins générés par l'algorithme. Le pire des cas est celui où toutes les banques se référencent mutuellement, toutes les combinaisons entre chemins centrés étant alors possibles. Néanmoins, on considère le cas où toutes les banques ne fournissent pas toutes les entités. Dans ce cas, le nombre de chemins construits par l'algorithme est :

$$C = (nr!) * \prod_{i=1}^{nr} \sum_{k=1}^{nbe_i} \mathrm{A}_{nbe_i}^{k}$$

où nr est le nombre d'entités prises en entrée par l'algorithme, et nbe_i est le nombre de banques contenant e_i ($1 \le i \le nr$, $nbe_i \le m$). La somme correspond au nombre de chemins centrés sur l'entité e_i, obtenus par la procédure *Chem_Centré_Entité* pour e_i. Le reste de la formule correspond au nombre de combinaisons de ces chemins centré-entité obtenues dans la procédure récursive, en considérant tous les ordres possibles des entités. La complexité en temps est donc très grande. Toutefois le nombre d'entités manipulées et de banques étant petit (de l'ordre de 5 entités et de 10 banques par requête), cette complexité reste raisonnable en pratique. Dans notre exemple, si le graphe des banques avait été complet avec nr=2, nbe_1=3 et nbe_2=4 on aurait eu $C = 2! * (\sum_{k=1}^{3} A_3^k) * (\sum_{k=1}^{4} A_4^k) = 2 * 60 * 64 = 7680$. Compte tenu des

références croisées réelles, seulement onze chemins banques_entités_complets ont été construits par *Chem_Banque_Entite_Complet*.

4.5.2 Améliorations possibles

Le nombre de chemins pouvant rapidement être trop élevé pour que le biologiste puisse tous les traiter, nous avons cherché à dégager un sous-ensemble de chemins pertinents qui soit de taille raisonnable. Pour cela, nous avons (i) caractérisé une classe de chemins particuliers : les chemins *Rép-équivalents* et (ii) défini des critères de sélection prenant en compte les préférences de l'utilisateur. Le point (ii) fait l'objet de la section suivante. Le point (i) nous amène à définir la notion de chemins banques_entités_complets donnant des réponses équivalentes. On notera $\mathbf{chem_1 \approx_{RE} chem_2}$ l'assertion $chem_1$ est *Rép-équivalent* à $chem_2$ dont on donne une définition récursive :

- si $chem_1=chem_2$ alors $chem_1 \approx_{RE} chem_2$;
- si $chem_1$ et $chem_2$ sont tels que tous leurs couples concernent la même banque et le même ensemble d'entités alors $chem_1 \approx_{RE} chem_2$;
- si $chem_1$=Concat_Chem(ch_1,ch'_1) et $chem_2$=Concat_Chem(ch_2,ch'_2) et $ch_1 \approx_{RE} ch_2$ et $ch'_1 \approx_{RE} ch'_2$ alors $chem_1 \approx_{RE} chem_2$;
- il n'y a pas d'autre manière d'obtenir des chemins banques_entités_complets *Rép-équivalents.*

Par exemple, 2=[(*ENS*,b), (*ENS*,g)] et 10=[(*ENS*,g), (*ENS*,b)] sont *Rép-équivalents*.
Notre algorithme est alors modifié pour détecter de tels chemins et pour les renvoyer de façon groupée à l'utilisateur. Remarquons que ces chemins qui proposeront des réponses identiques ne sont pas équivalents en terme de traitement effectué. Dans un chemin de la forme [(b_1,e_1), (b_1,e_2)] on accède d'abord à la vue de b_1 sur e_1 puis à la vue de b_1 sur e_2 (cf. l'ordre des opérateurs d'une requête en algèbre relationnelle).

5. Utilisation de critères de préférences

Nous avons vu en section 2 que les banques biomédicales peuvent avoir différents *focus* et peuvent être considérées comme plus ou moins *validées*. Nous permettons à chaque utilisateur de paramétrer le *niveau de validation* qu'il associe aux différentes banques. De même, nous avons vu que les liens qui sont proposés entre les banques peuvent eux aussi être plus ou moins *sûrs*. On va exploiter ces informations sur les banques et sur les liens pour filtrer les chemins à renvoyer et classer l'ensemble des réponses obtenues. On considère les 4 critères indiqués dans TAB 1. Pour chacun d'eux, on explicite leur utilisation pour sélectionner des chemins résultats et classer les chemins ainsi filtrés. Le *score de validité d'un chemin* est défini comme la somme de tous les niveaux des banques du chemin.

Critère	Seuil	Tri des chemins
Taille	Pas de chemins de taille supérieure à *seuil_taille_chem*	Par taille croissante
Focus	Pas de chemins ayant plus de *seuil_focus* banques consultées pour une autre entité que leur focus.	Par nombre décroissant de banques consultées pour une entité e qui est son focus.

BanquesValidées	Pas de chemins ayant plus de *seuil_bqe_niv_val(ni)* de banques consultées de niveau ni.	Par *score de validité* décroissant.
LiensSurs	Pas de chemins ayant plus de *seuil_lien_non_sur* liens non sûrs.	Par nombre croissant de liens non sûrs.

TAB 1 – Table des critères de sélection et de tri

Nous permettons maintenant à l'utilisateur de caractériser les chemins qu'il souhaite récupérer en sortie. Pour ce faire, il rentre les valeurs des seuils pour les critères qu'il souhaite prendre en compte parmi les 4 indiqués dans TAB 1. Ces critères sont pris en compte dans l'algorithme de la section précédente par l'introduction de fonctions booléennes contrôlant le non dépassement des seuils sur les critères. Les fonctions (1) *Contrôle_Init_Crit*, (2) *Contrôle_Crit* et (3) C*ontrôle_Crit_Chem* ont pour buts respectifs de tester, en fonction des seuils fixés, la possibilité de construire (1) un nouveau couple (b,e), (2) un nouveau chemin centré sur une entité e et potentiellement issu de la concaténation de deux chemins centrés sur l'entité e et (3) un nouveau chemin potentiellement issu de la concaténation de deux sous-chemins centrés sur des entités différentes. Par défaut, aucun critère n'est pris en compte (valeur MAX). D'autre part, dans cette seconde version de l'algorithme, les chemins sont renvoyés à l'utilisateur triés en fonction des valeurs qu'ils ont sur les différents critères. L'utilisateur pourra indiquer un ordre lexicographique pour ordonner la prise en compte de ces critères. On pourra alors envisager l'appel d'une fonction de tri qui ordonne les chemins résultats en fonction des indications de l'utilisateur.

6. Résultats sur l'exemple

Etudions les informations récoltées par différents chemins générés par *Chem_Banque_Entite_Complet*. Dans un premier temps nous ne considérons aucun critère de sélection ni de tri pour la requête « *Retrouver des informations sur le BAC dont le numéro est **rp11-703h8** ainsi que sur les gènes que ce BAC peut contenir* ». Un BAC est un morceau de chromosome composé d'une séquence de nucléotides et est identifié de façon unique à travers les sources par son numéro. Les réponses obtenues pour la requête correspondent à l'état des sources consultées le 1er Octobre 2003. Nous illustrons ci-dessous l'intérêt majeur pour le biologiste de pouvoir explorer différents chemins en soulignant deux points.

(i) Le chemin 2=[(*ENS*,b), (*ENS*,g)] propose dans ensEMBL des informations de localisation du BAC *rp11-703h8,* sur le Chromosome **11,** et des informations succinctes à propos de la séquence du BAC qui peut contenir des gènes. Ces renseignements peuvent être en soi suffisants ; néanmoins, si l'on cherche à obtenir des informations plus précises sur les séquences potentielles des gènes contenus sur ce BAC, on pourra suivre le chemin 3=[(*ENS*,b), (*ENS*,g), (*GB*,g)] qui propose en outre un lien vers GenBank (id AP003306) renseignant ces informations. Dans cet exemple, le chemin 2 propose d'accéder à une seule banque (*ENS*) et le chemin 3 à deux banques (*ENS* et *GB*). Considérons un autre exemple avec des chemins qui proposent respectivement d'accéder à deux et à trois banques. C'est le cas des chemins 4=[(*LL*,b), (*UG*,g)] et 5=[(*LL*,b), (*UG*,g), (*RS*,g)]. Le chemin 4 propose dans LocusLink une localisation du BAC *rp11-703h8* sur le chromosome **11** et donne une localisation cytogénétique précise de celui-ci en **11q12.2**. En suivant le lien vers UniGene (id Hs 448198), un gène semble être positionné sur le BAC. Ce gène coderait pour une protéine

dont la fonction est « transmembrane helix receptor »[2]. Là encore, ces informations peuvent être suffisantes, mais pour plus de détail, on peut suivre le chemin 5=[(*LL*,b), (*UG*,g), (*RS*,g)] qui propose en outre un lien de UniGene vers RefSeq (id XM_16900), source offrant des données plus précises sur la séquence dite *codante*.

Conclusion : un chemin de la forme [(b_1,e_1), (b_2,e_2), (b_3,e_2)] (où b1 est éventuellement égal à b2) peut donner des informations plus complètes que l'un de ses sous-chemins, ici [(b_1,e_1), (b_2,e_2)], bien que toutes les entités sous-jacentes à la requête soient présentes dans le sous-chemin.

(ii) Le chemin 7=[(*UCSC*,b), (*RS*,g)] propose dans UCSCGenome une localisation du BAC *rp11-703h8* sur le chromosome **4** au locus q12.

Conclusion : deux chemins, par exemple 4 et 7, alors qu'ils concernent un même objet (le BAC *rp11-703h8)* peuvent renvoyer des informations divergentes (localisation du BAC sur le chromosome 4 ou 11).

L'utilisation de critères pour filtrer et classer les chemins permet notamment d'aider l'utilisateur à trancher lorsqu'il est face à des informations divergentes. On va prendre en compte maintenant les informations indiquées sur la FIG. 2 concernant le *focus* et le niveau de *validité* des banques ainsi que les liens non *sûrs*. Supposons que l'utilisateur choisisse des seuils sur les différents critères qui imposent que les chemins renvoyés par l'algorithme soient (a) de taille maximale 2 (*seuil_taille_chem=2*), (b) tels que chaque banque est consultée pour des informations sur une entité seulement si cette entité est le focus de la banque (*seuil_focus=MAX*), (c) tels que les banques composant le chemin aient toutes un niveau au moins 3 (*seuil_bqe_niv_val(1)=0,seuil_bqe_niv_val(2)=0,* et pour n allant de 3 à 10 : *seuil_bqe_niv_val(n)=MAX*) , (d) tels qu'ils ne traversent qu'au plus un lien non sûr (*seuil_lien_non_sur=1*). Supposons de plus que l'utilisateur souhaite classer les chemins résultats ainsi sélectionnés avec l'ordre suivant sur les critères : (1) *Taille,* (2) *BanquesValidées,* (3) *LiensSurs*. Notons que la condition (b) implique un élagage selon le *focus* et n'intervient donc pas dans le classement.

Conséquences :

La liste L' des chemins renvoyés est L' = [7, 4, 9, 11, 5] avec
7=[(*UCSC*,b),(*RS*,g)], 4=[(*LL*,b),(*UG*,g)], 9=[(*RS*,g),(*LL*,b)],
11=[(*UG*,g),(*RS*,g), (*LL*,b)] et 5=[(*LL*,b),(*UG*,g), (*RS*,g)].

Le chemin 6 a une taille égale à 3 et n'est donc plus dans l'ensemble des réponses (cf. (a)). Les chemins 2, 3 et 10 sont eux aussi éliminés car ils proposent d'accéder à ensEMBL pour des informations sur l'entité gène alors que son focus est BAC (cf. (b)). Les chemins 1 et 8 ne sont plus non plus solution car ils proposent d'accéder à la banque GenBank qui est de niveau 2 (cf. (c)). Le classement des chemins de L' se fait par taille croissante, on a d'abord les chemins 7, 4, et 9 de taille 1 et puis les chemins 11 et 5 de taille 2. Les chemins 11 et 5 ont même valeur pour les tous les critères, leur ordre est donc indifférent. Par contre, le chemin 7 a un score de validité plus élevé que le chemin 4 en conséquence 7 précède 4 dans la liste des réponses. Les chemins 4 et 9 ont un ordre indifférent. L'utilisateur pourra donc *a priori* considérer que le BAC *rp11-703h8* est localisé sur le chromosome 4 (cf. chemin 7) et non pas sur le chromosome 11 (cf. chemin 4).

Le classement proposé à l'utilisateur tient compte de ses préférences et lui permet de choisir en cas d'informations divergentes.

[2] Cette information est cruciale car elle permet au biologiste de savoir que le BAC n'est pas une région *intergénique* mais contient bien un gène codant pour une protéine.

7. Conclusion et perspectives

L'algorithme *Chem_Banque_Entite_Complet* de construction de chemins que nous avons introduit dans cet article permet à un biologiste de ne pas avoir à sélectionner ni même à connaître *a priori* les sources susceptibles répondre à sa requête (transparence de l'interrogation). De plus, chaque chemin renvoyé par l'algorithme indique à l'utilisateur où récupérer et comment combiner les données des différentes banques répondant à sa requête (traçabilité des réponses).

Notons que *Chem_Banque_Entite_Complet* peut être utilisé dans différents contextes. Dans la plate-forme HKIS qui offre à ses utilisateurs des parseurs pour mettre les sources au format relationnel, il peut être considéré comme un module de construction de chemins permettant de savoir quelles sources rapatrier et dans quel ordre effectuer les jointures entre les données de ces sources ; il est d'ailleurs actuellement en cours d'implémentation. Mais cet algorithme de sélection de chemins et les critères de préférences que nous avons définis pourraient aussi être utilisés en amont d'autres systèmes fondés sur le format relationnel, comme DiscoveryLink [Haas *et al.,* 2001]. De façon encore plus générale, il existe de nombreux systèmes permettant l'accès à plusieurs sources biomédicales et l'uniformisation des formats des sources comme par exemple DBGet/LinkDB [Fujibuchi *et al.*, 1998] ou l'un des plus utilisés aujourd'hui, SRS[3] [Etzold *et al.*, 1996]. SRS recherche l'ensemble des fiches d'annotations des sources sélectionnées par l'utilisateur qui contiennent un texte entré par ce dernier. La liste des fiches répondant à sa requête lui est renvoyée et il peut ainsi accéder au contenu de chacune d'elles et suivre des liens de références croisées entre fiches. SRS se distingue de notre approche par le fait que l'utilisateur a directement accès à l'ensemble des informations contenues dans les banques sans voir le chemin qu'il est en train de suivre ou les autres chemins qu'il pourrait emprunter. Le principe de notre algorithme est assez général pour pouvoir s'adapter aussi à ce type de système et peut être proposé en amont en tant que module d'aide à l'utilisateur.

La démarche suivie dans [Mork *et al.*, 2002] qui décrit le médiateur *BioMed*, est plus proche de la nôtre. La notion de chemin entre les sources y est explicite et le langage PQL, basé sur XML, permet de définir des propriétés sur les chemins (par exemple, des chemins *sûrs*). Néanmoins, aucun algorithme de sélection de chemins n'est présenté et là encore, l'utilisateur récupère directement l'ensemble des résultats instanciés sans connaître la liste des chemins permettant de les obtenir.

Notre étude peut être mise en relation plus globalement avec les travaux sur les méta-données. Ceux qui concernent les données biologiques sont encore peu nombreux et portent le plus souvent, comme dans [Cheung *et al.* 1998] ou dans le *Medical Metadata Project*[4], plus sur la description des sources (URI, nom, …) ou sur la description du contenu des sources (entités biologiques…) que sur la qualité des données. Il est vrai qu'il existe de plus en plus de travaux sur la qualité des données comme par exemple [Lenca *et al.*, 2003], mais, à notre connaissance, peu d'entre eux aujourd'hui prennent en compte la spécificité des données biomédicales. Un des problèmes propres aux données biomédicales est qu'elles peuvent être divergentes sans que l'on puisse, *a priori*, décider quelles sont les données correctes ou pas. Par exemple, la notion de correction des données évoquée dans [Berti-

[3] Sequence Retrieval System, http://srs6.ebi.ac.uk/

[4] http://medir.ohsu.edu/~maletg/MedMetadata.HTM

Equille, 2003] ne s'applique pas directement à de telles données et il serait intéressant d'étudier une notion de qualité plus spécifique aux données biomédicales [Berti, 2001].

Remerciements

Les auteurs remercient tout particulièrement Emmanuel Barillot, François Radvanyi et Nicolas Stransky (Institut Curie), Philippe Boutruche, Stéphane Graziani et Hervé Perdrix (ISoft), ainsi que l'ensemble des partenaires du projet HKIS, pour les discussions fructueuses qu'ils ont eues avec eux. Ce travail a été partiellement financé par le projet européen HKIS (5e PCRD, Projet R&D, IST-2001-38153).

Références

[Berti, 2001] L. Berti. Integration of Biological Data and Quality-driven Source Negotiation. Proc. of the *Int. Conf. on Conceptual Modeling (ER'2001)*, LNCS 2224:256-269, 2001.

[Berti-Equille, 2003] L. Berti-Equille. Renseigner la qualité des connaissances par la fusion d'indicateurs sur la qualité des données. Actes *EGC' 2003* : 263-269, 2003.

[Cheung *et al.*, 1998] K. Cheung, P.M. Nadkarni et D. Shin. A metadata approach to query interoperation between molecular biology databases. *BioInformatics* 14(6):486-497, 1998

[Etzold *et al.*, 1996] T. Etzold, A. Ulyanov, et P. Argos. SRS: information retrieval system for molecular biology data banks. *Methods Enzymol*, 266: 114-128, 1996.

[Fujibuchi *et al.*, 1998] W. Fujibuchi, S. Goto, H. Migimatsu, I. Uchiyama, A. Ogiwara, Y. Akiyama, et M. Kanehisa. DBGET/LinkDB : an integrated database retrieval system. Proc. of the *Pacific Symp. Biocomputing* : 683-694, 1998.

[Haas *et al.*, 2001] L. M. Haas, P. M. Schwarz, P. Kodali, E. Kotlar, J. E. Rice, et W.C. Swope. DiscoveryLink: A system for integrated access to life sciences data source. *IBM Systems Journal*, 40(2): 263-269, 2001.

[Lenca *et al.*, 2003] Ph. Lenca, P. Meyer, Ph. Picouet et B. Vaillant. Aide multi-critère à la décision pour évaluer les indices de qualité des connaissances. Actes *EGC'2003*:263-269.

[Mork *et al.* 2002] P. Mork, A. Shaker, A. Halevy et P. Tarczy-Hornoch. PQL: A Declarative Query Language over Dynamic Biological Schemata. *AMIA Annual Symposium,* 2002.

[Veale *et al.*, 1987] D. Veale, T. Ashcroft, C. Marsh, G.J. Gibson, A.L. Harris. Epidermal growth factor receptors in non-small cell lung cancer. *Br J Cancer*, 55(51987):513-516, 1987.

Annexe 1

Procédure Chem_Centrés_Entité
Entrées : e : Une entité sous-jacente à la requête R ; B : Ensemble de banques contenant e
Sortie : CHEM_CENTRE_E : Ensemble des chemins de B centré-entité sur e
Début

```
L ← ∅ // Initialisation de l'ensemble L des chemins centré-entité relatifs à e qui restent à compléter
Pour toute banque β ∈ B faire
    Si Contrôle_Init_ Crit (β, e) alors // Si la banque et l'entité valident les critères
        L ← L ∪ { [(β, e)] }      // [(β, e)] est le début d'un chemin de L
```

```
Tant que L ≠ ∅ faire // c : premier des chemins de L en cours de traitement ; b : dernière banque de c
  Pour toute banque β ∈ B tq Cross_Refer (b, β) faire // la banque β pourrait étendre c
    Si non Contient_Banq(c, β) alors    // La propriété (3) doit être vérifiée
      Si Contrôle_ Crit (c, β, e) alors // Si le chemin c étendu par [(β, e)] valide les critères
        L ← L ∪ Concat_Chem (c, [(β, e)]) // on l'ajoute à L
  // Toutes les possibilités d'extension du chemin c ont été traitées, on l'ajoute donc à
  // CHEM_CENTRE_E et on le supprime des chemins à compléter
  L ← L \ {c} ; CHEM_CENTRE_E ← CHEM_CENTRE_E ∪ {c}
```

Fin Chem_Centrés_Entité

Procédure Construction_Rec

Entrées : E_PART: Ensemble d'entités sous-jacentes à la requête R
Tab_CHEM_CENTRE : Tableau des ensembles de chemins centré-entité, indicé par entité // rempli par la procédure Chem_Centré_Entité

Sortie : CHEM_PART : Ensemble des chemins banques_entités_complets passant par toutes les entités de E_PART

Début

```
CHEM_PART ← ∅      // Initialisation
Si |E_PART|= 1 alors // Cas d'arrêt : les chemins résultats sont centré-entité sur l'unique entité.
  Pour toute entité e de E_PART faire CHEM_PART ← Tab_CHEM_CENTRE (e)
Sinon Pour toute entité e de E_PART faire
        CHEM_SOUS_PART ← ∅
        // CHEM_SOUS_PART est un ensemble de banques_entités initialisé à ∅. Ensuite on fait
        // un appel récursif :on stocke l'ensemble de tous les chemins banques_entités passant par
        // toutes les entités de E_PART sauf e dans CHEM_SOUS_PART.
        Construction_Rec(E_PART\{e},Tab_CHEM_CENTRE,CHEM_SOUS_PART)
        // Pour chaque chemin c1 centré-entité sur e, on regarde s'il peut être complété par un chemin c2
        // de CHEM_SOUS_PART. Si c'est le cas, le chemin résultant de la concaténation passe bien par
        // toutes les entités de E_PART et est ajouté au résultat CHEM_PART de la procédure.
        Pour tout c1 de Tab_CHEM_CENTRE (e) faire // Soit b1 la dernière banque de c1
          Pour tout c2 de CHEM_SOUS_PART faire // Soit b2 la première banque de c2
            Si ((Cross_Refer (b1,b2) ou b1=b2) et Contrôle_Crit_Chem(c1,c2)) alors
            CHEM_PART ← CHEM_PART ∪ Concat_Chem (c1, c2)
```

Fin Construction_Rec

Summary

Our work is performed in the context of a project aiming at the elaboration of a platform for the genomic study of cancer. This platform - among other features - provides the user with several analysis scenarios. At each step of a scenario chosen by the user for his study, he can be constrained to ask queries requiring access to many sources and be faced with the task of selecting relevant sources. Our final goal is to help him with an algorithm which presents a selection of sources which fits his own preferences and is relevant to his query. For it, we investigate the characteristics of biomedical data and introduce several preference criteria useful for bioinformaticians. Our approach is illustrated through an elementary biomedical query.