Développement d'un outil de recherche automatisée des éléments génétiques mobiles de type ICE dans les génomes bactériens (ICE-Finder)

Yann Aubert

Aubert Yann

Master Microbiologie : Option Microbiologie Environnementale et Sanitaire

07/01/2013 21/06/2013

Développement d'un outil de recherche automatisée des éléments génétiques mobiles de type ICE dans les génomes bactériens (ICE-Finder)

Encadrement

Devignes Marie-Dominique LORIA UMR7503, équipe Orpailleur

G. Guédon et N. Leblond-Bourget UMR1128 DynAMic, équipe ICE-TeA,

Sommaire

I) Introduction

1) Problématique et contexte du stage

L'acquisition de gènes par transfert horizontal est un mécanisme clé d'évolution des génomes bactériens. Il implique le transfert d'ADN d'une bactérie à une autre, ces bactéries peuvent être de la même espèce ou d'espèces différentes, et son maintien dans la descendance de celle-ci. Il permet l'acquisition rapide de nouvelles fonctions d'adaptation pouvant être avantageuses pour la bactérie telles que la virulence, la résistance aux antibiotiques ou la synthèse de nouveaux métabolites (Ochman *et al*., 2000 ; Gogarten *et al*., 2005).

L'acquisition de gènes est généralement due à des éléments génétiques mobiles dont certains sont capables de se transférer d'une cellule bactérienne à une autre. Un mécanisme essentiel d'acquisition de gènes d'adaptation est la conjugaison, qui nécessite un contact direct entre deux cellules bactériennes. Un fragment d'ADN est transféré d'une cellule dite « donneuse » vers une autre cellule dite « receveuse » via un pore de conjugaison protéique, codé par un élément génétique mobile (appelé de ce fait élément conjugatif).

Bien que la quasi-totalité des éléments caractérisés capables de se transférer par conjugaison soient des plasmides, c'est-à-dire des éléments extrachromosomiques se maintenant par réplication dans la descendance, de nouvelles classes d'éléments génétiques intégrés dans les chromosomes bactériens et se transférant par conjugaison existent (Frost *et al*., 2005)(Guglielmini *et al*., 2011). Les mieux connus d'entre-eux, les ICE ('Integrative Conjugative Element') sont des éléments génétiques mobiles qui codent les fonctions nécessaires à leur excision sous forme circulaire, leur transfert via un pore de conjugaison, et leur intégration dans un nouveau génome qui assure ainsi leur maintien dans la descendance (Burrus *et al*., 2002). Leur transfert s'accompagne de leur réplication, celle-ci pouvant également participer à leur maintien, en particulier quand ils sont sous forme excisée. Quelques ICE sont connus et bien étudiés, comme Tn*916* (Cooksone*t al*., 2011) ou ICE*Bs1* (Burrus *et al.,* 2001). Une autre classe d'éléments se transférant par conjugaison concerne les IME ('Integrative Mobilizable Element'). A la différence des ICE, ces éléments ne codent aucune protéine du pore de conjugaison mais, sont capables d'utiliser le pore de conjugaison codé par un ICE (Burrus *et al*., 2002). Des IME n'ont été que très rarement identifiés. Les gènes acquis par transfert horizontal sont souvent regroupés au niveau de régions chromosomiques appelées îlots génomiques, dont le mécanisme d'acquisition a été très rarement identifié. Les données récentes suggèrent que beaucoup de ces îlots pourraient correspondre à des ICE ou IME ou en dériver, et qu'en conséquence ICE et IME pourraient être extrêmement répandus (Gugliemini *et al*., 2011, Haenni *et al*., 2010).

Ces dernières années, le nombre de génomes bactériens complets analysés s'est fortement accru. Tout d'abord, les nouvelles technologies de séquençage permettent le séquençage d'un génome de quelques mégabases en très peu de temps. De plus, l'augmentation de la puissance de calcul des processeurs et l'amélioration des algorithmes permet de détecter efficacement la présence de gènes. Enfin, l'augmentation des capacités de stockage permet de sauvegarder toutes ces informations dans des espaces de plus en plus petits. Cependant de nombreux problèmes demeurent. Si nous sommes capables de séquencer et d'identifier les régions codant des protéines, il n'est pas encore possible d'identifier la fonction de toutes les protéines. Il est encore moins possible de détecter, identifier et délimiter de manière claire tous les éléments génétiques mobiles, particulièrement ceux appartenant à des classes encore mal ou très mal connues comme les ICE et les IME.

Ce stage a été effectué dans le cadre d'un projet collaboratif entre le LORIA et le laboratoire DynAMic visant à créer une méthode informatisée et automatisée de détection des ICE et IME dans les génomes bactériens. Il a pour buts : (i) d'établir une procédure d'identification et de délimitation des ICE et IME dans ces génomes, en s'appuyant sur une base de données permettant de regrouper les informations collectées et de faciliter des apprentissages ultérieurs, (ii) d'automatiser cette procédure afin de créer, à terme, une plate-forme de recherche d'ICE accessible à la communauté scientifique.

Afin de mettre en place la procédure d'identification des ICE/IME et de lever les verrous méthodologiques, l'analyse a été restreinte aux génomes des bactéries du genre *Streptococcus*. Ce genre comprend à la fois des bactéries alimentaires « utiles » (*S. thermophilus*, ferment lactique alimentaire), des bactéries commensales pouvant être des pathogènes opportunistes (*S. salivarius, S. agalactiae*) et des pathogènes humains (*S. pneumoniae*, *S. pyogenes*, *S. mutans)* ou animaux (*S. agalactiae*). Ce genre a été choisi sur la base de résultats antérieurs, indiquant que les ICE et IME sont répandus au sein des génomes de souches des espèces *S. thermophilus* (Pavlovic *et al.*, 2004) et *S. agalactiae* (Brochet *et al*., 2008). Ainsi, ces éléments pourraient donc jouer un rôle essentiel dans les échanges de gènes au sein du genre *Streptococcus* et contribuer de façon importante à l'adaptation des streptocoques à leur environnement.

2) Caractéristiques biologiques et génétiques des ICE/IME

2.1 Une organisation en modules

Les ICE et les IME connus de streptocoques ont des tailles très variables, allant de 5 kb à 300 kb. Les plus petits ICE mesurent 18 kb et en dessous de cette valeur, il est admis que l'élément trouvé est un IME. Tous les éléments mobiles, dont les ICE et les IME, présentent une structure modulaire (Figure 1)(Burrus *et al*.,2002 ; Toussaint et Merlin, 2002 , Pavlovic *et al*., 2004).

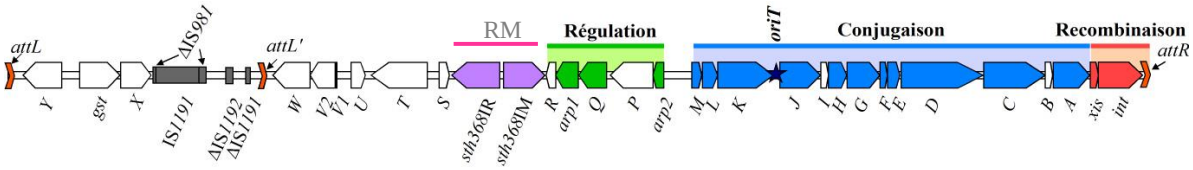

Figure 1 : Structure modulaire d'un ICE : l'exemple d'ICESt3 de *S. thermophilus*(d'après Bellanger *et al*., 2009).

RM, module codant un système de restriction-modification conférant une résistance à des bactériophages.

Ils se composent :

- (i) d'un module de recombinaison qui code une intégrase nécessaire à l'excision et intégration des ICE et IME. Les intégrases peuvent appartenir à 3 familles de protéines : les transposases à DDE, les intégrases à sérine et les intégrases à tyrosine.

- (ii) d'un module de conjugaison ou de mobilisation. Chez un ICE, ce module code l'ensemble des protéines nécessaires au transfert de l'ADN d'une cellule à l'autre ce qui inclut non seulement les protéines du pore de conjugaison mais aussi une enzyme, appelée relaxase, et une protéine de couplage. Chez les IME, ce module code toujours une relaxase, parfois une protéine de couplage et aucune protéine du pore de conjugaison.

- (iii) d'un module de régulation qui contrôle le processus d'excision et de conjugaison en réponse à des stimuli extérieurs.

  (iv) d'autres modules pouvant conférer un avantage adaptatif à la bactérie. Par exemple de nombreux ICE et IME portent des gènes de résistance à des antibiotiques (tétracycline, érythromycine…), à des métaux lourds (cadmium, plomb), des gènes de biosynthèse de bactériocine (telle la lactococine 972).

  La caractérisation d'ICE et d'IME permettrait donc d'identifier des gènes d'intérêt industriel transférables d'une souche bactérienne à une autre., ou des gènes pour des applications thérapeutiques.

2.2 Les différentes étapes du transfert

Les différentes étapes du transfert sont représentées en Figure 2. Dans la cellule donneuse, l'ICE se maintient, s'excise du chromosome de la cellule hôte, généralement par recombinaison site-spécifique (Burrus *et al*., 2002). Le mécanisme de transfert des ICE est mal connu mais présenterait des similitudes avec celui des plasmides conjugatifs des bactéries à coloration Gram négatives. Chez les Firmicutes (groupe de bactéries à coloration Gram positive incluant les streptocoques), l'ADN serait d'abord pris en charge par des protéines qui s'assemblent au niveau d'une séquence conservée appelée origine de transfert (*oriT*) pour former un complexe nommé relaxosome. L'acteur majeur du relaxosome est la relaxase qui coupe l'un des brins dans une position spécifique d'*oriT* et se lie de façon covalente à l'extrémité 5' du brin d'ADN coupé. Le complexe ADN-relaxase va être ensuite recruté par une protéine appelée protéine de couplage avant d'être transporté par la machinerie de conjugaison (ou transférosome) (Llosa *et al*., 2002 ; Walldene*t al*., 2010 ; Burton et Dubnau, 2010).

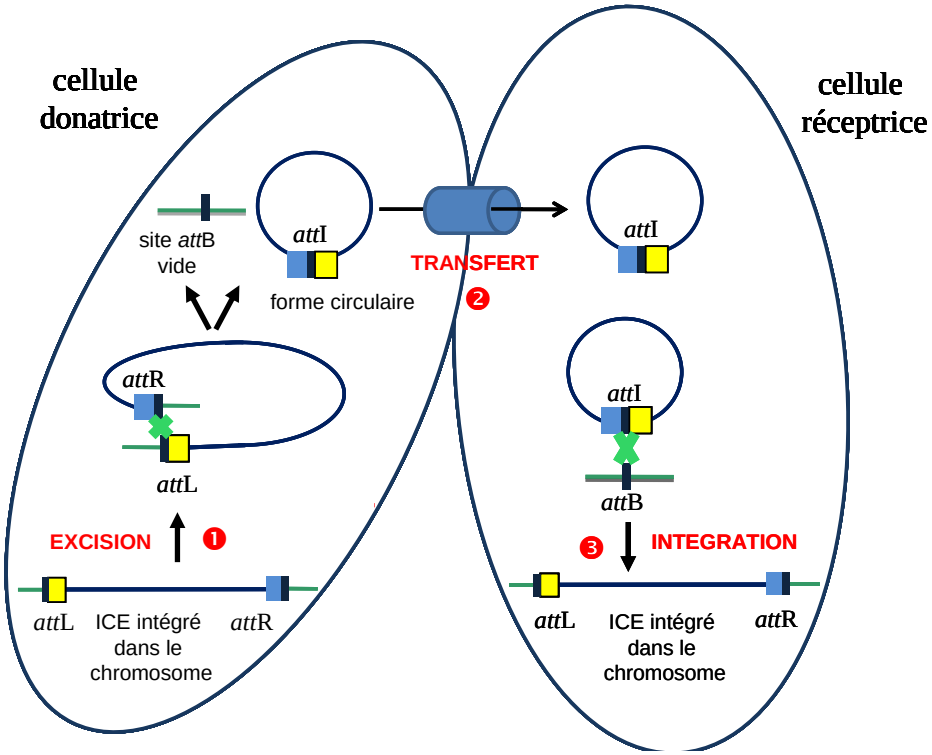

Figure 2 : Excision et transfert d'un élément intégratif conjugatif (ICE).

Chez les Firmicutes, la machinerie de conjugaison est mal connue. Plusieurs des protéines de ce pore sont cependant homologues aux protéines du pore de conjugaison des bactéries Gram négatives chez qui elle constitue un assemblage protéique complexe appelé système de sécrétion de type IV (T4SS pour « Type 4 Secretion System »). Le système le mieux connu est celui du plasmide Ti (pTi) d'*Agrobacterium tumefaciens* qui comprend 11 protéines (numérotées VirB1 à VirB11) en plus de la protéine de couplage (appelée VirD4) (Figure 3).

2.3 Les protéines caractéristiques des ICE/IME

Les données de la littérature indiquent que les ICE et les IME évoluent par échange de module. De ce fait, aucun des modules pris isolement n'est caractéristique de ces éléments, seule la combinaison de modules permet d'identifier les ICE/IME (Burrus *et al*., 2002). Un module de recombinaison pourra être identifié par la présence d'un gène codant une intégrase. Un module de conjugaison ou mobilisation pourra être repéré par la présence de gènes codant une relaxase et éventuellement d'une protéine de couplage. Ainsi, l'association des gènes codant ces 3 protéines (intégrase, relaxase et éventuellement protéine de couplage) est caractéristique d'un ICE ou d'un IME, c'est pourquoi ces classes de protéines ont été recherchées au sein des 72 génomes du genre *Streptococcus* et les hits positionnés dans les génomes.

2.4 Les sites d'intégration

Une très grande majorité des ICE et IME connus code des intégrases à tyrosine. Ces intégrases catalysent généralement une recombinaison site-spécifique entre deux petites séquences identiques appartenant à un site spécifique *attB* du chromosome bactérien et un site spécifique *attI* de l'élément. Le site *attB* correspond généralement à l'extrémité 3' d'un gène codant un ARNt ou à celle d'un gène codant une protéine le plus souvent ribosomique. D'autres ICE/IME éléments peuvent s'intégrer dans un site spécifique correspondant à l'extrémité d'un gène codant une protéine. L'intégration site-spécifique d'un ICE/IME codant une intégrase à tyrosine conduit à la formation de deux sites *attL* et *attR* flanquant l'élément intégré. Ces sites *att* portent des répétitions directes de 8 à 60 pb. Bien que codant une intégrase à tyrosine, les ICE de type Tn*916* sont les rares ICE qui s'intègrent de façon peu ou pas spécifique et ne génèrent pas la formation de tels DR lors de leur intégration.

Une minorité d'ICE et IME connus codent une intégrase à sérine. Ces intégrases, moins bien connues que les intégrases à tyrosine, catalysent généralement une recombinaison site-spécifique entre un site spécifique *attB* du chromosome bactérien et un site spécifique *attI* de l'élément. Le site d'intégration peut être interne à un gène ou correspondre à son extrémité 3'. Les DR flanquant peuvent n'être que de 2 pb. Dans certains cas, l'intégration n'est pas site-spécifique.

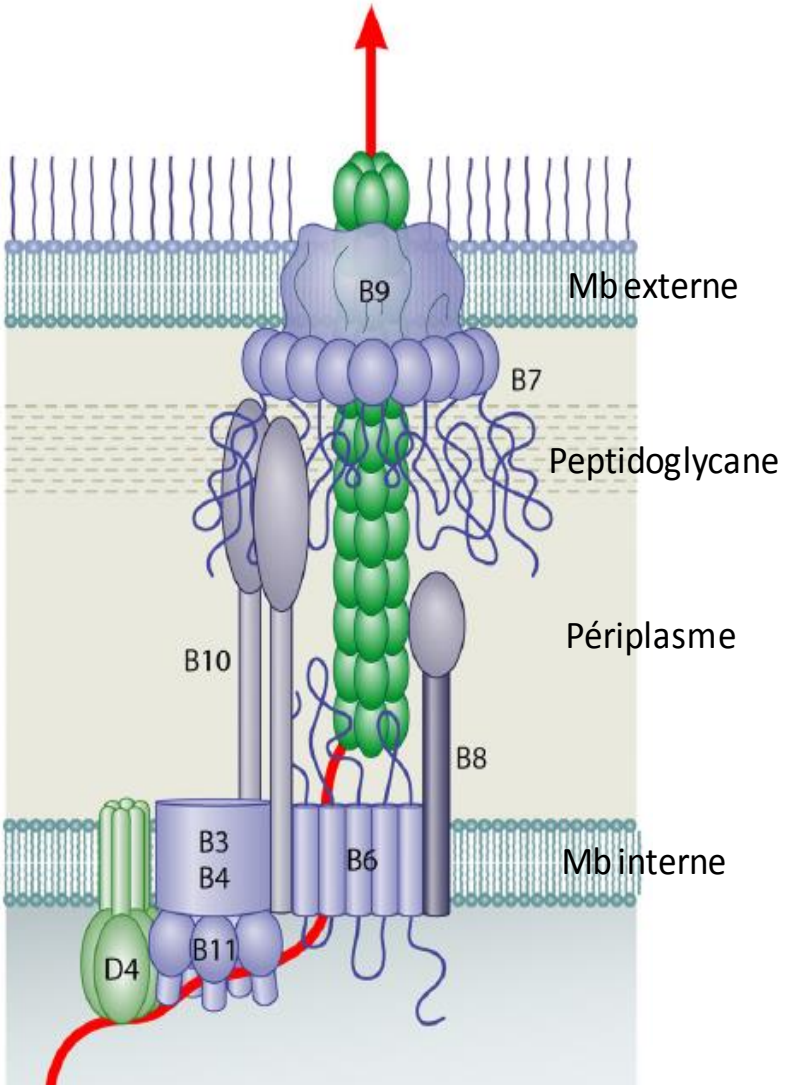

Figure 3 : Schéma d'un pore de conjugaison : le système VirB4/D4 du plasmide Ti (Alvarez-Martinez etChristie, 2009).

Quelques ICE et IME connus codent une intégrase appartenant aux transposases à DDE. Les mieux connus des éléments de streptocoques sont les ICE Tn*GBS1* et Tn*GBS2* de streptocoques (Brochet *et al.,* 2009). Leur intégration se fait exclusivement 15-16 pb en amont de la séquence -35 de promoteurs variés. Elle provoque une duplication de la séquence cible, l'élément intégré étant flanqué de répétions directes de 9 pb. D'autres éléments présentent cependant une faible spécificité d'intégration et des tailles de DR différentes.

Cette variabilité rend difficile la détermination des limites des ICE/IME dans les génomes.

2.5. Les éléments composites

Les ICE portent fréquemment d'autres éléments mobiles. Le cas le plus fréquent chez les streptocoques concerne les éléments de la famille Tn916 intégrés en diverses positions d'ICE de la famille Tn5252 (Mingoia *et al*., 2011).

De plus, les éléments qui ont une intégrase à tyrosine et qui s'intègrent de façon site spécifique peuvent, dans au moins certains cas, s'intégrer non seulement au site attB mais également attL ou attR flanquant un ICE résident et ainsi générer un élément composite (Figure 4).

3). Approches bio-informatiques pour l'identification et la localisation des ICE

3.1 Recherche d'îlots génomiques

Les ICE/IME font partie, dans les génomes, de régions appelées îlots génomiques (IG), dont les caractéristiques diffèrent du reste du génome, du fait du transfert horizontal. Diverses équipes ont développé des méthodes de recherche d'îlots génomiques, mais n'ont pas tenté d'identifier leur nature.

3.1.1Une approche par alignement de séquences

L'équipe de Langille *et al* (2008) a travaillé sur une méthode de prédiction des IG. Le module de recherche de ces îlots compare un génome à une liste de génomes références. Le génome à tester est aligné avec un génome de référence grâce au logiciel « Mauve ». Les régions conservées sont considérées comme des régions n'appartenant pas aux IG. Les régions non apparentées sont filtrées pour valider la présence d'IG. Le problème de cette méthode est qu'elle ne s'applique pas à des génomes pour lesquels il n'y a pas de génomes de référence dans la banque de données.

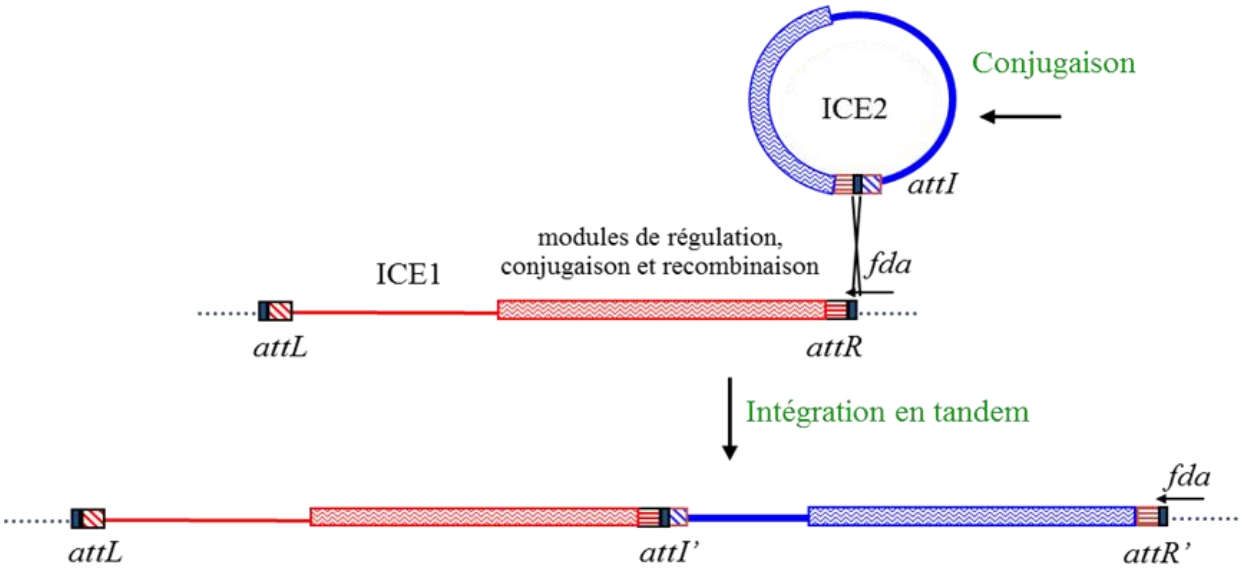

Figure 4 : Intégration en tandem de deux ICE. Dans ce schéma, l'ICE2 s'intègre dans le site attR de l'ICE1.

3.1.2 L'approche Islandviewer

Une autre approche, celle d'Islandviewer développée par l'équipe de Langille et Brinkman. (2009), a pour but de détecter et de visualiser des IG connus ou inconnus. Pour ce faire, plusieurs méthodes de détections (IslandPath-DIMOB (Hsiao*et al*., 2005), SIGIHMM (Waack *et al*., 2006)) ont été regroupées puis ils ont appliqué une méthode de recherche appelée IslandPick. Cette méthode permet la détection des îlots génomique par une analyse HMM (modèle de Markov caché) du génome via le logiciel SIGIHM. Cette analyse consiste à établir un modèle statistique de l'apparition de nucléotides dans un génome. Si une séquence s'éloigne trop de ce modèle statistique, alors elle sera considérée comme exogène. Bien qu'intéressante, cette méthode adaptée à la recherche des IG est inadaptée à la recherche d'ICE ou d'IME. En effet, l'utilisation de l'interface graphique développée par Languille et Brinkman (2009) pour rechercher des ICE connus, présents dans le génome de *Streptococcus agalactiae* A909 montre que les îlots génomiques connus et publiés sont bel et bien retrouvés mais les ICE et les IME n'apparaissent pas (soit parce qu'ils sont trop petits pour la recherche soit parce qu'ils ne présentent pas de différences significatives avec le modèle statistique).

3.1.3 L'approche GI-POP

L'équipe de Lee *et al* (2013) a créé un server GI-POP http://gipop.life.nthu.edu.tw ('Genomic Island Pipeline Of Prediction') qui permettrait d'identifier de manière efficace les îlots génomiques. La méthode de détection décrite dans la publication est intéressante. L'équipe commence par assembler les contig et les scaffold d'un génome mais d'après eux « certaines séquences peuvent être oubliées ».

Plusieurs étapes sont réalisées pour identifier les IG :

1. la découpe du génome en fragments de même longueur. Le profil de chaque fragment est extrait et est soumis à une SVM « support vector machine » pour classer les informations. La recherche d'IG se fait en analysant le pourcentage en bases G+C, la composition en acides aminés. Cela correspond à la recherche d'éléments génétiques mobiles par comparaison de séquences à partir IG connus ou de séquences n'appartenant pas aux IG (données du travail de Langille *et al.,* 2008)
2. un filtre est appliqué sur les résultats obtenus. Ce filtre est basé sur la longueur des éléments étudiés.
3. la délimitation des bornes des IG en recherchant des séquences codant des gènes d'ARNt ou des DR.

Le programme prend des séquences du génome et les compare avec celles du reste du génome. Si la séquence étudiée présente beaucoup d'analogie avec le reste du chromosome, le logiciel considère la séquence comme appartenant à l'organisme. Par

contre, si la séquence étudiée présente beaucoup de différences alors le programme considère qu'un îlot génomique est probablement présent dans cette séquence. GI-POP recherche les éléments répétés et les séquences codant les gènes d'ARNt afin de prédire la présence d'un îlot. Le facteur limitant est que les auteurs partent d'une base de données définies, et recherchent des éléments apparentés à cette base. Ainsi, des IG qui n'ont pas de représentant au sein de la base de données ne sont pas détectés. Il ne nous a pas été possible de tester ce modèle de prédiction des IG car le site où se situe le server GI-POP n'existe plus ou ne fonctionne pas.

3.1.4 L'approche ICEberg

L'équipe de Bi (Bi *et al.,* 2012) a travaillé sur le projet ICEberg. L'équipe a créé une base de données qui regroupe les informations publiées concernant les ICE mais non les IME. Leur site est accessible et permet de rechercher dans les génomes des ICE ou IME référencés. Toutefois, certains des ICE de la base de données ICEberg ne présentent aucune des protéines étiquettes toujours codées par des ICE (intégrase, relaxase, protéines de couplage), ce qui met en doute le fait qu'il s'agisse bien d'ICE. Malgré cela, il sera intéressant d'utiliser le site ICEberg, pour confronter les résultats obtenus avec cette méthode et celle que nous développons. .

3.1.5 Une approche basée sur des protéines étiquettes

L'équipe de Guglielmini *et al*. (2011) a travaillé sur une méthode de détection des modules de conjugaison à l'aide d'une analyse HMM. Pour réaliser cette analyse l'équipe a créé une base de données contenant les informations de 3 types de protéines étiquettes de tous les éléments conjugatifs ou mobilisables connus (essentiellement des plasmides provenant principalement de protéobactéries) : la relaxase, la protéine de couplage, et la protéine du pore de conjugaison VirB4. Ce type d'approche permet de détecter uniquement les modules de conjugaison ayant des similarités avec ceux présents dans la banque de référence. Comme il n'y a pas d'enrichissement de cette banque, la détection de modules de conjugaison reste limitée. Ces auteurs indiquent que leur méthode est performante chez les protéobactéries du fait d'une bonne connaissance des systèmes conjugatifs de plasmides chez ces bactéries à coloration Gram négatives, mais probablement moins chez les autres groupes dont les systèmes de conjugaison sont moins bien connus. Les auteurs n'ont pas recherché les intégrases. Ils se contentent de suggérer que les modules de conjugaison chromosomiques détectés appartiennent probablement à des ICE et que les gènes isolés de relaxases pourraient appartenir à des IME.

En conclusion il existe plusieurs méthodes pour localiser et pour identifier des îlots génomiques, mais les IG peuvent correspondre à des prophages, des plasmides intégrés, des

transposons, ou d'autres éléments génétiques que les ICE/IME. De plus, certains ICE ou IME sont trop petits pour être détectés comme des îlots génomiques. Donc la détection des ilots génomique n'est pas une méthode spécifique pour rechercher et borner les ICE/IME. La seule méthode spécifique de recherche d'ICE est celle de Guglielmini *et al*. (2011). Cependant, il faut noter que leur recherche s'appuie sur un jeu de protéines de références incluant notamment la protéine VirB4 qui est absente chez les IME. Leur méthode ne permet donc pas de détecter les IME.

4) Objectif du stage

4.1 L'approche d'ICE Finder

La Figure 5 schématise le projet ICE-Finder. Ce projet est centré sur une base de données ICEdb qui intègre tous les résultats de la recherche dans les génomes étudiés. La partie supérieure (traitement 'KoriBlast') correspond à la recherche des protéines étiquettes parmi les protéines codées par les génomes étudiés. La partie inférieure (traitements 'ICE-limits' et 'ICE-products') correspond à la détection des bornes des ICE via la détection de séquences répétées, la visualisation des ICE/IME et l'extraction des protéines codées.

4.2 L'existant à mon arrivée

Une base de données ICEdb existait déjà à mon arrivée et avait été peuplée de manière incomplète. Une première procédure de recherche des protéines étiquettes avait été définie à partir du logiciel KoriBlast et d'un ensemble de protéines de référence. En bref, les résultats provenant de ce logiciel sont filtrés pour éliminer les résultats incohérents et redondants. Les résultats ainsi obtenus correspondent aux protéines candidates. Celles-ci doivent être validées par un expert, puis intégrées dans la base de données. La base de données est gérée par un système de gestion de bases de données relationnelle PostgreSQL. Ce type de gestion permet de préserver l'intégrité des données stockées dans la base et de faciliter l'accès à ces données.

4.3 Les objectifs du stage

4.3.1 Enrichissement et consolidation de la base de données.

Le premier objectif du stage a été de compléter ICEdb et de consolider les informations qu'elle contient (notamment la correction de certaines erreurs d'insertion et de modélisation) et de créer et valider des clés étrangères permettant de contrôler l'intégrité des données. A mon arrivée la base de données comportait des tables contenant les informations d'un premier blastP sur les protéines de couplage, la table génome et les tables références existaient déjà.

.

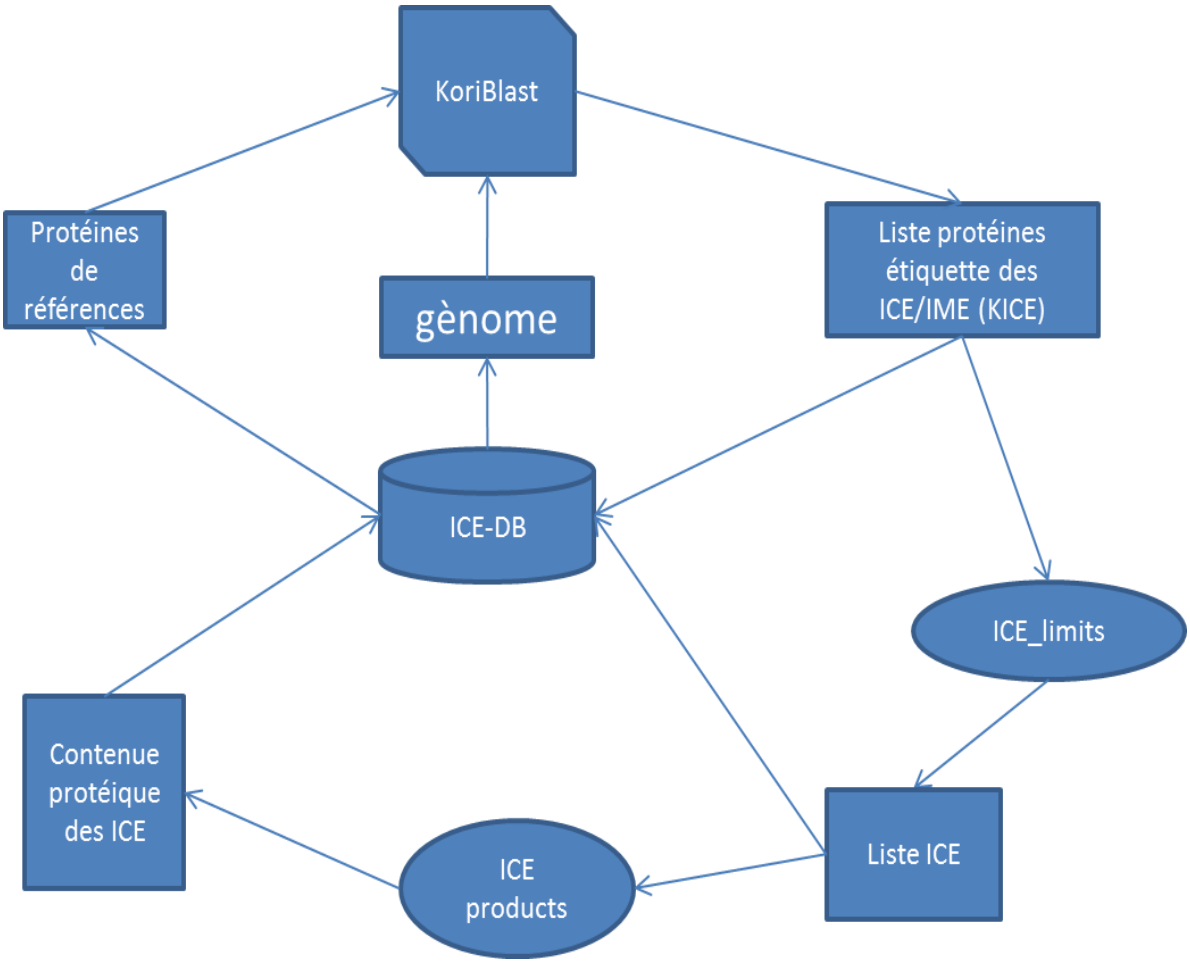

Figure 5 : Schéma du projet ICEfinder.

4.3.2 Détection des DR pour délimiter les ICE et visualisation

Le second objectif a consisté en la recherche des sites d'intégration des ICE (le plus souvent spécifiques) ainsi que leurs extrémités constituées de séquences répétées. Par exemple, les ICE de type ICE*St3* que nous avons caractérisé dans les génomes de *Streptococcus thermophilus* s'intègrent à l'extrémité 3' du gène *fda* (Burrus *et al.*, 2002). Ce travail permet de prédire les bornes des ICE et donc de les délimiter. Après validation, ces données devront être collectées pour enrichir la base de données.

Matériels et méthodes

1) Séquences

1.1 Protéines « étiquettes » de la Ref0

Un panel de protéines références (Ref0) a été fourni par l'équipe ICE TeA du laboratoire DynAMic. Il se compose de 40 protéines de couplage, 50 relaxases et 77 intégrases (51 intégrases à tyrosine, 13 intégrases à Serine, 13 transposases à DDE) (Tableau 1). Ces protéines sont codées exclusivement par des ICE et IME présents dans les génomes de Firmicutes et décrits dans la littérature. Seuls 14 des 72 génomes du genre *Streptococcus* codent ces protéines de référence, encore appelées protéines « étiquettes » : leur numéro d'accession est indiqué dans le Tableau 2.

1.2 Génomes

Les séquences des 72 chromosomes de streptocoques, disponibles dans les bases de données au moment de l'analyse, ont été utilisées afin de déterminer s'ils codent des homologues des protéines étiquettes Ref0. Les protéines codées par chacun de ces chromosomes sont téléchargés sous la forme d'un fichier-multifasta depuis le site du NCBI (ftp://ftp.ncbi.nih.gov/genomes/Bacteria/). Dans la suite du texte, le mot génome sera employé comme synonyme de chromosome.

2) Programmes externes utilisés

2.1 PostgreSQL(via l'interface pgAdminIII)

2.1.1 Généralités

ICE DB est une base de données créée pour la recherche d'ICE (Figure 6). Sur le modèle conceptuel, les différentes entités regroupent les informations relatives aux protéines (leur position, leur domaine…), les informations sur les génomes et les différentes expériences de BLASTp. La base de données est stockée sous le format postgreSQL.

Le logiciel utilisé pour pouvoir lire et modifier cette base de données est pgAdminIII.

Protéine de couplage		Relaxase		Intégrase					
				Transposase à DDE		Intégrase à Sérine		Intégrase à Tyrosine	
embl_id	genome_id	embl_id	genome_id	embl_id	genome_id	embl_id	genome_id	embl_id	genome_id
CAL97970	NC_009004	AAB06502	U50902	AAY32950	AY928180	AAB51419	U15027	AAA19427	BX571856
ABP89939	CP000407	AAB60013	BX571856	ACI48596	FJ231270	AAF35174	AF333235	AAA72427	L27649
ABV61247	NC_009848	AAC98434	L29324	ACS83558	FJ589781	AAO82010	NC_004668	AAC98428	L29324
ACI48608	FJ231270	AAF72355	AB374546	CAQ48507	AM990992	AAT72345	AF227521	AAF72368	AB374546
ADC91908	FJ231270	AAN00125	AE009948	CAQ48508	AM990992	BAA86648	AB033763	AAK17958	AF329848
CAR69098	NC_011900	AAN00848	AE009948	CAQ50273	AM990992	BAC67561	AY271717	AAO80040	NC_004668
CAZ51658	FM252031	AAN00876	AE009948	EDL64613	ABCF01000016	BAG06212	AY894416	AAO80333	NC_004668
CAZ55296	FM252032	AAN00972	AE009948	YP_001558636	NC_010001	CAJ67260	NC_009089	AAO81609	NC_004668
CAZ55792	FM252032	AAN57979	AE014133	YP_001642449	CP000904	CAJ68758	NC_009089	AAO81971	NC_004668
ADA65816	NC_013656	AAO80018	NC_004668	CAD46054	AL732656	CAJ70270	NC_009089	AAO81984	NC_004668
CAQ48513	AM990992	AAO81639	NC_004668	CAD46384	AL732656	CAP20376	AM909652	AAO82259	NC_004668
CAQ50424	AM990992	AAO82063	NC_004668	CAD46628	AL732656	YP_002744787	FM204884	AAX72413	CP000056
CAQ50279	AM990992	AAO82241	NC_004668	CAD46883	AL732656	YP_003028287	FM252032	ABV61272	NC_009848
CBG92846	FN555436	AAV61548	CP000023					ACW84384	EU351020
CBI14163	FN597254	AAV63465	CP000024					ADA65801	NC_013656
EFV96566	AEQQ01000099	AAX72431	CP000056					ADX23799	CP002215
ADX24160	CP002215	ABP89879	CP000407					ADX24144	CP002215
ADX23815	CP002215	ABV61248	NC_009848					BAA19318	NC_000964
CBW39305	FR671412	ACI48613	FJ231270					BAB56554	NC_002758
AEH56854	CP002843	ADA65815	NC_013656					CAB70622	AJ243106
AEH56878	CP002843	ADX23811	CP002215					CAC99175	NC_003210
BAA19323	NC_000964	ADX24159	CP002215					CAE52373	AJ586568
CAB12293	NC_000964	AEH56839	CP002843					CAJ67177	NC_009089
CAJ70243	NC_009089	AEH56852	CP002843					CAJ67935	NC_009089
CAJ70291	NC_009089	AEH56877	CP002843					CAJ70223	NC_009089
CAJ67204	NC_009089	BAA19324	NC_000964					CAL97951	NC_009004
AAB60012	BX571856	CAC67542	AJ278471					CAR69044	NC_011900
AAX72432	CP000056	CAC99189	NC_003210					CBG92867	FN555436
CAE52361	AJ586568	CAD46780	AL732656					CBW39317	FR671412
AAO82246	NC_004668	CAD46997	AL732656					EFF33467	ABQJ01000139
AAO82068	NC_004668	CAE52362	AJ586568					EFV96579	AEQQ01000099
AAO80014	NC_004668	CAJ67202	NC_009089					AAN00122	AE009948
AAN00877	AE009948	CAJ67248	NC_009089					AAN00846	AE009948
AAN57980	AE014133	CAJ67947	NC_009089					AAN00853	AE009948
CAD47728	AL732656	CAJ68743	NC_009089					AAN00970	AE009948
CAD46787	AL732656	CAJ70242	NC_009089					AAN57965	AE014133
CAD45886	AL732656	CAJ70290	NC_009089					CAD45856	AL732656
CAC99190	NC_003210	CAM32745	AM410044					CAD45882	AL732656
CAC67541	AJ278471	CAP20357	AM909652					CAD46973	AL732656

Protéine de couplage		Relaxase		Intégrase					
				Transposase à DDE		Intégrase à Sérine		Intégrase à Tyrosine	
AAF72343	AF192329	CAR69046	NC_011900					CAD46984	AL732656
		CAR69099	NC_011900					CAD47732	AL732656
		CAW99790	FM204884					ABP89869	CP000407
		CAZ51597	FM252031					CAZ51585	FM252031
		CAZ55342	FM252032					CAZ55861	FM252032
		CAZ55852	FM252032					CBI14137	FN597254
		CBG92849	FN555436					AEH56837	CP002843
		CBI14161	FN597254					AEH56850	CP002843
		CBW39306	FR671412					AEH56866	CP002843
		EFF33509	ABQJ01000139					AAV61550	CP000023
		EFV96567	AEQQ01000099					AAV63467	CP000024
								ABJ67043	CP000419

Tableau 1 : Liste des protéines de référence utilisées pour le traitement IceKoriBlastP. Pour chaque protéine, son identifiant embl_id est indiqué ainsi que l'identifiant du génome qui la code.

	genome_id													
	AE009948	AE014133	AL732656	CP000023	CP000024	CP000056	CP000407	CP000419	CP002215	CP002843	FM204884	FM252031	FM252032	FN597254
ref_DDE	0	0	4	0	0	0	0	0	0	0	0	0	0	0
ref_ser	0	0	0	0	0	0	0	0	0	0	1	0	1	0
ref_tyr	4	1	5	1	1	1	1	1	2	3	0	1	1	1
ref_relaxa	4	1	2	1	1	1	1	0	2	3	1	1	2	1
ref_coupl	1	1	3	0	0	1	1	0	2	2	0	1	2	1

Tableau 2 : Distribution des protéines de référence dans 14 des 72 génomes étudiés. Dans le tableau les valeurs indiquent le nombre de protéines de référence codées par les différents génomes.

Genome embl_id	taxon	Genome_refseq
AE004092	Bacteria/*Streptococcus_pyogenes*_M1_GAS_uid57845	NC_002737
AP012053	Bacteria/*Streptococcus_gallolyticus_gallolyticus*_ATCC43143	NC_017576
CP002641	Bacteria/*Streptococcus_suis*_D9	NC_017620
CP002651	Bacteria/*Streptococcus_suis*_ST1	NC_017950
CP002644	Bacteria/*Streptococcus_suis*_D12	NC_017621
CP002640	Bacteria/*Streptococcus_suis*_SS12	NC_017619
CP002888	Bacteria/*Streptococcus_salivarius*_57	NC_017594
CP002904	Bacteria/*Streptococcus_equi_zooepidemicus*_ATCC35246	NC_017582
FQ312027	Bacteria/*Streptococcus_pneumoniae*_OXC141	NC_017592
FR873481	Bacteria/*Streptococcus_salivarius*_CCHSS3	NC_015760
FR875178	Bacteria/*Streptococcus_thermophilus*_JIM8232	NC_017581
FR873482	Bacteria/*Streptococcus_salivarius*_JIM8777	NC_017595
BA000034	Bacteria/*Streptococcus_pyogenes*_SSI_1_uid57895	NC_004606
CP002121	Bacteria/*Streptococcus_pneumoniae*_AP200_uid52453	NC_014494
CP000921	Bacteria/*Streptococcus_pneumoniae*_Taiwan19F_14_uid59119	NC_012469
CP000261	Bacteria/*Streptococcus_pyogenes*_MGAS2096_uid58573	NC_008023
CP002843	*Streptococcus parasanguinis* ATCC 15912	NC_015678
FQ312030	Bacterie/*Streptococcus_pneumoniae*_INV104	NC_017591
CP002340	Bacteria/*Streptococcus_thermophilus*_NDO03	NC_017563
CP002176	Bacteria/*Streptococcus_pneumoniae*_670_6B_uid52533	NC_014498
CP000410	Bacteria/*Streptococcus_pneumoniae*_D39_uid58581	NC_008533
CP002633	Bacteria/*Streptococcus_suis*_ST3_uid66327	NC_015433
AE014074	Bacteria/*Streptococcus_pyogenes*_MGAS315_uid57911	NC_004070
AE009949	Bacteria/*Streptococcus_pyogenes*_MGAS8232_uid57871	NC_003485
CP003068	Bacteria/*Streptococcus_pyogenes*_Alab49	NC_017596
CP000024	*Streptococcus thermophilus* CNRZ1066	NC_006449
CP000837	Bacteria/*Streptococcus_suis*_GZ1	NC_017617
CP002465	Bacteria/*Streptococcus_suis*_JS14	NC_017618
CP002215	Bacteria/*Streptococcus_dysgalactiae_equisimilis*_ATCC12394	NC_017567
CP002570	Bacteria/*Streptococcus_suis*_A7	NC_017622
FQ312029	Bacteria/*Streptococcus_pneumoniae*_INV200	NC_017593
CP000114	Bacteria/*Streptococcus_agalactiae*_A909_uid57935	NC_007432
AE005672	Bacteria/*Streptococcus_pneumoniae*_TIGR4_uid57857	NC_003028
FM252032	Bacteria/*Streptococcus_suis*_BM407_uid59321	NC_012926
CP000419	Bacteria/*Streptococcus_thermophilus*_LMD_9_uid58327	NC_008532

CP000262	Bacteria/*Streptococcus_pyogenes*_MGAS10750_uid58575	NC_008024
AP010935	Bacteria/*Streptococcus_dysgalactiae_equisimilis*_GGS_124_uid59103	NC_012891
CP000387	Bacteria/*Streptococcus_sanguinis*_SK36_uid58381	NC_009009
FM252031	Bacteria/*Streptococcus_suis*_SC84_uid59323	NC_012924
CP000919	Bacteria/*Streptococcus_pneumoniae*_JJA_uid59121	NC_012466
FM204883	Bacteria/*Streptococcus_equi*_4047_uid59259	NC_012471
AE009948	Bacteria/*Streptococcus_agalactiae*_2603V_R_uid57943	NC_004116
FM204884	Bacteria/*Streptococcus_equi_zooepidemicus*_uid59261	NC_012470
CP000259	Bacteria/*Streptococcus_pyogenes*_MGAS9429_uid58569	NC_008021
CP000407	Bacteria/*Streptococcus_suis*_05ZYH33_uid58663	NC_009442
FN568063	Bacteria/*Streptococcus_mitis*_B6_uid46097	NC_013853
CP000003	Bacteria/*Streptococcus_pyogenes*_MGAS_10394	NC_006086
AP012054	Bacteria/*Streptococcus_pasteurianus*_ATCC_43144_uid68019	NC_015600
CP000829	Bacteria/*Streptococcus_pyogenes*_NZ131_uid59035	NC_011375
CP000408	Bacteria/*Streptococcus_suis*_98HAH33_uid58665	NC_009443
CP001129	Bacteria/*Streptococcus_equi_zooepidemicus*_MGCS10565_uid59263	NC_011134
CP000056	Bacteria/*Streptococcus_pyogenes*_MGAS6180_uid58335	NC_007296
AM946015	Bacteria/*Streptococcus_uberis*_0140J	NC_012004
AP010655	Bacteria/*Streptococcus_mutans*_NN2025_uid46353	NC_013928
FN597254	Bacteria/*Streptococcus_gallolyticus*_UCN34	NC_013798
CP001993	Bacteria/*Streptococcus_pneumoniae*_TCH8431_19A_uid49735	NC_014251
AM295007	Bacteria/*Streptococcus_pyogenes*_Manfredo_uid57847	NC_009332
AM946016	Bacteria/*Streptococcus_suis*_P1_7_uid32235	NC_012925
CP000260	Bacteria/*Streptococcus_pyogenes*_MGAS10270_uid58571	NC_008022
CP000017	Bacteria/*Streptococcus_pyogenes*_MGAS5005_uid58337	NC_007297
AE014133	Bacteria/*Streptococcus_mutans*_UA159_uid57947	NC_004350
CP000918	Bacteria/*Streptococcus_pneumoniae*_70585_uid59125	NC_012468
CP001015	Bacteria/*Streptococcus_pneumoniae*_G54_uid59167	NC_011072
CP000023	Bacteria/*Streptococcus_thermophilus*_LMG_18311_uid58219	NC_006448
AL732656	Bacteria/*Streptococcus_agalactiae*_NEM316_uid61585	NC_004368
CP002925	Bacteria/*Streptococcus_pseudopneumoniae*_IS_7493	NC_015875
CP001033	Bacteria/*Streptococcus_pneumoniae*_CGSP14_uid59181	NC_010582
CP000920	Bacteria/*Streptococcus_pneumoniae*_P1031_uid59123	NC_012467
CP000936	Bacteria/*Streptococcus_pneumoniae*_Hungary19A_6_uid59117	NC_010380

CP000725	Bacteria/*Streptococcus_gordonii*_Challis_substr__CH1_uid57667	NC_009785
CP002471	Bacteria/*Streptococcus_parauberis*_KCTC_11537_uid67355	NC_015558
AE007317	Bacteria/*Streptococcus_pneumoniae*_R6_uid57859	NC_003098

Tableau 3 : Liste des 72 génomes utilisés dans cette étude.

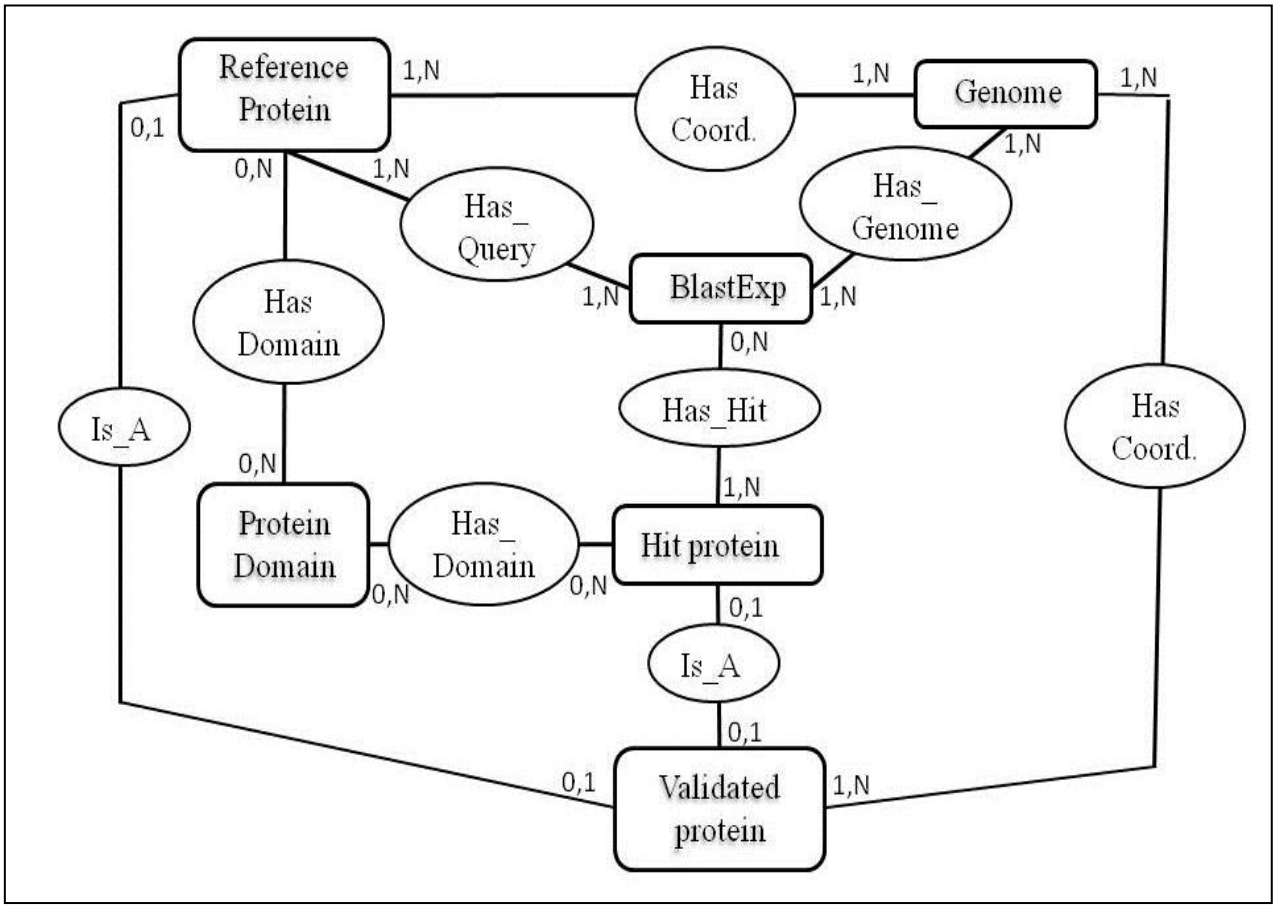

Figure 6 : Modèle conceptuel de la base de données ICEdb. Les rectangles représentent les entités et les ovales représentent les associations existant entre ces entités.

2.1.2 Le modèleICEdb

Plusieurs tables ont été créées pour structurer la base de données :

1. les tables contenant les protéines de référence (Ref0) et les tables contenant les informations sur les différents génomes
2. les tables contenant les informations des différents blasts.
3. les tables répertoriant l'ensemble des protéines homologues aux protéines de référence (Ref0) retrouvées par analyse de chaque génome par Koriblast. Ces protéines sont appelées « hits ».
4. les tables contenant les protéines candidates pour la validation.
5. les tables contenant les protéines validées par l'expert (« hits »)
6. les tables contenant les informations sur les domaines caractérisés au sein de chacune de ces protéines.

Ce modèle de base de données a pour but d'harmoniser les différentes tables, afin que chaque personne puisse se les approprier et les utiliser de manière efficace.

2.2Nrdb90.pl

'Nrdb' est un script Perl édité par (Holm et Sander, 1998). Ce script permet de scanner et de classer des séquences protéiques ayant un certain degré de similarité. Il est utile pour regrouper en cluster les séquences ayant au moins 90% d'identité entre elles. Chaque cluster possède un représentant. C'est ce représentant qui jouera le rôle de protéine de référence et qui sera utilisé dans KoriBlast en tant que query. Ce script a été téléchargé à partir du site http://ekhidna.biocenter.helsinki.fi/downloads/rsdb/nrdb90.pl

2.3 Koriblast

KoriBlast est un logiciel édité par l'entreprise Korilog. Ce logiciel permet d'exécuter le programme blast de manière locale et d'appliquer des filtres permettant l'élimination de hits non pertinents. L'outil Koriblast a permis de rechercher les protéines « étiquettes » codées par les génomes de streptocoques.

Une procédure de Blastp réalisée à l'aide de Koriblast permet à partir des fichiers contenant les séquences des protéines étiquettes (« query ») de rechercher, dans le fichier multifasta protéiques de chacun des génomes étudiés, les protéines homologues (hit). L'ensemble des résultats obtenus (hits) sont alors soumis à une procédure appelée, filtre, qui permet de ne retenir que les résultats pertinents, c'est-à-dire répondant à différents critères de sélection définis par l'expert (E-value, pourcentage d'identité, recouvrement de la séquence requête, longueur du 'hit', …). Les protéines répondant aux critères de sélection sont appelées protéines étiquettes validées. Pour chaque type de protéines étiquettes, plusieurs filtres ont dû être testés avant de trouver un compromis satisfaisant (voir procédure ICEKoriblastP).

2.4 Artémis

Artemis est un logiciel de visualisation et d'annotation de séquences génomiques. Son interface graphique permet d'accéder à la séquence nucléotidique des génomes, de positionner les ORF ('Open Reading Frame') et d'accéder à la séquence des protéines qu'elles codent. Artemis a été utilisé pour distinguer, parmi les ORF des génomes, celles codant les protéines étiquettes validées et identifier les gènes qui les bordent. Ce logiciel permet également de visualiser des séquences répétées. L'ouverture de ce logiciel ainsi que le chargement de tous les fichiers utiles peuvent se faire de manière automatique. Afin de visualiser facilement les différentes catégories de protéines étiquettes Artemis a été paramétré de sorte que les intégrases apparaissent en vert, les relaxases en jaune et les protéines de couplage en rose. Les séquences répétées potentielles apparaissent en rouge. Artémis peut se lancer manuellement, à la condition de suivre l'ordre, indiqué ci-dessous, d'ouverture des fichiers :

1) le fichier 'genome.fasta', contenant la séquence du génome au format fasta ;

2) le fichier'genome.embl' ;

3) le fichier 'resultat.embl' avec les coordonnées des gènes codant les protéines étiquettes validées ;

4) le fichier dri.txt en fonction des besoins (ce type de fichier contient les drHit trouvées grâce à blast2seq).

2.5Perl

Perl est un langage de programmation, créé par Larry Wall en 1987. Il permet de créer des commandes simples qui seront lues ligne par ligne. Chaque commande est terminée par un point-virgule. Ce langage est très pratique pour remanier ou créer des fichiers textes (les fichiers embl sont des fichiers textes avec une syntaxe particulière), en effet le langage perl permet d'intégrer des expressions régulières dans les lignes de commandes. Ce langage est utilisé pour la création de fichiers lisibles par Artémis.

2.6 BLAST2SEQ

Blast2Seq (bl2seq) est un logiciel qui permet comparer les séquences 2 à 2. Au cours de ce travail, sa fonction Blastn a été utilisée pour rechercher les bornes des ICE. Celles-ci sont caractérisées, dans le cas des ICE codant une intégrase à tyrosine, par la présence d'une séquence (de 8 à 60 nt) répétée de part et d'autre de l'ICE. Les intégrases à tyrosine étant localisées à une des extrémités des ICE, leur présence permet de localiser une des bornes et

de rechercher à sa proximité une séquence qui sera répétée à l'autre extrémité de l'élément. L'autre borne sera localisée à une distance maximale de 300kb du gène codant l'intégrase. Pour identifier les séquences répétées de part et d'autre de l'élément ; la méthodologie a consisté à définir pour chaque intégrase, une séquence requête (appelée seqDRquery) et de l'utiliser dans des expériences de Blastn pour rechercher des séquences identiques (ou similaires) dans les 300 kb en amont du gène codant l'intégrase. Les résultats de cette analyse sont stockés dans deux fichiers :

- un fichier contenant les séquences hits obtenus (« DRhits').
- un fichier contenant le détail des différents alignements entre la séquence seqDRquery et les DRhits

Voici un exemple de ligne de commande permettant l'exécution du logiciel : « bl2seq.exe –i CP002843.1-fasta.fasta -I 1936816, 2236826 -j CP002843.1-fasta.fasta -J 1936826,1936963 -pblastn-r4 -q-5 -G12 -E8 -e0.5 -W8 -o out.txt -D1 »

–i correspond au chemin d'accès au génome au format fasta contenant la séquence query,

-I correspond à la localisation sur le génome de la séquence query,

-j et –J correspondent au chemin d'accès au génome au format fasta contenant la zone de recherche et les positions de la zone de recherche.

-p (programme) correspond au type de BLAST voulu.

-ret -q correspondent aux pénalités pour, respectivement, les concordances (« match ») et les discordances (« mismatch ») de nucléotides entre 2 séquences.

-G et -E correspondent aux pénalités pour, respectivement, l'ouverture et l'extension d'un décalage entre 2 séquences (« GAP »).

-e (evalue) correspond au filtre pour les E-values maximales.

-W (word) correspond au plus petit hit trouvé.

-o (output) correspond au fichier de sortie.

-D1 permet de créer deux fichier différents, si –D1 est présent dans la ligne, le fichier de sortie sera un fichier lisible par Artémis. Si –D1 est absent de la ligne alors le fichier de sortie sera un rapport détaillé des alignements. Dans cette étude, ces deux types de fichiers ont été générés. Ce type de recherche sera effectué pour toutes les intégrases identifiées et validées. Le logiciel est téléchargeable à l'adresse suivante : ftp://ftp.ncbi.nih.gov/blast/executables/blast+/

3) Méthodes utilisées et développées au cours du stage

3.1 ICEKoriBlastP

Cette méthode recouvre l'ensemble des étapes visant à identifier les protéines étiquettes codées par les génomes des streptocoques et à positionner les gènes qui les codent. Ces étapes incluent : l'étape BlastP proprement dite et le filtrage des résultats de Koriblast à l'aide de filtres spécifiques de chaque type de protéines (Tableau 4). Pour les protéines de couplage, deux filtres doivent être appliqués en raison des deux fourchettes de longueur qui servent à filtrer les 'hits'(protéine de couplage 1, protéine de couplage 2). La Figure 7 donne un exemple de fichier de résultats obtenus après cette étape ICEKoriBlastP.

3.2 Traitement et intégration des données dans ICEdb

Les résultats fournis par ICEKoriblastP sont redondants. En effet, puisque chaque séquence protéique de référence est utilisée comme requête sur l'ensemble génomes, une même protéine (hit) peut être sélectionnée plusieurs fois si elle présente un bon alignement avec plusieurs protéines requêtes. Il est donc nécessaire de ne garder que les résultats obtenus avec la meilleure protéine requête pour chaque protéine 'hit'. Pour ce faire deux méthodes sont possibles :

a. En ouvrant le fichier contenant les résultats de Koriblast à l'aide de Microsoft Excel. Il faut appliquer un classement aux données affichées. Les protéines 'hit' sont classsées en fonction de l'ordre alphabétique de leur identifiant embl_id. Puis, elles sont classées par ordre décroissant sur leur pourcentage d'identité. Ce classement permet de repérer les doublons et de les éliminer. Excel gardera ainsi la ligne correspondant à l'embl_id ayant le plus grand pourcentage d'identité.
b. En passant par la base de données ICEdb, il faut d'abord y intégrer les résultats obtenus. Pour cela, le fichier contenant les résultats doit être modifié à l'aide d'un éditeur de texte classique. Les signes =,] ,(,) doivent être remplacés par un « ; ». Les informations sont ensuite importées dans Excel où des entêtes ou des colonnes qui ne serviront pas par la suite seront supprimées. Le fichier est ensuite sauvegardé en format csv (« comma separated values ») en choisissant le « ; » comme séparateur, afin d'être intégré à la base de données sous la forme d'une table de type 'travail', notée par exemple 'travail_relaxase' pour les relaxases.

.

	E-value	Hit coverage	Query coverage	Hit lengh
Protéines de couplage 1	1,00E-05	25%	25%	>180 ; <700
Protéines de couplage 2	1,00E-05	25%	25%	>1000 ; <1200
relaxase	1,00E-04	30%	30%	>180
intégrase à tyrosine	1,00E-04	25%	25%	>320
intégrase à serrine	1,00E-04	25%	25%	>320
transposase à DDE	1,00E-04	25%	25%	>250

Tableau 4 : détail des filtres utilisés dans KoriBlast.

#	Q. Name	Q. Length	Q. Coverage	H. Accession	H. Definition	H. Length	H. Frame	H. Coverage	E-value	Identity	Positive	Align. Length
1	tr|F5WXU8|F5WXU8_STRG1	430	94,90%	AER20413	integrase family protein [Streptococcus suis D12]	390	1	99,20%	9,00E-48	33,50%	53,70%	415
1	tr|F5WXU8|F5WXU8_STRG1	430	56,70%	AER19884	integrase [Streptococcus suis D12]	324	1	76,50%	5,00E-10	27,90%	44,70%	262
1	tr|F5WXU8|F5WXU8_STRG1	430	89,80%	AER18449	phage integrase family site-specific recombinase [Streptococcus suis D12]	376	1	95,50%	4,00E-08	26,50%	43,70%	407
1	tr|F5WXU8|F5WXU8_STRG1	430	74,70%	AER20430	phage integrase family site-specific recombinase [Streptococcus suis D12]	373	1	79,10%	3,00E-05	25%	42,20%	332
2	tr|B9DTA7|B9DTA7_STRU0	381	100%	AER20413	integrase family protein [Streptococcus suis D12]	390	1	99,50%	9,00E-80	43,30%	63,80%	390
2	tr|B9DTA7|B9DTA7_STRU0	381	89,20%	AER18449	phage integrase family site-specific recombinase [Streptococcus suis D12]	376	1	89,60%	1,00E-16	27,50%	43%	356
2	tr|B9DTA7|B9DTA7_STRU0	381	76,60%	AER20430	phage integrase family site-specific recombinase [Streptococcus suis D12]	373	1	79,40%	2,00E-16	28,10%	43,90%	303
2	tr|B9DTA7|B9DTA7_STRU0	381	80,60%	AER19884	integrase [Streptococcus suis D12]	324	1	99,10%	1,00E-15	25,50%	46%	326
3	tr|B5XID0|B5XID0_STRPZ	378	81,50%	AER19884	integrase [Streptococcus suis D12]	324	1	99,40%	2,00E-36	33%	55,70%	327
3	tr|B5XID0|B5XID0_STRPZ	378	92,10%	AER20413	integrase family protein [Streptococcus suis D12]	390	1	91%	9,00E-16	27,20%	44,80%	368
3	tr|B5XID0|B5XID0_STRPZ	378	78,60%	AER20430	phage integrase family site-specific recombinase [Streptococcus suis D12]	373	1	79,90%	5,00E-12	27,10%	47,10%	310
3	tr|B5XID0|B5XID0_STRPZ	378	69,30%	AER18449	phage integrase family site-specific recombinase [Streptococcus suis D12]	376	1	69,40%	4,00E-07	25,60%	44,10%	270
4	tr|B9DVQ1|B9DVQ1_STRU0	383	99,20%	AER20413	integrase family protein [Streptococcus suis D12]	390	1	96,90%	2,00E-13	24,40%	43%	405
4	tr|B9DVQ1|B9DVQ1_STRU0	383	59,80%	AER20430	phage integrase family site-specific recombinase [Streptococcus suis D12]	373	1	60,10%	5,00E-09	28,20%	46,10%	241
4	tr|B9DVQ1|B9DVQ1_STRU0	383	96,30%	AER18449	phage integrase family site-specific recombinase [Streptococcus suis D12]	376	1	94,40%	5,00E-08	26,20%	42,80%	390
4	tr|B9DVQ1|B9DVQ1_STRU0	383	64,80%	AER19884	integrase [Streptococcus suis D12]	324	1	79,30%	1,00E-06	26%	44%	273
5	tr|B1I8G4|B1I8G4_STRPI	388	100%	AER20413	integrase family protein [Streptococcus suis D12]	390	1	99,50%	2,00E-145	68,80%	81,20%	388
5	tr|B1I8G4|B1I8G4_STRPI	388	77,80%	AER19884	integrase [Streptococcus suis D12]	324	1	95,40%	1,00E-14	24,20%	44,70%	322
5	tr|B1I8G4|B1I8G4_STRPI	388	89,90%	AER18449	phage integrase family site-specific recombinase [Streptococcus suis D12]	376	1	90,20%	3,00E-14	25,30%	45,90%	364

Figure 7 : Fichier de sortie d'ICEkoriblastP après application des filtres

a. L'étape suivante est de créer une vue 'max_identité'(view) grâce à une requête sql (figure 8). Cette requête va sélectionner le pourcentage d'identité maximum présent pour chaque embl_id. Grâce à cette vue et à la table, il est possible de créer une nouvelle vue (figure 9) contenant les informations de la table travail restreintes par la vue « max_identité ». Cette nouvelle vue sera de type 'found' et contiendra l'ensemble des protéines étiquettes candidates pour l'identification d'ICE.

Un script a été créé pour modifier de manière automatique les fichiers de sortie du traitement IceKoriBlastP. Il s'agit du script Perl '01signatureICE-F.pl' (annexe1.1). Il prend, en entrée, le fichier de sortie KoriBlast où quelques caractères doivent être supprimés comme : « / », « ; », « . », » ,« "». Ce script va permettre la création de deux fichiers : l'un des fichiers est destiné à l'intégration des données dans ICEdb, l'autre correspond au formatage des données pour le script suivant nommé '02visuDR_ICE-FDR.pl'.

Une fois les protéines étiquettes validées par l'expert les informations sont stockées dans une nouvelle table (« validation »)

3.3 Traitement des listes de protéines étiquettes pour la visualisation et la recherche de DR

Le traitement des listes de protéines étiquettes validées pour la visualisation et la recherche de DR est réalisé par le script '02visuDR_ICE-FDR.pl' (annexe 1.2). Ce script prend, en entrée, un fichier csv (point-virgule) nommé genome.csv, composé de 11 colonnes appelées ici position (pos1; pos2; pos 3; pos4; pos5; pos6; pos7; pos8; pos9; pos10; pos11). S'il manque une information sur une position alors un espace vide est laissé. Chaque colonne doit contenir les informations suivantes :

pos1 : embl_id de la protéine cette information est primordiale pour la bonne exécution du logiciel

pos2 : description de la protéine, récupérée à partir du fichier embl

pos3 : le numéro d'accession du génome au format embl (genome_id)

pos4 : la position de début de l'ORF de la protéine

pos5 : la position de fin de l'ORF de la protéine

pos6 : la 2ème position de début de l'ORF de la protéine si la protéine est en deux morceaux (marquée par le term 'join' dans le fichier EMBL)

pos7 : la 2ème position de fin de l'ORF de la protéine si la protéine est en deux morceaux (marquée par le term 'join' dans le fichier EMBL)

```
CREATE OR REPLACE VIEW max_identity_percent_recombinases_tyr_secound_round AS
 SELECT DISTINCT travail_recombinase_tyr_secound_round.embl_id,
max(travail_recombinase_tyr_secound_round.identity_percent) AS max
 FROM travail_recombinase_tyr_secound_round
 GROUP BY travail_recombinase_tyr_secound_round.embl_id
 ORDER BY travail_recombinase_tyr_secound_round.embl_id,
max(travail_recombinase_tyr_secound_round.identity_percent);
```

Figure 8 requête pour la création de la vue « max_identité »

```
(CREATE OR REPLACE VIEW found_recomb_tyr88_secound_round_blastexp_14 AS
 SELECT DISTINCT
travail_recombinase_tyr88_secound_round.query_number,travail_recombinase_tyr88_secound_round.
query_name,travail_recombinase_tyr88_secound_round.q_coverage_percent,
travail_recombinase_tyr88_secound_round.e_value,
max_identity_percent_recomb_tyr88_secound_round.max,
travail_recombinase_tyr88_secound_round.align_length,
travail_recombinase_tyr88_secound_round.hit_coverage,
travail_recombinase_tyr88_secound_round.genome_id,
convertion_table_second_round_recombinase_tyr88.uniprot_id,
biomart_hit_round_recombinases_tyr88.taxon, biomart_hit_round_recombinases_tyr88.pfam_id,
biomart_hit_round_recombinases_tyr88.start_h, biomart_hit_round_recombinases_tyr88.stop_h,
biomart_hit_round_recombinases_tyr88.interpro_id,
biomart_hit_round_recombinases_tyr88.domaine_name,
travail_recombinase_tyr88_secound_round.q_length,
travail_recombinase_tyr88_secound_round.h_length
  FROM biomart_hit_round_recombinases_tyr88, convertion_table_second_round_recombinase_tyr88,
max_identity_percent_recomb_tyr88_secound_round, travail_recombinase_tyr88_secound_round
 WHERE convertion_table_second_round_recombinase_tyr88.embl_id::text =
travail_recombinase_tyr88_secound_round.embl_id::text AND
convertion_table_second_round_recombinase_tyr88.uniprot_id::text =
biomart_hit_round_recombinases_tyr88.uniprot::text AND
max_identity_percent_recomb_tyr88_secound_round.embl_id::text =
travail_recombinase_tyr88_secound_round.embl_id::text AND
max_identity_percent_recomb_tyr88_secound_round.genome_id::text =
travail_recombinase_tyr88_secound_round.genome_id::text AND
max_identity_percent_recomb_tyr88_secound_round.max =
travail_recombinase_tyr88_secound_round.identity_percent
 ORDER BY travail_recombinase_tyr88_secound_round.query_number,
travail_recombinase_tyr88_secound_round.query_name,
travail_recombinase_tyr88_secound_round.q_coverage_percent,
travail_recombinase_tyr88_secound_round.e_value,
max_identity_percent_recomb_tyr88_secound_round.max,
travail_recombinase_tyr88_secound_round.align_length,
travail_recombinase_tyr88_secound_round.hit_coverage,
travail_recombinase_tyr88_secound_round.genome_id,
convertion_table_second_round_recombinase_tyr88.uniprot_id,
biomart_hit_round_recombinases_tyr88.taxon, biomart_hit_round_recombinases_tyr88.pfam_id,
biomart_hit_round_recombinases_tyr88.start_h, biomart_hit_round_recombinases_tyr88.stop_h,
biomart_hit_round_recombinases_tyr88.interpro_id,
biomart_hit_round_recombinases_tyr88.domaine_name,
```

```
        travail_recombinase_tyr88_secound_round.q_length,
        travail_recombinase_tyr88_secound_round.h_length;
        ALTER TABLE found_recomb_tyr88_secound_round_blastexp_14
OWNER TO ice;
```

Figure 9 requête de la création de la vue found

pos8 : brin codant +ou – ; si la protéine est codée par le brin complémentaire la pos8 doit contenir 'complement'

pos9 : information sur des protéines en deux parties, si la protéine est en deux parties la pos 9 doit contenir 'join' et les pos 6 et pos7 doivent être complétées

pos10 : le type de protéines étiquettes (protéine de couplage, relaxase ou intégrase Tyr, Ser ou Transposase DDE)

pos11 : numéro du locus_tag obtenu dans le fichier embl du génome

Cette mise en forme a été effectuée par le 'script 01signatureICE-F.pl'. Une fois le fichier d'entrée créé, le script peut être activé. Le script demandera à l'utilisateur s'il souhaite que les résultats soient affichés à l'aide d'Artémis. Il faudra aussi que l'utilisateur définisse la distance de recherche des séquences répétées dans les génomes (dans notre cas, 300Kb en amont de l'intégrase). Pour chaque génome, les différents fichiers téléchargés (genome.embl et genome.fasta), et les fichiers résultats (résultats.embl, dri.txt, drirapport.txt) sont rangés dans un dossier portant son identifiant genome_id.

Le programme '02visuDR_ICE-FDR.pl' réalise l'essentiel du procédé d'automatisation, il analyse le fichier contenant les protéines étiquettes de chaque génome:

- il crée pour chaque génome un fichier contenant les protéines étiquettes qu'il code (Résultat.embl) (plus il y a de génomes plus il y a de fichiers).

- il télécharge les informations nécessaires à la visualisation des protéines étiquettes (séquence du ou des génomes à analyser : fichier 'genome.fasta' téléchargeable au NCBI, ensemble des protéines du ou des génomes : fichier genome.embl))

-il recherche les DR affiliés à chaque intégrase en créant une séquence requête notée 'seqDRquery', utilisée avec le logiciel Bl2seq pour faire un BlastN local, sur des zones précises du génome.

- le script crée deux fichiers : un fichier contenant les résultats de Bl2seq contenant les séquences répétées potentielles repérées pour chaque intégrase (fichier 'DRI.txt'), lisible par le logiciel Artémis, et un fichier contenant le détail des alignements produits par Bl2seq entre les séquences répétées potentielles et la SeqDRquery.

- il ouvre les fichiers produits avec le logiciel Artémis. Cela permet de visualiser les gènes codant les protéines étiquettes ainsi que les gènes qui les environnent et d'identifier les bornes des ICE ou IME par l'identification des séquences répétées qui les délimitent.

3.4 Récupération du contenu en séquences codantes des ICE/IME

Le script '03content_ ICE-FDR.pl'(annexe 1.3) permet la récupération du contenu en séquences codantes des ICE identifiés. Une fois qu'un ICE/IME est délimité (par la recherche des séquences répétées), le programme prend, en entrée, les coordonnées de début et de fin de l'ICE/IME sur le génome et télécharge les annotations des séquences codantes comprises entre ces coordonnées à partir du format embl du génome. Il télécharge ce fichier au format texte ou XML en vue de faciliter son exploitation ultérieure. Il crée, en sortie, un fichier 'Resultat.embl', où sont sauvegardés les contenus en séquences codantes des ICE/IME bornés. Ces données seront alors examinées par l'expert et pourront, après validation, être rentrées dans la base de données ICEdb.

3.5Modim

Modim ('Model-Driven Data Integration for Mining'(Bresso*et al.,* 2011)) est un programme qui permet de récupérer des informations se trouvant dans des pages au format HTML ou XML, locales ou distantes, et de copier ces informations dans la base de données. Ce logiciel a été utilisé pour récupérer de façon automatique, pour chaque protéine étiquette de la base de données (c'est-à-dire pour chaque embl_id validé après le traitement IceKoriBlastP), les positions de la séquence codante, le nom du génome, le taxon, le locus tag, etc. MODIM permet, en outre, d'enregistrer directement toutes ces informations dans la base de données. Les tables concernées ont comme identifiant le préfixe 'modim'. Une table finale sera synthétisée avec les informations provenant de trois tables : la table 'found' et la table 'modim'.

III) Résultats

1) Enrichissement et consolidation de la base de données ICEdb

1.1 Intégration des protéines candidates ('hits') et des protéines validées

L'intégration des protéines candidates trouvées par ICEKoriBlastP dans ICEdb concerne deux types de tables :

- les tables 'travail' regroupent les résultats bruts provenant d'ICEKoriBlastp

- les tables 'found' regroupent les protéines ayant passé les filtres (cribles de sélection) qui doivent être validées ou non par l'expert.

Lors de la tentative d'intégration des protéines validées dans ICEdb, plusieurs erreurs provenant d'ancienne intégration ont empêché l'enrichissement :

- l'absence de certains génomes concernés dans les tables de référence

- des problèmes de saisie tels que l'insertion d'un espace à la fin d'une saisie

- la structure inadéquate de certaines tables

Pour résoudre ces problèmes, des modifications de la structure des tables et des liens qui existent entre elles ont dû être apportées.

1.2 Modification des clés primaires

Les clés primaires sont des éléments importants dans la consolidation des tables présentes dans la base de données. Ces clés permettent d'être sûr que chaque ligne de la table correspond à un ensemble unique de valeurs. Elles permettent donc d'éviter l'intégration de lignes doublons dans les tables. Par exemple, dans certaines tables, la clé primaire n'utilisait comme critère que la colonne des identifiants UniProt (uniprot_id). Or un uniprot_id peut renvoyer à plusieurs identifiants EMBL (embl_id). En effet, un uniprot_id renvoie à un ensemble de protéines ayant la même séquence en acide aminés, quel que soit le génome qui les code et le nombre de copies du gène présent dans ces génomes. Au contraire, la banque de données EMBL donne un identifiant différent aux protéines identiques selon le génome de provenance et la position de son gène dans ce génome. Or, diverses relaxases de séquence identique et possédant le même uniprot_id sont codées par des génomes de souches différentes, en particulier de la même espèce, et, dans quelques cas, par des copies multiples du même gène dans le même chromosome. Il en est de même pour diverses protéines de couplage et intégrases. Avec une clé primaire positionnée sur l'uniprot_id, il était impossible d'entrer de nouvelles lignes ayant le même uniprot_id mais des embl_id différents. Une nouvelle clé a été mise en place qui intègre comme critère la colonne des

uniprot_id et la colonne des embl_id. Ainsi il est possible de compléter les tables concernées.

1.3 Modification des clés étrangères

Les clés étrangères jouent un rôle majeur dans la consolidation de la base de données. Ces clés permettent de lier les tables entre elles et empêchent l'insertion de lignes isolées. Par exemple, une clé étrangère existe entre la table des protéines de référence et la table des génomes, faisant correspondre dans les deux tables l'attribut identifiant le génome. Si un utilisateur veut entrer une nouvelle protéine de référence trouvée dans un génome et que ce génome est absent de la table des génomes, l'insertion de cette ligne sera refusée par le système de gestion de la base de données. Certaines tables ne présentaient pas de clés étrangères. Lors de la tentative de création de ces clés, il est apparu que certaines informations étaient absentes des tables concernées. Il a fallu rechercher les données et compléter ces tables afin de pouvoir créer les différentes clés étrangères. De plus, certaines clés étrangères déjà existantes ont dû être modifiées afin de prendre en compte les nouvelles informations définies par les changements de clés primaires.

1.4 Modification de la syntaxe des cellules et du contenu des colonnes

Une fois les différentes clés modifiées, la correction des informations a pu être faite.

- Certaines cellules présentaient un espace à la fin de la saisie. Or, les espaces sont considérés dans une base de données comme des caractères particuliers (\n), donc par exemple pour un génome ('CP01345') si dans la base de données cette information est entrée avec un espace ('CP01345 '), la base de données considère l'information comme différente de l'information initiale.

- Certaines cellules présentaient des erreurs de saisie. Par exemple le génome CP00047 était inscrit dans la table génome de manière erroné CPOOO47 (caractère O au lieu du chiffre 0).

- La colonne genome_id de la table génome contenait des informations imprécises. En effet, au début cette colonne référençait les génomes avec un identifiant RefSeq (ex : NC_nnnnnn). Or ce type d'identifiant n'est pas toujours affecté aux génomes récemment séquencés. Nous avons donc décidé d'utiliser à la place des identifiants EMBL, en créant une nouvelle colonne à la table et en plaçant la clé primaire sur cette colonne. Les identifiants RefSeq ont été conservés comme simples attributs dans la table.

La Figure 10 représente le modèle relationnel de la base de données avant (A) et après consolidation (B). On peut voir que dans la Figure 10B la majorité des tables se réfèrent à la table génomes.

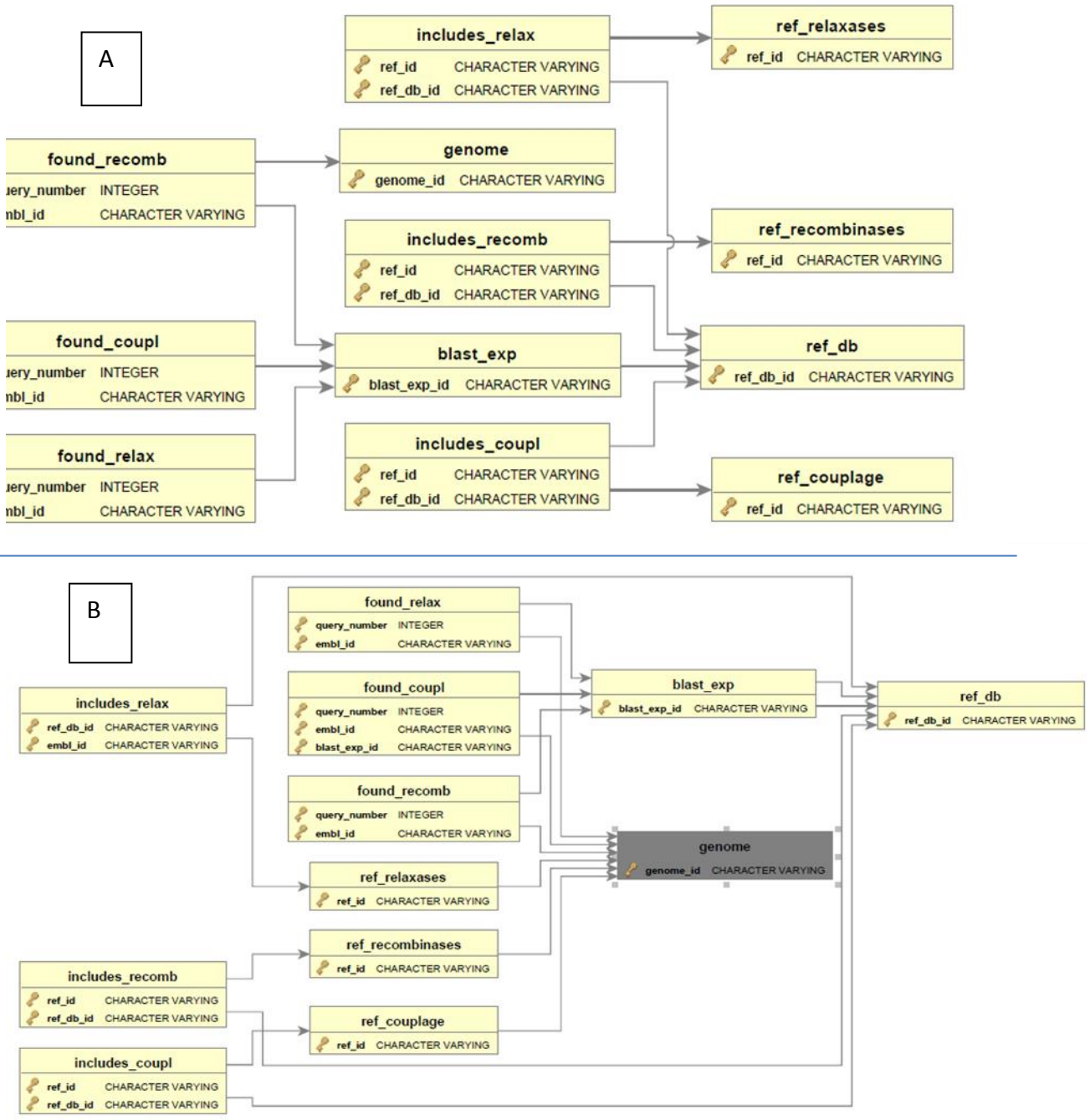

Figure 10 : Modèle relationnel simplifié de la base de données ICEdb. En (A), la base de données le 1 janvier 2013, en (B), la base de données le 1 juin 2013.

1.5 Contenu de la base de données ICEdb

La Figure 11 montre l'enrichissement en protéines étiquettes des ICE et IME pour chaque génome étudié. Le panneau (A) montre la distribution des protéines étiquettes de référence pour les 72 génomes étudiés : seuls 14 génomes sont concernés. Le panneau (B) montre la distribution des protéines étiquettes identifiées par ICEKoriBlastP pour les 72 génomes étudiés : 71/72 génomes sont concernés. Le génome de *Streptococcus pneumoniae* D39 (CP000410) ne code en effet aucune des protéines étiquettes recherchées. Le nombre de protéines détectées varie beaucoup entre les génomes (Fig. 10B). Alors que *Streptococcus gallolyticus subsp. gallolyticus* ATCC 43143 (AP012053) présente 19 nouvelles protéines étiquettes détectées, chez sept génomes une seule nouvelle protéine étiquette est décelée.

Le Tableau 5 récapitule les résultats en montrant le nombre de génomes codant les différents types de protéines étiquettes (A), le nombre de génomes codant différentes combinaisons de protéines étiquettes (B), et le comptage global des protéines étiquettes (C). Cinquante chromosomes codent au moins une relaxase. Comme les relaxases impliquées dans le transfert conjugatif (des plasmides, des ICE ou des IME) sont caractéristiques des éléments conjugatifs ou mobilisables (Smilie *et al*., 2010), ceci suggère donc que 50 souches pourraient posséder au moins un ICE ou un IME. Le Tableau 5B montre que 42 génomes codent les trois types de protéines étiquettes. Sept chromosomes codent une relaxase et une intégrase, suggérant la présence d'IME. Par ailleurs, un chromosome code une protéine de couplage et une relaxase et 5 chromosomes codent une protéine de couplage et une intégrase. Le Tableau (5C) correspond à un comptage des différentes protéines étiquettes. Comme cette analyse a identifié 110 relaxases, ceci suggère que nous pourrions visualiser et borner au maximum 131 nouveaux ICE/IME.

2) Algorithme pour la délimitation des ICE/IME

L'algorithme utilisé pour la délimitation des ICE /IME repose sur les hypothèses suivantes :

- l'ICE/IME est généralement flanqué de courtes répétitions directes de 8 à 60 pb le plus souvent parfaites ('direct repeats' ou DR) qui résultent de son intégration. Cependant, elles sont parfois un peu dégénérées, plus courtes (jusqu'à 2 pb), voire absentes.
- l'une de ces DR recouvre le plus souvent l'extrémité 3' du gène d'insertion (par exemple un gène d'ARNt ou codant une protéine ribosomique). Cependant, l'une des 2 DR recouvre parfois l'extrémité 5' du gène d'insertion au lieu de son extrémité 3'. Les 2 copies du DR peuvent également se trouver dans une région intergénique.

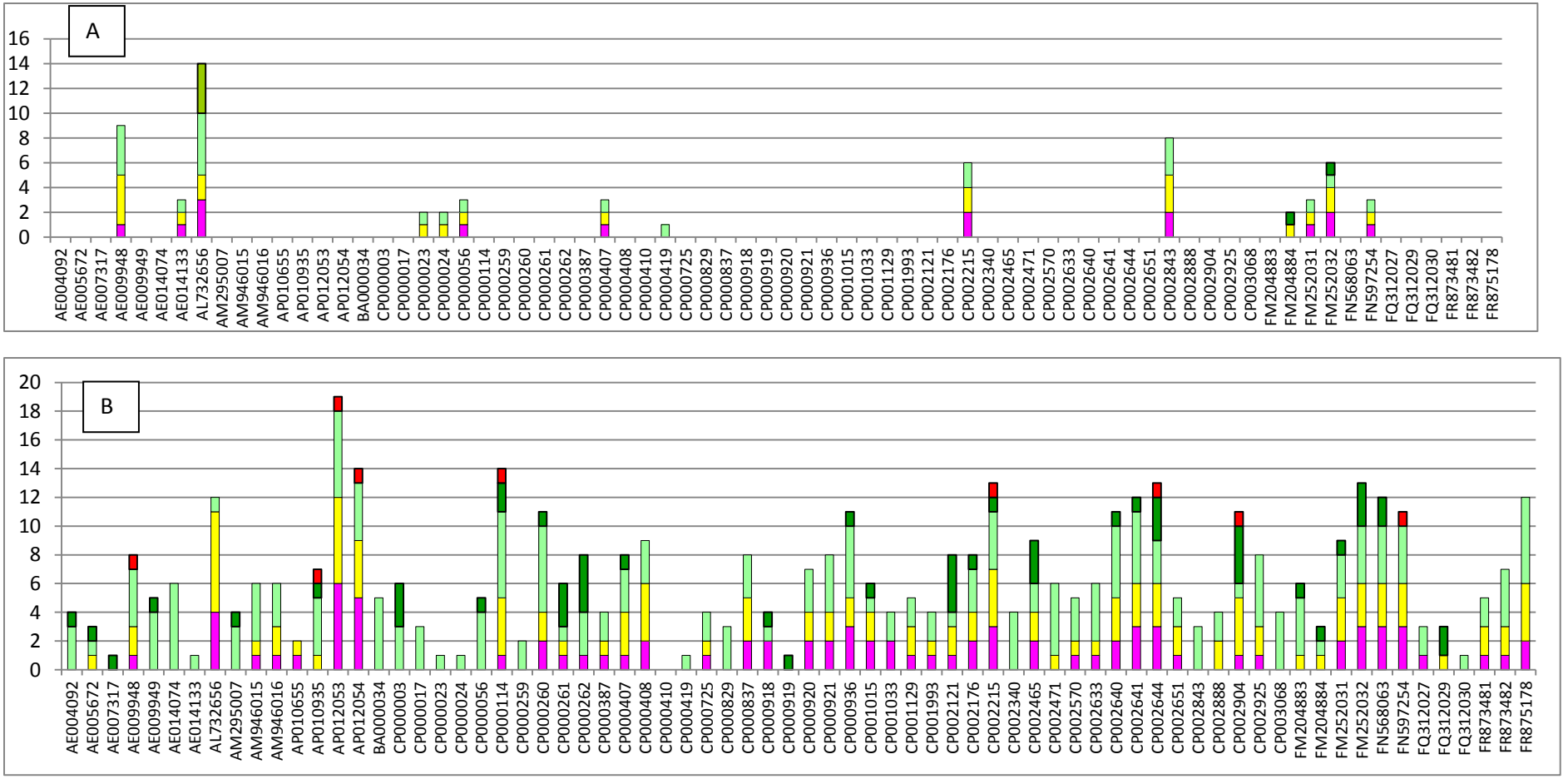

Figure 11 : Distribution des protéines étiquettes à travers les 72 génomes de l'étude. Les protéines de couplage sont représentées en rose, les relaxases sont représentées en jaune, les intégrases à tyrosine sont représentées en vert clair, les intégrases à sérine en vert foncé)et les transposases à DDE en rouge). Les génomes possédant des protéines étiquettes de référence sont présentés dans le tableau A. Le tableau B montre l'enrichissement obtenu pour chaque génome sans les protéines de référence

A

		génome codant	
		ref	blast
type protéique	transposase à DDE	1	9
	intégrase à sérine	2	31
	intégrase à tyrosine	13	66
	relaxase	13	47
	protéine de couplage	10	47

B

	génome codant
3 types de protéines étiquettes	42
protéine de couplage + relaxase	1
protéine de couplage + un type d'intégrase	5
relaxase + un type d'intégrase	7
intégrase	16
protéine de couplage	0
aucune protéine étiquette	1

C

	protéines étiquettes	total
ref	protéine de couplage	15
	relaxase	21
	intégrase à serrine	2
	intégrase à tyrosine	23
	transposase à DDE	4
hit	protéine de couplage	85
	relaxase	110
	intégrase à serrine	53
	intégrase à tyrosine	205
	transposase à DDE	9

Tableau 5 : Trois tableaux récapitulatifs des protéines étiquettes retrouvées après la procédure ICEKoriBlast, la validation par l'expert et l'insertion dans la base de données ICEdb. Le Tableau (A) montre le nombre de génomes codant au moins un type de protéines étiquettes de référence (ref) ou un type de protéines étiquettes nouvellement identifié (blast). Le tableau B montre le nombre de génomes possédant les différentes combinaisons possibles de gènes codant des protéines étiquettes. Le Tableau (C) présente un comptage des différents types de protéines étiquettes pour l'ensemble des 72 génomes.

- l'une de ces DR est généralement localisée dans la région située immédiatement en aval de l'intégrase chez les ICE et IME de firmicutes. La DR la plus proche est parfois séparée du gène de l'intégrase par quelques gènes.

Ces 3 conditions sont remplies simultanément dans une grande majorité des cas observés lorsque l'ICE ou IME code une intégrase à tyrosine catalysant une intégration site-spécifique (Burrus *et al* ., 2002). Elles ne sont généralement pas toutes vérifiées dans les autres cas. Ainsi, les extrémités sont constituées de répétitions inversées dégénérées ou parfaites s'il code une intégrase à DDE (Brochet *et al*., 2009). Globalement, l'algorithme recherchera donc les répétitions flanquantes ou terminales. Il est cependant possible qu'elles n'existent pas ou qu'elles soient trop courtes ou dégénérées pour être distinguées de ressemblances dues au seul hasard.

L'algorithme utilise la fonction d'alignement local du programme blast2seq et comprend cinq étapes.

a. Définition d'une séquence requête pour le programme d'alignement. Cette séquence, notée SeqDRquery est volontairement la plus longue possible et recouvre les 30 dernières paires de bases du gène de l'intégrase, la séquence intergénique jusqu'au prochain gène, les 30 premières paires de bases du gène le plus proche de l'intégrase.
b. Définition de la région à explorer : sur 300 kb en amont de la SeqDRquery.
c. Exécution du programme blast2seq avec les séquences définies en a. et b. et les paramètres précisés dans la section Matériel et Méthodes.
d. Analyse des résultats, récupération des séquences alignées, notées DRhits, et mise au format pour la visualisation sous Artemis.
e. Sélection manuelle et visuelle des répétitions candidates capables de délimiter un ICE/IME. Les critères requis sont les suivants.
    a. (i) Une répétition candidate ne doit pas se situer au milieu d'une ORF. En effet si une répétition se situait au milieu d'une ORF cela signifierait que l'ORF se prolonge dans l'ICE. Ceci ne pourrait se produire que si l'ICE ou l'IME s'est intégré dans une ORF et que l'insertion a provoqué la formation de DR, or ceci n'a jamais été décrit pour les ICE et IME connus de firmicutes. La répétition candidate se situe généralement sur les derniers nucléotides d'un gène codant un ARNt ou d'une ORF.
    b. (ii) Le pourcentage d'identité doit être supérieur à 80%. Il est en général compris entre 90 et 100% pour les DR candidats validés.
    c. (iii) La E-value doit être la plus faible possible.
    d. (iv) La séquence candidate doit être lue dans le même sens que la SeqDRquery, dans le cas d'une intégrase à sérine ou tyrosine.

3) Validation de la recherche des ICE et IME

3.1 Exemple d'un ICE déjà annoté dans la littérature, ICE*Ssu*$_{BM407}$2

La Figure 12 montre une zone du génome de *Streptococcus suis BM407* (FM252032.1). La recherche de DR associés à l'intégrase à tyrosine CAZ55861 a été faite en utilisant comme seqDRquery la région allant de l'extrémité 3' du gène de l'intégrase à celle de l'ORF en aval CAZ55862, extrémités des deux ORF incluses. Cette seqDRquery a permis de mettre en évidence six répétitions candidates ou DRhits. Quatre de ces DRhits sont localisées en milieu d'ORF et sont donc rejetées. Une DRhit est localisée dans la seqDRquery. Elle correspond à une répétition interne à la seqDRquery. Une DRhit de 18 paires de bases est localisée à la fin d'une ORF et a une orientation identique à celle de la seqDRquery. Cette DRhit répond aux critères de sélection : pourcentage d'identité de 94,44%, E-value de 0,1. L'E-value semble élevée mais c'est souvent le cas avec de courtes séquences, comme c'est le cas pour cette DRhit (18 pb). Cette DRhit est placé à 80,5 kb en amont du gène de l'intégrase CAZ55861 utilisée pour former la seqDRquery et inclut la fin du gène *rplL* codant la protéine ribosomique L12 (Figure 12B). Ce DRhit trouvé est exactement co-localisé avec la séquence DR déjà annotée dans le génome. Si on utilise le DRhit trouvé et que l'on utilise ces positions dans le logiciel BL2seq on retrouve l'autre borne déjà annotée. Cet ICE de 80,5kb, ICE*Ssu*$_{BM407}$2, a déjà été décrit et appartient à la famille de Tn*5252* (Holden *et al*., 2009). Les limites de l'ICE ont donc été correctement identifiées par ce travail.

Selon Holden *et al*. (2009), cet ICE contient un ICE de type Tn*916* dont les frontières ont préalablement été identifiées par comparaison de séquences avec d'autres ICE de ce type, ayant une séquence très proche. Ces ICE ne s'intègrent pas de manière site spécifique et ne génèrent pas de DR. La recherche utilisant la seqDRquery associée à l'intégrase de cet élément des DRhit liées à l'intégrase CAZ55849 de cet ICE trouve cependant une DRhit de 34 paires de bases, qui ne correspond cependant pas à la limite de l'élément. Cette DRhit se localise à 75 paires de bases en amont de la limite 5' de l'élément

3.2 Validation sur un génome déjà analysé manuellement par l'équipe ICE-TeA

En 2008, l'équipe ICE-TeA a répertorié les ICE, les IME et les îlots génomiques en dérivant (CIME) présents dans différents génomes de *Streptococcus agalactiae* (Figure 13) (Brochet *et al*., 2008), notamment du génome de *S. agalactiae* A909 (CP000114). Ce génome ne posséde au départ aucun des gènes codant les protéines étiquettes de référence. Afin de valider la méthode ICEFinder, j'ai utilisé notre protocole décrit précédemment et cela m'a effectivement permis de retrouver ou corriger les annotations suivantes déjà publiées :

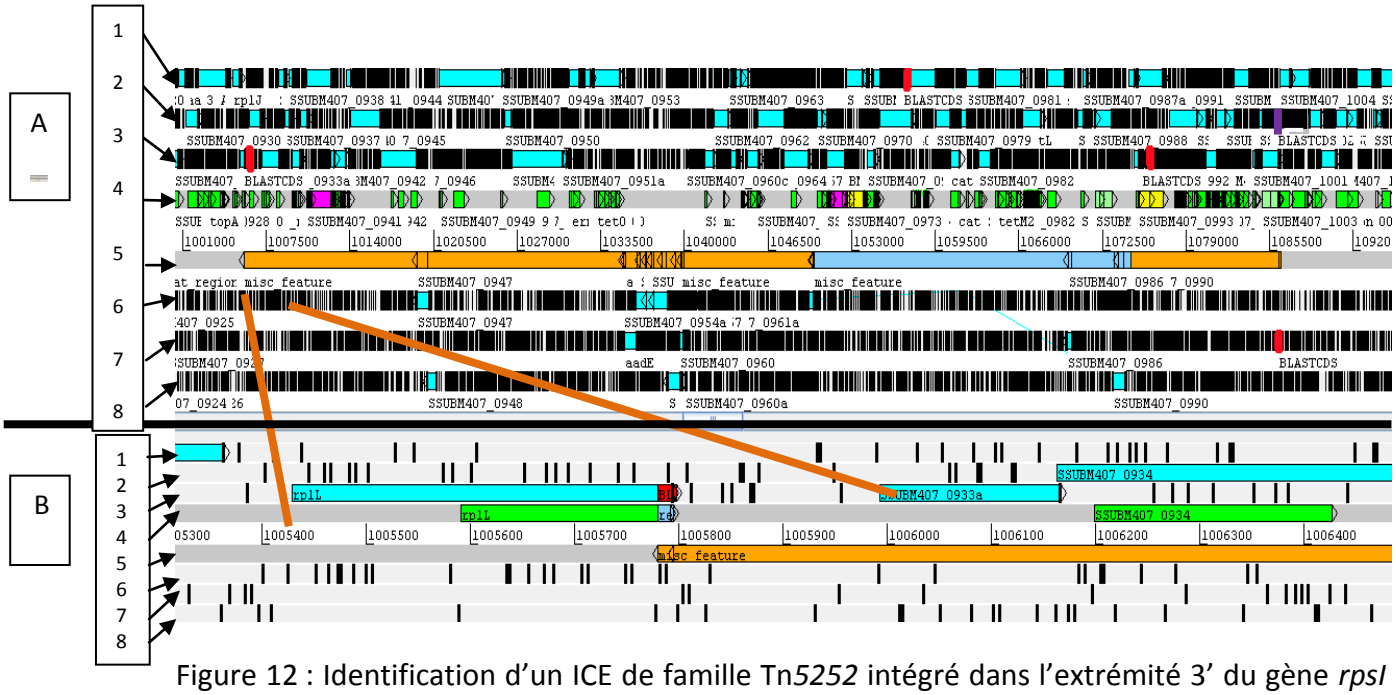

Figure 12 : Identification d'un ICE de famille Tn*5252* intégré dans l'extrémité 3' du gène *rpsI* chez *Streptococcus suis* BM407 (FM252032.1). (A), vue d'ensemble entre les positions 1001000 et 1092066. La figure ne couvre pas les 300 kb utilisés pour la recherche de DRhit, dans cet intervalle, seules 3 des 6 DRhit sont présentes. (B), vue détaillée d'une extrémité de l'élément de famille Tn*5252*. Les lignes 1 2 3 6 7 8 correspondent respectivement aux différents cadres de lectures 1 2 3 -1 -2 -3. Les segments en bleu ciel correspondent aux ORF. .La ligne 4 indique la position des CDS (vert fluorescent). Les segments roses correspondent aux gènes de protéines de couplage, les segments jaunes aux gènes relaxases et les segments vert clair aux gènes d'intégrases. L'espace entre les lignes 4 et 5 donne la position des éléments. Ces positions signifient la même chose dans la partie B. L'analyse a mis en évidence un ICE de famille Tn*5252* (en orange sur la ligne 5) contenant un ICE de famille Tn*916* (en bleu sur la ligne 5). Les DRhit candidates sont représentées en rouge et sont annotées BLASTCDS. La SeqDRquery est colorée en mauve sur la vue d'ensemble (A) L'agrandissement (B) permet d'identifier le gène *rplL* dans l'extrémité 3' duquel est inséré l'ICE de famille Tn*5252* (ligne B4). La région colorée en bleu pâle correspond au DR déjà annoté dans le fichier de séquence de ce génome et en rouge le DR identifié par cette analyse.

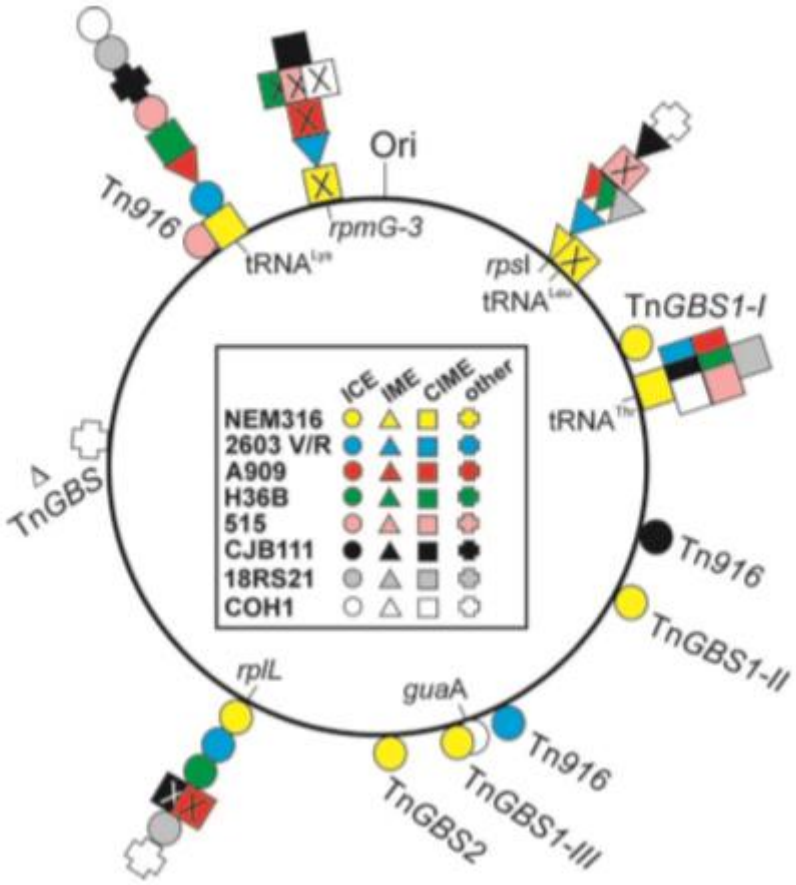

Figure 13 : Localisation des ICE, IME et îlots génomiques censées en dérivant (CIME, other) ainsi que des sites d'insertions dans les génomes de *Streptococcus agalactiae* (Brochet *et al.,*2008). Le génome de la souche A909, décrit dans la partie 2.2 contient les éléments correspondant aux formes géométriques colorées en rouge.

- Le premier IME, IME_*A909_rpsI* se localise dans les positions 241188-249898 et mesure 8,7 kb. Il code une relaxase et une intégrase à tyrosine déjà identifiées. L'analyse de ses DR a permis de retrouver son site d'intégration déjà identifié : l'extrémité 3' du gène *rpsI* codant la protéine ribosomique S9.
- Un élément est localisé entre les positions 1308836 et 1318971 et mesure 10,2 kb. Il est annoté dans la publication comme un îlot génomique CIME_*A909_rplL*, car aucune relaxase n'avait été identifiée à l'époque. L'analyse des données réalisée lors de ce travail a révélé un gène codant une relaxase putative (SAK_1327) et retrouvé celui de l'intégrase à tyrosine déjà identifié. L'élément annoté était censé coder une protéine de couplage qui n'est pas retrouvée lors de cette analyse. Cette « protéine » résultait en fait de la traduction d'un pseudogène tronqué en 5'. Dans notre analyse, le produit de cette traduction ne franchit pas le filtre utilisé du fait d'un taux de couverture trop faible par rapport aux protéines requêtes. L'analyse des DR a permis de retrouver le site d'intégration déjà identifié : l'extrémité 3' du gène *rplL* codant la protéine ribosomique L12. Par ailleurs, cet élément ne code aucune protéine du pore de conjugaison (Brochet *et al*., 2008) et ne peut donc être un ICE. Comme il code une relaxase et une intégrase, il pourrait s'agir d'un IME. Cependant, si l'on tient compte de la troncature du gène de protéine de couplage, cela suggère un élément muté, donc plutôt un dérivé d'IME.
- Un IME, IME_*A909_tRNA*^{Lys}, est localisé en positions 1939018-1957583 et mesure 18,2 kb. En effet, à cette position se trouvent un gène codant une intégrase à tyrosine et un codant une relaxase, très proches l'un de l'autre. Cependant, la publication dit que le site d'intégration se trouve au niveau d'un gène codant un ARNt^{Lys} situé à 2 kb en aval de l'intégrase. L'intégrase n'étant pas à côté du site d'intégration, il était impossible de borner cet IME avec notre méthode. J'ai donc tenté de définir une nouvelle seqDRquery à l'extrémité de ce gène de ARNt^{Lys} et en lançant le programme de recherche de DR, certaines séquences répétées sont retrouvées et devront être examinées par l'expert car cet IME selon les données publiées, résulte d'une accrétion et présente de ce fait 4 DR. (Figure 14).

- Un élément, est localisé en positions 2042688-2052587 et mesure 9,9 kb (Figure 15). Il est annoté dans la publication comme CIME_*A909_rpmG* car aucune relaxase n'avait été identifiée à l'époque. L'analyse a cependant révélé une relaxase putative (SAK_1327), la protéine de couplage et l'intégrase à tyrosine déjà annotées. Le programme retrouve les DR flanquant cet élément. Par ailleurs, cet élément ne code aucune protéine du pore de conjugaison (Brochet *et al*., 2008) et ne peut donc être un ICE. Comme il code relaxase et intégrase, ce pourrait être un IME

Globalement, l'analyse a permis de retrouver les éléments déjà décrits et leurs limites sauf pour IME_*A909_tRNA*^{Lys}. Elle a aussi permis de corriger des faux positifs ou négatifs concernant la présence/absence de gènes de relaxases ou de protéines de couplage.

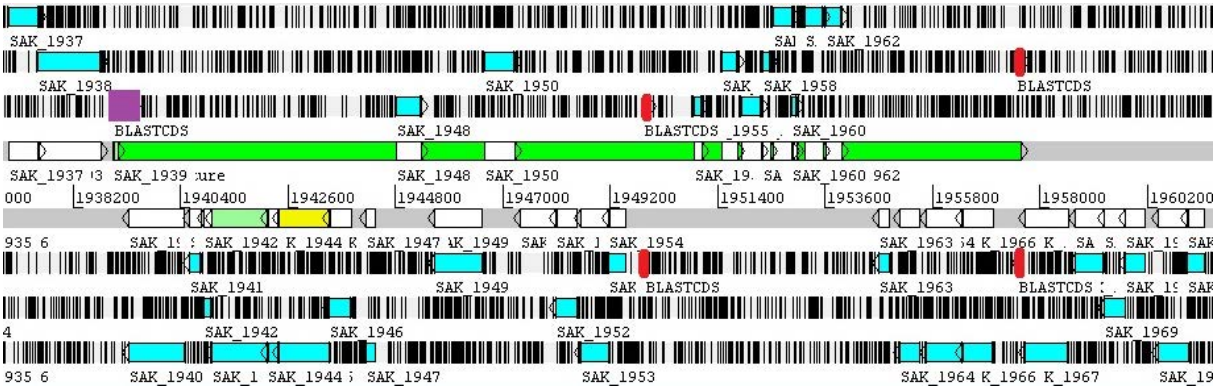

Figure 14 : Identification d'un IME intégré dans un gène d'ARNt^{Lys} chez *S. agalactiae A909*. La figure ne couvre pas les 300 kb utilisés pour la recherche de DRhit, dans cet intervalle, seules 4 des 5 DRhit sont présentes. Cette représentation est identique à celle de la Figure 12 et comporte la même légende à l'exception : (i) des segments blancs correspondant aux gènes présents sur le chromosome, (ii) du segment mauve correspond à la seqDRquery. Les DRhit trouvées en construisant une SeqDRquery en aval du gène d'ARNt^{Lys} sont schématisés en rouge.

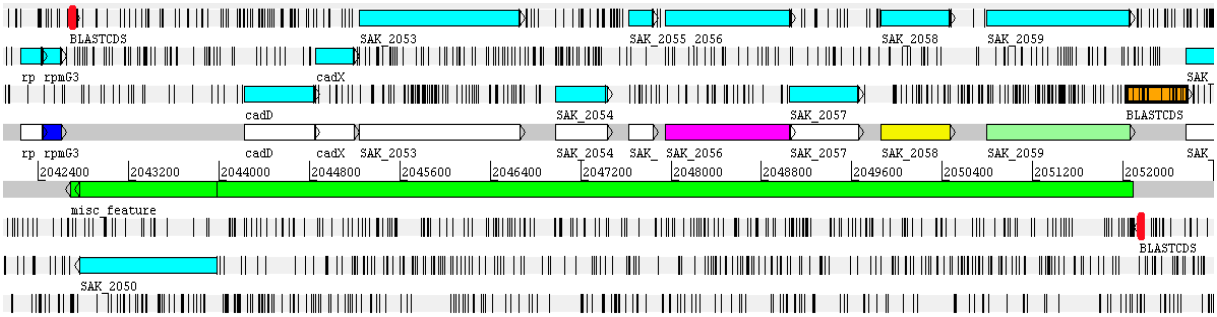

Figure 15 : Identification d'un IME intégré dans le gène *rpmG* chez *S. agalactiae* A909. La figure ne couvre pas les 300 kb utilisés pour la recherche de DRhit. Cette représentation est identique à celle de la Figure 12 et comporte la même légende à l'exception : (i) des segments blanc correspondant aux gènes présents sur le chromosome,(ii) la SeqDRquery est colorée en orange et le site d'insertion (rpmG3) est coloré en bleu foncé. LesDRhit candidates sont colorées en rouge

3.3 Un cas des plus originaux : un IME composite de *Streptococcus parasanguinis* ATCC 15912

Le chromosome de *Streptococcus parasanguinis* ATCC 15912 (CP002843) code 11 protéines étiquettes dont 2 protéines de couplage, 3 relaxases et 6 intégrases à tyrosine. Ceci suggérait que 3 éléments sont susceptibles d'être bornés. Les gènes codant les protéines de couplage, les relaxases et 3 des intégrase sont localisés dans la même région, ce qui suggérait une structure originale qui a été analysée.

3.3.1 Elément intégré dans l'extrémité 3' du gène rpsI

Six DRhits candidats sont détectés à partir de la seqDRquery construite à partir de l'intégrase à tyrosine AEH56837. Trois DR candidates se localisent en milieu d'ORF et ne sont donc pas valides. Les trois autres DR candidates se localisent dans un espace inter-génique (Figure 16 DR colorées en vert pâle). Parmi celles-ci, une DR de 16 paires de bases présente un pourcentage d'identité inférieur à 80% avec son homologue dans la SeqDRquery, deux DRhit présentent un pourcentage d'identité égale à 100%, mais une seule de ces 2 dernières candidates est dans le même sens que la SeqDRquery.

Cette approche a donc permis de définir un élément faisant 52,6 kb qui porte les gènes codant 2 protéines de couplage, 3 relaxases et 3 intégrases à tyrosine. Cet élément, flanqué de répétitions directes de 16 pb, est intégré dans l'extrémité 3' du gène *rpsI* codant la protéine ribosomique S9. La configuration des gènes codant les différentes protéines étiquettes fait penser à un élément composite.

3.3.2 IME interne

En prenant comme séquence requête la SeqDRquery construite à partir de l'intégrase à tyrosine AEH56850(présent dans l'élément bleu) codée par un gène interne de l'élément de 52,6 kb, on détecte 12 répétitions candidates représentaient en bleu foncé. Parmi celles-ci, sept se localisent dans des ORF, les cinq autres se trouvant dans des espaces inter-géniques. Deux de ces cinq candidates se localisent au début d'une ORF ou dans une zone inter-génique mais se localisent à l'extérieur de l'élément décrit précédemment. Les 3 candidates restantes présentent un pourcentage d'identité supérieur à 80 %, et ils sont dans le même sens que la SeqDRquery. Leurs E-values varient de 0,11 à 2^{-6} Si l'on prend en compte la DR candidate dont l'alignement avec la SeqDRquery présente l'E-value la plus faible, l'élément aurait une taille de 13,2 kb et coderait une intégrase à tyrosine, une relaxase et une protéine de couplage. Comme par ailleurs, il ne code pas de protéine du pore de conjugaison, cet élément serait un IME.

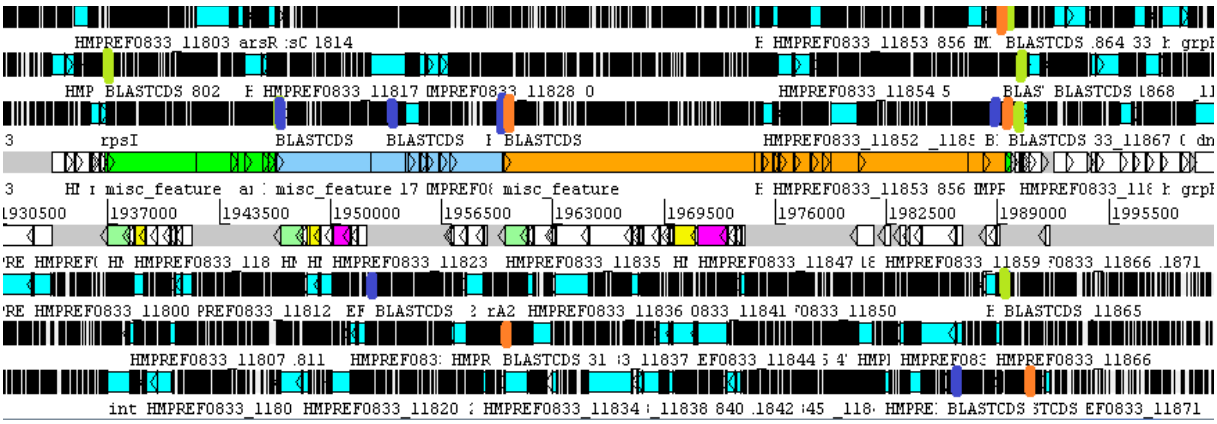

Figure 12 : Identification d'un IME composite dans le génome de *S. parasanguinis* ATCC 15912 (CP002843). La figure ne couvre pas les 300 kb utilisés pour la recherche de DRhit, dans cet intervalle, seules une partie des DRhit sont présentes. Cette représentation est identique à celle de la Figure 12 et comporte la même légende. Le code couleur utilisé pour les protéines étiquettes est le même que celui décrit dans le paragraphe 3.1. La couleur des IME, des ICE bornés est choisie de manière aléatoire afin de bien les différencier. Ici, pour mieux les distinguer les uns des autres, les seqDRquery et les DR candidates présentent une couleur voisine de l'élément auquel apprtient l'intégrase qui sert de point de départ à la définition de la seqDRquery. Par exemple, dans le cas de l'IME coloré en vert, les répétitions candidates seront colorées en vert pâle.

3.3.1.3 ICE interne

Onze DRhit (correspondant aux barres verticales de couleur orange) candidates sont détectées à partir de la SeqDRquery construite à partir de l'intégrase à tyrosine AEH56866 codée par un gène interne de l'élément de 52,6 kb situé en dehors de l'IME défini dans le 3.3.2. Parmi ceux-ci, quatre DR candidates se situent à l'intérieur d'ORF, deux se situent dans la SeqDRquery elle-même, et quatre se localisent dans des régions inter-géniques. L'une de ces quatre candidates se localise à la fin d'une ORF, elle a un pourcentage d'identité de 94,7%, a une taille de 19 paires de bases et est dans le même sens que la SeqDRquery. Mais ce candidat est trop éloigné de l'IME décrit précédemment en 3.3.2. En effet L'ICE que nous étudions ici est intégré dans l'IME donc la borne de l'ICE et par conséquent la DRhit ne peut pas se situer à l'extérieur de l'IME. Les trois autres candidates se situent dans l'IME précédemment décrit. Deux d'entre eux présentent un pourcentage d'identité inférieur à 80% (77,1% pour une taille de 35 paires de bases)(66,2% pour une taille de 144 paires de bases) et ne sont donc pas retenus, le dernier présente un pourcentage d'identité égal à 93,94% pour une taille de 33 paires de bases et une E-value de 9^{-8} de plus il est lu dans le même sens que la SeqDRquery. Il pourrait correspondre à la limite d'un élément. Cet élément serait flanqué de répétitions directes de 34 pb et aurait une taille de 29,3 kb. Il contient non seulement des gènes d'intégrase, de relaxase et de protéines de couplage mais aussi des gènes codant du pore de conjugaison et serait donc un ICE.

3.3.1.4. Structure globale.

L'analyse précédente suggère qu'un élément intégré dans *rpsI* contenait un IME et un ICE intégrés. Si l'on retire ces deux éléments, les séquences restantes incluent un gène codant une intégrase à tyrosine et un gène codant une relaxase. Cela suggère que ces séquences correspondent à un IME. Globalement un IME et un ICE seraient intégrés dans un IME lui-même intégré dans l'extrémité 3' du gène *rpsI*.

4 Conclusion

La création des scripts a permis un gain de temps important dans le processus de délimitation des ICE par rapport à une détection et une délimitation manuel. En effet l'analyse des protéines étiquettes retrouvées dans les 72 génomes (sans l'étape manuelle de choix parmi les DR candidates) peut être réalisée en environ 3 h. Le choix final des bornes doit cependant se faire manuellement. En effet le choix des bornes peut être multiple et seul l'utilisateur pourra décider de valider ou non les bornes détectées.

En appliquant cette méthode pour borner les ICE et les IME, 109 structures ont pu être bornées dans 47 des 72 génomes étudiés alors que 131 relaxases et potentiellement 131 éléments pourraient être présents. Les tailles de ces structures varient de 6 kb à 200 kb. 19 structures ont leur site d'intégration à la fin de gènes codant des ARNt. Certains IME/ICE

n’ont pas pu être bornés car la DR n’est pas à côté de l’intégrase, ou l’intégrase n’a pas été identifiée.

IV) Discussion

1) Enrichissement et consolidation de la base de données ICEdb

La base de données ICEdb a subi de profondes modifications à la suite de ce travail. Tout en restant fidèle au modèle conceptuel, le modèle relationnel a évolué et s'est enrichi de contraintes d'intégrités (clés étrangères par exemple). Cette architecture permet maintenant d'éviter un grand nombre d'erreurs lors de l'insertion manuelle des informations.

Cette architecture peut encore être amenée à changer. En effet, une étape reste à faire dans l'enrichissement de cette base. Une fois les bornes validées et les informations sur le contenu des ICE et des IME téléchargées, ces nouvelles données devront y être intégrées. Ceci impliquera non seulement la création de nouvelles tables mais aussi de nouvelle clés étrangères. Il sera important de discuter avec les utilisateurs pour identifier les informations utiles à stocker dans cette base lors de ces étapes.

L'enrichissement en protéines susceptibles d'être codées par les ICE a été en partie réalisé lors de mon stage de Master 1. Cet enrichissement a été important au vu du petit nombre de génomes étudiés. En effet, nous détectons au total 515 nouvelles protéines codées par les 72 génomes de streptocoques, à partir de 65 protéines de départ. La présence simultanée d'au moins un gène de relaxase et un gène d'intégrase dans 49 génomes peut s'expliquer par la présence d'ICE ou d'IME. Cependant, plusieurs génomes présentent d'autres combinaisons de ces protéines. Ainsi, un chromosome code une protéine de couplage et une relaxase, 5 codent une protéine de couplage et une intégrase, 16 génomes ne codent que des intégrases et 1 ne code aucune des 3 types de protéines étiquettes (intégrase, relaxase, couplage). L'absence d'un, deux ou 3 des types de protéines dans ces 23 génomes peut s'expliquer d'au moins 3 façons :

i) Ces types de protéines sont réellement absents. Dans ce cas, ces génomes ne portent ni IME, ni ICE. Dans les 2 premiers cas, ces chromosomes peuvent cependant porter des îlots génomiques dérivant d'ICE ou d'IME par délétion et/ou pseudogénisation des gènes codant les protéines recherchées. De plus, des intégrases incluses dans la banque de référence après expansion de celle-ci appartiennent clairement à des prophages (Guédon, com. pers.), les intégrases de prophage ne présentent pas de caractéristiques qui permettent de les différencier facilement de celles des ICE et IME. Seule l'analyse des gènes flanquant ceux codant ces intégrases permet de les différencier : par exemple la présence de gènes de capsides ou encore l'absence de gènes codant une relaxase et/ou une protéine de couplage indiquera que cette intégrase correspond probablement à la présence de prophages intégrés.

ii) Il est cependant possible que certaines protéines n'aient pas été détectées en raison de filtres trop sévères ou inadaptés, utilisés lors de l'enrichissement par ICEKoriblastP. Cela avait par exemple été initialement le cas de la protéine de couplage de Tn*GBS1* publiée récemment (Brochet *et al*., 2009), qui avait échappé à la détection en raison de restriction sur la longueur. Cependant, l'allègement des filtres pourrait conduire à l'inclusion dans la banque de protéines indésirables et/ou un accroissement considérable du travail d'expertise pour les éliminer.

iii) Cette absence pourrait également être due au fait que la ou les protéines étiquettes pour les ICE et IME de ces génome appartiendraient à des familles ou des sous-familles de protéines non encore identifiées dans les ICE ou IME de firmicutes non détectables avec les protéines références utilisées. Lors de la construction de la banque, les analyses faites pour vérifier la validité des protéines trouvées avaient révélé que certains gènes d'intégrases à tyrosine et/ou de protéines de couplages étaient adjacents à des gènes de relaxase appartenant à des familles jamais identifiées chez les ICE/IME de firmicutes, voire dans des systèmes conjugatifs. Ces relaxases avaient alors été incluses dans la banque Ref1 pour faire le run suivant, ce qui a permis l'identification des relaxases d'IME intégrés dans l'extrémité 3' de *rplL* et du chromosome de *S. agalactiae* A909, qui n'avaient pas été identifiées par Brochet *et al*. (2009).

2) Visualisation des ICE étiquetés par les gènes des protéines caractéristiques, et détection des bornes

Le langage Perl est simple à apprendre. Son utilisation a permis la modification de fichiers texte, de télécharger les fichiers utiles aux analyses, d'appeler et d'exécuter des programmes externes. L'utilisation du logiciel Artémis permet de visualiser rapidement la localisation des différents gènes et des séquences répétées qui pourraient constituer les bornes des éléments. Il permet aussi d'enregistrer les ICE et les IME bornés.

Une fois les 72 génomes analysés, des bornes ont pu être proposées pour 109 ICE et IME. Il était donc intéressant de confronter nos résultats à ceux de la littérature et des bases de données. Ce travail a été fait manuellement.

a. La comparaison des résultats obtenus au cours de ce travail avec le contenu des ICE et des IME présents sur la base de données ICEberg (Bi *et al*., 2012) montre que la procédure utilisée permet de retrouver tous les ICE présents chez les génomes analysés tant que des protéines étiquettes sont clairement identifiées dans ICEberg. Si les ICE et les IME ne possèdent pas de protéines étiquettes connus dans ICEberg, il est difficile pour nous de détecter l'élément.
b. Les résultats obtenus ont également été comparés avec des résultats antérieurs de l'équipe ICE-TeA (Brochet *et al*., 2008) : la procédure mise en place au cours de ce travail permet de repérer tous les ICE et les IME annotés dans cette publication ainsi

que leurs bornes, sauf pour IME_*515_tRNA*^{Lys} où l'analyse des DR reste à valider. De plus, certains des îlots génomiques annotés comme CIME (éléments dérivant d'ICE ou d'IME par délétion) dans la publication de Brochet *et al.* (2008) sur la base de l'absence de gènes de relaxase possédaient en fait des relaxases atypiques et étaient donc des IME. L'étude de Brochet *et al*. (2008), seule recherche systématique d'ICE et IME dans des génomes, bien qu'ayant trouvé de nombreux ICE et IME, avait donc sous-estimé le nombre d'IME présents dans les génomes analysés.

3) Perspectives

3.1 Consolidation et extension de la banque

La base de donnée devra être modifiée afin d'éliminer les anciennes tables et avoir une vue d'ensemble de cette base. Il faudra également y intégrer les informations relatives aux ICE et aux IME découverts après validation par l'expert.

L'enrichissement de la banque devra être poursuivi en élargissant et validant l'analyse sur un ensemble plus large de bactéries. Pour commencer, ce travail sera élargi aux Firmicutes, groupe auxquels les streptocoques appartiennent et dont actuellement 330 génomes sont disponibles. Pour mener à bien l'analyse des génomes de Firmicutes, les filtres appliqués pour Koriblast devront être redéfinis pour prendre en compte la diversité des protéines étiquettes.. En effet, lors de test sur des génomes bactériens de firmicutes pris au hasard sur le site du NCBI, en utilisant comme jeu de protéines de référence l'ensemble des protéines validées de la base de données, et en utilisant les filtres définis pour les 72 génomes de streptocoques étudiés les résultats obtenus présentaient quelques incohérences. En particulier, parmi les gènes de relaxases identifiés dans ces génomes au terme de la procédure ICEKoriBlastP, quelques-uns sont annotés comme codant des protéines de résistance aux antibiotiques. Ceci suggère que ces gènes aient été mal annotés ou encore qu'ils ne codent pas de relaxase et qu'en conséquence le filtre utilisé n'est pas adéquat. Il sera donc nécessaire de resserrer les filtres pour éviter de récupérer des résultats incohérents.

De plus, une recherche systématique de modules de conjugaison chromosomiques, dans plus de mille génomes de bactéries et archées, a révélé qu'un grand nombre de ces modules peuvent appartenir à des ICE et que des gènes de relaxase sans module de conjugaison peuvent appartenir à des IME (Guglielmini *et al*., 2011) . Cependant, cette recherche n'est fructueuse que dans les groupes (principalement les protéobactéries) où un grand nombre de modules de conjugaison (essentiellement plasmidiques) avait déjà été identifiés puis séquencés. Pour les groupes où la conjugaison est mal connue, la recherche est, beaucoup moins fructueuse. Ceci suggère que les comparaisons et/ou analyses de séquences ne détectent des protéines étiquettes (relaxases et protéines de couplage) que si elles sont assez proches de leurs protéines de références. Il est donc important de partir d'un jeu de

protéines étiquettes de référence le plus diversifié possible. Il pourrait par exemple être utile d'inclure des protéines référence provenant d'autres groupes que les firmicutes et surtout de modules conjugatifs de plasmides.

Par ailleurs, il serait intéressant de créer une banque pour un ou des types de protéines du pore de conjugaison présentes dans tous les systèmes de conjugaison, par exemple l'ATPase de famille VirB4. Ceci permettrait de différencier les ICE des IME sur la base de la présence ou l'absence de ces protéines.

3.2 Validation des bornes et recherche d'éléments composites.

Il faut dans un premier temps analyser et valider les DR sélectionnées. En effet 109 ICE/IME bornés représentent 218 bornes potentielles à analyser. Mais il est probable que certaines de ces bornes aient été mal sélectionnées. De plus, la présence de 131 gènes de relaxase suggère que les bornes de 22 éléments pourraient manquer. L'analyse des ICE et des IME a montré que certains DR n'étaient pas localisés à proximité des intégrases. Dans ce cas, la définition de la SeqDRquery pose problème. On pourra réutiliser les DR validées et/ou les sites d'intégration connus par exemple en identifiant des consensus de DR ou la nature du gène d'associé à tel ou tel type d'intégrases. L'analyse de l'ensemble des DR validées permettra de créer une nouvelle base regroupant ces données, par exemple les séquences des DR, leur longueur, la nature du site cible et les caractéristiques de chaque DR, associées aux différentes familles d'intégrases. Par ailleurs, il serait intéressant d'adapter les règles de recherche de répétitions en fonction de la famille d'intégrase et/ou de ce qui est découvert. En effet, les règles suivies étaient basées sur les caractéristiques des IME et ICE s'intégrant de façon site-spécifique grâce à une intégrase à tyrosine. Il faudrait définir des règles de recherche de bornes de l'élément adaptées aux ICE et IME utilisant des intégrases à sérine ou DDE et/ou s'intégrant de façon peu spécifiques.

Dans ce cadre, il faut souligner que les seules analyses systématiques réalisées en ce sens sur *S. thermophilus* et *S. agalactiae* ont révélé de nombreux cas d'accrétion en tandem et que de plus en plus de cas sont décrits en dehors des recherches systématiques (Bellanger *et al*., 2009 ; Pavlovic *et al.,* 2004). Pour cela, des DR devront être recherchées aussi bien en amont qu'en aval des gènes d'intégrases, et il faudra prendre en compte le fait qu'un gène d'intégrase unique puisse être associé à plus de 2 DR dans un même chromosome et/ou que des intégrases apparentées codées par des gènes localisés dans la même région du même génome pourraient avoir des DR plus ou moins similaires.

Dans ce cadre, le cas de *Streptococcus parasanguinis* ATCC 15912 (CP002843) est particulièrement intéressant. En effet, l'étude des éléments portés par le génome de cette souche suggère qu'un ICE et un IME sont intégrés dans un autre IME. Ceci est un fait nouveau car jusqu'à présent l'insertion d'un ICE ou IME dans un autre IME n'avait jamais été décrite. Cependant, les intégrases à tyrosine de ces deux éléments internes (AEH56850 et AEH56866) possèdent une identité protéique de 60%. Habituellement des intégrases à

tyrosine aussi proches ont une spécificité identique ou très similaire, ce qui pourrait suggérer une accrétion en tandem de l'IME et l'ICE au sein de l'IME intégré dans l'extrémité 3' de *rpsI* de *S. parasanguinis* ATCC 15912. Dans ce cas, au moins l'un des types de DR flanquant un des éléments internes aurait été mal identifié et/ou 3 copies valides d'un même DR pourraient être présentes dans cette région.

Bibliographie

Alvarez-Martinez, C.E., and Christie, P.J. (2009). Biological diversity of prokaryotic type IV secretion systems.Microbiol. Mol. Biol. Rev. *73*, 775–808.

Bellanger, X., Roberts, A.P., Morel, C., Choulet, F., Pavlovic, G., Mullany, P., Decaris, B., and Guédon, G. (2009).Conjugative transfer of the integrative conjugative elements ICESt1 and ICESt3 from Streptococcus thermophilus. J. Bacteriol. *191*, 2764–2775.

Bertram, J., Strätz, M., and Dürre, P. (1991).Natural transfer of conjugative transposon Tn916 between gram-positive and gram-negative bacteria. J. Bacteriol. *173*, 443–448.

Bi, D., Xu, Z., Harrison, E.M., Tai, C., Wei, Y., He, X., Jia, S., Deng, Z., Rajakumar, K., and Ou, H.-Y. (2012). ICEberg: a web-based resource for integrative and conjugative elements found in Bacteria. Nucleic Acids Res. *40*, D621–626.

Bresso, E., Ndiaye, B., Smaïl-Tabbone, M., Souchet, M., and Devignes, M.-D. (2011). MODIM: Model-Driven Data Integration for Mining.

Brochet, M., Couvé, E., Glaser, P., Guédon, G., and Payot, S. (2008). Integrative conjugative elements and related elements are major contributors to the genome diversity of Streptococcus agalactiae. J. Bacteriol. *190*, 6913–6917.

Brochet, M., Da Cunha, V., Couvé, E., Rusniok, C., Trieu-Cuot, P., and Glaser, P. (2009). Atypical association of DDE transposition with conjugation specifies a new family of mobile elements. Mol. Microbiol. *71*, 948–959.

Burrus, V., Bontemps, C., Decaris, B., and Guédon, G. (2001). Characterization of a novel type II restriction-modification system, Sth368I, encoded by the integrative element ICESt1 of Streptococcus thermophilus CNRZ368. Appl. Environ. Microbiol.*67*, 1522–1528.

Burrus, V., Pavlovic, G., Decaris, B., and Guédon, G. (2002). Conjugative transposons: the tip of the iceberg. Mol. Microbiol. *46*, 601–610.

Ciric, L., Jasni, A., de Vries, L.E., Agersø, Y., Mullany, P., and Roberts, A.P. (2013).The Tn916/Tn1545Family of Conjugative Transposons.

Cookson, A.L., Noel, S., Hussein, H., Perry, R., Sang, C., Moon, C.D., Leahy, S.C., Altermann, E., Kelly, W.J., and Attwood, G.T. (2011). Transposition of Tn916 in the four replicons of the Butyrivibrio proteoclasticus B316(T) genome. FEMS Microbiol.Lett.*316*, 144–151.

Frost, L.S., Leplae, R., Summers, A.O., and Toussaint, A. (2005). Mobile genetic elements: the agents of open source evolution. Nat. Rev. Microbiol**. *3***, 722–732.

Gogarten, J.P., and Townsend, J.P. (2005).Horizontal gene transfer, genome innovation and evolution.Nat. Rev. Microbiol. ***3***, 679–687.

Guglielmini, J., Quintais, L., Garcillán-Barcia, M.P., de la Cruz, F., and Rocha, E.P.C. (2011). The repertoire of ICE in prokaryotes underscores the unity, diversity, and ubiquity of conjugation. PLoS Genet. ***7***, e1002222.

Haenni, M., Saras, E., Bertin, S., Leblond, P., Madec, J.-Y., and Payot, S. (2010). Diversity and mobility of integrative and conjugative elements in bovine isolates of Streptococcus agalactiae, S. dysgalactiae subsp. dysgalactiae, and S. uberis. Appl. Environ. Microbiol.***76***, 7957–7965.

Holden, M.T.G., Hauser, H., Sanders, M., Ngo, T.H., Cherevach, I., Cronin, A., Goodhead, I., Mungall, K., Quail, M.A., Price, C., et al. (2009). Rapid evolution of virulence and drug resistance in the emerging zoonotic pathogen Streptococcus suis. PLoS ONE ***4***, e6072.

Holm, L., and Sander, C. (1998).Removing near-neighbour redundancy from large protein sequence collections. Bioinformatics ***14***, 423–429.

Hsiao, W.W.L., Ung, K., Aeschliman, D., Bryan, J., Finlay, B.B., and Brinkman, F.S.L. (2005). Evidence of a large novel gene pool associated with prokaryotic genomic islands. PLoS Genet. ***1***, e62.

Langille, M.G.I., and Brinkman, F.S.L. (2009). IslandViewer: an integrated interface for computational identification and visualization of genomic islands. Bioinformatics ***25***, 664–665.

Langille, M.G.I., Hsiao, W.W.L., and Brinkman, F.S.L. (2008).Evaluation of genomic island predictors using a comparative genomics approach.BMC Bioinformatics ***9***, 329.

Lee, C.-C., Chen, Y.-P.P., Yao, T.-J., Ma, C.-Y., Lo, W.-C., Lyu, P.-C., and Tang, C.Y. (2013). GI-POP: a combinational annotation and genomic island prediction pipeline for ongoing microbial genome projects. Gene ***518***, 114–123.

Lu, F., and Churchward, G. (1995). Tn916 target DNA sequences bind the C-terminal domain of integrase protein with different affinities that correlate with transposon insertion frequency. J. Bacteriol. ***177***, 1938–1946.

Mingoia, M., Tili, E., Manso, E., Varaldo, P.E., and Montanari, M.P. (2011). Heterogeneity of Tn5253-like composite elements in clinical Streptococcus pneumoniae isolates. Antimicrob.Agents Chemother.***55***, 1453–1459.

Ochman, H., Lawrence, J.G., and Groisman, E.A. (2000). Lateral gene transfer and the nature of bacterial innovation. Nature ***405***, 299–304.

Pavlovic, G., Burrus, V., Gintz, B., Decaris, B., and Guédon, G. (2004). Evolution of genomic islands by deletion and tandem accretion by site-specific recombination: ICESt1-related elements from Streptococcus thermophilus. Microbiology (Reading, Engl.) ***150***, 759–774.

Rutherford, K., Parkhill, J., Crook, J., Horsnell, T., Rice, P., Rajandream, M.A., and Barrell, B. (2000). Artemis: sequence visualization and annotation. Bioinformatics *16*, 944–945.

Smillie, C., Garcillán-Barcia, M.P., Francia, M.V., Rocha, E.P.C., and de la Cruz, F. (2010).Mobility of plasmids.Microbiol. Mol. Biol. Rev. ***74***, 434–452.

Toussaint, A., and Merlin, C. (2002).Mobile elements as a combination of functional modules. Plasmid ***47***, 26–35.

Waack, S., Keller, O., Asper, R., Brodag, T., Damm, C., Fricke, W.F., Surovcik, K., Meinicke, P., and Merkl, R. (2006).Score-based prediction of genomic islands in prokaryotic genomes using hidden Markov models.BMC Bioinformatics *7*, 142.

Annexes

1) Programme 01signatureICE-F.pl

```
print "avant de commencer verifiz qu'il ny' a AUCUN  join dans le fichier vrifiez qu'il aucun ; dans les descriptif\n";

print" exemple [protein=ddddddd; fffffff] doit être transformé en [protein=ddddddd, fffffff]\n";

print "Si c'est régles sont respecter apuyer sur enter pour continuer";

# voici un exmple de ligne qui doti être utiliser

#
33;relaxase|tr|G7SEB0|G7SEB0_STRSU;618;|AE004092.1_cdsid_AAK33555.1;[gene=SPy_0570] [protein=putative drug resistance protein] [protein_id=AAK33555.1] [location=complement(460288..461655)] ;455;83,10%;7,00E-14;208;23,10%;45,10%;386

# le nombre de |  est important ainsi que le _cdsid_

$ok=<stdin>;

$result = "resultats.csv";

open (rea, "$result");

($Second,$Minute,$Hour,$DayOfMonth,$Month,
$Year,$WeekDay,$DayOfYear,$IsDST)=localtime(time);

$RealMonth=$Month + 1;

$RealYears=$Year+1900;

$date="%02d:%02d:%02d_%02d/%02d/%d",$Hour,$Minute,$Second,$DayOfMonth,$RealMonth,$RealYears;

# création de l'arborésence des résultats avec les résultats avec la date

$DRE="resultats.$DayOfMonth-$RealMonth-$RealYears";

mkdir $DRE;

$drf="01signatureICE-F";

mkdir $drf;

rename ("$drf","$DRE/$drf");

rmdir $drf;

$GLOB= "$DRE/$drf/retrav.csv";

open (reao, ">>$GLOB");

@rea=<rea>;

foreach $lignes(@rea){

$lignes=~s/\"//;

print $lignes;

$lignes=~s/\"//;

print $lignes;

$lignes=~s/\|/\;/;

print $lignes;

$lignes=~s/\|/\;/;

print $lignes;

$lignes=~s/\|/\;/;

print $lignes;

$lignes=~s/\|/\;/;

print $lignes;

$lignes=~s/\|/\;/;

print $lignes;

$lignes=~s/\|/\;/;

print $lignes;

$lignes=~s/\_cdsid\_/;/;

print $lignes;

$lignes=~s/\[gene\=/;/;

print $lignes;

$lignes=~s/\] \[protein\=/;\"/;

print $lignes;

$lignes=~s/\[protein\=/;no;\"/;

print $lignes;

$lignes=~s/\] \[protein_id\=/\";/;

print $lignes;

$lignes=~s/\[protein_id\=/;no;no;/;

print $lignes;

if ($lignes=~/join/){

$lignes=~s/\]
\[location\=complement\(join\(/complement;join;;/;

print$lignes

$lignes=~s/\] \[location\=join\(/;;join;/;

print$lignes;

$lignes=~s/\)\)\]/;/;

print $lignes;

$lignes=~s/\)\]/;/;

print $lignes;

$lignes=~s/(?<=..[0-9])\,(?=[0-9]..)/\;/;

print$lignes;
```

```
}

$lignes=~s/\] \[location\=complement\(/;complement;;/;

print $lignes;

$lignes=~s/\] \[location\=/;;;/;

print $lignes;

$lignes=~s/\.\./;/;

print $lignes;

$lignes=~s/\.\./;/;

print $lignes;

$lignes=~s/\)\]/;;;/;

print $lignes;

$lignes=~s/\]/;;;/;

print $lignes;


print reao "$lignes"

}

close rea;

close reao;

open (reaa, $GLOB);

$genome="$DRE/$drf/genome.csv";

$database="$DRE/$drf/database.csv";

while ($inser=<reaa>){

($pos1,$pos2,$pos3,$pos4,$pos5,$pos6,$pos7,$pos8,$pos9,$pos10,$pos11,$pos12,$pos13,$pos14,$pos15,$pos16,$pos17,$pos18,$pos19,$pos20,$pos21,$pos22,$pos23,$pos24,$pos25,$pos26,$pos27)=split(/;/,$inser);

chomp $pos27;

open (indb,">>$genome");

open( db,">>$database");

 print db "$pos1\;$pos2\;$pos4\;$pos23\;$pos25\;$pos26\;$pos27\;$pos8\;$pos6\;$pos21\n";

print indb "$pos9\;$pos12\;$pos8\;$pos16\;$pos17\;$pos18\;$pos19\;$pos14\;$pos15\;$pos2\;$pos11\n"}

print db"query_number\;query_type\;query_uniprot\;E-value\;identity_percent\;positive_percent\;align_length\;hit_genome_id\;query_length\;hit_length\n";

print indb "FIN\;FIN\;FIN\;FIN\;FIN\;FIN\;FIN\;FIN\;FIN\n";

close reaa;

close db;

close in
```

2)Programme 02visuDR_ICE-FDR.pl

```
#ouverture du fichier genome pour les differents fichier s

use LWP::Simple;

use URI::URL;

$arte=0;

# lancement d'artemis a la fin de l'analyse

print "Voulez vous lancer artemis pour chaque genome? ( n( no) ou y(yes))( respectez la lettres en minuscule : ";

$arte= <stdin>;

chomp $arte;

# Définition de la taille de recherche des DR

print "jusqu'où";

$length=<stdin>;

chomp $length;

# def de l'heure et e la date pour le nom de fichier

($Second,$Minute,$Hour,$DayOfMonth,$Month,
$Year,$WeekDay,$DayOfYear,$IsDST)=localtime(time);

$RealMonth=$Month + 1;

$RealYears=$Year+1900;

$date="%02d:%02d:%02d_%02d/%02d/%d",$Hour,$Minute,$Second,$DayOfMonth,$RealMonth,$RealYears;

# création de l'arborésence des résultats avec les résultats avec la date

$resultats="resultats.$DayOfMonth-$RealMonth-$RealYears";

mkdir $resultats;

# utilisation du fichier de sortie du script 01 ou non

print "utiliser le fichier de sortie de 01 y ou n";

$genn=<stdin>;

chomp $genn;

if $(genn= 'y'){

$genome="$resultats/01signatureICE-F/genome.csv";}

else {print"ou se situe le fichier conteant les info";
```

```
$genome=<stdin>;chomp $genome}

open(INDB,$genome);

$drf="02visuICE_F";

mkdir $drf;

rename ("$drf","$resultats/$drf");

rmdir $drf;

$Listetraite="$resultats/$drf/listegenomeanalyse.txt";

open(ARI,">>$resultats/$drf/listegenomeanalyse.txt");

# def des differente valeur utilisé dans le script

$pos3avant=0;

$pos3apres=0;

$pos7avant=0;

$pos7apres=0;

$pos3bis=0;

$dossierbis=0;

$filebis=0;

# création d'une boulce de lecture ligne par ligne

while ($ligne=<INDB>){

# def des différente position du tableau

# ATTENTION  les differentes positions doivent être dans l'ordre

# pos1 embl_id

# pos2 les notes

# pos3 le genome_id

# pos4 start

# pos5 stop

# $pos6 start stop join

# $pos7 stop join

# pos8 complement ou non

# pos9 JOIN

# pos10 sa fonction

# pos11 locus_tag

($pos1,$pos2,$pos3,$pos4,$pos5,$pos6,$pos7,$pos8,$pos9,$p
os10,$pos11)=split(/;/,$ligne);

# elimination du /n a la fin de la ligne

chomp $pos11;

# création des dossier avec les embl_id

$dossier="$pos3";

if ($pos3 ne $pos3avant)

{ print "\n creation du dossier contenant les résultats\n";

$drt=0;

chomp $drt;

$drtot=0;

mkdir $dossier;

rename("$dossier", "$resultats/$drf/$dossier");

rmdir $dossier}

# recherche du fichier embl corespondant au genome

print "\n recuperation du genome complet au format embl \n"
if ($pos3 ne $pos3avant);

$Lgenome="$resultats/$drf/listegenome.txt";

open(lg,">>$resultats/$drf/listegenome.txt");

# telechargement des différent fichier

$embl=
"http://www.ncbi.nlm.nih.gov/sviewer/viewer.cgi?tool=portal&
sendto=on&log$=seqview&db=nuccore&dopt=gbwithparts&sor
t=&val=".$pos3."&extrafeat=976&fmt_mask=294912&maxplex
=1" if ($pos3 ne $pos3avant);

$embl2="$resultats/$drf/$dossier/$pos3-genome.embl";

$fasta=
"http://www.ncbi.nlm.nih.gov/sviewer/viewer.cgi?tool=portal&
sendto=on&log$=seqview&db=nuccore&dopt=fasta&sort=&val
=".$pos3."&extrafeat=976&fmt_mask=294912&maxplex=1"    if
($pos3 ne $pos3avant);

$fasta2="$resultats/$drf/$dossier/$pos3-fasta.fasta";

if ($pos3 ne $pos3avant) {getstore($fasta,$fasta2);

open(fast ,$fasta2);

@fasta=<fast>;

foreach(@fasta){if (/>/)

{print" le nom e la bacterie est $fasta[0]";

if ($fasta[0]=~/(?<=\| ).*?(?=\,)/){

$nameb=$&;

print "$nameb est son nom\n"}}}};

# integrer les fichier dans leur dossier

# création du fichier en fonction du changement de genome_id

$file="$pos3-$nameb.resultats$DayOfMonth-$RealMonth-
$RealYears" if ($pos3 ne $pos3avant);

print  ARI  "$resultats/$drf/$dossier/$file.embl;".$pos3."\n"  if
($pos3 ne $pos3avant);
```

```
open(out,">>$resultats/$drf/$dossier/$file.embl");

print out "\n     gene          " ;

#ajout de l'info complement ou non $pos8

if ($pos9 eq"join"){

if ($pos8 eq "complement")

{print out  "complement(join($pos4..$pos5,$pos6..$pos7))"}

else {print out  "join($pos4..$pos5,$pos6..$pos7)"}}

else{

if($pos8 eq"complement")

{print out  "complement($pos4..$pos5)"}

else {print out  "$pos4..$pos5"}}

# ajout de l'info de la protéine

print out "\n                /note=$pos2";

# ajout de la fonction $pos8

print out "\n                /function=\"$pos10\"";

print out "\n                /origid=\"$pos1\"";

print out "\n                /locus_tag=\"$pos11\"";

# defintion de la couleur $pos8

if ($pos10 eq "couplage")

{print out "\n                /color=6"}

elsif ($pos10 eq "relaxase")

{print out"\n                /color=7"}

# integrastion a ce niveau de la recherche de DR

else{print out"\n                /color=8";

# ouverture du fichier source

$elem=0;

if ($pos8 eq "complement") {$elem="".$pos4."..".$pos5."";print
"\n putain tu va marcher connard"}

else {$elem="     CDS          ".$pos4."..".$pos5.""}

$i=0;

$max=0;

$findata=$#data;

open (direct,$embl2);

# ouverture du fichier embl conteant les info de genome

# open (teteur,$genome);

# convertir ce fichier en tableau

@data=<direct>;

# definition des posison a cehrcher  elle se presentara
pos3..pos4

print "zone a trouver\n";

print "la position recherche est $elem\n";

# trouvé une ligne d'interet dans tout le fichier si l'integrase est
sur le brin complementaire

if ($pos8 eq "complement") { print "recerche de la prot\éine en
amont de l'integrase";


foreach(@data){if  (/     source        1/)

{$data[$i]=~ /(?<=\.\.).*?(?=\n)/;

$max=$&;

print" taille max du genome est de $max";

$i++}

elsif (/$elem/){

print "ligne".$i."\n lignecomplete $data[$i]";

# amorçage de la remonté du fhichier dans le cas ou integrase
est sur le brin complementaire la suite sera $i++ pour amorcer
la desente

$i--;

# recherche du gene en amont

while($i>-2)

{if ($data[$i] =~/     CDS          complement/)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

if ($data[$i]=~ /(?<=\.\.).*?(?=\))/)

{$st=$&;

print "\n ici la borne a utilise est $st \n";

# changement de la valeur $i pour

$i=-3}}

# cas ou le gene est sur le brin complementaire

elsif ( $data[$i] =~/     CDS          /)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

if ($data[$i]=~ /(?<=\.\.).*?(?=\n)/)

{$st=$&;

print "\n ici la borne a utilise est $st \n";
```

```
# changement de la valeur $i pour

$i=-3}}

elsif ($data[$i] =~/     tRNA          complement/)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

if ($data[$i]=~ /(?<=\.\.).*?(?=\))/)

{$st=$&;

print "\n ici la borne a utilise est $st \n";

# changement de la valeur $i pour

$i=-3}}

# cas ou le gene est sur le brin complementaire

elsif ($data[$i] =~/     tRNA          /)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

# if ($data[$i]=~ /(?<=     CDS          ).*?(?=\.)/)

if ($data[$i]=~ /(?<=\.\.).*?(?=\n)/)

{$st=$&;

print "\n ici la borne a utilise est $st \n";

# changement de la valeur $i pour

$i=-3}}

elsif ($data[$i] =~/     rRNA          complement/)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

# if ($data[$i]=~ /(?<=     CDS          ).*?(?=\.)/)

if ($data[$i]=~ /(?<=\.\.).*?(?=\(\))/)

{$st=$&;

print "\n ici la borne a utilise est $st \n";

# changement de la valeur $i pour

$i=-3}}

# cas ou le gene est sur le brin complementaire

elsif ($data[$i] =~/     rRNA          /)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

# if ($data[$i]=~ /(?<=     CDS          ).*?(?=\.)/)

if ($data[$i]=~ /(?<=\.\.).*?(?=\n)/)

{$st=$&;

print "\n ici la borne a utilise est $st \n";

# changement de la valeur $i pour

$i=-3}}

else{$i--; print "$i\n"}}}

else {$i++}}}

else {

foreach(@data){if    (/          source          1/){$data[$i]=~
/(?<=\.\.).*?(?=\n)/;

$max=$&;

print" taille max du genome $max";

$i++}

elsif (/$elem/){print "ligne".$i."\n lignecomplete $data[$i]";

$findata=$#data;

# amorçage de la remonté du fhichier dans le cas ou integrase
est sur le brin complementaire la suite sera $i++ pour amorcer
la desente

$i++;

# recherche du gne en amont

while($i<=$findata)

{

# cas ou le gene est sur le brin complementaire

if ($data[$i] =~/     CDS          complement/)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

# if ($data[$i]=~ /(?<=     CDS          ).*?(?=\.)/)

if ($data[$i]=~ /(?<=\().*?(?=\.\.)/)

{$st=$&;

print "\n ici la borne a utilise est $staaaaaaaaaa \n";

# changement de la valeur $i pour

$i=$findata+100}}

elsif ( $data[$i] =~/     CDS          /)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

# if ($data[$i]=~ /(?<=     CDS          ).*?(?=\.)/)

if ($data[$i]=~ /(?<=     CDS          ).*?(?=\.\.)/)

{$st=$&;
```

```
print "\n ici la borne a utilise est $st \n";

# changement de la valeur $i pour

$i=$findata+100}}

elsif ($data[$i] =~/     tRNA          complement/)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

# if ($data[$i]=~ /(?<=     CDS          ).*?(?=\.)/)

if ($data[$i]=~ /(?<=\().*?(?=\.\.)/)

{$st=$&;

print "\n ici la borne a utilise est $st \n";

# changement de la valeur $i pour

$i=$findata+100}}

elsif($data[$i] =~/     tRNA          /)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

# if ($data[$i]=~ /(?<=     CDS          ).*?(?=\.)/)

if ($data[$i]=~ /(?<=     tRNA          ).*?(?=\.\.)/)

{$st=$&;

print "\n ici la borne a utilise est $st \n";

# changement de la valeur $i pour

$i=$findata+100}}

# cas ou le gene est sur le brin complementaire

elsif ($data[$i] =~/     rRNA          complement/)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

# if ($data[$i]=~ /(?<=     CDS          ).*?(?=\.)/)

if ($data[$i]=~ /(?<=\().*?(?=\.\.)/)

{$st=$&;

print "\n ici la borne a utilise est $st \n";

# changement de la valeur $i pour

$i=$findata+100}}

elsif($data[$i] =~/     rRNA/)

{print "\n la nouvelle".$i." \n et ça ligne complet $data[$i] \n";

# selection de la position stop uniquement

# if ($data[$i]=~ /(?<=     CDS          ).*?(?=\.)/)

if ($data[$i]=~ /(?<=     rRNA          ).*?(?=\.\.)/)

{$st=$&;

print "\n ici la borne a utilise est $st \n";

# changement de la valeur $i pour

$i=$findata+100}}

# cas ou le gene est sur le brin complementaire

else{$i++; print "$i\n";if ($i==$findata) {print "pas de protenavallafinde recherche sera la taille max du genome";

$st=$max}}}}

else {$i++}}}

if ($pos8 eq "complement"){

$sp=$st-30;

$ss=$st-$length;

if ($sp<=0){$sp=1}

if ($ss<=0){$ss=1}

$hit=$sp+$length;

$binte=$pos4+30;

print "$st jusque $sp \n";

if($hit>$max) {$hit=$max}

if ($binte>$max){$binte=$max}

# ALORS configuration de bl2seq -i et -j permette d'identifier le fichier d'entré -I et -J permet e selectioner les zone pour le blast

# -r et-q permet de definir les penalité pour le match ou mismatch du blast -G -E les penatlité d'ouverture et d'extension des gap

# -e la evalue max ( je l'ai detuite des observation de plusieur blast) -W la longueur mini d'allignement

$blast2seq="bl2seq.exe -i\"".$fasta2."\" -I".$ss.",".$hit." -j\"".$fasta2."\" -J".$sp.",".$binte." -pblastn -r4 -q-5 -G12 -E8 -e0.5 -W8 -o out.txt -D1";

$blast3seq="bl2seq.exe -i\"".$fasta2."\" -I".$ss.",".$hit." -j\"".$fasta2."\" -J".$sp.",".$binte." -pblastn -r4 -q-5 -G12 -E8 -e0.5 -W8 -o out3.txt";

# bl2seq.exe -i"resultats.21-4-2013\CP000407-Streptococcus-suis-05ZYH33\CP000407-fasta.fasta" -I93816,393816 -j"resultats.21-4-2013\CP000407-Streptococcus-suis-05ZYH33\CP000407-fasta.fasta" -J93832,93948 -pblastn -r4 -q-5 -G12 -E8 -e0.5 -W8 -o testvisu2.txt -D1

print $blast2seq;

system ($blast2seq);

system ($blast3seq);

$dr="out.txt";
```

```
$dr2="$resultats/$drf/$dossier/".$pos1."DR.txt";

open(dr,$dr);

open(dr2,">>$dr2");

$dr3="out3.txt";

$dr4="$resultats/$drf/$dossier/".$pos1."DRIrapport.txt";

open(dr3,$dr3);

open(dr4,">>$dr4");

while ($LLL=<dr>){ print dr2 $LLL}

while ($LLL3=<dr3>){ print dr4 $LLL3}

if ($pos3 ne $pos3avant) {$concaav=$conca;

$conca=" ";

$conca="$conca \+ $dr2"}

else {$conca="$conca \+ $dr2"}}

else {

# construction de la seqDRquery

$sp=$st+30;

$ss=$st+$lengh;

if ($sp>$max){$sp=$max}

if ($ss>$max){$ss=$max}

$hit=$sp-$length;

$binte=$pos5-30;

print "dimention de la recherche $st jusque $sp \n";

if($hit<0) {$hit=0}

if ($binte<0){$binte=1}

$blast2seq="bl2seq.exe -i\"".$fasta2."\" -I".$hit.",".$ss." -j\"".$fasta2."\" -J".$binte.",".$sp." -pblastn -r4 -q-5 -G12 -E8 -e0.5 -W8 -o out.txt -D1";

$blast3seq="bl2seq.exe -i\"".$fasta2."\" -I".$hit.",".$ss." -j\"".$fasta2."\" -J".$binte.",".$sp." -pblastn -r4 -q-5 -G12 -E8 -e0.5 -W8 -o out3.txt";

print $blast2seq;

system ($blast2seq);

system ($blast3seq);

$dr="out.txt";

$dr2="$resultats/$drf/$dossier/".$pos1."DR.txt";

$dr3="out3.txt";

$dr4="$resultats/$drf/$dossier/".$pos1."DRIrapport.txt";

open(dr,$dr);

open(dr2,">>$dr2");

open(dr3,$dr3);

open(dr4,">>$dr4");

while ($LLL=<dr>){ print dr2 $LLL}

while ($LLL3=<dr3>){ print dr4 $LLL3}

if ($pos3 ne $pos3avant) {$concaav=$conca;

$conca=" ";

$conca="$conca \+ $dr2"}

else {$conca="$conca \+ $dr2"}}}

$pos7apres=$pos3;

$pos3avant=$pos3;

# execution d'artemis si la condition le permet

# print "$arte-$pos7avant-$pos7apres-" ;

if ($pos7avant eq 0)

{$pos7avant=$pos3;

$pos3bis=$pos3;

$dossierbis=$dossier;

$filebis=$file;

print " \n modification des valeurs d'entrées \n"}

elsif( $pos7avant ne $pos7apres)

{print "\n fin d'ecriture du fichier resultat. \n fin d'ecriture du fichier multifasta.\n";

print "\n changement de localisation génétique";

if($arte eq "y")

{print "\n lancement Artémis ";

print"\n ERREUR DOSSSIER";

print "\n changement de dossier chargement du dossier anterieur";

print"\n input ok lancement d'Artemis\n";

$pid = fork();

if($pid==0)

{$artemis="artemis_v6.jar/C:/Users/yann/Desktop/BDvsartemishmm/".$resultats."/".$drf."/\"".$dossierbis."\"/".$pos3bis."-fasta.fasta + C:/Users/yann/Desktop/BDvsartemishmm/".$resultats."/".$drf."/\"".$dossierbis."\"/".$pos3bis."-genome.embl + C:/Users/yann/Desktop/BDvsartemishmm/".$resultats."/".$drf."/\"".$dossierbis."\"/\"".$filebis."\".embl ".$concaav."";
```

```
# + C:/Users/yann/Desktop/BDvsartemishmm/".$resultats."/".$drf."/\"".$dossierbis."\"/".$pos3bis."-analysegff.gff";

system ($artemis);

exit;}

$concaav=" ";

$pos7apres=$pos3;

$pos7avant=$pos3;

$pos3bis=$pos3;

$dossierbis=$dossier;

$filebis=$file}

else {print " \n annulation du lancement d'artemis par l'utilisateur ";

print "000000000 passage au genome suivant000000000000000".$pos3."bbbbb".$pos7avant."bbbbb".$pos7apres."\n"}}

else {print " \n pas de modification de la localisation genomique";

print "\n passage a la ligne autorise\n"}}
```

3) programme 03content_ ICE-FDR

```
use LWP::Simple;

use URI::URL;

print " voulez vous ouvrir un fichier contenant les résultats (y) \n ou voulez vous entrainer chaque fichier contenant les résultats (n)";

$rep1=<stdin>;

chomp $rep1;

($Second,$Minute,$Hour,$DayOfMonth,$Month,
$Year,$WeekDay,$DayOfYear,$IsDST)=localtime(time);

$RealMonth=$Month + 1;

$RealYears=$Year+1900;

$date="%02d:%02d:%02d_%02d/%02d/%d",$Hour,$Minute,$Second,$DayOfMonth,$RealMonth,$RealYears;

# création de l'arborésence des résultats avec les résultats avec la date

$DRE="resultats.$DayOfMonth-$RealMonth-$RealYears";

mkdir $DRE;

$drf="03content_ ICE-FDR";

mkdir $drf;

rename ("$drf","$DRE/$drf");

rmdir $drf;

$drg="xml";

mkdir $drg;

rename ("$drg","$DRE/$drf/$drg");

rmdir $drg;

$drh="gbk";

mkdir $drh;

rename ("$drh","$DRE/$drf/$drh");

rmdir $drh;

if ($rep1 eq "y"){

print "ou se situe le fichier enssemble( ecrire tout le chemin )\n";

$rep2=<stdin>;

chomp $rep2;

open (resl1 ,$rep2);

while ($ligne=<resl1>){

# $pos1 est le chemins de destination

# $pos2 est le nom du génome

($pos1,$pos2)=split(/;/,$ligne);

print"fichier liste ouvert\n";

print "$ligne\n";

chomp $pos2;

open(resl2, $pos1);

$nm=1;

@resl=<resl2>;

$i=0;

foreach (@resl){

if (/FT   misc_feature    /){

print"ice trouvé $resl[$i]\n";

$findata=$#data;

# selection des coodronné de l'ice
```

```
if($resl[$i]=~/(?<=FT misc_feature complement\().*?(?=\.\.)/){print"start $&\n $resl[$i]\n";$start=$&}

elsif($resl[$i]=~/(?<=FT misc_feature ).*?(?=\.\.)/){print"start $&\n $resl[$i]\n";$start=$&} else {print "$resl[$i]"}

if($resl[$i]=~/(?<=\.\.).*?(?=\))/){$stop=$&;print"start $&\n $resl[$i]\n"}

elsif($resl[$i]=~/(?<=\.\.).*?(?=\n)/){$stop=$&;print"start $&\n $resl[$i]\n"}

$sav="http://www.ncbi.nlm.nih.gov/sviewer/viewer.cgi?tool=portal&sendto=on&log$=seqview&db=nuccore&dopt=gbwithparts&sort=&val=".$pos2."&from=".$start."&to=".$stop."&extrafeat=976&maxplex=1";

$sav2= "$DRE/$drf/$drh/".$pos2."-".$nm.".gbk";

getstore ($sav,$sav2);

$sav5="http://www.ncbi.nlm.nih.gov/sviewer/viewer.cgi?tool=portal&sendto=on&log$=seqview&db=nuccore&dopt=gbc_xml&sort=&val=".$geno."&from=".$start."&to=".$stop."&extrafeat=976&fmt_mask=424&maxplex=1";

$sav6= "$DRE/$drf/$drg/".$pos2."-".$nm.".xml";

getstore ($sav5,$sav6);

$i++;

$nm++;}

else {$i++}}}}

else{

print "ou se situe le fichier ( ecrire tout le chemin )\n";

$rep3=<stdin>;

print"de quel genome il est question\n";

$geno=<stdin>;

chomp $geno;

open (resl3 ,$rep3);

$nm=1;

@resl=<resl3>;

$i=0;

foreach (@resl){

if (/FT misc_feature /){

print"ice trouvé $resl[$i]\n";

$findata=$#data;

# selection des coodronné de l'ice

if($resl[$i]=~/(?<=FT misc_feature complement\().*?(?=\.\.)/){print"start $&\n $resl[$i]\n";$start=$&}

elsif($resl[$i]=~/(?<=FT misc_feature ).*?(?=\.\.)/){print"start $&\n $resl[$i]\n";$start=$&} else {print "$resl[$i]"}

if($resl[$i]=~/(?<=\.\.).*?(?=\))/){$stop=$&;print"start $&\n $resl[$i]\n"}

elsif($resl[$i]=~/(?<=\.\.).*?(?=\n)/){$stop=$&;print"start $&\n $resl[$i]\n"}

$sav="http://www.ncbi.nlm.nih.gov/sviewer/viewer.cgi?tool=portal&sendto=on&log$=seqview&db=nuccore&dopt=gbwithparts&sort=&val=".$geno."&from=".$start."&to=".$stop."&extrafeat=976&maxplex=1";

$sav2= "$DRE/$drf/$drh/".$geno."-".$nm.".gbk";

getstore ($sav,$sav2);

$sav5="http://www.ncbi.nlm.nih.gov/sviewer/viewer.cgi?tool=portal&sendto=on&log$=seqview&db=nuccore&dopt=gbc_xml&sort=&val=".$geno."&from=".$start."&to=".$stop."&extrafeat=976&fmt_mask=424&maxplex=1";

$sav6= "$DRE/$drf/$drg/".$geno."-".$nm.".xml";

getstore ($sav5,$sav6);

$i++;

$nm++;}

else {$i++}}}
```

Développement d'un outil de recherche automatisée des éléments génétiques mobiles de type ICE dans les génomes bactériens (ICE-Finder)

Les ICE (« integrative conjugating elements ») et les IME (« integrative mobilizable elements ») sont des éléments importants du transfert horizontal de gènes. Aujourd'hui, il existe des méthodes pour détecter des éléments génétiques mobiles mais aucune n'est spécialisée dans la recherche d'ICE ou d'IME. Ce stage a été effectué dans le cadre d'un projet collaboratif visant à créer une méthode informatisée et automatisée de détection des ICE et IME dans les génomes bactériens. La méthodologie se base sur l'identification et le positionnement dans les génomes de gènes codant 3 classes de protéines caractéristiques de ces éléments (les protéines de couplage, les relaxases et les intégrases) et sur la création et l'enrichissement d'une base de données rassemblant et connectant les données entre elles. Au cours du stage, un programme prototype (ICEFinder) de recherche automatique d'ICE/IME a été développé afin de réaliser les tâches suivantes :

(1) Détecter les gènes codant les protéines d'intérêt dans des génomes bactériens
(2) Positionner ces gènes sur les génomes
(3) Rechercher et localiser les bornes potentielles des ICE/IME
(4) Récupérer le contenu en gènes de ces élements génétiques mobiles

L'exécution de ce programme, sur des génomes complets de streptocoques, a permis de caractériser et de borner 109 ICE/IME présents dans 49 des 72 génomes étudiés.

Development of a tool for automatic search of mobile genetic elements ICE in bacterial genomes (ICE-Finder)

ICE ("integrative conjugating elements") and IME ("integrative mobilizable elements") are important elements of horizontal gene transfer. Today, there are methods to detect mobile genetic elements but none is specialized in the research of ICE or IME. This training was conducted in a collaborative project to create a computerized and automated method for the detection of ICE and IME in bacterial genomes. The methodology is based on identifying and positioning in the genomes of three classes of genes encoding proteins which are specific of these elements (coupling protein, relaxase , integrase) and development of a database gathering data on these genes. During the study, a prototype program (ICEFinder) for automatic search of ICE / IME has been developed to perform the following tasks:

(1) Detect the genes encoding the proteins of interest in bacterial genomes

(2) Locate the genes on the genome

(3) Search and locate potential terminals ICE / IME

(4) Get the gene content of these mobile genetic elements

The execution of this program on the whole genomes of streptococci has been used to characterize and limit 109 ICE / IME in 49 of the 72 genomes studied.