N° d'ordre 236-2006 Année 2006

THÈSE

Présentée

devant l'université Claude Bernard Lyon-1

pour l'obtention

du DIPLÔME DE DOCTORAT

(arrêté du 25 avril 2002)

présentée et soutenue publiquement le 4 décembre 2006

par

Anne-Muriel ARIGON

Développements d'outils pour l'aide à l'identification dans les grandes banques de familles de gènes.

Directeurs de thèse : Manolo GOUY et Guy PERRIÈRE

JURY :	Jean-Pierre FLANDROIS	Président
	Richard CHRISTEN	Rapporteur
	Christine FROIDEVAUX	Rapporteur
	Yves VAN DE PEER	Examinateur
	Manolo GOUY	Directeur
	Guy PERRIÈRE	Directeur

UNIVERSITÉ CLAUDE BERNARD-LYON 1

Président de l'Université	**M. le professeur L. COLLET**
Vice-Président du Conseil Scientifique	M. le professeur J. F. MORNEX
Vice-Président du Conseil d'Administration	M. le professeur R. GARRONE
Vice-Président des Etudes et de la Vie Universitaire	M. le professeur G. ANNAT
Secrétaire Génèral	M. G. GAY

SECTEUR SANTE

Composantes

UFR de Médecine Lyon R.T.H. Laënnec	Directeur : M. le Professeur D. VITAL-DURAND
UFR de Médecine Lyon Grange-Blanche	Directeur : M. le Professeur X. MARTIN
UFR de Médecine Lyon-Nord	Directeur : M. le Professeur F. MAUGUIERE
UFR de Médecine Lyon-Sud	Directeur : M. le Professeur F.N. GILLY
UFR d'Ontologie	Directeur : M. O. ROBIN
Institut des Sciences Pharmaceutiques et Biologiques	Directeur : M. le Professeur F. LOCHER
Institut Techniques de Réadaptation	Directeur : M. le Professeur MATILLON
Département de Formation et Centre de Recherche en Biologie Humaine	Directeur : M. le Professeur P. FARGE

SECTEUR SCIENCES

Composantes

UFR de Physique	Directeur : M. le Professeur J. L. VIALLE
UFR de Biologie	Directeur : M. le Professeur H. PINON
UFR de Mécanique	Directeur : M. le Professeur H. BEN HADID
UFR de Génie Electrique et des Procédés	Directeur : M. le Professeur A. BRIGUET
UFR de Sciences de la Terre	Directeur : M. le Professeur P. HANTZPERGUE
UFR de Mathèmatique	Directeur : M. le Professeur M. CHAMARIE
UFR d'Informatique	Directeur : M. le Professeur M. EGEA
UFR de Chimie Biochimie	Directeur : M. le Professeur J. P. SCHARFF
UFR de STAPS	Directeur : M. le Professeur R. MASSARELLI
Observatoire de Lyon	Directeur : M. le Professeur R. BACON
Institut des Sciences et des Techniques de l'Ingénieur de Lyon	Directeur : M. le Professeur J. P. PUAUX
DÈpartement de premier cycle Sciences	Directeur : M. J. C. DUPLAN Maître de conférences
IUT A	Directeur : M. le Professeur M. ODIN
IUT B	Directeur : M. le Professeur G. MAREST
Institut de Science Financière et d'Assurances	Directeur : M. le Professeur J. C. AUGROS

REMERCIEMENTS

Je tiens tout d'abord à remercier Manolo et Guy qui m'ont permis de réaliser cette thèse et de pouvoir travailler dans le domaine de la bioinformatique. Merci pour votre confiance et pour avoir toujours été là afin de me guider, me soutenir et me permettre de mener ces travaux de thèse à leur terme. J'ai vraiment eu plaisir à travailler avec vous.

Je souhaite également remercier les directeurs ancien et actuel du LBBE, Christian et Dominique, pour m'avoir accueilli au sein du laboratoire pendant ces 3 années de thèse, ainsi qu'à Misou, Nathalie et Agnès qui m'ont aidé à gérer toutes les démarches administratives, toujours dans la bonne humeur.

Je remercie les membres de mon jury de thèse (Richard Christen, Jean-Pierre Flandrois, Christine Froideveaux et Yves Van de Peer) qui ont accepté d'examiner mon travail et plus particulièrement Richard Christen et Christine Froideveaux qui ont accepté le rôle de rapporteurs de mon mémoire.

J'aimerais aussi remercier les membres du laboratoire, en particulier ceux de l'équipe BGE et BAOBAB qui m'ont aidé au cours de ma thèse ainsi que tous ceux que j'ai croisé pendant ces 3 ans et qui m'ont permis d'avancer dans mon travail. Un grand merci plus particulièrement à mes « co-bureaux » anciens et actuels, aussi bien dans le « débarras », la bibliothèque (pendant les multiples catastrophes) et maintenant dans les nouveaux locaux. Merci pour votre aide, votre soutien, cette bonne ambiance et pour les dégustations de chocolat et autres gâteaux !

Je souhaite également remercier Jérôme et Ricco avec qui j'ai travaillé pendant mes 3 années de monitorat. J'ai pu ainsi découvrir le métier d'enseignant dans une ambiance très conviviale. Merci pour votre aide ainsi que pour tous vos conseils qui m'ont été très utiles. Merci également aux membres de ERIC et particulièrement aux thèsards (Cécile, Gaëlle, Hadj,...) qui m'ont aidé pendant ces 3 ans et avec qui j'ai passé de bons moments.

Merci aussi à Maryvonne et Anne avec qui j'ai eu le plaisir de travailler à nouveau et qui m'ont permis de finaliser mon travail de DEA. Merci pour votre aide, votre soutien et vos conseils.

Je voudrais remercier mes parents, Jérôme, Anh et Emma qui m'ont soutenu et encouragé tout au long de ma thèse et qui ont eu la patience de relire toutes ces pages. Merci également à tous mes amis qui m'ont entourée et soutenue.

Enfin, un grand merci tout particulièrement à Benoît qui a toujours été là, même aux heures les plus avancées de la nuit, pour relire mon travail et écouter mes nombreuses et interminables répétitions de soutenance.

Merci à tous... !

Table des matières

Résumé **iv**

Abstract **vii**

Introduction **ix**

1 Le contexte bioinformatique **1**

1 La génétique et la génomique 1

1.1 Un peu d'histoire 1

1.2 L'information génétique 3

1.2.1 La cellule 3

1.2.2 L'ADN 4

1.2.3 Les chromosomes 6

1.2.4 Les gènes et la synthèse protéique 7

1.2.4.1 La réplication 8

1.2.4.2 La transcription 9

1.2.4.3 La traduction 10

1.3 L'évolution des séquences 11

2 Le stockage de données biologiques 12

2.1 Les banques généralistes 13

2.2 Les banques spécialisées 13

3 L'analyse de séquences 15

3.1 Comparaison de séquences, recherche de similarité et alignement par paire 15

3.1.1 Le principe 16

3.1.2 Les matrices de similarité 17

3.1.3 Les algorithmes et les programmes 19

3.2 Les alignements multiples 22

3.2.1 Les méthodes d'alignement progressif 22

3.2.2 D'autres approches 24

3.2.3 Les différents formats d'alignement 24
3.3 Les arbres phylogénétiques 25
3.3.1 Les différentes méthodes 26
3.3.2 Enraciner un arbre 28
3.3.3 Evaluer la qualité d'un arbre 29
3.3.4 Les formats d'arbres 30

2 L'identification **31**
1 La taxonomie 31
1.1 Une brève histoire 31
1.2 Les classifications 32
1.2.1 La classification phénotypique 32
1.2.2 La classification phylogénétique 32
2 L'identification de séquences 33
2.1 Définition 33
2.2 Les méthodes d'identification 33
2.2.1 L'approche phénotypique 33
2.2.2 L'approche moléculaire 34
2.2.2.1 L'hybridation moléculaire 35
2.2.2.2 L'amplification génique ou PCR ("polymerase chain reaction") 36
2.2.2.3 Le séquençage 38
2.2.2.4 Les codes barres d'ADN ("DNA barcode") 39
3 Les outils bioinformatiques existants 40
3.1 BIBI, MitALib et PhyID/CD 40
3.2 RDP II 42
3.3 RIDOM 42
3.4 MicroSeq 42
3.5 TaxI 43
4 Les motivations 43

3 L'outil d'identification développé : HoSeqI **47**
1 Les objectifs 47
2 L'accès aux données 48
2.1 Les banques utilisées 48
2.2 L'outil d'interrogation 49
3 Le principe et le choix des méthodes à utiliser 50
3.1 La recherche de la famille à laquelle appartient la séquence requête . 51
3.1.1 La recherche de similarité 51
3.1.2 L'identification de la famille 53
3.2 L'alignement de la séquence requête avec les séquences de la famille 54
3.3 La reconstruction phylogénétique 58

4 L'implémentation . . . 62
4.1 L'architecture . . . 62
4.2 Les langages utilisés . . . 62
4.3 L'algorithme d'HoSeqI . . . 63
4.3.1 Le premier module principal . . . 65
4.3.2 Le module secondaire permettant le choix des listes de programmes d'alignements et de phylogénies . . . 67
4.3.3 Le module secondaire permettant de lancer en différé sur le serveur le calcul de l'alignement et la reconstruction phylogénétique . . . 69
4.3.4 Le deuxième module principal . . . 70
4.3.5 Le troisième module principal . . . 70
4.4 L'interface web et la présentation des résultats . . . 70
5 Conclusions . . . 75

4 Deux utilisations et applications des méthodes développées dans HoSeqI 77
1 L'ajout des séquences de génomes aux banques de familles de gènes homologues . . . 77
1.1 Les bactéries du genre *Frankia* . . . 78
1.2 La détection de transferts horizontaux chez les bactéries . . . 79
1.3 Une application basée sur les modules de l'application Web HoSeqI pour l'ajout de séquences de *Frankia* à HOGENOM . . . 81
1.3.1 Le principe . . . 81
1.3.2 L'algorithme . . . 82
1.4 Les traitements des résultats . . . 83
1.5 Conclusions . . . 86
2 L'identification automatique de séquences d'ARNr 16S de bactéries . . . 92
2.1 Les séquences d'ARNr 16S de bactéries . . . 92
2.2 Les séquences chimères . . . 93
2.3 Un outil basé sur des modules d'HoSeqI . . . 94
2.3.1 La base de données utilisée . . . 94
2.3.2 Le principe . . . 94
2.3.2.1 Le repérage de séquences chimères . . . 95
2.3.2.2 L'identification . . . 95
2.3.3 L'algorithme . . . 96
2.3.3.1 La détection de chimères . . . 97
2.3.3.2 L'identification de séquences d'ARNr 16S . . . 101
2.4 Conclusions . . . 104

Conclusions et perspectives 105

Références bibliographiques 107

Annexe A

Résultats de la détection d'éventuels transferts horizontaux chez *Frankia*, par l'analyse phylogénétique **117**

Annexe B

Résultats de la détection d'éventuels transferts horizontaux chez *Frankia*, par l'analyse liée à l'absence de gènes chez *Streptomyces* **139**

Annexe C

Articles **161**

1 Article1 : Bioinformatic sequence identification from sequence family databases 162

2 Article2 : HoSeqI : automated homologous sequence identification in gene family databases 167

Résumé

Le nombre de séquences génomiques disponibles augmente très vite du fait du développement de méthodes de séquençage massif. La classification de ces séquences est nécessaire et permet l'étude de leurs relations évolutives. Des outils bioinformatiques automatisés sont indispensables pour effectuer ces opérations d'identification de façon précise et rapide. Nous avons développé HoSeqI (Homologous Sequence Identification), un système permettant d'automatiser l'identification de séquences dans de grandes banques de familles de gènes homologues. HoSeqI propose une interface accessible sur internet (http ://pbil.univ-lyon1.fr/software/HoSeqI/) afin d'identifier une séquence et de visualiser l'alignement et la phylogénie obtenus. Un autre programme, dérivé d'HoSeqI, a été implémenté pour l'ajout automatique de séquences génomiques aux banques de familles. Enfin, un travail sur l'identification automatique de séquences bactériennes d'ARN 16S et la détection de séquences chimères a été effectué.

Mots-clés : Identification automatique, similarité, alignement, phylogénie, chimère.

Abstract

The number of available genomic sequences is growing very fast, due to the development of massive sequencing techniques. Sequence classification is needed and contributes to the assessment of their evolutionary relationships. Automated bioinformatics tools are thus necessary to carry out these identification operations in an accurate and fast way. We developed HoSeqI (Homologous Sequence Identification), a software environment allowing this kind of automated sequence identification using homologous gene family databases. HoSeqI is accessible through a Web interface (http ://pbil.univ-lyon1.fr/software/HoSeqI/) allowing to identify a sequence and to visualize obtained alignment and phylogeny. We also implemented another set of programs - derivated from the one used in HoSeqI - in order to automatically add genomic sequences to family databases. At last, some developments aimed at automated identification of 16S RNA bacterial sequences and detection of chimeric sequences.

Keywords : Automated identification, similarity, alignment, phylogeny, chimera.

Introduction

Aujourd'hui, le nombre d'espèces connues (entre 1,5 et 1,8 millions) n'est pas déterminé avec exactitude à cause des difficultés liées à la notion d'espèce. De plus, personne ne sait exactement combien d'espèces existent et certains estiment ce nombre entre 8 et 12 millions (Wikipédia, 2006). L'étendue de la diversité des espèces vivantes n'est donc que partiellement connue et de nombreuses espèces restent encore à identifier.

Le principe de l'identification consiste à déterminer à quel groupe d'espèces, un organisme analysé se rattache. Les méthodes traditionnelles d'identification utilisent la morphologie et d'autres caractères phénotypiques pour différencier les organismes. Cependant, ces techniques ne permettent pas systématiquement d'avoir une identification exacte et rapide, et, dans certains cas, on ne dispose pas de caractères phénotypiques pour identifier toutes les espèces. Le développement des méthodes moléculaires a permis de palier les limites de l'approche phénotypique en comparant les séquences d'ADN des organismes étudiés à celles d'espèces connues. Ces méthodes permettent de définir l'espèce d'une séquence analysée sur la base de relations de parenté (relations phylogénétiques).

Au cours des dix dernières années, le séquençage d'ADN à grande échelle a permis d'augmenter considérablement la quantité de données biologiques disponibles. L'identification moléculaire manuelle de séquences devient alors trop longue et pénible à réaliser pour de grands jeux de données. Il est donc nécessaire de mettre à disposition des biologistes, des outils bioinformatiques afin de traiter automatiquement ces données. Par exemple, un projet international réunissant un grand nombre d'organisations de différents pays (Consortium Barcode of Life) a été mis en place dans le but d'appliquer le concept de code barre génétique afin d'accélérer considérablement le rythme de l'inventaire de la biodiversité, ainsi que de permettre à quiconque d'identifier des organismes sans ambiguïté.

Les outils dédiés à l'identification doivent utiliser des méthodes spécifiques aux données analysées. L'objectif de ces travaux de thèse consiste à développer des outils permettant l'identification automatique de séquences dans les grandes banques de familles de gènes homologues. Nous avons également travaillé sur l'identification bactérienne et la détection automatique de séquences chimères.

Ce document se compose de quatre parties. Dans la première, nous définissons le contexte bioinformatique en introduisant les différentes notions de génétique et génomique

puis nous expliquons comment il est possible de stocker et analyser des données biologiques. Dans la deuxième partie, nous détaillons les différentes méthodes d'identification ainsi que les outils existants. La troisième partie est consacrée à l'outil bioinformatique développé, HoSeqI (Homologous Sequence Identification), qui permet l'identification automatique de séquences dans de grandes banques de familles de gènes homologues. Enfin, dans la quatrième partie, nous décrivons deux autres applications basées sur les méthodes utilisées dans HoSeqI : la première permet d'ajouter automatiquement toutes les séquences de génomes à une banque de famille de gènes ; la deuxième est dédiée à l'identification bactérienne et intègre un module de détection automatique de séquences chimères.

Chapitre 1

Le contexte bioinformatique

La bioinformatique est une discipline récente qui consiste à analyser l'information biologique, essentiellement sous la forme de séquences biologiques et de structures de protéines. Deux de ses principaux objectifs sont l'identification des gènes et la prédiction de leur fonction, deux problèmes au centre de la génomique. La bioinformatique est constituée en partie par l'ensemble des concepts et des techniques nécessaires à l'interprétation de l'information génétique (séquences) et structurale (repliement 3-D). Cette discipline est donc une branche théorique de la biologie. Son but est d'effectuer la synthèse des données disponibles à l'aide de modèles et de théories, d'énoncer des hypothèses généralisatrices, sur l'évolution des espèces par exemple, et de formuler des prédictions, comme la fonction ou la localisation de gènes. Elle s'appuie, bien sûr, à la fois sur les concepts de la biologie et de l'informatique, mais également sur des outils issus de la chimie et de la physique.

1 La génétique et la génomique

1.1 Un peu d'histoire

Cette section a été réalisée en assemblant des informations provenant de (Bernot et Alibert), (Gayon) et (Dardel, 2002).

La génétique, étude de l'hérédité, est une discipline récente de la biologie. La génétique moderne remonte aux travaux de Mendel, qui le premier établit les lois de l'hérédité. Il publia ses résultats en 1866, mais ils passèrent alors à peu près inaperçus (Mendel, 1866; Mendel, 1961). Mendel comprend qu'un caractère héréditaire peut exister sous différentes versions – les allèles –, les unes dominantes, les autres récessives. Il énonce les lois de la transmission de certains traits héréditaires. L'élément céllulaire responsable de cette transmission est le gène (section 1.2.4). Les gènes se transmettent entre les générations et fonctionnent de manière indépendante.

En 1900, trois botanistes, De Vries, Correns et Tschermak, "redécouvrent" les lois de l'hybridation de Mendel. En moins d'une décennie, une science nouvelle est fondée sur cette base, Bateson la nomme "génétique". Dans le milieu des années 1910, ce sont les travaux de

Morgan, sur la drosophile, qui conduisent au développement de la théorie chromosomique de l'hérédité. Les gènes sont alors localisés sur les chromosomes (section 1.4), et les travaux de Sturtevant, permettent d'ordonner les gènes le long des chromosomes, constituant les premières cartes génétiques. Cependant, jusqu'au milieu du $XX^{ème}$ siècle, les gènes, bien que localisés sur les chromosomes, demeurent des entités théoriques. Leurs caractères physiques, leur nature matérielle, tout comme leur mode d'action demeurent mystérieux. Le support matériel de l'hérédité, l'acide désoxyribonucléique, ou ADN (section 1.3), fut identifié comme tel en 1944. La structure en double hélice de l'ADN est découverte par Watson et Crick en 1953. Au début des années 1960, Jacob et Monod (Jacob & Monod, 1961) découvrent le modèle de régulation de la production des protéines chez les bactéries et virus. L'ADN est transcrit en ARN messager et celui-ci est traduit en séquences polypeptidiques, c'est-à-dire en protéines. La découverte du code génétique (section 1.2.4) par Nirenberg et al. est contemporaine de ces travaux sur la régulation (Nirenberg *et al.*, 1963).

Dans les années 1970, la génétique et la biologie moléculaire entrent dans une nouvelle phase. Le génie génétique est fondé sur des découvertes et des procédés techniques permettant d'isoler un (ou plusieurs) gène(s) d'un organisme, de le(s) modifier et de le(s) transférer dans un autre organisme. Cette technique a rendu possible une nouvelle génétique moléculaire. Une accélération considérable dans ce processus a été observée dans les années 80-90. Les méthodes de maîtrise de l'expression des gènes, de cartographie et de séquençage des génomes ont été développées, et c'est surtout à cette période qu'a émané une volonté internationale d'élucider complètement la séquence de génomes particuliers. En 1995, pour la première fois, la séquence du génome d'une cellule vivante (*Haemophilus influenzae*, une bactèrie responsable d'infections bronco-pulmonaires chez les jeunes enfants) a été déterminée. Au cours des 10 dernières années, les progrès ont été tout à fait spectaculaires. Aujourd'hui les séquences de plusieurs centaines de génomes complets sont connues, provenant de domaines très différents du Vivant : bactéries, archées, champignons, invertébrés, insectes, plantes, vertébrés. Le décryptage du génome humain est une prouesse technique à la mesure des avancées théoriques de la génétique. Ces avancées permettent des études à l'échelle de tous les génes d'une espèce.

La biologie moléculaire est donc entrée depuis 1995 dans l'ère de la génomique. Il s'agit d'étudier les génomes et en particulier l'ensemble des gènes, leur disposition sur les chromosomes, leur séquence, leur fonction, leur évolution et leur rôle. On dispose maintenant de l'information génétique exhaustive sur un nombre croissant d'organismes vivants et il est aujourd'hui possible d'aborder de manière globale un certain nombre de problèmes complexes dont on avait jusqu'à présent qu'une connaissance fragmentaire : évolution, voies métaboliques, interaction de la cellule avec l'extérieur, mécanismes globaux de régulation et de contrôle. Avec la mise à disposition de toutes ces informations, et notamment d'un nombre considérable de séquences ayant parfois un ancêtre commun lointain, la génomique comparative s'est particulièrement développée. En effectuant

des comparaisons de séquences entre plusieurs organismes, il est possible d'enrichir les connaissances que l'on a sur un gène ou un groupe de gènes. C'est en utilisant cette approche que sont le plus souvent effectuées des assignations de fonctions ainsi que de nombreuses études ayant trait à la phylogénie moléculaire (section 1.3).

L'accélération du processus de séquençage, permise en particulier par la robotisation et la parallélisation des méthodes d'analyse, nécessite un soutien de plus en plus important de l'outil informatique. C'est un outil incontournable pour permettre l'assemblage des milliers ou millions de fragments de génomes issus des automates de séquençage ainsi que pour extraire et analyser l'information contenue dans les banques de séquences.

1.2 L'information génétique

L'information génétique détermine les caractères d'un organisme. Elle est transmise des parents à leurs descendants. Différents éléments jouent un rôle dans le support de l'information génétique : la cellule, l'ADN, les chromosomes, les gènes et les protéines.

Pour cette partie, nous avons utilisé des informations provenant de (Tracqui & Demongeot, 2003) et (Maftah & Julien, 2003).

1.2.1 La cellule

Tous les êtres vivants, animaux, végétaux et microbes, sont composés de cellules. Par exemple, on estime à quelques milliers de milliards le nombre de cellules composant le corps humain. Depuis la formulation de la théorie cellulaire, habituellement attribuée au botaniste Matthias Jakob Schleiden et au zoologiste Theodor Schwannn (1839), il est reconnu que la cellule est l'unité de base du monde vivant. On distingue deux types de cellules correspondant à deux catégories d'organismes vivants classés selon la complexité de l'agencement de leur matériel génétique : les procaryotes divisées en deux groupes (les bactéries et les archées) et dont les cellules sont dépourvues de noyau, et les eucaryotes dont les cellules comprennent un noyau. Tous les organismes pluricellulaires sont constitués de cellules eucaryotes. Les eucaryotes multicellulaires sont composés de différentes sortes de cellules spécialisées qui dépendent alors les unes des autres pour survivre au sein d'un organisme. Cette dépendance est relative car la plupart des cellules possèdent tout ce qui est indispensable pour vivre et se multiplier, sous une forme isolée et complètement séparée de leur organisme d'origine, si elles sont mises dans un environnement qui procure les nutriments, hormones et facteurs de croissance appropriés. La cellule eucaryote (figure 1.1) présente toujours la même structure : une membrane, un cytoplasme et un noyau. La membrane individualise et isole du milieu extérieur. Le cytoplasme contient les organites, ces organites sont les instruments des différentes fonctions cellulaires. Le noyau de la cellule est séparé du cytoplasme par une enveloppe et renferme l'information génétique. Chez les procaryotes (figure 1.2), dans la plupart des cas, le matériel génétique n'est pas clairement

séparé du cytoplasme par une membrane. L'information génétique est la même dans toutes les cellules, qui la copient à chaque division cellulaire. L'expression de cette information est spécifique selon la fonction et le rôle de la cellule, mais l'information inutilisée reste présente.

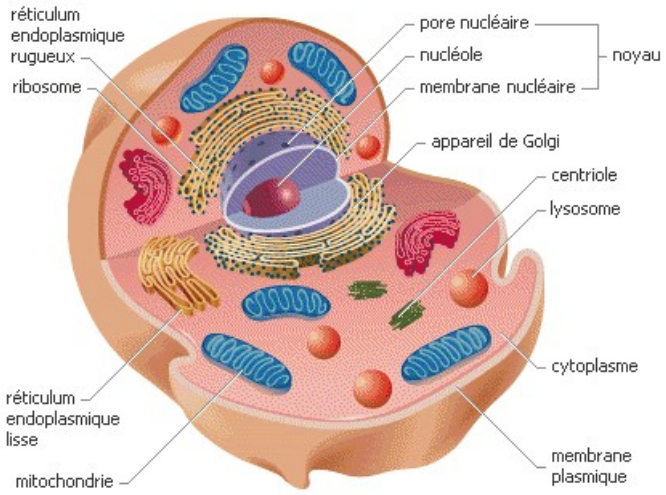

FIG. *1.1: Exemple de la structure d'une cellule eucaryote*

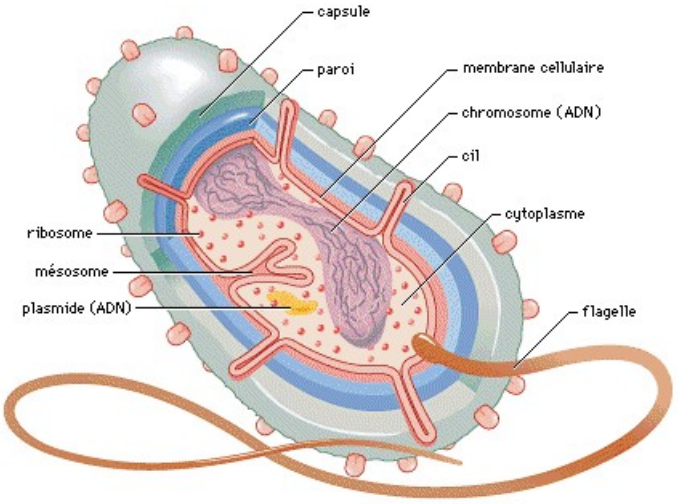

FIG. *1.2: Exemple de la structure d'une cellule procaryote (une bactérie)*

1.2.2 L'ADN

L'Acide DésoxyriboNucléique (ADN) est le support universel de l'information génétique chez les êtres vivants et le support de l'hérédité car il constitue leur génome et se transmet en totalité ou en partie lors des processus de reproduction. Il est constitué de

deux chaînes enroulées en double hélice (figure 1.3). Chaque hélice est formée par un long enchaînement de molécules élémentaires : les nucléotides. Chaque nucléotide comprend un sucre, le désoxyribose, un résidu phosphate et une des 4 bases azotées : adénine (A), guanine (G) pour les bases puriques et cytosine (C), thymine (T) pour les bases pyrimidiques. La nature de la base forme l'unique différence entre les nucléotides. Il existe donc 4 nucléotides différents. Par abus de langage, un nucléotide est souvent confondu avec le nom de sa base azotée. Les deux chaînes sont toujours étroitement reliées entre elles par des liaisons hydrogène (également appelées ponts hydrogène ou encore simplement liaisons H ou ponts H) établies entre deux bases de nucléotides face à face. Ces deux chaînes s'associent entre elles par complémentarité des bases. Les nucléotides sont complémentaires entre eux. Ainsi, l'adénine est complémentaire à la thymine (A-T) et la guanine est complémentaire à la cytosine (C-G). Deux liaisons hydrogène retiennent ensemble la paire A-T et trois retiennent la paire G-C. On dit que chaque chaîne ou brin est complémentaire de l'autre car chaque nucléotide d'une chaîne se lie à son partenaire complémentaire de l'autre côté. De plus, la molécule d'ADN est orientée étant donné que le lien entre les nucléotides ne se fait que dans un sens : de 5' vers 3' (5' et 3' sont les numéros des atomes de carbone du sucre). Ce lien est constitué par une liaison phosphodiester entre le phosphate Ⓟsitué sur le carbone 5' du sucre d'un nucléotide et une fonction alcool (-OH) située sur le carbone 3' du sucre du nucléotide suivant. Les deux brins complémentaires ont une orientation opposée.

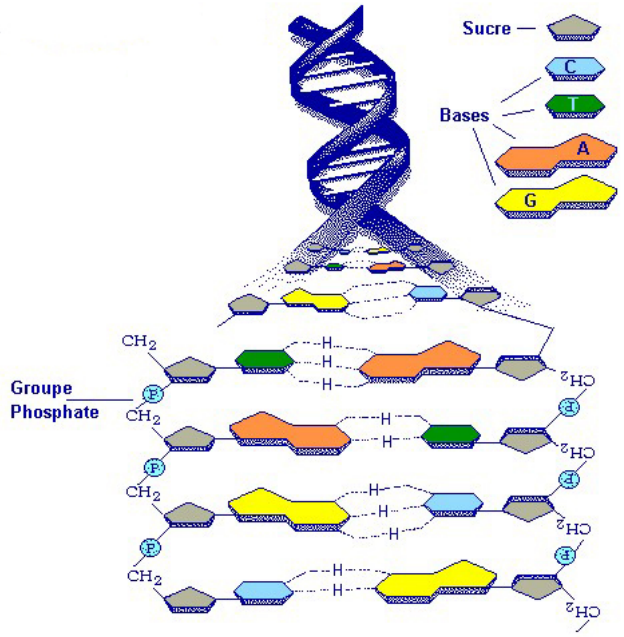

FIG. *1.3: La molécule d'ADN*

1.2.3 Les chromosomes

Chez les procaryotes, les deux extrémités de l'hélice sont souvent reliées de manière covalente et forment un ADN circulaire. Chez les eucaryotes, l'ADN est inséré dans une structure particulière complexe constituant le matériel génétique fonctionnel, appelé chromatine. Entre deux divisions cellulaires, la chromatine existe sous forme d'un écheveau de fibres. Avant la division cellulaire, la chromatine se condense en chromosomes. Chaque chromosome est constitué de deux bâtonnets parallèles, les chromatides, réunies en une région centrale, le centromère. Chaque chromatide se termine par une structure particulière appelé télomère (figure 1.4).

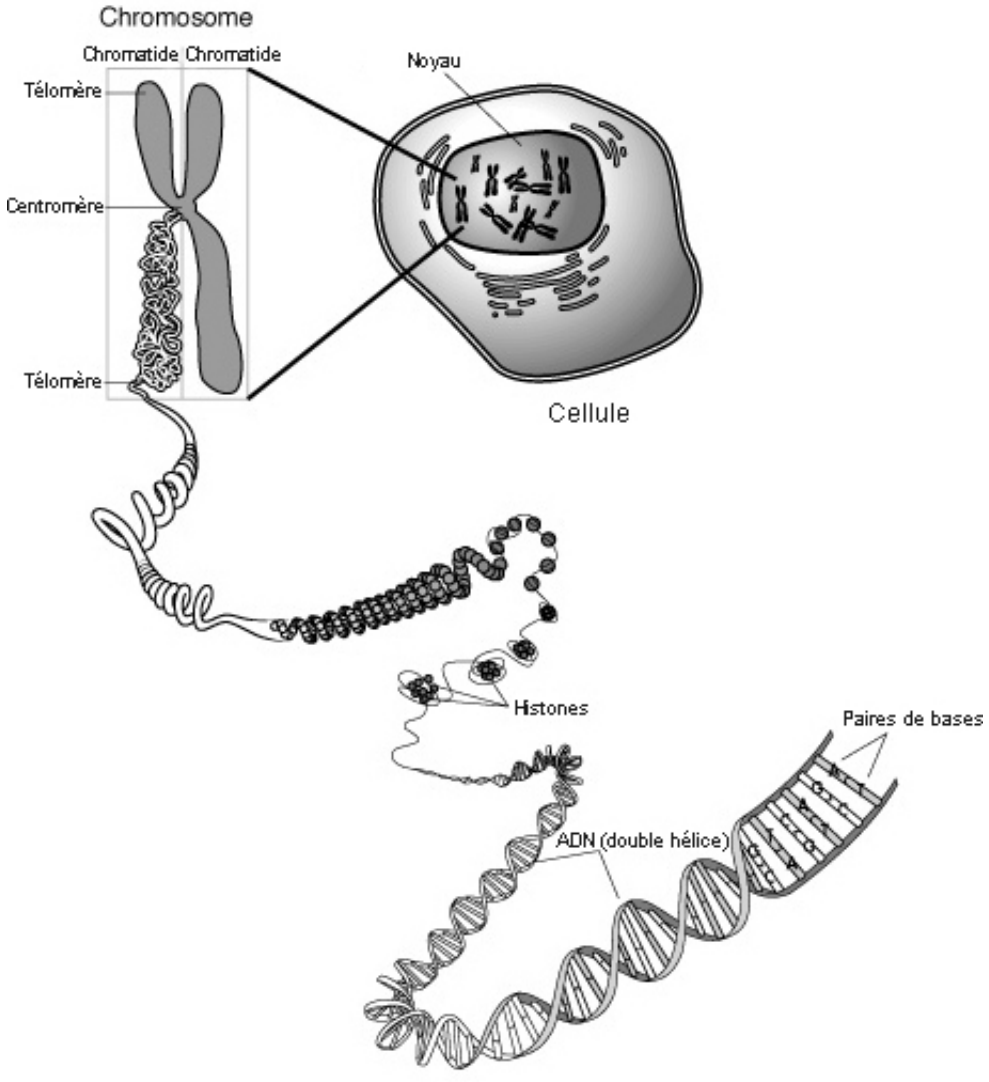

FIG. *1.4: La structure d'un chromosome*

Les deux chromatides d'un même chromosome sont strictement identiques et sont appelées chromatides sœurs. Chaque espèce a un nombre de chromosomes déterminé. Les chromosomes existent en un seul exemplaire dans les cellules sexuelles (cellules germinales

ou gamètes), cellules dites haploïdes, alors qu'ils sont présents en paires dans toutes les autres cellules (les cellules somatiques ou cellules non sexuelles), dites diploïdes. Les deux éléments d'une paire de chromosomes sont dits homologues et chacun des éléments provient d'un parent. Ainsi les cellules germinales humaines contiennent 23 chromosomes et les autres cellules contiennent 46 chromosomes répartis en 23 paires. On peut compter 22 paires d'autosomes (chromosome ne jouant pas de rôle dans la détermination du sexe) et 1 paire de chromosomes sexuels définissant le sexe du porteur : X pour la femelle (XX), Y pour le mâle (XY).

1.2.4 Les gènes et la synthèse protéique

Un gène est un fragment d'ADN qui correspond à un caractère héréditaire et constitue l'unité d'information génétique. Sa taille peut varier de quelques centaines à plusieurs dizaines de milliers de nucléotides. L'expression des caractères que l'ensemble des gènes gouvernent, telle qu'on peut l'observer, constitue le phénotype. Chaque individu présente un phénotype particulier, c'est-à-dire un certain nombre de caractères directement observables dont certains sont héréditaires, comme la couleur des cheveux, la taille, *etc.* Le gène porte l'information nécessaire à la biosynthèse d'un produit biologiquement actif, souvent une protéine. Les protéines sont des molécules complexes constituées d'un enchaînement précis d'acides aminés. Il en existe 20 différents (figure 1.5) et de leur ordre dépendent les propriétés de la protéine.

Code à une lettre	Code à trois lettre	Description
A	Ala	Alanine
B	Asx	Asparagine ou acide aspartique
C	Cys	Cystéine
D	Asp	Acide aspartique
E	Glu	Acide glutamique
F	Phe	Phenylalanine
G	Gly	Glycine
H	His	Histidine
I	Ile	Isoleucine
K	Lys	Lysine
L	Leu	Leucine
M	Met	Méthionine
N	Asn	Asparagine
P	Pro	Proline
Q	Gln	Glutamine
R	Arg	Arginine
S	Ser	Serine
T	Thr	Thréonine
V	Val	Valine
W	Trp	Tryptophane
Y	Tyr	Tyrosine
Z	Glx	Glutamine ou acide glutamique

FIG. *1.5: Listes des 20 acides aminés composant des protéines*

Le passage de la séquence du gène, c'est-à-dire de l'enchaînement de 4 nucléotides différents, à la séquence de la protéine, un enchaînement de 20 acides aminés différents,

correspond à la synthèse protéique. Elle se décompose en deux étapes : la transcription et la traduction. Chaque gène contient les instructions pour coder une ou plusieurs protéines. Chez les eucaryotes, le gène est constitué de régions codantes ou exons, séparées par des régions non-codantes ou introns. Seules les exons se retrouvent, sous forme traduite, dans la protéine. Enfin il faut savoir que l'information génétique se transmet généralement selon trois processus : la réplication, la transcription et la traduction.

1.2.4.1 La réplication

Lors de chaque division cellulaire, la totalité de l'ADN doit être dupliquée. La duplication d'une molécule d'ADN parent en deux molécules d'ADN filles est appelée Réplication ou Duplication. Elle permet un dédoublement de l'ADN par l'appariement complémentaire des bases. Pendant la réplication (figure 1.6), la double hélice est déroulée localement grâce à des protéines spécifiques appelées hélicases et les deux brins sont séparés. Chaque brin sert de matrice pour la synthèse d'un nouveau brin complémentaire.

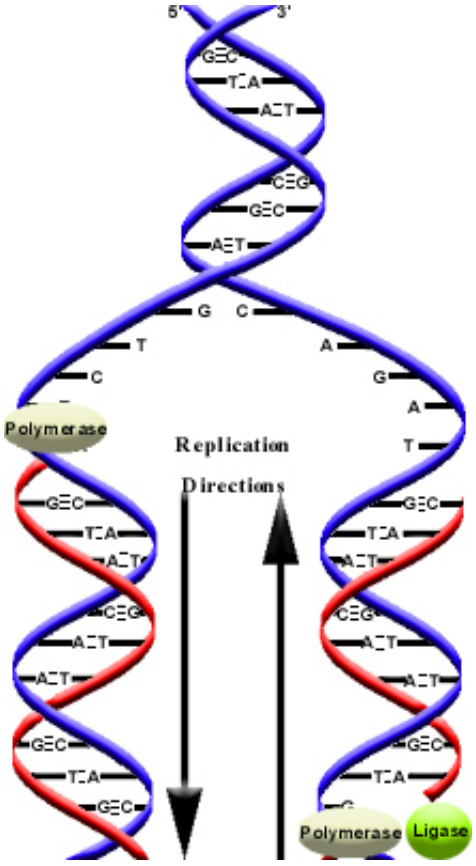

FIG. *1.6: Le mécanisme de la réplication*

La réplication est dite semi-conservative car dans chaque molécule fille d'ADN, un brin provient de la molécule parent, il est de ce fait conservé et un autre brin est nouvellement synthétisé. La réplication fournit donc deux molécules d'ADN identiques

entre elles et identiques à la molécule parent. La réplication s'effectue uniquement dans le sens 5'->3' et les brins complémentaires sont synthétisés grâce à l'ADN polymérase. Cette enzyme est chargée de récupérer les nucléotides libres et de les apparier aux nucléotides complémentaires du brin parent. Etant donné que les deux brins d'ADN sont antiparallèles, un des brins en formation, le brin avancé ou principal, s'allongera de façon continue. L'autre brin, dit retardé ou secondaire, se forme de façon discontinue par de petits fragments d'ADN, appelés fragments d'Okazaki. Les différents fragments d'ADN du brin retardataire sont ensuite réunis par l'enzyme ADN ligase.

Etant donné que l'ADN est porteur de l'information génétique de la cellule, il faut que la réplication se fasse avec une fidélité élevée afin de ne pas perdre d'informations. Cependant l'ADN est soumis à des agressions physiques, chimiques ou des accidents de polymérisation du brin synthétisé. Il peut ainsi subir des dégradations ou des changements permanents (mutations) s'il n'y a pas de protection et de réparations effectuées. Il existe donc des mécanismes de réparation de l'ADN garantissant à la cellule fille la transmission d'une information génétique presque identique à celle de la cellule parent. Les principaux mécanismes de réparation de l'ADN utilisent le fait que l'information génétique existe en deux exemplaires (sur chacun des brins de la double hélice). Lorsqu'il y a une erreur sur l'un des brins, l'ADN polymérase assure le remplacement du fragment endommagé par un fragment correct grâce à l'information portée sur le brin complémentaire intact.

1.2.4.2 La transcription

L'ADN est situé dans le noyau alors que la synthèse des protéines a lieu dans le cytoplasme au contact des ribosomes accolés au réticulum endoplasmique. Pour assurer la liaison entre les deux, il y a synthèse d'ARN messager (Acide RiboNucléique) dans le noyau de la cellule chez les eucaryotes et dans le cytoplasme chez les procaryotes. La synthèse d'ARN à partir d'ADN correspond à la transcription. L'ARN messager (ARNm) est constitué d'un enchaînement de nucléotides comme l'ADN. Il y a cependant quelques différences entre la structure de ARNm et celle de l'ADN : il est simple brin (constitué d'une seule chaîne de nucléotides), la thymine est remplacée par l'uracile et le sucre est un ribose et non un désoxyribose. La transcription (figure 1.7) nécessite une enzyme, l'ARN polymérase, qui se fixe sur une séquence particulière de l'ADN (la région promotrice) indiquant ainsi le début de la transcription. Cette enzyme assure la séparation des brins à transcrire et transcrit une des deux chaînes de l'ADN en une chaîne d'ARN. La synthèse de l'ARN s'effectue toujours dans le sens 5'->3' par addition des nucléotides d'ARN complémentaires de ceux du brin d'ADN. La transcription a donc pour rôle principal de transférer vers l'ARN l'information contenue dans la séquence d'un gène, c'est-à-dire de réaliser un intermédiaire, une copie conservant la signification de la séquence des bases de l'ADN.

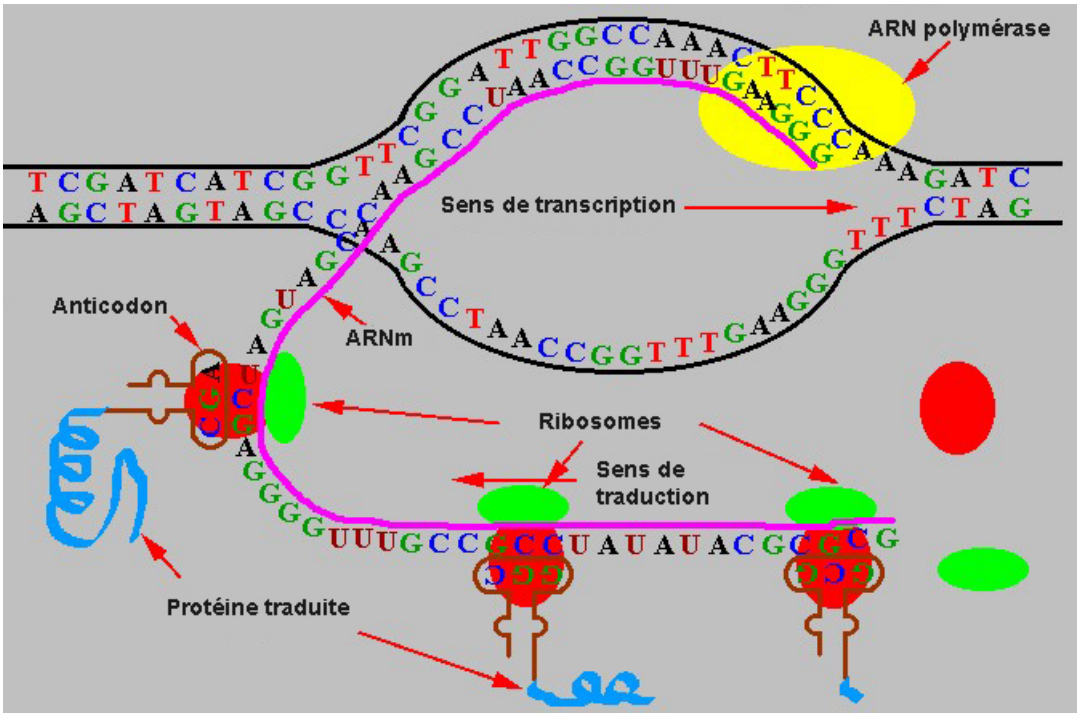

FIG. *1.7: Les mécanismes de transcription et de traduction*

1.2.4.3 La traduction

La traduction est l'ensemble des mécanismes qui permettent la synthèse des protéines à partir des ARNm. Chez les eucaryotes, l'ARNm passe à travers la membrane nucléaire et transporte dans le cytoplasme, où a lieu la traduction, l'information qu'il porte. Le transfert de l'information de la séquence nucléotidique à la séquence protéique fait intervenir un système de correspondance entre les deux types de séquences que l'on appelle code génétique (figure 1.8). Les séquences nucléotidiques de l'ARNm sont lues séquentiellement par triplets, appelés codons (AAC, TGC, etc.). Le code génétique donne la correspondance entre codons et acides aminés. Un codon code pour un certain acide aminé dans une protéine, par exemple CCA code pour la glycine. Il y a 64 (4^3) codons possibles. Plusieurs triplets pouvant coder pour le même acide aminé, le code est dit dégénéré. Parmi les différents codons, il existe un codon initiateur et trois codons stops qui permettent de délimiter le début et la fin de la zone à décoder pour former la protéine. Lors de la traduction (figure 1.7), l'ARNm se fixe à un complexe moléculaire composé d'ARN ribosomiques (ARNr) et de protéines, appelé ribosome. Celui-ci va permettre l'assemblage des acides aminés afin de synthétiser la protéine : il parcourt le brin d'ARNm et, grâce à l'ensemble des ARNs de transfert (ARNt), va ajouter un acide aminé à la protéine en cours de fabrication en fonction du codon lu. Il n'y a aucun chevauchement dans la lecture des codons successifs. Il existe donc trois manières, appelées phases de lecture, pour découper en codons la séquence. C'est le premier codon lu qui va définir la phase de lecture. Selon

cette phase la séquence en acides aminés sera différente. Lorsque le ribosome atteint le codon stop, il se détache de la protéine et du brin d'ARNm. La protéine est alors libérée dans l'organisme et acquiert ses fonctions.

LE CODE GENETIQUE

Première lettre	Deuxième lettre U		Deuxième lettre C		Deuxième lettre A		Deuxième lettre G	
U	UUU	phe (F)	UCU	ser (S)	UAU	tyr (Y)	UGU	cys (C)
	UUC		UCC		UAC		UGC	
	UUA	leu (L)	UCA		UAA	STOP	UGA	STOP
	UUG		UCG		UAG	STOP	UGG	trp (W)
C	CUU	leu (L)	CCU	pro (P)	CAU	HIS 5H)	CGU	arg (R)
	CUC		CCC		CAC		CGC	
	CUA		CCA		CAA	gln (Q)	CGA	
	CUG		CCG		CAG		CGG	
A	AUU	ile (I)	ACU	thr (T)	AAU	asn (N)	AGU	ser (S)
	AUC		ACC		AAC		AGC	
	AUA		ACA		AAA	lys (K)	GA	arg (R)
	AUG	met (M)	ACG		AAG		AGG	
G	GUU	val (V)	GCU	ala (A)	GAU	asp (D)	GGU	gly (G)
	GUC		GCC		GAC		GGG	
	GUA		GCA		GAA	glu (E)	GGA	
	GUG		GCG		GAG		GGG	

FIG. *1.8: Le code génétique*

1.3 L'évolution des séquences

L'évolution est définie par l'interaction entre les modifications au sein de chaque espèce et l'apparition d'espèces nouvelles (spéciation). Ces deux phénomènes sont accompagnés d'extinctions d'espèces. Le processus évolutif se base sur la diversité existant au sein de chaque espèce. Tout mécanisme susceptible de modifier les caractéristiques d'une population, génération après génération, peut engendrer une évolution. Le plus célèbre est probablement la sélection naturelle, qui a été proposée par Charles Darwin et décrite dans son ouvrage majeur l'Origine des espèces, en 1859. Diverses théories se sont par la suite succédées pour conduire au consensus actuel (théorie synthétique de l'évolution). Celui-ci n'est d'ailleurs pas total, comme en témoignent certaines divergences dont la plus connue est celle ayant opposé Stephen Jay Gould et Richard Dawkins, sur l'intérêt d'introduire la notion d'équilibres ponctués.

La plupart des individus d'une espèce donnée possèdent la même formule chromosomique (le nombre et la forme structurale des chromosomes), le même caryotype (photographie ordonnée des chromosomes) et les mêmes gènes, et pourtant aucun n'est identique.

Cela s'explique en partie par la diversité génétique due aux polymorphismes. Les principaux mécanismes qui conduisent au polymorphisme sont les modifications accidentelles de l'ADN appelées mutations et le brassage génétique (intra et interchromosomique) assuré par la reproduction sexuée. Par différents processus les séquences d'ADN subissent des mutations ponctuelles provoquant des substitutions, des insertions ou des délétions de nucléotides et donc une altération de l'information génétique portée par un individu :

- une substitution correspond au remplacement d'un nucléotide par un autre porteur d'une base différente,
- une insertion est l'ajout dans la chaîne d'un ou d'une portion de chaîne de nucléotides
- une délétion implique la suppression d'un ou d'une portion de la chaîne de nucléotides.

L'occurrence de telles mutations peut modifier la fonction de protéines ou d'ARN. L'étude de ces processus de mutation est difficile, car ils ne peuvent être étudiés que de manière indirecte, par comparaison de séquences d'ADN entre des individus de la même espèce ou d'espèces différentes. C'est à partir de ces comparaisons que l'on va tenter de reconstruire l'histoire évolutive des séquences.

Une des méthodes employées pour essayer de reconstruire l'histoire évolutive des séquences est la phylogénie. La phylogénie moléculaire est une branche de la systématique qui est l'étude et la description de la diversité des êtres vivants, la recherche de la nature et des causes de leurs différences et de leurs ressemblances, la mise en évidence des relations de parenté existant entre eux et l'élaboration d'une classification traduisant ces liens de parenté. La phylogénie moléculaire consiste à déterminer l'arbre phylogénétique d'un ensemble de séquences homologues données, c'est-à-dire la configuration la plus probable pour rendre compte du degré de parenté existant entre ces séquences. Les feuilles et les noeuds d'un arbre phylogénétique correspondent à des taxons qui sont définis comme étant des regroupements d'organismes reconnus en tant qu'unités formelles (Lecointre & Guyader, 2001).

2 Le stockage de données biologiques

Aujourd'hui les méthodes rapides de séquençages (cf. chapitre 2, page 31 section 2.2.2, page 34) sont utilisées fréquemment et le nombre de nouvelles séquences augmente rapidement. Toutes les données issues du séquençage doivent être traitées et analysées afin d'obtenir le plus grand nombre d'informations. Il faut ainsi stocker ces séquences et toutes les informations obtenues. Pour cela, de grandes bases de données de séquences ont été mises en oeuvre pour permettre un accès facile aux données. Les premières banques de données en biologie moléculaire ont traité des informations structurales sur les protéines, puis très rapidement, des séquences protéiques et nucléotidiques. Il existe différents types de bases de données biologiques : celles qui sont dites généralistes et qui stockent des séquences provenant de tous les organismes et celles dites spécialisées qui se consacrent plus

particulièrement à un organisme ou à une thématique donnée.

2.1 Les banques généralistes

Il existe plusieurs banques généralistes publiquement accessibles. La principale banque généraliste de séquences nucléotidiques est produite par trois partenaires : EMBL data library (Cochrane *et al.*, 2006) en Europe, GenBank (Benson *et al.*, 2006) aux Etats-Unis et DDBJ (Okubo *et al.*, 2006) au Japon. La plupart des données de ces banques proviennent de soumissions effectuées par les auteurs. D'autres regroupent des séquences protéiques telles que UNIPROT (Wu *et al.*, 2006), GenPept, HAMAP (High-quality Automated and Manual Annotation of microbial Proteomes) (Gattiker *et al.*, 2003)), *etc.*

De la même manière que pour les banques de séquences nucléotidiques, leur organisation se base autour des annotations biologiques et biochimiques d'une part, et des séquences d'autre part. GenPept correspond à la traduction de l'ensemble des parties codantes de GenBank. La principale banque de protéines est UNIPROT. En effet, elle possède de nombreux atouts : redondance minimale, références croisées, qualité d'annotation, *etc.* Elle correspond à la fusion de SWISS-PROT (Wu *et al.*, 2006), TrEMBL et PIR (Protein Information Resource) (Wu *et al.*, 2003). Les séquences contenues dans SWISS-PROT sont issues de la traduction des gènes annotés dans EMBL, d'autres banques protéiques, de publication scientifiques et de quelques soumissions d'auteurs. TrEMBL est la version protéique de la banque nucléotidique EMBL. Elle contient la traduction de toutes les parties codantes annotées de EMBL en excluant les protéines présentes dans SWISS-PROT. PIR, qui maintenant n'existe plus, fournissait des informations organisées selon des critères taxonomiques et de similarité. Enfin, HAMAP est un projet développé par le groupe SWISS-PROT. Son but est d'annoter automatiquement les protéines provenant des projets de séquençage des génomes microbiens. La banque contient également des collections de familles de protéines microbiennes générées par des experts et utilisées pour l'annotation automatique.

Ces banques généralistes permettent donc de centraliser toutes les séquences connues. Cependant, il existe tout de même un grand nombre d'erreurs, notamment au niveau des annotations des séquences ainsi qu'une redondance des informations dans certaines banques.

2.2 Les banques spécialisées

Pour pallier ces inconvénients, l'augmentation exponentielle du volume, de la diversité des séquences et la diversité des études, des banques spécialisées ont été développées. Ces développements ont permis l'introduction d'informations spécifiques à chacune permettant ainsi d'avoir des banques adaptées aux besoins des utilisateurs. Elles répondent pour la plupart soit à des besoins ponctuels, soit à des besoins liés à des secteurs d'activité bien

précis.

Parmi celles-ci, ont été développées des banques thématiques se consacrant à un domaine bien précis. Ainsi, certaines regroupent des données sur les structures moléculaires tridimensionnelles telles que la PDB (Protein Data Bank) (Berman *et al.*, 2000). D'autres s'intéressent à la structure en domaine des séquences protéiques comme la banque ProDom (Servant *et al.*, 2002). Il y en a également qui centralisent des données sur les signatures, caractéristiques de certaines protéines telle que PROSITE (Hulo *et al.*, 2006). D'autres encore traitent des séquences et des structures d'ARN. Il existe deux principales banques stockant des séquences d'ARN ribosomique et intégrant des alignements (section 3.2) et des arbres phylogénétiques (section 3.3) : la banque américaine RDP (Ribosomal Database Project) (Cole *et al.*, 2005) qui est mise à jour régulièrement et celle européenne, Ribosomal RNA Database, (Wuyts *et al.*, 2004), qui n'est plus maintenue.

De plus, il existe des banques proposant une classification de gènes protéiques sous forme de familles. La construction de ce type de banques débute par une recherche de similarité entre toutes les protéines d'un ensemble donné, effectuée par BLAST (Altschul *et al.*, 1990; Altschul *et al.*, 1997) ou FASTA (Pearson & Lipman, 1988) (section 3.1). Puis en utilisant des critères de similarité, ces protéines sont regroupées en familles. Plusieurs banques de ce type existent. Par exemple, ProtFam (Mewes *et al.*, 2000) construite à partir des séquences de PIR propose quatre niveaux de similarité entre protéines ainsi que des alignements et des dendrogrammes, ou encore ProtoMap qui est l'équivalent de ProtFam (Yona *et al.*, 2000) pour la banque SWISS-PROT. Certaines de ces banques ne se consacrent qu'à un groupe d'organismes comme HOVERGEN (Homologous Vertebrate Genes Database) (Duret *et al.*, 1999) qui contient les séquences de tous les gènes des vertébrés figurant dans GenBank et d'autres comme HOGENOM (Homologous Sequences from Complete Genomes Database) ou COG (Cluster of Orthologous Groups of protein) (Tatusov *et al.*, 2001) regroupent les séquences de génomes complets. Dans ces banques, les séquences sont toutes groupées en familles de gènes homologues c'est-à-dire que les séquences d'une même famille ont toutes un ancêtre commun. Elles sont également préalablement alignées.

Pour faciliter l'accès aux informations d'un organisme donné, des banques génomiques ont été mises en place afin de stocker un ou plusieurs génomes uniquement. Par exemple, Ensembl (Birney *et al.*, 2006) regroupe des génomes allant des mammifères (homme, souris, *etc*) aux Chordés primitifs (*Ciona intestinalis*) ainsi que des génomes d'eucaryotes (levure, mouche, ver, *etc*) et un nombre limité de génomes d'insectes. Une version d'Ensembl structurée en familles de gènes homologues, HomolEns (Database of Homologous Sequences from Ensembl Complete Genomes), a été développée au Pôle Bio-Informatique Lyonnais.

Enfin la banque Genome Reviews (Kersey *et al.*, 2005) permet d'avoir une vue mise à jour, standardisée et largement annotée des séquences des génomes d'organismes complètement séquencés. Elle correspond à une version nettoyée des entrées de EMBL, GenBank et DDBJ.

Les banques de familles de gènes homologues telles que HOVERGEN ou HOGENOM nous intéressent plus particulièrement car c'est avec ce type de banques que nous avons principalement travaillé. Elles fournissent pour chaque famille un alignement et une phylogénie. Ces banques permettent la comparaison de séquences homologues qui constitue une étape essentielle pour la compréhension des mécanismes d'évolution moléculaire ou encore pour connaître la fonction d'un gène.

Parmi les séquences homologues, on peut distinguer les séquences orthologues, c'est-à-dire les séquences qui divergent après un événement de spéciation, et les séquences paralogues, c'est-à-dire qui divergent après une duplication du gène ancestral. Il est important de faire cette distinction pour la phylogénie moléculaire (section 3.3) car il est nécessaire de travailler avec des orthologues pour reconstruire l'histoire des spéciations mais également pour l'analyse comparative de séquences (recherche de régions conservées fonctionnellement, prédiction de fonction de nouveaux gènes, *etc*). Cependant la distinction orthologue/paralogue est parfois difficile à établir à cause de données biologiques trop divergentes ou manquantes, d'une mauvaise ou du manque d'annotation des séquences dans les bases de données, *etc*. Ce type de banque permet donc d'accéder directement à des familles de gènes homologues ainsi qu'à leur alignement, à l'arbre phylogénétique et aux informations taxonomiques correspondantes, aidant ainsi la détermination des orthologues et des paralogues.

Le processus de détermination de l'homologie est une tâche complexe impliquant plusieurs étapes dans l'analyse des séquences (section 3) : l'utilisation d'un critère de similarité pour rechercher les familles de gènes, l'alignement des séquences appartenant à une même famille et la construction de l'arbre phylogénétique correspondant.

3 L'analyse de séquences

La génomique comparative est souvent utilisée dans l'analyse de séquence. L'analyse comparative de séquences est une approche efficace pour identifier des séquences, détecter les régions fonctionnellement importantes dans les protéines ou les séquences d'acides nucléiques, étudier l'ensemble de la structure de génomes ou même étudier l'histoire évolutive des séquences.

3.1 Comparaison de séquences, recherche de similarité et alignement par paire

La recherche de similarité entre séquences est un élément fondamental qui constitue souvent la première étape des analyses de séquences. Cela consiste à comparer deux séquences en repérant les régions proches entre elles. Pour cela, il faut rechercher les régions qui comptabilisent un maximum de caractères communs et un minimum de changements lorsqu'on les superpose l'une à l'autre c'est-à-dire quand elles sont alignées. Le taux de similarité permet d'avoir une indication sur l'existence d'une homologie entre les séquences. Plus le taux de similarité entre deux séquences est haut, plus il est probable que ces

séquences soient homologues, deux séquences sont homologues si elles ont un ancêtre commun. Ainsi la recherche de similarité permet d'avoir des informations sur l'évolution des séquences et un fort taux de similarité entre des séquences indique une origine évolutive commune. Dans la construction de banques de familles de gènes homologues comme celles décrites à la section précédente (section 2), la notion de similarité est utilisée pour regrouper les séquences en familles. Les banques telles que HOVERGEN ou HOGENOM permettent ainsi, par association, d'obtenir des informations importantes sur la structure, la fonction ou l'évolution des biomolécules. La comparaison de séquences est largement utilisée dans les recherches de motifs à travers une séquence, dans la caractérisation de régions communes ou similaires entre deux ou plusieurs séquences, dans la comparaison d'une séquence avec l'ensemble ou sous-ensemble des séquences d'une banque, ou bien encore dans l'établissement d'un alignement multiple (section 3.2) sur lequel sont basées les analyses d'évolution moléculaire. Il faut également savoir que la recherche de similarité à partir d'une séquence protéique est plus sensible que les comparaisons faites à partir d'une séquence d'ADN. En effet, l'ADN est constitué de quatre bases uniquement alors que les séquences protéiques sont constituées de vingt acides aminés différents. La comparaison de séquences protéiques est donc plus fine que la comparaison de séquences nucléiques. Enfin les banques protéiques sont en général plus petites que les banques nucléiques ce qui permet d'obtenir des résultats plus rapidement lorsque l'on compare une séquence aux séquences d'une banque.

3.1.1 Le principe

Les processus de comparaison et de recherche de similarité entre deux séquences utilisent l'alignement de celles-ci. Un alignement de deux séquences met en évidence les ressemblances qu'il existe entre elles : il essaie de faire correspondre les résidus (nucléotides ou acides aminés) de la première séquence avec ceux de la deuxième (figure 1.9).

```
Séquence 1: NRPPTSEELPSCYTGDDWSG
Séquence 2: KRSP--E--SS--TTEDWSG
```

FIG. *1.9: Exemple d'alignement entre la séquence 1 et la séquence 2*

Lorsque dans une paire de résidus apparaît le même résidu dans les deux séquences, cela suggère que le résidu en ce site n'a pas changé depuis la divergence des deux séquences. On parle alors d'appariement. A l'inverse, on parle de mésappariement lorsque dans la paire, les résidus de chaque séquence sont différents, ce qui indique qu'au moins une substitution s'est produite dans une des deux séquences depuis la divergence entre les deux. Enfin il existe un troisième type de paires constitué d'un résidu d'une des séquences et d'un caractère spécial appelé brèche ou résidu nul représenté le plus souvent par "-", ce qui montre qu'une délétion ou une insertion (désignées par la contraction indel pour INsertionDELetion) s'est faite dans l'une des deux séquences. Les substitu-

tions, insertions et délétions correspondent à des événements de mutation dans la séquence.

Pour comparer des séquences, il faut rechercher l'alignement de score maximum c'est-à-dire faire correspondre un maximum de résidus d'une séquence avec ceux de l'autre en insérant entre les résidus une brèche, dans le but de maximiser le score de l'alignement. Afin d'obtenir ce score, le principe est d'associer à chaque paire de résidus et à chaque brèche un score élémentaire et de faire la somme de ces scores sur la longueur de l'alignement. Une fonction permet de calculer le score élémentaire. Il existe plusieurs méthodes pour définir cette fonction. Ces méthodes utilisent des matrices de scores qui rendent compte de tous les états possibles en fonction de l'alphabet utilisé dans la description des séquences. De plus, la plupart des méthodes nuancent le calcul du score en donnant des pénalités plus ou moins importantes aux brèches (Gotoh, 1982; Gotoh, 1993).

3.1.2 Les matrices de similarité

Les matrices de similarité ou matrice de substitution ou encore matrices de scores permettent de définir le score élémentaire à attribuer à la conservation d'un résidu et au remplacement d'un résidu par un autre. Pour les séquences nucléiques, il existe peu de matrices car ces séquences sont constituées uniquement des quatre nucléotides A, T, G, C qui jouent un rôle équivalent dans la structure et la fonction de la molécule. Ainsi dans la matrice d'identité ou unitaire, toutes les bases sont considérées comme équivalentes. Il existe également d'autres matrices privilégiant certaines associations ou donnant la fréquence pour chacune des douze possibilités telle que la matrice de transition-transversion qui tient compte du fait que les mutations de A en G et de G en A d'une part, de C en T et de T en C d'autre part, sont plus fréquentes que les autres. Pour les séquences protéiques, les matrices sont plus complexes car il faut tenir compte du fait que les vingt acides aminés apparaissent dans les séquences avec des fréquences différentes et que certains acides aminés ont des fonctions biochimiques ou des structures très proches et sont donc plus facilement interchangeables. Une des premières matrices à utiliser ce principe a été celle déduite de la dégénérescence du code génétique (Fitch, 1966). Cette matrice se base sur le nombre de changements minimum pour passer d'un acide aminé à l'autre. Depuis de nombreuses matrices ont été créées et l'on peut classer celles-ci en deux catégories : les matrices liées à l'évolution et les matrices liées aux caractères physico-chimiques.

Parmi les matrices issues d'études montrant le caractère de substitution des acides aminés au cours de l'évolution, les matrices de type PAM (Point Accepted Mutation), de type BLOSUM (BLOcks SUbstitution Matrix) et la matrice de Gonnet sont celles qui ont été les plus utilisées.

Les matrices de type PAM représentent les échanges possibles ou acceptables d'un acide aminé par un autre (probabilités de mutation) lors de l'évolution des protéines (Dayhoff *et al.*, 1978; Schwartz & Dayhoff, 1978). Une matrice de probabilité a été calculée à partir d'alignements globaux de séquences de protéines de fonctions identiques

chez différentes espèces apparentées (71 familles de séquences protéiques peu divergentes). Chaque élément de cette matrice donne la probabilité qu'un acide aminé A soit remplacé par un acide aminé B durant une étape d'évolution. Cette matrice de probabilité de mutation correspond en fait à une mutation acceptée pour 100 sites durant un temps d'évolution particulier. On parle ainsi de la matrice de transition PAM1. Si l'on multiplie la matrice par elle-même un certain nombre de fois, on obtient une matrice PAMX qui donne des probabilités de transition pour des distances d'évolution plus grandes. Les matrices de substitution sont dérivées des matrices de transition : en prenant compte des fréquences relatives de mutation et en prenant le logarithme de chaque élément des matrices de transition, on construit les matrices de substitution. Celles-ci donnent les scores pour toutes les paires d'acides aminés possibles. Par exemple, la matrice de substitution PAM250 contient les scores relatifs à la probabilité d'occurrence des différentes mutations après un temps d'évolution de 250 mutations pour 100 acides aminés (figure 1.10). Bien que ces matrices soit basées sur un échantillon assez large et représente assez bien les probabilités de substitution d'un acide aminé en un autre suivant que cette mutation engendre ou pas des changements dans la structure ou la fonctionnalité des protéines, elles présentent néanmoins un certain nombre d'inconvénients. En effet, elles sont basées sur des données de petites protéines globulaires et le nombre de séquences protéiques à l'origine des matrices PAM en 1978 était restreint. Plus récemment, la matrice JTT (Jones *et al.*, 1992) a été créée en utilisant la même méthode que pour la matrice de Dayhoff mais avec des données de banques plus récentes.

Ala(A)	2																	
Cys(C)	-2	12																
Asp(D)	0	-5	4															
Glu(E)	0	-5	3	4														
Phe(F)	-4	-4	-6	-5	9													
Gly(G)	1	-3	1	0	-5	5												
His(H)	-1	-3	1	1	-2	-2	6											
Ile(I)	-1	-2	-2	-2	1	-3	-2	5										
Lys(K)	-1	-5	0	0	-5	-2	0	-2	5									
Leu(L)	-2	-6	-4	-3	2	-4	-2	2	-3	6								
Met(M)	-1	-5	-3	-2	0	-3	-2	2	0	4	6							
Asn(N)	0	-4	2	1	-4	0	2	-2	1	-3	-2	2						
Pro(P)	1	-3	-1	-1	-5	-1	0	-2	-1	-3	-2	-1	6					
Gln(Q)	0	-5	2	2	-5	-1	3	-2	1	-2	-1	1	0	4				
Arg(R)	-2	-4	-1	-1	-4	-3	2	-2	3	-3	0	0	0	1	6			
Ser(S)	1	0	0	0	-3	1	-1	-1	0	-3	-2	1	1	-1	0	2		
Thr(T)	1	-2	0	0	-3	0	-1	0	0	-2	-1	0	0	-1	-1	1	3	
Val(V)	0	-2	-2	-2	-1	-1	-2	4	-2	2	2	-2	-1	-2	-2	-1	0	4
Trp(W)	-6	-8	-7	-7	0	-7	-3	-5	-3	-2	-4	-4	-6	-5	2	-2	-5	-6	17	
Tyr(Y)	-3	0	-4	-4	7	-5	0	-1	-4	-1	-2	-2	-5	-4	-4	-3	-3	-2	0	10
	Ala (A)	Cys (C)	Asp (D)	Glu (E)	Phe (F)	Gly (G)	His (H)	Ile (I)	Lys (K)	Leu (L)	Met (M)	Asn (N)	Pro (P)	Gln (Q)	Arg (R)	Ser (S)	Thr (T)	Val (V)	Trp (W)	Tyr (Y)

FIG. *1.10: La matrice PAM250*

Dans les matrices de type BLOSUM (Henikoff & Henikoff, 1992), le degré de substitution des acides aminés a été mesuré en observant des blocs d'acides aminés issus de séquences de protéines plus ou moins divergentes (500 familles de protéines). Chaque bloc est obtenu par l'alignement multiple (section 3.2) sans insertion-délétion de courtes régions très conservées. Puis tous les segments de séquences ayant un pourcentage d'identité

minimum au sein de leur bloc sont regroupés. On en déduit des fréquences de substitution pour chaque paire d'acides aminés et on calcule ensuite une matrice logarithmique de probabilité appelée BLOSUM. A chaque pourcentage d'identité correspond une matrice particulière. Ainsi la matrice BLOSUM60 est obtenue en utilisant un seuil d'identité de 60% .

Enfin la matrice de Gonnet (Gonnet *et al.*, 1992) est calculée d'après un très large échantillon (Prot). Une mesure de distance entre acides-aminés classique a été utilisée pour estimer l'alignement entre protéines. Puis, les résultats ont permis d'estimer une nouvelle matrice de distance, qui a servi ensuite à estimer l'alignement entre protéines et à calculer une nouvelle matrice. Cette procédure a été répétée de façon itérative jusqu'à convergence.

Dans certains cas des matrices liées aux caractéristiques physico-chimiques des protéines sont utilisées afin de révéler au mieux certaines de ces caractéristiques communes à deux protéines. Quelques unes sont basées sur une estimation des proximités physico-chimiques entre les aminoacides (Grantham, 1974), d'autres utilisent le caractère hydrophile ou hydrophobe des protéines et leur structure secondaire ou tertiaire telles que la matrice d'hydrophobicité (Levitt, 1976) ou la matrice de structure secondaire (Levin *et al.*, 1986). Plus récemment l'augmentation du nombre de structures tridimensionnelles déterminées, a permis d'établir des matrices basées sur la comparaison de ces structures comme la matrice établie par Risler et al. (Risler *et al.*, 1988) et la matrice de Johnson et Overington (Johnson & Overington, 1993).

3.1.3 Les algorithmes et les programmes

Les programmes de comparaison de séquences ont pour but de repérer les régions identiques ou très proches entre deux séquences et de distinguer celles qui sont significatives et qui ont un sens biologique de celles qui sont observées par hasard. Il existe deux types d'algorithmes de recherche de similarité : ceux qui calculent une similarité globale et ceux qui recherchent une similarité locale.

Les algorithmes qui calculent un score de similarité globale se basent sur la totalité des séquences. L'algorithme de Needleman et Wunsch (Needleman & Wunsch, 1970) a tout d'abord été développé pour réaliser l'alignement global de deux séquences. C'est le premier à avoir utilisé une approche dynamique. La première étape consiste à construire une matrice des scores élémentaires (valeur déterminée pour chaque élément par une matrice de substitution telles que PAM, BLOSUM, *etc*). Puis, on calcule

$$S(i,j) = MAX[S(i-1,j) + \delta(-,j)],\ [S(i-1,j-1) + \delta(i,j)],\ [S(i,j-1) + \delta(i,-)]$$

où $S(i,j)$ est le score obtenu pour la case d'indice i et j, $\delta(i,j)$ est le score suivant la matrice de substitution et $\delta(-,j)$ ou $\delta(i,-)$ sont les scores pour une brèche.
Enfin, pour déterminer l'alignement global optimal, il suffit de trouver le score sommé maximum dans la matrice transformée. Celui-ci indique le score associé au chemin

(alignement) optimal. L'alignement lui-même est obtenu grâce à la méthode du *back tracking* qui remonte à travers la matrice (de la position finale à la position initiale) et reconstitue le chemin des choix qui ont été à l'origine de cette valeur finale. Un exemple d'alignement de deux séquences par cette méthode est présenté à la (figure 1.11).

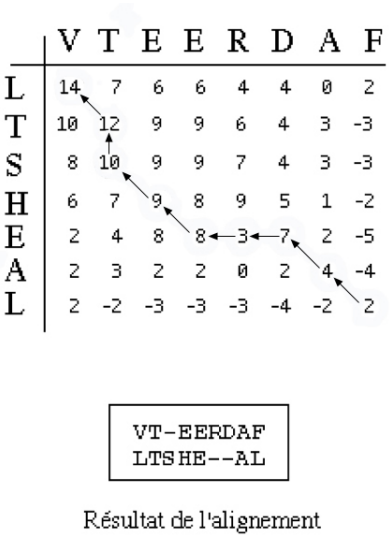

FIG. *1.11: Exemple d'alignement de deux séquences protéiques (LTSHEAL et VTEERDAF) par la méthode de Needleman et Wunsch. Le système de score utilisé est la matrice PAM250 (figure 1.10), les brèches valant 0. Une flèche en diagonale indique un appariement ou une substitution, une flèche verticale correspond à une insertion sur la séquence verticale et une flèche horizontale est une insertion sur la séquence horizontale. Le chemin formé par les flèches permet de reconstruire cet alignement, lequel est présenté sous la matrice. Il n'existe pour cet exemple qu'un seul alignement optimal.*

FASTA (Pearson & Lipman, 1988) est un logiciel de comparaison de séquences avec insertions/délétions qui utilise une méthode heuristique. Il produit un alignement global en joignant des alignements locaux. Ce programme commence par repérer les régions de plus dense identité avec la séquence recherchée. Puis il fait une recherche de similarité sans insertion/délétion en recalculant le score des régions de plus forte identité. Il essaie ensuite de joindre les régions précédentes afin d'étendre la meilleure similarité. Enfin l'alignement optimal des deux séquences est effectué par programmation dynamique en considérant uniquement les régions définies précédemment.

De part leur principe, les algorithmes de recherche de similarité globale vont donner des alignements de moins bonne qualité dans le cas de séquences très divergentes. De plus, dans beaucoup de cas, les homologies entre séquences sont réduites à des régions limitées des séquences puisque beaucoup de protéines résultent d'une combinaison de segments qui ont été ré-agencés lors de l'évolution.

Des algorithmes de recherche de similarité se basant sur de courtes régions pour calculer une similarité locale ont donc été développés. Ces méthodes ont l'avantage d'être beaucoup plus rapides.

L'algorithme de Smith et Waterman (Smith & Waterman, 1981) est directement inspiré de celui de Needleman et Wunsch. La principale différence est que n'importe quelle case de la matrice de comparaison peut être considérée comme point de départ pour le calcul des scores sommes et que tout score somme qui devient inférieur à zéro stoppe la progression du calcul des scores sommes. La case pointée est alors réinitialisée à zéro et peut être considérée comme nouveau point de départ. La transformation de la matrice se calcule en utilisant la formule suivante :

$$S(i,j) = MAX0,\ [S(i-1,j) + \delta(-,j)],\ [S(i-1,j-1) + \delta(i,j)],\ [S(i,j-1) + \delta(i,-)]$$

où $S(i,j)$ est le score obtenu pour la case d'indice i et j, $\delta(i,j)$ est le score suivant la matrice de substitution et $\delta(-,j)$ ou $\delta(i,-)$ sont les scores pour une brèche.

BLAST (Altschul *et al.*, 1990; Altschul *et al.*, 1997) est un programme de recherche de similarité locale qui utilise une méthode heuristique. Il débute par la recherche de tous les petits segments (mots) de longueur W dans la séquence. Puis chacun de ces segments est comparé avec les séquences de la banque afin d'identifier les identités strictes. Enfin, ces identités strictes servent de point d'ancrage à partir desquels l'alignement est étendu dans les deux directions le long de chaque séquence afin d'améliorer le score cumulé. La signification des alignements est évaluée statistiquement en fonction de la longueur et de la composition de la séquence, de la taille de la banque et de la matrice de scores utilisée. BLAST présente ses résultats en une liste de séquences ayant un alignement significatif, chacune associée à un score (score dérivé du score brut de l'alignement) et une *E-value* (nombre d'alignements différents que l'on peut espérer trouver dans les banques avec un score supérieur ou égal au score d'alignement obtenu) : plus la *E-value* est faible, plus le score de l'alignement est significatif. L'algorithme initial de BLAST ne permet ni les insertions ni les délétions, mais il est très rapide et il attribue une valeur statistique au score obtenu. L'algorithme initial a été modifié plusieurs fois pour répondre à différents besoins. Ainsi, BLAST2 est une version de BLAST qui autorise les insertions et les délétions (mais la statistique n'est plus exacte) alors que PSI-BLAST est une version qui construit des profils à partir d'alignements itératifs. Des filtres ont été également conçus pour éliminer les régions répétitives et segments de "faible complexité" qui bruitent les résultats.

L'algorithme de Smith et Waterman n'utilise pas d'approximation et il est plus sensible que BLAST et FASTA. Il permet donc d'obtenir un alignement optimal exact entre régions locales. Cependant son temps d'exécution est long. Par ailleurs, FASTA est plus sensible que BLAST pour la recherche dans les banques nucléiques mais il est beaucoup plus long.

3.2 Les alignements multiples

Lorsqu'une recherche de similarité d'une séquence est effectuée par rapport à l'ensemble des séquences d'une banque de données, il est fréquent de trouver plusieurs séquences présentant une forte similarité avec la séquence étudiée. Pour pouvoir comparer simultanément toutes ces séquences entre elles, il faut aligner ces séquences ensemble et construire ainsi un alignement multiple. L'alignement multiple a de nombreuses applications qui ne sont pas possibles en comparant deux séquences telles que l'identification de positions et d'acides aminés importants, la visualisation de domaines, la construction de phylogénies (section 3.3) ou une aide à la modélisation, certains algorithmes de prédiction de structures secondaires exploitant les alignements multiples (Geourjon & Deleage, 1995).

Le problème de l'alignement multiple est complexe : il s'agit d'aligner plusieurs séquences dans leur intégralité. Plusieurs méthodes d'alignement existent mais aucune n'est parfaite. Les méthodes qui garantissent les meilleurs alignements demandent un temps d'exécution trop long et un espace mémoire trop important pour une utilisation sur un grand nombre de séquences. En effet, la méthode par programmation dynamique de Needleman et Wunsch peut être étendue à N séquences et permet d'obtenir un alignement multiple optimal. Le programme MSA (Lipman *et al.*, 1989) est une implémentation d'un algorithme (Carrillo & Lipman., 1988) généralisant l'approche par programmation dynamique. Cependant, s'il y a plus d'une dizaine de séquences, le calcul devient impossible à cause de la croissance exponentielle du nombre d'alignements. D'autres méthodes ont été développées en utilisant des heuristiques afin d'améliorer la rapidité d'exécution et pour diminuer l'espace mémoire requis, mais ces méthodes ne garantissent pas de trouver l'alignement optimal. Parmi les méthodes d'alignement les plus communes, certaines se basent sur l'alignement progressif (section 3.2.1), d'autres sur des approches itératives ou encore sur la consistance (section 3.2.2). Elles peuvent également construire un alignement global ou un alignement local.

3.2.1 Les méthodes d'alignement progressif

La méthode la plus couramment utilisée est l'approche heuristique qui se base sur l'alignement progressif (Feng & Doolittle, 1987). Cette méthode est très rapide, requiert un espace mémoire peu important et offre de bonnes performances avec des séquences relativement bien conservées. Le principe de l'alignement progressif est de construire un alignement multiple en utilisant des alignements par paires. Il s'effectue en 3 étapes :

- calcul du score d'alignement ou de la distance entre toutes les paires de séquences
- construction d'un arbre guide à l'aide des distances d'alignements calculées précédemment, ce qui représente les relations de parenté entre les séquences
- alignement des séquences en suivant l'arbre guide

Plusieurs programmes d'alignement multiple global utilisent cette approche tels que CLUSTAL W (Thompson *et al.*, 1994), MULTALIN (Corpet, 1988), MUSCLE (Edgar, 2004), *etc.* Ces différents programmes diffèrent sur la méthode utilisée dans l'une des trois étapes.

CLUSTAL W permet de choisir, à la première étape, entre une programmation dynamique ou des méthodes heuristiques. Il utilise, pour la construction de l'arbre guide, l'algorithme de Neighbor-Joining (Saitou & Nei, 1987) (section 3.3). Enfin pour la troisième étape, CLUSTAL W utilise des profils (matrices consensus) pour aligner les séquences en suivant l'arbre guide.

MULTALIN ajoute une boucle de programmation : après l'alignement des séquences suivant l'arbre guide, l'algorithme revient à la première étape de construction d'une matrice des scores de toutes les paires de séquences de l'alignement, puis l'arbre guide est reconstruit ; s'il correspond au précédent, l'algorithme est arrêté, sinon on refait l'alignement, puis la matrice et l'arbre guide. L'algorithme boucle tant que l'on n'obtient pas une convergence.

Un alignement par MUSCLE s'effectue en trois étapes. Il ajoute deux itérations à l'algorithme après l'alignement progressif. Dans la première itération, il s'agit d'améliorer l'arbre obtenu précédemment. Un calcul de matrice de distance est effectué à partir de l'alignement obtenu par l'alignement progressif, en utilisant une méthode différente de celle utilisée pour le premier calcul de la matrice de distance. Puis l'arbre guide et l'alignement correspondant sont construits. Si l'arbre correspond au précédent alors on passe à la deuxième itération, sinon on recommence l'opération. La deuxième itération correspond à un perfectionnement de l'alignement : l'arbre obtenu à partir de l'alignement de la première itération est coupé en deux sous-arbres, puis le profil de l'alignement multiple de chaque sous-arbre est calculé, un nouvel alignement est alors re-calculé à partir des profils, enfin si le score de l'alignement est amélioré alors celui-ci est gardé sinon il est abandonné. Le programme boucle ainsi jusqu'à obtenir une convergence ou le nombre de fois que l'utilisateur l'a souhaité. MUSCLE propose deux variantes permettant d'aligner des séquences plus rapidement : MUSCLE-prog et MUSCLE-fast. MUSCLE-prog effectue uniquement les deux premières étapes *i.e.* l'alignement progressif et la première itération. MUSCLE-fast, quant à lui, n'effectue que la première étape *i.e.* l'alignement progressif. MUSCLE-fast est plus rapide que MUSCLE-prog mais il perd en qualité au niveau de l'alignement. Les performances de MUSCLE ainsi que de MUSCLE-prog et MUSCLE-fast ont été testées sur plusieurs ensembles d'alignements de référence comme BAliBASE (Thompson *et al.*, 1999; Bahr *et al.*, 2001). Ces méthodes paraissent associer précision et rapidité, les deux variantes de MUSCLE permettant d'aligner un grand nombre de séquences très rapidement.

Ces programmes permettent également d'ajouter rapidement une séquence ou un ensemble de séquences à un alignement pré-calculé en utilisant le principe d'alignement de profils (Gribskov *et al.*, 1987). Ce principe consiste à déduire, à partir de l'alignement, une matrice d'occurences de chaque caractère à des positions spécifiques de la séquence.

Il en résulte une séquence consensus. Le profil peut alors être utilisé pour être aligné avec un autre profil ou une séquence.

Parmi les programmes utilisant l'approche progressive, il existe également le logiciel MENTALIGN (Dufayard, 2004) ou encore l'algorithme plus récent POA (Lee *et al.*, 2002; Grasso & Lee, 2004). MENTALIGN est un algorithme incrémental qui a été spécifiquement développé afin de permettre le calcul de grands alignements contenant des milliers de séquences. Il permet d'ajouter, rapidement, une à une, des nouvelles séquences à un alignement en utilisant un arbre guide. POA, quant à lui, utilise une représentation en graphe d'un alignement multiple de séquences (PO-MSA), qui peut lui-même être aligné directement par un programme dynamique par paire, éliminant la nécessité de réduire l'alignement multiple de séquences à un profil.

3.2.2 D'autres approches

D'autres programmes se basent sur des méthodes différentes. T-COFFEE (Notredame *et al.*, 2000; Notredame *et al.*, 1998; Poirot *et al.*, 2003) permet la combinaison de collections d'alignements par paires et multiples, globales ou locales dans un seul modèle. Il permet également d'estimer le niveau de consistance de chaque position à l'intérieur du nouvel alignement avec le reste des alignements. MABIOS (Abdeddaïm, 1997) utilise des algorithmes d'alignement par blocs : il sélectionne des blocs (alignement de segments de même longueur de séquences sans caractère nul) compatibles dans les séquences puis aligne les séquences entre les blocs. Il permet également d'ajouter rapidement une séquence à un alignement déjà existant. Il existe encore beaucoup d'autres algorithmes d'alignement tels que DIALIGN (Morgenstern, 2004) qui combine les méthodes locales et globales et utilise la comparaison segment par segment, MAFFT (Katoh *et al.*, 2002; Katoh *et al.*, 2005) qui est une implémentation d'une méthode basée sur les Transformés de Fourrier rapides (FFT), permettant ainsi une détection rapide de segments homologues ou encore plus récemment PROBCONS (Do *et al.*, 2005) qui est une méthode basée sur la consistance, comme T-COFFEE, utilisant une librairie de pair-HMM (pair Hidden Markov Models) (Durbin *et al.*, 1988) et M-COFFEE (Wallace *et al.*, 2006), extension de T-COFFEE, qui combine plusieurs méthodes d'alignement multiple.

3.2.3 Les différents formats d'alignement

Il existe divers formats pour représenter un alignement dans les programmes d'alignement. Parmi ces formats, on peut citer les formats CLUSTAL, FASTA, PHYLIP, GCG/MSF, NBRF/PIR, MASE, NEXUS, *etc.* Chaque format ajoute des informations spécifiques en début ou fin de fichier et organise les séquences de leur propre manière. Par exemple le format CLUSTAL regroupe les séquences alignées en bloc de longueur fixée où chaque ligne correspond à une séquence alors que dans le format FASTA, les séquences alignées sont en entier, les unes sous les autres avec pour chacune une première ligne de

description (figure 1.12). Le format PHYLIP est celui utilisé pour les données en entrée des programmes du package PHYLIP qui propose un ensemble d'outils pour faire de la reconstruction phylogénétique (section 3.3).
Certains de ces formats correspondent aux formats de stockage des séquences dans les banques comme le format FASTA ou le format PIR.

```
>ACYP1_DROME 119
ATHNVHSCEFEVFGRVQGVNFRRHALRKAKTLGLRGWCMNSSRGTVKGYIEGRPAEMDVM
KEWLRTTGSPLSSIEKVEFSSQRERDRYGYANFHIKPDPHENRPVHEGLGSSSSHHDSN
>H6ST1_CHICK 408
MKRAGRTMVERTSKFLLIVAASVCFMLILYQYVGPGLSLGAPSGRPYAEEPDLFPTPDPH
YVKKYYFPVRELERELAFDMKGEDVIVFLHIQKTGGTTFGRHLVQNVRLEVPCDCRPGQK
KCTCYRPNRRETWLFSRFSTGWSCGLHADWTELTNCVPGVLGRRESAPNRTPRKFYYITL
LRDPVSRYLSEWRHVQRGATWKTSLHMCDGRTPTPEELPSCYEGTDWSGCTLQEFMDCPY
NLANNRQVRMLADLSLVGCYNMSFIPENKRAQILLESAKKNLKDMAFFGLTEFQRKTQYL
FERTFNLKFIRPFMQYNSTRAGGVEVDNDTIRRIEELNDLDMQLYDYAKDLFQQRYQYKR
QLERMEQRIKNREERLLHRSNEALPKEETEEQGRLPTEDYMSHIIEKW
```

FIG. *1.12: Exemple de deux séquences au format FASTA*

3.3 Les arbres phylogénétiques

Pour cette partie, nous avons utilisé des informations provenant de (Graur & Li, 2000).

Afin d'étudier l'histoire évolutive d'un ensemble de séquences homologues, il est indispensable de déterminer l'arbre phylogénétique de ces séquences. Un arbre phylogénétique est une représentation graphique des relations de parenté qui existent entre des organismes (figure 1.13).

Il représente l'évolution d'un groupe de taxons. Les nœuds internes de l'arbre représentent les séquences ancestrales. Les feuilles (ou noeuds externes ou terminaux) sont les séquences connues et représentent les unités taxonomiques (OTU pour Operational Taxonomic Unit). Les branches correspondent aux lignées évolutives. L'ensemble des branches de l'arbre est appelé la topologie. Il existe plusieurs représentations : l'arbre peut être raciné (l'ancêtre commun à toutes les feuilles a été placé) et être ainsi orienté ou il peut être non raciné et donc non orienté. Lorsque l'arbre est orienté cela signifie qu'il existe une évolution temporelle et les longueurs plus ou moins longues des branches correspondent à la mesure de la quantité d'évolution. Cette quantité d'évolution s'exprime en nombre de substitutions par site. La plupart des méthodes de construction d'arbres phylogénétiques donnent des arbres non racinés. Il faut par la suite raciner l'arbre obtenu. La plupart du temps, pour construire un arbre phylogénétique, des séquences homologues alignées sont utilisées.

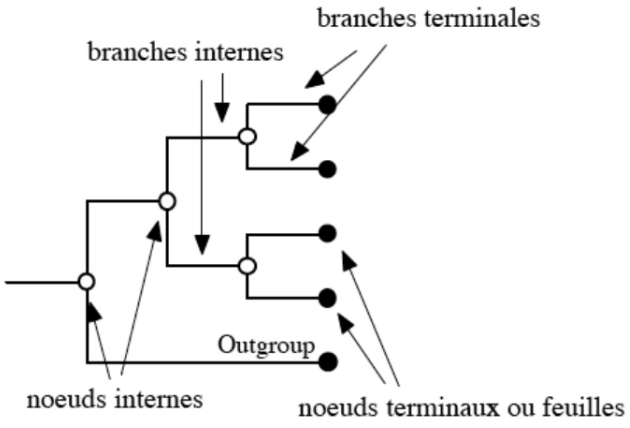

FIG. *1.13: Un arbre phylogénétique. L'outgroup ou groupe externe permet de raciner l'arbre et d'orienter les événements dans l'arbre (section 3.3.2). Extrait de (Daubin, 2002)*

Enfin il faut savoir que l'histoire évolutive des gènes reproduit celle des espèces qui les portent sauf si il y a eu des transferts horizontaux, c'est-à-dire des transferts de gènes entre espèces ou si il y a eu des duplications géniques. Dans ce cas, l'arbre des espèces peut être différent de l'arbre des gènes.

3.3.1 Les différentes méthodes

Il existe trois grands types de méthodes permettant la reconstruction d'arbres phylogénétiques :

- les méthodes basées sur les distances évolutives,
- les méthodes par parcimonie,
- les méthodes de maximum de vraisemblance

Les méthodes de distances sont des méthodes de reconstruction d'arbres phylogénétiques sans racine basées sur les mesures de distances entre séquences prises deux à deux, c'est-à-dire le nombre de substitutions de nucléotides ou d'acides aminés entre ces deux séquences. Elles permettent de reconstruire des arbres en partant des ressemblances observées entre chaque paire d'unités évolutives. On parle de la ressemblance globale établie à partir du maximum d'observations disponibles.

Deux étapes sont nécessaires pour la reconstruction d'arbre en utilisant les méthodes de distances :

- Calcul d'une matrice de distances à partir de l'alignement de départ.
- Construction de l'arbre phylogénétique en utilisant la matrice de distances

Le calcul de distances peut se faire simplement en comparant les séquences et en évaluant leur similitude et leur différence (distance observée). Il est également possible de calculer une distance évolutive en utilisant des modèles évolutifs qui permettent de prendre en compte les processus évolutifs qui agissent sur les séquences. En effet, un

simple comptage des différences entre deux séquences ne donne pas forcément le nombre réel d'événements de mutation, il peut y avoir des substitutions cachées ou multiples. Ainsi il existe plusieurs modèles évolutifs dont les plus connus sont celui de Jukes-Cantor (Jukes & Cantor, 1969) ou celui de Kimura à deux paramètres (Kimura, 1980).

Plusieurs méthodes de distances ont été développées et permettent de construire un arbre phylogénétique à partir d'une matrice de distance. Parmi ces méthodes, il existe les algorithmes UPGMA (Unweighted Pair Group Method with Arithmetic mean) et Neighbor-Joining (Saitou & Nei, 1987). Le principe de UPGMA est de regrouper ensemble les séquences les plus proches en faisant l'hypothèse d'horloge moléculaire. Cette hypothèse suppose que toutes les lignées évoluent à la même vitesse depuis leur divergence, c'est-à-dire depuis que leur dernier ancêtre commun a subi une spéciation et qu'il y a eu séparation en deux espèces distinctes. La méthode Neighbor-Joining, quant à elle, permet de construire l'arbre dont la somme des longueurs des branches est minimale en recherchant les séquences les plus proches à chaque étape de regroupement sans impliquer l'hypothèse d'horloge moléculaire. Les méthodes de distances sont rapides et donnent de bons résultats pour des séquences ayant une forte similarité.

Différents programmes de construction phylogénétique ont été développés et utilisent ces méthodes de distances. Parmi ces logiciels, FASTME (Desper & Gascuel, 2002) est basé sur le principe du minimum d'évolution qui consiste à retenir l'arbre dont la longueur estimée de sa topologie (somme des longueurs des branches) est la plus petite. BIONJ (Gascuel, 1997) et QUICKTREE (Howe *et al.*, 2002) sont des implémentations de la méthode Neighbor-Joining qui est une approximation du principe du maximum d'évolution. Ces programmes garantissent une grande rapidité d'exécution même si la qualité de l'arbre n'est pas optimale.

Les méthodes par parcimonie et celles de maximum de vraisemblance sont basées sur les caractères et s'intéressent au nombre de mutations (substitutions / insertions /délétions) qui affectent chacun des sites de la séquence.

La parcimonie consiste à minimiser le nombre de substitutions nécessaires pour passer d'une séquence à une autre dans une topologie de l'arbre. Les méthodes de maximum de parcimonie recherchent toutes les topologies possibles afin de trouver l'arbre optimal c'est-à-dire l'arbre dont la topologie requiert le nombre minimal de substitutions nécessaires totalisé sur l'ensemble des sites pour passer d'une séquence à une autre de la topologie. Ces méthodes sont relativement lentes car le temps nécessaire pour cette exploration croit rapidement avec le nombre de séquences. Un des algorithmes développant la méthode du maximum de parcimonie est l'algorithme de Fitch (Fitch, 1977).

Les méthodes de maximum de vraisemblance recherchent l'arbre dont la topologie est la plus vraisemblable étant donnés les séquences et le modèle d'évolution des séquences choisi. Ces méthodes évaluent, en terme de probabilités, l'ordre des branchements et la longueur des branches d'un arbre sous un modèle évolutif donné. Elles ne comparent pas

les données deux à deux, mais estiment la vraisemblance de chaque site pour la topologie au regard du modèle évolutif choisi. La topologie choisie par la méthode sera celle qui maximise la vraisemblance de l'alignement. Ces méthodes nécessitent des temps de calcul très importants.

Plusieurs packages de programmes proposent des implémentations des méthodes de parcimonie et de maximum de vraisemblance comme par exemple PHYLIP (Felsenstein, 1989), PAUP (Swofford, 2003) ou le programme PHYML (Guindon & Gascuel, 2003).

3.3.2 Enraciner un arbre

Lorsqu'un arbre n'est pas raciné, cela signifie que l'ancêtre commun à toutes les feuilles n'a pas été trouvé. Etant donné qu'un arbre phylogénétique est un arbre d'évolution, il ne peut être considéré en tant que tel que lorsqu'il est raciné. Afin de pouvoir obtenir des informations au niveau évolutif, il faut donc raciner l'arbre et définir l'emplacement du gène ancestral commun. Pour cela, il existe différentes méthodes : la méthode de l'outgroup, du point médian ou le racinement par un paralogue.

La méthode de l'*outgroup* consiste à ajouter aux séquences traitées, avant la construction de l'arbre, un groupe externe (outgroup) correspondant à une séquence éloignée. La divergence entre le groupe externe et les autres séquences doit être antérieure à la divergence entre les séquences traitées. Ainsi le nœud-racine sera placé entre la séquence ajoutée et le noeud connectant cette séquence aux autres. Le groupe externe ne doit pas être trop éloigné des séquences traitées sinon cela peut impliquer des erreurs de topologie car il est difficile dans ce cas d'estimer les distances entre le groupe externe et les autres séquences. De plus, il existe un risque d'attraction des longues branches avec l'outgroup (Philippe & Germot, 2000). L'attraction des longues branches est un artéfact de reconstruction qui se traduit par le mauvais positionnement des séquences évoluant plus vite ou moins vite que la moyenne. En particulier, les séquences qui évoluent plus vite apparaissent d'émergence beaucoup trop précoce dans les phylogénies. Le groupe externe ne doit pas non plus être trop proche car dans ce cas ce n'est pas un vrai groupe externe et la séquence ajoutée en tant que groupe externe risque de se retrouver parmi les séquences traitées dans l'arbre phylogénétique. Enfin, plusieurs groupes externes peuvent être utilisés afin d'améliorer la topologie de l'arbre.

Dans le cas où il n'y a pas de groupe externe qui puisse permettre un racinement de l'arbre, il est possible de positionner la racine par une méthode mathématique en faisant l'hypothèse que la vitesse d'évolution tend à être uniforme sur l'ensemble des branches de l'arbre (hypothèse de l'horloge moléculaire). C'est la méthode du point médian (mid-point rooting) qui consiste à positionner la racine approximativement à égale distance de toutes les feuilles.

Enfin, s'il existe une duplication ancestrale clairement identifiée, il est possible d'utiliser un gène paralogue pour raciner l'arbre phylogénétique.

3.3.3 Evaluer la qualité d'un arbre

Lorsqu'un arbre phylogénétique est construit, il faut évaluer la fiabilité de cet arbre c'est-à-dire évaluer celle de chaque branche interne de l'arbre puisque l'information phylogénétique réside dans l'ensemble de ses branches internes. Pour cela, plusieurs méthodes statistiques sont disponibles. La méthode la plus couramment utilisée est le bootstrap. Cette méthode part du principe que les caractères évoluent de manière indépendante. Elle a été inventée par Bradley Efron en 1979 et introduite en phylogénie par Felsenstein en 1985 dans le package PHYLIP. Cette méthode consiste à réaliser, à partir d'un alignement d'origine, un grand nombre (au moins 100) d'alignements "artificiels" de même taille par tirage aléatoire avec remise. Puis, pour chaque alignement "artificiel" obtenu, l'arbre est reconstruit et comparé à l'arbre d'origine. Pour chaque branche interne de l'arbre d'origine, on compte le nombre de fois où celle-ci est présente dans les arbres obtenus artificiellement. Cette fréquence avec laquelle on retrouve un sous-arbre est la valeur de bootstrap (figure 1.14). Elle indique la robustesse statistique de la branche interne. Ainsi plus cette valeur est élevée plus la fiabilité de la branche est importante.

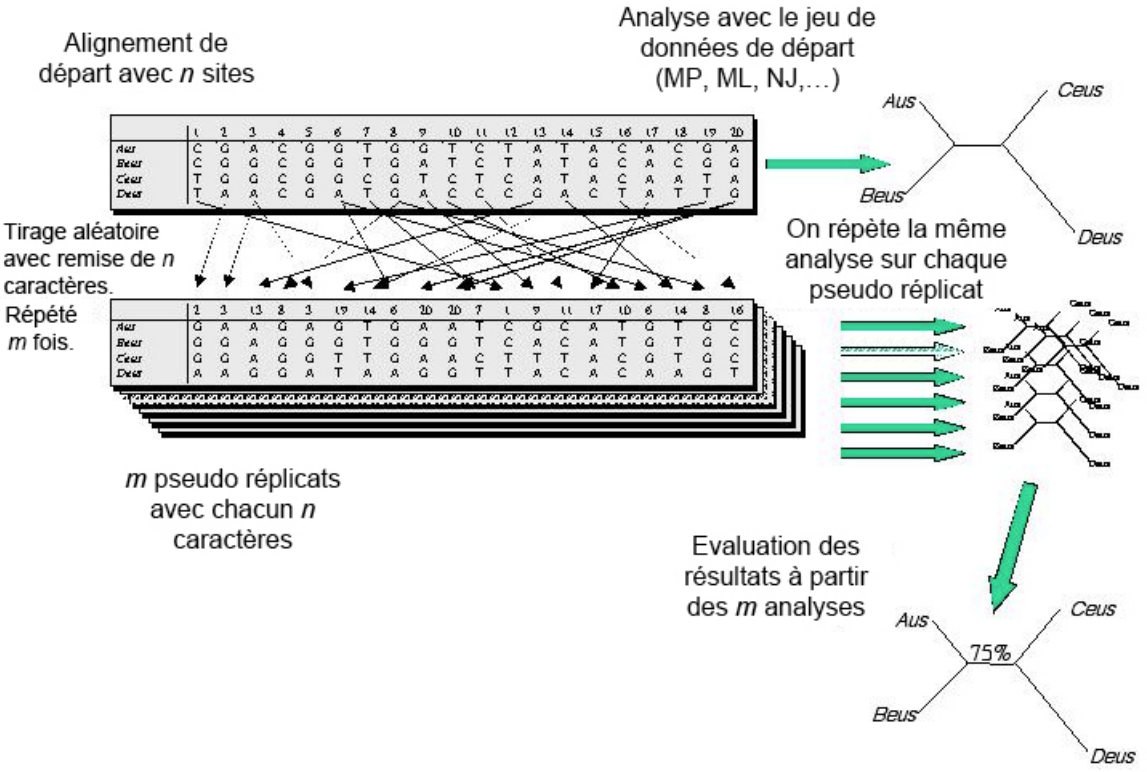

FIG. *1.14: Principe du bootstrap. Extrait de (Calteau, 2005).*

3.3.4 Les formats d'arbres

Il existe différents formats pour représenter les arbres phylogénétiques tels que les formats Newick ou Nexus. La plupart des programmes de phylogénie utilisent le format Newick créé par Cayley (1857), maintenant devenu un format standard. Ce format représente les arbres sous forme linéaire par une série de parenthèses imbriquées, regroupant les noms des séquences et séparés par des virgules (figure 1.15). Dans ce format, la configuration décrite par les parenthèses représente la topologie de l'arbre.

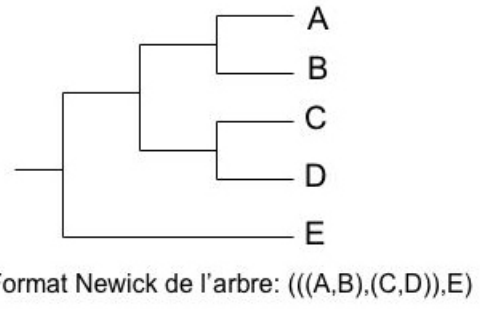

FIG. *1.15: Exemple de format Newick*

Chapitre 2

L'identification

L'identification est une des principales applications de la systématique qui est constituée de deux disciplines, la taxonomie et la nomenclature. La taxonomie est la science qui permet de classer rationnellement les organismes vivants en groupes d'affinité. Cette discipline est étroitement liée à l'identification puisque le but de cette dernière est de placer un nouvel individu dans la classification existante. La nomenclature, quant à elle, a pour objectif de donner un nom à chaque groupe défini dans la taxonomie selon des règles publiées. Cela permet d'unifier le langage scientifique et d'améliorer la communication entre les utilisateurs.

1 La taxonomie

1.1 Une brève histoire

Le mot taxinomie ou taxonomie (taxis = arrangement et nomos = usage, règlement) a été proposé en 1813 par le botaniste suisse Augustin-Pyramus de Candolle afin de désigner la science des lois de la classification des formes vivantes. Carl von Linné (1707 - 1778) posa les fondations de la systématique, et fut l'auteur d'une classification dont les grands principes furent la base de la systématique scientifique jusqu'au milieu du XX^{ème} siècle. Linné codifie les différents niveaux hiérarchiques ou taxons et définit les sept rangs traditionnels : règne, embranchement, classe, ordre, famille, genre, espèce. Cette classification traditionnelle fait encore, en ce début du XXI^{ième} siècle, partie du bagage culturel commun. En 1859, Charles Darwin recommande une classification purement généalogique. S'il y a eu évolution, les espèces doivent être classées selon leur degré d'apparentement évolutif. Il arrive ainsi à l'idée-clé de descendance avec modification, proposée dans *De l'origine des espèces* (1859), et qui structure désormais la pensée phylogénétique. Il a donc fallu attendre Darwin pour comprendre que l'ordre de la Nature est le reflet de l'histoire évolutive des organismes sur Terre. Trouver la classification naturelle et retrouver cette histoire correspond donc à la même chose. Cependant, après cette percée conceptuelle fondamentale, il faudra attendre près d'un siècle pour que celle-ci devienne opérationnelle, et d'abord pour accepter la généalogie comme inaccessible (qui descend de qui ?) pour mieux se concen-

trer sur la phylogénie (qui est plus proche parent de qui ?). Dans la deuxième moitié du XX^{ème} siècle est apparue l'approche phylogénétique pour laquelle le critère fondamental du choix de la classification est qu'elle doit refléter strictement la phylogénie, c'est-à-dire les degrés d'apparentement entre espèces. La notion même d'une telle phylogénie est une conséquence de la théorie de l'évolution, et le succès prédictif des arbres phylogénétiques une des preuves de cette théorie. La taxonomie est donc une discipline de synthèse en constante évolution. Le but ultime est de mettre au point une classification naturelle des espèces pour dégager les entités évolutives, en tenant compte des rapports existant entre elles et de leur degré de complexité. Des bouleversements interviennent périodiquement dans la dénomination et l'ordre des groupes taxonomiques ou taxons.

1.2 Les classifications

Une classification est représentée par un classement arborescent (un arbre). Cet arbre part d'une racine et inclut des êtres vivants existant ou ayant existé, jusqu'aux individus. Chaque nœud définit un taxon, qui groupe tous les sous-taxons qu'engendre le nœud. Le terme classification est souvent donné comme synonyme de taxonomie. La classification permet donc d'établir selon des critères de similarités, une hiérarchie des organismes dont les niveaux correspondent à des groupes taxonomiques. Il existe différentes classifications dont la classification phénotypique et la classification phylogénétique.

1.2.1 La classification phénotypique

La classification traditionnelle basée sur des traits phénotypiques est issue de la classification du vivant établie par Linné. Cette classification utilise la comparaison de caractères considérés comme importants tels que des caractéristiques morphologiques observables ou l'habitat. Elle divise le monde du vivant en cinq règnes : les procaryotes (bactéries et archées), les protistes (eucaryotes unicellulaires), les champignons (eucaryotes multicellulaires), les végétaux (eucaryotes multicellulaires) et les animaux (eucaryotes multicellulaires). Cependant une telle classification ne reflète qu'une quantité d'information réduite car les critères ne sont pas tout le temps suffisamment précis pour discerner les différentes espèces. De plus, le choix des critères qualifiés "d'importants" est subjectif et il peut varier d'un auteur à un autre ce qui est une source potentielle d'instabilité.

1.2.2 La classification phylogénétique

La classification phylogénétique, initiée par Willi Hennig en 1950, est basée sur les caractères anatomiques des êtres vivants analysés différemment et par la suite sur les caractères moléculaires. Cette classification permet de mieux visualiser les embranchements du vivant tels que ceux constitués par différenciations progressives au cours du temps. Le principe est de classer les organismes selon les similarités ou les différences qu'il existe entre leurs séquences d'acides nucléiques. Une des grandes évolutions de l'approche phylogénétique est que cette classification illustre les principes d'évolution et de parenté des espèces

alors que celle développée par Linné était basée sur l'hypothèse que toutes les espèces sont apparues en même temps et que celles-ci étaient fixes.

2 L'identification de séquences

2.1 Définition

Le but de l'identification est d'attribuer une unité taxonomique inconnue à un groupe taxonomique défini au préalable dans une classification préétablie. La classification est donc indispensable à l'identification de nouveaux individus puisque l'assignation à un groupe ne peut se faire que si le groupe a déja été décrit. Ainsi pour identifier un nouveau taxon ou une nouvelle séquence, il faut trouver le taxon connu le plus proche de celui à identifier. L'identification ne peut se faire qu'en utilisant la classification existante qui constitue une base de connaissance nécessaire. Les méthodes utilisées pour l'identificaion et la robustesse du résultat sont donc intimement liées aux travaux réalisés en amont. L'identification est utilisée dans de nombreux domaines, tels que la microbiologie, la médecine, l'environnement, *etc.* Dans le domaine médical, les méthodes d'identification sont utilisées pour détecter et reconnaître les micro-organismes impliqués dans des pathologies, ce qui permet ainsi de choisir le traitement le plus approprié. Les méthodes d'identification peuvent également être utilisées dans le domaine agro-alimentaire comme outils de la traçabilité alimentaire. Dans d'autres contextes tels que l'identification d'espèces ou de taxons à partir de marqueurs moléculaires d'organismes environnementaux, la confrontation d'une nouvelle séquence avec une banque de données, ou la mise à jour des banques de séquences, l'assignation d'une nouvelle séquence à une collection est nécessaire. Par exemple, dans les banques de familles de gènes homologues, l'identification d'une séquence inconnue consiste à connaître la famille de gènes homologues à laquelle cette séquence appartient, ce qui permet l'étude de ses relations évolutives.

2.2 Les méthodes d'identification

L'évolution des méthodes d'identification correspond à celle des méthodes de classification. Ainsi on peut distinguer les méthodes reposant sur l'étude phénotypique et celles basées sur des approches moléculaires. Beaucoup de méthodes ont été développées pour l'identification bactérienne et notamment pour le domaine médical.

2.2.1 L'approche phénotypique

De la même manière que la classification, l'identification phénotypique s'appuie sur des études morphologiques, biochimiques, sérologiques, *etc.* Il s'agit de comparer divers caractères phénotypiques du taxon à étudier avec ceux des taxons déjà connus. Les caractères phénotypiques peuvent être de simples observations ou des tests préliminaires mais cela peut également être des tests métaboliques, de la sérologie, *etc.*

Au début du XX^{ème} siècle, le nombre limité de tests biochimiques ou phénotypiques a conduit à une caractérisation des espèces inadéquate et imprécise. En microbiologie, c'est à partir de 1960 qu'apparaît un premier manuel d'identification développé dans les laboratoires de microbiologie clinique (ROCHE). Puis BioMérieux a rapidement commercialisé des galeries API 20E garantissant des résultats en moins de 20 heures à partir de 20 tests différents. En 1978, Microscan est développé et permet une identification et des tests de sensibilité. Enfin, c'est en 1983 que les premiers automates d'identification bactérienne apparaissent sur le marché. C'est le cas du produit VITEK 2 (BioMérieux, Marcy l'Etoile, France) ou du système PHOENIX (Becton Dickinson Microbiology Systems, Cockeysville, Md.) qui automatisent toutes les étapes nécessaires à la réalisation des tests d'identification.

Cependant, les techniques phénotypiques ne sont pas forcément adaptées à une identification exacte et rapide et peuvent prendre un certain temps. En zoologie, par exemple, l'identification morphologique peut nécessiter l'obtention d'individus adultes, ce qui exige l'élevage ardu et coûteux de spécimens. Il faut donc pouvoir accélérer les processus d'identification. Par ailleurs, le nombre de tests phénotypiques est limité et restreint, ainsi, ce type d'identification. La plasticité phénotypique et la variabilité génétique des caractères utilisés pour l'identification phénotypique peuvent mener à des identifications incorrectes. Enfin, étant donné que les clefs d'identification morphologique sont souvent efficaces seulement pour une étape de vie ou un genre particulier, beaucoup d'individus ne peuvent être identifiés et l'utilisation de telles clefs exige souvent un si haut niveau d'expertise que les diagnostics erronés sont communs.

2.2.2 L'approche moléculaire

Les limitations de l'identification phénotypique ont conduit au développement d'une nouvelle approche basée sur l'analyse des séquences : l'identification moléculaire. Le principe est de comparer les séquences d'ADN des taxons à étudier. La majorité des méthodes utilisées actuellement repose sur une amplification pour augmenter la sensibilité. De nombreuses méthodes d'identification moléculaire ont fait l'objet d'expérimentations, en particulier à l'aide de procédés de détection d'acides nucléiques tels que l'hybridation moléculaire et les techniques d'amplification génique *in vitro*. Ces derniers permettent, en produisant un nombre très élevé de séquences nucléiques identiques, d'améliorer la sensibilité des tests. Différentes méthodes d'identification ont été développées telles que le séquençage et, plus récemment, l'utilisation de codes barres génétiques. Par rapport aux approches phénotypiques, ce type d'identification permet d'obtenir des résultats plus précis, objectifs et reproductibles. De plus, il donne la possibilité d'identifier des organismes non cultivables et de faire des analyses (semi-)quantitatives. Il n'y a plus besoin de conditions spécifiques de croissance ou de tests préliminaires. Les séquences peuvent être facilement partagées, transportées ou stockées dans des banques et les taches d'identification peuvent être automatisées. Enfin, l'identification moléculaire permet une approche

phylogénétique et l'analyse de l'évolution des séquences d'ADN.

2.2.2.1 L'hybridation moléculaire

Les méthodes d'identification moléculaire peuvent se baser sur l'hybridation (Meijer *et al.*, 2000). Cette dernière permet de mettre en évidence au sein d'une cellule ou d'un tissu, une séquence d'acide nucléique. C'est une réaction hautement spécifique permettant la détection et l'identification de la séquence recherchée parmi les milliers de séquences d'un génome, ou dans un mélange de séquences de différents organismes. Elle est basée sur le principe de complémentarité des bases nucléiques, plus particulièrement entre l'ADN et le brin d'ARN de séquences complémentaires. L'hybridation moléculaire désigne l'association qui peut avoir lieu entre deux acides nucléiques simples brins de séquences complémentaires et qui conduit à la formation d'un double brin ou duplex. Cette association s'effectue par l'établissement de liaisons hydrogènes spécifiques : deux liaisons entre l'adénine (A) et la thymine (T) (ou l'uracile U) et trois entre la cytosine (C) et la guanine (G). La formation et la stabilité des duplex dépendent de nombreux facteurs en plus de la composition en bases : longueur des duplex, complexité de la séquence, *etc.*

L'hybridation est à la base de nombreuses techniques de biologie moléculaire impliquant la mise en présence d'au moins deux brins simples d'acides nucléiques dans des conditions physico chimiques précises. Le brin, dont on connaît au moins une partie de la séquence, est une sonde, l'autre brin, celui que l'on souhaite caractériser constitue la cible. Une sonde nucléotidique est donc un segment de nucléotides qui permet de rechercher de manière spécifique un fragment d'acide nucléique que l'on désire étudier. Ce peut être une séquence d'ADN ou d'ARN, mais obligatoirement monocaténaire. Sa taille est très variable : oligonucléotide de 20-30 nucléotides ou, à l'opposé, de plusieurs centaines de nucléotides. La sonde est complémentaire et antiparallèle du fragment recherché. Dans un mélange complexe où s'effectue l'hybridation moléculaire, la sonde doit être facilement repérable grâce à un marquage avec un radioisotope (marquage chaud), mais il existe également des sondes sans marquage par un radioisotope (sondes froides).

Les nouveaux développements de l'hybridation ont conduit à de nouvelles applications tel que les puces à ADN (Ye *et al.*, 2001). Le principe de fonctionnement des puces repose sur l'utilisation de sondes (ADN ou ARN). Des sondes moléculaires, le plus souvent des oligonucléotides, sont fixées sur une surface miniaturisée, généralement de l'ordre de quelques centimètres carrés. Lors d'une analyse, un échantillon contenant des fragments d'un acide nucléique cible, est déposé sur la puce à ADN. La mise en présence des séquences cibles marquées et des sondes conduit à la formation par hybridation de duplex. Après une étape de lavage, permettant d'éliminer les cibles non hybridées, une analyse de la surface de la puce permet le repérage des hybridations effectives grâce aux signaux émis par les marqueurs étiquettant la cible. Il résulte de cette analyse une empreinte

d'hybridation qui, par un traitement informatique adéquat, permettra d'accéder à des informations plus ou moins complexes et complètes, telles la présence de fragments particuliers dans l'échantillon, la détermination des séquences, l'étude des mutations, *etc.* Le principe des puces à ADN est présenté à la figure 2.1. Enfin, il existe des sondes PNA (Peptide Nucleic Acid) (Stender *et al.*, 2002). Elles constituent des analogues de l'ADN, leurs propriétés physico-chimiques leur conférant une sensibilité et une spécificité supérieure aux sondes ADN.

Cependant la difficulté de cette technique réside dans l'obtention de la sonde c'est-à-dire d'une séquence courte et spécifique d'une seule espèce. De plus la sensibilité des sondes est limitée et autorise rarement leur emploi pour le diagnostic direct. Ce manque de sensibilité qui, pendant longtemps, a représenté l'obstacle majeur, est à présent résolu avec l'apparition des techniques d'amplification génique.

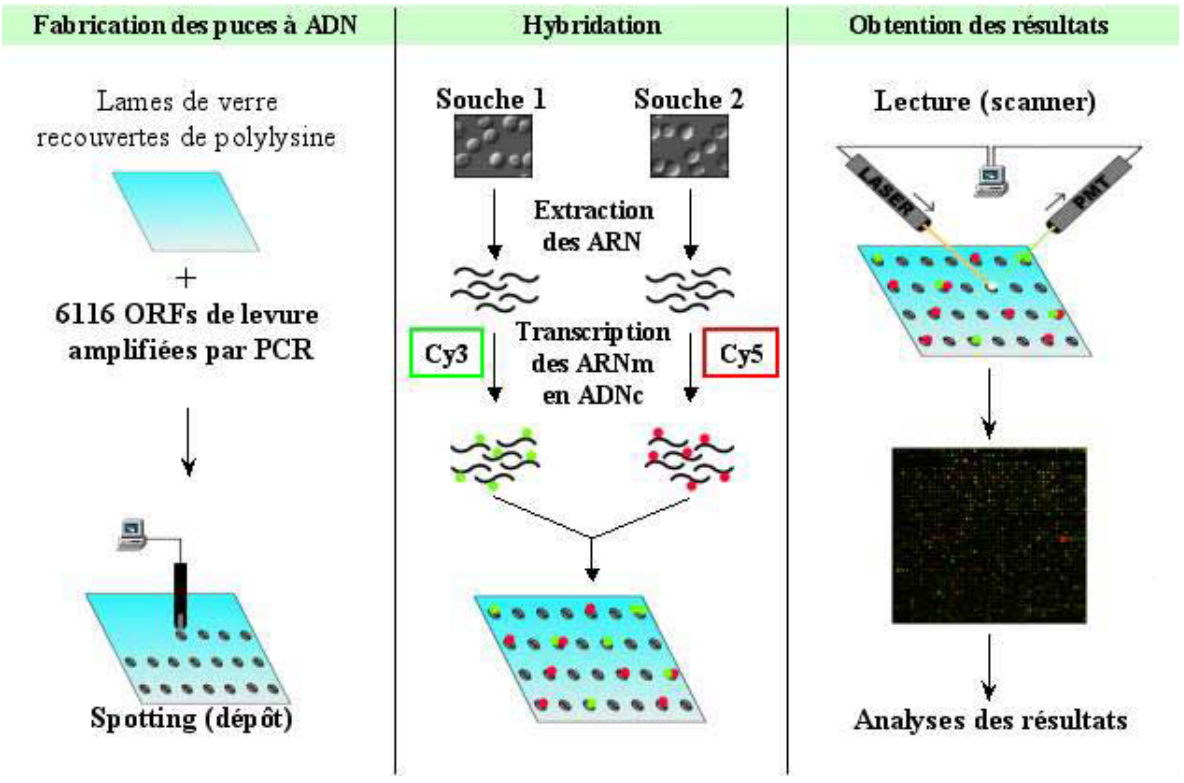

FIG. *2.1: Principales étapes du principe des puces à ADN.*

2.2.2.2 L'amplification génique ou PCR ("polymerase chain reaction")

La PCR est une technologie qui a bouleversée la biologie moléculaire et s'est implantée très rapidement dans les laboratoires. La première publication sur la PCR a été faite en 1986 par K. Mullis (Mullis *et al.*, 1986). En 1988, la première PCR a

été réalisée par R. Saiki (Saiki *et al.*, 1988). Les premiers kits PCR commerciaux, Amplicor concernaient le diagnostic médical, puis ils ont été destinés au diagnostic alimentaire.

Le principe de la réaction d'amplification de gène est très simple. La méthode permet de recopier un segment d'ADN ou d'ARN en de nombreux exemplaires, grâce à une ADN polymérase thermostable extraite de *Thermus aquaticus* (la Taq polymerase) et à deux amorces oligonucléotidiques qui encadrent le segment amplifié. La PCR consiste en une succession cyclique de trois étapes (figure 2.2) :

- La dénaturation thermique : les deux brins d'ADN sont séparés par élévation de la température supérieure à la température de dénaturation de l'ADN.
- L'hybridation des amorces : une amorce sens et une amorce antisens vont s'hybrider sur les brins d'ADN et délimitent ainsi la séquence à amplifier. La température réactionnelle doit être inférieure à la température de fusion des amorces pour permettre leur hybridation.
- L'élongation : la Taq polymérase (ADN polymérase ADN dépendante résistante à température élevée) allonge les amorces en incorporant des désoxynucléotides complémentaires de la séquence de la matrice à laquelle elle est hybridée.

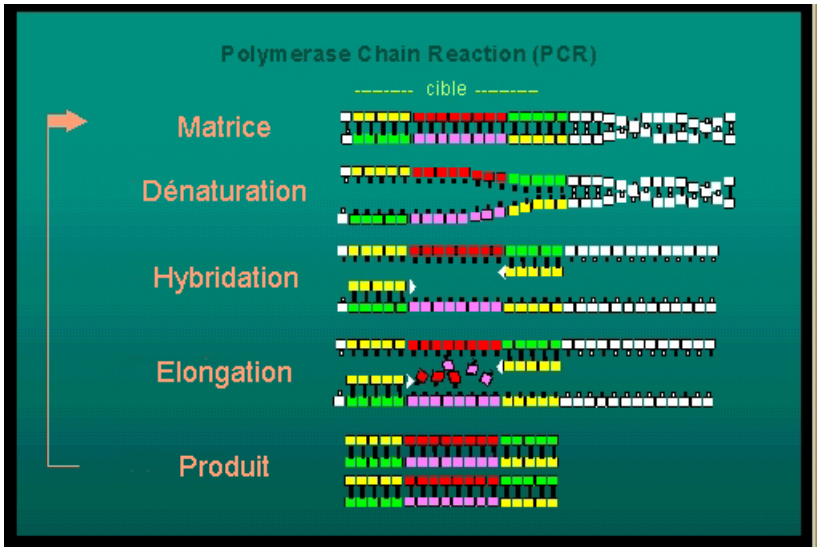

FIG. *2.2: Les différentes étapes de la PCR.*

Cette technique permet donc d'amplifier un fragment d'ADN ce qui permet d'augmenter la sensibilité de détection. Chaque réaction est spécifique et la difficulté de la mise au point des techniques PCR se situe dans le choix de l'amorce et de la zone à amplifier.

Des techniques associées ont été développées comme par exemple :

- La PCR emboîtée ou PCR gigogne (Nested PCR) qui s'effectue en deux étapes successives, avec deux couples d'amorces différents, le second liant des séquences situées à l'intérieur du premier amplicon. Elle permet une meilleure sensibilité du

résultat.
- La PCR multiplexe (Multiplex PCR) qui permet d'amplifier plus d'un amplicon à la fois, généralement en ajoutant un couple d'amorces par type désiré.
- La PCR en temps réel (Real-time PCR) ou Q-PCR (Quantitative PCR) qui consiste à mesurer la quantité d'ADN polymérisé à chaque cycle grâce à un marqueur fluorescent.
- La RT-PCR (RT-PCR pour Reverse Transcriptase PCR) qui associe une RT (synthèse du brin complémentaire d'un ARN avec des désoxyribonucléotides en utilisant une ADN polymérase ARN dépendante) suivie d'une PCR. Elle permet de pouvoir séquencer, cloner ou mesurer un transcrit généralement très faiblement représenté.
- La méthode NASBA (Nucleic Acid Sequence-based Amplification ; Compton, 1991) qui est une méthode d'amplification isotherme reposant sur la réplication rétrovirale et utilisant trois enzymes.

Les techniques de PCR sont donc largement utilisées pour augmenter la sensibilité et permettre une meilleure identification.

2.2.2.3 Le séquençage

Le séquençage consiste à déterminer la succession de nucléotides composant le brin d'ADN étudié. Cette technique utilise les connaissances qui ont été acquises depuis une trentaine d'années sur les mécanismes de la réplication de l'ADN : elle repose sur l'allongement par l'ADN polymérase d'un brin à partir d'une amorce, en utilisant un autre brin d'ADN comme matrice. La méthode de séquençage (méthode des didésoxyribonucléotides) proposée par Frederick Sanger (Sanger *et al.*, 1977) est universellement employée pour séquencer l'ADN. Depuis 1977, la méthode a considérablement évolué grâce à la mise au point de séquenceurs automatiques et du marquage des nucléotides à l'aide de fluorochromes.

La technique d'identification basée sur le séquençage direct consiste à séquencer un segment d'ADN et à comparer cette séquence avec les séquences connues et stockées en banques de données. Cela permet d'identifier le taxon étudié et de le classer dans la collection de données. Il est ainsi possible, par exemple, de comparer la séquence inconnue avec celles des banques de gènes homologues afin de déterminer la famille de séquences homologues à laquelle appartient le taxon inconnu.

Lorsqu'une région spécifique d'un chromosome est séquencée, des banques de gènes spécifiques correspondants sont utilisées pour la comparaison. Par exemple pour l'identification bactérienne, des gènes spécifiques sont utilisés, et en particulier ceux codant pour l'ARN. Il existe différents types d'ARN (cf. chapitre 1, page 1 section 1.2.4, page 7). Les ARNr sont trés utilisés en taxonomie et en identification bactérienne. Ils ont été choisis en identification pour plusieurs raisons : c'est une molécule ubiquiste, sa structure est bien

conservée car c'est une molécule indispensable à toute cellule pour la biosynthèse des protéines et enfin les ARNr sont abondants dans la cellule et donc facilement purifiables. Les ARNr s'associent à des protéines pour former les ribosomes constitués de deux sous-unités. Chez les procaryotes, la petite sous-unité 30S est constituée d'une molécule d'ARN 16S et de 21 chaînes polypeptidiques. La grande sous-unité 50S est constituée d'une molécule d'ARN 5S, d'une molécule d'ARN 23S et de 34 chaînes polypeptidiques. Chez les eucaryotes, la petite sous-unité 40S est composée d'une molécule d'ARNr 18S et de 33 protéines. La grande sous-unité 60S est composée de trois molécules d'ARNr 5S, 28S et 5.8S et de 49 protéines. Parmi les gènes codant pour les différentes sous-unités ribosomiques, le gène codant pour l'ARNr 16S est le plus utilisé en identification bactérienne pour déterminer le genre et l'espèce car sa structure secondaire est plus conservée.

Les gènes spécifiques utilisés dans l'identification doivent être rigoureusement choisis car ils doivent permettre de différencier l'ensemble des espèces d'un genre. Ces gènes ne doivent pas non plus avoir des régions soumises à de trop fortes variations car cela impliquerait une perte d'homogénéité intra-spécifique. Enfin, les gènes peuvent subir des transferts horizontaux (échange de matériel génétique entre des organismes non forcément apparentés), ce qui peut entraîner des erreurs d'identification.

2.2.2.4 Les codes barres d'ADN ("DNA barcode")

Récemment, l'utilisation de codes barres d'ADN a été proposée (Hebert *et al.*, 2003a) afin de faciliter l'identification et la découverte d'espèces. Un projet international visant à établir le code barre de toutes les espèces animales et végétales a été mis en place. Le Consortium Barcode of Life (CBOL) réunit plus de 50 organisations de 22 pays des six continents. Les partenaires établissent des codes barres d'ADN afin de constituer un inventaire géant de la vie sur terre. C'est une technique qui utilise des courtes séquences de gènes situés à une position standard du génome comme marqueurs moléculaires pour des espèces. Chaque espèce a un code barre d'ADN différent, ce qui permet d'utiliser ces codes barres pour identifier les spécimens, découvrir, caractériser, décrire de nouvelles espèces et améliorer la taxonomie.

Hebert et al. ont établi que le gène mitochondrial de la sous-unité 1 de l'oxydase du cytochrome c peut être utilisé comme code barre pour identifier les espèces du règne animal (Hebert *et al.*, 2003a; Hebert *et al.*, 2003b). Les séquences de codes barres d'ADN sont très courtes (environ 500 paires de bases) relativement au génome entier et elles peuvent être obtenues rapidement et assez facilement. Ils ont également montré que la diversité dans les séquences d'acides aminés codées par la section 5' du gène mitochondrial était suffisante pour placer convenablement les espèces à l'intérieur des catégories taxonomiques les plus hautes (des phylums aux ordres). Comme deuxième étape dans les procédures de codes barres d'ADN, les "microcodes" ont été proposés (Summerbell *et al.*, 2005) pour permettre une identification plus rapide, moins coûteuse ou plus facile à utiliser. Ces "microcodes" correspondent à des codes barres d'oligonucléotide et font moins de 25 paires de bases.

Hebert et al. (Hebert *et al.*, 2003b) argumentent contre l'utilisation des gènes codant pour l'ARNr 16S et 12S comme marqueurs moléculaires à cause des problèmes d'alignements que posent la présence d'insertions et de délétions multiples dans ces gènes. Cependant, beaucoup d'études montrent que les gènes codant pour l'ARNr 28S (Tautz *et al.*, 2003; Markmann & Tautz, 2005), l'ITS (Blaxter, 2003) et l'ARNr 16S (Vences *et al.*, 2005; Steinke *et al.*, 2005) peuvent être utilisés comme marqueurs moléculaires dans les systèmes de codes barres d'ADN.

3 Les outils bioinformatiques existants

L'identification consiste à comparer une séquence inconnue à un ensemble de séquences connues d'une banque. Trouver la ou les séquences de la banque qui sont les plus proches de la séquence requête permet d'identifier celle-ci. Pour trouver ces séquences, il est nécessaire d'utiliser un ensemble de processus parfois complexes à manipuler. Tout d'abord il faut faire une recherche de similarité dans la banque pour sélectionner les séquences les plus proches de celle analysée. Puis les séquences retenues sont alignées avec celle traitée. Enfin l'arbre phylogénétique correspondant doit être construit. De plus, les résultats doivent parfois être vérifiés manuellement. Lorsque ces tâches doivent être faites séquentiellement, l'identification de séquences devient un travail long et pénible. Le nombre de séquences disponibles dans les banques augmente de façon exponentielle avec les techniques de séquençage massif et identifier les gènes de protéines inconnues est une étape cruciale de l'annotation de génomes. Des outils bioinformatiques automatisés sont donc nécessaires pour effectuer ces opérations de façon précise et rapide. Des applications ont été développées afin d'aider les biologistes à analyser et identifier de nouvelles séquences. Ces outils ont une approche différente selon les données qu'ils traitent et les banques qu'ils utilisent. Nous présentons, ici, quelques uns de ces outils. Beaucoup de méthodes d'identification sont utilisées pour l'identification de micro-organismes. Ainsi plusieurs méthodes ont été développées pour permettre l'identification de bactéries. L'identification est également utilisée pour tracer les aliments. Récemment, un outil utilisant les codes barres d'ADN a été développé.

3.1 BIBI, MitALib et PhyID/CD

Au sein du laboratoire de Biométrie et Biologie Evolutive, trois expériences de développement de systèmes d'identification basés sur l'utilisation des séquences biologiques ont été conduites. Tout d'abord le système BIBI (Bioinformatics Bacterial Identification, Devulder et al., 2003), dédié principalement à l'identification de bactéries pathogènes, puis le système MitALib (Mitochondrial sequences Aligned Library, Delucinge, 2003), dédié à la traçabilité en alimentation humaine et enfin PhyID/CD (Flandrois *et al.*, 2005) pour l'identification bactérienne et virale. Ces trois systèmes se basent sur l'emploi de banques de données, associées à un ensemble d'outils permettant de positionner une séquence requête par rapport aux séquences figurant dans ces banques. La méthodologie

employée par ces trois systèmes est cependant assez différente.

BIBI est un outil spécifiquement développé pour l'identification bactérienne. L'utilisateur sélectionne l'une des six banques de séquences bactériennes proposées par le serveur, puis le programme effectue une recherche de similarité conduite au moyen du logiciel BLAST. Cette première étape permet d'identifier un ensemble de séquences, dont le nombre maximum est fixé par l'utilisateur, proches de la séquence requête. L'étape suivante consiste en le filtrage et le traitement du fichier de sortie de BLAST. Les séquences considérées comme les plus similaires sont alignées avec la séquence requête au moyen d'un logiciel d'alignement multiple tel que CLUSTAL W puis MUSCLE dans la dernière version de BIBI (BIBI light edition). Enfin, à partir de cet alignement, un arbre phylogénétique est calculé au moyen de la méthode NJ (Neighbour-Joining), également implémentée dans CLUSTAL W. L'utilisateur a alors la possibilité de visualiser sur cet arbre quelle est la séquence la plus proche de la séquence requête. La banque Européenne d'ARNr utilise l'algorithme de BIBI afin de permettre à un utilisateur de faire des analyses rapides de phylogénies d'ARNr (Wuyts *et al.*, 2004).

Dans le cas de MitALib, une seule banque, contenant des séquences mitochondriales, est disponible sur le serveur. La différence avec les banques disponibles au travers de BIBI est que ces séquences sont toutes groupées en familles homologues et préalablement alignées. La première étape dans le processus d'identification est similaire à celle utilisée par BIBI puisqu'il s'agit d'une recherche de similarité utilisant BLAST. Par contre, seule la séquence présentant le meilleur score BLAST va directement servir à l'identification. En effet, une fois cette séquence identifiée, le système va alors récupérer l'alignement correspondant à la famille à laquelle elle appartient. L'étape suivante consiste en l'alignement de la séquence requête avec le profil de l'alignement préexistant. Cette étape est réalisée au moyen du programme MULTALIN. Une fois cet alignement réalisé, un arbre phylogénétique est construit avec le programme BIONJ. Encore une fois, l'identification se fait alors par visualisation directe de la proximité des séquences sur l'arbre phylogénétique.

PhyID/CD a été conçu afin de permettre la validation d'ensembles de séquences de référence utilisés pour identifier des séquences inconnues, soit lors d'études phylogénétiques et taxonomiques soit lors de développements d'outils d'identification. Cette application utilise un ensemble de séquences pré-alignées auquel elle ajoute la séquence à identifier grâce au programme d'alignement multiple MUSCLE. L'arbre phylogénétique est ensuite reconstruit en utilisant la méthode Neighbor-Joining ou une méthode de maximum de parcimonie. Il est alors possible de visualiser dans l'arbre la séquence traitée par rapport aux séquences de l'ensemble pré-aligné choisi et de l'identifier. Le programme propose également de découper la séquence analysée en plusieurs sous-séquences qui seront, par la suite, traitées une par une.

3.2 RDP II

Le projet américain *Ribosomal Database Project* propose une banque regroupant des séquences de gènes d'ARNr. Il fournit également des services d'analyse tels que le *RDP classifier* accessible via un site Web. Cet outil permet de placer de nouvelles séquences d'ARNr 16S de bactéries et d'archées dans la hiérarchie RDP afin de donner un premier placement des séquences analysées dans la taxonomie. Il utilise un algorithme de classification basé sur une approche bayésienne naïve (*naïve Bayesian rRNA classifier*). Il donne également une estimation de la confiance à apporter à chaque placement taxonomique. Outre cet outil d'identification, sont également proposés d'autres services comme le *Hierarchy Browser* permettant de naviguer rapidement dans les données, le *Sequence Match* utilisé pour rechercher les séquences similaires à une séquence requête ou le *Probe Match* qui analyse la spécificité de séquences sondes et d'amorces par rapport à la banque RDP.

3.3 RIDOM

Le serveur web RIDOM (Ribosomal Differenciation Of Medical Organisms) fournit différents services pour les besoins de l'identification médicale (Harmsen *et al.*, 1999; Harmsen *et al.*, 2002; Harmsen *et al.*, 2003). La procédure d'identification se fait à partir d'une séquence d'ADN ribosomique de la petite sous-unité du spécimen à étudier (fragment du gène ARNr 16S). Puis une recherche de similarité permet d'obtenir le nom d'espèce ou du genre de la séquence requête. Si les premiers résultats ne sont pas satisfaisants, deux autres identifications sont proposées : 1) une identification à partir de l'ITS (Intergenic transcribed spacer) correspondant à la région comprise entre le gène codant pour l'ARNr 16S et celui codant pour l'ARNr 23S ; 2) une identification à partir d'une méthode de différenciation phénotypique conventionnelle. La banque utilisée pour l'identification est une banque spécifique développée par les auteurs de RIDOM de manière à avoir des données les plus exactes possibles. Elle privilégie la qualité à l'exhaustivité et regroupe uniquement des séquences d'ARNr 16S et d'ITS. L'interface Web proposée se base principalement sur les logiciels FASTA et CLUSTAL W. Elle utilise également le paquetage PHRED/PHRAP pour estimer la probabilité d'erreur (Ewing & Green, 1998). Enfin l'utilisateur peut contrôler la qualité des résultats en visualisant les chromatogrammes des séquences.

3.4 MicroSeq

MicroSeq (Microbial identification System) est une solution commerciale complète développée par Applied Biosystems (Foster City, Californie) pour l'identification de bactéries, de levures et de champignons filamenteux. Le système MicroSeq a une approche phylogénétique basée sur les séquences de gènes codant pour l'ARNr. Il existe deux systémes MicroSeq : le système d'identification bactérien "MicroSeq 16S rDNA" et le système d'identification fongique "MicroSeq D2 LSU". Le premier système permet d'identifier des gènes bactériens. Il est constitué de deux kits différents : le kit "MicroSeq 500 16S rDNA" fournit une identification à partir d'un fragment des 500 premières paires de bases de gènes d'ADNr

16S et le kit "MicroSeq Full Gene 16S rDNA" utilise l'intégralité du gène bactérien d'ADNr 16S. Ce système a été utilisé pour identifier différentes espèces telles que *Mycobactérium* (Patel *et al.*, 2000; Cloud *et al.*, 2002; Hall *et al.*, 2003a), *Nocardia* (Cloud *et al.*, 2004) ou pour identifier des gènes de bactéries isolés (Woo *et al.*, 2003; Fontana *et al.*, 2005). Le deuxième système est utilisé pour identifier des levures et des champignons filamenteux et se base sur une région des gènes d'ARNr de la grande sous-unité (LSU). Il a par exemple été utilisé pour identifier des levures communément trouvées dans les cliniques (Hall *et al.*, 2003b). Les deux systèmes comprennent tout le nécessaire pour l'amplification, le séquençage et l'analyse des résultats (MicroSeq® Analysis Software) en permettant la recherche de similarité, l'alignement et la construction d'arbres phylogénétiques.

3.5 TaxI

Un outil, nommé TaxI, se basant sur les codes barres d'ADN a été développé (Steinke *et al.*, 2005) afin de permettre l'identification de séquences d'ADN. Cette application est basée sur des calculs de divergences de séquences entre la séquence requête (le taxon utilisé en tant que code barre) et chaque séquence de la banque de référence définie par l'utilisateur. Le programme considère toutes les paires de séquences possibles puis utilise T-Coffee pour aligner toutes ces paires. Enfin la divergence entre chaque paire de séquences est déterminée à partir des alignements obtenus. Plusieurs méthodes de calcul de distances sont proposées dont le modèle de Jukes-Cantor et celui de Kimura à deux paramètres. La plus petite distance est identifiée et le fichier de sortie présente toutes les distances d'évolution entre les paires de séquences.

4 Les motivations

Nous avons vu que l'identification de nouvelles séquences est utilisée dans de nombreux cas tels que la classification d'une séquence dans un ensemble prédéfini de séquences connues, l'aide au diagnostic médical, la traçabilité alimentaire, la mise à jour de banque de données, *etc.* De plus, le nombre de séquences inconnues augmente exponentiellement avec le développement du séquençage. Ainsi, de plus en plus de séquences ont besoin d'être identifiées et des outils automatisés sont nécessaires pour mener à bien cette tâche. Selon les données à traiter l'outil utilisé est différent. En effet, les données biologiques sont complexes. Ainsi les outils d'identification sont souvent dépendants du type de séquences et donc des banques de séquences pour lesquels ils ont été développés. Les divers outils que nous avons décrit précédemment sont spécifiques à un domaine ou à un type de données comme par exemple BIBI qui a été développé spécifiquement pour l'identification bactérienne, RIDOM pour l'identification médicale ou encore MitALib pour l'identification alimentaire, *etc.*

Différentes banques de familles de séquences telles que HOVERGEN et HOGENOM regroupent les séquences homologues de gènes protéiques en familles et fournissent un

alignement et un arbre pour chacune de ces familles. Ces banques peuvent êtres utilisées de différentes manières, et notamment dans le but de faire des analyses phylogénétiques. L'ajout d'une seule séquence à une famille donnée de ces banques peut avoir beaucoup de répercussions sur la topologie de l'arbre phylogénétique associé. Ces changements peuvent être situés près de la séquence introduite, mais ils peuvent également être situés dans des noeuds profonds. Dans un tel cas, l'information phylogénétique apportée par la famille entière doit être prise en considération. En outre, étant donné que HOVERGEN et HOGENOM contiennent de nombreuses familles avec plusieurs milliers de séquences, il faut de puissants algorithmes afin d'ajouter rapidement une séquence à un grand alignement.

Actuellement les outils d'identification de séquences disponibles comme ceux présentés précédemment sont développés pour traiter des données spécifiques telles que des séquences d'ARN ribosomique.

L'approche utilisée par MitALib présente l'avantage de ne pas avoir à recalculer l'alignement complet puisque la séquence requête est simplement alignée sur le profil de la famille. En effet, aussi bien dans le cas de BIBI que de MitALib, l'étape limitante du processus d'identification est celle correspondant au calcul de l'alignement. Par ailleurs, ces deux systèmes ne sont effectivement fonctionnels dans le cadre d'une utilisation en ligne que parce que les banques sur lesquelles ils travaillent sont de petite taille. De ce fait, ils peuvent retourner des résultats en un temps CPU considéré comme acceptable par l'utilisateur. Dans le cas des données traitées par MitALib et BIBI, il suffit de ne récupérer qu'un petit nombre de séquences, présentant les plus fortes similarités avec la séquence requête, avant de calculer l'alignement.

De même que pour MitAlib, dans le cas de PhyID/CD, l'alignement n'est pas recalculé intégralement. La séquence est simplement ajoutée à l'ensemble de séquences pré-alignées choisi. Il faut donc au préalable calculer l'alignement de toutes les séquences avec lesquelles l'identification doit être effectuée.

De la même manière que BIBI et MitALib, RIDOM et MicroSeq utilisent des banques de petite taille, garantissant ainsi une rapidité d'exécution. Ce sont des systèmes fermés : le choix des séquences intégrées dans la banque et la vitesse de mise à jour dépend de la subjectivité et de la réactivité des concepteurs. De plus MicroSeq est un système commercial.

Enfin, l'outil de classification de RDP permet d'identifier des séquences d'ARNr uniquement. Il utilise un algorithme de classification basé sur une approche bayésienne naïve qui nécessite d'entraîner l'outil de classification sur un jeu d'essai de séquences de référence d'ARNr 16S. Taxi, quant à lui, se base sur des codes barres d'ADN qui correspondent à des gènes d'ARNr 16S ou des gènes mitochondriaux de la sous-unité 1 de l'oxydase du cytochrome c.

Ainsi tous ces outils ne peuvent pas être employés efficacement avec des grandes

banques de familles de gènes homologues telles que HOVERGEN et HOGENOM. Nous avons donc développé une application, appelé HoSeqI (Homologous Sequence Identification), permettant l'identification automatique de séquences homologues et leur classification à l'intérieur de grandes banques de familles de gènes homologues.

Chapitre 3

L'outil d'identification développé : HoSeqI

Afin de répondre aux différents besoins d'identification dans les grandes banques de familles de gènes homologues telles qu'HOGENOM, nous avons développé une application, appelée HoSeqI (Homologous Sequence Identification). Elle permet de classer automatiquement de nouvelles séquences dans les familles de gènes homologues afin de les identifier. Cette application, accessible par une interface Web, intègre différents programmes de recherche de similarité, d'alignements multiples et de constructions d'arbres phylogénétiques ainsi que des outils spécifiques. L'identification se fait par une approche phylogénétique, permettant ainsi l'analyse des relations évolutives entre séquences.

1 Les objectifs

Les différents outils que nous avons présentés au chapitre précédent sont utilisés pour des gènes spécifiques et sont, pour la plupart, destinés à des identifications bactériennes. Une application d'identification dédiée aux banques de familles homologues doit utiliser une approche différente puisqu'il ne s'agit pas du même type d'identification. En effet, il faut qu'elle soit adaptée aux particularités de l'identification de nouvelles séquences dans ce type de banque.

Différents problèmes doivent être abordés lorsqu'il s'agit de travailler avec des banques de familles de gènes. Tout d'abord, il faut étudier l'accès aux données. Les bases de données ne sont pas toutes construites sur le même modèle et utilisent des systèmes de gestion différents. Ainsi l'outil d'interrogation permettant d'accéder aux données doit être étudié afin d'utiliser les fonctionnalités adaptées à nos besoins et trouver les informations qui nous intéressent le plus rapidement possible.

De plus, lors de l'ajout d'une nouvelle séquence à une famille, l'ensemble de celle-ci doit être pris en compte (cf. chapitre 2, page 31 section 4, page 43). En outre, ces bases de données peuvent contenir des familles de plusieurs milliers de séquences. Cela nécessite

donc l'utilisation d'algorithmes capables de traiter un grand nombre de séquences.

Enfin, l'application développée est destinée à être utilisée *via* Internet. Il s'agit de développer une application Web qui permettra d'obtenir des résultats rapidement. Il est donc indispensable d'utiliser des méthodes et des programmes pouvant effectuer les tâches demandées dans un intervalle de temps acceptable pour une telle application. De plus l'interface doit être facile à manipuler et doit permettre d'interagir le plus efficacement possible avec l'utilisateur.

2 L'accès aux données

2.1 Les banques utilisées

Nous avons travaillé avec des banques disponibles au Pôle Bio-Informatique Lyonnais (PBIL : http ://pbil.univ-lyon1.fr) regroupant les séquences en familles de gènes homologues.

Les banques HOGENOM, HOVERGEN, HomolEns (cf. chapitre 1, page 1 section 2.2, page 13) et HOVERGEN clean ont été développées au laboratoire de Biométrie et Biologie Evolutive de Lyon :

- HOGENOM regroupe les séquences de génomes complets.
- HOVERGEN contient les séquences de tous les gènes des vertébrés figurant dans GenBank.
- Une version "nettoyée" de HOVERGEN est proposée : HOVERGEN clean. Dans cette version, les alignements et les arbres sont plus fiables mais contiennent moins de séquences : les séquences partielles qui couvrent moins de 80% de l'alignement des protéines complètes sont exclues.
- HomolEns contient les gènes homologues des organismes séquencés entièrement disponibles sur Ensembl.

Une version de la banque HAMAP (cf. chapitre 1, page 1 section 2.1, page 13) est également proposée. Elle contient des séquences protéiques de Uniprot trouvées dans les familles HAMAP.

Nous présentons ici un exemple de méthode de construction de familles de gènes homologues. Le principe utilisé pour les banques HOVERGEN, HOGENOM et la version de HAMAP proposée au PBIL est le suivant (Perrière *et al.*, 2000) : pour des séquences protéiques complètes, deux séquences sont incluses dans la même famille si elles sont similaires à 50% ou plus sur au moins 80% de leur longueur. De plus, pour la construction des familles, les liens transitifs simples sont utilisés (si une séquence A requiert les conditions pour être dans la même famille qu'une séquence B et qu'une séquence C est placée dans la même famille que la séquence A, alors les séquences A, B, C sont incluses dans la même famille). Une fois que toutes les séquences complètes sont traitées et que la classification en familles est construite, les séquences partielles (*i.e.* des séquences qui correspondent à une portion de gène uniquement) peuvent être ajoutées. Une séquence partielle similaire à

une séquence protéique complète est incluse à la famille correspondante si elle remplit les conditions requises pour les séquences complètes et si sa longueur est supérieure ou égale à 100 acides aminés ou à la moitié de la longueur de la séquence protéique complète. Ainsi une séquence partielle peut être attribuée à plusieurs familles de gènes homologues et ne peut jamais être l'unique séquence constituant une famille. Toutes les séquences partielles qui ne sont pas incluses dans des familles sont regroupées. L'information selon laquelle la séquence n'est pas complète mais partielle est stockée dans la banque comme mot-clé associé à la séquence.

A partir des annotations des séquences protéiques, le lien vers les séquences nucléotidiques est récupéré. Ces banques sont ainsi constituées d'une part, par les séquences protéiques au format Uniprot et d'autre part, par les séquences nucléotidiques au format EMBL. Des liens croisés existent entre ces deux collections.

Dans la version actuelle de ces banques disponibles au PBIL, les alignements et les arbres phylogénétiques sont calculés et stockés pour les familles de moins de 500 séquences. Pour les autres familles (plus de 500 séquences), les alignements et les arbres ne sont pas calculés.

Nous avons décidé de travailler avec les banques protéiques car la recherche de similarité à partir de séquences protéiques est plus sensible, plus fine et plus rapide que les comparaisons faites à partir de séquences nucléotidiques (cf. chapitre 1, page 1 section 3.1, page 15).

2.2 L'outil d'interrogation

Ces banques utilisent le système de gestion de bases de données ACNUC (Gouy *et al.*, 1985), développé au Laboratoire de Biométrie et Biologie Evolutive. Ce système permet de structurer les informations d'une banque de données de séquences utilisant les formats GenBank, EMBL, SWISS-PROT et PIR. Les éléments d'information structurés sont utilisés comme critères de sélection des données pour l'interrogation des banques de séquences et donc pour retrouver des séquences. Ces éléments sont : le nom de la séquence, la longueur en nucléotide, la date d'entrée dans la banque, le numéro d'accession, les mots-clés, l'espèce de provenance ainsi que tous les niveaux de classification taxonomique des organismes, les noms d'auteurs, le journal et l'année de publication, les références bibliographiques et enfin les noms des éléments figurant dans les *features* (nature de la molécule séquencée, nature de l'organite, ...).

Le système ACNUC définit des séquences en régions fonctionnelles correspondant à des sous-séquences ayant une signification biologique particulière (parties codantes, molécules d'ARNt, d'ARNr et d'autres types d'ARN). Ainsi, il y a d'une part les séquences-mères qui sont les séquences telles qu'elles figurent dans la banque et les séquences-filles correspondant aux sous-séquences. Pour une base de données, ACNUC

définit un ensemble de douze index (ACCESS, AUTHOR, BIBLIO, EXTRACT, KEYWORDS, LOCUS, LONGL, SHORTL, SMJYT, SPECIES, SUBSEQ, TEXT) associés aux fichiers plats qui contiennent les entrées. Ces index sont des fichiers binaires à accès direct.

Associé au système ACNUC, un langage d'interrogation permet d'accéder aux informations d'une base de données à travers ses index grâce à des requêtes. Ce langage est accessible *via* le programme de commande en ligne Query mais également les interfaces Query-WIN et WWW-Query (Perrière & Gouy, 1996). Des requêtes complexes utilisant plusieurs critères de sélection liés par des opérateurs logiques sont possibles. Ce programme utilise une bibliothèque écrite en langage C ANSI. Les fonctions de cette bibliothèque permettent d'interroger des bases de données ACNUC et sont facilement utilisables dans des programmes développés par des utilisateurs afin d'effectuer des traitements plus complexes pour accéder aux données. Il existe également des API ACNUC écrites en Python et en R (Charif & Lobry, 2006).

3 Le principe et le choix des méthodes à utiliser

Le but de l'application est de permettre la classification d'une nouvelle séquence à l'intérieur d'un ensemble de séquences d'une banques de familles de gènes homologues. Cette identification exige l'enchaînement de différentes tâches et programmes parfois complexes à manipuler. Trois étapes sont nécessaires pour identifier des séquences dans les banque de familles (figure 3.1) :

- Rechercher la famille de gènes homologues à laquelle appartient la séquence inconnue.
- Aligner la séquence inconnue avec les séquences de la famille trouvée.
- Reconstruire l'arbre phylogénétique de la famille en y incluant la nouvelle séquence.

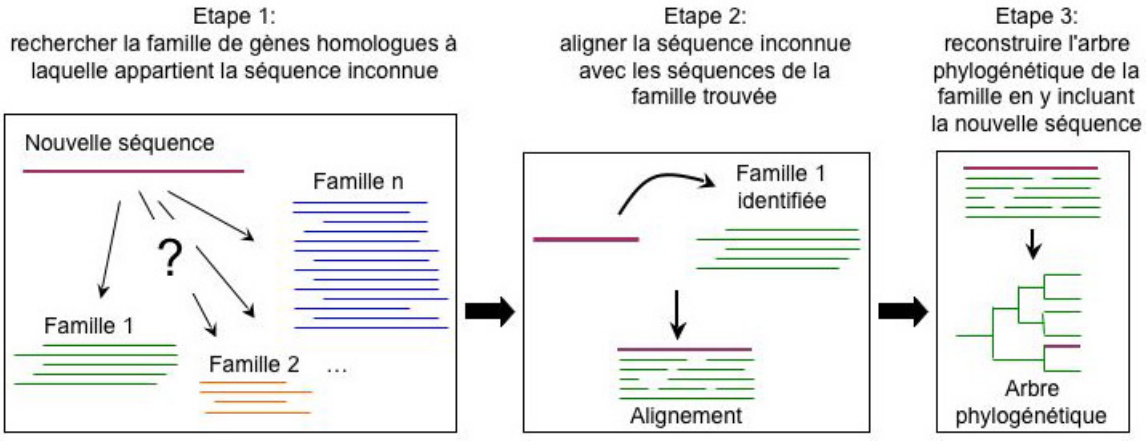

FIG. *3.1: Les trois étapes nécessaires à l'identification de nouvelles séquences dans les banques de familles de gènes homologues.*

3.1 La recherche de la famille à laquelle appartient la séquence requête

Les banques avec lesquelles nous travaillons regroupent les séquences en familles de gènes homologues. Pour classer et identifier une nouvelle séquence à l'intérieur de ces banques, la première étape consiste à trouver à quelle famille de la banque la séquence requête appartient. Il s'agit de trouver la famille dont les séquences sont les plus similaires à la séquence requête. Pour cela, il faut donc commencer par faire une recherche de similarité et comparer la nouvelle séquence avec les séquences de la banque pour ensuite rechercher la famille dans laquelle la séquence requête peut être classée.

3.1.1 La recherche de similarité

Plusieurs logiciels permettent de faire des recherches de similarité, et les plus connus sont BLAST et FASTA (cf. chapitre 1, page 1 section 3.1, page 15). Chacun des deux logiciels a ses avantages et ses inconvénients. BLAST est plus rapide que FASTA pour une recherche classique effectuée sur la totalité des banques de données protéiques ou nucléique. Il favorise la vitesse par rapport à la sensibilité. Pour la recherche sur des chaînes nucléotidiques, FASTA est la référence bien qu'il effectue les recherches en quelques heures contre deux à trois minutes pour BLAST. Etant donné que nous travaillons avec des banques protéiques et que nous avons besoin d'une certaine rapidité d'exécution pour l'application Web, nous avons décidé d'utiliser BLAST (version BLAST2) qui permet une recherche rapide, faisant apparaître les alignements possibles.

BLAST possède en fait cinq programmes distincts de comparaison avec les banques de données : BLASTN qui permet de comparer une séquence nucléotidique avec les séquences d'une banque nucléique, BLASTP qui permet de comparer une séquence protéique avec les séquences d'une banque protéique, BLASTX qui permet de comparer une séquence nucléotidique en la traduisant dans les six phases avec les séquences d'une banque protéique, TBLASTN qui permet de comparer une séquence protéique avec les séquences qu'il traduit dans les six phases d'une banque nucléique et TBLASTX qui permet de comparer une séquence nucléotidique en la traduisant dans les six phases avec les séquences qu'il traduit dans les six phases d'une banque nucléique. L'outil que nous développons doit permettre d'identifier une séquence nucléotidique ou protéique soumise par un utilisateur par rapport aux séquences d'une banque protéique. Nous avons choisi d'utiliser BLASTP aussi bien avec les séquences protéiques que les séquences nucléotidiques. Nous ajoutons pour ces dernières une étape de traduction en utilisant le logiciel TRANSEQ (package EMBOSS) (Rice *et al.*, 2000). Par défaut, la traduction se fait dans les phases de lecture mais l'utilisateur peut indiquer la phase qu'il souhaite utiliser. Nous n'utilisons pas BLASTX car celui-ci traduit automatiquement la séquence nucléotidique dans les six phases de lecture et ne permet pas à un utilisateur de choisir la phase de traduction. De plus, les séquences protéiques traduites ne sont pas disponibles en entier à la fin du traitement de BLASTX.

L'unité fondamentale de BLAST est le HSP (High-scoring Segment Pair). Il correspond à une région similaire, la plus longue possible, entre deux séquences ayant un score supérieur ou égal à un score seuil. Les HSPs sont donc des alignements locaux dont le score dépasse un score fixé. Le HSP ayant le meilleur score parmi tous les HSPs sur l'ensemble de deux séquences est appelé MSP (Maximal-scoring Segment Pair). Plusieurs HSPs peuvent être trouvés sur la même séquence, c'est-à-dire qu'il peut y avoir plusieurs alignements locaux entre deux séquences avec un score significatif (figure 3.2).

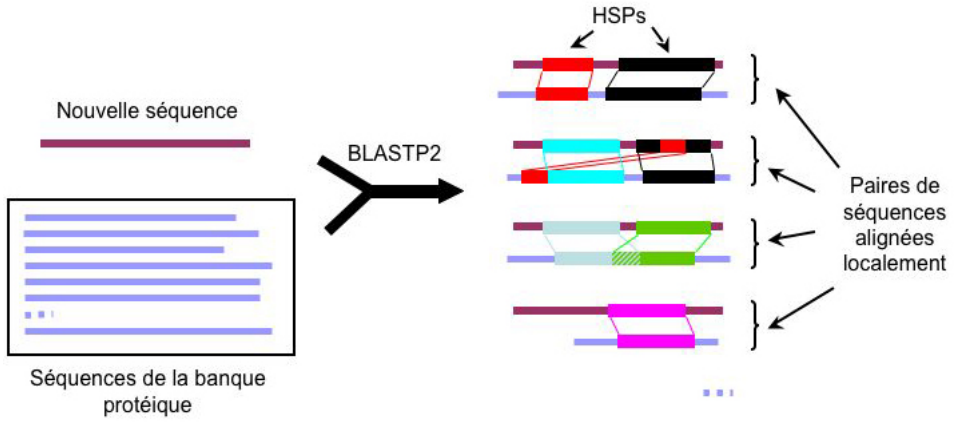

FIG. *3.2: Recherche de similarité entre une nouvelle séquence et les séquences d'une banque avec BLASTP2. Représentation des paires de séquences alignées localement et de leurs HSPs. Une paire est constituée de la séquence requête et d'une des séquences de la banque.*

En sortie du programme, BLAST fournit un fichier avec les résultats de la comparaison effectuée. En début de fichier se trouve un résumé avec le score et la *E-value* (cf. chapitre 1, page 1 section 3.1, page 15) globaux obtenus pour la comparaison de chaque séquence de la banque avec la séquence requête, par ordre décroissant du score global, et ordre croissant de la *E-value* globale. Dans le cas où il y a plusieurs HSPs, le score global correspond au maximum des différents scores calculés et la *E-value* globale est déterminée par une méthode de "somme statistique" (Karlin & Altschul, 1993). Puis chacun des alignements locaux (HSPs) est présenté sous forme d'alignement par paire avec différentes informations telles que le score, la *E-value*, une valeur *identities* (proportion des paires de résidus identiques entre deux séquences alignées exprimée en pourcentage), une valeur *positives* (proportion des paires de résidus avec des poids positifs entre deux séquences alignées exprimée en pourcentage) et une valeur *Gap* (proportion de brèches c'est-à-dire d'insertions et de délétions entre deux séquences alignées exprimée en pourcentage). Il faut être prudent sur l'interprétation des résultats obtenus par BLAST. Il faut savoir que la signification statistique ne reflète pas forcément la signification biologique et inversement. Plus la *E-value* est faible, plus le score de l'alignement est significatif. Dans la sortie de BLAST, les séquences présentées en premier sont donc celles qui sont les plus similaires à

la séquence requête (plus haut score et plus basse *E-value*) et qui sont susceptibles d'être homologues à la séquence requête. Afin de garder un maximum d'informations apportées par les résultats de BLAST, il a fallu tenir compte de cette possibilité d'avoir plusieurs HSPs lors de l'implémentation de l'outil d'identification.

Pour toute comparaison de séquences, une matrice de similarité est indispensable au calcul du score (cf. chapitre 1, page 1 section 3.1.2, page 17). Il n'y a pas de matrice idéale et le choix de la matrice dépend du type d'analyse à faire. Pour des séquences relativement similaires et courtes, il est préférable d'utiliser une matrice BLOSUM élevée ou PAM faible. A l'inverse une matrice BLOSUM faible ou PAM élevée permet de comparer des séquences plus divergentes et plus longues. Généralement les matrices plutôt basées sur les comparaisons de séquences (BLOSUM, Gonnet) sont meilleures pour détecter les alignements locaux et donnent de meilleurs résultats que celles basées principalement sur le modèle de Dayhoff (PAM). BLAST utilise par défaut la matrice BLOSUM62. Cette matrice donne les meilleurs résultats pour détecter la majorité des faibles similarités entre séquences protéiques.

Une recherche de similarité en utilisant BLASTP2 entre la séquence requête et les séquences des banques de familles de gènes permet donc d'obtenir un ensemble de séquences proches de la séquence requête, classées par ordre décroissant de similarité avec pour chaque comparaison le ou les alignements locaux obtenus et les informations associées.

3.1.2 L'identification de la famille

Une fois que la recherche de similarité est effectuée, il est nécessaire d'analyser les résultats obtenus afin d'identifier les séquences les plus proches de celle soumise. Pour cela, nous considérons chaque paire de séquences composée de la séquence requête et d'une séquence sujet, *i.e.* une séquence de la banque alignée localement avec celle analysée. Chaque comparaison de paires est étudiée afin de déterminer si pour une même paire il existe plusieurs alignements locaux (HSPs). Dans le cas où plusieurs HSPs sont détectés entre deux séquences, il faut évaluer s'ils doivent être tous pris en compte ou pas. Dans le cas où il existe un chevauchement significatif (plus de quelques acides aminés) entre des HSPs, le HSP ayant le plus haut score (le MSP) est retenu uniquement. Par contre, lorsque les HSPs ne se chevauchent pas, ils doivent être tous pris en compte (figure 3.3).

Les HSPs retenus sont regroupés selon la séquence de la banque à laquelle ils correspondent. Un score cumulé et une *E-value* globale sont calculés. On ne retient que les séquences sujets pour lesquelles la *E-value* globale obtenue est inférieure à un seuil. Ce seuil a été fixé à la valeur suivante : il s'agit de la *E-value* du meilleur alignement obtenu par BLAST x 10^5. Ce seuil permet de ne pas prendre en compte uniquement la meilleure séquence sujet (*i.e.* celle pour laquelle le score cumulé et la *E-value* globale sont les meilleurs), mais un ensemble suffisamment conséquent pour permettre de déterminer

la famille à laquelle appartient la séquence requête.

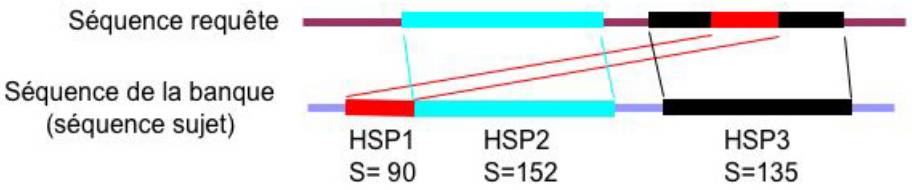

FIG. *3.3: Sélection des HSPs pour extraire les séquences nous permettant de déterminer la famille de gènes homologues à laquelle appartient la nouvelle séquence. Ici le HSP1 et le HSP3 se chevauchent. Un seul de ces deux HSPs est retenu. Le HSP sélectionné est celui de plus haut score, c'est-à-dire le HSP3 qui a un score de 135 (contre un score de 90 pour le HSP1). Le HSP2 est également sélectionné car il ne chevauche aucun autre HSP.*

Les séquences partielles ne sont pas prises en compte pour la détermination de la famille. En effet, ces séquences peuvent être attribuées à aucune, une ou plusieurs familles. Une courte séquence partielle peut présenter une forte similarité avec la séquence requête alors que les séquences complètes de la famille correspondante obtiendraient un taux de similarité moins fort. Tout cela peut engendrer des problèmes lors de l'identification de la famille de la séquence requête. De plus lorsqu'une séquence partielle appartient à une famille, il existe forcément une séquence protéique complète dans la même famille. Nous n'avons donc pas considéré les séquences partielles dans notre analyse puisque les séquences complètes suffisent pour identifier correctement la famille de la nouvelle séquence.

La famille de la banque à laquelle appartient chacune des séquences sélectionnées précédemment est déterminée grâce à l'utilisation de fonctions de la bibliothèque ACNUC. Par une méthode que nous développerons plus tard (section 4.3.1, page 65), il est alors possible de déterminer la famille dont les séquences sont les plus similaires à la nouvelle séquence. Celle-ci peut ainsi être attribuée à la famille de gènes homologues trouvée afin de permettre son identification. Dans certains cas (expliqués à la section 4.3.1, page 65) plusieurs familles peuvent être proposées.

3.2 L'alignement de la séquence requête avec les séquences de la famille

Lorsque la famille de gènes homologues de la séquence requête a été identifiée, il faut classer cette séquence par rapport aux autres séquences de la famille. Pour cela la nouvelle séquence doit être alignée avec les séquences de la famille.

Une première étape est celle de l'extraction des données. Cela consiste à récupérer les séquences de la famille et à les convertir au bon format afin de pouvoir ensuite utiliser un programme pour les aligner avec la nouvelle séquence. Il existe deux cas :

– soit la famille de gènes homologues est constituée de moins de 500 séquences et l'alignement est pré-calculé et stocké dans la banque,
– soit la famille de gènes homologues regroupe plus de 500 séquences et l'alignement n'est pas pré-calculé.

Pour les familles de moins de 500 séquences, il suffit de récupérer le fichier de l'alignement des séquences de la famille dans la banque et de le convertir au format FASTA.

Lorsque l'alignement n'existe pas, toutes les séquences de la famille sont extraites directement à partir de la banque et stockées dans un fichier au format FASTA en utilisant des fonctions et un programme de conversion de la bibliothèque ACNUC.

Dés lors que les séquences ou l'alignement de la famille sont obtenus, la séquence requête peut être alignée avec les séquences de la famille. Pour cela différents programmes d'alignements multiples sont disponibles (cf. chapitre 1, page 1 section 3.2.1, page 22). Ils doivent permettre d'aligner un grand nombre de séquences rapidement et d'ajouter une séquence à un alignement pré-existant. Parmi les programmes présentés au premier chapitre, notre choix s'est porté sur les programmes ayant une approche basée sur l'alignement progressif car ils offrent une rapidité de traitement et les alignements obtenus sont de bonne qualité lorsque les séquences sont similaires. Ainsi nous avons choisi de proposer l'ensemble des programmes suivant : CLUSTAL W, MULTALIN, MUSCLE (MUSCLE-prog, MUSCLE-fast) et MENTALIGN. Même si ces programmes sont basés sur la même approche, ils n'offrent pas tout à fait les mêmes possibilités et ne sont pas performants de la même manière en fonction du nombre de séquences traitées. Nous avons ajouté à cette liste de programmes le logiciel MABIOS qui utilise une approche d'alignement par bloc particulièrement efficace lorsque les séquences ne sont pas globalement similaires mais qu'elles partagent des ressemblances locales.

Chacun des programmes fonctionne avec des paramètres différents : format du fichier en entrée, présence ou non de la séquence à ajouter dans ce fichier, utilisation d'un fichier d'arbre phylogénétique de la famille, format du fichier de sortie, *etc.* A partir des fichiers au format FASTA obtenus après la phase d'extraction des données, il est donc nécessaire de faire des pré- ou post-traitement spécifiques à chaque programme tels que des conversions afin d'avoir les bons formats de données.

Le but de l'outil développé étant de permettre à un utilisateur d'identifier une séquence rapidement dans les banques de familles de gènes *via* une interface Web, nous avons évalué les performances en termes de temps d'exécution de ces différents programmes en utilisant des données provenant de banques de familles homologues. Ainsi nous avons réalisé des tests permettant de connaître la rapidité d'exécution des programmes en fonction du nombre de séquences traitées. Ces tests ont été réalisés avec un ordinateur MACINTOSH (Power Mac G5, processeur PowerPC 970 1.6 GHz, 768 Mo de RAM, Mac OS X). Nous avons utilisé les familles de séquences de l'ancienne banque HOBACGEN (Homologous Bacterial Genes Database, Perrière et al., 2000) développée au PBIL et de HOVERGEN.

Nous avons choisi d'utiliser HOBACGEN afin d'avoir un banc d'essai stable pour réaliser les tests. Cette banque regroupe en familles homologues toutes les séquences de bactéries (bactéries et archées) et de levures disponibles dans SWISS-PROT et TrEMBL. Le jeu de données utilisé contient des familles de gènes homologues ayant un nombre de séquences allant de 50 à 18000. Tous les programmes ont été testés avec chacune des familles afin d'évaluer leur temps d'exécution. Le temps d'ajout d'une séquence à l'alignement obtenu pour chaque famille a également été évalué pour tous les programmes. Les résultats sont résumés dans le tableau suivant (figure 3.4).

	MABIOS	CLUSTALW	MULTALIN	MENTALIGN	MUSCLE	MUSCLE-prog	MUSCLE-fast
HBG000049 (Hobacgen) - 49 séquences (134 à 240 aa)	18s	10s	2s	112s	4s	2s	
ajout de 1 séquence	1 s	1 s		6s			
HBG000089 (Hobacgen) - 139 séquences (183 à 520 aa)	16m 27s	7m 37s	33s	13m 14s	2m 54s	11s	3s
ajout de 1 séquence	12 s	2 s		29 s			
HBG000097 (Hobacgen) - 263 séquences (100 à 332 aa)	17m 20s	10m 40s	23s	42m 14s	2m 24s	9s	
ajout de 1 séquence	35s	11s	1s	14s			
HBG012447 (Hobacgen) - 404 séquences (122 à 293 aa)	982s	13m 8s	88s	31m 3s	2m 10s	11s	
ajout de 1 séquence	25s	97s		7s			
HBG000383 (Hobacgen) - 667 séquences (101 à 842 aa)	4h 28m30s	2h 9m 50s	10m	7h 31m 19s			
ajout de 1 séquence	5m 45s	12m 47s		42s			
HBG016829 (Hobacgen) - 1239 séquences (62 à 373 aa)	4h 27m 25s		16m 40s	2h 45m 5s	4h 43m 42s	1m 29s	26s
ajout de 1 séquence	12m 40s		1s	5s	24s	24s	23s
HBG021718 (Hobacgen) - 2836 séquences (75 à 636 aa)			2h 16m 34s	18h 38m 12s		13m 43s	2m 30s
ajout de 1 séquence				19s	5m 35s	4m 41s	4m 39s
HBG017625 (Hovergen) - 5490 séquences (91 à 639 aa)			7h 31m 12s	1j 21h 48m 3s		27m 58s	9m 06s
ajout de 1 séquence				24 sec	10m 39s	8m 35s	8m 29s
HBG016692 (Hovergen) - 7233 séquences (88 à 484 aa)			8h 24 m 36s	2j 8h 30m		47m 25s	11m 13s
ajout de 1 séquence			3-4s	34s		12m 18s	12m 1s
HBG017694 (Hovergen) - 18861 séquences (100 à 529 aa)							1h 38m 8s
ajout de 1 séquence			1m				

FIG. *3.4: Tableau de tests de performance des programmes d'alignements multiples CLUSTAL W, MABIOS, MULTALIN, MUSCLE, MUSCLE-prog, MUSCLE-fast et MENTALIGN. Ces programmes sont présentés en colonnes et les familles testées sont présentées en lignes. Pour chaque famille et chaque logiciel est indiqué le temps global de calcul de l'alignement et de l'ajout d'une séquence. Lorsque la case est barrée, cela signifie que le programme n'a pas pu calculer l'alignement (la procédure ne s'est pas terminée normalement). Lorsque la case est vide ou grisée, cela signifie que le test ne s'est pas fait parce que l'information obtenue n'était pas utile : soit le programme aurait donné un temps d'exécution trop long pour une utilisation sur Internet (case vide), soit les autres résultats nous suffisaient (case grisée).*

Les résultats des tests présentés dans le tableau de la figure 3.4 nous permettent de comparer les différents programmes d'alignements au niveau du temps de traitement en fonction du nombre de séquences à aligner. Il apparaît que MUSCLE-prog et MUSCLE-fast sont les plus rapides et permettent d'aligner de grands jeux de données en peu de temps. En outre, CLUSTAL W et MABIOS sont les plus lents et ne peuvent pas traiter de grands ensembles de séquences. Par ailleurs, MUSCLE-fast permet d'aligner un très grand nombre

de séquences (plus de 18000 séquences). Enfin, MENTALIGN et MULTALIN permettent d'aligner de grands jeux de données mais ils sont relativement longs. Par contre, ils sont très rapides pour ajouter une séquence à un alignement existant. Nous avons également remarqué que des problèmes surviennent lorsque certains programmes tels que CLUSTAL W et MABIOS sont utilisés pour calculer un alignement de plus de 500 séquences (exécution trop lente, mémoire insuffisante, ...). Par exemple, si la famille à traiter contient plus de 3000 séquences, l'alignement des séquences de la famille et de la séquence requête doit être recalculé en entier. Pour un tel traitement, le programme MULTALIN n'est pas proposé car cela prendrait plus de deux heures. Ce qui n'est pas envisageable pour un service proposé *via* Internet. Ces divers résultats vont nous permettre, par la suite, d'établir des filtres pour proposer une liste de programmes appropriés aux données traitées.

D'autres programmes d'alignements multiples ont également été testés avec le même jeu de données mais n'ont pas encore été intégrés à la première version de l'outil. Il s'agit de POA, PROBCONS, différentes versions de MAFFT et DIALIGN. Les tests ont pour l'instant été effectués pour évaluer le temps d'exécution de chacun des programmes pour aligner les séquences de chacune des familles du jeu de données. Le tableau de la figure 3.5 présente les résultats obtenus.

	POA	PROBCONS	DIALIGN	MAFFT-i	MAFFT-2	MAFFT-1
HBG000049 (Hobacgen) - 49 séquences (134 à 240 aa)	7s	69s	41s	3s	1s	1s
HBG000089 (Hobacgen) - 139 séquences (183 à 520 aa)	5m 23s	1h 22m 23s	31m	1m 33	4s	2s
HBG000097 (Hobacgen) - 263 séquences (100 à 332 aa)	3m 2s	45m 34s	1h 50m 23s	46s	5s	4s
HBG012447 (Hobacgen) - 404 séquences (122 à 293 aa)	2m 22s	1h 35m 49s		2m 26s	9s	6s
HBG000383 (Hobacgen) - 667 séquences (101 à 842 aa)	45m 30s			18m 42s	39s	24s
HBG016829 (Hobacgen) - 1239 séquences (62 à 373 aa)	1h 20m24s			4h 11m 58s	2m 23s	1m 17s
HBG021718 (Hobacgen) - 2836 séquences (75 à 636 aa)	11h 50m 24s			4 jours	23m 51s	9m 56s
HBG017625 (Hovergen) - 5490 séquences (91 à 639 aa)	23h 24m				40m 2s	18m 49s
HBG016692 (Hovergen) - 7233 séquences (88 à 484 aa)	9h 52m 12s				42m 24s	20m 20s
HBG017694 (Hovergen) - 18861 séquences (100 à 529 aa)						

FIG. *3.5: Tableau de tests de performance des programmes d'alignements multiples POA, PROBCONS, différentes versions de MAFFT et DIALIGN. Ces programmes sont présentés en colonnes et les familles testées sont présentées en lignes. Pour chaque famille et chaque logiciel est indiqué le temps global de calcul de l'alignement. Lorsque la case est barrée, cela signifie que le programme n'a pas pu calculer l'alignement (la procédure ne s'est pas terminée normalement). Lorsque la case est vide, cela signifie que le test ne s'est pas fait car l'information obtenue n'était pas utile : le programme aurait donné un temps d'exécution trop long pour une utilisation sur Internet.*

De la même manière que pour les programmes intégrés à HoSeqI, le tableau des résultats des tests de la figure 3.5 nous permet de comparer les différents programmes présentés

précédemment au niveau du temps d'exécution selon le nombre de séquences traitées. Nous remarquons donc que MAFFT-1 et MAFFT-2 sont les plus rapides et leur temps d'exécution est relativement comparable à celui de MUSCLE-prog. Par ailleurs, PROBCONS et DIALIGN sont les plus lents, PROBCONS ne permettant pas d'aligner un très grand nombre de séquences (problème de mémoire à l'exécution pour 667 séquences). Enfin, MAFFT-i et POA ont des temps de traitements équivalents à MUSCLE. Ces résultats devront être pris en compte lorsque ces programmes d'alignements seront intégrés à HoSeqI.

3.3 La reconstruction phylogénétique

Après avoir calculé l'alignement des séquences de la famille sélectionnée et de la nouvelle séquence, il faut ensuite reconstruire l'arbre phylogénétique. Ainsi il sera possible d'analyser la position phylogénétique de la séquence soumise, de l'identifier et d'étudier son histoire évolutive.

Parmi les différentes méthodes permettant la reconstruction d'arbres phylogénétiques, nous avons opté pour les méthodes de distances (cf. chapitre 1, page 1 section 3.3.1, page 26). Elles ont l'avantage d'être beaucoup plus rapides que les autres méthodes même si la qualité des arbres obtenus n'est pas optimale. Ainsi nous avons choisi la liste suivante des programmes implémentant une méthode de distance : BIONJ, FASTME et QUICKTREE. Nous avons ajouté à cette liste le programme PHYML, basé sur une méthode de maximum de vraisemblance, cependant cette méthode est beaucoup plus lente. Une stratégie possible est de construire un arbre phylogénétique rapidement avec les méthodes de distance, puis PHYML peut être utilisé afin d'obtenir un arbre de meilleure qualité.

Afin de pouvoir évaluer la qualité de l'arbre obtenu, une procédure de bootstrap (cf. chapitre 1, page 1 section 1.14, page 29) a été implémentée.

Enfin, l'arbre obtenu par ces différentes méthodes n'est pas raciné (cf. chapitre 1, page 1 section 3.3.2, page 28). Nous avons choisi de le raciner "en son centre" par la méthode du *midpoint*. En effet, la méthode de l'*outgroup* n'est pas forcément adaptée au développement d'outils automatiques car il faut trouver un groupe externe qui pourrait être utilisé pour n'importe quelle famille de n'importe quelle banque proposée. Il existe une méthode de racinement automatique qui se base sur des arbres de référence (Dufayard *et al.*, 2005) pour trouver l'*outgroup*. Le principe est d'utiliser un algorithme de réconciliation qui compare l'arbre des gènes avec l'arbre de référence des espèces afin d'obtenir un arbre présentant les événements de duplication et de spéciation entre les gènes. Cet algorithme sert à explorer toutes les positions de la racine dans l'arbre des gènes. La position qui minimise le nombre de duplications géniques est retenue. Cependant, des arbres de référence ne sont pas toujours disponibles, notamment pour les procaryotes, il n'y a pas d'arbre consensuel de référence. Ce ne sont donc pas des méthodes appropriées à notre outil. Le programme permettant de raciner l'arbre est ADD_ROOT, une implémentation de la méthode du *midpoint*. Cet algorithme a été mis en oeuvre par Manolo Gouy dans le cadre du développement du logiciel NJplot permettant

la visualisation d'arbres phylogénétiques (Perrière & Gouy, 1996).

Les programmes de phylogénies fonctionnent avec des paramètres différents : divers types de fichiers en entrée avec des formats spécifiques, possibilité d'option de bootstrap, *etc.* Plusieurs cas sont possibles :

- PHYML se base sur un alignement de séquence pour construire l'arbre et propose l'option de bootstrap.
- FASTME et BIONJ se basent sur une matrice de distance et n'a pas d'option de bootstrap.
- QUICKTREE se basent soit sur un alignement, soit sur une matrice de distance et permet d'utiliser une option de bootstrap.

Dans le cas d'un programme se basant sur un alignement en entrée et proposant l'option de bootstrap, il suffit juste de convertir, dans le bon format, l'alignement de séquences grâce aux logiciels de conversion CLUSTAL W ou SREFORMAT (package HMMER) (Eddy, 1998) en fonction du format souhaité. Le programme de phylogénies peut alors reconstruire l'arbre à partir de l'alignement obtenu avec le nombre de réplicats de bootstrap souhaité.

Dans le cas d'un programme se basant sur une matrice de distance, nous utilisons le logiciel PROTDIST (package PHYLIP) qui permet de calculer la matrice de distance à partir de l'alignement des séquences protéiques, converti au bon format. Le modèle évolutif choisi est celui qui correspond à la distance de Kimura (Kimura, 1983) et qui permet de calculer la matrice rapidement avec PROTDIST (les autres paramètres choisis étant ceux proposés par défaut). Si le programme de phylogénie propose une option de bootstrap ou si aucune procédure de bootstrap n'est souhaitée, il est alors possible de reconstruire l'arbre phylogénétique à partir de la matrice de distance obtenue avec le nombre de réplicats souhaité. S'il n'y a pas d'option de bootstrap proposé, le logiciel SEQBOOT (package PHYLIP) doit être utilisé afin de générer un jeu de données en fonction du nombre de réplicats (les autres paramètres étant ceux par défaut). Puis PROTDIST est utilisé pour calculer les matrices correspondantes au jeu de données généré précédemment (avec les mêmes paramètres que précédemment). Le programme de phylogénies est ensuite utilisé pour reconstruire les arbres correspondant à toutes les matrices obtenues. Enfin le logiciel ADDBOOTSTRAP (développé par Manolo Gouy) permet de calculer les valeurs de bootstrap une fois que tous les arbres sont reconstruits.

Comme nous l'avons vu avec les programmes d'alignements multiples, il est utile de connaître la performance des programmes choisis sur des données identiques à celle utilisées par l'outil afin de proposer un ensemble de méthodes les plus appropriées aux données traitées. Ainsi afin d'évaluer la rapidité d'exécution des différents programmes de reconstructions plylogénétiques, nous avons testé chacun d'eux. De même que pour les programmes d'alignements, ces tests ont été réalisés avec un ordinateur MACINTOSH (Power Mac G5, processeur PowerPC 970 1.6 GHz, 768 Mo de RAM, Mac OS X). Nous

avons utilisé un jeu de données de familles de gènes homologues provenant d'HOGENOM et HOBACGEN contenant de 5 à 3000 séquences. Les quatre programmes ont été testés sur l'ensemble des familles avec ou sans l'option de bootstrap et en prenant un nombre de réplicats de bootstrap allant de 50 à 2000. Nous avons ainsi pu estimer les performances de chaque programme en fonction du nombre de séquences à traiter et du nombre de réplicats de bootstrap demandé. Le temps d'exécution évalué correspond, en fait, au temps nécessaire pour obtenir un arbre phylogénétique raciné à partir d'un alignement en utilisant les différents programmes. Dans les tableaux de la figure 3.6 et la figure 3.7 sont présentés les résultats au niveau temps d'exécution de chacun des programmes.

	quickTree						
	sans bootstrap	avec bootstrap: 50	avec bootstrap: 100	avec bootstrap: 500	avec bootstrap: 1000	avec bootstrap: 1500	avec bootstrap: 2000
HBG000008 (Hogenom) 14 séquences	0s	0s	1s	2s	3s	3s	3s
HBG000023 (Hogenom) 38 séquences	0s	1s	1s	5s	7s	4s	5s
HBG000049 (Hobacgen) 50-1 séquences	0s	1s	2s	7s	10s	20s	30s
HBG000002 (Hogenom) 95 séquences	1s	2s	3s	20s	24s	38s	56s
HBG000089 (Hobacgen) 140-1 séquences	5s	7s	14s	75s	5m 19s	5m 38s	6m 52s
HBG000097 (Hobacgen) 263 séquences	3s	17s	39s	6m 24s	5m 32		11m 15s
HBG012447 (Hogenom) 404 séquences	6s	47s	5m 41s	8m 54s			32m 38s
HBG000223 (Hogenom) 667 séquences	17s	6m 38s	8m 38s				3h 2m 2s
HBG016829 (Hobacgen) 1239 séquences	40s						
HBG015393 (Hogenom) 2009 séquences	2m 57s	56m 56s					
HBG021718 (Hobacgen) 2837 séquences	8m 16s						

FIG. *3.6: Tableaux de tests de performance en termes de temps d'exécution du programme de reconstructions phylogénétiques QUICKTREE. Dans chaque tableau, les familles testées sont présentées en lignes et les différentes utilisations de l'option de bootstrap en fonction du nombre de réplicats sont présentées en colonnes. Pour chaque famille et chaque utilisation du bootstrap, le temps global de reconstruction de l'arbre est indiqué. Lorsque la case est vide, cela signifie que le test ne s'est pas fait car l'information obtenue n'était pas utile : le programme aurait donné un temps d'exécution trop long pour une utilisation sur Internet.*

Les résultats des tests présentés dans les tableaux de la figure 3.6 et la figure 3.7 sont analysés de la même manière que ceux correspondant aux programmes d'alignements. Ici, ils nous permettent de comparer les programmes de reconstructions phylogénétiques en termes de temps de traitements, selon le nombre de réplicats de bootstrap et le nombre de séquences. Ainsi QUICKTREE, FASTME et BIONJ sont relativement équivalents au niveau du temps d'exécution, même si QUICKTREE paraît être plus rapide pour traiter des jeux de données plus importants (par exemple, 2000 séquences). Par ailleurs, avec l'option de bootstrap, PHYML ne permet pas de reconstruire un arbre phylogénétique en un temps relativement court, quelque soit le nombre de séquences. De même que pour les programmes d'alignements, nous allons constituer des filtres à partir de ces résultats afin de proposer une liste de programmes appropriés aux données traitées.

	FastMe						
	sans bootstrap	avec bootstrap: 50	avec bootstrap: 100	avec bootstrap: 500	avec bootstrap: 1000	avec bootstrap: 1500	avec bootstrap: 2000
HBG000008 (Hogenom) 14 séquences	0s	2s	1s	6s	37s	18s	21s
HBG000023 (Hogenom) 38 séquences	0s	1s	2s	10s	76s	31s	37s
HBG000049 (Hobacgen) 50-1 séquences	0s	3s	7s	36s	3m 36s	94s	1m 56s
HBG000002 (Hogenom) 95 séquences	1s	14s	36s	1m 57s	4m 43s	6m 36s	7m 2s
HBG000089 (Hobacgen) 140-1 séquences	3s	35s	1m 27s	5m 51s	10m 8s	19m 27s	20m 21s
HBG000097 (Hobacgen) 263 séquences	6s	2m 08s	4m 56s	18m 3s		55m 24s	
HBG012447 (Hogenom) 404 séquences	10s	4m 19s	7m 52s	35m 34s		plus de 1h	
HBG000223 (Hogenom) 667 séquences	25s	plus de 30m	plus de 30m	plus de 1h			
HBG016829 (Hobacgen) 1239 séquences	3m 5s						
HBG015393 (Hogenom) 2009 séquences	6m 38s						
HBG021718 (Hobacgen) 2837 séquences	14m 39s						

	BIONJ						
	sans bootstrap	avec bootstrap: 50	avec bootstrap: 100	avec bootstrap: 500	avec bootstrap: 1000	avec bootstrap: 1500	avec bootstrap: 2000
HBG000008 (Hogenom) 14 séquences	0s	1s	2s	6s	97s	22s	22s
HBG000023 (Hogenom) 38 séquences	0s	1s	3s	11s	19s	31s	37s
HBG000049 (Hobacgen) 50-1 séquences	1s	5s	8s	43s	75s	116s	2m 31s
HBG000002 (Hogenom) 95 séquences	1s	18s	34s	3m 37s	5m 30s	9m 46s	11m 1s
HBG000089 (Hobacgen) 140-1 séquences	3s	69s	2m 6s	11m 23s		31m 25s	
HBG000097 (Hobacgen) 263 séquences	7s	195s	6m 20s				
HBG012447 (Hogenom) 404 séquences	15s	plus de 30m	19m 34s	plus de 1h		plus de 1h	
HBG000223 (Hogenom) 667 séquences	1m 15s						
HBG016829 (Hobacgen) 1239 séquences	9m 52s						
HBG015393 (Hogenom) 2009 séquences							
HBG021718 (Hobacgen) 2837 séquences							

	PhyML						
	sans bootstrap	avec bootstrap: 50	avec bootstrap: 100	avec bootstrap: 500	avec bootstrap: 1000	avec bootstrap: 1500	avec bootstrap: 2000
HBG000020 (Hogenom) 5 séquences		23m 5s	plus de 30m				
HBG000008 (Hogenom) 14 séquences	1m 46s	plus de 30m					
HBG000023 (Hogenom) 38 séquences	6m 12s						
HBG000049 (Hobacgen) 49 séquences	plus de 20m						
HBG000002 (Hogenom) 95 séquences	plus de 1h						
HBG000089 (Hobacgen) 139 séquences	plus de 1h						
HBG000097 (Hobacgen) 263 séquences							
HBG012447 (Hogenom) 404 séquences							
HBG000223 (Hogenom) 667 séquences							
HBG016829 (Hobacgen) 1239 séquences							
HBG015393 (Hogenom) 2009 séquences							
HBG021718 (Hobacgen) 2837 séquences							

FIG. *3.7: Tableaux de tests de performance en termes de temps d'exécution des programmes de reconstructions phylogénétiques FASTME, BIONJ et PHYML réalisés dans les mêmes conditions que les tests précédents.*

4 L'implémentation

4.1 L'architecture

HoSeqI est une application Web implémentée sur le serveur PBIL du Pôle Bio-Informatique Lyonnais. Elle permet l'enchaînement de processus intégrant différents logiciels de recherche de similarité, d'alignements multiples et de construction d'arbres phylogénétiques ainsi que des programmes que nous avons développés spécifiquement afin de permettre l'identification automatique de nouvelles séquences dans les banques de familles homologues. Pour cela, l'application fait appel à différents modules correspondant chacun à un exécutable.

Les trois modules principaux correspondent aux trois étapes nécessaires pour identifier une nouvelle séquence dans les banques de familles de gènes homologues :

- Premier module principal : la détermination de la famille de gènes homologues à laquelle appartient la séquence requête.
- Deuxième module principal : l'alignement de la nouvelle séquence avec les séquences de la famille.
- Troisième module principal : la reconstruction de l'arbre phylogénétique associé.

A ces modules principaux sont ajoutés des modules secondaires permettant des traitements intermédiaires. Un module permet de déterminer les listes de programmes d'alignements et de phylogénies proposées à l'utilisateur et un autre exécute, en une seule étape, l'alignement et la reconstruction phylogénétique en différé sur le serveur de l'application.

4.2 Les langages utilisés

HoSeqI est accessible par une interface Web réalisée en HTML/PHP et Javascript afin de répondre à nos besoins. Cette application utilise également des applets Java pour l'affichage de certains résultats. Enfin les différents modules qu'utilise l'application sont des programmes implémentés en langage C ANSI.

PHP est un langage de script (interprété) exécuté du côté serveur (comme les scripts CGI, ASP, ...). La syntaxe du langage est proche de celle du langage C et du Perl. Il a été créé en 1994 par Rasmus Lerdorf. Depuis, beaucoup d'améliorations ont été apportées et plusieurs versions de PHP se sont succédées. L'application a été implémentée en utilisant la dernière version PHP5. Ses avantages sont multiples : il est simple à utiliser, gratuit, intégré à de nombreux serveurs Web comme Apache, il est possible d'inclure des scripts PHP directement dans une page HTML et il permet d'avoir un site Web dynamique. Il donne également la possibilité d'exécuter des programmes directement dans une page Web et de récupérer les résultats retournés par le programme dans le code PHP. Cette fonctionnalité nous a particulièrement intéressés puisqu'elle nous permet d'avoir une réelle indépendance entre les pages Web et les différents modules. En effet, chaque page web ne

fait qu'exécuter un programme et récupérer les résultats obtenus afin de pouvoir les traiter.

Javascript est un langage de scripts incorporé dans un document HTML mis au point par Netscape en 1995. Historiquement c'est même le premier langage de script pour le Web. Il s'agit d'un langage de programmation qui permet d'apporter des améliorations aux interfaces Web par l'exécution de commandes du côté client. Dans l'application, nous l'utilisons principalement pour vérifier que les données fournies par l'utilisateur sont correctes et pour permettre aux différentes pages de l'application d'être dynamiques.

Les applets Java correspondent à des programmes Java préalablement compilés et incorporés dans des documents HTML afin d'être exécutés *via* la page Web. Une machine virtuelle Java permettant d'interpréter le pseudo-code est chargée en mémoire du côté du client à chaque chargement de la page où est appelée l'applet java, permettant ainsi de lancer l'application Java. Les applets Java peuvent entraîner des ralentissements dûs au lancement de la machine virtuelle coté client. Elles constituent l'une des possibilités qu'il existe pour visualiser facilement et de façon interactive des alignements et des phylogénies à l'aide de programmes Java afin de permettre à l'utilisateur d'analyser les résultats de l'identification obtenus.

Le langage C est utilisé ici pour développer les divers modules d'identification. Il a été choisi pour sa portabilité, sa rapidité d'exécution mais également par souci de cohérence avec les autres développements du PBIL et notamment le système ACNUC et sa bibliothèque C. Cela nous a permis d'utiliser le langage d'interrogation de base de données fourni par ACNUC et d'accéder ainsi facilement aux données des banques.

4.3 L'algorithme d'HoSeqI

Nous avons vu précédemment que l'application était constituée de trois modules principaux et de deux modules secondaires. L'algorithme général de l'application correspond à l'enchaînement des différents modules avec, entre chacun d'eux, des interactions entre l'utilisateur et l'application ainsi que différents tests. Ces tests sont effectués lorsqu'il existe plusieurs possibilités dans le procédé d'identification. A la suite de l'exécution du premier module principal, trois tests sont réalisés. Le premier correspond au cas où plusieurs séquences protéiques obtenues par la traduction dans plusieurs phases de lecture d'une séquence requête nucléotidique, sont retenues (section 4.3.1). Deux autres tests sont effectués dans le cas où plusieurs familles sont déterminées pour la séquence requête (familles dont les séquences ne se recouvrent pas ou familles dont le score global est proche) (section 4.3.1). Un dernier test a lieu une fois que l'utilisateur a choisi les programmes d'alignements et de reconstructions phylogénétiques, afin de déterminer si le temps d'exécution (alignement et construction de l'arbre) permet un traitement immédiat des données (section 4.3.2). L'algorithme général est présenté dans la figure suivante (figure 3.8).

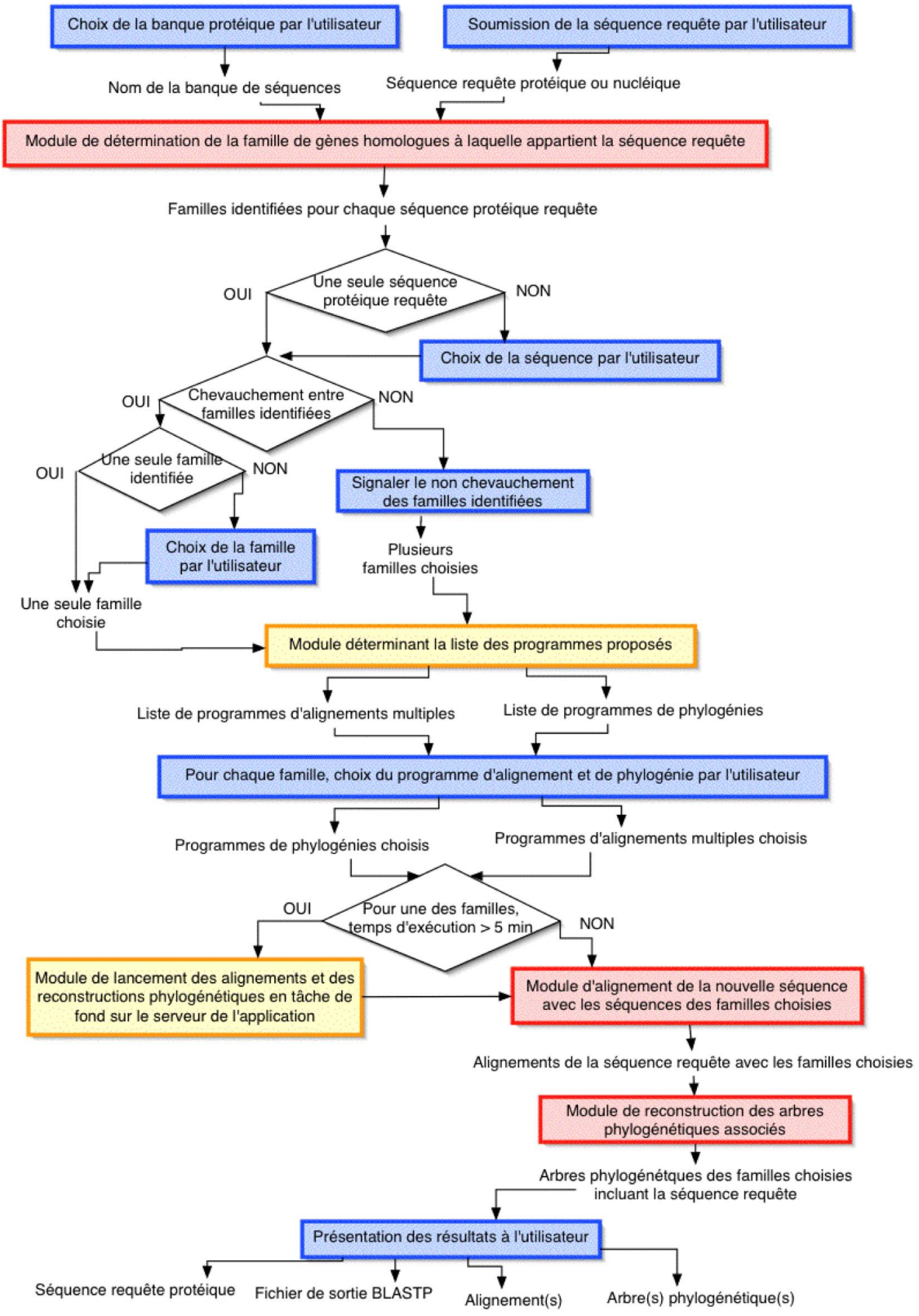

FIG. *3.8: Algorithme général de l'application HoSeqI. Les modules principaux sont représentés en rouge, les secondaires sont en jaune, les étapes d'interaction avec l'utilisateur sont en bleu et les tests sont représentés par des losanges. Tout ce qui n'est pas dans un carré ou un losange correspond à une entrée ou une sortie pour un traitement.*

4.3.1 Le premier module principal

Le procédé d'identification utilise BLASTP2 avec un seuil de *E-value* de 10^4 par défaut pour comparer la séquence requête aux entrées de la banque de familles de séquences protéiques choisie par l'utilisateur. Le programme arrête la comparaison de séquences lorsque la *E-value* obtenue est supérieure à ce seuil. Dans le cas où l'utilisateur soumet une séquence nucléotidique, celle-ci est traduite soit dans la ou les phases indiquées par l'utilisateur, soit dans les six phases. La recherche de similarité est faite pour chacune des séquences protéiques obtenues.

Les sorties de BLASTP2 sont ensuite analysées afin d'identifier à quelle(s) famille(s) la ou les séquences protéiques requêtes appartiennent. Pour chaque fichier de sortie, les HSPs sont sélectionnés et regroupés (section 3.1.2) par séquence. Pour chaque séquence sont calculés un score cumulé et une *E-value* globale. Pour obtenir ces valeurs globales, les scores de similarité de chaque HSP sélectionné sont additionnés et leurs *E-values* sont multipliées. Les séquences retenues pour la suite du traitement sont celles ayant une *E-value* globale inférieure au seuil que nous avons fixé, *i.e.* la *E-value* du meilleur alignement obtenu entre la séquence requête et les séquences de la banque x 10^5 (section 3.1.2).

Puis nous déterminons la famille à laquelle chaque séquence sélectionnée appartient, ce qui permet de regrouper toutes les séquences retenues selon leur famille. Pour chaque ensemble de séquences, c'est-à-dire pour chaque famille, le score global est calculé à partir des scores cumulés de chaque séquence composant la famille. Le calcul du score global correspond à une moyenne des scores cumulés pondérés par les rangs des séquences. Plus une séquence est similaire à la séquence requête, plus son rang est élevé. Par exemple, s'il y a dix séquences dans une famille, la séquence la plus similaire à la séquence requête aura comme rang la valeur 10, et celle qui est le moins similaire aura une valeur de rang de 1. Nous utilisons cette moyenne pondérée par les rangs de manière à accorder plus de poids aux séquences sujets qui obtiennent les meilleurs scores d'alignement avec la séquence requête. En effet, considérons le cas suivant : dans la liste des séquences sujets produisant un alignement significatif avec la séquence requête du fichier de sortie BLASTP2 il y a plusieurs séquences appartenant à une famille (famille 1) avec, au milieu de celles-ci, une séquence isolée appartenant à une autre famille (famille 2) (figure 3.9). Si le score global de chaque famille est calculé en utilisant une moyenne (non pondérée) des scores cumulés correspondant à chaque séquence, alors le score global de la famille 1 risque d'être inférieur à celui de la famille 2. La famille 2 serait donc sélectionnée comme étant celle à laquelle la séquence requête appartient. Or il est plus probable que la famille de la séquence requête soit la famille 1. En utilisant une moyenne des scores cumulés pondérés par les rangs des séquences, nous évitons ces types d'erreurs.

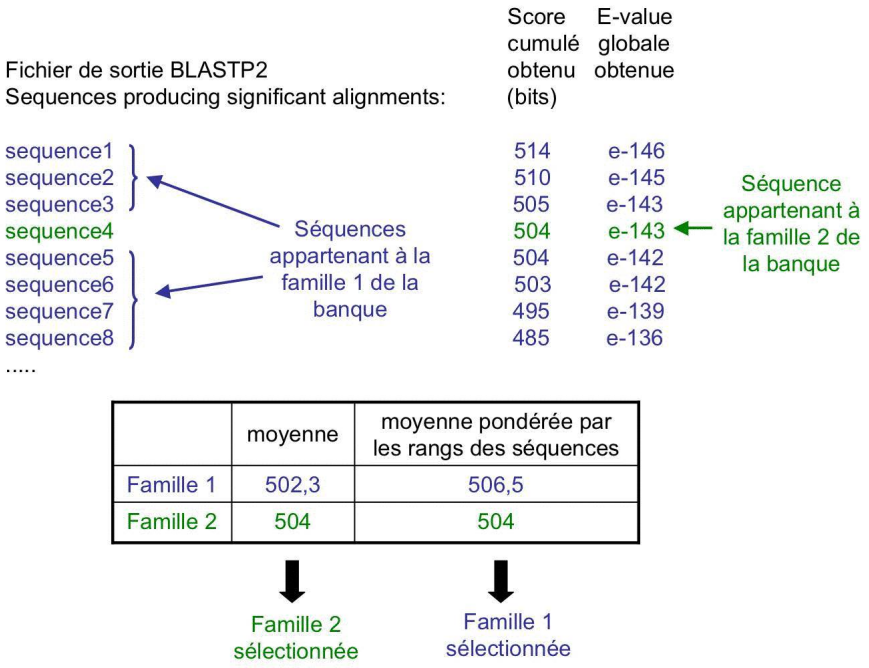

FIG. *3.9: Exemple d'un cas justifiant l'utilisation d'une moyenne des scores cumulés pondérés par les rangs des séquences pour calculer le score global d'une famille.*

En comparant les scores globaux, il est possible de déterminer les meilleures familles c'est-à-dire toutes celles de plus haut score ou dont le score est proche du plus haut score (à 5 *bits* près). Si la séquence soumise est nucléotidique et qu'elle est traduite dans différentes phases, plusieurs des séquences protéiques obtenues peuvent obtenir des résultats significatifs lors de la comparaison avec la banque effectuée en utilisant BLASTP2. Dans ce cas une sélection se fait également à cette étape : ne sont retenues que les séquences requêtes dont le score global des meilleures familles identifiées précédemment est le plus haut ou est proche du plus haut score (à 5 *bits* près).

De plus, la position initiale minimale et la position finale maximale des alignements de chaque séquence d'une famille avec la séquence requête sont notées afin de détecter d'éventuels non-chevauchements entre les familles de plus haut score global et les autres familles. Toutes les familles distinctes dont les séquences s'alignent avec la séquence requête sans chevauchement sont sélectionnées (figure 3.10). Cela permet de prendre en compte les séquences de familles de gènes homologues distinctes qui contiennent des régions non-chevauchantes et qui correspondent à des fusions de gènes.

En fin de traitement, on obtiendra donc une liste des meilleures familles (non-chevauchantes ou de meilleurs scores) pour chaque séquence protéique requête retenue. Un schéma explicatif de l'algorithme est présenté dans la figure 3.11.

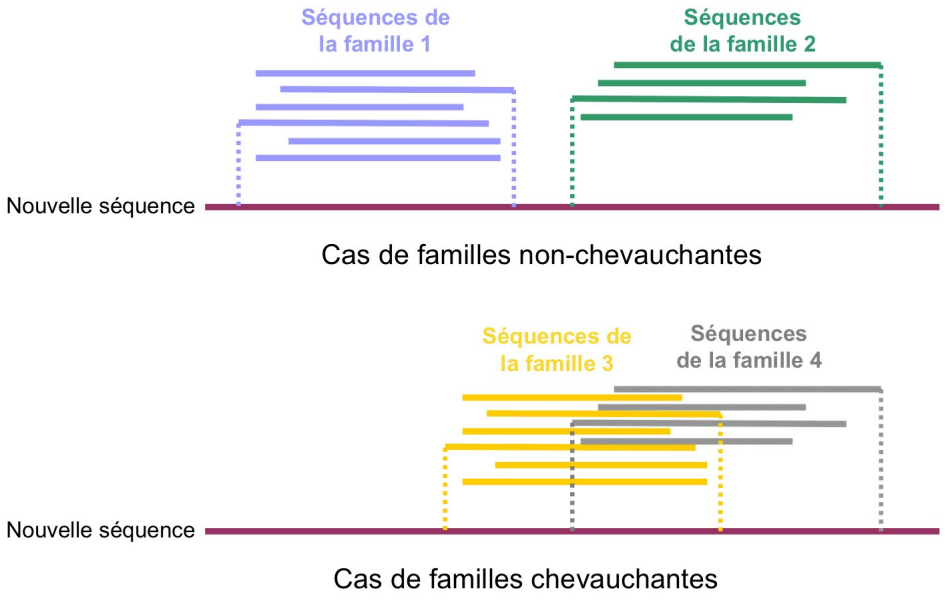

FIG. *3.10: Exemple de deux familles non-chevauchantes et deux familles chevauchantes.*

A la suite de ce module, comme il est montré dans l'algorithme général, s'il y a plusieurs séquences requêtes sélectionnées, l'utilisateur devra choisir celle qu'il souhaite utiliser pour l'identification. De la même manière, s'il y a plusieurs familles de score proche identifiées, l'utilisateur devra choisir celle avec laquelle il souhaite continuer le processus d'identification. Par contre, toutes les familles non-chevauchantes seront conservées.

4.3.2 Le module secondaire permettant le choix des listes de programmes d'alignements et de phylogénies

Le module secondaire permettant de déterminer la liste des programmes proposés est utilisé pour calculer le nombre de séquences des familles identifiées et regarder si le fichier d'alignement de la famille est stocké dans la banque. Cela nous permet de savoir si l'alignement des séquences de la famille avec la séquence requête doit être recalculé en entier ou si l'ajout de la nouvelle séquence requête à l'alignement pré-existant suffit. De plus, la connaissance du nombre de séquences nous aide à évaluer le temps d'exécution des programmes d'alignements et de reconstructions phylogénétiques.

En utilisant les tableaux de tests présentés précédemment (section 3.2 et 3.3), nous avons mis en place des filtres afin de proposer des traitements adaptés aux familles sélectionnées. Pour les programmes d'alignements, la liste des programmes est modifiée en fonction du nombre de séquences que la famille traitée contient. Ainsi lorsque les temps de traitements d'un programme sont inférieurs à cinq minutes (temps d'exécution maximum considéré comme acceptable pour une application Web), celui-ci est proposé à l'utilisateur.

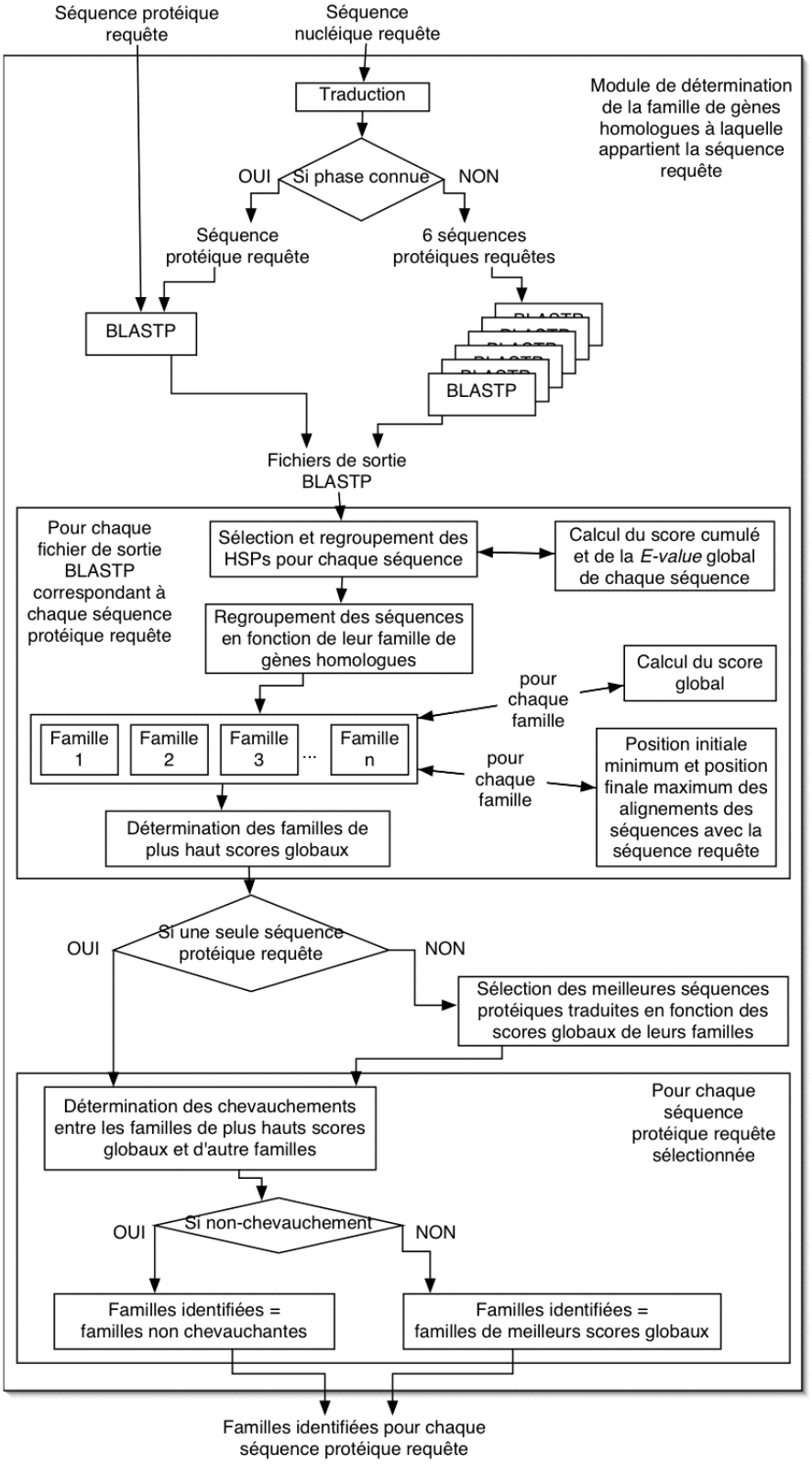

FIG. *3.11: Schéma explicatif de l'algorithme du premier module permettant la détermination de la ou des familles de gènes homologues auxquelles appartient la séquence requête*

Par contre, un programme est enlevé de la liste, s'il ne peut pas gérer le nombre de séquences de la famille traitée (figure 3.12). Pour les programmes de reconstructions phylogénétiques, à partir d'un certain seuil du nombre de séquences, en fonction du programme choisi par l'utilisateur, la construction de l'arbre nécessite un intervalle de temps que nous estimons trop grand pour pouvoir être réalisée dans l'immédiat par une application Web (plus de cinq minutes). Tous les traitements (calcul des alignements et constructions phylogénétiques) se feront alors en différé sur le serveur (figure 3.13). Dans tous les cas, si le nombre de séquences de la famille dépasse 2500 séquences, les calculs seront lancés automatiquement en différé car les programmes de reconstructions d'arbres ne peuvent s'exécuter en moins de cinq minutes au delà de 2500 séquences.

Programmes	Ajout d'une séquence à l'alignement pré-existant	Alignement recalculé en entier
MABIOS	500	100
CLUSTAL W	500	100
MULTALIN	2500	500
MENTALIGN	2500	100
MUSCLE	2000	500
MUSCLE-prog	2500	1500
MUSCLE-fast	2500	2500

FIG. *3.12: Tableau représentant le nombre de séquences maximum que chacun des programmes d'alignements peut traiter en un temps d'exécution inférieur à 5 minutes si on ajoute une séquence à un alignement pré-existant d'une part ou si on recalcule l'alignement complet d'autre part. Le nombre limite est 2500 car c'est le nombre maximum de séquences que peut traiter un des programmes de phylogénies en moins de 5 minutes.*

programme de phylogénie	arbre sans bootstrap	arbre avec bootstrap de 50	arbre avec bootstrap de 100	arbre avec bootstrap de 500	arbre avec bootstrap de 1000	arbre avec bootstrap de 1500
QUICKTREE	2500	450	350	150	120	100
BIONJ	1000	300	200	100	90	70
FastME	1500	400	250	100	100	70
PhyML	20	0	0	0	0	0

FIG. *3.13: Tableau représentant le nombre de séquences maximum que chacun des programmes de phylogénies peut traiter en un temps d'exécution inférieur à 5 minutes en fonction de l'option de bootstrap et du nombre de réplicats utilisé.*

L'utilisateur peut alors choisir parmi les programmes proposés un programme d'alignements multiples et un programme de reconstructions phylogénétiques pour chacune des familles sélectionnées.

4.3.3 Le module secondaire permettant de lancer en différé sur le serveur le calcul de l'alignement et la reconstruction phylogénétique

Le deuxième module secondaire permet de lancer en différé sur le serveur le calcul de l'alignement et la reconstruction phylogénétique en utilisant les programmes choisis par l'utilisateur. Celui-ci sera averti par mail lorsque les calculs seront terminés et qu'il

pourra accéder aux résultats, c'est-à-dire aux alignements et aux arbres phylogénétiques incluant la séquence requête et les séquences de chaque famille sélectionnée. Ce module lance en différé sur le serveur les deux modules principaux de calcul de l'alignement et de reconstruction plylogénétique. Il est utilisé uniquement dans le cas où le temps de traitement est trop grand pour une exécution immédiate de l'application Web.

4.3.4 Le deuxième module principal

Le deuxième module principal correspond aux alignements de la nouvelle séquence avec les séquences de sa ou ses familles de gènes homologues. Si plusieurs familles non-chevauchantes sont identifiées, l'alignement devra être calculé pour chacune des familles. Les programmes proposés (CLUSTAL W, MULTALIN, MENTALIGN, MUSCLE, MUSCLE-prog, MUSCLE-fast et MABIOS) exigent des traitements spécifiques des données avant de pouvoir calculer l'alignement. Ces traitements sont différents selon les données extraites, c'est-à-dire si l'alignement des séquences de la famille existe ou si l'on extrait juste l'ensemble des séquences de la famille non alignées.

4.3.5 Le troisième module principal

Le troisième module principal permet de reconstruire les arbres phylogénétiques contenant la séquence requête et les séquences de la ou des familles sélectionnées. De la même manière que pour le module précédent, cette reconstruction se fera pour chacune des familles identifiées. Tous ces calculs doivent être faits aussi rapidement que possible et doivent pouvoir être réalisés avec un grand nombre de séquences. Pour chaque programme de reconstructions phylogénétiques proposé (BIONJ, FASTME, QUICKTREE, PHYML), l'utilisateur peut utiliser l'option de bootstrap et indiquer le nombre de réplicats de bootstrap qu'il souhaite utiliser. L'arbre est ensuite automatiquement raciné par la méthode du *midpoint*.

4.4 L'interface web et la présentation des résultats

L'interface Web de l'application permet d'interagir avec l'utilisateur. Celui-ci est sollicité à chaque étape du processus d'identification afin de fournir aux différents modules les données et les paramètres utiles au bon déroulement des traitements. L'interface se découpe en plusieurs pages correspondant aux différentes étapes de l'algorithme.

Tout d'abord l'utilisateur doit fournir les différentes informations qui vont être utilisées pour la première étape de l'identification c'est-à-dire la détermination de la famille à laquelle appartient la séquence soumise. Cela se fait en utilisant un formulaire (figure 3.14) dans lequel l'utilisateur doit donner les informations suivantes :

- Le type de la séquence doit être renseigné.
- Si c'est une séquence nucléotidique, il faut indiquer si la phase de traduction est connue et la choisir si c'est le cas.

– La séquence requête doit être fournie.
– La banque de séquences protéiques à utiliser pour l'identification est à choisir dans une liste des banques disponibles sur le serveur.
– Les paramètres du programme BLASTP2 peuvent être modifiés.

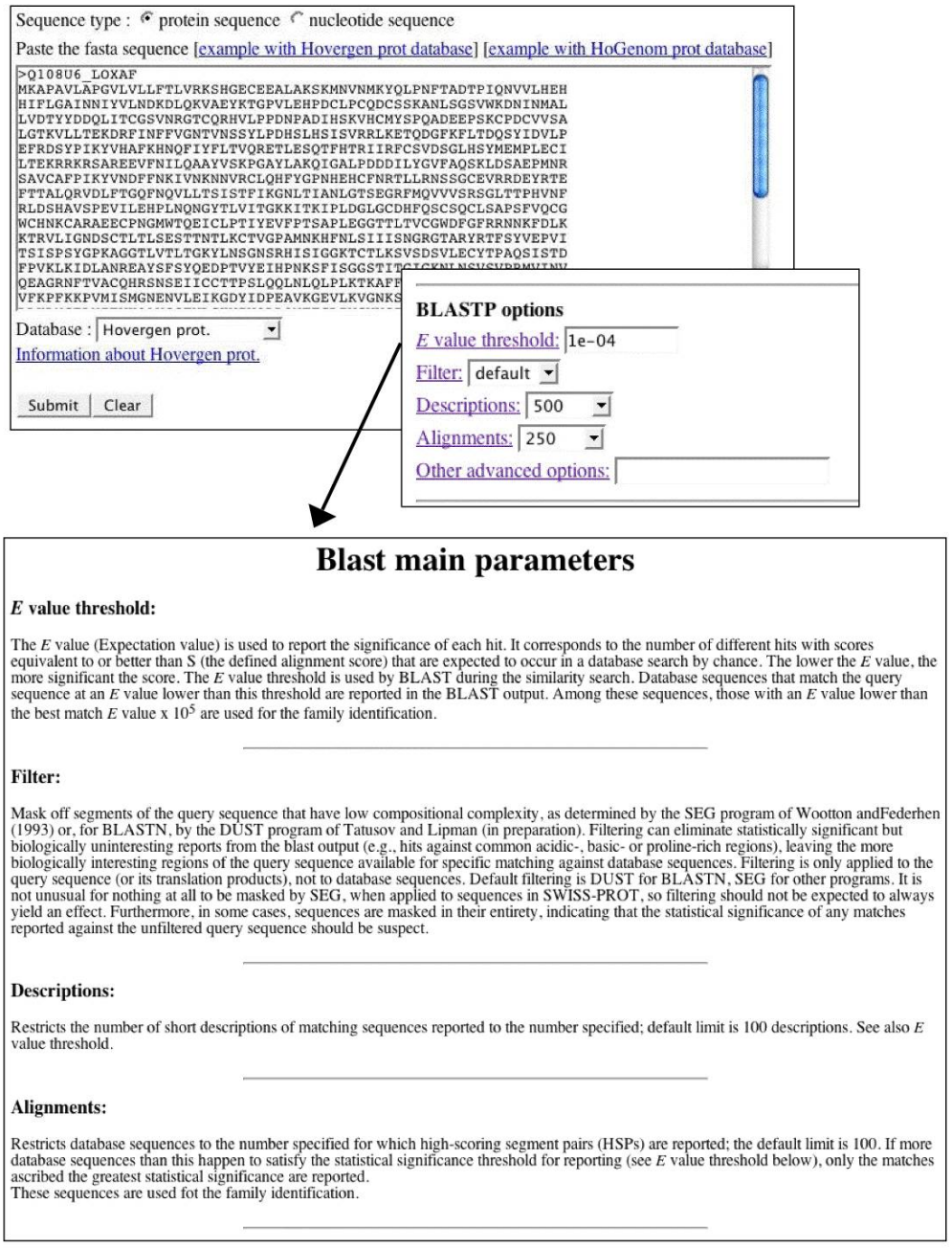

FIG. *3.14: Première page de l'application. Il s'agit ici de l'identification de la séquence protéique de nom Q108U6_LOXAF dans la banque HOVERGEN. Les paramètres par défaut de BLAST ne sont pas modifiés. Un lien sur une fiche d'explication permet d'obtenir des informations sur chaque paramètre de BLAST afin d'aider le choix de l'utilisateur.*

Les données du formulaire sont vérifiées, notamment au niveau de la syntaxe (la

séquence requête doit être au format FASTA, *etc*), puis ces informations sont utilisées comme paramètres d'entrée du premier module principal permettant la détermination de la famille de gènes à laquelle appartient la séquence requête.

Par ailleurs, l'utilisateur doit faire différents choix par l'intermédiaire de divers formulaires lors de la procédure d'identification. Ainsi, un formulaire permet de choisir entre les différentes séquences candidates dans le cas d'une séquence requête nucléotidique traduite dans plusieurs phases. Un autre formulaire permet de sélectionner la famille de gène que l'utilisateur veut utiliser pour l'identification dans le cas où plusieurs familles de score global proche sont retenues. Enfin il doit choisir les programmes d'alignements ou de phylogénies que l'application utilise pour l'identifciation. Pour l'aider dans ces choix, l'interface propose des liens sur les fichiers de sortie de BLASTP2 obtenus et sur les informations concernant chaque famille proposée (figure 3.15).

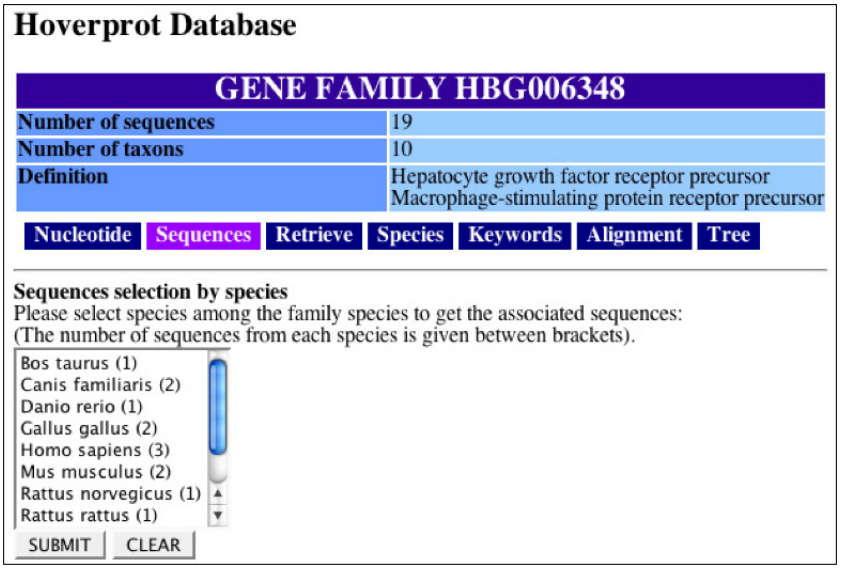

FIG. *3.15: Informations concernant la famille de gènes homologues HBG006348 de la banque HOVERGEN utilisée pour l'identification.*

De plus, pour les programmes d'alignements et de phylogénies, l'interface permet à l'utilisateur de modifier les valeurs des paramètres utilisés par défaut et d'utiliser, en particulier pour les programmes de reconstructions d'arbres, une option de bootstrap en indiquant le nombre de réplicats (figure 3.16). Toutes ces informations sont transmises en tant que paramètres aux différents modules de traitements. L'utilisateur est également informé dans le cas où les programmes et les options choisis entraînent des calculs trop longs qui seront effectués en différé sur le serveur et, dans ce cas, son adresse mail lui est demandée.

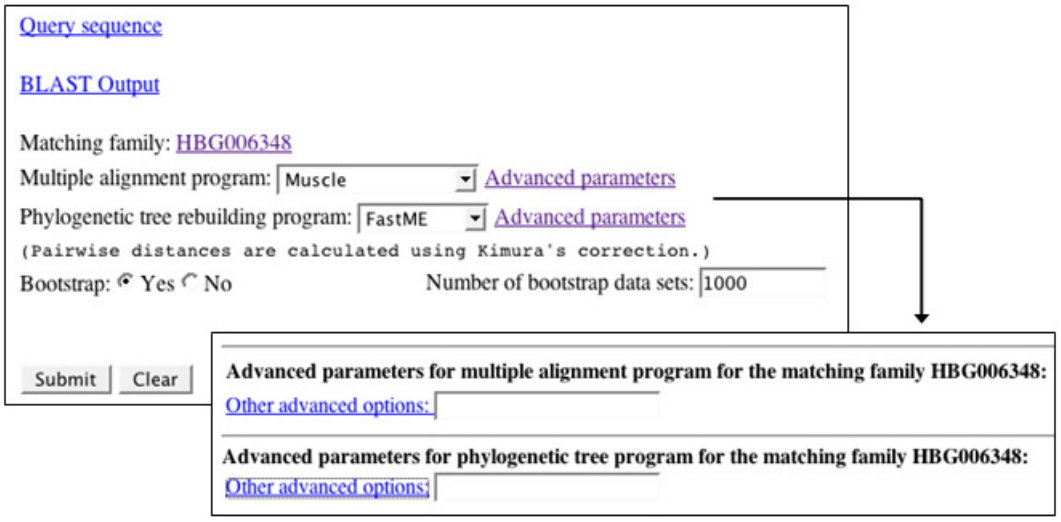

FIG. *3.16: Formulaire correspondant à l'étape de choix des programmes d'alignements multiples et de reconstructions phyligénétiques. Ici, MUSCLE a été choisi comme programme d'alignements multiples. FASTME est utilisé pour la reconstruction phylogénétique avec une option de bootstrap et un nombre de réplicats de 1000. Pour chaque programme, les paramètres par défaut peuvent être modifiés. Des liens sur des fiches d'explications permettent d'obtenir des informations sur les différents paramètres.*

Dans le cas où les traitements sont immédiats, une fois que tous les calculs sont effectués, les résultats sont présentés à travers une page Web. L'utilisateur a accès aux fichiers de sortie BLASTP2, aux informations concernant les familles sélectionnées, aux séquences requêtes, aux alignements multiples et aux arbres phylogénétiques (figure 3.17). Tous ces fichiers peuvent être téléchargés. De plus, les alignements et les arbres phylogénétiques obtenus peuvent être visualisés à l'aide de deux applets Java : Jalview (http ://www2.ebi.ac.uk/ michele/jalview/) (figure 3.18) et ATV (Zmasek & Eddy, 2001) (figure 3.19).

Il est ainsi possible d'étudier l'emplacement de la nouvelle séquence dans l'arbre phylogénétique de la ou des familles afin de l'identifier. L'applet Java ATV permet d'obtenir des informations sur toutes les séquences de l'arbre. En sélectionnant une séquence, on accède aux annotations de celle-ci dans la banque utilisée.

Lorsque les calculs sont exécutés en différé sur le serveur, l'utilisateur reçoit un mail avec des liens sur les différents résultats conservés sur le serveur pendant un mois. Ces liens lui permettent soit de télécharger les fichiers, soit de visualiser les résultats directement, notamment avec les applets Java pour les alignements et les arbres phylogénétiques.

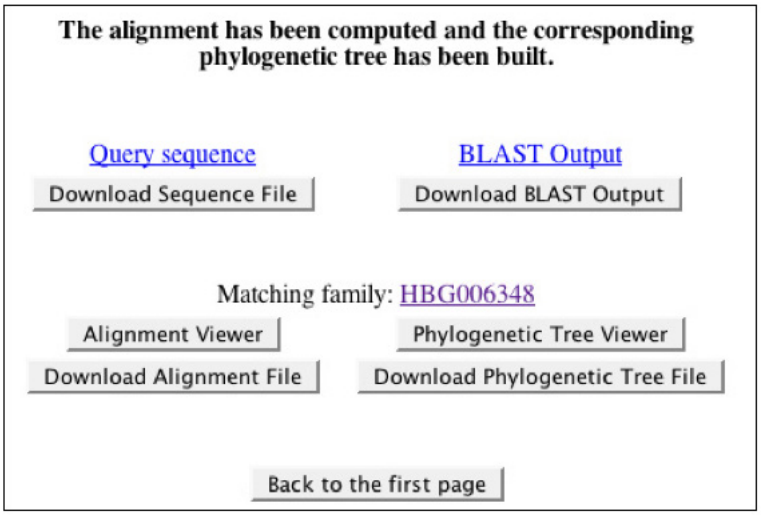

FIG. *3.17: Page Web présentant les résultats de l'identification à l'utilisateur. La séquence requête et le fichier de sortie BLAST peuvent être visualisés ou téléchargés. Les informations sur la famille de gènes homologues HBG006348 sélectionnée sont disponibles également. Enfin, l'alignement et l'arbre phylogénétique peuvent être visualisés ou téléchargés.*

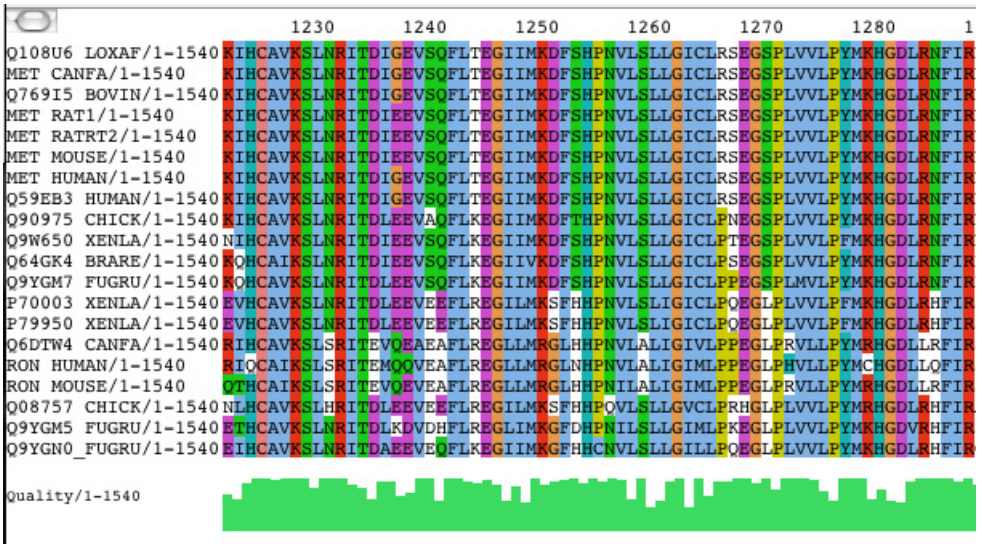

FIG. *3.18: Visualisation de l'alignement de la séquence requête avec les séquences de la famille sélectionnée en utilisant l'applet Java Jalview. La séquence requête Q108U6_LOXAF identifiée est la première de l'alignement.*

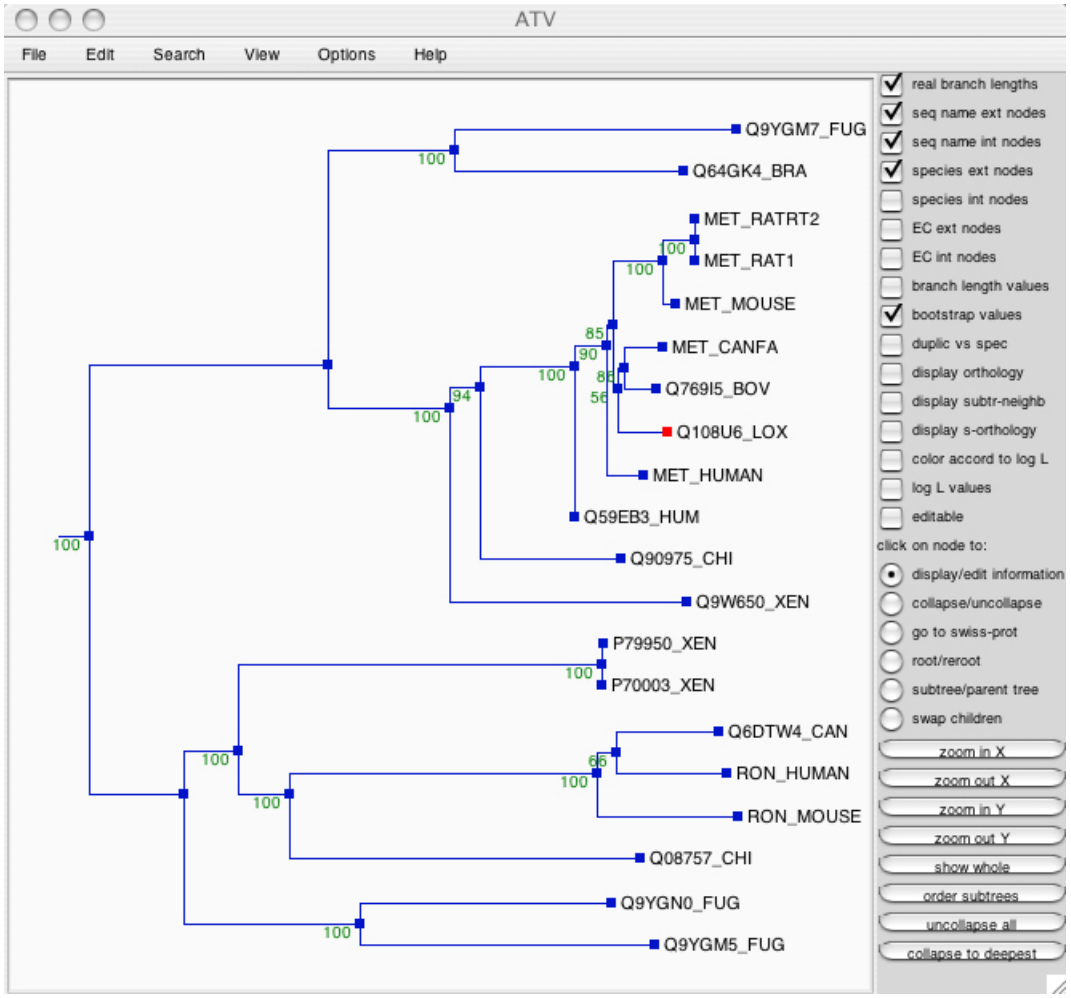

FIG. *3.19: Visualisation de l'arbre phylogénétique de la famille incluant la séquence requête en utilisant l'applet Java ATV. La séquence requête Q108U6_LOXAF correspond à celle dont la feuille de l'arbre est rouge. Les valeurs de bootstrap sont également visibles sur les noeuds de l'arbre.*

5 Conclusions

HoSeqI permet d'automatiser le procédé d'identification sur des grandes banques de familles de gènes homologues et, ce faisant, rend possible l'analyse phylogénétique des séquences soumises. L'application propose une interface Web (`http://pbil.univ-lyon1.fr/software/HoSeqI/`) facile à utiliser qui permet à un utilisateur d'identifier une séquence requête, c'est-à-dire de trouver la famille de la banque à laquelle elle appartient, puis, de visualiser l'alignement et l'arbre phylogénétique obtenus. Le principe d'identification est basé sur l'approche phylogénétique et permet à l'utilisateur de localiser la séquence dans l'arbre de la famille de gènes afin d'étudier son origine évolutive.

Dans un premier temps, l'application a été implémentée en local avec la banque

HOBACGEN. Nous avons testé l'application avec des séquences déjà présentes dans les banques de données afin de vérifier la cohérence de l'identification. La famille identifiée pouvait être facilement vérifiée et les séquences ajoutées devaient être placées à côté d'elles mêmes dans l'arbre phylogénétique reconstruit. Ces tests nous ont permis d'apporter diverses modifications afin d'améliorer l'algorithme de détermination de la ou des familles de gènes homologues sélectionnées et d'obtenir 100% de bons résultats avant d'installer l'application sur le serveur. Nous avons également testé la version d'HoSeqI installée sur le serveur PBIL en utilisant des séquences du clade *Elephantidae* provenant de la banque généraliste UNIPROT et non présentes dans les banques HOVERGEN et HOGENOM. En utilisant les informations des différentes annotations des séquences fournies par UNIPROT, nous avons pu vérifier les résultats obtenus et valider la qualité des identifications. Ainsi nous avons obtenu 95% des nouvelles séquences classées dans les bonnes familles et 5% de séquences pour lesquelles étaient proposées plusieurs familles pour l'identification. Les incertitudes d'identification peuvent être dues à la longueur de la séquence à traiter. S'il s'agit d'une séquence partielle, celle-ci pourra plus facilement s'aligner correctement avec un grand nombre de séquences, et donc plusieurs familles pourront être sélectionnées. D'autres incertitudes peuvent être dues au fait que certaines familles sont constituées de séquences proches entraînant la sélection de toutes ces familles lors de la détermination de celle de la nouvelle séquence.

Pour l'évaluation du temps de traitement, il faut considérer le temps d'exécution des différents modules de l'algorithme puisque, entre chacun d'eux, l'utilisateur peut être sollicité. Au niveau du temps d'exécution de la première étape de l'identification (le premier module), c'est-à-dire la détermination de la ou des familles auxquelles appartient la séquence soumise, le temps de traitement varie en moyenne entre une dizaine de secondes et 1 minute en fonction du volume de la banque utilisée et de la charge de travail du serveur. Cet intervalle de temps correspond en majeure partie au temps d'exécution du programme BLAST. Pour le calcul de l'alignement et la reconstruction phylogénétique, nous avons vu précédemment que le temps d'exécution dépendait du programme utilisé et du nombre de séquences traitées (section 3.2 et 3.3). Un temps global d'exécution a été évalué en utilisant les programmes d'alignements multiples MUSCLE-prog ou MUSCLE-fast et le programme de phylogénies QUICKTREE. Nous avons ainsi obtenu un temps de calcul de 30s pour une séquence requête attribuée à une famille de 143 séquences avec MUSCLE-prog et de 2 minutes et 30 secondes pour une famille de 1132 séquences avec MUSCLE-fast.

Chapitre 4

Deux utilisations et applications des méthodes développées dans HoSeqI

Les méthodes développées pour l'application Web HoSeqI nous ont permis de réaliser deux autres applications basées sur l'identification automatique de séquences. Nous avons développé une application pour l'ajout de séquences de génomes aux banques de familles de gènes homologues afin de permettre l'étude de l'évolution de ces génomes. Nous avons également travaillé sur le développement d'un outil de détection automatique de séquences chimères d'ARNr 16S et d'identification des séquences non-chimères.

1 L'ajout des séquences de génomes aux banques de familles de gènes homologues

Nous avons utilisé les différents développements de l'application Web HoSeqI décrite précédemment pour créer une autre application permettant une identification rapide de plusieurs milliers de séquences. Pour cela, nous avons modifié la version Web d'HoSeqI et créé un logiciel. Il est ainsi possible de donner à ce programme un fichier contenant n séquences et d'identifier automatiquement ces n séquences. Nous avons utilisé cette application pour ajouter les séquences de nouveaux génomes aux grandes banques de familles de gènes homologues développées au PBIL et permettre différentes études phylogénétiques. Le prototypage de cette fonctionnalité et l'analyse des résultats obtenus ont été réalisés dans le cadre d'une collaboration avec Philippe Normand (Laboratoire d'Ecologie Microbienne des Sols - UMR CNRS 5557 - Lyon1), Vincent Daubin et Simon Penel. Nous avons ainsi contribué à l'ajout des séquences protéiques de deux génomes de bactéries du genre *Frankia* à une version de la banque HOGENOM contenant déjà un génome de *Frankia*. Le but était d'étudier l'évolution des génomes de ces bactéries et notamment de pouvoir détecter d'éventuels transferts horizontaux de gènes.

1.1 Les bactéries du genre *Frankia*

Les bactéries du genre *Frankia* appartiennent à la classe des Actinobactéries (figure 4.1). Ces bactéries ont un génome de grande taille avec un pourcentage élevé en bases G et C (haut G+C%). On retrouve parmi les Actinobactéries notamment *Mycobacterium* (agents de la tuberculose et de la lèpre) et *Streptomyces* (bactéries du sol, à l'origine de nombreux antibiotiques). Douze groupes génomiques qui ont rang d'espèces sont décrites à ce jour chez *Frankia*.

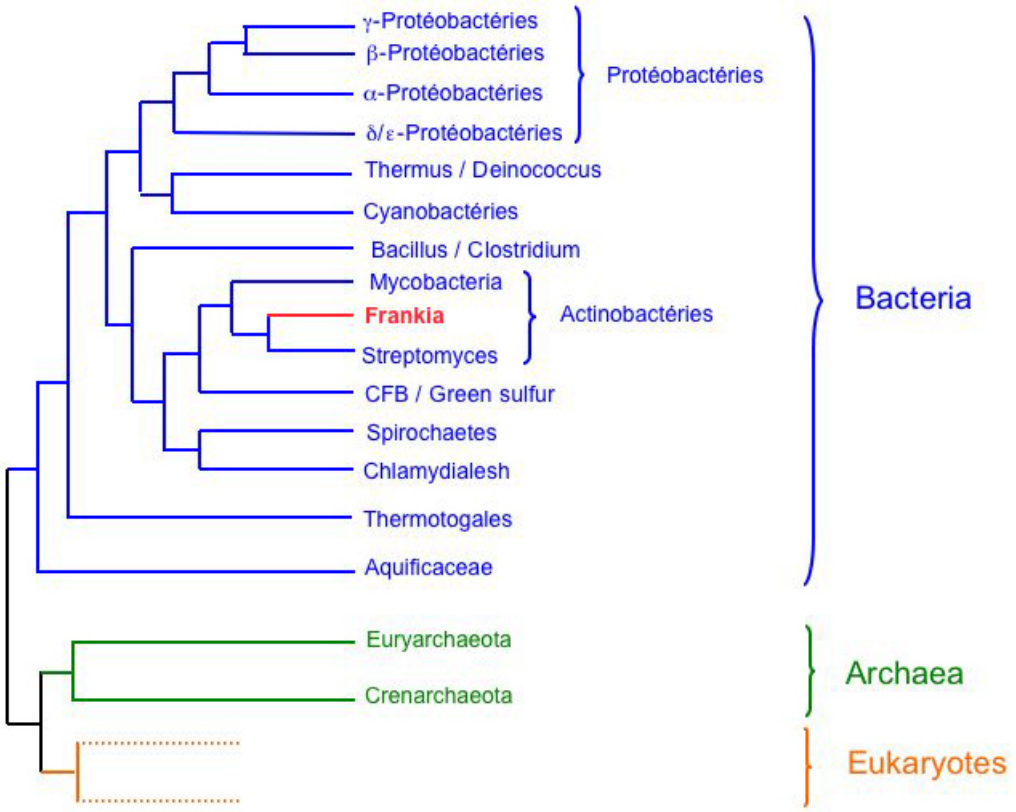

FIG. *4.1: Arbre phylogénétique du vivant basé sur (Brown* et al.*, 2001). Le genre Frankia est présenté en rouge dans l'arbre.*

Ces bactéries fixent l'azote, ceci en symbiose avec un large spectre de plantes dicotylédones, appelées actinorhiziennes (figure 4.2). Ces plantes, avec leurs bactéries symbiotiques, sont collectivement responsables d'environ 15% des entrées d'azote fixé biologiquement sur Terre. Lors de cette interaction, des nodules se forment au niveau des racines de la plante hôte (figure 4.2). A l'intérieur de ces nodules, la bactérie va pouvoir se multiplier sans compétition et transformer l'azote atmosphérique en ammoniac directement assimilable par la plante. Cette association présente un intérêt écologique majeur notamment dans le cadre de la reforestation de sols pauvres et dégradés ou de la prévention de la désertification de sols soumis à une sévère érosion. Par ailleurs, de

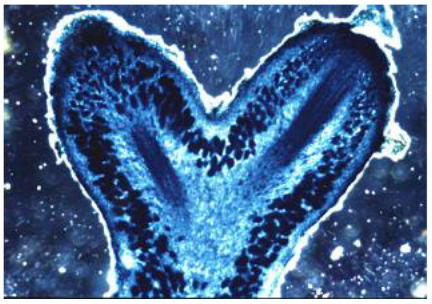
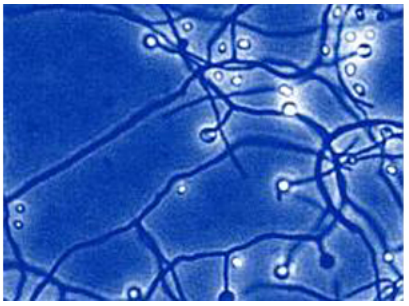

FIG. *4.2: La photo de gauche est une section longitudinale d'un nodule ramifié de racine d'aulne. (photo P. Normand). La photo de droite représente une culture de Frankia alni présentant des hyphes (filaments) ramifiées et septées (pourvues de cloisons transversales nommées septum) et des diazovésicules, cellules spécialisées dans la fixation de l'azote (photo Y. Hammad).*

nombreuses plantes actinorhiziennes sont utilisées par l'industrie pharmaceutique en raison d'une très grande production de molécules phénoliques possédant diverses activités : antimicrobiennes, antioxydantes, antivirales, anti-inflammatoires, antispasmodiques, antitumorales, *etc.*

L'équipe de Philipe Normand cherche à comprendre les bases physiologiques de cette symbiose en analysant les modifications qui se produisent au cours de la mise en place de celle-ci. Sur cette bactérie, pour laquelle aucun outil de transformation génétique n'est disponible, la connaissance des gènes impliqués dans la symbiose passe par l'étude du génome. Par des études de biodiversité et de phylogénie, aussi bien au niveau du partenaire bactérien que de la plante hôte, il est possible de chercher à comprendre comment s'est faite l'évolution de la symbiose.

Trois souches de *Frankia* étaient disponibles en 2006 : HFPCcI3 (CcI3), EAN1pec (EAN) et ACN14a (ACN). Les deux premières on été séquencées au DOE Joint Genome Institute en collaboration avec D. Benson (Université du Connecticut) et L. Tisa (Université de New Hampshire). La troisième souche (ACN) a été séquencée au Génoscope en collaboration avec P. Normand. Les génomes de ces souches sont circulaires et leur taille varie entre 5,38 millions de paires de bases (Mb) pour CcI3, 7,50 Mb pour ACN et 9,08 Mb pour EAN.

1.2 La détection de transferts horizontaux chez les bactéries

Un transfert horizontal de gène correspond à un échange de matériel génétique entre des organismes non forcément apparentés. C'est à partir de 1950, lorsque des résistances multiples à des antibiotiques ont été découvertes, que l'importance de ces transferts sur l'évolution des bactéries a commencé à être comprise (Davies, 1996). En effet, la rapidité avec laquelle certaines bactéries développaient une résistance au même spectre d'antibio-

tiques indiquait que ces caractéristiques étaient transférées entre les taxons plutôt que générées dans chaque lignée (Ochman *et al.*, 2000).

L'arrivée massive de données de séquences a rapidement montré que ces événements étaient trop nombreux pour être négligés. En 1991, Médigue et al. (Medigue *et al.*, 1991) font la première analyse mettant en évidence le caractère massif des transferts. Ils ont montré qu'il existait une déviation significative du modèle général de l'usage du code pour certains gènes montrant une relation claire avec des gènes de bactériophages et de flagelles. Il a donc été proposé que ces gènes avaient été acquis par transferts à partir de différentes sources. Hilario et Gogarten (Hilario & Gogarten, 1993) ont également signalé l'importance des transferts en observant qu'il était parfois impossible de construire des phylogénies congruentes à partir de différents jeux de gènes orthologues. En outre, Lawrence et Ochman (Lawrence & Ochman, 1998) ont estimé à 18% l'impact global des transferts horizontaux de gènes sur l'évolution du génome d'*Escherichia coli*, lui permettant ainsi d'acquérir des propriétés pour explorer d'autres niches écologiques, inaccessibles autrement.

Puis, le séquençage de multiples génomes bactériens a permis de réaliser des analyses de génomique comparative. Elles ont révélé qu'il pouvait y avoir de grandes différences dans les répertoires géniques des bactéries, même entre celles qui sont proches évolutivement. Ainsi l'évolution des génomes ne peut pas être uniquement expliquée par la transmission verticale des gènes (échange de matériel génétique entre des organismes appartenant à la même espèce). De plus, toutes les catégories de gènes sont susceptibles d'être transférées. Cependant il existe des différences au niveau de la fréquence de transmission en fonction des gènes (Jain *et al.*, 1999), certains sont plus transférés que d'autres, comme les classes de gènes non candidats aux analyses phylogénétiques (*i.e.* des gènes n'ayant pas assez d'homologues pour pouvoir être utilisés afin d'effectuer des analyses phylogénétiques) (Daubin *et al.*, 2003) et selon les groupes d'organismes les transferts ne sont pas les mêmes (Ochman *et al.*, 2000).

Il existe différentes méthodes pour identifier les gènes acquis par transfert horizontal : des approches paramétriques (utilisant des indices tels que l'usage du code, le contenu en GC, l'usage des mots, *etc*), des méthodes basées sur les similarités BLAST, l'utilisation de profils phylogénétiques et des approches basées sur les arbres phylogénétiques. Toutes ces méthodes reposent sur le fait que les séquences acquises ont des caractéristiques particulières qui les différencient des autres séquences du génome hôte.

L'analyse de la topologie des arbres phylogénétiques est l'un des plus fiables moyens pour identifier des événements de transferts horizontaux (figure 4.3). Si, par exemple, dans un arbre une séquence bactérienne est groupée avec ses homologues archées plutôt qu'avec les autres bactéries, on peut en déduire que cette séquence a probablement été transférée horizontalement chez cette bactérie à partir d'une espèce d'archée.

La topologie d'un arbre est un bon indicateur de l'histoire évolutive uniquement si elle est soutenue statistiquement par des analyses de bootstrap par exemple. Cette topologie peut être comparée à celle d'un arbre de référence afin de repérer des positions anormales

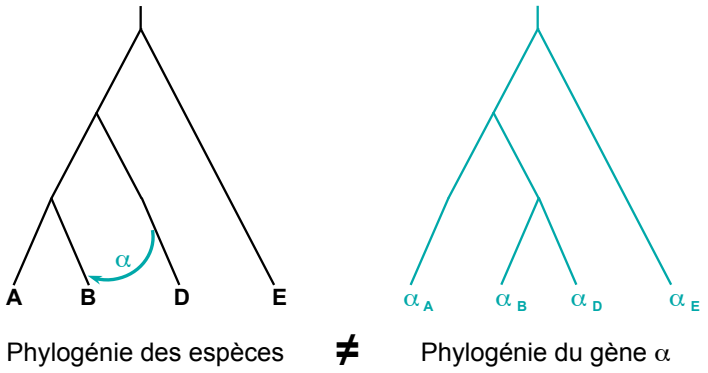

FIG. *4.3: Impact d'un transfert horizontal sur une phylogénie. Si une copie du gène α a été transférée de l'espèce D vers l'espèce B, alors la phylogénie reconstruite à partir des séquences de ce gène sera différente de la phylogénie des espèces. Extrait de (Calteau, 2005).*

de gènes par rapport à celles de leur espèce. Cependant l'arbre du vivant est loin d'être résolu puisque les relations entre certains phyla, particulièrement chez les procaryotes, ne sont pas établies avec certitude. La topologie de l'arbre de référence peut donc influer sur la détection des événements de transferts.

1.3 Une application basée sur les modules de l'application Web HoSeqI pour l'ajout de séquences de *Frankia* à HOGENOM

Nous avons développé une application afin de permettre l'ajout des séquences protéiques de génomes de *Frankia* de la souche CcI3 (4557 séquences) et de la souche EAN (7976 séquences) à une version locale de la banque HOGENOM, construite à partir de la version 2 (octobre 2004) de celle-ci. Cette banque contient les séquences de 182 autres génomes complets dont des génomes d'eucaryotes, d'archées et de bactéries et particulièrement 13 génomes d'Actinobactéries incluant les genres *Mycobacterium*, *Corynebaterium*, *Spreptomyces*, *Tropheryma* et *Bifidobacterium*. La version locale d'HOGENOM que nous utilisons correspond à la banque HOGENOM à laquelle ont été ajoutées les séquences du génome de la souche ACN (*Frankia alni*). Le but de l'outil développé est de permettre l'ajout de séquences de génomes à une banque de familles de gènes homologues. Pour ajouter ces séquences à une banque telle qu'HOGENOM, il faut les identifier et les classer dans la banque. Ce programme utilise donc les modules développés pour l'application Web HoSeqI afin de permettre l'identification d'un ensemble de n séquences dans une banque de familles de gènes.

1.3.1 Le principe

Le principe est tout d'abord de déterminer la famille de la banque à laquelle chacune des séquences de *Frankia* appartient, puis de calculer pour chaque famille les alignements

multiples en ajoutant les nouvelles séquences et enfin de reconstruire les arbres phylogénétiques correspondants (figure 4.4). C'est à partir de ces arbres qu'une étude des éventuels transferts horizontaux pourra s'effectuer.

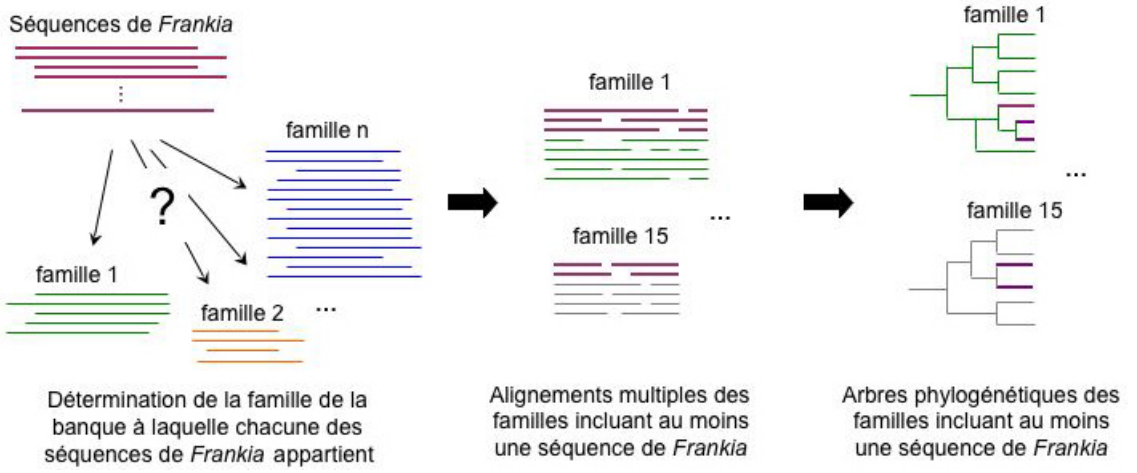

FIG. *4.4: Schéma du principe de l'ajout de séquences de génomes de Frankia à une version locale d'HOGENOM.*

1.3.2 L'algorithme

L'algorithme du programme correspond à l'enchaînement de certains modules d'HoSeqI ainsi que des traitements intermédiaires spécifiques à cette application. Etant donné que ce programme ne doit pas intérargir avec l'utilisateur (puisque les séquences sont toutes identifiées automatiquement les unes à la suite des autres) il n'y a pas de choix de programmes possible et de choix de familles dans le cas où plusieurs familles sont sélectionnées pour une même séquence. Ainsi nous avons défini à l'avance ces choix. Nous avons décidé d'utiliser comme programme d'alignement MUSCLE (MUSCLE-prog ou MUSCLE-fast en fonction du nombre de séquences, avec les paramètres par défaut) et QUICKTREE (paramètres par défaut) pour la reconstruction phylogénétique afin de favoriser la rapidité d'obtention des résultats. De plus, si plusieurs familles sont sélectionnées pour une même séquence (score global proche ou égal ou familles chevauchantes), cette séquence sera alors non traitée et son nom sera inscrit dans un fichier afin de constituer un historique (un fichier de *log* pour les familles de score global proche ou égal et un autre pour les familles non-chevauchantes). La version d'HOGENOM utilisée contenait déjà un génome de *Frankia alni*, ce qui facilite l'identification des familles car les séquences à ajouter sont fortement similaires aux séquences du génome déjà présentes dans la banque. Ainsi le cas de plusieurs familles déterminées pour une séquence ne doit pas être fréquent. De plus, un fichier de *log* est également créé pour les séquences dont le résultat du BLAST est nul, *i.e.* il n'y a aucune séquence suffisamment similaire à la séquence de *Frankia* traitée dans la banque. Pour la même raison que dans le cas de plusieurs familles déterminées, on s'attend à ce que cela soit très peu fréquent. Enfin

un dernier fichier de *log* est utilisé pour les séquences pour lesquelles le programme ne s'est pas exécuté correctement (mémoire insuffisante, un des logiciels tels que BLAST, un programme d'alignement ou de phylogénie ne se termine pas correctement, *etc*) .

Pour chaque séquence de *Frankia* à ajouter, le module de détermination de la famille de gènes est utilisé pour rechercher celle à laquelle appartient la séquence (avec, pour BLASTP, l'utilisation des paramètres par défaut). Puis, on regroupe par famille, dans un même fichier, les séquences de *Frankia*. Pour chaque ensemble de séquences, on utilise alors le module de choix des programmes proposés. Si le nombre de séquences de la banque constituant la famille est supérieur à 5000, alors le programme d'alignement choisi sera MUSCLE-fast, sinon ce sera MUSCLE-prog. Le module d'alignement multiple permet ensuite de re-calculer l'alignement entier de la famille en incluant les séquences de *Frankia*. Enfin, le module de reconstruction phylogénétique est utilisé pour construire les arbres phylogénétiques à partir des alignements obtenus précédemment. Après le traitement, on obtient donc les alignements des familles de la banque contenant au moins une séquence de *Frankia* ainsi que les arbres phylogénétiques correspondant et les différents fichiers de *log*. Un schéma de l'algorithme est présenté dans la figure 4.5.

1.4 Les traitements des résultats

Le programme décrit précédemment nous a permis d'ajouter, à la version d'HOGENOM utilisée, les séquences de la souche CcI3 et de la souche EAN du genre *Frankia*. Ces traitements ont été effectués en une à deux semaines. Sur les 12533 séquences traitées, 2450 (19,5%) séquences n'ont pas été identifiées. Pour la majorité de celles-ci (1821/12533 séquences ou 14,5%), nous n'avons pas obtenu de résultat lors de la recherche de similarité, *i.e.* il n'y avait pas de séquence similaire à ces séquences dans la banque. Elles correspondent à des séquences orphelines que l'on trouve fréquemment et qui ne ressemblent à aucune autre séquence. Enfin, plusieurs familles de score global proche ou égal ont été sélectionnées pour 600 séquences (4,8%) et des familles non-chevauchantes ont été déterminées pour 28 séquences (0,2%).

Un peu plus de 10000 séquences de *Frankia* ont donc été identifiées et les alignements et arbres phylogénétiques de 4435 familles contenant au moins une séquence de *Frankia* ont été reconstruits. Afin de mettre en évidence d'éventuels transferts horizontaux de gènes, les arbres phylogénétiques ont été analysés.

Dans un premier temps, il a été nécessaire de construire un arbre de réconciliation. Pour cela, il faut comparer chaque arbre des gènes résultant avec un arbre des espèces considéré comme référence afin d'obtenir un arbre de réconciliation présentant les événements de spéciation et de duplication entre les gènes (figure 4.6). L'algorithme RAP (Réconciliateur d'Arbres Phylogénétiques) développé par Dufayard et al. (Dufayard *et al.*, 2005) a été utilisé pour construire les arbres réconciliés de chaque arbre phylogénétique.

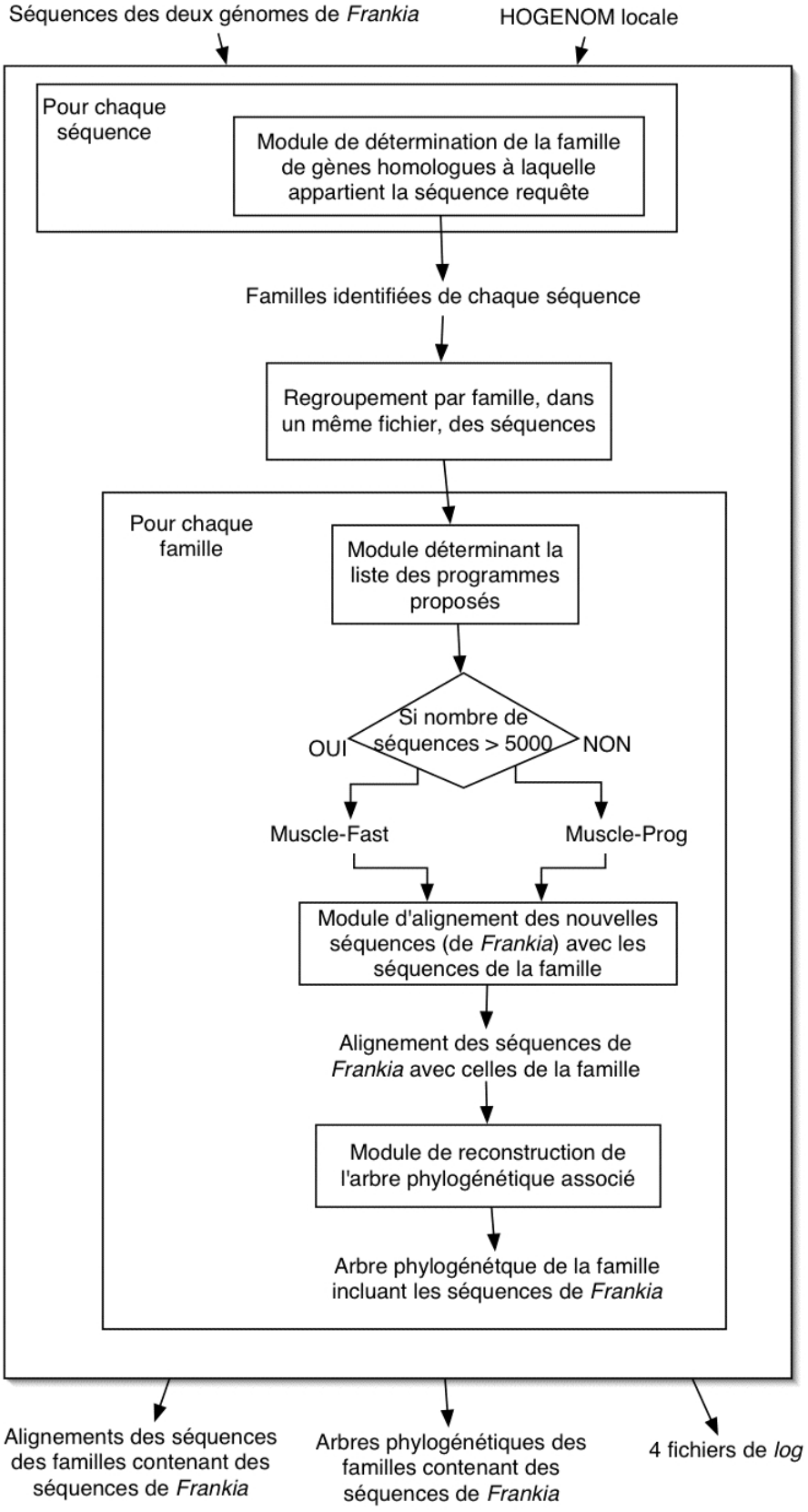

FIG. *4.5: Schéma de l'algortithme du programme d'ajout de séquences de génomes de Frankia à une version locale d'HOGENOM.*

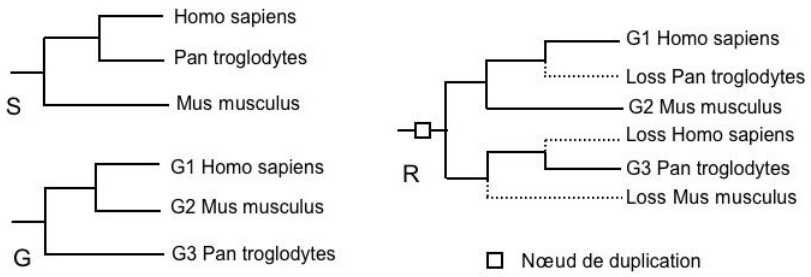

FIG. *4.6: Réconciliation d'arbre entre un arbre des gènes G et un arbre des espèces S montrant les différentes topologies. Le résultat est l'arbre réconcilié R. R est une variation de S, dans lequel des noeuds de duplication suivis de pertes ont été insérés dans le but d'expliquer l'incongruence avec G.*

Une fois que les arbres réconciliés ont été construits, en utilisant un éditeur graphique implémenté dans l'interface FamFetch (Perrière *et al.*, 2000), il a été possible de faire une recherche de motifs d'arbre sur l'ensemble des familles de la banque. Un motif d'arbre correspond à une structure d'arbre particulière avec divers paramètres d'évolution et des paramètres taxonomiques contenus dans les noeuds et les feuilles de l'arbre. La méthode de recherche de motifs est celle développée par Dufayard et al. (Dufayard *et al.*, 2005). On peut ainsi choisir une topologie pour le motif d'arbre et des contraintes sur les noeuds (duplication ou spéciation) et les feuilles (taxons à inclure ou exclure). La méthode utilisée permet de chercher toutes les familles de gènes de la banque pour lesquelles la topologie correspond à un motif particulier.

Afin d'analyser les transferts horizontaux de gènes dans les trois génomes de *Frankia*, une recherche de motifs a été utilisée pour repérer l'ensemble des gènes de *Frankia* dont les plus proches voisins phylogénétiques ne sont pas des Actinobactéries. Cette idée est représentée par le formalisme utilisé dans le motif présenté à la figure 4.7.

La recherche de motifs a permis d'identifier 387 familles correspondant au motif souhaité. Ces familles ont ensuite été utilisées pour des analyses phylogénétiques plus poussées. Pour cela, les alignements de départ ont tout d'abord été filtrés en utilisant le logiciel Gblocks (Castresana, 2000). Ce programme permet de sélectionner des blocs conservés dans les alignements tout en essayant de minimiser la perte de sites informatifs. Le programme élimine les positions mal alignées et les régions divergentes. Ces positions peuvent ne pas être homologues ou peuvent avoir été saturées par des substitutions multiples et il est préférable de les éliminer avant de faire une analyse phylogénétique.

Une fois que les alignements ont été nettoyés, le programme de reconstructions d'arbres phylogénétiques PHYML a été utilisé (modèle JTT avec hétérogénéité du taux d'évolution entre sites modélisée avec une loi gamma, 4 classes de vitesses et tous les paramètres estimés par le programme) avec 100 réplicats de bootstrap.

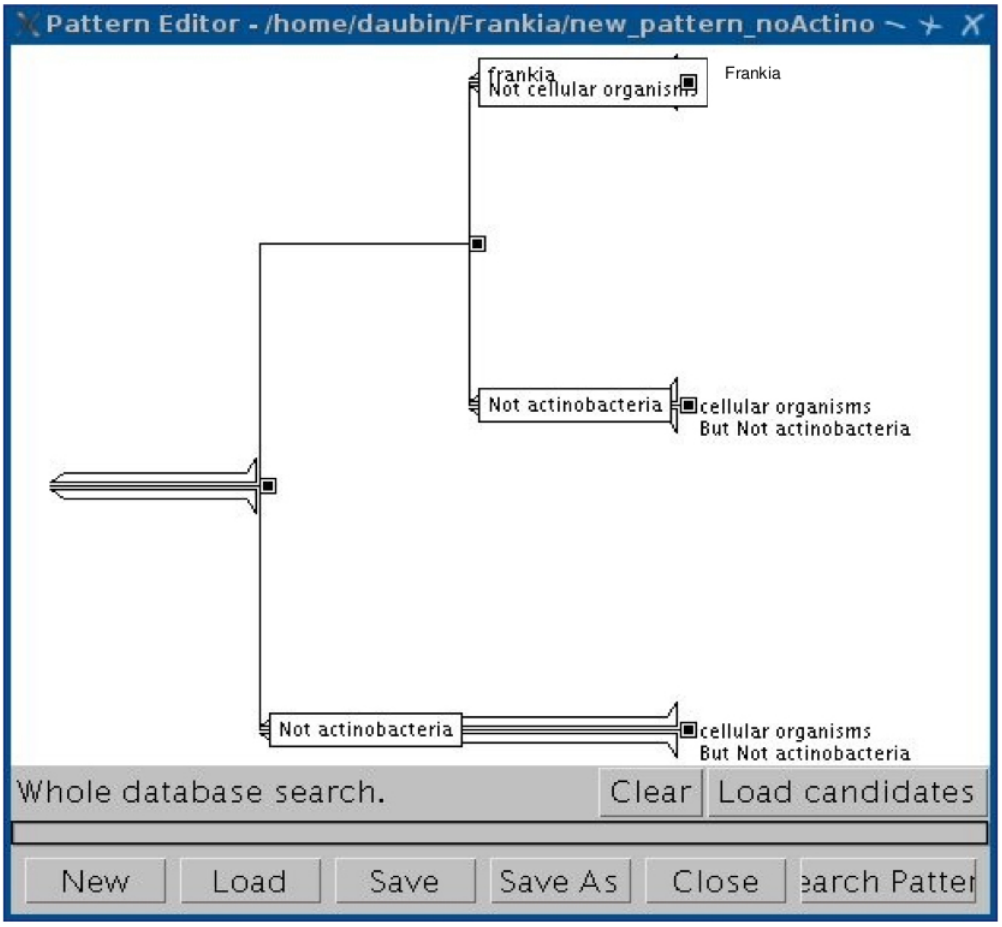

FIG. *4.7: Motif utilisé pour la requête effectuée sur la version locale de la banque HOGENOM*

1.5 Conclusions

Une analyse phylogénétque a été effectuée : tous les arbres obtenus précédemment ont été analysés manuellement et comparés à un arbre de référence tel que celui présenté à la figure 4.1 de la section 1.1, page 78, afin de détecter d'éventuels transferts horizontaux. Un transfert est détecté lorsque des protéines de *Frankia* sont significativement regroupées avec des séquences provenant d'espèces distantes. De plus, les protéines de *Frankia* qui ont des représentants dans une famille où aucune autre Actinobactérie n'est représentée, sont considérées comme probablement transférées. Sur les 387 familles candidates, 201 familles ont été validées comme présentant un transfert horizontal potentiel d'un gène vers *Frankia*. Dans chaque cas, les espèces les plus proches dans l'arbre sont notées comme représentants de la position phylogénétique d'un donneur potentiel. Un tableau en annexe (annexe A, page 117) présente les résultats de cette analyse. Dans ce tableau, on retrouve pour chaque famille, son annotation dans la banque, la valeur de bootstrap validant le transfert horizontal de gène détecté, les séquences de *Frankia* présentes dans la famille et les séquences les plus proches dans l'arbre.

Nous présentons ici trois exemples de transferts horizontaux de gènes vers *Frankia* détectés :

– Le premier exemple est celui d'un gène potentiellement acquis par l'ancêtre actuel des *Frankia* d'une Alpha-Protéobactérie (figure 4.8). Le gène étudié correspond à celui codant pour la Prephenate dehydratase. Il est intéressant de remarquer que toutes les Alpha-Protéobactéries de l'arbre sont également des bactéries fixatrices d'azote.

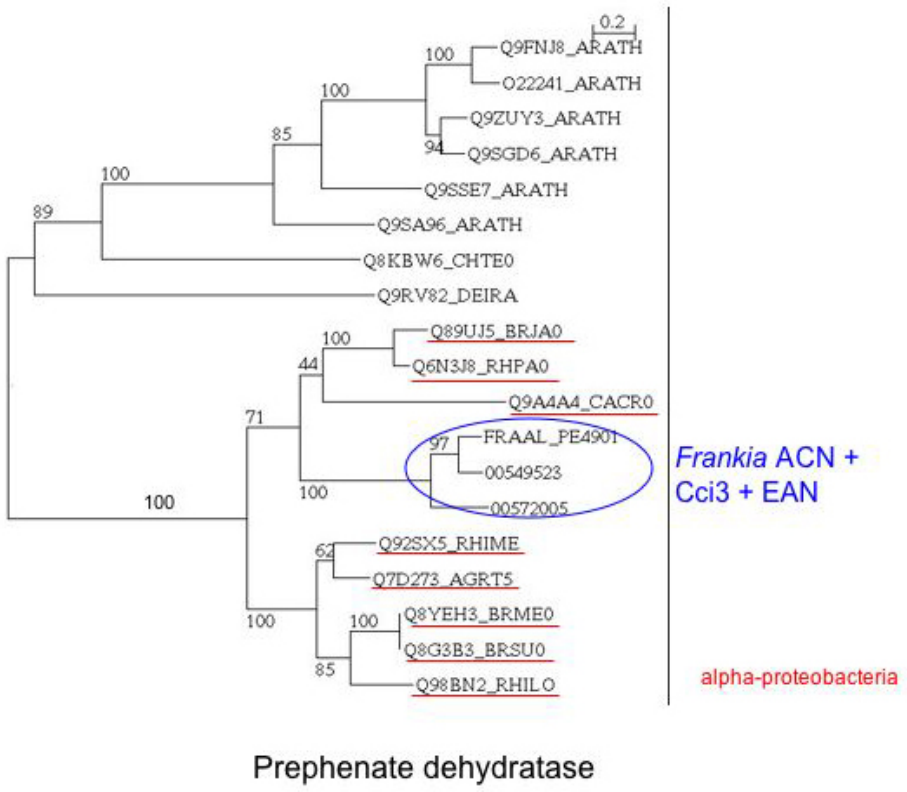

FIG. *4.8: Exemple de transfert horizontal d'un gène d'une Alpha-Protéobactérie vers Frankia ACN, EAN et CcI3.*

– Le deuxième exemple traite un gène potentiellement acquis par *Frankia* ACN d'une Alpha-Protéobactérie (figure 4.9). Le gène analysé est celui codant pour la Glutamine Synthétase III. Ce gène intervient dans la fixation d'azote chez les *Frankia* et les Alpha-Protéobactéries.

– Le troisième exemple est celui de transferts de deux gènes d'une Alpha-Protéobactérie qui se seraient suivis d'une duplication chez *Frankia* EAN (figure 4.10). Les deux gènes correspondent à ceux codant pour les sous-unités I et II de la Sulfate adénylyltransférase qui sont situés à coté sur un chromosome. Ces gènes ont une histoire évolutive très proche comme l'indique la ressemblance des deux arbres phylogénétiques.

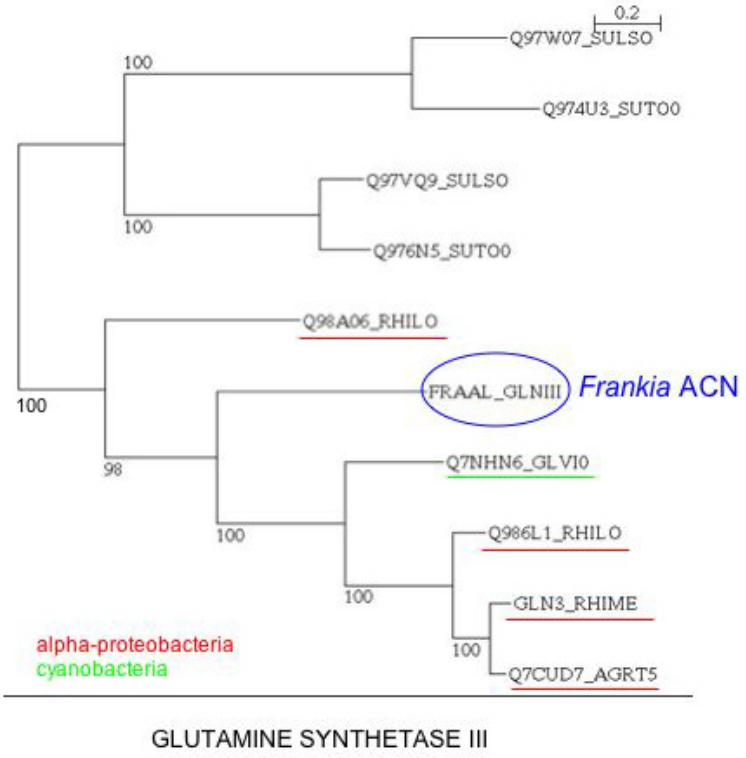

FIG. *4.9: Exemple de transfert horizontal d'un gène d'une Alpha-Protéobactérie vers Frankia ACN.*

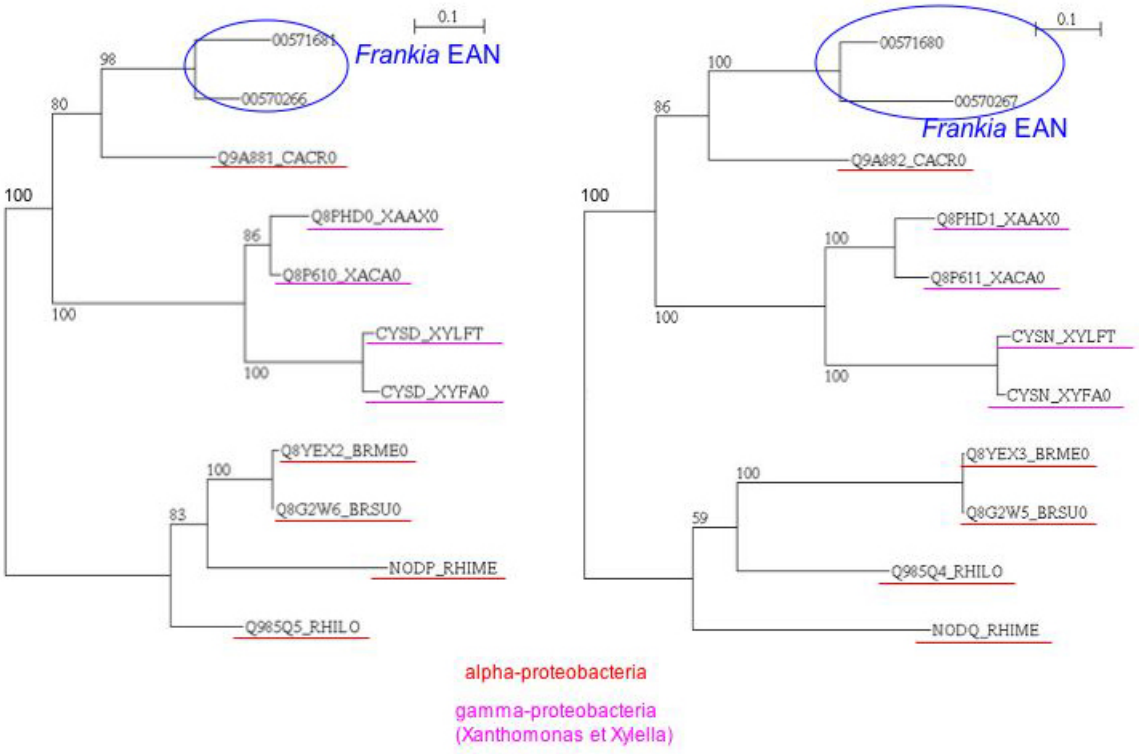

FIG. *4.10: Deux exemples de transferts horizontaux de deux gènes d'une Alpha-Protéobactérie, suivis d'une duplication chez Frankia EAN.*

L'analyse phylogénétique réalisée propose des estimations plutôt conservatrices du nombre de transferts horizontaux de gènes identifiés phylogénétiquement dans *Frankia*, étant donné que seuls des gènes qui ont des homologues dans des génomes d'organismes évolutivement lointains de *Frankia* sont utilisés pour l'analyse. Afin d'identifier d'autres gènes possibles qui pourraient avoir été acquis par *Frankia*, une analyse basée sur le critère d'incongruence phylogénétique (Zelwer & Daubin, 2004) a été utilisée.

Les gènes qui sont présents dans le génome de *Frankia* et absents des génomes de *Streptomyces* ont été recherchés. Pour cela, une analyse de toutes les familles contenant au moins une séquence de *Frankia* a été effectuée afin de ne conserver que les familles ne contenant pas de séquences de *Streptomyces*. 380 familles ont ainsi été détectées.

Streptomyces est le groupe soeur de *Frankia* dans la phylogénie actuelle des organismes complètement séquencés et possède un génome très grand. Bien que l'absence de gènes homologues à *Frankia* dans les génomes de *Streptomyces* ne soit pas une preuve de transferts horizontaux, il est possible que *Streptomyces* ait gardé la plupart des gènes de leur ancêtre commun, et la classe de gènes présents chez *Frankia* et non chez *Streptomyces* est probablement enrichie en transferts horizontaux. Un tableau regroupant l'ensemble des familles correspondant à cette analyse est présenté en annexe (annexe B page 139). Dans ce tableau, on retrouve pour chaque famille, son annotation dans la base de données, les séquences de *Frankia* présentes dans la famille et les séquences les plus proches d'Actinobactérie dans l'arbre.

Pour chacune des analyses effectuées (l'analyse phylogénétique et l'analyse liée à l'absence de gènes chez *Streptomyces*), nous avons compté le nombre de transferts horizontaux de gènes vers l'une des trois souches de *Frankia* correspondant aux catégories COG (Clusters of Orthologous Groups). La banque COG (cf. chapitre 1, page 1 section 2.2, page 13) contient les séquences de génomes complets. Elle regroupe, en familles, des gènes homologues en se basant sur les meilleures similarités BLAST réciproques. Cette classification en catégorie permet d'avoir des informations sur les fonctions des gènes ainsi que des indications sur l'évolution des organismes.

Pour chaque séquence de *Frankia*, présentes dans les familles d'HOGENOM (version locale) où un transfert horizontal a été détecté, nous avons utilisé BLASTP pour la comparer avec les séquences de la banque COG afin d'en extraire la plus similaire. Puis nous avons déterminé la catégorie COG de chacune des séquences similaires et trouvé ainsi celle associée à chaque séquence de *Frankia*. Nous avons ensuite regroupé les séquences de *Frankia* en fonction de la famille à laquelle elles appartenaient, ce qui nous a permis d'associer à chaque famille dans laquelle un transfert horizontal de gènes a été détecté une ou plusieurs catégories COG. Ainsi nous avons obtenu un premier classement donnant le nombre de transferts horizontaux de gènes vers *Frankia* correspondant à chacune des catégories COG. Ces résultats sont résumés dans les tableaux présentés dans la figure 4.11 et la figure 4.12.

COG category	ACN	CcI	EaN	Exemples
[A] RNA processing and modification	0	0	0	
[B] Chromatin structure and dynamics	0	0	0	
[C] Energy production and conversion	8	8	8	Malate/lactate dehydrogenases; Ni,Fe-hydrogenase; Nif
[D] Cell cycle control, cell division, chromosome partitioning	0	0	0	
[E] Amino acid transport and metabolism	10	6	14	Glutamine synthetase; Glutamate synthase
[F] Nucleotide transport and metabolism	3	2	1	Cytosine/adenosine deaminases; Folate-dependent phosphoribosylglycinamide formyltransferase PurN
[G] Carbohydrate transport and metabolism	8	6	10	Sugar kinases, ribokinase family; Predicted xylanase/chitin deacetylase;
[H] Coenzyme transport and metabolism	6	7	5	5,10-methylene-tetrahydrofolate dehydrogenase/Methenyl tetrahydrofolate cyclohydrolase; Geranylgeranyl pyrophosphate synthase;
[I] Lipid transport and metabolism	2	2	2	3-hydroxyisobutyrate dehydrogenase and related beta-hydroxyacid dehydrogenases; Methylmalonyl-CoA mutase, N-terminal domain/subunit;
[J] Translation, ribosomal structure and biogenesis	1	1	1	Tryptophanyl-tRNA synthetase;
[K] Transcription	12	2	12	NAD-dependent protein deacetylases, SIR2 family; NAD-dependent protein deacetylases, SIR2 family;
[L] Replication, recombination and repair	6	10	17	Transposase and inactivated derivatives; RecB family exonuclease;
[M] Cell wall/membrane/envelope biogenesis	2	0	1	Predicted pyridoxal phosphate-dependent enzyme apparently involved in regulation of cell wall biogenesis; Glycosyltransferase;
[N] Cell motility	1	1	1	Predicted periplasmic or secreted lipoprotein;
[O] Posttranslational modification, protein turnover, chaperones	1	1	0	ATPases of the AAA+ class; Organic radical activating enzymes;
[P] Inorganic ion transport and metabolism	12	7	11	Nitrogenase subunit NifH (ATPase); ABC-type Fe3+-hydroxamate transport system, periplasmic component;
[Q] Secondary metabolites biosynthesis, transport and catabolism	8	4	12	Aromatic ring hydroxylase; Isochorismate hydrolase;
[R] General function prediction only	26	18	22	Putative homoserine kinase type II (protein kinase fold); Predicted glycosyltransferases;
[S] Function unknown	9	6	16	Virulence-associated protein and related proteins;
[T] Signal transduction mechanisms	0	0	2	Adenylate cyclase, family 3 (some proteins contain HAMP domain);
[U] Intracellular trafficking, secretion, and vesicular transport	0	0	0	
[V] Defense mechanisms	3	1	2	Beta-lactamase class C and other penicillin binding proteins; Type I restriction-modification system methyltransferase subunit;

FIG. *4.11: Nombre de transferts horizontaux de gènes vers l'une des trois souches de Frankia correspondant aux catégories COG résultant de l'analyse phylogénétique.*

COG category	ACN	CcI	EaN	Exemples
[A] RNA processing and modification	0	0	0	
[B] Chromatin structure and dynamics	0	0	0	
[C] Energy production and conversion	16	2	23	Ni,Fe-hydrogenase
[D] Cell cycle control, cell division, chromosome partitioning	3	3	4	ATPases involved in chromosome partitioning;
[E] Amino acid transport and metabolism	8	8	18	Homoserine acetyltransferase;
[F] Nucleotide transport and metabolism	3	3	4	Predicted alternative thymidylate synthase;
[G] Carbohydrate transport and metabolism	10	8	18	2-polyprenyl-6-methoxyphenol hydroxylase and related FAD-dependent oxidoreductases;
[H] Coenzyme transport and metabolism	5	8	13	2-polyprenyl-6-methoxyphenol hydroxylase and related FAD-dependent oxidoreductases;
[I] Lipid transport and metabolism	8	2	31	Fatty acid desaturase;
[J] Translation, ribosomal structure and biogenesis	1	3	7	Glycyl-tRNA synthetase, alpha subunit; [J] COG0751 Glycyl-tRNA synthetase, beta subunit;
[K] Transcription	9	12	36	AraC-type DNA-binding domain-containing proteins;
[L] Replication, recombination and repair	7	19	31	Transposase and inactivated derivatives, IS30 family;
[M] Cell wall/membrane/envelope biogenesis	9	7	12	Glycosyltransferases involved in cell wall biogenesis;
[N] Cell motility	1	1	1	Flp pilus assembly protein TadC;
[O] Posttranslational modification, protein turnover, chaperones	1	0	2	Predicted metalloendopeptidase;
[P] Inorganic ion transport and metabolism	8	4	15	Siderophore-interacting protein;
[Q] Secondary metabolites biosynthesis, transport and catabolism	6	5	17	SAM-dependent methyltransferases; Phenylpropionate dioxygenase and related ring-hydroxylating dioxygenases, large terminal subunit;
[R] General function prediction only	43	36	67	Predicted aminoglycoside phosphotransferase; Predicted nucleic acid-binding protein, contains PIN domain;
[S] Function unknown	14	14	35	Uncharacterized conserved protein, contains double-stranded beta-helix domain;
[T] Signal transduction mechanisms	2	4	3	Regulator of polyketide synthase expression;
[U] Intracellular trafficking, secretion, and vesicular transport	0	0	0	
[V] Defense mechanisms	2	2	2	Type II restriction enzyme, methylase subunits;

FIG. *4.12: Nombre de transferts horizontaux de gènes vers l'une des trois souches de Frankia correspondant aux catégories COG résultant de l'analyse liée à l'absence de gènes chez Streptomyces.*

2 L'identification automatique de séquences d'ARNr 16S de bactéries

Nous avons établi une autre collaboration avec Philippe Normand (Laboratoire d'Ecologie Microbienne des Sols - UMR CNRS 5557 - Lyon1). Le projet avait pour but d'identifier automatiquement un ensemble de séquences d'ARNr 16S de bactéries issues de carottes glaciaires ou de sources hydrothermales afin de connaître, pour chaque séquence, leur espèce et leur genre. Pour cela, il est nécessaire de développer un outil d'identification automatique qui permette de traiter un ensemble de séquences en déterminant pour chacune l'espèce de provenance et le genre. De plus, pour ce type de données, avant d'identifier les nouvelles séquences, il serait intéressant de vérifier s'il ne s'agit pas de séquences chimères (définies plus bas). Nous avons donc travaillé sur le développement d'un outil d'identification automatique de séquences bactériennes d'ARNr 16S incluant un module de détection de séquences chimères.

2.1 Les séquences d'ARNr 16S de bactéries

Tous les micro-organismes connus possèdent au moins une copie des gènes codant pour les ARNr. Parmi ces gènes, celui codant pour l'ARNr 16S est principalement utilisé pour la détermination du genre et de l'espèce (cf. chapitre 2, page 31 section 2.2.2, page 34). Le séquençage de gènes codant pour l'ARNr 16S est considéré comme une des techniques de référence pour l'identification bactérienne au niveau de l'espèce (Woo *et al.*, 2003). Il permet de traiter des séquences difficilement identifiables avec les techniques classiques (Mignard & Flandrois, 2006).

Les séquences d'ARNr 16S ou ARNr de la petite sous-unité du ribosome sont connues pour un grand nombre de souches bactériennes et sont accessibles par interrogation de banques telles que GenBank, qui contient environ 432212 séquences d'ARNr 16S (version 158, février 2007), ou EMBL qui en contient 432691 (version 89, décembre 2006). Il existe plusieurs banques spécialisées regroupant uniquement des séquences d'ARNr 16S dont la banque américaine RDP et celle européenne, Ribosomal RNA Database (cf. chapitre 2, page 31 section 2.2, page 13). La RDP contient des séquences d'ARNr 16S mithochondiales, de procaryotes et d'eucaryotes. La banque européenne est constituée des séquences complètes ou presque complètes d'ARNr de la petite et de la grande sous-unité pour les archées, les bactéries et les eucaryotes. Elle est divisée en deux parties, Large Subunit rRNA database pour la grande sous-unité et Small Subunit rRNA database pour la petite sous-unité.

Ce type d'identification est différent de l'identification de séquences dans les banques de familles de gènes homologues. Ici, il ne faut pas rechercher la famille de gènes homologues à laquelle une séquence inconnue appartient et la reclasser à l'intérieur de cette famille. Identifier une séquence d'ARNr 16S consiste à replacer la séquence dans la taxo-

nomie des séquences de la banque et à déterminer l'espèce et le genre où rattacher cette séquence. Le processus d'identification sera donc légèrement différent du processus utilisé par les applications décrites précédemment comme HoSeqI ou l'outil d'ajout de séquences de génomes à une banque de familles homologues.

2.2 Les séquences chimères

Si les séquences d'ARNr 16S analysées proviennent d'une amplification par PCR puis d'un clonage, il se peut que certaines soient des séquences chimères c'est-à-dire qu'elles soient formées de différents fragments d'ADN d'origines diverses. Ces séquences sont composées habituellement de deux séquences parentes distinctes phylogénétiquement. Cela se produit lorsqu'un amplicon (séquence spécifique d'ADN à amplifier) terminé prématurément se ré-assemble avec un brin d'ADN étranger puis est copié en entier dans le cycle PCR suivant. Les chimères représentent un problème car elles suggèrent la présence d'organismes non existants, il faut donc pouvoir les détecter. Cependant lorsqu'une séquence est trop courte, on ne pourra pas savoir s'il s'agit d'une chimère ou pas car il ne sera pas possible d'obtenir des résultats significatifs.

Plusieurs outils ont été développés afin de permettre la détection de séquences chimères. Nous présentons ici quelques unes de ces applications. La méthode de Robison-Cox et al. (Robison-Cox *et al.*, 1995), le programme Check_chimera de la RDP (Cole *et al.*, 2005), le package mglobalCHI (Komatsoulis & Waterman, 1997), le logiciel Bellerophon (Huber *et al.*, 2004) et l'application PhyID/CD (cf. chapitre 2, page 31 section 3, page 40) utilisent différentes variantes de la méthode des plus proches voisins qui consiste à déterminer les séquences de la banque les plus proches des différents domaines de la séquence analysée. La plupart de ces méthodes sont basées sur le principe qu'une chimère montrerait des relations phylogénétiques diférentes selon les parties (debut et fin) de la séquence à analyser. Alors que tous les autres programmes comparent les séquences individuellement à une ou deux séquences parents à la fois, Bellerophon détecte toutes les séquences putatives chimères dans un alignement multiple de séquences en une seule analyse.

Un autre programme, CCODE (Chimera and Cross-Over DEtection) (Gonzalez *et al.*, 2005), propose une stratégie pour évaluer les séquences chimères. Cette stratégie est basée uniquement sur la comparaison de la séquence à analyser et de séquences connues, non-chimères.

Les différentes méthodes présentées ici donnent des résultats différents. Elles sont complémentaires et aucune n'est reconnue comme étant la meilleure. Elles permettent d'obtenir des informations pour aider l'utilisateur à savoir si la ou les séquences analysées sont des chimères ou pas.

2.3 Un outil basé sur des modules d'HoSeqI

Différents outils d'identification de séquences d'ARNr 16S existent déjà tels que BIBI, le "RDP classifier", *etc* (cf. chapitre 2, page 31 section 3, page 40). Cependant ces outils ne proposent pas d'étape de détection automatique de chimères avant l'identification ou ne permettent de traiter qu'une seule séquence à la fois. En outre, les outils de détection décrits précédemment nécessitent des traitements manuels afin de déterminer si la séquence traitée est une chimère ou non. Il n'est donc pas possible avec tous ces outils d'identifier un ensemble de séquences en permettant au préalable de détecter automatiquement les chimères.

Nous avons donc développé un outil dont le but est d'automatiser la détection de chimères parmi un ensemble de séquences d'ARNr 16S et d'identifier automatiquement les séquences non-chimères en utilisant une banque de séquences d'ARNr 16S. Pour cela, nous avons réalisé un module de détection de séquences chimères et modifié les modules et méthodes développés dans l'application Web HoSeqI et dans l'outil d'ajout de génomes aux banques de familles homologues.

2.3.1 La base de données utilisée

Pour l'application, nous avons choisi d'utiliser un jeu de données de séquences d'ARNr 16S de bactéries de souches cultivées proposé par Richard Christen et disponible sur le site Web `http://bioinfo.unice.fr/biodiv/Exemple_diversit%E9_16SrRNA/Analyse_de_la_biodiversite.html`. Ce jeu de données contient des séquences exactes, complètes et provenant de souches cultivées, ce qui permet d'avoir des résultats de meilleure qualité lors de recherches de similarité entre la séquence à analyser et les séquences de la banque. Si la séquence traitée est bien une séquence d'ARNr 16S de bactéries, il existera dans le jeu de données une séquence relativement similaire à la séquence requête. De plus, ce jeu de données contient des séquences de Chloroplastes afin de pouvoir les identifier plus facilement.

Nous avons utilisé ce jeu de données pour créer une base de données de séquences d'ARNr 16S au format ACNUC afin de pouvoir utiliser le système d'interrogation Query et ainsi accéder facilement aux différentes informations sur les séquences.

2.3.2 Le principe

Le programme doit permettre l'identification de n nouvelles séquences d'ARNr 16S les unes à la suite des autres tout en ayant vérifié au préalable qu'il ne s'agissait pas de séquences chimères. A la fin des traitements, l'application doit donc indiquer l'espèce et le genre auxquels appartient chaque nouvelle séquence non-chimère.

2.3.2.1 Le repérage de séquences chimères

Une séquence chimère correspond à un fusionnement de plusieurs séquences. Ainsi lorsque l'on compare une chimère avec un ensemble de séquences, on s'attend à retrouver plusieurs *matchs* (*i.e.* séquences de la banque résultant de la recherche de similarité), distants phylogénétiquement, qui obtiennent un bon score d'alignement avec différentes parties de la séquence chimère. La méthode utilisée pour détecter une séquence chimère se base sur le principe des plus proches voisins.

Tout d'abord, la séquence à traiter est découpée en deux demi-séquences d'égale longueur (deux parties 5' et 3'). Puis pour chaque demi-séquence, nous identifions l'ensemble des séquences les plus similaires en la comparant aux séquences de la banque d'ARNr 16S utilisée et en éliminant les redondances entre les deux ensembles. L'alignement de ces deux ensembles de séquences homologues est ensuite calculé et les deux demi-séquences sont ajoutées à cet alignement. Si, dans l'arbre phylogénétique correspondant, les deux demi-séquences sont éloignées *i.e.* si la distance phylogénétique, calculée entre le noeud le plus proche de la première demi-séquence et le noeud le plus proche de la deuxième (distance $d_{1,2}$, en jaune, dans la figure 4.13), est supérieure à un seuil, alors les deux parties de la séquence traitée proviennent d'organismes distants. La séquence traitée est donc considérée comme une chimère. Le principe de la détection de chimère est schématisé dans la figure 4.13.

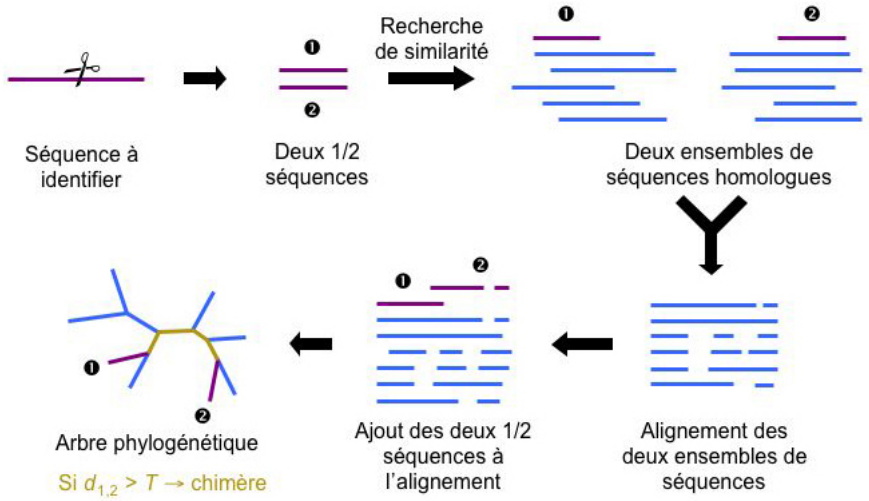

FIG. *4.13: Schéma représentant le principe utilisé pour la détection de séquences chimères.*

2.3.2.2 L'identification

L'identification consiste à trouver l'espèce et le genre auxquels appartient une séquence. Pour cela, il faut identifier les séquences les plus proches qui puissent provenir de cellules

bactériennes appartenant au même taxon que celle étudiée. Tout d'abord, on recherche l'ensemble des séquences les plus similaires à celle identifiée en la comparant aux entrées de la banque utilisée. Puis l'alignement de ces séquences est calculé en incluant celle analysée, et l'arbre phylogénétique correspondant est reconstruit. Enfin on repère dans l'arbre phylogénétique la ou les séquences les plus proches de celle traitée, et on détermine leur espèce. Cette dernière sera celle de provenance de la nouvelle séquence. Le schéma de la figure 4.14 présente ce principe.

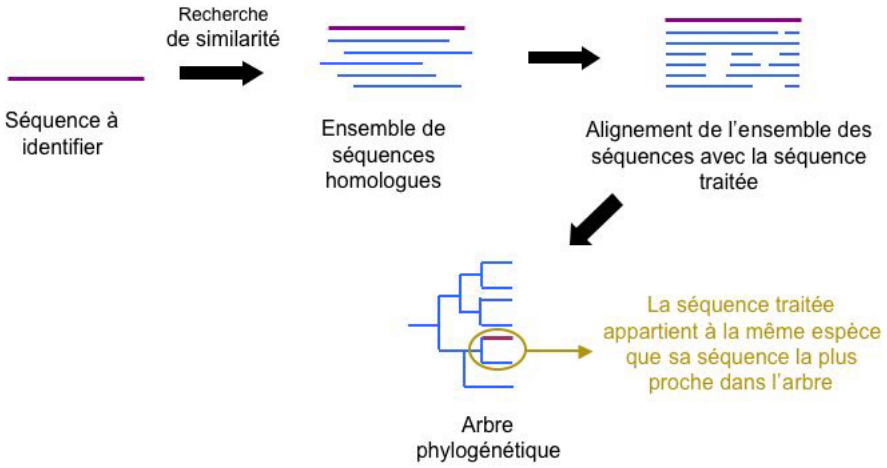

FIG. *4.14: Schéma représentant le principe utilisé pour l'identification de séquences d'ARNr 16S.*

2.3.3 L'algorithme

L'algorithme général de ce programme ressemble à celui utilisé pour l'application d'ajout de séquences de génomes aux banques de familles homologues. En effet, il s'agit également ici d'un enchaînement de modules sans interaction avec l'utilisateur. Les différents programmes utilisés pour les calculs d'alignements et les constructions phylogénétiques devront être choisis au préalable. Deux modules sont utilisés dans ce programme :

- le module de détection de chimères,
- le module d'identification de séquences d'ARNr 16S c'est-à-dire de détermination de l'espèce de la séquence traitée, qui correspond à une modification du premier module principal de l'application Web HoSeqI (*i.e.* le module de détermination de la famille de gènes homologues à laquelle appartient la séquence requête).

Le premier module est exécuté pour chacune des n séquences d'ARNr 16S afin de savoir s'il s'agit d'une séquence chimère. Puis, pour chaque séquence non-chimère, le deuxième module est exécuté, permettant d'obtenir ainsi l'espèce à laquelle appartient la séquence. Enfin en utilisant une fonction ACNUC, il est possible d'accéder à toute la taxonomie de chacune des espèces identifiées afin d'obtenir le genre, le phylum, *etc.* A la fin du traitement, on obtient donc un fichier dans lequel est indiquée, pour chaque séquence non-chimère, la

liste des espèces auxquelles elle appartient potentiellement ainsi que pour chaque espèce sa taxonomie. De la même manière que pour l'application d'ajout de séquences, des fichiers de *log* sont générés afin de connaître les séquences chimères, celles ayant plusieurs espèces potentielles et celles n'ayant pas pu être analysées. L'algorithme général est présenté dans la figure 4.15.

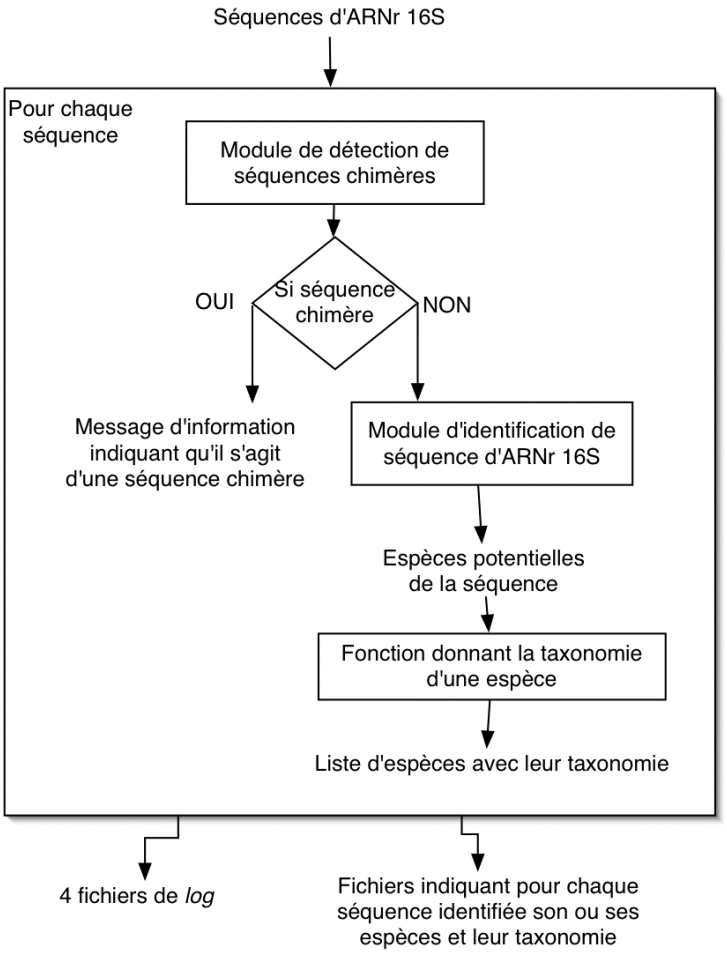

FIG. *4.15: Schéma représentant l'algorithme général de l'application d'identification de séquences d'ARNr 16S.*

2.3.3.1 La détection de chimères

Comme nous l'avons vu dans le principe, plusieurs étapes sont nécessaires à la détection de séquences chimères. Nous pouvons résumer l'algorithme utilisé en trois étapes majeures :

- Découpage de la séquence traitée et recherche de similarité afin de définir deux ensembles de séquences homologues à chacune des demi-séquences.
- Alignement des deux ensembles définis et ajout des deux demi-séquences.
- Calcul de la distance phylogénétique entre les deux demi-séquences.

Ce traitement est effectué pour toutes les séquences dans le cas où elles sont suffisamment longues *i.e.* la longueur de la séquence est supérieure à 1000 paires de base. Dans le cas contraire, la séquence est trop courte pour permettre la détection d'une chimère et est considérée comme non-chimère afin de pouvoir poursuivre l'identification.

Dans la première étape, la comparaison des deux demi-séquences avec les séquences de la banque d'ARNr 16S est réalisée par le programme de recherche de similarité BLASTN utilisé avec les paramètres par défaut (cf. chapitre 1, page 1 section 3.1.3, page 19). Chacun des fichiers obtenus est analysé afin d'extraire deux ensembles de séquences homologues à chacune des demi-séquences traitées. Successivement, une séquence (son nom) de l'un des fichiers puis de l'autre est sélectionnée en éliminant celles qui sont redondantes jusqu'à obtenir deux ensembles d'un total de 30 séquences (ce nombre a été fixé de manière à obtenir un compromis entre le temps d'exécution et la qualité des résultats). Un programme de conversion de la bibliothèque ACNUC permet d'obtenir un fichier contenant les 30 séquences homologues à l'une des deux demi-séquences au format FASTA.

La deuxième étape consiste à calculer l'alignement des deux ensembles de séquences sélectionnées précédemment avec les deux demi-séquences. Pour cela, tout d'abord le programme d'alignement multiple MUSCLE-prog est utilisé, avec les paramètres par défaut, pour aligner les deux ensembles. Puis les demi-séquences sont ajoutées progressivement l'une après l'autre en utilisant l'option d'alignement de profils de MUSCLE-prog.

La troisième étape correspond au calcul de la distance phylogénétique entre les deux demi-séquences. A partir de l'alignement obtenu précédemment, le programme DNADIST (package PHYLIP) permet de calculer la matrice des distances entre chaque séquence. Le modèle évolutif choisi est celui de Kimura à deux paramètres (cf. chapitre 1, page 1 section 3.3.1, page 26) qui permet de calculer la matrice rapidement avec DNADIST (les autres paramètres choisis étant ceux proposés par défaut). L'arbre phylogénétique est ensuite reconstruit à l'aide du programme FASTME (paramètres par défaut). Ainsi, il est facile de retrouver, dans l'arbre, la distance phylogénétique entre les deux demi-séquences : celle-ci correspond à la distance $d_{1,2}$ en jaune sur la figure 4.13. Si cette distance est supérieure à un seuil fixé, cela signifie que les deux demi-séquences ont de grandes chances de provenir d'organismes distants et donc que la séquence traitée correspond certainement au regroupement de différentes séquences éloignées phylogénétiquement. Dans ce cas, la séquence est considérée comme chimère et ne sera pas identifiée.

Dans le cas de séquence non-chimère, il arrive souvent que, dans l'alignement, les deux demi-séquences ne se recouvrent pas (figure 4.16). Le calcul de la matrice pose alors des problèmes car la distance calculée entre les deux demi-séquences n'a aucun sens étant donné qu'elles ne partagent aucune similarité.

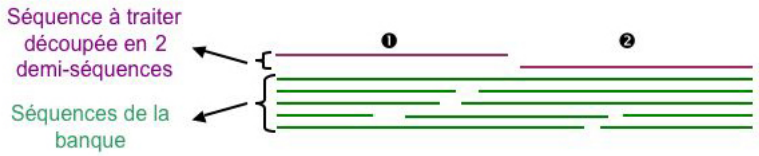

FIG. *4.16: Schéma représentant le cas de deux demi-séquences non recouvrantes dans l'alignement avec les séquences de la banque, homologues à l'une et/ou l'autre des demi-séquences.*

La matrice de distance doit alors être calculée de façons différentes suivant les cas rencontrés :

- A : pour toutes les séquences de la banque, le calcul de la distance se fait sur l'ensemble des sites,
- B : pour les distances entre les séquences de la banque et chacune des demi-séquences, le calcul se fait sur les sites correspondant à chaque demi-séquence,
- C : pour les deux demi-séquences, la distance est calculée entre la première demi-séquence complétée par une partie de sa séquence la plus similaire et la deuxième demi-séquence complétée par une partie de sa séquence la plus similaire.

Les trois cas sont schématisés dans la figure 4.17. Il est ainsi possible d'obtenir les distances entre chacune des séquences de l'alignement, et en particulier la distance entre les deux demi-séquences, afin de déterminer si la séquence d'ARNr 16S traitée est une chimère ou pas.

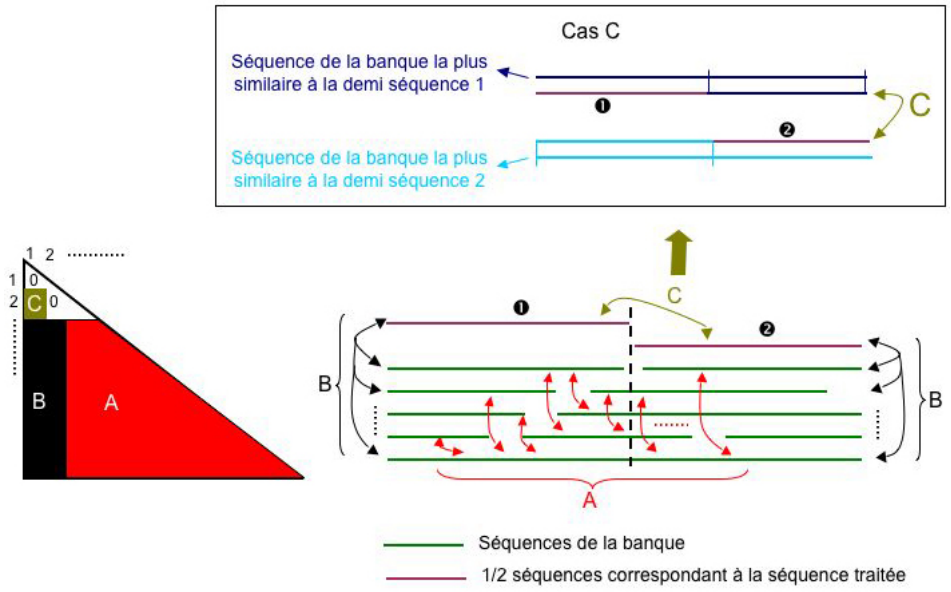

FIG. *4.17: Schéma représentant les différentes étapes du calcul de la matrice de distance correspondant à l'alignement entre les séquences de la banque sélectionnées et les deux demi-séquences.*

Dans le cas où il s'agit d'une chimère, son identification ne s'effectue pas et son nom est noté dans un fichier de *log*. L'algorithme utilisé pour la détection de chimère est représenté à la figure figure 4.18.

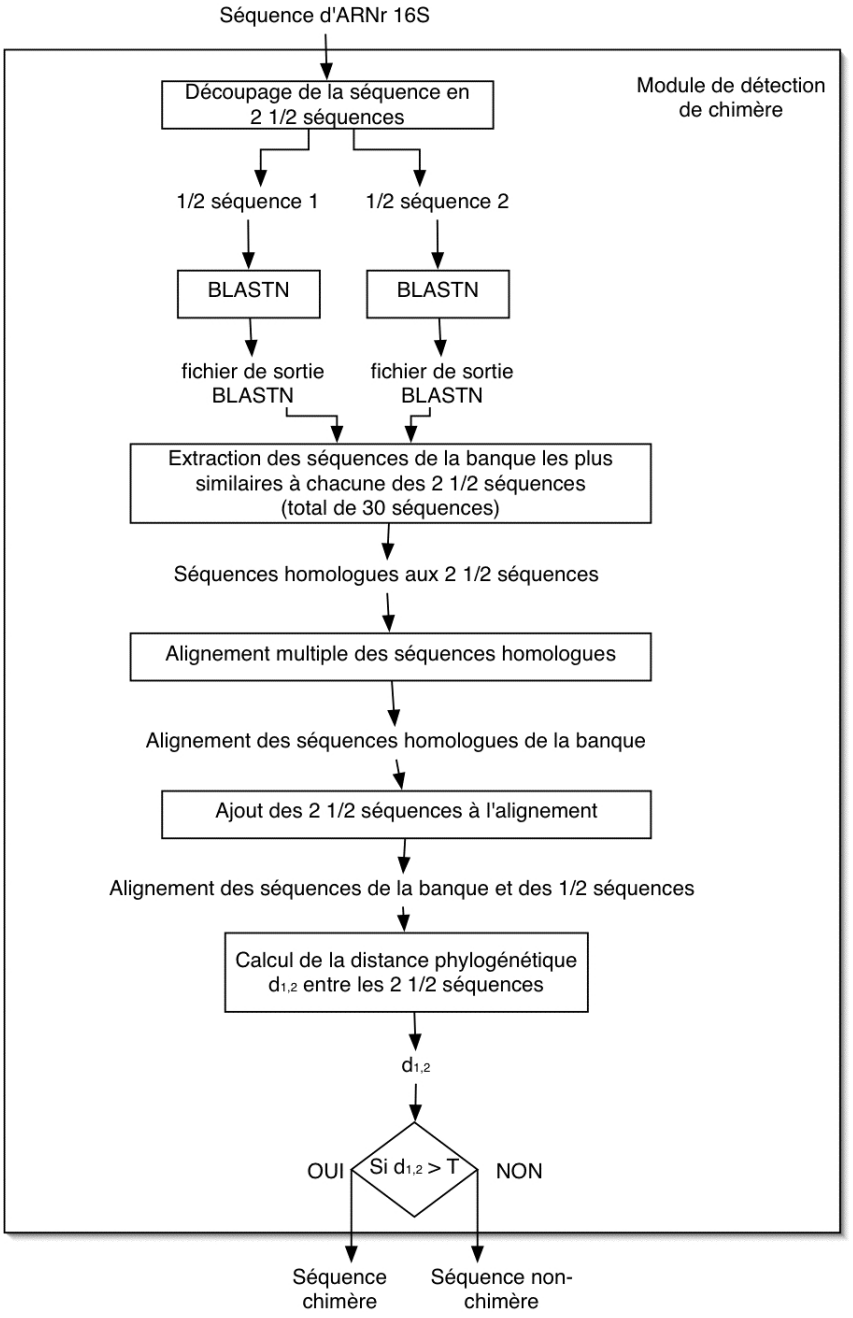

Fig. *4.18: Schéma représentant l'algorithme de détection de séquences chimères.*

2.3.3.2 L'identification de séquences d'ARNr 16S

Le module d'identification d'ARNr 16S a pour but de trouver l'espèce de la séquence non-chimère analysée. Pour ce module, nous avons modifié un module de l'application Web HoSeqI (le module de détermination de la famille de gènes homologues à laquelle appartient la séquence requête, cf. chapitre 3, page 47 section 4.3.1, page 65). Dans l'algorithme de ce dernier, nous avons ajouté un test afin de connaître le type de séquence à traiter. Ainsi, s'il s'agit de séquences d'ARNr 16S, les traitements changent et correspondent aux étapes suivantes :

- Rechercher l'ensemble de séquences les plus similaires à la séquence à identifier.
- Calculer l'alignement de ces séquences en incluant la séquence requête et reconstruire l'arbre phylogénétique correspondant.
- Extraire dans l'arbre phylogénétique la ou les séquences les plus proches de la séquence traitée et déterminer leur espèce.

La première étape est la comparaison de la séquence à identifier avec les séquences de la banque. Cette comparaison se fait en utilisant le programme de recherche de similarité BLASTN (paramètres par défaut). Le fichier de sortie BLASTN est ensuite analysé afin d'extraire les séquences les plus similaires. Les n premières séquences du fichier sont extraites de manière à obtenir au moins 30 séquences avec au moins quatre espèces distinctes représentées. Cela permet d'obtenir suffisamment de séquences d'espèces distinctes pour pouvoir correctement identifier la séquence traitée.

La deuxième étape correspond au calcul de l'alignement des séquences précédemment sélectionnées avec celle à identifier. Cet alignement se fait en utilisant MUSCLE-prog (avec les paramètres par défaut). Puis, à partir de l'alignement, l'arbre phylogénétique est reconstruit à l'aide du programme FASTME (paramètres par défaut) en ayant au préalable calculé la matrice de distance grâce à DNADIST (le modèle évolutif choisi étant celui de Kimura à deux paramètres et les autres paramètres par défaut). Enfin l'arbre obtenu est raciné en son centre par le programme ADD_ROOT.

La dernière étape consiste à déterminer l'espèce de la ou des séquences les plus proches de la séquence traitée dans l'arbre phylogénétique obtenu. Il est donc nécessaire d'analyser l'arbre afin de retrouver les séquences ayant le même ancêtre commun avec la séquence à identifier. Pour cela, nous avons utilisé l'analyseur syntaxique XML LIBXML2 mettant à disposition une bibliothèque de fonctions écrites en C. Le format XML (eXtensible Markup Language) correspond en effet à une représentation arborée. Les fonctionnalités proposées par LIBXML2 permettent d'analyser un document XML et d'extraire facilement des données. Par exemple, il est possible d'obtenir les n frères ou parents d'un noeud.

Le fichier contenant l'arbre phylogénétique au format NEWICK est converti au format XML en utilisant le programme RETREE (package PHYLIP). Une fonction C, faisant appel aux divers outils proposés par la librairie LIBXML2, permet d'extraire les informations de l'arbre phylogénétique. Il est ainsi facile de trouver la ou les séquences les plus proches de celle à identifier dans l'arbre. En utilisant une fonction de la bibliothèque ACNUC, on détermine l'espèce de la ou des séquences extraites de l'arbre. Cette espèce est proposée comme étant celle à laquelle la séquence d'ARNr 16S analysée appartient. Dans le cas où les séquences extraites n'appartiennent pas à la même espèce, toutes les espèces trouvées sont proposées comme espèces potentielles de la séquence d'ARNr 16S analysée. Le schéma de la figure 4.19 représente les différents cas traités. Dans le cas où plusieurs espèces correspondent à une même séquence, les noms de la séquence et des différentes espèces sont notés dans un fichier de *log*.

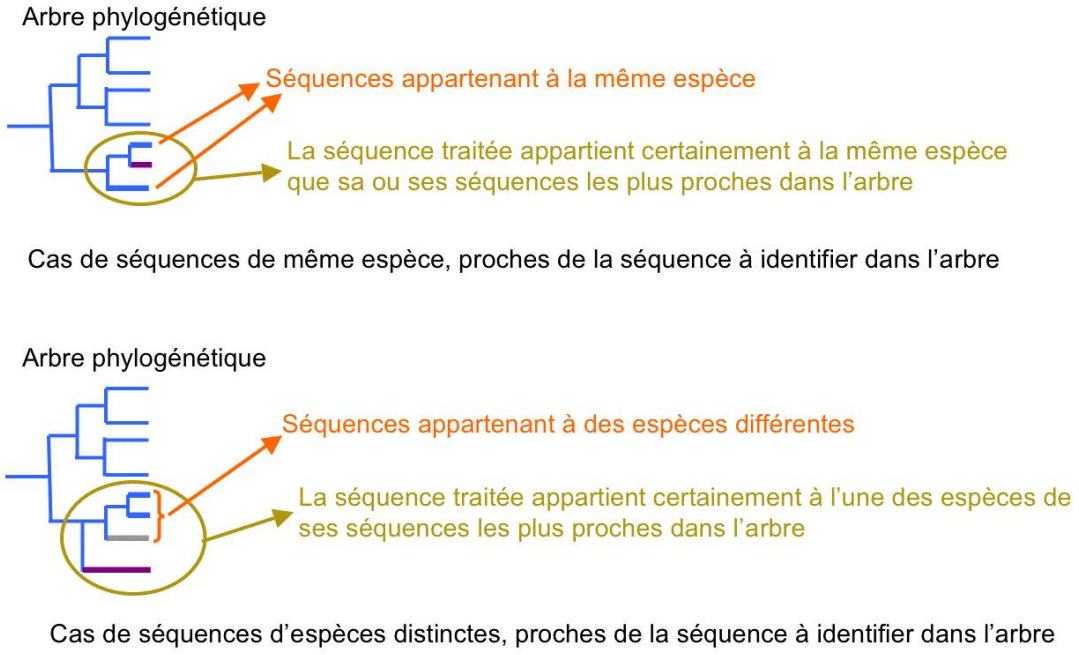

FIG. *4.19: Schéma représentant l'analyse de l'arbre afin de déterminer l'espèce de la séquence à identifier en fonction de sa position dans l'arbre phylogénétique.*

A la fin du traitement, on obtient un fichier qui récapitule la liste des séquences identifiées ainsi que les espèces et leur taxonomie. De plus, d'autres informations sont notées dans deux autres fichiers de *log* : les séquences n'ayant aucune séquence similaire dans la banque et les séquences pour lesquelles il y a eu une erreur d'exécution. L'algorithme utilisé pour l'identification de séquences d'ARNr 16S est présenté à la figure figure 4.20.

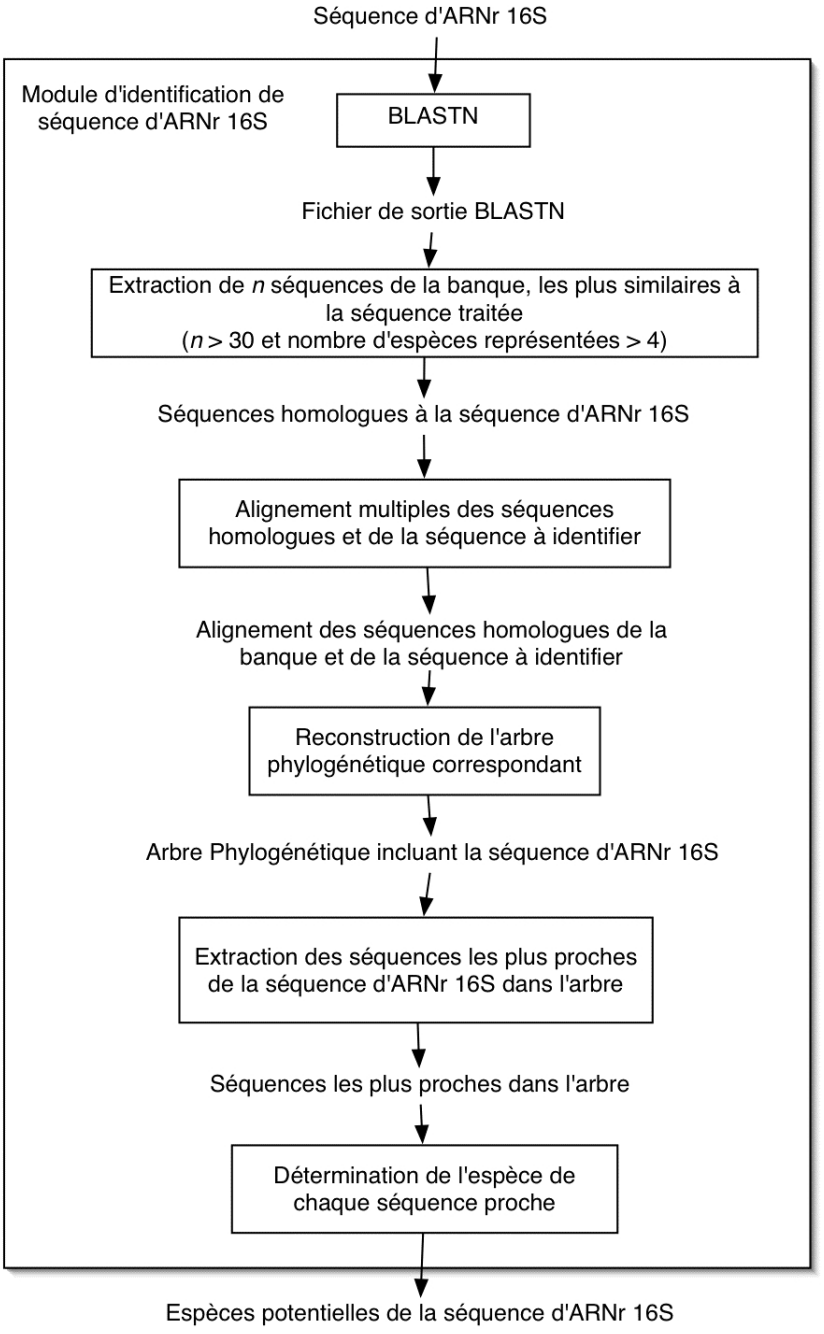

FIG. *4.20: Schéma représentant l'algorithme d'identification de séquences d'ARNr 16S .*

2.4 Conclusions

L'application d'identification automatique de séquences bactériennes d'ARNr 16S est encore en phase de tests afin de vérifier l'exactitude des traitements et de déterminer leur temps d'exécution. Il est difficile avec les séquences d'ARNr 16S d'obtenir des résultats avec 100% d'exactitude, *i.e.* d'être certain que les résultats obtenus pour la détection de chimère et l'identification sont exacts pour toutes les séquences traitées. En effet, ces séquences sont des séquences très bien conservées et il existe peu de dissimilarité entre elles. De plus, les logiciels existants de détections de chimères tels que ceux présentés à la section 2.2, page 93 donnent des résultats différents. En outre, même si un organisme international vérifie et publie les noms valides attribués aux bactéries, il existe encore certains problèmes : de nombreuses espèces ont changé de nom avec les analyses phylogénétiques, les anciens noms sont parfois encore utilisés, la correspondance nom d'espèces et séquence n'est pas toujours exacte, les descriptions des séquences ne comportent pas tout le temps toutes les informations utiles comme le nom de la souche, la taxonomie au dessus du genre reste souvent approximative, *etc.* Tout cela complexifie donc la tâche d'identification et de détection de chimères pour des séquences d'ARNr 16S.

Il faut donc à présent réaliser des tests afin, d'une part, de fixer le seuil pour la distance phylogénétique utilisée pour déterminer si une séquence est chimère ou pas, et, d'autre part, de vérifier la qualité des résultats et déterminer les temps de traitement. Pour cela, nous allons utiliser des séquences connues non présentes dans la banque, chimères et non-chimères. Puis nous allons comparer les résultats de notre application à ceux obtenus avec d'autres applications telles que Check_chimera ou Bellerophon puis le "RDP classifier".

Conclusions et perspectives

L'objectif de cette thèse était de développer des outils bioinformatiques afin de permettre l'identification automatique de nouvelles séquences dans des grandes banques de familles de gènes homologues. L'application Web développée, HoSeqI (Homologous Sequence Identification), permet ce type d'identification. La méthode utilisée se base sur une approche phylogénétique, permettant ainsi l'analyse des relations évolutives entre séquences. HoSeqI propose une interface facile d'utilisation qui permet à un utilisateur d'identifier rapidement une nouvelle séquence dans une banque de familles de gènes homologues choisie, c'est-à-dire de trouver la famille à laquelle appartient la séquence analysée, puis, de visualiser l'alignement et l'arbre phylogénétique obtenus. Il est ainsi facile de localiser la nouvelle séquence dans l'arbre de la famille de gènes homologues afin d'étudier son histoire évolutive. HoSeqI est actuellement disponible sur le site Web du PBIL.

Nous souhaitons faire évoluer cette application afin de proposer un choix plus large de programmes d'alignements multiples adaptés à différents types de données tels que POA, PROBCONS, différentes versions de MAFFT et DIALIGN. De plus, les prochaines versions des banques de gènes homologues proposées au PBIL intégreront les alignements multiples et les arbres phylogénétiques pour les grandes familles (plus de 500 séquences). Cela permettra un gain de temps au niveau du calcul de l'alignement dans HoSeqI puisque plus aucun alignement ne sera recalculé en entier, il suffira d'ajouter la nouvelle séquence aux alignements des séquences des familles pré-existant dans la banque. Il sera alors utile d'avoir un programme d'ajout de séquences à un arbre phylogénétique existant sans devoir reconstruire l'arbre en entier.

A partir des développements effectués pour HoSeqI, nous avons également développé une autre application afin d'ajouter les séquences de nouveaux génomes aux banques de familles de gènes homologues. Cette application nous a permis d'ajouter les séquences protéiques de deux génomes de bactéries du genre *Frankia* (EAN et CcI3) à une version de la banque HOGENOM, contenant déjà un génome de *Frankia* (ACN). Il a ainsi été possible d'étudier l'évolution de ces trois génomes de bactéries et notamment de détecter d'éventuels transferts horizontaux de gènes. Par une étude phylogénétique des arbres obtenus pour les familles de la banque contenant au moins une séquence de *Frankia*, nous avons pu mettre

en évidence l'existence d'un certain nombre de ces transferts potentiels. De plus, une analyse liée à l'absence de gènes chez *Streptomyces* et basée sur le critère d'incongruence phylogénétique nous a permis d'identifier d'autres gènes qui pourraient avoir été acquis par *Frankia*.

Ce logiciel d'ajout de séquences à une banque de familles homologues est pour l'instant utilisé sur le serveur PBIL, ce qui peut entraîner d'importants temps de traitements dans le cas de calculs d'alignements ou de reconstructions phylogénétiques pour de grandes familles de séquences. Il serait donc intéressant de paralléliser les tâches exécutées par l'application afin de pouvoir utiliser les ressources du centre de calcul de l'IN2P3 regroupant plusieurs centaines de machines. Cela permettrait de bénéficier d'une grande rapidité d'exécution mais également d'utiliser des programmes d'alignements ou de phylogénies plus précis qui demandent plus de ressources et qu'il ne serait pas possible de lancer sur le serveur PBIL. Il sera ainsi possible de proposer un choix plus large entre différents programmes d'alignements et de reconstructions phylogénétiques.

Une troisième application a été développée lors de ces travaux de thèse. Celle-ci est dédiée à la détection automatique de séquences chimères et à l'identification de séquences bactériennes d'ARNr 16S. Cet outil permet de déterminer les espèces de provenance d'un ensemble de séquences d'ARNr 16S après avoir vérifié, au préalable, pour chacune d'elles qu'il ne s'agissait pas de séquences chimères.

Des tests sont encore nécessaires pour valider cet outil. De plus, cette application utilisant les mêmes modules que les outils précédents, elle pourra bénéficier de l'utilisation du centre de calcul de l'IN2P3 afin d'augmenter la rapidité d'exécution et permettre un choix plus large dans les programmes d'alignements et de phylogénies. En outre, afin d'augmenter la rapidité des traitements, nous pourrions utiliser une petite banque de représentants de séquences d'ARNr 16S, comme celle proposée par Richard Christen. Celle-ci permettrait d'obtenir rapidement une taxonomie approximative dans laquelle il serait possible de sélectionner un sous-jeu de données que l'on utiliserait ensuite pour l'identification. Par ailleurs, nous souhaitons ajouter un calcul d'indice de diversité génétique entre les séquences identifiées. Enfin, le processus d'identification bactérienne et de détection automatique de chimères pourrait être ajouté à la version Web d'HoSeqI en donnant accès, dans l'interface, à la banque d'ARNr 16S utilisée.

Références bibliographiques

Abdeddaïm S. (1997). Fast and sound two-step algorithms for multiple alignment of nucleic sequences. *Int. J. Artif. Intell. Tools*, **6**, 179–192.

Altschul S. F., Gish W., Miller W., Myers E. W. & Lipman D. J. (1990). Basic local alignment search tool. *J Mol Biol*, **215**(3), 403–10.

Altschul S. F., Madden T. L., Schaffer A. A., Zhang J., Zhang Z., Miller W. & Lipman D. J. (1997). Gapped BLAST and PSI-BLAST : a new generation of protein database search programs. *Nucleic Acids Res*, **25**(17), 3389–402.

Bahr A., Thompson J. D., Thierry J. C. & Poch O. (2001). BAliBASE (Benchmark Alignment dataBASE) : enhancements for repeats, transmembrane sequences and circular permutations. *Nucleic Acids Res*, **29**(1), 323–6.

Benson D. A., Karsch-Mizrachi I., Lipman D. J., Ostell J. & Wheeler D. L. (2006). GenBank. *Nucleic Acids Res*, **34**(Database issue), D16–20.

Berman H. M., Westbrook J., Feng Z., Gilliland G., Bhat T. N., Weissig H., Shindyalov I. N. & Bourne P. E. (2000). The Protein Data Bank. *Nucleic Acids Res*, **28**(1), 235–42.

Bernot A. & Alibert O. La naissance de la biologie moléculaire. Genoscope. `http://www.genoscope.cns.fr/externe/HistoireBM/.`

Birney E., Andrews D., Caccamo M., Chen Y., Clarke L., Coates G., Cox T., Cunningham F., Curwen V., Cutts T., Down T., Durbin R., Fernandez-Suarez X. M., Flicek P., Graf S., Hammond M., Herrero J., Howe K., Iyer V., Jekosch K., Kahari A., Kasprzyk A., Keefe D., Kokocinski F., Kulesha E., London D., Longden I., Melsopp C., Meidl P., Overduin B., Parker A., Proctor G., Prlic A., Rae M., Rios D., Redmond S., Schuster M., Sealy I., Searle S., Severin J., Slater G., Smedley D., Smith J., Stabenau A., Stalker J., Trevanion S., Ureta-Vidal A., Vogel J., White S., Woodwark C. & Hubbard T. J. (2006). Ensembl 2006. *Nucleic Acids Res*, **34**(Database issue), D556–61.

Blaxter M. (2003). Molecular systematics : Counting angels with DNA. *Nature*, **421**(6919), 122–4.

Brown J. R., Douady C. J., Italia M. J., Marshall W. E. & Stanhope M. J. (2001).

Universal trees based on large combined protein sequence data sets. *Nat Genet*, **28**(3), 281–5.

Calteau A. (2005). *Relations évolutives entre les bactéries et les archées hyperthermophiles*. Thèse de doctorat, Université Claude Bernard - Lyon 1.

Carrillo H. & Lipman. D. (1988). The multiple sequence alignment problem in biology. *SIAM Journal Applied Mathematics*, **48**, 1073–1082.

Castresana J. (2000). Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. *Mol Biol Evol*, **17**(4), 540–52.

Charif D. & Lobry J. (2006). SeqinR 1.0-2 : a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis in Structural approaches to sequence evolution : Molecules, networks, populations (U. Bastolla, M. Porto, H.E. Roman and M. Vendruscolo Eds.). *Biological and Medical Physics, Biomedical Engineering, in press*.

Cloud J. L., Conville P. S., Croft A., Harmsen D., Witebsky F. G. & Carroll K. C. (2004). Evaluation of partial 16S ribosomal DNA sequencing for identification of nocardia species by using the MicroSeq 500 system with an expanded database. *J Clin Microbiol*, **42**(2), 578–84.

Cloud J. L., Neal H., Rosenberry R., Turenne C. Y., Jama M., Hillyard D. R. & Carroll K. C. (2002). Identification of Mycobacterium spp. by using a commercial 16S ribosomal DNA sequencing kit and additional sequencing libraries. *J Clin Microbiol*, **40**(2), 400–6.

Cochrane G., Aldebert P., Althorpe N., Andersson M., Baker W., Baldwin A., Bates K., Bhattacharyya S., Browne P., van den Broek A., Castro M., Duggan K., Eberhardt R., Faruque N., Gamble J., Kanz C., Kulikova T., Lee C., Leinonen R., Lin Q., Lombard V., Lopez R., McHale M., McWilliam H., Mukherjee G., Nardone F., Pastor M. P., Sobhany S., Stoehr P., Tzouvara K., Vaughan R., Wu D., Zhu W. & Apweiler R. (2006). EMBL Nucleotide Sequence Database : developments in 2005. *Nucleic Acids Res*, **34**(Database issue), D10–5.

Cole J. R., Chai B., Farris R. J., Wang Q., Kulam S. A., McGarrell D. M., Garrity G. M. & Tiedje J. M. (2005). The Ribosomal Database Project (RDP-II) : sequences and tools for high-throughput rRNA analysis. *Nucleic Acids Res*, **33**(Database issue), D294–6.

Compton J. (1991). Nucleic acid sequence-based amplification. *Nature*, **350**(6313), 91–2.

Corpet F. (1988). Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res*, **16**(22), 10881–90.

Dardel F. K. e. F. (2002). *Bioinformatique : Génomique et post-génomique*. Palaiseau : Ecole Polytechnique.

Daubin V. (2002). *Phylogénie et évolution des génomes procaryotes*. Thèse de doctorat, Université Calude Bernard - Lyon 1.

Daubin V., Moran N. A. & Ochman H. (2003). Phylogenetics and the cohesion of bacterial genomes. *Science*, **301**(5634), 829–32.

Davies J. (1996). Origins and evolution of antibiotic resistance. *Microbiologia*, **12**(1), 9–16.

Dayhoff M., Schwartz R. & Orcutt B. (1978). A model of evolutionary change in proteins. *In Dayhoff, M. O. [ed] Atlas of protein sequece and structure*, **5**(National Biomedical Research Foundation, Washington, DC), 345–352.

Delucinge C. (2003). *MitALib - outil bioinformatique de traçabilité alimentaire*. Rapport interne, Laboratoire de biométrie et Biologie Evolutive - UMR CNRS 5558 - Lyon 1.

Desper R. & Gascuel O. (2002). Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *J Comput Biol*, **9**(5), 687–705.

Devulder G., Perriere G., Baty F. & Flandrois J. P. (2003). BIBI, a bioinformatics bacterial identification tool. *J Clin Microbiol*, **41**(4), 1785–7.

Do C. B., Mahabhashyam M. S., Brudno M. & Batzoglou S. (2005). ProbCons : Probabilistic consistency-based multiple sequence alignment. *Genome Res*, **15**(2), 330–40.

Dufayard J. F. (2004). *Algorithmes incrémentaux pour l'alignement multiple et la phylogénie de grandes familles de séquences homologues*. Thèse de doctorat, Université Joseph Fournier - Grenoble.

Dufayard J. F., Duret L., Penel S., Gouy M., Rechenmann F. & Perriere G. (2005). Tree pattern matching in phylogenetic trees : automatic search for orthologs or paralogs in homologous gene sequence databases. *Bioinformatics*, **21**(11), 2596–603.

Durbin R., Eddy S., Krogh A. & Mitchison G. (1988). *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge UK.

Duret L., Perrière G. & Gouy M. (1999). Hovergen : Database and software for comparative analysis of homologous vertebrate genes. *In Bioinformatics Databases and Systems (ed. S.I. Letovsky)*, p. pp. 13–29.

Eddy S. R. (1998). Profile hidden Markov models. *Bioinformatics*, **14**(9), 755–63.

Edgar R. C. (2004). MUSCLE : multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*, **32**(5), 1792–7.

Ewing B. & Green P. (1998). Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res*, **8**(3), 186–94.

Felsenstein J. (1989). PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics*, **5**, 164–166.

Feng D. F. & Doolittle R. F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J Mol Evol*, **25**(4), 351–60.

Fitch W. (1966). Mutation values for the interconversion of amino acid pair. *J. Mol. Biol.*, **16**, 9–16.

Fitch W. M. (1977). Phylogenies constrained by the crossover process as illustrated by human hemoglobins and a thirteen-cycle, eleven-amino-acid repeat in human apolipoprotein A-I. *Genetics*, **86**(3), 623–44.

Flandrois J. P., S. M., E. D., Gouy M. & Devulder G. (2005). Génération et visualisation de la phylogénie des Bacteria pour l'étude des incohérences taxinomie-phylogénie. In

C. G. G. Perrière, A. Guénoche, Ed., *Actes des Journées Ouvertes Biologie Informatique Mathématiques (JOBIM)*, p. 277–285, Lyon.

Fontana C., Favaro M., Pelliccioni M., Pistoia E. S. & Favalli C. (2005). Use of the MicroSeq 500 16S rRNA gene-based sequencing for identification of bacterial isolates that commercial automated systems failed to identify correctly. *J Clin Microbiol*, **43**(2), 615–9.

Gascuel O. (1997). BIONJ : an improved version of the NJ algorithm based on a simple model of sequence data. *Mol Biol Evol*, **14**(7), 685–95.

Gattiker A., Michoud K., Rivoire C., Auchincloss A. H., Coudert E., Lima T., Kersey P., Pagni M., Sigrist C. J., Lachaize C., Veuthey A. L., Gasteiger E. & Bairoch A. (2003). Automated annotation of microbial proteomes in SWISS-PROT. *Comput Biol Chem*, **27**(1), 49–58.

Gayon J. Histoire de l'hérédité et de la génétique. Cité des Sciences et de l'Industrie. `http://www.cite-sciences.fr/francais/ala_cite/expo/tempo/defis/histoire/index2.htm`.

Geourjon C. & Deleage G. (1995). SOPMA : significant improvements in protein secondary structure prediction by consensus prediction from multiple alignments. *Comput Appl Biosci*, **11**(6), 681–4.

Gonnet G. H., Cohen M. & Benner S. (1992). Exhaustive matching of the entire protein sequence database. *Science*, **256**, 1443–1444.

Gonzalez J. M., Zimmermann J. & Saiz-Jimenez C. (2005). Evaluating putative chimeric sequences from PCR-amplified products. *Bioinformatics*, **21**(3), 333–7.

Gotoh O. (1982). An improved algorithm for matching biological sequences. *J Mol Biol*, **162**(3), 705–8.

Gotoh O. (1993). Optimal alignment between groups of sequences and its application to multiple sequence alignment. *Comput Appl Biosci*, **9**(3), 361–70.

Gouy M., Gautier C., Attimonelli M., Lanave C. & di Paola G. (1985). ACNUC–a portable retrieval system for nucleic acid sequence databases : logical and physical designs and usage. *Comput Appl Biosci*, **1**(3), 167–72.

Grantham R. (1974). Amino acid difference formula to help explain protein evolution. *Science*, **185**(4154), 862–4.

Grasso C. & Lee C. (2004). Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems. *Bioinformatics*, **20**(10), 1546–56.

Graur D. & Li W.-H. (2000). *Fundamentals of molecular evolution*. Sinauer, Sunderland, Massachusetts, 2eme édition.

Gribskov M., McLachlan A. D. & Eisenberg D. (1987). Profile analysis : detection of distantly related proteins. *Proc Natl Acad Sci U S A*, **84**(13), 4355–8.

Guindon S. & Gascuel O. (2003). A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol*, **52**(5), 696–704.

Hall L., Doerr K. A., Wohlfiel S. L. & Roberts G. D. (2003a). Evaluation of the MicroSeq system for identification of mycobacteria by 16S ribosomal DNA sequencing and its integration into a routine clinical mycobacteriology laboratory. *J Clin Microbiol*, **41**(4), 1447–53.

Hall L., Wohlfiel S. & Roberts G. D. (2003b). Experience with the MicroSeq D2 large-subunit ribosomal DNA sequencing kit for identification of commonly encountered, clinically important yeast species. *J Clin Microbiol*, **41**(11), 5099–102.

Harmsen D., Dostal S., Roth A., Niemann S., Rothganger J., Sammeth M., Albert J., Frosch M. & Richter E. (2003). RIDOM : comprehensive and public sequence database for identification of Mycobacterium species. *BMC Infect Dis*, **3**, 26.

Harmsen D., Rothganger J., Frosch M. & Albert J. (2002). RIDOM : Ribosomal Differentiation of Medical Micro-organisms Database. *Nucleic Acids Res*, **30**(1), 416–7.

Harmsen D., Rothganger J., Singer C., Albert J. & Frosch M. (1999). Intuitive hypertext-based molecular identification of micro-organisms. *Lancet*, **353**(9149), 291.

Hebert P. D., Cywinska A., Ball S. L. & deWaard J. R. (2003a). Biological identifications through DNA barcodes. *Proc Biol Sci*, **270**(1512), 313–21.

Hebert P. D., Ratnasingham S. & deWaard J. R. (2003b). Barcoding animal life : cytochrome c oxidase subunit 1 divergences among closely related species. *Proc Biol Sci*, **270 Suppl 1**, S96–9.

Henikoff S. & Henikoff J. (1992). Amino acid substitution matrices from protein blocks. *Proc. Nat. Acad. Sci. USA*, **89**, 10915–10919.

Hilario E. & Gogarten J. P. (1993). Horizontal transfer of ATPase genes–the tree of life becomes a net of life. *Biosystems*, **31**(2-3), 111–9.

Howe K., Bateman A. & Durbin R. (2002). QuickTree : building huge Neighbour-Joining trees of protein sequences. *Bioinformatics*, **18**(11), 1546–7.

Huber T., Faulkner G. & Hugenholtz P. (2004). Bellerophon : a program to detect chimeric sequences in multiple sequence alignments. *Bioinformatics*, **20**(14), 2317–9.

Hulo N., Bairoch A., Bulliard V., Cerutti L., De Castro E., Langendijk-Genevaux P. S., Pagni M. & Sigrist C. J. (2006). The PROSITE database. *Nucleic Acids Res*, **34**(Database issue), D227–30.

Jacob F. & Monod J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, **3**, 318–56.

Jain R., Rivera M. C. & Lake J. A. (1999). Horizontal gene transfer among genomes : the complexity hypothesis. *Proc Natl Acad Sci U S A*, **96**(7), 3801–6.

Johnson M. S. & Overington J. P. (1993). A structural basis for sequence comparisons. An evaluation of scoring methodologies. *J Mol Biol*, **233**(4), 716–38.

Jones D. T., Taylor W. R. & Thornton J. (1992). The rapid generation of mutation data matrices from protein sequences. *Comput. Applic. Biosci.*, **8**, 275–282.

Jukes T. & Cantor C. R. (1969). Evolution of protein molecules. *In H. N. Munro, ed., Mammalian Protein Metabolism*, p. 21–132.

Karlin S. & Altschul S. F. (1993). Applications and statistics for multiple high-scoring segments in molecular sequences. *Proc Natl Acad Sci U S A*, **90**(12), 5873–7.

Katoh K., Kuma K., Toh H. & Miyata T. (2005). MAFFT version 5 : improvement in accuracy of multiple sequence alignment. *Nucleic Acids Res*, **33**(2), 511–8.

Katoh K., Misawa K., Kuma K. & Miyata T. (2002). MAFFT : a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res*, **30**(14), 3059–66.

Kersey P., Bower L., Morris L., Horne A., Petryszak R., Kanz C., Kanapin A., Das U., Michoud K., Phan I., Gattiker A., Kulikova T., Faruque N., Duggan K., McLaren P., Reimholz B., Duret L., Penel S., Reuter I. & Apweiler R. (2005). Integr8 and Genome Reviews : integrated views of complete genomes and proteomes. *Nucleic Acids Res*, **33**(Database issue), D297–302.

Kimura M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J Mol Evol*, **16**(2), 111–20.

Kimura M. (1983). *The Neutral Theory of Molecular Evolution*. Cambridge : Cambridge University Press.

Komatsoulis G. A. & Waterman M. S. (1997). A new computational method for detection of chimeric 16S rRNA artifacts generated by PCR amplification from mixed bacterial populations. *Appl Environ Microbiol*, **63**(6), 2338–46.

Lawrence J. G. & Ochman H. (1998). Molecular archaeology of the Escherichia coli genome. *Proc Natl Acad Sci U S A*, **95**(16), 9413–7.

Lecointre G. & Guyader H. L. (2001). *Classification phylogénétique du vivant*. Paris : Belin.

Lee C., Grasso C. & Sharlow M. F. (2002). Multiple sequence alignment using partial order graphs. *Bioinformatics*, **18**(3), 452–64.

Levin J., Robson B. & Garnier J. (1986). An algorithm for secondary structure determination in proteins based on sequence similarity. *FEBS Lett.*, **205**, 303–308.

Levitt M. (1976). A simplified representation of protein conformations for rapid simulation of protein folding. *J Mol Biol*, **104**(1), 59–107.

Lipman D. J., Altschul S. F. & Kececioglu J. D. (1989). A tool for multiple sequence alignment. *Proc Natl Acad Sci U S A*, **86**(12), 4412–5.

Maftah A. & Julien R. (2003). *Biologie moléculaire, 2eme édition*. Paris : Dunot.

Markmann M. & Tautz D. (2005). Reverse taxonomy : an approach towards determining the diversity of meiobenthic organisms based on Ribosomal RNA signature sequences. *Phil. Trans. R. Soc.*, **360**, 1917–1924.

Medigue C., Rouxel T., Vigier P., Henaut A. & Danchin A. (1991). Evidence for horizontal gene transfer in Escherichia coli speciation. *J Mol Biol*, **222**(4), 851–6.

Meijer A., Roholl P. J., Gielis-Proper S. K., Meulenberg Y. F. & Ossewaarde J. M. (2000). Chlamydia pneumoniae in vitro and in vivo : a critical evaluation of in situ detection methods. *J Clin Pathol*, **53**(12), 904–10.

Mendel G. (1866). Versuche über pflanzen-hybriden. *Verhandlungen des Naturforschenden Vereins*, **4 : 3-47**(Cité page(s) 8.).

Mendel G. (1961). L'oeuvre de Grégor Mendel : Recherches sur divers hybrides végétaux. *Bulletin de "l'Union des Naturalistes de l'Enseignement Public"*, **3 : 1-37**(http ://www.cndp.fr/magsvt/genes/mendel.htm.), Cité page(s) 8.

Mewes H. W., Frishman D., Gruber C., Geier B., Haase D., Kaps A., Lemcke K., Mannhaupt G., Pfeiffer F., Schuller C., Stocker S. & Weil B. (2000). MIPS : a database for genomes and protein sequences. *Nucleic Acids Res*, **28**(1), 37–40.

Mignard S. & Flandrois J. P. (2006). 16S rRNA sequencing in routine bacterial identification : A 30-month experiment. *J Microbiol Methods*.

Morgenstern B. (2004). DIALIGN : multiple DNA and protein sequence alignment at BiBiServ. *Nucleic Acids Res*, **32**(Web Server issue), W33–6.

Mullis K., Faloona F., Scharf S., Saiki R., Horn G. & Erlich H. (1986). Specific enzymatic amplification of DNA in vitro : the polymerase chain reaction. *Cold Spring Harb Symp Quant Biol*, **51 Pt 1**, 263–73.

Needleman S. B. & Wunsch C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, **48**(3), 443–53.

Nirenberg M. W., Matthaei J. H., Jr. O. W. J., Martin R. G. & Barondes S. H. (1963). Approximation of genetic code via cell-free protein synthesis directed by template RNA. *Federation Proceedings*, **22**, 55– 61.

Notredame C., Higgins D. G. & Heringa J. (2000). T-Coffee : A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*, **302**(1), 205–17.

Notredame C., Holm L. & Higgins D. G. (1998). COFFEE : an objective function for multiple sequence alignments. *Bioinformatics*, **14**(5), 407–22.

Ochman H., Lawrence J. G. & Groisman E. A. (2000). Lateral gene transfer and the nature of bacterial innovation. *Nature*, **405**(6784), 299–304.

Okubo K., Sugawara H., Gojobori T. & Tateno Y. (2006). DDBJ in preparation for overview of research activities behind data submissions. *Nucleic Acids Res*, **34**(Database issue), D6–9.

Patel J. B., Leonard D. G., Pan X., Musser J. M., Berman R. E. & Nachamkin I. (2000). Sequence-based identification of Mycobacterium species using the MicroSeq 500 16S rDNA bacterial identification system. *J Clin Microbiol*, **38**(1), 246–51.

Pearson W. R. & Lipman D. J. (1988). Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A*, **85**(8), 2444–8.

Perrière G., Duret L. & Gouy M. (2000). HOBACGEN : database system for comparative genomics in bacteria. *Genome Res*, **10**(3), 379–85.

Perrière G. & Gouy M. (1996). WWW-query : an on-line retrieval system for biological sequence banks. *Biochimie*, **78**(5), 364–9.

Philippe H. & Germot A. (2000). Phylogeny of eukaryotes based on ribosomal RNA : long-branch attraction and models of sequence evolution. *Mol Biol Evol*, **17**(5), 830–4.

Poirot O., O'Toole E. & Notredame C. (2003). Tcoffee@igs : A web server for computing, evaluating and combining multiple sequence alignments. *Nucleic Acids Res*, **31**(13), 3503–6.

Rice P., Longden I. & Bleasby A. (2000). EMBOSS : the European Molecular Biology Open Software Suite. *Trends Genet*, **16**(6), 276–7.

Risler J., Delorme M., Delacroix H. & Henaut A. (1988). Amino acid substitutions in structurally related proteins. A pattern recognition approach. *J. Mol. Biol.*, **204**, 019–1029.

Robison-Cox J. F., Bateson M. M. & Ward D. M. (1995). Evaluation of nearest-neighbor methods for detection of chimeric small-subunit rRNA sequences. *Appl Environ Microbiol*, **61**(4), 1240–5.

Saiki R. K., Gelfand D. H., Stoffel S., Scharf S. J., Higuchi R., Horn G. T., Mullis K. B. & Erlich H. A. (1988). Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase. *Science*, **239**(4839), 487–91.

Saitou N. & Nei M. (1987). The neighbor-joining method : a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, **4**(4), 406–25.

Sanger F., Nicklen S. & Coulson A. (1977). DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci U S A*, **74**, 5463–7.

Schwartz R. M. & Dayhoff M. O. (1978). Matrices for detecting distant relationships. *In Dayhoff, M. O. [ed] Atlas of protein sequece and structure*, **5**(National Biomedical Research Foundation, Washington, DC), 353–358.

Servant F., Bru C., Carrere S., Courcelle E., Gouzy J., Peyruc D. & Kahn D. (2002). ProDom : automated clustering of homologous domains. *Brief Bioinform*, **3**(3), 246–51.

Smith T. F. & Waterman M. S. (1981). Identification of common molecular subsequences. *J Mol Biol*, **147**(1), 195–7.

Steinke D., Vences M., Salzburger W. & Meyer A. (2005). TaxI : a software tool for DNA barcoding using distance methods. *Philos Trans R Soc Lond B Biol Sci*, **360**(1462), 1975–80.

Stender H., Fiandaca M., Hyldig-Nielsen J. J. & Coull J. (2002). PNA for rapid microbiology. *J Microbiol Methods*, **48**(1), 1–17.

Summerbell R. C., Levesque C. A., Seifert K. A., Bovers M., Fell J. W., Diaz M. R., Boekhout T., de Hoog G. S., Stalpers J. & Crous P. W. (2005). Microcoding : the second step in DNA barcoding. *Philos Trans R Soc Lond B Biol Sci*, **360**(1462), 1897–903.

Swofford D. (2003). PAUP*. Phylogenetic Analysis Using Parsimony (*and Other Methods). Version 4.

Tatusov R. L., Natale D. A., Garkavtsev I. V., Tatusova T. A., Shankavaram U. T., Rao B. S., Kiryutin B., Galperin M. Y., Fedorova N. D. & Koonin E. V. (2001).

The COG database : new developments in phylogenetic classification of proteins from complete genomes. *Nucleic Acids Res*, **29**(1), 22–8.

Tautz D., Arctander P., Minelli A., Thomas R. H. & Volger A. P. (2003). A plea for DNA taxonomy. *Trends Ecol. Evol.*, **18**, 70–74.

Thompson J. D., Higgins D. G. & Gibson T. J. (1994). CLUSTAL W : improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, **22**(22), 4673–80.

Thompson J. D., Plewniak F. & Poch O. (1999). BAliBASE : a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, **15**(1), 87–8.

Tracqui P. & Demongeot J. (2003). *Eléments de biologie à l'usage d'autres disciplines. De la structure aux fonctions*. Grenoble : EDP Sciences.

Vences M., Thomas M., van der Meijden A., Chiari Y. & Vieites D. R. (2005). Comparative performance of the 16S rRNA gene in DNA barcoding of amphibians. *Front Zool*, **2**(1), 5.

Wallace I. M., O'Sullivan O., Higgins D. G. & Notredame C. (2006). M-Coffee : combining multiple sequence alignment methods with T-Coffee. *Nucleic Acids Res*, **34**(6), 1692–9.

Wikipédia (2006). Espèce - Wikipédia, l'encyclopédie libre - `http://fr.wikipedia.org/wiki/Esp%C3%A8ce`.

Woo P. C., Ng K. H., Lau S. K., Yip K. T., Fung A. M., Leung K. W., Tam D. M., Que T. L. & Yuen K. Y. (2003). Usefulness of the MicroSeq 500 16S ribosomal DNA-based bacterial identification system for identification of clinically significant bacterial isolates with ambiguous biochemical profiles. *J Clin Microbiol*, **41**(5), 1996–2001.

Wu C. H., Apweiler R., Bairoch A., Natale D. A., Barker W. C., Boeckmann B., Ferro S., Gasteiger E., Huang H., Lopez R., Magrane M., Martin M. J., Mazumder R., O'Donovan C., Redaschi N. & Suzek B. (2006). The Universal Protein Resource (UniProt) : an expanding universe of protein information. *Nucleic Acids Res*, **34**(Database issue), D187–91.

Wu C. H., Yeh L. S., Huang H., Arminski L., Castro-Alvear J., Chen Y., Hu Z., Kourtesis P., Ledley R. S., Suzek B. E., Vinayaka C. R., Zhang J. & Barker W. C. (2003). The Protein Information Resource. *Nucleic Acids Res*, **31**(1), 345–7.

Wuyts J., Perriere G. & Van De Peer Y. (2004). The European ribosomal RNA database. *Nucleic Acids Res*, **32**(Database issue), D101–3.

Ye R. W., Wang T., Bedzyk L. & Croker K. M. (2001). Applications of DNA microarrays in microbial systems. *J Microbiol Methods*, **47**(3), 257–72.

Yona G., Linial N. & Linial M. (2000). ProtoMap : automatic classification of protein sequences and hierarchy of protein families. *Nucleic Acids Res*, **28**(1), 49–55.

Zelwer M. & Daubin V. (2004). Detecting phylogenetic incongruence using BIONJ : an improvement of the ILD test. *Mol Phylogenet Evol*, **33**(3), 687–93.

Zmasek C. M. & Eddy S. R. (2001). ATV : display and manipulation of annotated phylogenetic

trees. *Bioinformatics*, **17**(4), 383–4.

Annexe A
Résultats de la détection d'éventuels transferts horizontaux chez *Frankia*, par l'analyse phylogénétique

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG000186	PROBABLE_HEXULOSE-6-PHOSPHATE_SYNTHASE_	(no_actino_in_family)	:FRAAL_PYRF	:00549430	:00573277	Q74D58_GESU0(_Geobacter_sulfurreducens_PCA) Q72DM8_DESVH(_Desulfovibrio_vulgaris_subsp._vulgaris_str.) PYRF_GLVI0(_Gloeobacter_violaceus_PCC_7421) PYRF_CACR0(_Caulobacter_crescentus_CB15) PYRF_AGRT5(_Agrobacterium_tumefaciens_str._C58) PYRF_RHILO(_Mesorhizobium_loti) PYRF_BRME0(_Brucella_melitensis_16M) PYRF_BRSU0(_Brucella_suis_1330) PYRF_RHIME(_Sinorhizobium_meliloti) PYRF_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q6NCY0_RHPA0(_Rhodopseudomonas_palustris_CGA009) PYRF_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.) PYRF_XACA0(_Xanthomonas_campestris_pv._campestris_str.) PYRF_XYFA0(_Xylella_fastidiosa_9a5c) PYRF_XYLFT(_Xylella_fastidiosa_Temecula1)
HBG000357	GLUTAMINE_SYNTHETASE_1;_PROBABLE_GLUTAMINE_SYNTHETASE_2_Q7CUD7_AGRT5_GLN3_RHIME_100_Q986L1_RHILO_100_Q7NHN6_GLVI0_100	98	:FRAAL_GLNIII			Q7CUD7_AGRT5(_Agrobacterium_tumefaciens_str._C58),GLN3_RHIME(_Sinorhizobium_meliloti),Q986L1_RHILO(_Mesorhizobium_loti),Q7NHN6_GLVI0(_Gloeobacter_violaceus_PCC_7421)
HBG000481	L-LACTATE_DEHYDROGENASE_1;_L-LACTATE_DEHYDROGENASE_2;_L-LACTATE_DEHYDROGENASE_3;_L-LACTATE_DEHYDROGE_	(no_actino_in_family)			:00568757	Q6KLP9_BAAN0(_Bacillus_anthracis_str._'Ames_Ancestor') Q81KZ7_BACAA(_Bacillus_anthracis_str._Ames) Q72ZE5_BACE0(_Bacillus_cereus_ATCC_10987) Q817F9_BACCR(_Bacillus_cereus_ATCC_14579) MDH_BASU0(_Bacillus_subtilis_subsp._subtilis_str.) Q8EPE2_OCEIH(_Oceanobacillus_iheyensis) MDH_BAHA0(_Bacillus_halodurans_C-125) Q8CQ25_STEP0(_Staphylococcus_epidermidis_ATCC_12228) Q8A0W0_BATH0(_Bacteroides_thetaiotaomicron_VPI-5482) MDH_CHTE0(_Chlorobium_tepidum_TLS) Q8PVJ7_MEMA0(_Methanosarcina_mazei_Go1) Q8TSH7_MEAC0(_Methanosarcina_acetivorans_C2A) Q7NHJ3_GLVI0(_Gloeobacter_violaceus_PCC_7421) Q55383_SYNY3(_Synechocystis_sp._PCC_6803) Q8YP78_ANASP(_Nostoc_sp._PCC_7120) Q7UNC6_RHOBA(_Pirellula_sp.) Q6L0C3_PITO0(_Picrophilus_torridus_DSM_9790) MDH_THEAC(_Thermoplasma_acidophilum) Q979N9_THEVO(_Thermoplasma_volcanium)
HBG000504	PROBABLE_THREONINE_ALDOLASE_	(no_actino_in_family)	:FRAAL_LTAE	:00546307	:00569141	Q7NHQ7_GLVI0(_Gloeobacter_violaceus_PCC_7421) Q9RUR9_DEIRA(_Deinococcus_radiodurans) Q72KW7_THTH0(_Thermus_thermophilus_HB27) Q89N26_BRJA0(_Bradyrhizobium_japonicum_USDA_110)
HBG000583	NITROGENASE_IRON-MOLYBDENUM_COFACTOR_BIOSYNTHESIS_PROTEIN_NIFE_	100	:FRAAL_NIFE	:00546166	:00571386	Q6NA13_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q7MRF9_WOLSU(_Wolinella_succinogenes) NIFE_RHILO(_Mesorhizobium_loti) NIFE_RHIME(_Sinorhizobium_meliloti) NIFE_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q6N0Z2_RHPA0(_Rhodopseudomonas_palustris_CGA009) NIFE_ANASP(_Nostoc_sp._PCC_7120)
HBG000583	NITROGENASE_IRON-MOLYBDENUM_COFACTOR_BIOSYNTHESIS_PROTEIN_NIFE	100	:FRAAL_NIFD	:00547747	:00571857	Q7MRF8_WOLSU(_Wolinella_succinogenes) Q92ZL4_RHIME(_Sinorhizobium_meliloti) Q98AP6_RHILO(_Mesorhizobium_loti) NIFD_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q6N0Z0_RHPA0(_Rhodopseudomonas_palustris_CGA009) NIFD_ANASP(_Nostoc_sp._PCC_7120) Q749C3_GESU0(_Geobacter_sulfurreducens_PCA)
HBG000584	NITROGENASE_IRON_PROTEIN_1;_NITROGENASE_IRON_PROTEIN_2	93	:FRAAL_NIFH	:00547746	:00571856	NIH1_ANASP(_Nostoc_sp._PCC_7120) NIH2_ANASP(_Nostoc_sp._PCC_7120)
HBG001883	3'-PHOSPHOADENOSINE_5'-PHOSPHOSULFATE_SULFOTRANSFERASE;_AGR_C_1496P;_NODULATION_PROTEIN_NODP;_SULFAT	80			:00570266 :00571681	Q9A881_CACR0(_Caulobacter_crescentus_CB15)
HBG002269	ARGININE_BIOSYNTHESIS_BIFUNCTIONAL_PROTEIN_ARG7_MITOCHONDRIAL_PRECURSOR_[INCLUDES:_GLUTAMATE_N-ACET	99			:00568065	ARGJ_BRJA0(_Bradyrhizobium_japonicum_USDA_110) ARGJ_RHPA0(_Rhodopseudomonas_palustris_CGA009) ARGJ_AGRT5(_Agrobacterium_tumefaciens_str._C58) ARGJ_RHIME(_Sinorhizobium_meliloti) ARGJ_RHILO(_Mesorhizobium_loti) ARGJ_BRME0(_Brucella_melitensis_16M) ARGJ_BRSU0(_Brucella_suis_1330) ARGJ_CACR0(_Caulobacter_crescentus_CB15)
HBG002300	FEMO_COFACTOR_BIOSYNTHESIS_PROTEIN_NIFB;_HYPOTHETICAL_PROTEIN_MJ1093	98	:FRAAL_NIFB	:00546159	:00571393	NIFB_ANASP(_Nostoc_sp._PCC_7120) NIFB_RHIME(_Sinorhizobium_meliloti) Q98AT9_RHILO(_Mesorhizobium_loti) NIFB_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q6N0X9_RHPA0(_Rhodopseudomonas_palustris_CGA009)
HBG003007	BIFUNCTIONAL_SHORT_CHAIN_ISOPRENYL_DIPHOSPHATE_SYNTHASE_[INCLUDES:_FARNESYL_PYROPHOSPHATE_SYNTHETASE	60		:00549359		Q7U881_SYNPX(_Synechococcus_sp._WH_8102) Q7V6P0_PROMM(_Prochlorococcus_marinus_str._MIT_9313) Q7VBG5_PRMA0(_Prochlorococcus_marinus_subsp._marinus_str.) Q7V119_PROMP(_Prochlorococcus_marinus_subsp._pastoris_str.)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG003011	HYPOTHETICAL_PROTEIN_BUSG401;_HYPOTHETICAL_PROTEIN_HI1189;_HYPOTHETICAL_PROTEIN_MJ1645;_HYPOTHETICAL	(no_actino_in_family)		:00548689		Q9A3P3_CACR0(_Caulobacter_crescentus_CB15) Q7NSB7_CHVI0(_Chromobacterium_violaceum_ATCC_12472) Q8XZF2_RASO0(_Ralstonia_solanacearum_GMI1000) Q6NAJ5_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q89SC0_BRJA0(_Bradyrhizobium_japonicum_USDA_110)
HBG003442	GLUTAMATE_SYNTHASE_LARGE_SUBUNIT-LIKE_PROTEIN;_HYPOTHETICAL_PROTEIN_MJ1351	100	:FRAAL_PE1890			Q882P1_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q986L0_RHILO(_Mesorhizobium_loti) GLXD_RHIME(_Sinorhizobium_meliloti) Q7CUD6_AGRT5(_Agrobacterium_tumefaciens_str._C58)
HBG003806	FRUCTOSE-BISPHOSPHATE_ALDOLASE_CLASS_I;_PROBABLE_FRUCTOSE-BISPHOSPHATE_ALDOLASE_CLASS_I;_PUTATIVE_AL_	(no_actino_in_family)		:00548928 :00549318		Q6LZE3_METMP(_Methanococcus_maripaludis) Y400_MEJA0(_Methanocaldococcus_jannaschii_DSM_2661) Q8TVI0_MEKA0(_Methanopyrus_kandleri_AV19) Y579_METTH(_Methanothermobacter_thermautotrophicus_str._Delta_H) Y108_ARFU0(_Archaeoglobus_fulgidus_DSM_4304) Q8PXE9_MEMA0(_Methanosarcina_mazei_Go1) Q8THC6_MEAC0(_Methanosarcina_acetivorans_C2A) Q72EV8_DESVH(_Desulfovibrio_vulgaris_subsp._vulgaris_str.) YF54_AQAE0(_Aquifex_aeolicus_VF5) Q8PWG2_MEMA0(_Methanosarcina_mazei_Go1) Q8TTJ2_MEAC0(_Methanosarcina_acetivorans_C2A) Y309_HALN1(_Halobacterium_sp._NRC-1) Y230_ARFU0(_Archaeoglobus_fulgidus_DSM_4304)
HBG004209	PROBABLE_DIHYDROOROTATE_DEHYDROGENASE_ELECTRON_TRANSFER_SUBUNIT_	(no_actino_in_family)		:00547738		Q8KB97_CHTE0(_Chlorobium_tepidum_TLS) Q7LWY8_PYFU0(_Pyrococcus_furiosus_DSM_3638) Q9V045_PYRAB(_Pyrococcus_abyssi) O59014_PYHO0(_Pyrococcus_horikoshii_OT3) Q8U2E4_PYFU0(_Pyrococcus_furiosus_DSM_3638) Q9V0C3_PYRAB(_Pyrococcus_abyssi) Q729F2_DESVH(_Desulfovibrio_vulgaris_subsp._vulgaris_str.) Q74H08_GESU0(_Geobacter_sulfurreducens_PCA) Q8KD07_CHTE0(_Chlorobium_tepidum_TLS) Q97CU7_THEVO(_Thermoplasma_volcanium) Q9HM24_THEAC(_Thermoplasma_acidophilum)
HBG004390	102AA_LONG_HYPOTHETICAL_PERIPLASMIC_DIVALENT_CATION_TOLERANCE_PROTEIN_CUTA;_106AA_LONG_HYPOTHETICAL_	(no_actino_in_family)	:FRAAL_PE2149	:00545800	:00572633	Q7VTA5_BORPE(_Bordetella_pertussis) Q7W2C8_BORPA(_Bordetella_parapertussis) Q7WR96_BORBR(_Bordetella_bronchiseptica) Q82SF1_NIEU0(_Nitrosomonas_europaea_ATCC_19718) Q8XV40_RASO0(_Ralstonia_solanacearum_GMI1000) Q7NQ89_CHVI0(_Chromobacterium_violaceum_ATCC_12472) Q8PPZ7_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.) Q87BC1_XYLFT(_Xylella_fastidiosa_Temecula1) Q9PFN8_XYFA0(_Xylella_fastidiosa_9a5c) Q8PD25_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8YL42_ANASP(_Nostoc_sp._PCC_7120) Q8EIY0_SHON0(_Shewanella_oneidensis_MR-1) Q72IZ0_THTH0(_Thermus_thermophilus_HB27) Q9RS33_DEIRA(_Deinococcus_radiodurans) Q7NDP4_GLVI0(_Gloeobacter_violaceus_PCC_7421) Q8DL76_SYNEL(_Synechococcus_elongatus) Q6NBW9_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q8KC19_CHTE0(_Chlorobium_tepidum_TLS) Q7UKK3_RHOBA(_Pirellula_sp.) Q7VGV2_HEHE0(_Helicobacter_hepaticus_ATCC_51449) Q8SVR6_ENCCU(_Encephalitozoon_cuniculi) Q7MRU0_WOLSU(_Wolinella_succinogenes) Q92J45_RICO0(_Rickettsia_conorii_str._Malish_7) Q9ZDY6_RIPR0(_Rickettsia_prowazekii_str._Madrid_E)
HBG005220	HYDROGENASE-1_SMALL_CHAIN_PRECURSOR;_HYDROGENASE-2_SMALL_CHAIN_PRECURSOR;_PERIPLASMIC_[NIFE]_HYDROGE	100			:00571319	Q74GX1_GESU0(_Geobacter_sulfurreducens_PCA)
HBG005731	6-PYRUVOYL_TETRAHYDROBIOPTERIN_SYNTHASE_PRECURSOR_	(no_actino_in_family)		:00548688		PTPS_ESCO1(_Escherichia_coli_O157:H7_EDL933) Q8FEI6_ECOL6(_Escherichia_coli_O6) PTPS_SHFL1(_Shigella_flexneri_2a_str._2457T) PTPS_ECO57(_Escherichia_coli_O157:H7) PTPS_ESCO0(_Escherichia_coli_K12) PTPS_SHFL0(_Shigella_flexneri_2a_str._301) Q7AMD4_SALTI(_Salmonella_typhi) Q7CPW5_SATY0(_Salmonella_typhimurium_LT2) Q8XGC8_SAEN0(_Salmonella_enterica_subsp._enterica_serovar) Q7N8L7_PHLU0(_Photorhabdus_luminescens_subsp._laumondii_TTO1) Q74XS6_YEPE2(_Yersinia_pestis_biovar_Medievalis_str.) Q8D199_YEPE0(_Yersinia_pestis_KIM) Q8ZBN5_YEPE1(_Yersinia_pestis_CO92) PTPS_BUAP1(_Buchnera_aphidicola_str._Sg_(Schizaphis) Q9KSF7_VICH0(_Vibrio_cholerae_O1_biovar_eltor) Q87NI5_VIBPA(_Vibrio_parahaemolyticus) Q8DAE8_VIVU0(_Vibrio_vulnificus_CMCP6) Q7MJS2_VIBVY(_Vibrio_vulnificus_YJ016) Q88KE9_PSEPK(_Pseudomonas_putida_KT2440) Q87ZJ4_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q9I0H2_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q728E6_DESVH(_Desulfovibrio_vulgaris_subsp._vulgaris_str.) Q87EY7_XYLFT(_Xylella_fastidiosa_Temecula1) Q9PGV5_XYFA0(_Xylella_fastidiosa_9a5c) Q8PCW2_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PGK2_XAAX0(_Xanthomonas_axonopodis_pv._citri_st
HBG005850	HIGH-AFFINITY_BRANCHED-CHAIN_AMINO_ACID_TRANSPORT_SYSTEM_PERMEASE_PROTEIN_BRAD;_HIGH-AFFINITY_BRANCH	93			:00570312	Q6N3H3_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q89CH2_BRJA0(_Bradyrhizobium_japonicum_USDA_110)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from Alni	sequences from Cci3	sequences from EAN	sister group in tree
HBG006099	PROTEIN_GLCG_	100			:00570582	Q88I09_PSEPK(_Pseudomonas_putida_KT2440) Q89HE9_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q98FH1_RHILO(_Mesorhizobium_loti) Q8FYT5_BRSU0(_Brucella_suis_1330) Q8YJ13_BRME0(_Brucella_melitensis_16M) Q981X1_RHILO(_Mesorhizobium_loti) Q6LPW3_PHOPR(_Photobacterium_profundum) Q9I3K9_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q88F04_PSEPK(_Pseudomonas_putida_KT2440)
HBG006387	NADPH_DEHYDROGENASE_2;_NADPH_DEHYDROGENASE_3;_PROBABLE_NADH-DEPENDENT_FLAVIN_OXIDOREDUCTASE_YQIG;_PR	100	:FRAAL_PE185			Q6N0R8_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q7WCF4_BORPA(_Bordetella_parapertussis) Q7WQF9_BORBR(_Bordetella_bronchiseptica)
HBG007761	EXU_REGULON_TRANSCRIPTIONAL_REGULATOR;_GLC_OPERON_TRANSCRIPTIONAL_ACTIVATOR;_HYPOTHETICAL_TRANSCRIPT	89	:FRAAL_PE6171			Q7WEI8_BORBR(_Bordetella_bronchiseptica) Q7W370_BORPA(_Bordetella_parapertussis) Q7VSP7_BORPE(_Bordetella_pertussis) Q7WFB1_BORBR(_Bordetella_bronchiseptica) Q7W3Y1_BORPA(_Bordetella_parapertussis) Q7VUU7_BORPE(_Bordetella_pertussis)
HBG010213	XANTHINE_DEHYDROGENASE _FAD_BINDING_SUBUNIT_	88			:00569261 :00569738	Q7WCD7_BORPA(_Bordetella_parapertussis) Q7WQE3_BORBR(_Bordetella_bronchiseptica)
HBG010886	HYPOTHETICAL_25.3_KDA_PROTEIN_IN_RPL28-SEH1_INTERGENIC_REGION;_HYPOTHETICAL_27.6_KDA_PROTEIN_IN_THI2	(no_actino_in_family)	:FRAAL_PE4334			Q8DAD4_VIVU0(_Vibrio_vulnificus_CMCP6) Q7MJT7_VIBVY(_Vibrio_vulnificus_YJ016) Q87NK1_VIBPA(_Vibrio_parahaemolyticus) Q9RWG4_DEIRA(_Deinococcus_radiodurans) Q7NV46_CHVI0(_Chromobacterium_violaceum_ATCC_12472) Q7N1B9_PHLU0(_Photorhabdus_luminescens_subsp._laumondii_TTO1)
HBG011935	HYPOTHETICAL_OXIDOREDUCTASE_HI1010;_HYPOTHETICAL_OXIDOREDUCTASE_SLR0229;_HYPOTHETICAL_OXIDOREDUCTASE	100			:00571212	Q7UTL3_RHOBA(_Pirellula_sp.)
HBG011935	HYPOTHETICAL_OXIDOREDUCTASE_HI1010;_HYPOTHETICAL_OXIDOREDUCTASE_SLR0229;_HYPOTHETICAL_OXIDOREDUCTASE	100	:FRAAL_PE1509	:00549467		2)Q8PDM7_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PQK1_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.)
HBG012112	GLUCOSE-6-PHOSPHATE_1-DEHYDROGENASE_1_CHLOROPLAST_PRECURSOR;_GLUCOSE-6-PHOSPHATE_1-DEHYDROGENASE_2	98			:00568967 :00572395	G6PD_RHIME(_Sinorhizobium_meliloti) Q7D148_AGRT5(_Agrobacterium_tumefaciens_str._C58) Q8FVP8_BRSU0(_Brucella_suis_1330) Q8YCL5_BRME0(_Brucella_melitensis_16M) Q989A2_RHILO(_Mesorhizobium_loti)
HBG012169	PUTATIVE_ARSENICAL_PUMP-DRIVING_ATPASE_1;_PUTATIVE_ARSENICAL_PUMP-DRIVING_ATPASE_2_	(no_actino_in_family)	:FRAAL_PE5094	:00547849	:00569564	ARS1_AQAE0(_Aquifex_aeolicus_VF5) Q7NKT7_GLVI0(_Gloeobacter_violaceus_PCC_7421) Q8KAT5_CHTE0(_Chlorobium_tepidum_TLS) Q8DHV1_SYNEL(_Synechococcus_elongatus) ARSA_SYNY3(_Synechocystis_sp._PCC_6803) Q8YTL7_ANASP(_Nostoc_sp._PCC_7120) ARSA_CHTE0(_Chlorobium_tepidum_TLS) Q8KDR5_CHTE0(_Chlorobium_tepidum_TLS) Q8KG52_CHTE0(_Chlorobium_tepidum_TLS) Q8RIN4_FUNU0(_Fusobacterium_nucleatum_subsp._nucleatum_ATCC) Q8RIN3_FUNU0(_Fusobacterium_nucleatum_subsp._nucleatum_ATCC) Q6KZR9_PITO0(_Picrophilus_torridus_DSM_9790) Q979S7_THEVO(_Thermoplasma_volcanium) Q9HL03_THEAC(_Thermoplasma_acidophilum) Q73EL2_BACE0(_Bacillus_cereus_ATCC_10987)
HBG012217	PHOSPHORIBOSYLGLYCINAMIDE_FORMYLTRANSFERASE_CHLOROPLAST_PRECURSOR	84		:00547921		Q9RSU6_DEIRA(_Deinococcus_radiodurans)
HBG014219	PROBABLE_PYRUVATE-FLAVODOXIN_OXIDOREDUCTASE_	(no_actino_in_family)	:FRAAL_PE3253	:00549155		Q8YVR3_ANASP(_Nostoc_sp._PCC_7120) NIFJ_SYNY3(_Synechocystis_sp._PCC_6803) NIFJ_ANASP(_Nostoc_sp._PCC_7120) Q8KC02_CHTE0(_Chlorobium_tepidum_TLS) Q74GZ6_GESU0(_Geobacter_sulfurreducens_PCA) O83909_TRPA0(_Treponema_pallidum_subsp._pallidum_str.) Q73PY0_TRDE0(_Treponema_denticola_ATCC_35405) Q6LQN5_PHOPR(_Photobacterium_profundum)
HBG016186	CYSN/CYSC_BIFUNCTIONAL_ENZYME_[INCLUDES:_SULFATE_ADENYLYLTRANSFERASE_SUBUNIT_1;_ELONGATION_FACTOR_1-	86			:00570267 :00571680	Q9A882_CACR0(_Caulobacter_crescentus_CB15)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG016812	ADENOSYLHOMOCYSTEINASE;_PUTATIVE_ADENOSYLHOMOCYSTEINASE_2;_PUTATIVE_ADENOSYLHOMOCYSTEINASE_3	100	:FRAAL_AHCY			SAHH_GLVI0(_Gloeobacter_violaceus_PCC_7421) SAHH_ANASP(_Nostoc_sp._PCC_7120) SAHH_SYNY3(_Synechocystis_sp._PCC_6803) SAHH_SYNEL(_Synechococcus_elongatus)
HBG016897	PROBABLE_ISOCHORISMATASE;_VIBRIOBACTIN-SPECIFIC_ISOCHORISMATASE	96			:00568222	Q7DC80_PSAE0(_Pseudomonas_aeruginosa_PAO1)
HBG017189	HYPOTHETICAL_SUGAR_KINASE_PH1459;_HYPOTHETICAL_SUGAR_KINASE_YDJE;_PROTEIN_IOLC	80	:FRAAL_PE770 :FRAAL_PE3962			Q89RE0_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q7NI68_GLVI0(_Gloeobacter_violaceus_PCC_7421)
HBG017216	HOMOSERINE/HOMOSERINE_LACTONE_EFFLUX_PROTEIN;_HYPOTHETICAL_PROTEIN_HI1307;_HYPOTHETICAL_PROTEIN_PA47	50	:FRAAL_PE4701			Q81AX1_BACCR(_Bacillus_cereus_ATCC_14579)
HBG017750	HYPOTHETICAL_ABC_TRANSPORTER_PERMEASE_PROTEIN_YDEY;_HYPOTHETICAL_ABC_TRANSPORTER_PERMEASE_PROTEIN_YD	100	:FRAAL_RBSC			Q9CLI1_PASMU(_Pasteurella_multocida)
HBG019061	BIFUNCTIONAL_METHYLENETETRAHYDROFOLATE_DEHYDROGENASE/CYCLOHYDROLASE_MITOCHONDRIAL_PRECURSOR_[INCLUD	75	:FRAAL_FOLD :FRAAL_PE4216	:00547408 :00548957	:00573052	Q987T6_RHILO(_Mesorhizobium_loti) O83714_TRPA0(_Treponema_pallidum_subsp._pallidum_str.)
HBG023009	HYPOTHETICAL_PROTEIN_YGAY_	78	:FRAAL_PE2900			YGAY_ECO57(_Escherichia_coli_O157:H7) YGAY_ESCO1(_Escherichia_coli_O157:H7_EDL933) Q8FEQ7_ECOL6(_Escherichia_coli_O6) YGAY_ESCO0(_Escherichia_coli_K12) Q7C0B8_SHFL1(_Shigella_flexneri_2a_str._2457T) Q83QG6_SHFL0(_Shigella_flexneri_2a_str._301) Q7C7S3_SAEN0(_Salmonella_enterica_subsp._enterica_serovar) Q8Z4E1_SALTI(_Salmonella_typhi) Q8ZML0_SATY0(_Salmonella_typhimurium_LT2) Q7CK77_YEPE0(_Yersinia_pestis_KIM) Q8ZBX1_YEPE1(_Yersinia_pestis_CO92) Q74WZ8_YEPE2(_Yersinia_pestis_biovar_Medievalis_str.) Q7N778_PHLU0(_Photorhabdus_luminescens_subsp._laumondii_TTO1) Q88LS8_PSEPK(_Pseudomonas_putida_KT2440) Q7VWY6_BORPE(_Bordetella_pertussis) Q7WH34_BORBR(_Bordetella_bronchiseptica) Q7W9M3_BORPA(_Bordetella_parapertussis) Q8PD50_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PQ23_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.)
HBG023997	DNA-INVERTASE_FROM_LAMBDOID_PROPHAGE_E14;_DNA-INVERTASE_HIN;_PUTATIVE_DNA-INVERTASE_FROM_LAMBDOID_PR	84		:00548548 :00547238	:00570322	Q6MB56_PASP0(_Parachlamydia_sp._UWE25) Q82WD8_NIEU0(_Nitrosomonas_europaea_ATCC_19718) Q7AQU9_SALTI(_Salmonella_typhi) Q82W59_NIEU0(_Nitrosomonas_europaea_ATCC_19718) Q9RZH9_DEIRA(_Deinococcus_radiodurans)
HBG024002	AMINODEOXYCHORISMATE_LYASE;_BRANCHED-CHAIN_AMINO_ACID_AMINOTRANSFERASE_1_MITOCHONDRIAL_PRECURSOR;_B	91	:FRAAL_ILVE	:00548779	:00569641	Q9HNF8_HALN1(_Halobacterium_sp._NRC-1) ILVE_METTH(_Methanothermobacter_thermautotrophicus_str._Delta_H) Q8KC21_CHTE0(_Chlorobium_tepidum_TLS)
HBG024148	P-PROTEIN_[INCLUDES:_CHORISMATE_MUTASE;_PROBABLE_PREPHENATE_DEHYDRATASE	71	:FRAAL_PE4901	:00549523	:00572005	Q9A4A4_CACR0(_Caulobacter_crescentus_CB15) Q6N3J8_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q89UJ5_BRJA0(_Bradyrhizobium_japonicum_USDA_110)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG031619	POTENTIAL_TYROSINE_RECOMBINASE_XERD-LIKE;_PROBABLE_INTEGRASE/RECOMBINASE_PROTEIN_MJ0367;_TYROSINE_RE	83			:00569776 :00567027	Q981F2_RHILO(_Mesorhizobium_loti) Q989L3_RHILO(_Mesorhizobium_loti)
HBG036201	NITROGENASE_IRON-MOLYBDENUM_COFACTOR_BIOSYNTHESIS_PROTEIN_NIFN	100	:FRAAL_NIFK	:00547748	:00571858	Q749C4_GESU0(_Geobacter_sulfurreducens_PCA) Q7M8U9_WOLSU(_Wolinella_succinogenes) Q92ZL3_RHIME(_Sinorhizobium_meliloti) Q98AP5_RHILO(_Mesorhizobium_loti) NIFK_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q6N0Z1_RHPA0(_Rhodopseudomonas_palustris_CGA009) NIFK_ANASP(_Nostoc_sp._PCC_7120)
HBG042664	HYPOTHETICAL_PROTEIN_MJ1031;_HYPOTHETICAL_PROTEIN_MTH1382;_PUTATIVE_MALONATE_TRANSPORTER_	83	:FRAAL_PE1984	:00545918	:00569709	Q8KF63_CHTE0(_Chlorobium_tepidum_TLS)
HBG046136	TRYPTOPHANYL-TRNA_SYNTHETASE_1;_TRYPTOPHANYL-TRNA_SYNTHETASE_2;_TRYPTOPHANYL-TRNA_SYNTHETASE_MITOCH	67	:FRAAL_TRPS	:00546660	:00570492	Q81B50_BACCR(_Bacillus_cereus_ATCC_14579) Q6KJ43_BAAN0(_Bacillus_anthracis_str._'Ames_Ancestor') Q81N11_BACAA(_Bacillus_anthracis_str._Ames) Q734M8_BACE0(_Bacillus_cereus_ATCC_10987)
HBG088695	PUTATIVE_ADENYLATE_CYCLASE_3_	(no_actino_in_family)			:00569760	Q98LH2_RHILO(_Mesorhizobium_loti) Q89CQ9_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q92LP1_RHIME(_Sinorhizobium_meliloti) CYA3_RHIME(_Sinorhizobium_meliloti) Q98GL1_RHILO(_Mesorhizobium_loti) Q98LH3_RHILO(_Mesorhizobium_loti)
HBG103603	PUTATIVE_SELENIUM-BINDING_PROTEIN;_SELENIUM-BINDING_PROTEIN_1;_SELENIUM-BINDING_PROTEIN_2	95			:00568286	Q89C46_BRJA0(_Bradyrhizobium_japonicum_USDA_110)
HBG113612	HYPOTHETICAL_PROTEIN_YEAK_	(no_actino_in_family)		:00549645		YEAK_ESCO0(_Escherichia_coli_K12) YEAK_ECO57(_Escherichia_coli_O157:H7) YEAK_ESCO1(_Escherichia_coli_O157:H7_EDL933) Q8FGW3_ECOL6(_Escherichia_coli_O6) Q7C1Q1_SHFL1(_Shigella_flexneri_2a_str._2457T) Q83L68_SHFL0(_Shigella_flexneri_2a_str._301) Q7CAA3_SAEN0(_Salmonella_enterica_subsp._enterica_serovar) Q8Z6E6_SALTI(_Salmonella_typhi) Q8ZPW5_SATY0(_Salmonella_typhimurium_LT2) Q7N6I8_PHLU0(_Photorhabdus_luminescens_subsp._laumondii_TTO1) Q7NYV7_CHVI0(_Chromobacterium_violaceum_ATCC_12472)
HBG116595	PUTATIVE_DIOXYGENASE_SSO1533;_PUTATIVE_DIOXYGENASE_VC1345_	(no_actino_in_family)			:00571205	Q6KY68_BAAN0(_Bacillus_anthracis_str._'Ames_Ancestor') Q81VG4_BACAA(_Bacillus_anthracis_str._Ames) Q73EU6_BACE0(_Bacillus_cereus_ATCC_10987) Q81IU4_BACCR(_Bacillus_cereus_ATCC_14579) Q8EFK8_SHON0(_Shewanella_oneidensis_MR-1) Q6KZ96_PITO0(_Picrophilus_torridus_DSM_9790) YF33_SULSO(_Sulfolobus_solfataricus) Q6LI43_PHOPR(_Photobacterium_profundum) Q7NZF2_CHVI0(_Chromobacterium_violaceum_ATCC_12472) YD45_VICH0(_Vibrio_cholerae_O1_biovar_eltor) Q87Q00_VIBPA(_Vibrio_parahaemolyticus) Q7MLD1_VIBVY(_Vibrio_vulnificus_YJ016) Q8D940_VIVU0(_Vibrio_vulnificus_CMCP6)
HBG218422	COBALAMIN_BIOSYNTHESIS_PROTEIN_CBID;_PUTATIVE_COBALT-PRECORRIN-6A_SYNTHASE_[DEACETYLATING]	100			:00572346	CBID_AGRT5(_Agrobacterium_tumefaciens_str._C58) CBID_BRME0(_Brucella_melitensis_16M) CBID_BRSU0(_Brucella_suis_1330)
HBG218803	2 3 4 5-TETRAHYDROPYRIDINE-2 6-DICARBOXYLATE_N-SUCCINYLTRANSFERASE	97	:FRAAL_DAPD	:00546562	:00571119	Q8KAA7_CHTE0(_Chlorobium_tepidum_TLS)
HBG223564	HYPOTHETICAL_PROTEIN_MJ0964	89			:00573177	Q88H50_PSEPK(_Pseudomonas_putida_KT2440)
HBG223564	HYPOTHETICAL_PROTEIN_MJ0964	87			:00574156 :00574157	Q7NIT4_GLVI0(_Gloeobacter_violaceus_PCC_7421)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG225003	CONSERVED_PROTEIN_ASSOCIATED_WITH_ACETYL-COA_C-ACYLTRANSFERASE;_HYPOTHETICAL_PROTEIN_AF0132;_HYPOTHE	74	:FRAAL_PE2508			O30036_ARFU0(_Archaeoglobus_fulgidus_DSM_4304)
HBG227697	6-PHOSPHOFRUCTOKINASE;_PHOSPHOFRUCTOKINASE;_PUTATIVE_PYROPHOSPHATE--FRUCTOSE-6-PHOSPHATE_1-PHOSPHOTR_	(no_actino_in_family)	:FRAAL_PE4376	:00547658	:00572343	O51669_BOBU0(_Borrelia_burgdorferi_B31) O83146_TRPA0(_Treponema_pallidum_subsp._pallidum_str.) Q73RM4_TRDE0(_Treponema_denticola_ATCC_35405) Q72AD6_DESVH(_Desulfovibrio_vulgaris_subsp._vulgaris_str.) Q75FU3_LEIN1(_Leptospira_interrogans_serovar_Copenhageni_str.) Q8EXU6_LEIN0(_Leptospira_interrogans_serovar_Lai_str.)
HBG228390	HYPOTHETICAL_PROTEIN_AF1126;_HYPOTHETICAL_PROTEIN_AQ_134;_HYPOTHETICAL_PROTEIN_MA2509;_HYPOTHETICAL	(no_actino_in_family)		:00548352	:00574384	O66531_AQAE0(_Aquifex_aeolicus_VF5) Q8TVJ9_MEKA0(_Methanopyrus_kandleri_AV19) Q8PSK4_MEMA0(_Methanosarcina_mazei_Go1) Q8TMY5_MEAC0(_Methanosarcina_acetivorans_C2A) O29139_ARFU0(_Archaeoglobus_fulgidus_DSM_4304) Q6KYW1_PITO0(_Picrophilus_torridus_DSM_9790) Q97AL1_THEVO(_Thermoplasma_volcanium) Q9HKU3_THEAC(_Thermoplasma_acidophilum)
HBG230386	HYPOTHETICAL_CONSERVED_PROTEIN;_HYPOTHETICAL_PROTEIN_AQ_1682;_HYPOTHETICAL_PROTEIN_DR0837;_HYPOTHETI	(no_actino_in_family)			:00571216	Q72LB2_THTH0(_Thermus_thermophilus_HB27) Q9RW32_DEIRA(_Deinococcus_radiodurans) Q9X0Y4_THMA0(_Thermotoga_maritima_MSB8) O67593_AQAE0(_Aquifex_aeolicus_VF5)
HBG233977	BLR3164_PROTEIN;_HYPOTHETICAL_PROTEIN_CJ0189C;_HYPOTHETICAL_PROTEIN_TP0894;_HYPOTHETICAL_PROTEIN_XAC_	(no_actino_in_family)			:00567554 :00567382	Q82WK8_NIEU0(_Nitrosomonas_europaea_ATCC_19718) Q6N2T8_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q89QG4_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q7MRC6_WOLSU(_Wolinella_succinogenes) Q7US44_RHOBA(_Pirellula_sp.) Q6LWC2_PHOPR(_Photobacterium_profundum) P73622_SYNY3(_Synechocystis_sp._PCC_6803) Q8P4E1_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PG01_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.) Q9PIU0_CAJE0(_Campylobacter_jejuni_subsp._jejuni_NCTC) Q8A4E9_BATH0(_Bacteroides_thetaiotaomicron_VPI-5482) O83864_TRPA0(_Treponema_pallidum_subsp._pallidum_str.) Q73JX7_TRDE0(_Treponema_denticola_ATCC_35405)
HBG235734	PYOVERDIN_CHROMOPHORE_BIOSYNTHETIC_PROTEIN_PVCC_	(no_actino_in_family)			:00568292 :00568293	Q8ZQ50_SATY0(_Salmonella_typhimurium_LT2) Q7C977_SAEN0(_Salmonella_enterica_subsp._enterica_serovar) Q8Z7Q4_SALTI(_Salmonella_typhi) Q83II1_SHFL0(_Shigella_flexneri_2a_str._301) Q74US5_YEPE2(_Yersinia_pestis_biovar_Medievalis_str.) Q7CHV9_YEPE0(_Yersinia_pestis_KIM) Q8ZFE5_YEPE1(_Yersinia_pestis_CO92) Q7N069_PHLU0(_Photorhabdus_luminescens_subsp._laumondii_TTO1) Q7N7X6_PHLU0(_Photorhabdus_luminescens_subsp._laumondii_TTO1) Q9CKT3_PASMU(_Pasteurella_multocida) Q9HWT7_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q7N9R8_PHLU0(_Photorhabdus_luminescens_subsp._laumondii_TTO1) PVCC_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q7MZM5_PHLU0(_Photorhabdus_luminescens_subsp._laumondii_TTO1)
HBG242462	7;8-DIHYDRO-8-OXOGUANINE-TRIPHOSPHATASE;_HYPOTHETICAL_PROTEIN_PA0990;_HYPOTHETICAL_PROTEIN_YBGD;_MUT	(no_actino_in_family)	:FRAAL_MUTT			Q739M4_BACE0(_Bacillus_cereus_ATCC_10987) Q6KTQ9_BAAN0(_Bacillus_anthracis_str._'Ames_Ancestor') Q81RJ7_BACAA(_Bacillus_anthracis_str._Ames) Q81EE8_BACCR(_Bacillus_cereus_ATCC_14579) Q9K8B7_BAHA0(_Bacillus_halodurans_C-125) Q796N8_BASU0(_Bacillus_subtilis_subsp._subtilis_str.) Q9I4X9_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q6N695_RHPA0(_Rhodopseudomonas_palustris_CGA009)
HBG242848	ALL3940_PROTEIN;_BACTERIOFERRITIN;_HYPOTHETICAL_CONSERVED_PROTEIN;_HYPOTHETICAL_PROTEIN_	(no_actino_in_family)	:FRAAL_PE786			Q81U91_BACAA(_Bacillus_anthracis_str._Ames) Q6KWA5_BAAN0(_Bacillus_anthracis_str._'Ames_Ancestor') Q73CH8_BACE0(_Bacillus_cereus_ATCC_10987) Q81H21_BACCR(_Bacillus_cereus_ATCC_14579) Q8ERY8_OCEIH(_Oceanobacillus_iheyensis) Q72IL4_THTH0(_Thermus_thermophilus_HB27) Q7UWW0_RHOBA(_Pirellula_sp.) Q6MMV9_BDEBA(_Bdellovibrio_bacteriovorus) Q8YQ94_ANASP(_Nostoc_sp._PCC_7120)
HBG243449	PUTATIVE_TRANSKETOLASE_N-TERMINAL_SECTION	71			:00569768	Q7CYA0_AGRT5(_Agrobacterium_tumefaciens_str._C58)
HBG243450	1-DEOXY-XYLULOSE_5-PHOSPHATE_SYNTHASE;_322AA_LONG_HYPOTHETICAL_TRANSKETOLASE;_AGR_C_3491P;_BLR2169_P	100			:00569769	Q7CY99_AGRT5(_Agrobacterium_tumefaciens_str._C58)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG244605	PUTATIVE_THIAMINE_BIOSYNTHESIS_PROTEIN_HI0357_	(no_actino_in_family)	:FRAAL_PE1556			Q81IG2_BACCR(_Bacillus_cereus_ATCC_14579) Q6KXU2_BAAN0(_Bacillus_anthracis_str._'Ames_Ancestor') Q81Z90_BACAA(_Bacillus_anthracis_str._Ames) Q73E68_BACE0(_Bacillus_cereus_ATCC_10987) Q97J86_CLAC0(_Clostridium_acetobutylicum_ATCC_824) Q9WYV4_THMA0(_Thermotoga_maritima_MSB8) Q8DMZ7_STRR6(_Streptococcus_pneumoniae_R6) Q97N69_STPN0(_Streptococcus_pneumoniae_TIGR4) Q8RGQ0_FUNU0(_Fusobacterium_nucleatum_subsp._nucleatum_ATCC) Q8G7Y1_BILO0(_Bifidobacterium_longum_NCC2705) Q82NG5_STRAW(_Streptomyces_avermitilis) Q98JV9_RHILO(_Mesorhizobium_loti) Q9K9G5_BAHA0(_Bacillus_halodurans_C-125) Q81HQ9_BACCR(_Bacillus_cereus_ATCC_14579) Q6KWX9_BAAN0(_Bacillus_anthracis_str._'Ames_Ancestor') Q81UX8_BACAA(_Bacillus_anthracis_str._Ames) Q73DB5_BACE0(_Bacillus_cereus_ATCC_10987) Q6LJH2_PHOPR(_Photobacterium_profundum) Q87JW5_VIBPA(_Vibrio_parahaemolyticus) Q9CLH1_PASMU(_Pasteurella_multocida) Y357_HAIN0(_Haemophilus_influenzae_Rd_KW20) Q7CS34_AGRT5(_Agrobacterium_tumefaciens_str._C58) Q8G2V1_BRSU0(_Brucella_suis_1330)
HBG245069	HYPOTHETICAL_PROTEIN_RP368	60	:FRAAL_PE3812			Q98MD6_RHILO(_Mesorhizobium_loti)
HBG246216	4-CARBOXYMUCONOLACTONE_DECARBOXYLASE;_CARBOXYMUCONOLACTONE_DECARBOXYLASE_RELATED_PROTEIN;_HYPOTHETIC	(no_actino_in_family)			:00567714	Q7VG63_HEHE0(_Helicobacter_hepaticus_ATCC_51449) Q6LZ22_METMP(_Methanococcus_maripaludis) Q8TTM1_MEAC0(_Methanosarcina_acetivorans_C2A)
HBG247100	HYPOTHETICAL_PROTEIN_AQ_888;_HYPOTHETICAL_PROTEIN_SMC01448;_MLL6598_PROTEIN	(no_actino_in_family)			:00571226	Q72IR5_THTH0(_Thermus_thermophilus_HB27) Q92NQ2_RHIME(_Sinorhizobium_meliloti) Q6MCX0_PASP0(_Parachlamydia_sp._UWE25)
HBG248632	BH3796_PROTEIN;_METHYLMALONYL-COA_MUTASE_PROTEIN;_METHYLMALONYL-COA_MUTASE	(no_actino_in_family)	:FRAAL_MCM	:00548799	:00569662	Q9K6D3_BAHA0(_Bacillus_halodurans_C-125) Q6MLZ8_BDEBA(_Bdellovibrio_bacteriovorus) Q8Y2U5_RASO0(_Ralstonia_solanacearum_GMI1000) Q72TB6_LEIN1(_Leptospira_interrogans_serovar_Copenhageni_str.) Q8F222_LEIN0(_Leptospira_interrogans_serovar_Lai_str.)
HBG250996	BLR2725_PROTEIN;_HYPOTHETICAL_ISOCHORISMATASE_FAMILY_PROTEIN;_HYPOTHETICAL_PROTEIN;_ISOCHORISMATASE	100	:FRAAL_DHBB		:00570528	Q89RN8_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q8FBK3_ECOL6(_Escherichia_coli_O6)
HBG250996	BLR2725_PROTEIN;_HYPOTHETICAL_ISOCHORISMATASE_FAMILY_PROTEIN;_HYPOTHETICAL_PROTEIN;_ISOCHORISMATASE	100	:FRAAL_PE6405		:00569935	Q887E2_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q987G0_RHILO(_Mesorhizobium_loti) Q887E0_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q987G2_RHILO(_Mesorhizobium_loti) Q89H44_BRJA0(_Bradyrhizobium_japonicum_USDA_110)
HBG251511	ALL2531_PROTEIN;_IRON-MOLIBDENUM_COFACTOR_PROCESSING_PROTEIN;_NIFX;_NIFX_NITROGEN_FIXATION_PROTEIN;	(no_actino_in_family)	:FRAAL_NIFX	:00546164	:00571388	Q6N0Z4_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q79UV9_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q44146_ANASP(_Nostoc_sp._PCC_7120) Q8YU30_ANASP(_Nostoc_sp._PCC_7120) Q749D8_GESU0(_Geobacter_sulfurreducens_PCA) Q7MRG1_WOLSU(_Wolinella_succinogenes)
HBG256008	BH0857_PROTEIN;_HYPOTHETICAL_PROTEIN;_POLYSACCHARIDE_DEACETYLASE_DOMAIN_PROTEIN;_POLYSACCHARIDE_DEAC	(no_actino_in_family)	:FRAAL_PE3214			Q72FU7_DESVH(_Desulfovibrio_vulgaris_subsp._vulgaris_str.) Q74FF0_GESU0(_Geobacter_sulfurreducens_PCA) Q7VG22_HEHE0(_Helicobacter_hepaticus_ATCC_51449) Q9KEJ5_BAHA0(_Bacillus_halodurans_C-125)
HBG258403	AGR_C_3674P;_BH0695_PROTEIN;_BLL5290_PROTEIN;_CONSERVED_HYPOTHETICAL_PROTEIN;_GLL1436_PROTEIN;_HYPOT	(no_actino_in_family)			:00570621	Q7NKP1_GLVI0(_Gloeobacter_violaceus_PCC_7421) Q98C00_RHILO(_Mesorhizobium_loti) Q6MH97_BDEBA(_Bdellovibrio_bacteriovorus) Q8PZJ2_MEMA0(_Methanosarcina_mazei_Go1) Q8TK15_MEAC0(_Methanosarcina_acetivorans_C2A) Q9KF04_BAHA0(_Bacillus_halodurans_C-125) Q72UJ3_LEIN1(_Leptospira_interrogans_serovar_Copenhageni_str.) Q8F0G0_LEIN0(_Leptospira_interrogans_serovar_Lai_str.)
HBG259116	BLL6614_PROTEIN;_HYPOTHETICAL_PROTEIN;_PUTATIVE_ESTERASE	(no_actino_in_family)			:00567731	Q9K3V2_STCO0(_Streptomyces_coelicolor_A3(2)) Q89FT5_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q730C7_BACE0(_Bacillus_cereus_ATCC_10987)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG261763	FORMIMINOTETRAHYDROFOLATE_CYCLODEAMINASE;_FORMIMINOTRANSFERASE-CYCLODEAMINASE_FAMILY_PROTEIN;_FORMIM_	(no_actino_in_family)			:00573066	Q8RFG2_FUNU0(_Fusobacterium_nucleatum_subsp._nucleatum_ATCC) Q9X1P6_THMA0(_Thermotoga_maritima_MSB8) Q97CL2_THEVO(_Thermoplasma_volcanium) Q9HI68_THEAC(_Thermoplasma_acidophilum) Q73RN9_TRDE0(_Treponema_denticola_ATCC_35405) Q8RCH3_THTE0(_Thermoanaerobacter_tengcongensis_MB4) Q8XJ93_CLPE0(_Clostridium_perfringens_str._13) Q97GQ6_CLAC0(_Clostridium_acetobutylicum_ATCC_824) Q891R4_CLTE0(_Clostridium_tetani_E88) Q8A4B2_BATH0(_Bacteroides_thetaiotaomicron_VPI-5482) Q7MX85_POGI0(_Porphyromonas_gingivalis_W83) Q99XR3_STPY0(_Streptococcus_pyogenes_M1_GAS) Q8NZ50_STRP8(_Streptococcus_pyogenes_MGAS8232) Q79W26_STPY1(_Streptococcus_pyogenes_SSI-1) Q8K5L9_STRP3(_Streptococcus_pyogenes_MGAS315)
HBG262752	ALR1000_PROTEIN;_ALR5307_PROTEIN;_BLL2752_PROTEIN;_BLL5217_PROTEIN;_BLL7645_PROTEIN;_GLR3165_PROTEIN	100	:FRAAL_PE1703		:00567278	Q72HU5_THTH0(_Thermus_thermophilus_HB27) Q7UUY4_RHOBA(_Pirellula_sp.) Q8PSY0_MEMA0(_Methanosarcina_mazei_Go1) Q8TNB2_MEAC0(_Methanosarcina_acetivorans_C2A)
HBG264140	HYPOTHETICAL_METHYLTRANSFERASE_PH0819_	(no_actino_in_family)		:00549569		Q8TWP0_MEKA0(_Methanopyrus_kandleri_AV19) O27812_METTH(_Methanothermobacter_thermautotrophicus_str._Delta_H) Y819_PYHO0(_Pyrococcus_horikoshii_OT3) Q74E14_GESU0(_Geobacter_sulfurreducens_PCA)
HBG270951	BLL0125_PROTEIN;_BLR3349_PROTEIN;_DIHYDROOROTASE;_HYPOTHETICAL_CONSERVED_PROTEIN;_HYPOTHETICAL_PROTE	61			:00569340	Q89Y30_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q98E16_RHILO(_Mesorhizobium_loti)
HBG271743	AGR_L_1767P;_AGR_L_3333P;_BLR2818_PROTEIN;_HYPOTHETICAL_PROTEIN_RSP0831;_HYPOTHETICAL_PROTEIN_SCO189_	(no_actino_in_family)	:FRAAL_PE2584			Q9RYT6_DEIRA(_Deinococcus_radiodurans) Q9HSQ3_HALN1(_Halobacterium_sp._NRC-1) Q89RF4_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q8XRK0_RASO0(_Ralstonia_solanacearum_GMI1000) Q888H1_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q88NN6_PSEPK(_Pseudomonas_putida_KT2440) Q92T73_RHIME(_Sinorhizobium_meliloti) Q7CRQ0_AGRT5(_Agrobacterium_tumefaciens_str._C58) Q7CTR6_AGRT5(_Agrobacterium_tumefaciens_str._C58) Q829Q0_STRAW(_Streptomyces_avermitilis) Q9X9X6_STCO0(_Streptomyces_coelicolor_A3(2))
HBG272055	AGR_L_1239P;_AGR_PAT_315P;_BLL4801_PROTEIN;_BLL5452_PROTEIN;_BLR3671_PROTEIN;_HYPOTHETICAL_PROTEIN_S	100	:FRAAL_PE5281			Q89KU9_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q98IN9_RHILO(_Mesorhizobium_loti)
HBG272194	AGR_C_3389P;_AGR_L_1192P;_AGR_L_244P;_HYPOTHETICAL_PROTEIN;_HYPOTHETICAL_TRANSCRIPTIONAL_REGULATOR;	100	:FRAAL_PE4705			Q89HX9_BRJA0(_Bradyrhizobium_japonicum_USDA_110)
HBG272519	AGR_C_2998P;_HYPOTHETICAL_PROTEIN_RA0090;_METHLYTRANSFERASE;_UBIE/COQ5_FAMILY;_METHYLTRANSFERASE;_ML	100			:00570375	Q98BY2_RHILO(_Mesorhizobium_loti)
HBG274450	HYPOTHETICAL_PROTEIN_MJ0386_	(no_actino_in_family)	:FRAAL_PE5583	:00549418	:00572662	Q8U1T8_PYFU0(_Pyrococcus_furiosus_DSM_3638) O27155_METTH(_Methanothermobacter_thermautotrophicus_str._Delta_H) Y386_MEJA0(_Methanocaldococcus_jannaschii_DSM_2661) O30237_ARFU0(_Archaeoglobus_fulgidus_DSM_4304) Q8TJV8_MEAC0(_Methanosarcina_acetivorans_C2A) Q74N46_NAEQ0(_Nanoarchaeum_equitans_Kin4-M) Q6L318_PITO0(_Picrophilus_torridus_DSM_9790) Q97CJ7_THEVO(_Thermoplasma_volcanium) Q7MTF3_POGI0(_Porphyromonas_gingivalis_W83) Q8RED2_FUNU0(_Fusobacterium_nucleatum_subsp._nucleatum_ATCC) Q9KFX8_BAHA0(_Bacillus_halodurans_C-125) Q8PFX9_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.) Q7NYP3_CHVI0(_Chromobacterium_violaceum_ATCC_12472) Q72WF4_DESVH(_Desulfovibrio_vulgaris_subsp._vulgaris_str.) Q8KDC5_CHTE0(_Chlorobium_tepidum_TLS) Q8DSM5_STMU0(_Streptococcus_mutans_UA159) Q99YS8_STPY0(_Streptococcus_pyogenes_M1_GAS) Q72NB8_LEIN1(_Leptospira_interrogans_serovar_Copenhageni_str.) Q82W51_NIEU0(_Nitrosomonas_europaea_ATCC_19718) Q6L364_PITO0(_Picrophilus_torridus_DSM_9790) Q8R6X6_THTE0(_Thermoanaerobacter_tengcongensis_MB4) Q9X2B6_THMA0(_Thermotoga_maritima_MSB8)
HBG276254	HYPOTHETICAL_PROTEIN_TM0786;_HYPOTHETICAL_PROTEIN_TV0039;_HYPOTHETICAL_PROTEIN_TA0034;_MLL1513_PROTE	73	:FRAAL_PE4504		:00572742	Q82XT5_NIEU0(_Nitrosomonas_europaea_ATCC_19718) Q83DR4_COBU0(_Coxiella_burnetii_RSA_493)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG277085	ALR4493_PROTEIN;_GLYCOSYL_TRANSFERASE_PROTEIN;_GLYCOSYL_TRANSFERASE;_GROUP_2_FAMILY_PROTEIN;_PREDICT_	(no_actino_in_family)		:00547567	:00570499 :00568503	Q97GP6_CLAC0(_Clostridium_acetobutylicum_ATCC_824) Q97GP5_CLAC0(_Clostridium_acetobutylicum_ATCC_824) Q7MXD7_POGI0(_Porphyromonas_gingivalis_W83) Q8YNR8_ANASP(_Nostoc_sp._PCC_7120) Q832M8_ENFA0(_Enterococcus_faecalis_V583) Q8Y1K2_RASO0(_Ralstonia_solanacearum_GMI1000) Q7MVD0_POGI0(_Porphyromonas_gingivalis_W83) Q8A834_BATH0(_Bacteroides_thetaiotaomicron_VPI-5482) Q72QJ4_LEIN1(_Leptospira_interrogans_serovar_Copenhageni_str.) Q7CM42_LEIN0(_Leptospira_interrogans_serovar_Lai_str.)
HBG278079	PROTEASE_SYNTHASE_AND_SPORULATION_NEGATIVE_REGULATORY_PROTEIN_PAI_2_	(no_actino_in_family)	:FRAAL_PE3748 :FRAAL_PE2538		:00566693	Q7VVP7_BORPE(_Bordetella_pertussis) Q7WAS6_BORPA(_Bordetella_parapertussis) Q7WJY5_BORBR(_Bordetella_bronchiseptica) Q8XRZ2_RASO0(_Ralstonia_solanacearum_GMI1000) Q9HWJ9_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q7N6U4_PHLU0(_Photorhabdus_luminescens_subsp._laumondii_TTO1) Q98NS0_RHILO(_Mesorhizobium_loti) Q9PBU7_XYFA0(_Xylella_fastidiosa_9a5c) Q7NTU8_CHVI0(_Chromobacterium_violaceum_ATCC_12472) Q98A31_RHILO(_Mesorhizobium_loti) Q87U10_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q88C40_PSEPK(_Pseudomonas_putida_KT2440) Q7W076_BORPE(_Bordetella_pertussis) Q7WCA5_BORPA(_Bordetella_parapertussis) Q7WQA9_BORBR(_Bordetella_bronchiseptica) Q9RTS3_DEIRA(_Deinococcus_radiodurans) Q6MH79_BDEBA(_Bdellovibrio_bacteriovorus) Q8PAC8_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PM19_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.) Q6KPY8_BAAN0(_Bacillus_anthracis_str._'Ames_Ancestor') Q81YH8_BACAA(_Bacillus_anthracis_str._Ames) Q9KFR5_BAHA0(_Bacillus_halodurans_C-125) PAIB_BASU0(_Bacillus_subtilis_subsp._subtilis_str.) Q734X7_BACE0(_Bacillus_cereus_ATCC_10987)
HBG279067	HYPOTHETICAL_PROTEIN_CP0246;_HYPOTHETICAL_PROTEIN_HI0948;_PROTEIN_HI0321	100		:00548622		Q7NEK5_GLVI0(_Gloeobacter_violaceus_PCC_7421)
HBG280032	ALR0568_PROTEIN;_GLL1362_PROTEIN;_HYPOTHETICAL_PROTEIN_XAC2848;_HYPOTHETICAL_PROTEIN_XCC2687	(no_actino_in_family)	:FRAAL_PE2348		:00574186	Q7NKW4_GLVI0(_Gloeobacter_violaceus_PCC_7421) Q8YZB1_ANASP(_Nostoc_sp._PCC_7120) Q8P7C4_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PIP7_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.)
HBG280187	IRON-UTILIZATION_PERIPLASMIC_PROTEIN_PRECURSOR;_MAJOR_FERRIC_IRON_BINDING_PROTEIN_PRECURSOR	100		:00547607	:00568334	Q8ZCM4_YEPE1(_Yersinia_pestis_CO92) Q74SL6_YEPE2(_Yersinia_pestis_biovar_Medievalis_str.) Q7CJD9_YEPE0(_Yersinia_pestis_KIM) Q98G45_RHILO(_Mesorhizobium_loti)
HBG280188	ABC-TYPE_FE3+_TRANSPORT_SYSTEM;_PERMEASE_COMPONENT;_AGR_C_1092P;_AGR_C_717P;_FBPB;_IRON;_INNER_MEMBR	100		:00547555	:00568335	Q98G44_RHILO(_Mesorhizobium_loti) Q74SL5_YEPE2(_Yersinia_pestis_biovar_Medievalis_str.) Q8D0V7_YEPE0(_Yersinia_pestis_KIM) Q8ZCM3_YEPE1(_Yersinia_pestis_CO92)
HBG280639	AGR_C_3379P;_AGR_L_2793P;_CONSERVED_HYPOTHETICAL_PROTEIN;_HYPOTHETICAL_CYTOSOLIC_PROTEIN_BMEI1119;_H	60		:00547227		Q98JQ9_RHILO(_Mesorhizobium_loti) Q8XX01_RASO0(_Ralstonia_solanacearum_GMI1000)
HBG281221	AGR_C_4391P;_HYPOTHETICAL_PROTEIN_CT0928;_HYPOTHETICAL_PROTEIN_VC2743;_HYPOTHETICAL_PROTEIN_VP0123;	(no_actino_in_family)	:FRAAL_PE2484			Q98KH7_RHILO(_Mesorhizobium_loti) Q7CX38_AGRT5(_Agrobacterium_tumefaciens_str._C58) Q7NV70_CHVI0(_Chromobacterium_violaceum_ATCC_12472)
HBG281583	BLR1197_PROTEIN;_BLR2982_PROTEIN;_EPOXIDE_HYDROLASE;_EPHB;_EPOXIDE_HYDROLASE_PROTEIN;_PROBABLE_EPOXI	(no_actino_in_family)	:FRAAL_PE1292			Q988M0_RHILO(_Mesorhizobium_loti) Q89V60_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q7NXY2_CHVI0(_Chromobacterium_violaceum_ATCC_12472) Q98GK2_RHILO(_Mesorhizobium_loti) Q89QZ0_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q9HX27_PSAE0(_Pseudomonas_aeruginosa_PAO1)
HBG282919	NON-HAEM_IRON_PROTEIN;_NIGERYTHRIN;_NON-HAEM_IRON_PROTEIN;_PUTATIVE_RUBRERYTHRIN;_RUBREDOXIN:RUBRERY	88	:FRAAL_PE5115	:00547866 :00547998	:00569580	Q96XZ7_SUTO0(_Sulfolobus_tokodaii_str._7) Q97VI1_SULSO(_Sulfolobus_solfataricus)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG285504	BH1513_PROTEIN;_BLR2726_PROTEIN;_CONSERVED_HYPOTHETICAL_PROTEIN;_GLR3054_PROTEIN;_HYPOTHETICAL_PROTE	100	:FRAAL_PE6300		:00570522	Q89RN7_BRJA0(_Bradyrhizobium_japonicum_USDA_110)
HBG292172	ALL2220_PROTEIN;_HYPOTHETICAL_PROTEIN;_MLL1439_PROTEIN	(no_actino_in_family)		:00548198		Q9I039_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q98KK1_RHILO(_Mesorhizobium_loti) Q7NWZ0_CHVI0(_Chromobacterium_violaceum_ATCC_12472)
HBG292542	BLL2664_PROTEIN;_BLR2659_PROTEIN;_BLR4890_PROTEIN;_HIGHLY_CONSERVED_PROTEIN_PRECURSOR;_HYPOTHETICAL	85			:00572167	Q7W381_BORPA(_Bordetella_parapertussis) Q7WEK0_BORBR(_Bordetella_bronchiseptica) Q6NCK7_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q89KL5_BRJA0(_Bradyrhizobium_japonicum_USDA_110)
HBG292658	HYPOTHETICAL_PROTEIN_MJ0963	100			:00570830	Q7NLA0_GLVI0(_Gloeobacter_violaceus_PCC_7421)
HBG297240	PUTATIVE_DIOXYGENASE_RV3406/MT3514/MB3440	62			:00569272	Q7N7U7_PHLU0(_Photorhabdus_luminescens_subsp._laumondii_TTO1)
HBG299033	HYPOTHETICAL_PROTEIN_MJ0378_	(no_actino_in_family)	:FRAAL_PE5582	:00549410	:00572661	Q6L363_PITO0(_Picrophilus_torridus_DSM_9790) Q8R6X5_THTE0(_Thermoanaerobacter_tengcongensis_MB4) Q9X2B7_THMA0(_Thermotoga_maritima_MSB8) O66692_AQAE0(_Aquifex_aeolicus_VF5) Q7MTF2_POGI0(_Porphyromonas_gingivalis_W83) Q895W8_CLTE0(_Clostridium_tetani_E88) Q8RED1_FUNU0(_Fusobacterium_nucleatum_subsp._nucleatum_ATCC) Q8TJV7_MEAC0(_Methanosarcina_acetivorans_C2A) O58938_PYHO0(_Pyrococcus_horikoshii_OT3) Q8U1T7_PYFU0(_Pyrococcus_furiosus_DSM_3638) Y378_MEJA0(_Methanocaldococcus_jannaschii_DSM_2661) O27156_METTH(_Methanothermobacter_thermautotrophicus_str._Delta_H) O30236_ARFU0(_Archaeoglobus_fulgidus_DSM_4304) Q74N45_NAEQ0(_Nanoarchaeum_equitans_Kin4-M) Q894S2_CLTE0(_Clostridium_tetani_E88) Q6L317_PITO0(_Picrophilus_torridus_DSM_9790) Q97CJ6_THEVO(_Thermoplasma_volcanium) Q6ZEC7_SYNY3(_Synechocystis_sp._PCC_6803) Q8YZS6_ANASP(_Nostoc_sp._PCC_7120) Q8YWN6_ANASP(_Nostoc_sp._PCC_7120) Q745V9_THTH0(_Thermus_thermophilus_HB27) Q99YS7_STPY0(_Streptococcus_pyogenes_M1_GAS) Q9KFX9_BAHA0(_Bacillus_halodurans_C-125) Q72WF5_DESVH(_Desulfovibrio_vulgaris_subsp._vulgaris_str.)
HBG303737	HYPOTHETICAL_PROTEIN_SPY0948	(no_actino_in_family)		:00548041 :00548040		Q87NL5_VIBPA(_Vibrio_parahaemolyticus) Q9A034_STPY0(_Streptococcus_pyogenes_M1_GAS)
HBG304911	BETA-LACTAMASE_PRECURSOR;_BLL0805_PROTEIN;_BLL2798_PROTEIN;_BLL5708_PROTEIN;_BLL5709_PROTEIN;_BLR544	62	:FRAAL_PE6369			Q89ID1_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q6N609_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q89IU3_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q6N9E1_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q89ID0_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q6N4Q3_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q89J43_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q89F61_BRJA0(_Bradyrhizobium_japonicum_USDA_110)
HBG304995	BLL5454_PROTEIN;_FADE36;_HYPOTHETICAL_CONSERVED_PROTEIN;_HYPOTHETICAL_PROTEIN_CC0299;_HYPOTHETICAL_P	100			:00566937	Q9I2R6_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q88G01_PSEPK(_Pseudomonas_putida_KT2440) Q882A5_PSSY0(_Pseudomonas_syringae_pv._tomato_str.)
HBG306542	PUTATIVE_HTH-TYPE_TRANSCRIPTIONAL_REGULATOR_YCJC;_PUTATIVE_HTH-TYPE_TRANSCRIPTIONAL_REGULATOR_YDCN	100	:FRAAL_PE1892			Q882P4_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q7CUD3_AGRT5(_Agrobacterium_tumefaciens_str._C58) Q92T82_RHIME(_Sinorhizobium_meliloti) Q986J7_RHILO(_Mesorhizobium_loti)
HBG306685	HYPOTHETICAL_PROTEIN;_HYPOTHETICAL_TRANSCRIPTIONAL_REGULATOR;_PROBABLE_TRANSCRIPTIONAL_REGULATOR;_TE	96	:FRAAL_PE2460		:00571495	Q88MQ5_PSEPK(_Pseudomonas_putida_KT2440) Q9HXW2_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q8P5K6_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PH01_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.)
HBG306897	2-OXO-HEPTA-3-ENE-1;7-DIOATE_HYDRATASE;_2-OXO-HEPTA-3-ENE-1;7-DIOIC_ACID_HYDRATASE;_2-HYDROXYPENTA-2	64	:FRAAL_PE3409			Q7W8E9_BORPA(_Bordetella_parapertussis) Q7WM10_BORBR(_Bordetella_bronchiseptica)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG307942	ACETYLTRANSFERASE;_GNAT_FAMILY;_HYPOTHETICAL_CONSERVED_PROTEIN;_HYPOTHETICAL_PROTEIN_YFIL;_IAA_ACETY_	(no_actino_in_family)		:00548513		Q8ET83_OCEIH(_Oceanobacillus_iheyensis) O32248_BASU0(_Bacillus_subtilis_subsp._subtilis_str.) Q73AR9_BACE0(_Bacillus_cereus_ATCC_10987) Q6KUT7_BAAN0(_Bacillus_anthracis_str._'Ames_Ancestor') Q81SP9_BACAA(_Bacillus_anthracis_str._Ames) Q81FK8_BACCR(_Bacillus_cereus_ATCC_14579) Q837L3_ENFA0(_Enterococcus_faecalis_V583) Q88TR1_LAPL0(_Lactobacillus_plantarum_WCFS1) Q98JV2_RHILO(_Mesorhizobium_loti) Q9CHW9_LALA0(_Lactococcus_lactis_subsp._lactis_Il1403) Q8PY48_MEMA0(_Methanosarcina_mazei_Go1) Q8R615_FUNU0(_Fusobacterium_nucleatum_subsp._nucleatum_ATCC) Q8A8A9_BATH0(_Bacteroides_thetaiotaomicron_VPI-5482) Q893D7_CLTE0(_Clostridium_tetani_E88) Q97DM7_CLAC0(_Clostridium_acetobutylicum_ATCC_824) Q929Q2_LIIN0(_Listeria_innocua_Clip11262) Q71XP6_LIMO0(_Listeria_monocytogenes_str._4b_F2365) Q8Y5E7_LISMO(_Listeria_monocytogenes)
HBG308943	BLR7360_PROTEIN;_BLR7361_PROTEIN;_HYPOTHETICAL_PROTEIN_RSP1350	(no_actino_in_family)	:FRAAL_PE3945			Q89DS9_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q8XGZ4_RASO0(_Ralstonia_solanacearum_GMI1000) Q6NAA4_RHPA0(_Rhodopseudomonas_palustris_CGA009)
HBG312466	FE3+_ABC_TRANSPORTER;_PERIPLASMIC_IRON-BINDING_PROTEIN;_IRON;_PERIPLASMIC-BINDING_PROTEIN_OF_ABC_TRA	(no_actino_in_family)			:00570930	Q8KFB7_CHTE0(_Chlorobium_tepidum_TLS) Q8YQ09_ANASP(_Nostoc_sp._PCC_7120) Q8A5Y8_BATH0(_Bacteroides_thetaiotaomicron_VPI-5482) Q8A6D2_BATH0(_Bacteroides_thetaiotaomicron_VPI-5482)
HBG320067	CONSERVED_HYPOTHETICAL_PROTEIN;_HYPOTHETICAL_PROTEIN_YBEH;_HYPOTHETICAL_PROTEIN_YIIE;_MLR1523_PROTEI	(no_actino_in_family)	:FRAAL_PE4435			Q7P1V8_CHVI0(_Chromobacterium_violaceum_ATCC_12472) Q98KD7_RHILO(_Mesorhizobium_loti) Q835M7_ENFA0(_Enterococcus_faecalis_V583) Q9CH76_LALA0(_Lactococcus_lactis_subsp._lactis_Il1403)
HBG320565	PUTATIVE_HTH-TYPE_TRANSCRIPTIONAL_REGULATOR_AF1627;_PUTATIVE_HTH-TYPE_TRANSCRIPTIONAL_REGULATOR_AF17	(no_actino_in_family)		:00549831		Q9RRN4_DEIRA(_Deinococcus_radiodurans) Q9A662_CACR0(_Caulobacter_crescentus_CB15) Q9HWV1_PSAE0(_Pseudomonas_aeruginosa_PAO1)
HBG321374	GLYOXALASE/BLEOMYCIN_RESISTANCE_PROTEIN/DIOXYGENASE_FAMILY_PROTEIN;_HYPOTHETICAL_PROTEIN_CC2142	(no_actino_in_family)	:FRAAL_PE3200		:00570988	Q9A6F2_CACR0(_Caulobacter_crescentus_CB15) Q8ECN6_SHON0(_Shewanella_oneidensis_MR-1) Q72U26_LEIN1(_Leptospira_interrogans_serovar_Copenhageni_str.) Q8CXS0_LEIN0(_Leptospira_interrogans_serovar_Lai_str.)
HBG325734	BENZOATE_MEMBRANE_TRANSPORT_PROTEIN_PRECURSOR;_BENZOATE_MEMBRANE_TRANSPORT_PROTEIN;_PUTATIVE;_BENZOA	79		:00548714		Q92N69_RHIME(_Sinorhizobium_meliloti)
HBG326072	4-CARBOXYMUCONOLACTONE_DECARBOXYLASE_PROTEIN;_4-CARBOXYMUCOLACTONE_DECARBOXYLASE;_AGR_L_653P;_AGR_PA	85			:00568052	Q7W124_BORPA(_Bordetella_parapertussis) Q7WNR4_BORBR(_Bordetella_bronchiseptica)
HBG333858	AGR_L_1305P;_HYPOTHETICAL_PROTEIN_CT1203;_HYPOTHETICAL_PROTEIN_MA1733;_HYPOTHETICAL_PROTEIN_MM2633;	(no_actino_in_family)	:FRAAL_PE4892	:00547601	:00572471	Q7CUA9_AGRT5(_Agrobacterium_tumefaciens_str._C58) Q8KD51_CHTE0(_Chlorobium_tepidum_TLS) Q8PTS9_MEMA0(_Methanosarcina_mazei_Go1) Q8TQ19_MEAC0(_Methanosarcina_acetivorans_C2A)
HBG338129	4-HYDROXY-2-OXOVALERATE_ALDOLASE;_ALDOLASE_PROTEIN;_HYPOTHETICAL_PROTEIN;_MW0095_PROTEIN;_SA0118_PRO_	(no_actino_in_family)			:00569338	Q99X93_STAAM(_Staphylococcus_aureus_subsp._aureus_Mu50) Q7A1Z4_STAAW(_Staphylococcus_aureus_subsp._aureus_MW2) Q7A863_STAAN(_Staphylococcus_aureus_subsp._aureus_N315) Q8XSP4_RASO0(_Ralstonia_solanacearum_GMI1000) Q8P6C1_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PHS4_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG338173	GLL2605_PROTEIN;_HYPOTHETICAL_PROTEIN_XAC0801;_HYPOTHETICAL_PROTEIN_XCC2910;_HYPOTHETICAL_PROTEIN_ML	(no_actino_in_family)		:00546357 :00549875		Q98J83_RHILO(_Mesorhizobium_loti) Q7NHD2_GLVI0(_Gloeobacter_violaceus_PCC_7421) Q98IZ2_RHILO(_Mesorhizobium_loti) Q8P6Q7_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PP87_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.)
HBG339599	TRANSPOSASE_FOR_TRANSPOSON_GAMMA-DELTA_	(no_actino_in_family)			:00570597 :00566682	Q82W60_NIEU0(_Nitrosomonas_europaea_ATCC_19718) Q7AQU8_SALTI(_Salmonella_typhi) Q88BQ4_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q8CM48_SHON0(_Shewanella_oneidensis_MR-1) TNPA_ESCO0(_Escherichia_coli_K12) Q7ARH3_YEPE2(_Yersinia_pestis_biovar_Medievalis_str.) Q7ARH3_YEPE1(_Yersinia_pestis_CO92) Q8PRG5_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.) Q8PRM4_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.)
HBG340842	ASL1824_PROTEIN;_ASL2714_PROTEIN;_CONSERVED_PROTEIN;_HYPOTHETICAL_CONSERVED_PROTEIN;_HYPOTHETICAL_PH_	(no_actino_in_family)	:FRAAL_PE212	:00547300	:00568768	Q8TIM1_MEAC0(_Methanosarcina_acetivorans_C2A) Q8YVZ2_ANASP(_Nostoc_sp._PCC_7120) Q74XS2_YEPE2(_Yersinia_pestis_biovar_Medievalis_str.) Q7VZH0_BORPE(_Bordetella_pertussis) Q7W5Z1_BORPA(_Bordetella_parapertussis) Q7MBH8_VIBVY(_Vibrio_vulnificus_YJ016) Q886E5_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q7W8Q4_BORPA(_Bordetella_parapertussis) Q8TNZ2_MEAC0(_Methanosarcina_acetivorans_C2A) Q8YTK3_ANASP(_Nostoc_sp._PCC_7120) Q73LM4_TRDE0(_Treponema_denticola_ATCC_35405) Q98J18_RHILO(_Mesorhizobium_loti) Q92K60_RHIME(_Sinorhizobium_meliloti) Q9A3L9_CACR0(_Caulobacter_crescentus_CB15) Q93MC0_CLPE0(_Clostridium_perfringens_str._13) Q73KG6_TRDE0(_Treponema_denticola_ATCC_35405) Q8P1Q8_STRP8(_Streptococcus_pyogenes_MGAS8232) Q72HU0_THTH0(_Thermus_thermophilus_HB27) Q8PVN8_MEMA0(_Methanosarcina_mazei_Go1) Q8TSN0_MEAC0(_Methanosarcina_acetivorans_C2A) Q8Q0H6_MEMA0(_Methanosarcina_mazei_Go1) P72914_SYNY3(_Synechocystis_sp._PCC_6803) Q9X138_THMA0(_Thermotoga_maritima_MSB8) Q8U199_PYFU0(_Pyrococcus_furiosus_DSM_3638)
HBG345268	TYPE_I_RESTRICTION_ENZYME_ECOKI_M_PROTEIN;_TYPE_I_RESTRICTION_ENZYME_STYSJI_M_PROTEIN	(no_actino_in_family)	:FRAAL_HSDM			Q8XTZ9_RASO0(_Ralstonia_solanacearum_GMI1000) Q8TN79_MEAC0(_Methanosarcina_acetivorans_C2A) Q7UE19_RHOBA(_Pirellula_sp.) Q7UE34_RHOBA(_Pirellula_sp.) Q8DQ91_STRR6(_Streptococcus_pneumoniae_R6) Q97RD6_STPN0(_Streptococcus_pneumoniae_TIGR4)
HBG346775	HYPOTHETICAL_PROTEIN_MJ0377_	(no_actino_in_family)	:FRAAL_PE5581		:00572660	Q895W9_CLTE0(_Clostridium_tetani_E88) Q8RED0_FUNU0(_Fusobacterium_nucleatum_subsp._nucleatum_ATCC) O66693_AQAE0(_Aquifex_aeolicus_VF5) Q7MTF1_POGI0(_Porphyromonas_gingivalis_W83)
HBG349420	MULTIDRUG-EFFLUX_TRANSPORTER_1_REGULATOR	98	:FRAAL_PE3696			Q8YZW0_ANASP(_Nostoc_sp._PCC_7120)
HBG352690	AMINE_OXIDASE;_COPPER_AMINE_OXIDASE;_PUTATIVE_COPPER_AMINE_OXIDASE;_SPAC2E1P3.04_PROTEIN_	(no_actino_in_family)			:00569958	Q97XM1_SULSO(_Sulfolobus_solfataricus) Q989X2_RHILO(_Mesorhizobium_loti) Q8YRL5_ANASP(_Nostoc_sp._PCC_7120) Q9SLB1_ARATH(_Arabidopsis_thaliana)
HBG352748	HYPOTHETICAL_PROTEIN_YAAA;_HYPOTHETICAL_PROTEIN_YBCJ_	(no_actino_in_family)	:FRAAL_PE5665	:00546768	:00566833	Q839Z3_ENFA0(_Enterococcus_faecalis_V583) Q890K6_LAPL0(_Lactobacillus_plantarum_WCFS1) Q81JD3_BACCR(_Bacillus_cereus_ATCC_14579) Q6KYS9_BAAN0(_Bacillus_anthracis_str._'Ames_Ancestor') Q81W33_BACAA(_Bacillus_anthracis_str._Ames) Q73FK3_BACE0(_Bacillus_cereus_ATCC_10987) YAAA_BASU0(_Bacillus_subtilis_subsp._subtilis_str.) Q8EU86_OCEIH(_Oceanobacillus_iheyensis) Q98RK5_MYPU0(_Mycoplasma_pulmonis_UAB_CTIP) Q6KIT5_MYMO0(_Mycoplasma_mobile_163K) Q8RDL4_THTE0(_Thermoanaerobacter_tengcongensis_MB4) Q9RUJ5_DEIRA(_Deinococcus_radiodurans) Q7UVU5_RHOBA(_Pirellula_sp.) Q7V1N5_PROMP(_Prochlorococcus_marinus_subsp._pastoris_str.) Q7VC42_PRMA0(_Prochlorococcus_marinus_subsp._marinus_str.) Q7V7D1_PROMM(_Prochlorococcus_marinus_str._MIT_9313) Q7U803_SYNPX(_Synechococcus_sp._WH_8102) Q7NP33_GLVI0(_Gloeobacter_violaceus_PCC_7421) Q8YS38_ANASP(_Nostoc_sp._PCC_7120) Q74FY8_GESU0(_Geobacter_sulfurreducens_PCA) Q8RHA8_FUNU0(_Fusobacterium_nucleatum_subsp._nucleatum_ATCC) Q8XPG0_CLPE0(_Clostridium_perfringens_str._13) Q97N33_CLAC0(_Clostridium_acetobutylicum_ATCC_824)
HBG352788	HYPOTHETICAL_PROTEIN_HI0322;_HYPOTHETICAL_PROTEIN_HI0947;_HYPOTHETICAL_PROTEIN_MJ0915;_HYPOTHETICAL	58		:00546024		Q82WV4_NIEU0(_Nitrosomonas_europaea_ATCC_19718) Q6ZEN6_SYNY3(_Synechocystis_sp._PCC_6803) Q82T21_NIEU0(_Nitrosomonas_europaea_ATCC_19718) Q6ZEJ5_SYNY3(_Synechocystis_sp._PCC_6803)
HBG352788	HYPOTHETICAL_PROTEIN_HI0322;_HYPOTHETICAL_PROTEIN_HI0947;_HYPOTHETICAL_PROTEIN_MJ0915;_HYPOTHETICAL	83		:00548623		Q7NEK4_GLVI0(_Geobacter_sulfurreducens_PCA)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG356981	HYPOTHETICAL_UPF0087_PROTEIN_BH0655;_HYPOTHETICAL_UPF0087_PROTEIN_MTH1285;_HYPOTHETICAL_UPF0087_PROT	62	:FRAAL_PE2474		:00568443	Q98DX1_RHILO(_Mesorhizobium_loti)
HBG357414	AGR_L_2789P;_ACYL-COA_DEHYDROGENASE_FAMILY_PROTEIN;_ACYL-COA_DEHYDROGENASE-RELATED_PROTEIN;_ACYL-COA	93	:FRAAL_PE3867	:00549600	:00574085 :00574000 :00572891	Q89G93_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q9I0P0_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q885P1_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q88HX7_PSEPK(_Pseudomonas_putida_KT2440)
HBG357589	HYPOTHETICAL_PROTEIN_HI0902;_HYPOTHETICAL_PROTEIN_MJ0441_	(no_actino_in_family)	:FRAAL_PE4244			Q74GH8_GESU0(_Geobacter_sulfurreducens_PCA) Q8RDE0_THTE0(_Thermoanaerobacter_tengcongensis_MB4) Q928M9_LIIN0(_Listeria_innocua_Clip11262) Q71X21_LIMO0(_Listeria_monocytogenes_str._4b_F2365) Q8Y4N2_LISMO(_Listeria_monocytogenes) Q9CDK1_LALA0(_Lactococcus_lactis_subsp._lactis_Il1403) Q88XE9_LAPL0(_Lactobacillus_plantarum_WCFS1) Q6ML59_BDEBA(_Bdellovibrio_bacteriovorus) Q7NE90_GLVI0(_Gloeobacter_violaceus_PCC_7421) Q8KEE7_CHTE0(_Chlorobium_tepidum_TLS) Q7W0X7_BORPA(_Bordetella_parapertussis) Q7WNA1_BORBR(_Bordetella_bronchiseptica) Q8FUP2_BRSU0(_Brucella_suis_1330) Q8YDQ8_BRME0(_Brucella_melitensis_16M)
HBG357754	AGR_C_513P;_BLL5372_PROTEIN;_BLR0684_PROTEIN;_CONSERVED_HYPOTHETICAL_PROTEIN;_HYPOTHETICAL_CYTOSOLIC	80			:00571794	Q930M0_RHIME(_Sinorhizobium_meliloti) Q89JB1_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q984S8_RHILO(_Mesorhizobium_loti)
HBG360917	HYPOTHETICAL_PROTEIN_BUSG463;_HYPOTHETICAL_PROTEIN_HI1191;_HYPOTHETICAL_PROTEIN_MJ1347;_HYPOTHETICAL_	78		:00548690		Q8Y1F2_RASO0(_Ralstonia_solanacearum_GMI1000) Q87CW1_XYLFT(_Xylella_fastidiosa_Temecula1) Q9PCC3_XYFA0(_Xylella_fastidiosa_9a5c) Q8P6F6_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PHW1_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.) Q82XN5_NIEU0(_Nitrosomonas_europaea_ATCC_19718) Q9I4Z2_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q87Y44_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q88NI3_PSEPK(_Pseudomonas_putida_KT2440) Q83A28_COBU0(_Coxiella_burnetii_RSA_493) Q7UFU6_RHOBA(_Pirellula_sp.) Q74FW9_GESU0(_Geobacter_sulfurreducens_PCA)
HBG361087	HYPOTHETICAL_PROTEIN_MJ0126;_HYPOTHETICAL_PROTEIN_MJ0128;_HYPOTHETICAL_PROTEIN_MJ0141;_HYPOTHETICAL_	(no_actino_in_family)	:FRAAL_PE2644 :FRAAL_PE4454	:00546334 :00546441 :00546436	:00566641	Q7NFK1_GLVI0(_Gloeobacter_violaceus_PCC_7421) Q8DGT1_SYNEL(_Synechococcus_elongatus) Q72I47_THTH0(_Thermus_thermophilus_HB27) Q9JZ71_NEME0(_Neisseria_meningitidis_MC58) Q7CXB8_AGRT5(_Agrobacterium_tumefaciens_str._C58) Q8CMC8_SHON0(_Shewanella_oneidensis_MR-1) Q87AU7_XYLFT(_Xylella_fastidiosa_Temecula1) Q82VK7_NIEU0(_Nitrosomonas_europaea_ATCC_19718) Q9RTF8_DEIRA(_Deinococcus_radiodurans) Q6ND33_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q9A3X1_CACR0(_Caulobacter_crescentus_CB15) Q8ABE2_BATH0(_Bacteroides_thetaiotaomicron_VPI-5482) Q8KEJ4_CHTE0(_Chlorobium_tepidum_TLS) P73498_SYNY3(_Synechocystis_sp._PCC_6803) P72997_SYNY3(_Synechocystis_sp._PCC_6803) Q6ZEW3_SYNY3(_Synechocystis_sp._PCC_6803) Q73LK8_TRDE0(_Treponema_denticola_ATCC_35405)
HBG362129	ASPARATE-PROTON_SYMPORTER_	(no_actino_in_family)	:FRAAL_PE2998			Q83DX2_COBU0(_Coxiella_burnetii_RSA_493) Q83AT4_COBU0(_Coxiella_burnetii_RSA_493) Q83EA0_COBU0(_Coxiella_burnetii_RSA_493) Q83EG8_COBU0(_Coxiella_burnetii_RSA_493) Q45577_BASU0(_Bacillus_subtilis_subsp._subtilis_str.) Q8P4N3_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PGA0_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.) Q7NZQ0_CHVI0(_Chromobacterium_violaceum_ATCC_12472) Q8XT33_RASO0(_Ralstonia_solanacearum_GMI1000) Q74LB7_LAJO0(_Lactobacillus_johnsonii_NCC_533) Q8CN47_STEP0(_Staphylococcus_epidermidis_ATCC_12228) Q7A3P7_STAAN(_Staphylococcus_aureus_subsp._aureus_N315) Q99RI1_STAAM(_Staphylococcus_aureus_subsp._aureus_Mu50) Q8NUY6_STAAW(_Staphylococcus_aureus_subsp._aureus_MW2) ASPP_BASU0(_Bacillus_subtilis_subsp._subtilis_str.) Q8P842_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PJI9_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.) Q88NF2_PSEPK(_Pseudomonas_putida_KT2440) Q7MB35_PHLU0(_Photorhabdus_luminescens_subsp._laumondii_TTO1) Q97C85_THEVO(_Thermoplasma_volcanium) Q9HLS7_THEAC(_Thermoplasma_acidophilum) Q6L0S3_PITO0(_Picrophilus_torridus_DSM_9790) Q978X9_THEVO(_Thermoplasma_volcanium) Q9HL81_THEAC(_Thermo
HBG365857	ALR5370_PROTEIN;_HYPOTHETICAL_PROTEIN_CC3481;_HYPOTHETICAL_PROTEIN_RA0149;_PROBABLE_METHYLTRANSFERAS	(no_actino_in_family)	:FRAAL_PE865		:00569057 :00569079	Q9I5E0_PSAE0(_Pseudomonas_aeruginosa_PAO1) Q9A2S5_CACR0(_Caulobacter_crescentus_CB15) Q98JK6_RHILO(_Mesorhizobium_loti) Q930P6_RHIME(_Sinorhizobium_meliloti) Q8YLD2_ANASP(_Nostoc_sp._PCC_7120)
HBG379470	HYPOTHETICAL_ABC_TRANSPORTER_PERMEASE_PROTEIN_HI0355;_NITRATE_TRANSPORT_PERMEASE_PROTEIN_NRTB;_PUTAT	91	:FRAAL_PE1419			Q98K60_RHILO(_Mesorhizobium_loti) Q7CX64_AGRT5(_Agrobacterium_tumefaciens_str._C58) Q92MY9_RHIME(_Sinorhizobium_meliloti) Q92MY8_RHIME(_Sinorhizobium_meliloti) Q7CX63_AGRT5(_Agrobacterium_tumefaciens_str._C58) Q98K61_RHILO(_Mesorhizobium_loti)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG379470	HYPOTHETICAL_ABC_TRANSPORTER_PERMEASE_PROTEIN_HI0355;_NITRATE_TRANSPORT_PERMEASE_PROTEIN_NRTB;_PUTAT	76	:FRAAL_PE4252 :FRAAL_PE4356 :FRAAL_PE143		:00574094 :00570072	Q6MLI8_BDEBA(_Bdellovibrio_bacteriovorus) Q7UNX6_RHOBA(_Pirellula_sp.)
HBG381186	HYPOTHETICAL_UPF0245_PROTEIN_AF0229;_HYPOTHETICAL_UPF0245_PROTEIN_AQ_1922;_HYPOTHETICAL_UPF0245_PROT	(no_actino_in_family)		:00548929 :00549317		YJ22_AQAE0(_Aquifex_aeolicus_VF5) Q72EV7_DESVH(_Desulfovibrio_vulgaris_subsp._vulgaris_str.) Y229_ARFU0(_Archaeoglobus_fulgidus_DSM_4304) Q9LH99_ARATH(_Arabidopsis_thaliana)
HBG387677	AGR_PAT_263P;_BSL4960_PROTEIN;_HYPOTHETICAL_PROTEIN_DR1636;_HYPOTHETICAL_PROTEIN_YCZJ	(no_actino_in_family)	:FRAAL_PE6845	:00547720	:00573372	Q7VX03_BORPE(_Bordetella_pertussis) Q7WL96_BORBR(_Bordetella_bronchiseptica) Q7W7V6_BORPA(_Bordetella_parapertussis) Q6N648_RHPA0(_Rhodopseudomonas_palustris_CGA009) Q89KE7_BRJA0(_Bradyrhizobium_japonicum_USDA_110) Q7D3P8_AGRT5(_Agrobacterium_tumefaciens_str._C58) Q9RTW6_DEIRA(_Deinococcus_radiodurans) YCZJ_BASU0(_Bacillus_subtilis_subsp._subtilis_str.)
HBG388998	FLAVONOL_SYNTHASE-LIKE_PROTEIN;_GIBBERELLIN_OXIDASE-LIKE_PROTEIN;_HYOSCYAMINE_6-DIOXYGENASE_HYDROXYL_	(no_actino_in_family)	:FRAAL_PE5337			Q8EE01_SHON0(_Shewanella_oneidensis_MR-1) Q7MEA6_VIBVY(_Vibrio_vulnificus_YJ016) Q8D7A0_VIVU0(_Vibrio_vulnificus_CMCP6) Q87HU0_VIBPA(_Vibrio_parahaemolyticus) Q6LJ58_PHOPR(_Photobacterium_profundum) Q7P216_CHVI0(_Chromobacterium_violaceum_ATCC_12472) Q6MQL3_BDEBA(_Bdellovibrio_bacteriovorus) Q9ABM7_CACR0(_Caulobacter_crescentus_CB15) Q8P7S3_XACA0(_Xanthomonas_campestris_pv._campestris_str.) Q8PJ32_XAAX0(_Xanthomonas_axonopodis_pv._citri_str.) Q881B6_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q889J1_PSSY0(_Pseudomonas_syringae_pv._tomato_str.) Q9I6Y5_PSAE0(_Pseudomonas_aeruginosa_PAO1)
HBG000369	GLUTATHIONE-REGULATED POTASSIUM-EFFLUX SYSTEM	(no_actino_in_family)	FRAAL_PE6867	;00549337		Q984W6_RHILO(MESORHIZOBIUM_LOTI) Q92LV5_RHIME(SINORHIZOBIUM_MELILOTI) Q6N0L6_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009) Q9I4B4_PSAE0(PSEUDOMONAS_AERUGINOSA_PAO1)
HBG000450	DIHYDROXY-ACID DEHYDRATASE 1; DIHYDROXY-ACID DEHYDRATASE 2; DIHYDROXY-ACID DEHYDRATASE 3; DIHYDROXY-	(no_actino_in_family)			;00568966	Q8XPS8_RASO0(RALSTONIA_SOLANACEARUM_GMI1000) Q7P1R9_CHVI0(CHROMOBACTERIUM_VIOLACEUM_ATCC_12472) Q9JYX1_NEME0(NEISSERIA_MENINGITIDIS_MC58) Q9JTV9_NEME1(NEISSERIA_MENINGITIDIS_Z2491) EDD_HEPY0(HELICOBACTER_PYLORI_26695) EDD_HELPJ(HELICOBACTER_PYLORI_J99) Q989A4_RHILO(MESORHIZOBIUM_LOTI) Q8YCL7_BRME0(BRUCELLA_MELITENSIS_16M) Q7D150_AGRT5(AGROBACTERIUM_TUMEFACIENS_STR._C58) EDD_RHIME(SINORHIZOBIUM_MELILOTI) Q8PKU4_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q8P8U5_XACA0(XANTHOMONAS_CAMPESTRIS_PV._CAMPESTRIS_ST) Q9PEG6_XYFA0(XYLELLA_FASTIDIOSA_9A5C) Q87EG8_XYLFT(XYLELLA_FASTIDIOSA_TEMECULA1) Q9A6N2_CACR0(CAULOBACTER_CRESCENTUS_CB15) Q88P43_PSEPK(PSEUDOMONAS_PUTIDA_KT2440) Q887K4_PSSY0(PSEUDOMONAS_SYRINGAE_PV._TOMATO_STR._DC3) EDD_PSAE0(PSEUDOMONAS_AERUGINOSA_PAO1) EDD_ECO57(ESCHERICHIA_COLI_O157-H7) EDD_ESCO0(ESCHERICHIA_COLI_K12) EDD_ESCO1(ESCHERICHIA_COLI_O157-H7_EDL933) Q7C1B3_SHFL1(SHIGELLA_FLEXNERI_2A_STR._2457T) Q83R68_SHFL0(SHIGELLA_FLEXNERI_2A_STR._301) Q8FGR9_ECOL6(ESCHERICHIA_COLI_O6) Q83T46_SAEN0(SALMONELLA_ENTERICA_SUBSP._ENTERICA_SERO)
HBG002916	HYPOTHETICAL UPF0144 PROTEIN AQ_1732; HYPOTHETICAL UPF0144 PROTEIN BB0504; HYPOTHETICAL UPF0144 PROT	(no_actino_in_family)	FRAAL_PE5716	;00546809	;00572729	Q6MCB9_PASP0(CANDIDATUS_PROTOCHLAMYDIA_AMOEBOPHILA_UWE25) Q895I4_CLTE0(CLOSTRIDIUM_TETANI_E88) YI16_CLAC0(CLOSTRIDIUM_ACETOBUTYLICUM_ATCC_824) YG72_CLPE0(CLOSTRIDIUM_PERFRINGENS_STR._13) YD72_THTE0(THERMOANAEROBACTER_TENGCONGENSIS_MB4) YJ13_FUNU0(FUSOBACTERIUM_NUCLEATUM_SUBSP._NUCLEATUM) YA18_TRPA0(TREPONEMA_PALLIDUM_SUBSP._PALLIDUM_STR) Q73R54_TRDE0(TREPONEMA_DENTICOLA_ATCC_35405) Y504_BOBU0(BORRELIA_BURGDORFERI_B31) Q74E27_GESU0(GEOBACTER_SULFURREDUCENS_PCA) Q728D2_DESVH(DESULFOVIBRIO_VULGARIS_SUBSP._VULGARIS_S) Q6MNQ3_BDEBA(BDELLOVIBRIO_BACTERIOVORUS) Q6ML54_BDEBA(BDELLOVIBRIO_BACTERIOVORUS) Y8B7_BATH0(BACTEROIDES_THETAIOTAOMICRON_VPI-5482) Q7MX21_POGI0(PORPHYROMONAS_GINGIVALIS_W83) Y734_CHTE0(CHLOROBIUM_TEPIDUM_TLS) YI57_THMA0(THERMOTOGA_MARITIMA_MSB8) YH32_AQAE0(AQUIFEX_AEOLICUS_VF5) YG43_STRP8(STREPTOCOCCUS_PYOGENES_MGAS8232) YG33_STPY0(STREPTOCOCCUS_PYOGENES_M1_GAS) YG33_STPY1(STREPTOCOCCUS_PYOGENES_SSI-1) YG33_STRP3(STREPTOCOCCUS_PYOGENES_MGAS315) Y295_STAG0(STREPTOCOCCUS_AGALACTIAE_2603V/R) Y295_STAG1(STREPTOCOCCUS_AGALACTIAE_NEM316) Y475_STMU0(STREPTOCOCCUS_MUTANS_UA159) YH39_STPN0(STREPTOCOCCUS_PNEUMONIAE_TIGR4) YH39_ST

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG011804	PTS SYSTEM MANNITOL-SPECIFIC IIBC COMPONENT	(no_actino_in_family)			;00568971;00568972	PTMB_CLAC0(CLOSTRIDIUM_ACETOBUTYLICUM_ATCC_824) Q838N3_ENFA0(ENTEROCOCCUS_FAECALIS_V583) Q838M9_ENFA0(ENTEROCOCCUS_FAECALIS_V583) PTMB_STRR6(STREPTOCOCCUS_PNEUMONIAE_R6) PTMB_STPN0(STREPTOCOCCUS_PNEUMONIAE_TIGR4) PTMB_STMU0(STREPTOCOCCUS_MUTANS_UA159) Q8FE43_ECOL6(ESCHERICHIA_COLI_O6) PTYC_ECO57(ESCHERICHIA_COLI_O157-H7) PTYC_ESCO0(ESCHERICHIA_COLI_K12) PTYC_ESCO1(ESCHERICHIA_COLI_O157-H7_EDL933) Q87IV5_VIBPA(VIBRIO_PARAHAEMOLYTICUS)
HBG012485	DNA REPAIR PROTEIN RADC HOMOLOG; PUTATIVE RADC-LIKE PROTEIN VC0510; PUTATIVE RADC-LIKE PROTEIN VC178	(no_actino_in_family)			;00572854	RADC_SYNEL(SYNECHOCOCCUS_ELONGATUS) RADC_ANASP(NOSTOC_SP._PCC_7120) RADC_SYNY3(SYNECHOCYSTIS_SP._PCC_6803) Q7NFQ9_GLVI0(GLOEOBACTER_VIOLACEUS_PCC_7421) Q72Q17_LEIN1(LEPTOSPIRA_INTERROGANS_SEROVAR_COPENHAGE) Q8F655_LEIN0(LEPTOSPIRA_INTERROGANS_SEROVAR_LAI_STR)
HBG014478	UVRABC SYSTEM PROTEIN A	(no_actino_in_family)		;00548298	;00569413	Q7UMV4_RHOBA(PIRELLULA_SP.) Q749I5_GESU0(GEOBACTER_SULFURREDUCENS_PCA) Q8XR67_RASO0(RALSTONIA_SOLANACEARUM_GMI1000) Q7VU82_BORPE(BORDETELLA_PERTUSSIS) Q7W4R1_BORPA(BORDETELLA_PARAPERTUSSIS) Q7WG87_BORBR(BORDETELLA_BRONCHISEPTICA) Q6MAN4_PASP0(CANDIDATUS_PROTOCHLAMYDIA_AMOEBOPHILA_UWE25) Q822K6_CHCA0(CHLAMYDOPHILA_CAVIAE_GPIC) UVRA_CHPN0(CHLAMYDOPHILA_PNEUMONIAE_AR39) UVRA_CHPN1(CHLAMYDOPHILA_PNEUMONIAE_CWL029) UVRA_CHPN2(CHLAMYDOPHILA_PNEUMONIAE_J138) UVRA_CHPN3(CHLAMYDOPHILA_PNEUMONIAE_TW-183) UVRA_CHTR0(CHLAMYDIA_TRACHOMATIS_D/UW-3/CX) UVRA_CHLMU(CHLAMYDIA_MURIDARUM)
HBG017262	ALR0901 PROTEIN; CONSERVED HYPOTHETICAL PROTEIN; HYPOTHETICAL PROTEIN PA3683; HYPOTHETICAL PROTEIN S	(no_actino_in_family)	FRAAL_PE414	;00545457	;00569096	Q88MR1_PSEPK(PSEUDOMONAS_PUTIDA_KT2440) Q9HXV7_PSAE0(PSEUDOMONAS_AERUGINOSA_PAO1) Q886R5_PSSY0(PSEUDOMONAS_SYRINGAE_PV._TOMATO_STR._DC3) Q8DAN8_VIVU0(VIBRIO_VULNIFICUS_CMCP6) Q7MJ73_VIBVY(VIBRIO_VULNIFICUS_YJ016) Q87QV3_VIBPA(VIBRIO_PARAHAEMOLYTICUS) Q9KSU1_VICH0(VIBRIO_CHOLERAE_O1_BIOVAR_ELTOR_STR._N16) Q6LT54_PHOPR(PHOTOBACTERIUM_PROFUNDUM) YECE_ESCO0(ESCHERICHIA_COLI_K12) YECE_ECO57(ESCHERICHIA_COLI_O157-H7) YECE_ESCO1(ESCHERICHIA_COLI_O157-H7_EDL933) Q8FGQ7_ECOL6(ESCHERICHIA_COLI_O6) Q7AMT2_SALTI(SALMONELLA_TYPHI) Q7CQC6_SATY0(SALMONELLA_TYPHIMURIUM_LT2) Q8XFY8_SAEN0(SALMONELLA_ENTERICA_SUBSP._ENTERICA_SERO) Q74U50_YEPE2(YERSINIA_PESTIS_BIOVAR_MEDIEVALIS_STR._9) Q7CIB9_YEPE0(YERSINIA_PESTIS_KIM) Q8ZEV1_YEPE1(YERSINIA_PESTIS_CO92) Q7UN34_RHOBA(PIRELLULA_SP.) Q8YYE9_ANASP(NOSTOC_SP._PCC_7120)
HBG030412	BLUE COPPER OXIDASE CUEO PRECURSOR; PROTEIN SUFI PRECURSOR	(no_actino_in_family)	FRAAL_PE4284			Q8CQF6_STEP0(STAPHYLOCOCCUS_EPIDERMIDIS_ATCC_12228) Q88ZG5_LAPL0(LACTOBACILLUS_PLANTARUM_WCFS1) Q8YCF0_BRME0(BRUCELLA_MELITENSIS_16M) Q8FVW7_BRSU0(BRUCELLA_SUIS_1330) Q7N890_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDI) CUEO_YEPE2(YERSINIA_PESTIS_BIOVAR_MEDIEVALIS_STR._9) CUEO_YEPE1(YERSINIA_PESTIS_CO92) CUEO_YEPE0(YERSINIA_PESTIS_KIM) CUEO_SATY0(SALMONELLA_TYPHIMURIUM_LT2) CUEO_SALTI(SALMONELLA_TYPHI) CUEO_SAEN0(SALMONELLA_ENTERICA_SUBSP._ENTERICA_SERO) Q83ME9_SHFL0(SHIGELLA_FLEXNERI_2A_STR._301) Q7UDR7_SHFL1(SHIGELLA_FLEXNERI_2A_STR._2457T) Q8CWE2_ECOL6(ESCHERICHIA_COLI_O6) CUEO_ESCO0(ESCHERICHIA_COLI_K12) CUEO_ESCO1(ESCHERICHIA_COLI_O157-H7_EDL933) CUEO_ECO57(ESCHERICHIA_COLI_O157-H7) Q9CPE1_PASMU(PASTEURELLA_MULTOCIDA) Q7VPL4_HADU0(HAEMOPHILUS_DUCREYI_35000HP) Q7N0E3_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDI) Q74RN4_YEPE2(YERSINIA_PESTIS_BIOVAR_MEDIEVALIS_STR._9) Q8ZI41_YEPE1(YERSINIA_PESTIS_CO92) Q7CGI0_YEPE0(YERSINIA_PESTIS_KIM) SUFI_SATY0(SALMONELLA_TYPHIMURIUM_LT2) SUFI_SALTI(SALMONELLA_TYPHI) SUFI_SAEN0(SALMONELLA_ENTERICA_SUBSP._ENTERICA_SERO)
HBG056113	GLUTAMINE AMIDOTRANSFERASE-LIKE PROTEIN MTH191; GLUTAMINE AMIDOTRANSFERASE-LIKE PROTEIN GLXB	(no_actino_in_family)	FRAAL_PE1888			Q882P3_PSSY0(PSEUDOMONAS_SYRINGAE_PV._TOMATO_STR._DC3) Q7CUD4_AGRT5(AGROBACTERIUM_TUMEFACIENS_STR._C58) GLXB_RHIME(SINORHIZOBIUM_MELILOTI) Q986K8_RHILO(MESORHIZOBIUM_LOTI) Y191_METTH(METHANOTHERMOBACTER_THERMAUTOTROPHICUS_S) Q8TWH1_MEKA0(METHANOPYRUS_KANDLERI_AV19)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG113825	GLUTATHIONE SYNTHETASE; PUTATIVE GLUTATHIONE SYNTHASE; RIBOSOMAL PROTEIN	(no_actino_in_family)			;00570717	GSHB_SYNEL(SYNECHOCOCCUS_ELONGATUS) GSHB_ANASP(NOSTOC_SP._PCC_7120) GSHB_SYNY3(SYNECHOCYSTIS_SP._PCC_6803) GSHB_GLVI0(GLOEOBACTER_VIOLACEUS_PCC_7421) GSHB_PROMM(PROCHLOROCOCCUS_MARINUS_STR._MIT_9313) GSHB_SYNPX(SYNECHOCOCCUS_SP._WH_8102) GSHB_PRMA0(PROCHLOROCOCCUS_MARINUS_SUBSP._MARINUS_S) GSHB_PROMP(PROCHLOROCOCCUS_MARINUS_SUBSP._PASTORIS) GSHB_BRSU0(BRUCELLA_SUIS_1330) GSHB_BRME0(BRUCELLA_MELITENSIS_16M) GSHB_RHILO(MESORHIZOBIUM_LOTI) GSHB_RHIME(SINORHIZOBIUM_MELILOTI) GSHB_AGRT5(AGROBACTERIUM_TUMEFACIENS_STR._C58) GSHB_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009) GSHB_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) GSHB_CACR0(CAULOBACTER_CRESCENTUS_CB15) Q73I56_WOLPM(WOLBACHIA_SP._WMEL) GSHB_BORPE(BORDETELLA_PERTUSSIS) GSHB_BORBR(BORDETELLA_BRONCHISEPTICA) GSHB_BORPA(BORDETELLA_PARAPERTUSSIS) GSHB_RASO0(RALSTONIA_SOLANACEARUM_GMI1000) GSHB_NEME1(NEISSERIA_MENINGITIDIS_Z2491) GSHB_NEME0(NEISSERIA_MENINGITIDIS_MC58) GSHB_CHVI0(CHROMOBACTERIUM_VIOLACEUM_ATCC_12472) GSHB_NIEU0(NITROSOMONAS_EUROPAEA_ATCC_19718) GSHB_XACA0(XANTHOMONAS_CAMPESTRIS_PV._CAMPESTRIS_ST)
HBG224822	HYPOTHETICAL PROTEIN AF1688; MEMBRANE PROTEIN; PUTATIVE	(no_actino_in_family)	FRAAL_PE6820			Q72E09_DESVH(DESULFOVIBRIO_VULGARIS_SUBSP._VULGARIS_S) O28585_ARFU0(ARCHAEOGLOBUS_FULGIDUS_DSM_4304)
HBG225410	BLL7863 PROTEIN; HYDROLASE; HYPOTHETICAL PROTEIN; METALLO-BETA-LACTAMASE FAMILY PROTEIN; MLR3962 PRO	(no_actino_in_family)	FRAAL_PE6186		;00570362	Q6KQ76_BAAN0(BACILLUS_ANTHRACIS_STR._'AMES_ANCESTOR') Q81MV6_BACAA(BACILLUS_ANTHRACIS_STR._AMES) Q734H7_BACE0(BACILLUS_CEREUS_ATCC_10987) Q81AY3_BACCR(BACILLUS_CEREUS_ATCC_14579) Q81C53_BACCR(BACILLUS_CEREUS_ATCC_14579) O31787_BASU0(BACILLUS_SUBTILIS_SUBSP._SUBTILIS_STR._1) Q98F33_RHILO(MESORHIZOBIUM_LOTI) Q89CD4_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110)
HBG240156	BLR1748 PROTEIN; DUF269; HYPOTHETICAL PROTEIN FLGK; MLR5912 PROTEIN	(no_actino_in_family)	FRAAL_PE6806	;00546163	;00571389	Q79UV8_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q6N0Z5_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009) Q98AP1_RHILO(MESORHIZOBIUM_LOTI) Q44147_ANASP(NOSTOC_SP._PCC_7120) Q7MRG2_WOLSU(WOLINELLA_SUCCINOGENES)
HBG240283	HYPOTHETICAL PROTEIN XAC0505; HYPOTHETICAL PROTEIN XCC0491; SLL0543 PROTEIN	(no_actino_in_family)	FRAAL_PE1476	;00546836	;00571086	Q55401_SYNY3(SYNECHOCYSTIS_SP._PCC_6803) Q8PQ27_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q8PD54_XACA0(XANTHOMONAS_CAMPESTRIS_PV._CAMPESTRIS_ST)
HBG240327	ALR4836 PROTEIN; HYPOTHETICAL PROTEIN XAC1124; HYPOTHETICAL PROTEIN SLR0325; TLL1464 PROTEIN	(no_actino_in_family)			;00571281	Q8PNE7_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q82XI5_NIEU0(NITROSOMONAS_EUROPAEA_ATCC_19718) Q55530_SYNY3(SYNECHOCYSTIS_SP._PCC_6803) Q8YMU1_ANASP(NOSTOC_SP._PCC_7120) Q8DIW7_SYNEL(SYNECHOCOCCUS_ELONGATUS) Q7V417_PROMM(PROCHLOROCOCCUS_MARINUS_STR._MIT_9313) Q7U3R5_SYNPX(SYNECHOCOCCUS_SP._WH_8102)
HBG240498	HYPOTHETICAL PROTEIN; SIMILAR TO UNKNOWN PROTEIN; SLL0783 PROTEIN	(no_actino_in_family)	FRAAL_PE504 FRAAL_PE5959		;00572590	Q7N7C3_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDI) Q55950_SYNY3(SYNECHOCYSTIS_SP._PCC_6803) Q7U6B3_SYNPX(SYNECHOCOCCUS_SP._WH_8102)
HBG242000	HYPOTHETICAL PROTEIN YQJY	(no_actino_in_family)	FRAAL_PE3741		;00572708;00571056	Q8ETH4_OCEIH(OCEANOBACILLUS_IHEYENSIS) YQJY_BASU0(BACILLUS_SUBTILIS_SUBSP._SUBTILIS_STR._1) Q81BV7_BACCR(BACILLUS_CEREUS_ATCC_14579) Q735R2_BACE0(BACILLUS_CEREUS_ATCC_10987) Q6KR64_BAAN0(BACILLUS_ANTHRACIS_STR._'AMES_ANCESTOR') Q81NW4_BACAA(BACILLUS_ANTHRACIS_STR._AMES) Q7P0N4_CHVI0(CHROMOBACTERIUM_VIOLACEUM_ATCC_12472)
HBG242771	BACTERIOCIN O-METYLTRANSFERASE; PUTATIVE; MACROCIN O-METHYLTRANSFERASE	(no_actino_in_family)	FRAAL_PE2038	;00545881	;00569715	Q6KVS1_BAAN0(BACILLUS_ANTHRACIS_STR._'AMES_ANCESTOR') Q81TP7_BACAA(BACILLUS_ANTHRACIS_STR._AMES) Q73BU0_BACE0(BACILLUS_CEREUS_ATCC_10987) Q81GJ3_BACCR(BACILLUS_CEREUS_ATCC_14579) Q98ID4_RHILO(MESORHIZOBIUM_LOTI)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG245920	PUTATIVE THIH PROTEIN; THIAMIN BIOSYNTHESIS PROTEIN THIH; THIAMIN BIOSYNTHESIS PROTEIN; THIAZOLE MOI	(no_actino_in_family)	FRAAL_THIH	;00547833	;00569551	Q8AA16_BATH0(BACTEROIDES_THETAIOTAOMICRON_VPI-5482) Q7MT74_POGI0(PORPHYROMONAS_GINGIVALIS_W83) Q8DDL9_VIVU0(VIBRIO_VULNIFICUS_CMCP6) Q7MGM8_VIBVY(VIBRIO_VULNIFICUS_YJ016) Q87KF5_VIBPA(VIBRIO_PARAHAEMOLYTICUS) Q6LVX6_PHOPR(PHOTOBACTERIUM_PROFUNDUM) Q9KVS3_VICH0(VIBRIO_CHOLERAE_O1_BIOVAR_ELTOR_STR._N16) Q8EEE2_SHON0(SHEWANELLA_ONEIDENSIS_MR-1) Q8D250_WIGBR(WIGGLESWORTHIA_GLOSSINIDIA_ENDOSYMBIONT) Q7N967_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDI) Q8ZAP8_YEPE1(YERSINIA_PESTIS_CO92) Q74RE1_YEPE2(YERSINIA_PESTIS_BIOVAR_MEDIEVALIS_STR._9) Q8D1H2_YEPE0(YERSINIA_PESTIS_KIM) THIH_SATY0(SALMONELLA_TYPHIMURIUM_LT2) Q7C6P3_SAEN0(SALMONELLA_ENTERICA_SUBSP._ENTERICA_SERO) Q8Z321_SALTI(SALMONELLA_TYPHI) Q8FB82_ECOL6(ESCHERICHIA_COLI_O6) Q7A959_ECO57(ESCHERICHIA_COLI_O157-H7) Q8X6Z3_ESCO1(ESCHERICHIA_COLI_O157-H7_EDL933) Q7BZH5_SHFL1(SHIGELLA_FLEXNERI_2A_STR._2457T) Q83PC1_SHFL0(SHIGELLA_FLEXNERI_2A_STR._301) THIH_ESCO0(ESCHERICHIA_COLI_K12) Q8KEJ2_CHTE0(CHLOROBIUM_TEPIDUM_TLS)
HBG250727	POLYSACCHARIDE BIOSYNTHESIS PROTEIN; PUTATIVE AMINOTRANSFERASE	(no_actino_in_family)	FRAAL_PE4640			Q6N4I7_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009) Q8EEB1_SHON0(SHEWANELLA_ONEIDENSIS_MR-1)
HBG251167	HELICASE PROTEIN; POSSIBLE ATP-DEPENDENT RNA HELICASE	(no_actino_in_family)		;00548584		Q6N7N9_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009) Q92SA9_RHIME(SINORHIZOBIUM_MELILOTI)
HBG251703	HUPU PROTEIN; HYDROGENASE SMALL SUBUNIT; [NIFE] UPTAKE HYDROGENASE SMALL SUBUNIT; UPTAKE HYDROGENASE	(no_actino_in_family)	FRAAL_HUPS2	;00549673		Q7A2H6_ANASP(NOSTOC_SP._PCC_7120) O66987_AQAE0(AQUIFEX_AEOLICUS_VF5) Q79U56_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q6NB69_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009)
HBG255619	AGR_C_4128P; ALL1162 PROTEIN; BH2241 PROTEIN; BLR4796 PROTEIN; CONSERVED HYPOTHETICAL PROTEIN; ENZYM	(no_actino_in_family)		;00548686		Q72DG6_DESVH(DESULFOVIBRIO_VULGARIS_SUBSP._VULGARIS_S) Q92N45_RHIME(SINORHIZOBIUM_MELILOTI) Q7CXG8_AGRT5(AGROBACTERIUM_TUMEFACIENS_STR._C58) Q9A515_CACR0(CAULOBACTER_CRESCENTUS_CB15) Q89KV4_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q6N5U5_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009) Q983K4_RHILO(MESORHIZOBIUM_LOTI) Q8YHJ4_BRME0(BRUCELLA_MELITENSIS_16M) Q8G0B6_BRSU0(BRUCELLA_SUIS_1330) Q6MRJ2_BDEBA(BDELLOVIBRIO_BACTERIOVORUS) Q9ZJJ9_HELPJ(HELICOBACTER_PYLORI_J99) O25959_HEPY0(HELICOBACTER_PYLORI_26695) Q9K161_NEME0(NEISSERIA_MENINGITIDIS_MC58) Q9JSR7_NEME1(NEISSERIA_MENINGITIDIS_Z2491) Q8A7G0_BATH0(BACTEROIDES_THETAIOTAOMICRON_VPI-5482) Q7MUX8_POGI0(PORPHYROMONAS_GINGIVALIS_W83) Q8DUL0_STMU0(STREPTOCOCCUS_MUTANS_UA159) Q97P67_STPN0(STREPTOCOCCUS_PNEUMONIAE_TIGR4) Q8DNP8_STRR6(STREPTOCOCCUS_PNEUMONIAE_R6) Q8CTG5_STEP0(STAPHYLOCOCCUS_EPIDERMIDIS_ATCC_12228) Q7A1H9_STAAW(STAPHYLOCOCCUS_AUREUS_SUBSP._AUREUS_MW2) Q7A6T4_STAAN(STAPHYLOCOCCUS_AUREUS_SUBSP._AUREUS_N315) Q99VP5_STAAM(STAPHYLOCOCCUS_AUREUS_SUBSP._AUREUS_MU50) O31678_BASU0(BACILLUS_SUBTILIS_SUBSP._SUBTILIS_STR._1) Q9KAP6_BAHA0(BACILLUS_HALODURANS_C-125) C
HBG255641	HYPOTHETICAL PROTEIN MJ1651	(no_actino_in_family)	FRAAL_PE4509	;00548888	;00573837	Q72H50_THTH0(THERMUS_THERMOPHILUS_HB27) Q72DL4_DESVH(DESULFOVIBRIO_VULGARIS_SUBSP._VULGARIS_S)
HBG255898	HYPOTHETICAL PROTEIN YFAU	(no_actino_in_family)	FRAAL_GARL			Q7D052_AGRT5(AGROBACTERIUM_TUMEFACIENS_STR._C58) Q92TE1_RHIME(SINORHIZOBIUM_MELILOTI) Q8YDN6_BRME0(BRUCELLA_MELITENSIS_16M) Q8FUR3_BRSU0(BRUCELLA_SUIS_1330) Q8ZQ47_SATY0(SALMONELLA_TYPHIMURIUM_LT2) Q8Z7P9_SALTI(SALMONELLA_TYPHI) Q7C982_SAEN0(SALMONELLA_ENTERICA_SUBSP._ENTERICA_SERO) Q83P19_SHFL0(SHIGELLA_FLEXNERI_2A_STR._301) Q7BYH4_SHFL1(SHIGELLA_FLEXNERI_2A_STR._2457T) Q7N7W9_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDI) Q74US3_YEPE2(YERSINIA_PESTIS_BIOVAR_MEDIEVALIS_STR._9) Q8ZFE7_YEPE1(YERSINIA_PESTIS_CO92) Q8D095_YEPE0(YERSINIA_PESTIS_KIM) Q9HWQ3_PSAE0(PSEUDOMONAS_AERUGINOSA_PAO1) Q9CKS1_PASMU(PASTEURELLA_MULTOCIDA) Q9RYM1_DEIRA(DEINOCOCCUS_RADIODURANS) Q7WPE4_BORBR(BORDETELLA_BRONCHISEPTICA) Q8XZ15_RASO0(RALSTONIA_SOLANACEARUM_GMI1000) Q8XPM5_RASO0(RALSTONIA_SOLANACEARUM_GMI1000)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG257156	BH0339 PROTEIN; CRISPR-ASSOCIATED PROTEIN; TM1801 FAMILY; CONSERVED HYPOTHETICAL PROTEIN; HYPOTHETIC	(no_actino_in_family)		;00549408		Q746C0_THTH0(THERMUS_THERMOPHILUS_HB27) Q8KDC2_CHTE0(CHLOROBIUM_TEPIDUM_TLS) Q8F872_LEIN0(LEPTOSPIRA_INTERROGANS_SEROVAR_LAI_STR) Q72NC0_LEIN1(LEPTOSPIRA_INTERROGANS_SEROVAR_COPENHAGE) Q72WF7_DESVH(DESULFOVIBRIO_VULGARIS_SUBSP._VULGARIS_S) Q8PFY2_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q7NYP6_CHVI0(CHROMOBACTERIUM_VIOLACEUM_ATCC_12472) Q99YS5_STPY0(STREPTOCOCCUS_PYOGENES_M1_GAS) Q8DSM0_STMU0(STREPTOCOCCUS_MUTANS_UA159) Q9KFY1_BAHA0(BACILLUS_HALODURANS_C-125)
HBG257158	CRISPR-ASSOCIATED PROTEIN; CT1134 FAMILY; CONSERVED HYPOTHETICAL PROTEIN; HYPOTHETICAL CONSERVED PRO	(no_actino_in_family)		;00549406		Q8KDC0_CHTE0(CHLOROBIUM_TEPIDUM_TLS) Q72NC2_LEIN1(LEPTOSPIRA_INTERROGANS_SEROVAR_COPENHAGE) Q8F870_LEIN0(LEPTOSPIRA_INTERROGANS_SEROVAR_LAI_STR) Q72WF9_DESVH(DESULFOVIBRIO_VULGARIS_SUBSP._VULGARIS_S) Q8PFY4_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q7NYP8_CHVI0(CHROMOBACTERIUM_VIOLACEUM_ATCC_12472) Q746C2_THTH0(THERMUS_THERMOPHILUS_HB27)
HBG259002	CONSERVED PROTEIN; HYPOTHETICAL PROTEIN ECS1346; HYPOTHETICAL PROTEIN YPO0290; Z1606 PROTEIN	(no_actino_in_family)	FRAAL_PE5887	;00546945	;00573072	Q891X7_CLTE0(CLOSTRIDIUM_TETANI_E88) Q72YC8_BACE0(BACILLUS_CEREUS_ATCC_10987) Q8X9R1_ESCO1(ESCHERICHIA_COLI_O157-H7_EDL933) Q8X2L2_ECO57(ESCHERICHIA_COLI_O157-H7) Q74XH6_YEPE2(YERSINIA_PESTIS_BIOVAR_MEDIEVALIS_STR._9) Q7CKQ6_YEPE0(YERSINIA_PESTIS_KIM) Q8ZJ38_YEPE1(YERSINIA_PESTIS_CO92) Q888S9_PSSY0(PSEUDOMONAS_SYRINGAE_PV._TOMATO_STR._DC3)
HBG263297	TRANSCRIPTIONAL REGULATOR; SIR2 FAMILY	(no_actino_in_family)	FRAAL_PE3259		;00571686	Q748C0_GESU0(GEOBACTER_SULFURREDUCENS_PCA)
HBG268243	BLL6088 PROTEIN; HYPOTHETICAL PROTEIN	(no_actino_in_family)	FRAAL_PE6168			Q88FY1_PSEPK(PSEUDOMONAS_PUTIDA_KT2440) Q7TT89_BORPA(BORDETELLA_PARAPERTUSSIS) Q79G63_BORBR(BORDETELLA_BRONCHISEPTICA) Q7TTJ1_BORPE(BORDETELLA_PERTUSSIS) Q89HA6_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110)
HBG273045	AGR_PAT_360P; MLR6716 PROTEIN	(no_actino_in_family)	FRAAL_PE5558	;00548982	;00572344	Q988J5_RHILO(MESORHIZOBIUM_LOTI) Q7D3I7_AGRT5(AGROBACTERIUM_TUMEFACIENS_STR._C58)
HBG277266	SIMILARITIES WITH AAA SUPERFAMILY ATPASE; SIMILAR TO PUTATIVE AAA-FAMILY ATPASE GENE	(no_actino_in_family)	FRAAL_PE817			Q7MZ46_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDI) Q7N8Y9_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDI)
HBG277378	HYPOTHETICAL PROTEIN; SIMILAR TO PUTATIVE TRANSFERASE PROTEIN; TRANSFERASE PROTEIN	(no_actino_in_family)	FRAAL_PE4659			Q8XQJ2_RASO0(RALSTONIA_SOLANACEARUM_GMI1000) Q880F6_PSSY0(PSEUDOMONAS_SYRINGAE_PV._TOMATO_STR._DC3) Q7N025_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDI)
HBG278023	BLL2692 PROTEIN; GLYCOSYLTRANSFERASE; PUTATIVE; SIMILAR TO PROBABLE GLYCOSYLTRANSFERASE	(no_actino_in_family)	FRAAL_PE1959		;00571247	Q9RYI3_DEIRA(DEINOCOCCUS_RADIODURANS) Q89RS0_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q7N615_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDI) Q7N614_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDI)
HBG279287	ALL4478 PROTEIN; CONSERVED HYPOTHETICAL PROTEIN; GLR3419 PROTEIN; HYPOTHETICAL CONSERVED PROTEIN; HY	(no_actino_in_family)	FRAAL_PE4142		;00572787	Q7UEE6_RHOBA(PIRELLULA_SP.) Q7UJZ3_RHOBA(PIRELLULA_SP.) Q72MY2_LEIN1(LEPTOSPIRA_INTERROGANS_SEROVAR_COPENHAGE) Q8F8Q9_LEIN0(LEPTOSPIRA_INTERROGANS_SEROVAR_LAI_STR) Q7NFV5_GLVI0(GLOEOBACTER_VIOLACEUS_PCC_7421) Q8YNT3_ANASP(NOSTOC_SP._PCC_7120) Q8NVF7_STAAW(STAPHYLOCOCCUS_AUREUS_SUBSP._AUREUS_MW2) Q7A4D1_STAAN(STAPHYLOCOCCUS_AUREUS_SUBSP._AUREUS_N315) Q99SC4_STAAM(STAPHYLOCOCCUS_AUREUS_SUBSP._AUREUS_MU50) Q8CRM0_STEP0(STAPHYLOCOCCUS_EPIDERMIDIS_ATCC_12228) Q8ELP8_OCEIH(OCEANOBACILLUS_IHEYENSIS)
HBG279344	ALL3942 PROTEIN; CONSERVED HYPOTHETICAL PROTEIN; GLL3290 PROTEIN; HYPOTHETICAL PROTEIN	(no_actino_in_family)		;00548577		Q7NG83_GLVI0(GLOEOBACTER_VIOLACEUS_PCC_7421) Q8YQ92_ANASP(NOSTOC_SP._PCC_7120) Q8F5W1_LEIN0(LEPTOSPIRA_INTERROGANS_SEROVAR_LAI_STR) Q72Q97_LEIN1(LEPTOSPIRA_INTERROGANS_SEROVAR_COPENHAGE)
HBG279626	ALL1996 PROTEIN; ALR0892 PROTEIN; GLL2422 PROTEIN	(no_actino_in_family)			;00568011	Q8YYF8_ANASP(NOSTOC_SP._PCC_7120) Q8YVI2_ANASP(NOSTOC_SP._PCC_7120) Q7NHW3_GLVI0(GLOEOBACTER_VIOLACEUS_PCC_7421)
HBG280614	HYPOTHETICAL PROTEIN; PUTATIVE LIPOPROTEIN	(no_actino_in_family)	FRAAL_PE3693		;00569432	Q8XZX2_RASO0(RALSTONIA_SOLANACEARUM_GMI1000) Q7NQ19_CHVI0(CHROMOBACTERIUM_VIOLACEUM_ATCC_12472)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG282767	HYDROLASE; ISOCHORISMATASE FAMILY; PUTATIVE ISOCHORISMATASE	(no_actino_in_family)	FRAAL_PE6163			Q7TTE5_BORBR(BORDETELLA_BRONCHISEPTICA) Q7TTC8_BORPA(BORDETELLA_PARAPERTUSSIS) Q7TTD2_BORPE(BORDETELLA_PERTUSSIS) Q88FY5_PSEPK(PSEUDOMONAS_PUTIDA_KT2440) Q7WJL0_BORBR(BORDETELLA_BRONCHISEPTICA) Q7WAG3_BORPA(BORDETELLA_PARAPERTUSSIS)
HBG283588	HYPOTHETICAL PROTEIN	(no_actino_in_family)		;00549553	;00572351	Q87U59_PSSY0(PSEUDOMONAS_SYRINGAE_PV._TOMATO_STR._DC3) Q7U6C0_SYNPX(SYNECHOCOCCUS_SP._WH_8102)
HBG284413	RESTRICTION MODIFICATION SYSTEM S CHAIN HOMOLOG; TYPE I RESTRICTION MODIFICATION ENZYME; S SUBUNIT	(no_actino_in_family)	FRAAL_PE3318			Q8TN78_MEAC0(METHANOSARCINA_ACETIVORANS_C2A) Q7UE18_RHOBA(PIRELLULA_SP.) Q7UE33_RHOBA(PIRELLULA_SP.)
HBG290308	POSSIBLE NA+/H+ ANTIPORTER; CPA1 FAMILY	(no_actino_in_family)	FRAAL_PE5407			Q8TMM3_MEAC0(METHANOSARCINA_ACETIVORANS_C2A) Q8PVS1_MEMA0(METHANOSARCINA_MAZEI_GO1) Q7V831_PROMM(PROCHLOROCOCCUS_MARINUS_STR._MIT_9313)
HBG292084	HYPOTHETICAL PROTEIN RSC2803	(no_actino_in_family)	FRAAL_PE2408			Q7WN68_BORBR(BORDETELLA_BRONCHISEPTICA) Q7W0U8_BORPA(BORDETELLA_PARAPERTUSSIS) Q7VV37_BORPE(BORDETELLA_PERTUSSIS) Q7WCI3_BORPA(BORDETELLA_PARAPERTUSSIS) Q7WQI9_BORBR(BORDETELLA_BRONCHISEPTICA) Q7VZV3_BORPE(BORDETELLA_PERTUSSIS) Q8XVM7_RASO0(RALSTONIA_SOLANACEARUM_GMI1000)
HBG292168	AGR_C_4219P; HYPOTHETICAL PROTEIN SMC01522; POSSIBLE TRANSCRIPTIONAL REGULATOR; TETR FAMILY; PROBABL	(no_actino_in_family)			;00571522	Q55363_SYNY3(SYNECHOCYSTIS_SP._PCC_6803) Q81GX2_BACCR(BACILLUS_CEREUS_ATCC_14579) Q8CQC0_STEP0(STAPHYLOCOCCUS_EPIDERMIDIS_ATCC_12228) Q8CUW2_OCEIH(OCEANOBACILLUS_IHEYENSIS) Q89MG6_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q7CXC3_AGRT5(AGROBACTERIUM_TUMEFACIENS_STR._C58) Q6N2Z2_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009) Q92N23_RHIME(SINORHIZOBIUM_MELILOTI) Q7W9N5_BORPA(BORDETELLA_PARAPERTUSSIS) Q7VWE0_BORPE(BORDETELLA_PERTUSSIS) Q7WH22_BORBR(BORDETELLA_BRONCHISEPTICA) Q9I461_PSAE0(PSEUDOMONAS_AERUGINOSA_PAO1) Q7VZY8_BORPE(BORDETELLA_PERTUSSIS) Q7W4I0_BORPA(BORDETELLA_PARAPERTUSSIS) Q7WG03_BORBR(BORDETELLA_BRONCHISEPTICA)
HBG292212	BLR3400 PROTEIN; PUTATIVE DIOXYGENASE HYDROXYLASE COMPONENT	(no_actino_in_family)	FRAAL_HCAE			Q7WHQ8_BORBR(BORDETELLA_BRONCHISEPTICA) Q7W6S4_BORPA(BORDETELLA_PARAPERTUSSIS) Q7VWV0_BORPE(BORDETELLA_PERTUSSIS) Q89PT0_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110)
HBG298989	HYPOTHETICAL PROTEIN; PREVENT-HOST-DEATH FAMILY PROTEIN	(no_actino_in_family)		;00546025		Q82VL3_NIEU0(NITROSOMONAS_EUROPAEA_ATCC_19718) Q884T6_PSSY0(PSEUDOMONAS_SYRINGAE_PV._TOMATO_STR._DC3)
HBG299248	CYTIDINE AND DEOXYCYTIDYLATE DEAMINASE FAMILY PROTEIN; CYTIDINE AND DEOXYCYTIDYLATE DEAMINASE ZINC-B	(no_actino_in_family)	FRAAL_PE417			Q8PNZ8_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q8PCB7_XACA0(XANTHOMONAS_CAMPESTRIS_PV._CAMPESTRIS_ST) Q8KDQ1_CHTE0(CHLOROBIUM_TEPIDUM_TLS) Q82Y41_NIEU0(NITROSOMONAS_EUROPAEA_ATCC_19718)
HBG304863	GLR3050 PROTEIN; HYPOTHETICAL PROTEIN XAC2124; HYPOTHETICAL PROTEIN XCC1644; METHYLTRANSFERASE; PUTA	(no_actino_in_family)	FRAAL_PE305			Q88HU6_PSEPK(PSEUDOMONAS_PUTIDA_KT2440) Q880M1_PSSY0(PSEUDOMONAS_SYRINGAE_PV._TOMATO_STR._DC3) Q8PKP5_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q8PA49_XACA0(XANTHOMONAS_CAMPESTRIS_PV._CAMPESTRIS_ST) Q9RZ23_DEIRA(DEINOCOCCUS_RADIODURANS) Q7NCC9_GLVI0(GLOEOBACTER_VIOLACEUS_PCC_7421)
HBG310614	BLL2138 PROTEIN; INTEGRASE/RECOMBINASE; PUTATIVE INTEGRASE/RECOMBINASE	(no_actino_in_family)			;00569777	Q925Y2_RHIME(SINORHIZOBIUM_MELILOTI) Q89TA1_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q981F4_RHILO(MESORHIZOBIUM_LOTI)
HBG311732	EMB|CAB66926.1; HYPOTHETICAL PROTEIN T16K5.230	(no_actino_in_family)			;00569788;00567947	Q8A1H6_BATH0(BACTEROIDES_THETAIOTAOMICRON_VPI-5482) Q8A5X4_BATH0(BACTEROIDES_THETAIOTAOMICRON_VPI-5482) Q9M2X0_ARATH(ARABIDOPSIS_THALIANA) Q9FJW9_ARATH(ARABIDOPSIS_THALIANA)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG338640	AGR_PAT_755P; CELL CYCLE HISTIDINE KINASE CCKA; FIXL-RELATED HISTIDINE KINASE; HISTIDINE KINASE/RESP	(no_actino_in_family)	FRAAL_PE767 FRAAL_PE2440	;00549857	;00572601;00567018	Q92Z74_RHIME(SINORHIZOBIUM_MELILOTI) Q89HT3_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q89FI4_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q6N2W4_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009) Q87X95_PSSY0(PSEUDOMONAS_SYRINGAE_PV._TOMATO_STR._DC3) Q8PHI1_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q8PHG4_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q8P912_XACA0(XANTHOMONAS_CAMPESTRIS_PV._CAMPESTRIS_ST) Q8YVK7_ANASP(NOSTOC_SP._PCC_7120) Q7NFU0_GLVI0(GLOEOBACTER_VIOLACEUS_PCC_7421) Q8PMZ2_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q8PBD6_XACA0(XANTHOMONAS_CAMPESTRIS_PV._CAMPESTRIS_ST) Q9I1T8_PSAE0(PSEUDOMONAS_AERUGINOSA_PAO1) Q88G90_PSEPK(PSEUDOMONAS_PUTIDA_KT2440) Q89T74_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q89JL9_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q88HD1_PSEPK(PSEUDOMONAS_PUTIDA_KT2440) Q882B2_PSSY0(PSEUDOMONAS_SYRINGAE_PV._TOMATO_STR._DC3) Q8PLW7_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q8PA38_XACA0(XANTHOMONAS_CAMPESTRIS_PV._CAMPESTRIS_ST) Q8PN01_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30)
HBG338871	POSSIBLE TRANSCRIPTIONAL REGULATOR; TETR FAMILY; PROBABLE TRANSCRIPTIONAL REGULATOR; TRANSCRIPTIONAL	(no_actino_in_family)			;00572597	Q9HX43_PSAE0(PSEUDOMONAS_AERUGINOSA_PAO1) Q8PQQ6_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q8PDT8_XACA0(XANTHOMONAS_CAMPESTRIS_PV._CAMPESTRIS_ST) Q89EE6_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q6NAQ2_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009)
HBG339206	HYPOTHETICAL PROTEIN PA2734; HYPOTHETICAL PROTEIN VC1768; HYPOTHETICAL TYPE I RESTRICTION-MODIFICATI	(no_actino_in_family)		;00546296		Q8PSU8_MEMA0(METHANOSARCINA_MAZEI_GO1) Q6ZE86_SYNY3(SYNECHOCYSTIS_SP._PCC_6803) Q9KR75_VICH0(VIBRIO_CHOLERAE_O1_BIOVAR_ELTOR_STR._N16) Q6LTT0_PHOPR(PHOTOBACTERIUM_PROFUNDUM) Q9P9Z8_XYFA0(XYLELLA_FASTIDIOSA_9A5C) Q879W9_XYLFT(XYLELLA_FASTIDIOSA_TEMECULA1) Q9I0A8_PSAE0(PSEUDOMONAS_AERUGINOSA_PAO1) Q8PIJ8_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30)
HBG339655	TRANSPOSASE	(no_actino_in_family)			;00567476	Q8PSS8_MEMA0(METHANOSARCINA_MAZEI_GO1) Q8Q0G6_MEMA0(METHANOSARCINA_MAZEI_GO1) Q8PU92_MEMA0(METHANOSARCINA_MAZEI_GO1) Q8PRR2_MEMA0(METHANOSARCINA_MAZEI_GO1)
HBG339675	HYPOTHETICAL PROTEIN MA1921; HYPOTHETICAL PROTEIN MA1925; HYPOTHETICAL PROTEIN MM3350; SIMILAR TO UN	(no_actino_in_family)			;00571791	Q8TPJ0_MEAC0(METHANOSARCINA_ACETIVORANS_C2A) Q8TPI6_MEAC0(METHANOSARCINA_ACETIVORANS_C2A) Q8PRU3_MEMA0(METHANOSARCINA_MAZEI_GO1) Q7N8F9_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDI)
HBG339854	PREDICTED PROTEIN; TRANSPOSASE	(no_actino_in_family)			;00567618	Q8TSF9_MEAC0(METHANOSARCINA_ACETIVORANS_C2A) Q8PU93_MEMA0(METHANOSARCINA_MAZEI_GO1) Q8TLU0_MEAC0(METHANOSARCINA_ACETIVORANS_C2A) Q8Q0G5_MEMA0(METHANOSARCINA_MAZEI_GO1) Q8PSS7_MEMA0(METHANOSARCINA_MAZEI_GO1) Q8PRQ9_MEMA0(METHANOSARCINA_MAZEI_GO1)
HBG340899	HYPOTHETICAL PROTEIN MA1634; HYPOTHETICAL PROTEIN MA2137; METHYLTRANSFERASE	(no_actino_in_family)		;00548917	;00571266	Q8TNZ0_MEAC0(METHANOSARCINA_ACETIVORANS_C2A) Q8Q0W5_MEMA0(METHANOSARCINA_MAZEI_GO1) Q8TQB4_MEAC0(METHANOSARCINA_ACETIVORANS_C2A)
HBG345256	HYPOTHETICAL PROTEIN DR1127; KLAA PROTEIN	(no_actino_in_family)		;00548359	;00571675	Q9RVA2_DEIRA(DEINOCOCCUS_RADIODURANS) Q8TN38_MEAC0(METHANOSARCINA_ACETIVORANS_C2A)
HBG345915	5-FORMYLTETRAHYDROFOLATE CYCLO-LIGASE; BH0836 PROTEIN; HYPOTHETICAL PROTEIN APE0157; HYPOTHETICAL PR	(no_actino_in_family)		;00548066		Q9HP36_HALN1(HALOBACTERIUM_SP._NRC-1) Q9YFU3_AEPE0(AEROPYRUM_PERNIX_K1) Q8TUZ6_MEKA0(METHANOPYRUS_KANDLERI_AV19) Q97ZW6_SULSO(SULFOLOBUS_SOLFATARICUS) Q976L8_SUTO0(SULFOLOBUS_TOKODAII_STR._7) Q8ZUZ3_PYRAE(PYROBACULUM_AEROPHILUM) Q9H954_HUMAN(HOMO_SAPIENS) Q9H878_HUMAN(HOMO_SAPIENS) Q8BUM0_MOUSE(MUS_MUSCULUS) Q9SRE0_ARATH(ARABIDOPSIS_THALIANA) Q9KEL6_BAHA0(BACILLUS_HALODURANS_C-125)
HBG350748	HEME BIOSYNTHESIS PROTEIN; HYPOTHETICAL PROTEIN RSC1728	(no_actino_in_family)	FRAAL_PE3004			Q8YVE9_ANASP(NOSTOC_SP._PCC_7120) Q8XYN0_RASO0(RALSTONIA_SOLANACEARUM_GMI1000)
HBG352301	ALL0217 PROTEIN; ALR4619 PROTEIN; F9K20.18 PROTEI	(no_actino_in_family)	FRAAL_PE4775		;00573569	Q8YNE7_ANASP(NOSTOC_SP._PCC_7120) Q8Z081_ANASP(NOSTOC_SP._PCC_7120) Q9ZV96_ARATH(ARABIDOPSIS_THALIANA)
HBG379794	HYPOTHETICAL PROTEIN VCA0880; HYPOTHETICAL PROTEIN VCA0881	(no_actino_in_family)			;00570596	Q9KL67_VICH0(VIBRIO_CHOLERAE_O1_BIOVAR_ELTOR_STR._N16) Q9KL66_VICH0(VIBRIO_CHOLERAE_O1_BIOVAR_ELTOR_STR._N16)

Annexe B
Résultats de la détection d'éventuels transferts horizontaux chez *Frankia*, par l'analyse liée à l'absence de gènes chez *Streptomyces*

famille hogenom	annotations hogenom	sequences from AInl	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG000160	; 00546194		;00546194		Q7TXC2_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) CSTA_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) CSTA_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73V96_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q8NSL6_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q6NFT0_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG002278	; 00569730; Hypothetical 56.5 kDa protein in HXT8 5'region and in HXT17-COS10 intergenic region; Hypot			;00569730	Q8FMX1_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG002667	; 00548274; 00571008; Biotin carboxyl carrier protein of acetyl-CoA carboxylase; Biotinylated protein	FRAAL_PE6004	;00548274	;00571008	BTB7_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q73UP4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) BTB7_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) BTB7_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) BTB7_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG004408	; 00567687; 00567751; 00569015; 00569348; 00569844; 00570174; 00570371; 00572676; Hypothetical 35.9 kD			;00572676	Q73V74_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG004408	; 00567687; 00567751; 00569015; 00569348; 00569844; 00570174; 00570371; 00572676; Hypothetical 35.9 kD			;00570174; 00570371;00569844	Q742W7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG006411	; 00549469; 00571062; Glutamate racemase 1; Glutamate racemase 2; Glutamate racemase 3	FRAAL_MURI	;00549469	;00571062	MURI_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) MURI_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) MURI_MYLE0(MYCOBACTERIUM_LEPRAE_TN) MURI_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) MURI_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) MURI_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73X85_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q6NFN4_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG007288	; 00547646; 00570238; Hypothetical protein C106.17c in chromosome II; Probable homoserine O-acetyltran	FRAAL_METX	;00547646	;00570238	METX_MYLE0(MYCOBACTERIUM_LEPRAE_TN) METX_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) METX_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) METX_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73UB0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) METX_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) METX_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NIZ3_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG012784	; 00572008; Hypothetical protein Rv2575/MT2651/Mb2605; Hypothetical protein ypfJ	FRAAL_PE5600		;00572008	Q8NQ18_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FPM4_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NGZ3_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) YP75_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YP75_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YP75_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q741N0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG013177	; 00549049; Putative colanic acid biosynthesis acetyltransferase wcaB	FRAAL_CYSE	;00549049		Q73Y34_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O32979_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q7D7B1_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) P95231_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TYT0_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG014768	; 00545893; 00569508; 00569509	FRAAL_SYGA	;00545893	;00569508; 00569509	Q83I31_TROW8(TROPHERYMA_WHIPPLEI_TW08/27) Q83G33_TROWT(TROPHERYMA_WHIPPLEI_STR._TWIST)
HBG015420	; 00549674; 00571318; Hydrogenase-2 large chain precursor; Quinone-reactive Ni/Fe-hydrogenase large ch			;00571318	Q6NIU3_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG017948	; 00548946; 00549753; 00571476; Probable methylisocitrate lyase 1; Probable methylisocitrate lyase 2	FRAAL_PRPB		;00571476	Q744P5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) PRB2_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) PRB1_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FRP2_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG017972	; 00569727; 00569860; Beta-glucuronidase precursor			;00569860; 00569727	Q8FMX0_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)

famille hogenom	annotations hogenom	sequences from AIni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG030443	; 00545553; 00571696; 00572801; Asparaginyl-tRNA synthetase 1; Asparaginyl-tRNA synthetase 2; Asparagi		;00545553	;00571696	Q8NM92_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FMH1_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NFA7_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) SYK1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) SYK1_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) SYK1_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q743Y2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) SYK_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q8G3V0_BILO0(BIFIDOBACTERIUM_LONGUM_NCC2705)
HBG031619	; 00547301; 00548086; 00548757; 00567027; 00567248; 00569776; 00570377; 00571642; 00572453; 00573021;			;00571642; 00567248	XERC_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) XERC_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) XERC_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) XERC_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q73VQ5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG031619	; 00547301; 00548086; 00548757; 00567027; 00567248; 00569776; 00570377; 00571642; 00572453; 00573021;	FRAAL_XERD	;00548757	;00573021	Q740E3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) XERD_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) XERD_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) XERD_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) XERD_MYLE0(MYCOBACTERIUM_LEPRAE_TN)
HBG113977	; 00548508; 00549586; 00570392; 00571740; 00573953; Putative carboxymethylenebutenolidase	FRAAL_PE4887			Q73Z00_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O86353_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D7J3_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TZ31_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q9CC04_MYLE0(MYCOBACTERIUM_LEPRAE_TN)
HBG121935	; 00548954; Cytochrome P450 26A2; Cytochrome P450 88A3; Cytochrome P450 90A1; Cytochrome P450 90C1; Pu		;00548954		Q7TXC6_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) C136_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) C136_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73VA5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG124083	; 00567692; Endothelin-converting enzyme 1; Endothelin-converting enzyme-like 1; Hypothetical zinc met	FRAAL_PE4107		;00567692	Q83N63_TROW8(TROPHERYMA_WHIPPLEI_TW08/27) Q83FG7_TROWT(TROPHERYMA_WHIPPLEI_STR._TWIST)
HBG125743	; 00570339; 00573351; 2;3-dihydroxyphenylpropionate 1;2-dioxygenase; MhpB; Similar to 2			;00570339	Q742Z0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG125843	; 00545775; 00574363; Protein mraZ	FRAAL_MRAZ	;00545775	;00574363	Q83N09_TROWT(TROPHERYMA_WHIPPLEI_STR._TWIST) Q83HJ5_TROW8(TROPHERYMA_WHIPPLEI_TW08/27)
HBG218035	; 00549713; 00571866; 35 kDa protein	FRAAL_PE5727	;00549713	;00571866	Q8NLJ3_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FLQ7_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NEJ1_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) Q7TXZ5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) 35KD_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) 35KD_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73W06_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q6NGQ3_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) Q8FPC8_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q8NP60_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)

famille hogenom	annotations hogenom	sequences from AIni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
					MMA1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) MMA1_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) MMA1_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73SF7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q73SW0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q6MX39_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U1X5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q745E2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q8VKH1_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q79FX7_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U1K0_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q79FX8_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q8VKH2_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U1K1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q73SF8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q9CBK3_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q7TWK2_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) CFA1_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) CFA1_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q9CB40_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q73SV9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG218472	; 00547544; 00573197; Cyclopropane-fatty-acyl-phospholipid synthase 2; Methoxy mycolic acid synthase 1	FRAAL_PE3451	;00547544	;00573197	Q6MX38_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9R5_MYTU0(
HBG218601	; 00549607; Cobalamin biosynthesis protein cobIJ [Includes: Precorrin-2 C	FRAAL_COBI	;00549607		COBI_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) COBI_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) COBI_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73YZ1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG218790	; 00545836; 00570278; 00571679; 00573455; 00573592; CysQ protein homolog			;00573455	Q73Y98_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG218790	; 00545836; 00570278; 00571679; 00573455; 00573592; CysQ protein homolog	FRAAL_PE2107	;00545836	;00573592	Q73YS8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) CYSQ_MYLE0(MYCOBACTERIUM_LEPRAE_TN) CYSQ_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) CYSQ_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) CYSQ_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q8NS37_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FR46_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NIF1_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG219066	; 00571300; Putative fatty-acid--CoA ligase fadD33	FRAAL_PE6538		;00571300	Q9CD50_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q741W0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q7D8Z9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O05598_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U0V1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG219332	; 00549258; 00568165; Cyanoglobin; GlbN; Hemoglobin-like protein HbN	FRAAL_GLBN	;00549258	;00568165	Q740T9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) GLBN_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) GLBN_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) GLBN_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG219785	; 00549650; 00573628; Lat; Probable L-lysine-epsilon aminotransferase	FRAAL_LAT	;00549650	;00573628	Q73UF8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) LAT_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) LAT_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) LAT_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG220119	; 00567404; 00570338; 00573350; Hypothetical protein			;00570338	Q742Z1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG220393	; NADH PYROPHOSPHATASE NUDC; NADH pyrophosphatase	FRAAL_PE398			NUDC_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) NUDC_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TX14_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221095	; 00548997; Hypothetical protein MT1019; Hypothetical protein Mb1017c		;00548997		Q7U0W6_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) O05573_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D909_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q742B6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)

famille hogenom	annotations hogenom	sequences from AIni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG221118	; 00572914	FRAAL_PE5534		;00572914	Q73V42_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O05781_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TX88_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D643_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG221135	; 00568610; Acetyltransferase; GNAT family; POSSIBLE TRANSFERASE			;00568610	O05841_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D5W4_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TWY6_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221152	; 00545697; 00568467	FRAAL_TMK	;00545697	;00568467	Q8NSC3_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FRJ3_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NIS4_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) Q73UK7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q9CCJ3_MYLE0(MYCOBACTERIUM_LEPRAE_TN) O05891_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D5U8_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TWW8_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221154	; 00545916; 00569707	FRAAL_PE1986	;00545916	;00569707	Q73UK4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O05895_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D5U5_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TWW3_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221167	; 00566844			;00566844	O06085_MYLE0(MYCOBACTERIUM_LEPRAE_TN) P96398_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7DA93_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U2J7_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q73TR6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG221171	; 00573852; CONSERVED MEMBRANE PROTEIN; Conserved membrane protein; Hypothetical protein MT1652	FRAAL_PE3788		;00573852	Q740P1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O06133_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D897_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TZV9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q9CC50_MYLE0(MYCOBACTERIUM_LEPRAE_TN)
HBG221215	; 00548900; IS1540; transposase; POSSIBLE TRANSPOSASE; Possible transposase		;00548900		Q7TWI9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221236	; 00573894; Hypothetical protein MT0357; ISONIAZID INDUCTIBLE GENE PROTEIN INIA			;00573894	Q7U279_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221237	; 00573901; Hypothetical protein MT0357.1; ISONIAZID INDUCTIBLE GENE PROTEIN INIC			;00573901	O06294_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9Z5_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U278_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221243	; Hypothetical protein MLCB4.22c; Hypothetical protein MT0372; Hypothetical protein Mb0363c	FRAAL_PE4782 FRAAL_PE4781			Q73WE1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q73XD0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O69594_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q73T49_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q7U269_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) O06307_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9Y8_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG221250	; CONSERVED INTEGRAL MEMBRANE PROTEIN; Hypothetical protein MT3561; Probable conserved transmembrane p	FRAAL_PE4195			Q8VJ05_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TWH0_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) O06321_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG221255	; 00567061; Conserved hypothetical protein; Hypothetical protein MT3578			;00567061	O06337_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D5G9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TWF9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221256	; 00566608			;00566608	Q8VIZ6_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O06338_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TWF8_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q743K3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG221257	; 00566648; 00567116; Hypothetical protein MT3583; POSSIBLE TRANSMEMBRANE PROTEIN			;00566648; 00567116	Q7D5G6_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O06342_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)

famille hogenom	annotations hogenom	sequences from AlnI	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG221278	; 00546004; 00568389	FRAAL_PE6584	;00546004	;00568389	Q744A5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O69539_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q7D565_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O06380_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TW31_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q8NM80_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FMF9_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NF89_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG221286	; Hypothetical protein Cgl0455; Hypothetical protein Mb0557c; U296w	FRAAL_PE1463			Q73Z39_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q50171_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q7U1T4_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) O06409_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73SN4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG221312	; 00547400; 00572257; 00574394; Hypothetical protein MT1150; Hypothetical protein Mb1149c			;00574394; 00572257	Q73WI7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O06570_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D8T8_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TVC0_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221320	; 00548947; 00571462; 2-methylcitrate dehydratase 1; 2-methylcitrate dehydratase 2	FRAAL_PRPD	;00548947	;00571462	Q744P4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q7U0K9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) O06582_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D8S9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q8FRP3_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) PRD1_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) PRD2_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG221345	; 00547651; Hypothetical protein Mb0828; Icc protein	FRAAL_PE3203 FRAAL_PE4367 FRAAL_PE3006	;00547651		Q50024_MYLE0(MYCOBACTERIUM_LEPRAE_TN) O06629_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D993_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U185_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221369	; 00569182; TRANSCRIPTIONAL REGULATORY PROTEIN; Transcriptional regulator; IclR family			;00569182	O06806_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TZK5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q8VJW7_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG221392	; 00566643; Probable peptidase RP174			;00566643	O07178_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U1Y4_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D9S4_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q73SX2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG221395	; 00572743; 29 kDa ANTIGEN CFP29; Bacteriocin CFP29; Hypothetical protein; LINOCIN M18	FRAAL_PE4503		;00572743	Q743F5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O07181_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D997_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U190_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221403	; 00545854; 00572962; Conserved hypothetical protein; Hypothetical protein MT2768	FRAAL_PE2082	;00545854	;00572962	Q7TY33_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D6R5_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O07196_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73W49_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG221421	; 00573218; Hypothetical protein Mb0307			;00573218	O07226_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U2B8_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221422	; 00567015; Hypothetical protein Mb0308			;00567015	O07227_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U2B7_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221423	; 00567014			;00567014	O07228_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7DA25_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U2B6_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221441	; 00568878; Hypothetical protein Mb0339			;00568878	Q73T92_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O07256_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U287_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221475	; 00569199			;00569199	O07748_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG221495	; 00545458; 00546430; 00549340; 00571349; Hypothetical protein Rv2596/MT2672 precursor	FRAAL_PE206	;00545458;00549340	;00571349	O07780_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9K3_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U1N8_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)

famille hogenom	annotations hogenom	sequences from AIni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG221495	; 00545458; 00546430; 00549340; 00571349; Hypothetical protein Rv2596/MT2672 precursor		;00546430		YP96_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YP96_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TY97_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221497	; 00548425; 00569795		;00548425		Q7D9K6_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O07783_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U1P1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) O06774_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9G5_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U1I2_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG221497	; 00548425; 00569796			;00569795	Q73YC9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) P96917_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U1L2_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG223284	; 00573799			;00573799	Q8NU48_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FUB9_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG224863	; 00568726; Hypothetical protein MJ0830; Hypothetical protein slr1717			;00568726	Q8FMZ0_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG225669	; 00548119; 00569623	FRAAL_PE5822	;00548119	;00569623	P95112_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D6B3_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TXH8_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q73VJ2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O33128_MYLE0(MYCOBACTERIUM_LEPRAE_TN)
HBG225729	; 00549354; Conserved hypothetical protein; Hypothetical protein Mb1279c		;00549354		O50462_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U0A7_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG225737	; 00568180	FRAAL_PE5344		;00568180	Q73SR2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O33364_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9Q3_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U1V4_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q743H7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q7U1B1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) P71829_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9A8_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG227155	; 00548321; Hypothetical protein Rv3408/MT3516		;00548321		P95252_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D7P4_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TZ71_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG227155	; 00548321; Hypothetical protein Rv3408/MT3517	FRAAL_PE552			Q7TWJ5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YY08_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YY08_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG227169	; Conserved hypothetical membrane protein; Hypothetical membrane protein; Hypothetical protein MT1272;	FRAAL_PE2606			Q8FM91_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG227862	; 00548793; 00569673	FRAAL_PE5857	;00548793	;00569673	Q7D6A1_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O53260_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TXG1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q9CBR7_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q73VG7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG227877	; 00570573; Conserved hypothetical protein; Hypothetical protein MT3166			;00570573	O53298_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TXA8_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D659_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG227886	; 00571268; Hypothetical protein CC1232; Hypothetical protein MT3258; Hypothetical protein Mb3194			;00571268	O53319_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TX42_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D605_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q73UZ1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG227923	; 00546031; 00568440; 00569212; Glutamine amidotransferases; class-II; Hypothetical protein Mb1090	FRAAL_PE3589 FRAAL_PE6623	;00546031	;00569212; 00568440	Q741S7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O53409_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D8X1_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U0R5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)

famille hogenom	annotations hogenom	sequences from AInl	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG227924	; 00573185; Hypothetical protein MT1092; Hypothetical protein Mb1091			;00573185	Q741S6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q7D8X0_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O53410_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U0R4_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG227942	; 00545356; 00567998; Hypothetical protein MT2074; Hypothetical protein Mb2041		;00545356		Q7TZ55_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D7M4_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O53464_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG227945	; 00545623; 00548324; 00570851; Hypothetical protein MT2078; Hypothetical protein Mb2045c		;00548324;00545623	;00570851	O53468_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D7M0_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TZ51_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG227948	; 00572065; Hypothetical protein MT2089; Hypothetical protein Mb2056c			;00572065	O53475_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D7L3_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TZ41_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG227951	; 00570288; Hypothetical protein MT2096; Hypothetical protein Mb2062			;00570288	Q745M0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q7TZ38_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D7K8_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O53480_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG227965	; 00547843	FRAAL_PE5084	;00547843		Q73YN8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O69570_MYLE0(MYCOBACTERIUM_LEPRAE_TN) O53513_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D7E6_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TYY4_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q8FNS0_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q8NNL4_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q6NGB0_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG227985	; 00567587; Conserved hypothetical protein; Hypothetical protein MT3614			;00567587	O53555_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D5E0_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TWC1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q743N9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG228005	; 00548940; Hypothetical protein MT0077; POSSIBLE MATURASE; Possible maturase		;00548940		O53616_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U2X2_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q8VKS1_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG228042	; 00569259; 00569739; Hypothetical protein MT0384.1; POSSIBLE MEMBRANE OXIDOREDUCTASE			;00569739	O53704_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U258_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D9X9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG228065	; 00546682; 00571224; Hypothetical protein MT0597; Hypothetical protein Mb0586c	FRAAL_PE6314	;00546682	;00571224	Q7D9M2_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O53768_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U1R3_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG228067	; 00546368; 00567591; 00572878		;00546368	;00572878;00567591	Q7U1R0_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) O53771_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9L9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG228116	; 00567846; 00569063; 00570265; 00573721			;00573721;00567846;00570265;00569063	O53904_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D8B6_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7VEZ2_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG230117	; 00548342; 00572010		;00548342		Q73S07_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG230117	; 00548342; 00572011			;00572010	O65936_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TZM6_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D810_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG230507	; 00547726; 00570005; 00573366	FRAAL_GSHA	;00547726	;00570005;00573366	Q744N5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O69672_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D512_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TVW1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG230528	; 00548052; Excisionase; putative; POSSIBLE EXCISIONASE		;00548052		O69717_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D4X9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TVS3_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)

famille hogenom	annotations hogenom	sequences from Alni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG232401	; 00546142; 00570527; 00572224; 00573316		;00546142	;00572224	O86367_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9G4_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U1H9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG232401	; 00546142; 00570527; 00572224; 00573317			;00570527	Q73UF1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q9CCL5_MYLE0(MYCOBACTERIUM_LEPRAE_TN) P96903_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D5R4_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TWS7_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG233388	; 00546606; 00572843	FRAAL_CHL	;00546606	;00572843	Q7D920_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U0Y8_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) P71552_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q742D2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG233424	; 00569838			;00569838	P71651_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D6J8_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TXU5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q73VV9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG233433	; 00566675; Hypothetical protein MT0051; POSSIBLE HYDROLASE; Possible hydrolase	FRAAL_PE3995		;00566675	Q744U6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) P71702_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7DAJ5_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U2Y9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG233464	; 00567463			;00567463	P71821_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9B3_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U1B6_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q743I6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG233480	; 00548180; Conserved hypothetical protein; Hypothetical protein MT3632		;00548180		P71872_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D5C8_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TWA2_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q743Q4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG233596	; 00569451	FRAAL_PE4546		;00569451	P71733_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D769_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TYM8_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q49770_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q73XT7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG234275	; 00547277	FRAAL_PE3804	;00547277		Q743L2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG235096	; 00568309; Hypothetical protein MT2604			;00568309	Q7TYD3_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q8VJG6_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) P95024_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG235105	; 00567733; 00569737; Hypothetical protein MT0719.2; Hypothetical protein Mb0711	FRAAL_PE3571		;00567733	Q73SC2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q7D9F0_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) P95038_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U1G3_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG235106	; 00567734; 00569736; COENZYME PQQ SYNTHESIS PROTEIN E PQQE; MoaA/nifB/pqqE family protein; PqqE	FRAAL_PQQE		;00567734	Q73SC1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) P95039_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9E9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U1G2_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG235108	; Glycosyl transferase; MEMBRANE SUGAR TRANSFERASE	FRAAL_PE2582			Q7U1F9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D9E7_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) P95042_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG235122	; 00574317			;00574317	Q73VA2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) P95098_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TXC5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q8VJ78_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG235125	; 00547736		;00547736		P95110_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D6B2_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TXH7_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q9CBR8_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q73VJ0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)

famille hogenom	annotations hogenom	sequences from AInI	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG235139	; Hypothetical protein MT1915; Hypothetical protein Mb1897	FRAAL_PE4764 FRAAL_PE3876			Q8VJV5_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) P95149_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TZE9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q73ZM6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG235158	; Hypothetical protein MT0285	FRAAL_PE5278			Q73V24_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG235165	; 00546187		;00546187		P95251_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D7P3_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TZ70_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG235182	; Luciferase-related protein; LuxA2 protein; POSSIBLE MONOOXYGENASE	FRAAL_PE2518			Q7TZ96_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D7R4_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) P95278_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG235274	; Hypothetical protein MT3940; Hypothetical protein Mb3862c	FRAAL_PE1944			Q7TVK1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) P96246_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q8VIT2_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q745I6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG235277	; 00573994			;00573994	Q73T23_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) P96259_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9V1_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U218_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q8NM32_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FMC2_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG235316	; 00567514			;00567514	Q73TN2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q9CD19_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q7DA75_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) P96419_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U2H8_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG235342	; 00547052; 00568810	FRAAL_PE1214	;00547052	;00568810	Q73UH2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q7TWU1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) P96883_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q9CCL2_MYLE0(MYCOBACTERIUM_LEPRAE_TN)
HBG235350	; 00545925; 00572758	FRAAL_PE1967	;00545925	;00572758	Q9CBJ4_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q73SH0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) P96924_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9I1_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U1K6_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG235397	; 00574126; Putative 2-dehydropantoate 2-reductase			;00574126	Q741M8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) PANE_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) PANE_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TYA5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG235553	; 00548566; 00569902; 00573527; Probable serine/threonine-protein kinase pknK			;00573527; 00569902	PKNK_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) PKNK_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TXA9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG240179	; 00548268; 00573784	FRAAL_PE6035	;00548268	;00573784	Q73US0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q49765_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q8VJ53_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q6MX01_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TX16_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG245973	; 00549252; Hypothetical protein ygbF	FRAAL_PE459	;00549252		Q6NEQ2_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG246218	; 00569189			;00569189	Q8G715_BILO0(BIFIDOBACTERIUM_LONGUM_NCC2705)
HBG246243	; 00548542; 00571494; 00574491	FRAAL_PE2459 FRAAL_PE2849	;00548542	;00571494; 00574491	Q8G773_BILO0(BIFIDOBACTERIUM_LONGUM_NCC2705)
HBG248414	; 00547044; 00568797; Hypothetical protein Rv0433/MT0448/Mb0441; Hypothetical protein ybdK	FRAAL_PE1203	;00547044	;00568797	Q8NM44_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FMD3_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NF59_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) Y433_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Y433_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Y433_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73T00_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)

famille hogenom	annotations hogenom	sequences from AIni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG249431	; AGR_L_3173p; AGR_pAT_bx5p; AGR_pTi_175p; DNA POLYMERASE III ALPHA CHAIN PROTEIN; DNA polymerase III;	FRAAL_PE2180			Q73U92_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q7TWL9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7D5L9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O50399_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG250056	; 00570579	FRAAL_DESA		;00570579	Q73UM3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q6MWZ3_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D5W1_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TWY2_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q6NIR3_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG250467	; 00546763; 00572985	FRAAL_CAPD	;00546763	;00572985	Q8NTG0_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FSM3_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG250777	; Bll4229 protein; Glr0328 protein; Hypothetical protein PA1402; Mll5689 protein; Putative O-methyltra	FRAAL_PE5597			Q73U53_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG252002	; 00547699; 00568558	FRAAL_AMIB	;00547699	;00568558	Q8NL57_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FSV9_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NEA1_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) Q73RT7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q9CCX8_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q79F87_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D4L9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TVC7_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG252032	; 00547730; 00568591; 00568684; Hypothetical protein; Phage-related regulatory protein cII	FRAAL_PE6829	;00547730	;00568591; 00568684	Q6NEG6_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG252035	; 00566998; Hypothetical protein; Putative DNA-binding protein			;00566998	Q6NEH4_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG252039	; 00547710; 00573673; Hypothetical protein Rv0049/MT0055/Mb0050	FRAAL_PE6857	;00547710	;00573673	O53130_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q744W0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Y049_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Y049_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Y049_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q8NLF5_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FLP3_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NEH9_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG252054	; 00571792			;00571792	Q6NEM8_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG252055	; 00573754			;00573754	Q8NLP4_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FLY3_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NEN0_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) P96243_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D4S2_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TVJ9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q9CDC2_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q745I9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG252146	; 00547250; Hypothetical protein	FRAAL_PE6617	;00546028	;00568433	Q8NM34_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FMC4_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NF52_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG252215	; 00546028; 00568433	FRAAL_PE4850	;00547597	;00573428	Q73V39_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q73UU4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q8FMS0_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NFL3_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG252313	; 00547597; 00573428; Hypothetical protein			;00571221	Q6NG89_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) Q8FNP7_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q8NNJ3_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG252352	; 00571221; Hypothetical protein Cgl2206; Putative oxidoreductase	FRAAL_THYX	;00549122	;00571881	THYX_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) THYX_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) THYX_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) THYX_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q73VZ6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)

famille hogenom	annotations hogenom	sequences from AIni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG252410	; 00549122; 00571881; Hypothetical protein; Thymidylate synthase thyX			;00572337	Q73U95_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q6NHC2_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) Q8FTH9_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q8NQI0_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG252500	; 00572337; Hypothetical protein; SAM-dependent methyltransferases	FRAAL_PE2805		;00573055	Q741R9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O53419_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D8W6_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U0Q8_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) O07138_MYLE0(MYCOBACTERIUM_LEPRAE_TN)
HBG252697	; 00573055			;00570498	Q8G7M7_BILO0(BIFIDOBACTERIUM_LONGUM_NCC2705) Q6NJN7_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) Q8NTH4_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FSN9_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG252828	; 00571317; Putative Ni/Fe-hydrogenase B-type cytochrome subunit			;00569185	Q8NU14_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FU83_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NKA6_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG253175	; 00570498	FRAAL_PE4994	;00549624		Q8FTW9_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG253416	; 00568264; Putative transcriptional regulator		;00546297		Q8FUH2_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG260986	; 00569185		;00549377		Q8VK46_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O50461_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U0A8_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG261788	; 00549624			;00571409	Q73RW4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261802	; 00546297	FRAAL_PE4176	;00546324		Q73RZ2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q9CCZ2_MYLE0(MYCOBACTERIUM_LEPRAE_TN)
HBG261804	; 00547470; 00549377; 00571365; Hypothetical protein MJ0071; Hypothetical relE protein	FRAAL_PE1137	;00547198	;00570150	YY12_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YY12_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YY12_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) YY12_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q73S01_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261812	; 00573662; Hypothetical protein	FRAAL_PE2030 FRAAL_PE1479	;00547224	;00572785	Q740V7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q73S53_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q73S55_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q73S54_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q740W3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261813	; 00571409; Bll1004 protein; Hypothetical protein CC3460	FRAAL_PE4118			Q73S56_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261814	; 00572225; Hypothetical protein	FRAAL_PE4136			Q73S58_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261815	; 00546324; Hypothetical protein ML2649	FRAAL_PE4137			Q73S59_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261822	; 00547198; 00570150; Hypothetical protein ML0386; Hypothetical protein Rv3412/MT3521/Mb3446		;00546209	;00567808	Q73S72_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) O07744_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q8VJU9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TZD0_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG261876	; 00547224; 00572785; AtfA_1; AtfA_2; Hypothetical protein	FRAAL_PE4774 FRAAL_PE2840		;00572503	YD60_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YD60_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YD60_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG261878	; 00548202; Bll3016 protein; Hypothetical protein RSp1652	FRAAL_PE3435	;00547012	;00571586	Q73T11_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261949	; Hypothetical protein	FRAAL_PE1951			Q73TQ8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261950	; Hypothetical protein	FRAAL_PE3790		;00571900	Q73TR4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261958	; 00546209; 00567808; Hypothetical protein MT1936; POSSIBLE TRANSMEMBRANE PROTEIN		;00549777		Q73TU8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261968	; 00545427; 00572907; Hypothetical protein	FRAAL_SSUD FRAAL_PE3427 FRAAL_PE3500		;00570975; 00566825	Q73TY4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261980	; 00567093; 00568497; 00572503; Hypothetical protein Rv1360/MT1405/Mb1395 precursor	FRAAL_PE2926			Q73U47_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)

famille hogenom	annotations hogenom	sequences from Alni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG262001	; 00547012; 00571586; Hypothetical protein; Oxidoreductase; putative; POSSIBLE OXIDOREDUCTASE	FRAAL_PE2157	;00545795	;00572639	Q73UD4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q73UD3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q740Y5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q9CBL9_MYLE0(MYCOBACTERIUM_LEPRAE_TN) YE81_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YE81_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YE81_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG262012	; 00574192; Hypothetical protein	FRAAL_PE4189			Q73UL3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262020	; 00574199; Hypothetical protein			;00570263	Q73UN7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q744Q2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262034	; 00546183; Hypothetical protein	FRAAL_PE4124	;00546414	;00566806	Q73UV0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262039	; 00566645; Hypothetical protein	FRAAL_PE3238	;00546523;00547481	;00568012	Q73V63_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q73UV9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262045	; 00571672; Hypothetical protein	FRAAL_PE3903			Q73UX2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262049	; 00571900; Hypothetical protein	FRAAL_PE6044	;00549732	;00573791	Q73UX6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262051	; 00549777; 00567656; 00567768; 00572270; 00573578; 00573580; Hypothetical protein	FRAAL_PE304			Q73UX8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262071	; 00572506; Hypothetical protein	FRAAL_PE6792			Q73V38_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262087	; 00566825; 00570975; Hypothetical protein			;00569026	Q73V71_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262089	; 00569175; 00572507; Hypothetical protein			;00570172	Q742W5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262089	; 00569175; 00572507; Hypothetical protein			;00569023; 00567680	Q73V78_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262089	; 00569175; 00572507; Hypothetical protein			;00572677	Q73V73_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262092	; 00570262; Hypothetical protein			;00569021; 00567682	Q73V79_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262098	; Hypothetical protein			;00571418	Q73V89_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262141	; 00568151; Hypothetical protein	FRAAL_PE4895	;00547603	;00569906	Q73VR0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262153	; 00568153; Hypothetical protein	FRAAL_PE3502			Q73VX9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262220	; 00545795; 00572639; Hypothetical protein Rv1481/MT1528/Mb1517; Possible membrane protein	FRAAL_PE3135			Q73WK8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262226	; Hypothetical protein	FRAAL_PE4207		;00570255	Q73WN3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q8VK56_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG262238	; 00570263; Hypothetical protein	FRAAL_PE3765		;00569308	Q73WW2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q9CCA3_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q79IB6_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7ARU8_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D8J9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551)
HBG262244	; 00546414; 00566806; Hypothetical protein			;00570215	Q742R0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262244	; 00546414; 00566806; Hypothetical protein			;00569847	Q744K6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q743B9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262244	; 00546414; 00566806; Hypothetical protein			;00573645; 00573402;00573638	Q73WY3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262247	; 00569167; 00569849; 00572230; Hypothetical protein			;00572811; 00567762;00567765	Q7D8X3_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O53407_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U0R7_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q744J2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262247	; 00569167; 00569849; 00572230; Hypothetical protein			;00573806	Q73WY7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262247	; 00569167; 00569849; 00572230; Hypothetical protein			;00568034	Q73VY0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262281	; 00546523; 00547481; 00568012; Hypothetical protein			;00566843	Q73XA3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262283	; AGR_L_1631p; Bll0572 protein; DNA topoisomerase; type I; putative; Hypothetical protein PA2244; Hypo	FRAAL_PE3068		;00571246	Q73XA7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q73XA8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262284	; 00549732; 00573791; Hypothetical protein			;00569205	Q73XA9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)

famille hogenom	annotations hogenom	sequences from AInl	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG262291	; Acyl-CoA dehydrogenase-related protein; Dehydrogenase; Hypothetical protein			;00568158; 00567790;00570343;00570192;00567074	Q73XF2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262291	; Acyl-CoA dehydrogenase-related protein; Dehydrogenase; Hypothetical protein		;00567091		,Q73XC0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262298	; Hypothetical protein	FRAAL_PE3106 FRAAL_PE3374			Q73XD6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262317	; 00569026; Aldehyde dehydrogenase; Hypothetical protein XCC0808	FRAAL_PE5336	;00546299	;00566979; 00573802	Q73XH3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262412	; 00567680; 00567686; 00567753; 00567785; 00568163; 00569016; 00569023; 00569878; 00570172; 00570951;	FRAAL_PE5497			Q73YD7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262434	; 00572678; Hypothetical protein	FRAAL_PE5107	;00547859	;00569574	Q7TYX5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YL91_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YL91_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73YM5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262442	; 00567684; 00569019; Hypothetical protein	FRAAL_PE5596		;00572563	Q73YU1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262480	; 00567303; 00567682; 00567787; 00568133; 00568143; 00569021; 00569813; 00570196; 00571144; 00571498;	FRAAL_PE3520			Q73Z56_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262508	; 00566952; 00569061; Hypothetical protein	FRAAL_PE4293		;00572198	Q73ZC1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262522	; 00571418; Hypothetical protein			;00572674	Q742V0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262534	; 00570305; 00573773; Hypothetical protein	FRAAL_PE761		;00572607	Q73ZK5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262537	; 00572915; Hypothetical protein	FRAAL_PE2309			Q73ZL5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262547	; 00572916; Hypothetical protein	FRAAL_PE2268	;00547430	;00568922	YI25_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YI25_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YI25_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73ZR3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262563	; 00568196; Hypothetical protein	FRAAL_PE3036		;00567740; 00569268	Q73ZW7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262566	; 00568604; Hypothetical protein	FRAAL_PE5965	;00547010		Q73ZX9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262599	; 00547603; 00569906; Hypothetical protein	FRAAL_PE3477			Q740G8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262656	; Hypothetical protein	FRAAL_PE5545			Q741F9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q9CCQ1_MYLE0(MYCOBACTERIUM_LEPRAE_TN) YE05_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) YE05_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YE05_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YE03_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) YE03_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YE03_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG262664	; 00570846; Hypothetical protein	FRAAL_PE4047			Q741I9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262675	; 00567777; 00570272; 00571161; 00572353; Hypothetical protein			;00571691	YP70_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YP70_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YP70_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q741M3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262680	; 00567730; 00572172; Hypothetical protein	FRAAL_PE6681	;00546069	;00568687	YP77_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YP77_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TYA3_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q741N5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262686	; 00569080; 00570271; Hypothetical protein			;00569241	YP98_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YP98_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YP98_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q741Q4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262695	; 00569101; 00573774; Hypothetical protein	FRAAL_KDPC	;00545453	;00570221	Q741U0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)

famille hogenom	annotations hogenom	sequences from AIni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG262725	; 00567462; 00569290; 00570201; 00570981; Hypothetical protein	FRAAL_PE6214	;00546608	;00572842	Q8NRJ2_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FQL5_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Y959_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Y959_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Y959_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q742D1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262727	; Hypothetical protein	FRAAL_PE4765			Q742D7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262762	; 00570255; Hypothetical protein MT1192			;00567665	Q742N8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262800	; 00569308			;00570337	Q742Z2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262808	; 00567298; 00567761; 00568160; 00568988; Hypothetical protein	FRAAL_PE2888	;00546411		Q743F1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262895	; 00568989; 00569743; 00571499; Hypothetical protein			;00570031	Q744U8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Y043_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Y043_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Y043_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG262906	; 00566696; 00567683; 00569020; 00569748; 00569846; 00569847; 00570215; 00573402; 00573638; 00573645;	FRAAL_PE502	;00546272	;00572899	Q744Y3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262935	; 00567762; 00567765; 00568034; 00570198; 00572811; 00573806	FRAAL_PE4180			Q745E1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262961	; 00566690; Hypothetical protein	FRAAL_PE3055			Q745P0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262961	; 00566690; Hypothetical protein	FRAAL_PE3559			Q745S1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262964	; 00566843; Hypothetical protein; Putative dehydratase/racemase	FRAAL_PE3558		;00566953	Q745Q2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262986	; 00571246; FadD35; Hypothetical protein	FRAAL_PE3336			Q7D5Z7_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O53329_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TX32_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG262986	; 00571246; FadD35; Hypothetical protein			;00569098	O07781_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9K4_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U1N9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG268237	; 00569205; Hypothetical protein			;00571464	Q73XA5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q73TW9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q7U2P3_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q79G00_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG268396	; 00566850; 00567074; 00567091; 00567104; 00567790; 00568158; 00569177; 00570192; 00570343; 00570668;	FRAAL_PE5192 FRAAL_PE5865			Q79VK1_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG269335	; 00567099; Hypothetical protein			;00571242	Q8NSE1_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FSZ8_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG269337	; 00567469; Hypothetical protein			;00570704	Q8NSD8_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FSZ4_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG269338	; Hypothetical protein			;00570705	Q8NSD7_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FSZ1_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG273169	; 00568033; 00571736; Hypothetical protein		;00547007		Q7D5N6_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TVB5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q93IG7_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG277781	; 00571427; Hypothetical protein	FRAAL_PE1873	;00548669	;00569470	O05911_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D935_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U111_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG278295	; 00567460; 00573853; Hypothetical protein		;00545718	;00570852	O53333_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D5Z3_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TX28_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG279910	; 00546299; 00566979; 00571416; 00573802; Hypothetical protein	FRAAL_PE207	;00545731;00549353		Q7D9K2_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O07779_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7U1N7_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG283044	; 00549760; Hypothetical protein	FRAAL_PE310			Q7TWB4_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG283082	; 00571017; Hypothetical protein			;00569242	Q7TY96_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)

famille hogenom	annotations hogenom	sequences from AIni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG284880	; 00571732; Hypothetical protein	FRAAL_PE5756	;00549134	;00571891	Q7D6H2_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) P71615_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7TXQ9_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q73VV6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG286523	; 00569774; 00571725; 00573331; Hypothetical protein	FRAAL_PE2158	;00545794	;00572640	Q9CBL8_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q740Y6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) YE80_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YE80_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YE80_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG289481	; 00547307; 00547873; 00548591; 00569066; Hypothetical protein			;00568815	Q741M1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) YP68_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YP68_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YP68_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG292795	; 00548696; Hypothetical protein	FRAAL_PE5304		;00571244	Q8NQG5_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FTH0_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG300255	; 00568990; Hypothetical protein		;00548588	;00568301; 00570735	Q73V70_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG301437	; Hypothetical protein	FRAAL_PE2271	;00547428	;00568924	Q8NQJ3_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FTJ5_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NHD4_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) YI27_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YI27_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YI27_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) O32919_MYLE0(MYCOBACTERIUM_LEPRAE_TN) Q73ZR1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG301452	; 00545587; 00547274; Hypothetical protein	FRAAL_PE1321	;00546895	;00570856	Q83NE8_TROW8(TROPHERYMA_WHIPPLEI_TW08/27) Q83GQ8_TROWT(TROPHERYMA_WHIPPLEI_STR._TWIST)
HBG301516	; 00547859; 00569574; Hypothetical protein Mb2214; Hypothetical protein Rv2191/MT2247	FRAAL_PE1254	;00547081	;00574108	Q8NSD5_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FRK6_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NIT5_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129) Q73UI9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO) Q7TTR5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q6MWZ0_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D5T2_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q9CCK7_MYLE0(MYCOBACTERIUM_LEPRAE_TN)
HBG309734	; 00549667; FtsQ			;00571223	Q7D6N0_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TXY5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) O33298_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q8FN21_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q6NEG0_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG310080	; 00572563; Hypothetical protein			;00567466	O53905_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D8B5_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TZX7_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q740U9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG310354	; 00570674; 00573347; Hypothetical protein			;00571451	Q8NTJ2_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG317553	; Hypothetical protein		;00549401;00548937;00547838;00546484	;00571292; 00567643;00566632;00567599;00567550;00567548;00567487	Q8FRV1_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q8CM86_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q8FLI0_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314) Q8FSH8_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG322756	; 00569007; Hypothetical protein			;00566599	Q6NIK6_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG322756	; 00569007; Hypothetical protein			;00566623	O53337_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D5Y9_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TX24_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG325635	; 00549697; Hypothetical protein	FRAAL_PE5655	;00546760	;00572981	Q8NLD0_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032) Q8FLM3_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)

famille HOGENOM	annotation HOGENOM	bootstrap/best argument for HT	sequences from AIni	sequences from Cci3	sequences from EAN	sister group in tree
HBG113825	GLUTATHIONE SYNTHETASE; PUTATIVE GLUTATHIONE SYNTHASE; RIBOSOMAL PROTEIN	(no_actino_in_family)			;00570717	GSHB_SYNEL(SYNECHOCOCCUS_ELONGATUS) GSHB_ANASP(NOSTOC_SP._PCC_7120) GSHB_SYNY3(SYNECHOCYSTIS_SP._PCC_6803) GSHB_GLVI0(GLOEOBACTER_VIOLACEUS_PCC_7421) GSHB_PROMM(PROCHLOROCOCCUS_MARINUS_STR._MIT_9313) GSHB_SYNPX(SYNECHOCOCCUS_SP._WH_8102) GSHB_PRMA0(PROCHLOROCOCCUS_MARINUS_SUBSP._MARINUS_S) GSHB_PROMP(PROCHLOROCOCCUS_MARINUS_SUBSP._PASTORIS) GSHB_BRSU0(BRUCELLA_SUIS_1330) GSHB_BRME0(BRUCELLA_MELITENSIS_16M) GSHB_RHILO(MESORHIZOBIUM_LOTI) GSHB_RHIME(SINORHIZOBIUM_MELILOTI) GSHB_AGRT5(AGROBACTERIUM_TUMEFACIENS_STR._C58) GSHB_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009) GSHB_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) GSHB_CACR0(CAULOBACTER_CRESCENTUS_CB15) Q73I56_WOLPM(WOLBACHIA_SP._WMEL) GSHB_BORPE(BORDETELLA_PERTUSSIS) GSHB_BORBR(BORDETELLA_BRONCHISEPTICA) GSHB_BORPA(BORDETELLA_PARAPERTUSSIS) GSHB_RASO0(RALSTONIA_SOLANACEARUM_GMI1000) GSHB_NEME1(NEISSERIA_MENINGITIDIS_Z2491) GSHB_NEME0(NEISSERIA_MENINGITIDIS_MC58) GSHB_CHVI0(CHROMOBACTERIUM_VIOLACEUM_ATCC_12472) GSHB_NIEU0(NITROSOMONAS_EUROPAEA_ATCC_19718) GSHB_XACA0(XANTHOMONAS_CAMPESTRIS_PV._CAMPESTRIS_ST)
HBG224822	HYPOTHETICAL PROTEIN AF1688; MEMBRANE PROTEIN; PUTATIVE	(no_actino_in_family)	FRAAL_PE6820			Q72E09_DESVH(DESULFOVIBRIO_VULGARIS_SUBSP._VULGARIS_S) O28585_ARFU0(ARCHAEOGLOBUS_FULGIDUS_DSM_4304)
HBG225410	BLL7863 PROTEIN; HYDROLASE; HYPOTHETICAL PROTEIN; METALLO-BETA-LACTAMASE FAMILY PROTEIN; MLR3962 PRO	(no_actino_in_family)	FRAAL_PE6186		;00570362	Q6KQ76_BAAN0(BACILLUS_ANTHRACIS_STR._'AMES_ANCESTOR') Q81MV6_BACAA(BACILLUS_ANTHRACIS_STR._AMES) Q734H7_BACE0(BACILLUS_CEREUS_ATCC_10987) Q81AY3_BACCR(BACILLUS_CEREUS_ATCC_14579) Q81C53_BACCR(BACILLUS_CEREUS_ATCC_14579) O31787_BASU0(BACILLUS_SUBTILIS_SUBSP._SUBTILIS_STR._1) Q98F33_RHILO(MESORHIZOBIUM_LOTI) Q89CD4_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110)
HBG240156	BLR1748 PROTEIN; DUF269; HYPOTHETICAL PROTEIN FLGK; MLR5912 PROTEIN	(no_actino_in_family)	FRAAL_PE6806	;00546163	;00571389	Q79UV8_BRJA0(BRADYRHIZOBIUM_JAPONICUM_USDA_110) Q6N0Z5_RHPA0(RHODOPSEUDOMONAS_PALUSTRIS_CGA009) Q98AP1_RHILO(MESORHIZOBIUM_LOTI) Q44147_ANASP(NOSTOC_SP._PCC_7120) Q7MRG2_WOLSU(WOLINELLA_SUCCINOGENES)
HBG240283	HYPOTHETICAL PROTEIN XAC0505; HYPOTHETICAL PROTEIN XCC0491; SLL0543 PROTEIN	(no_actino_in_family)	FRAAL_PE1476	;00546836	;00571086	Q55401_SYNY3(SYNECHOCYSTIS_SP._PCC_6803) Q8PQ27_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q8PD54_XACA0(XANTHOMONAS_CAMPESTRIS_PV._CAMPESTRIS_ST)
HBG240327	ALR4836 PROTEIN; HYPOTHETICAL PROTEIN XAC1124; HYPOTHETICAL PROTEIN SLR0325; TLL1464 PROTEIN	(no_actino_in_family)			;00571281	Q8PNE7_XAAX0(XANTHOMONAS_AXONOPODIS_PV._CITRI_STR._30) Q82XI5_NIEU0(NITROSOMONAS_EUROPAEA_ATCC_19718) Q55530_SYNY3(SYNECHOCYSTIS_SP._PCC_6803) Q8YMU1_ANASP(NOSTOC_SP._PCC_7120) Q8DIW7_SYNEL(SYNECHOCOCCUS_ELONGATUS) Q7V417_PROMM(PROCHLOROCOCCUS_MARINUS_STR._MIT_9313) Q7U3R5_SYNPX(SYNECHOCOCCUS_SP._WH_8102)
HBG240498	HYPOTHETICAL PROTEIN; SIMILAR TO UNKNOWN PROTEIN; SLL0783 PROTEIN	(no_actino_in_family)	FRAAL_PE504 FRAAL_PE5959		;00572590	Q7N7C3_PHLU0(PHOTORHABDUS_LUMINESCENS_SUBSP._LAUMONDII) Q55950_SYNY3(SYNECHOCYSTIS_SP._PCC_6803) Q7U6B3_SYNPX(SYNECHOCOCCUS_SP._WH_8102)
HBG242000	HYPOTHETICAL PROTEIN YQJY	(no_actino_in_family)	FRAAL_PE3741		;00572708;00571056	Q8ETH4_OCEIH(OCEANOBACILLUS_IHEYENSIS) YQJY_BASU0(BACILLUS_SUBTILIS_SUBSP._SUBTILIS_STR._1) Q81BV7_BACCR(BACILLUS_CEREUS_ATCC_14579) Q735R2_BACE0(BACILLUS_CEREUS_ATCC_10987) Q6KR64_BAAN0(BACILLUS_ANTHRACIS_STR._'AMES_ANCESTOR') Q81NW4_BACAA(BACILLUS_ANTHRACIS_STR._AMES) Q7P0N4_CHVI0(CHROMOBACTERIUM_VIOLACEUM_ATCC_12472)
HBG242771	BACTERIOCIN O-METYLTRANSFERASE; PUTATIVE; MACROCIN O-METHYLTRANSFERASE	(no_actino_in_family)	FRAAL_PE2038	;00545881	;00569715	Q6KVS1_BAAN0(BACILLUS_ANTHRACIS_STR._'AMES_ANCESTOR') Q81TP7_BACAA(BACILLUS_ANTHRACIS_STR._AMES) Q73BU0_BACE0(BACILLUS_CEREUS_ATCC_10987) Q81GJ3_BACCR(BACILLUS_CEREUS_ATCC_14579) Q98ID4_RHILO(MESORHIZOBIUM_LOTI)

famille hogenom	annotations hogenom	sequences from AInl	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG361091	; 00567665; 00570057; 00570308; Hypothetical protein	FRAAL_PE4547		;00569450	YO11_MYLE0(MYCOBACTERIUM_LEPRAE_TN) YO11_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YO11_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YO11_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q73XT6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG363776	; 00570209; 00572462; Hypothetical protein		;00549357;00548903;00549179;00548988	;00571089;00569941;00571516;00573688	Q7TXV2_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) O33334_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D6K4_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) O07773_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D9J8_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7U1N1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) O07795_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D4S7_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Q7TVK5_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Q7TXI1_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) P95116_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Q7D6B5_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YS86_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YS86_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YS86_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG365129	; 00570180; Hypothetical protein		;00547574	;00570503	Q8NTF6_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG368421	; 00570046; 00571708; Hypothetical protein	FRAAL_PE1293		;00571279	Q73SY2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG391911	; 00570030; 00570047; Hypothetical protein		;00548515		Q8G7U4_BILO0(BIFIDOBACTERIUM_LONGUM_NCC2705)
HBG402232	; 00571143; 00572680; Hypothetical protein			;00569729	UXAC_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG402496	; 00567952; 00567955; Hypothetical protein	FRAAL_PE4043			Y088_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Y088_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Y088_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG402502	; 00567681; 00567752; 00567786; 00569022; 00569877; 00570173; Hypothetical protein	FRAAL_PE5956	;00547009		Y090_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) Y090_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Y090_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG403525	; 00567060; 00570372; Hypothetical protein	FRAAL_PE4817			Y960_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) Y960_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) Y960_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG404412	; 00567075; 00567794; 00570176; Hypothetical protein		;00547584		YH38_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YH38_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YH38_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) YQ32_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97) YQ32_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YQ32_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV)
HBG405186	; 00567783; 00570181; Hypothetical protein	FRAAL_PE553			YY07_MYTU1(MYCOBACTERIUM_TUBERCULOSIS_H37RV) YY07_MYTU0(MYCOBACTERIUM_TUBERCULOSIS_CDC1551) YY07_MYBO0(MYCOBACTERIUM_BOVIS_AF2122/97)
HBG370199	; 00570337; Hypothetical protein; Similar to the 2-hydroxy-6-ketonona-2			;00573002	Q9CB21_MYLE0(MYCOBACTERIUM_LEPRAE_TN)
HBG336838	; 00570336; 00573342; Hypothetical protein		;00547620		Q8NMR9_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG337048	; 00570185; Hypothetical protein		;00547965	;00567209	Q8NPQ0_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG337212	; 00570186; Hypothetical protein		;00548926	;00573169	Q8NR94_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG337328	; 00567688; 00571406; Hypothetical protein		;00547141	;00570160	Q8NT18_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG337379	; 00546411; Hypothetical protein		;00547315	;00567892;00571209	Q8NTR5_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG337410	; 00568953; Hypothetical protein			;00571204	Q8NU58_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG336710	; 00569251; Hypothetical protein			;00570052	Q8NL87_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG336788	; 00570264; 00574225; Hypothetical protein			;00570553	Q8NM60_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG336790	; 00570031; 00572250; Putative HTH-type transcriptional regulator Rv0043c/MT0049/Mb0044c			;00570554	Q8NM62_COGL0(CORYNEBACTERIUM_GLUTAMICUM_ATCC_13032)
HBG327414	; 00546272; 00572899; Hydrolase; putative; Hypothetical protein			;00569815	Q8G3K3_BILO0(BIFIDOBACTERIUM_LONGUM_NCC2705)
HBG327466	; 00567144; Hypothetical protein		;00545448		Q8G3Z0_BILO0(BIFIDOBACTERIUM_LONGUM_NCC2705)

famille hogenom	annotations hogenom	sequences from AIni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG327518	; 00570331; 00572192; Hypothetical protein		;00547302		Q8G4A6_BILO0(BIFIDOBACTERIUM_LONGUM_NCC2705)
HBG325940	; Hypothetical protein			;00570510	Q8FPM1_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG326007	; 00566956; Hypothetical protein		;00548841		Q8FQ77_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG326008	; 00566953; 00567094; Hypothetical protein		;00548842		Q8FQ78_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG326040	; 00569098			;00572070	Q8FQC9_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG326079	; 00571464			;00574188	Q8FQP9_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG325678	; Hypothetical protein		;00545462		Q8FM23_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG325774	; 00547313; 00567120; 00571242; 00574532			;00569728	Q8FMZ1_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG325814	; 00570704			;00569154	Q8FN95_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG325824	; 00570705; Putative ATP-dependent helicase; Superfamily II DNA/RNA helicases; SNF2 family			;00569008	Q8FNC5_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG317514	; 00547007; IS1561'; transposase; TRANSPOSASE			;00569683	Q8CM04_COEF0(CORYNEBACTERIUM_EFFICIENS_YS-314)
HBG262916	; 00548669; 00569470			;00567144	Q745A6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262927	; 00545718; 00570852			;00570331; 00572192	Q745C7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262893	; 00545731; 00549353; Hypothetical protein Rv2595/MT2671.1/Mb2626			;00574225	Q744U4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262864	; Cytochrome P450; Probable cytochrome P450 monooxygenase 142 CYP142B			;00568953	Q744K2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262878	; 00569242; Hypothetical protein Rv2599/MT2674 precursor; Probable conserved membrane protein			;00569251	Q744Q1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262806	; 00549134; 00571891; Hypothetical protein MG371 homolog; Protein mgpA			;00571406	Q743A7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262803	; 00545794; 00572640; Hypothetical protein Rv1480/MT1527/Mb1516			;00570185	Q743A4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262804	; 00568763; 00568815; Hypothetical protein Rv2568c/MT2644/Mb2598c			;00570186	Q743A5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262801	; 00571244			;00573342	Q742Z3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262793	; 00548588; 00568301; 00570735			;00567794; 00570176	Q742X0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262795	; 00547428; 00568924; Hypothetical protein Rv1827/MT1875/Mb1858			;00570181	Q742X3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262790	; 00546895; 00570856; Hypothetical protein; Putative membrane protein			;00567952; 00567955	Q742W0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262791	; 00547081; 00574108			;00570173	Q742W6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262792	; 00545574; 00571223; Hypothetical protein MJ1220; Putative type I restriction enzyme HindVIIP M prote			;00570372	Q742W8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262779	; 00567466			;00570046	Q742T9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262782	; 00548958; 00571265; 00571451			;00570030; 00570047	Q742U5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262785	; 00569683; Hypothetical protein			;00572680	Q742U9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262771	; 00546484; 00547838; 00548937; 00549401; 00566632; 00567548; 00567550; 00567599; 00567643; 00569487;			;00570209; 00572462	Q742R5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262778	; 00545431; 00548933; 00566599; 00566623; Transposase insI for insertion sequence element IS30B/C/D			;00570180	Q742T8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262602	; 00546760; 00572981; Glycine/D-amino acid oxidases; Hypothetical protein			;00571683	Q740H3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262601	; 00545462; Hypothetical protein			;00569669	Q740H2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262574	; 00569243; Hypothetical protein MAP1032c; Hypothetical protein Rv2600/MT2674.1/Mb2631			;00573356	Q740A5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262583	; 00569728; Hypothetical protein			;00569316	Q740C1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262556	; 00548209; Hypothetical membrane protein; Hypothetical protein		;00546328		Q73ZV2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262498	; 00569154; Putative superoxide dismutase			;00570685	Q73Z98_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262479	; 00569008; Putative phenol 2-monooxygenase			;00570674	Q73Z53_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)

famille hogenom	annotations hogenom	sequences from Alni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG262485	; 00548201; Methionine synthase II; Putative epoxyalkane:coenzyme M transferase			;00569007	Q73Z70_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262494	; 00570510; Putative tyrosinase		;00549697		Q73Z92_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262436	; 00548841; Hypothetical protein		;00549667		Q73YR0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262406	; 00548842; Hypothetical protein			;00568990	Q73YC7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262414	; 00572070; Putative ABC transporter ATP binding protein		;00545587;00547274		Q73YE0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262384	; 00547236; Hypothetical protein; Phosphoglycerate mutase/fructose-2;6-bisphosphatase		;00548696		Q73Y82_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262349	; 00574188; Hypothetical protein			;00571732	Q73XX1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262350	; Conserved hypothetical protein; Hypothetical protein Cgl0617; Hypothetical protein MT3501			;00571725	Q73XX7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262364	; 00569855; Beta-glucosidase-related glycosidases; Putative beta-glucosidase		;00547307;00548591		Q73Y10_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262328	; 00548356; Hypothetical protein Cgl1404		;00549760		Q73XL6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262331	; 00548358; Hypothetical protein Cgl1403			;00571017	Q73XP7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262308	; 00548360; Hypothetical protein Cgl1402			;00571427	Q73XF6_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262310	; 00571449; Hypothetical protein			;00567460;00573853	Q73XG0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262292	; 00549625; 00567870			;00567099	Q73XC2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262293	; 00569815; Possible TetR-type transcriptional regulator			;00567469	Q73XC3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262306	; 00545448; Narrowly conserved hypothetical protein in transglutaminase family			;00568033;00571736	Q73XF4_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262279	; 00547302; Narrowly conserved hypothetical protein			;00566690	Q73X97_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262243	; 00549447; 00569592; Hypothetical protein			;00568989	Q73WY1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262242	; 00570052; Hypothetical protein Cgl3061			;00567298	Q73WX9_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262217	; 00573567; Putative HTH-type transcriptional regulator yxaD			;00567462;00569290;00570201;00570981	Q73WK1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262193	; 00570553; Hypothetical protein Cgl2720			;00573774	Q73WB2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262168	; 00570554; Hypothetical protein Cgl2718			;00567777	Q73W79_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262169	; 00547620; Regulator of polyketide synthase expression			;00567730	Q73W82_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262172	; 00547965; 00567209; Hypothetical protein Cgl1761			;00569080;00570271	Q73W85_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262156	; 00567693			;00570846	Q73VZ0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262139	; 00548926; 00573169			;00568604	Q73VQ0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262119	; 00547141; 00570160; Hypothetical protein Cgl0496			;00568196	Q73VG3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262107	; 00547315; 00567892; 00571209; Hypothetical protein Cgl0237			;00572916	Q73VB8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262099	; 00571204; Hypothetical protein Cgl0082			;00570305	Q73V90_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262106	; 00570578			;00572915	Q73VB7_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262091	; 00545708; 00546490; 00546728; 00548417; 00548901; 00549192; 00549193; 00549194; 00568064; 00573649			;00567684;00569019	Q73V76_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262097	; 00546152; Hypothetical protein HI1053; Hypothetical protein MJ0742; Hypothetical protein MJ1511			;00566952;00569061	Q73V88_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262090	; Ferrichrome-binding periplasmic protein; Hypothetical protein; Iron ABC transporter; solute-binding			;00572678	Q73V75_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG262036	; 00546024; 00546254; 00548512; 00548623; Hypothetical protein HI0322; Hypothetical protein HI0947; Hy			;00569167;00569849;00572230	Q73UV5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261984	; 00549722; Hypothetical protein Rv2265/MT2327/Mb2288			;00568151	Q73U60_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261985	; 00569214; 00571794			;00568153	Q73U62_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)

famille hogenom	annotations hogenom	sequences from AIni	sequences from Cci3	sequences from EAN	sister actinobacteria in tree
HBG261969	; 00569452; Hypothetical protein ML0607; Hypothetical protein Rv2569c/MT2645/Mb2599c			;00572507	Q73TY5_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261976	; 00569450; Hypothetical protein ML0605; Hypothetical protein Rv2411c/MT2484/Mb2434c; Hypothetical pro			;00570262	Q73U19_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261967	; 00548903; 00548988; 00549179; 00549357; 00569941; 00571089; 00571516; 00573688; Hypothetical protein			;00572506	Q73TY2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261917	; 00547574; 00570503		;00546183		Q73TE2_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261934	; 00571279; Hypothetical protein yqjT			;00566645	Q73TI8_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261883	; 00566606; 00571713; 00573002; Possible enoyl-CoA hydratase			;00574192	Q73T31_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261884	; 00548515; Hypothetical protein MJ0425; Hypothetical protein MJECL04			;00574199	Q73T32_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261855	; 00569729; Uronate isomerase			;00572907	Q73SR3_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261801	; Hypothetical protein Rv0088/MT0096/Mb0091			;00572225	Q73RZ1_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG261786	; 00547009; Hypothetical protein Rv0090/MT0099/Mb0093			;00573662	Q73RW0_MYAV0(MYCOBACTERIUM_AVIUM_SUBSP._PARATUBERCULO)
HBG252802	; Hypothetical protein Rv0960/MT0988/Mb0985			;00568264	Q6NK55_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG252585	; 00547584; Hypothetical protein Rv1738/MT1780/Mb1767; Hypothetical protein Rv2632c/MT2708/Mb2665c			;00571317	Q6NIU2_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)
HBG252134	; Hypothetical protein Rv3407/MT3515/Mb3441		;00547250		Q6NF29_CODI0(CORYNEBACTERIUM_DIPHTHERIAE_NCTC_13129)

Annexe C
Articles

1 Article1 : Bioinformatic sequence identification from sequence family databases

Arigon A.M., Perrière G. and Gouy M. (2005) Bioinformatic sequence identification from sequence family databases. In Actes des 6èmes Journées Ouvertes : Biologie, Informatique et Mathématiques, Perrière, G., Guénoche, A. et Geourjon C. (eds.), CNRS, Lyon, pp. 213-220.

Bioinformatic sequence identification from sequence family databases

Anne-Muriel Arigon[1], Guy Perrière[1] and Manolo Gouy[1]

[1] Laboratoire de Biométrie et Biologie Évolutive, UMR CNRS 5558, Université Claude Bernard – Lyon I, 43 bd. du 11 Novembre 1918, 69622 VILLEURBANNE Cedex, France
arigon@biomserv.univ-lyon1.fr
perriere@biomserv.univ-lyon1.fr
mgouy@biomserv.univ-lyon1.fr

Abstract: *We have developed a tool in order to identify sequences in relation to a sequence family database. This tool combines several algorithms: BLAST, multiple sequence alignment and phylogenetic tree building. After identification of the most similar gene family to the query sequence, this query sequence is added to the whole family alignment and the phylogenetic tree of the family is rebuilt including the query sequence. Thus, the query sequence can be easily located in its gene family.*

Keywords: identification, similarity search, alignment, phylogeny.

1 Introduction

In several biological contexts such as species or taxon identification from molecular markers of environmental organisms, confrontation of a new sequence to those of a database, or sequence database update, the classification of a new sequence compared to a sequence collection can be useful. Algorithms and bioinformatic tools are necessary to carry out these operations in a relevant and fast way.

Tools such as BIBI (Bioinformatic Bacterial Identification) [1] were mainly developed for pathogenic bacteria identification. This program uses BLAST [2,3] to perform the similarity search. Then multiple sequence alignment programs such as CLUSTAL W [4] are used to align the most similar sequences to the query sequence and finally to build the corresponding phylogenetic tree. The limiting stage of the identification process is alignment computation. Moreover, such a system is practical for on-line use because the databases on which it works are of small size. Thus, results can be obtained in a time acceptable to the user. This is related to the fact that it is an identification tool: it is enough to recover only a small number of sequences, presenting the strongest similarities with the query sequence, before computing the alignment.

With existing algorithms, the principle used by BIBI is thus inoperative for large sequence family databases such as HOVERGEN [5], or HOBACGEN [6]. In these databases, sequences are grouped in homologous families and prealigned. These databases can also be used as tools for phylogenetic analyses. The addition of only one sequence can have many repercussions on the topology of the phylogenetic tree associated to a given family. These changes may be located in the tree near the query sequence, but they may also be located at the level of major nodes. In such a case, it is thus impossible to limit comparisons to the most similar sequences, and it is the whole family which should be taken into account. Moreover, the HOVERGEN and HOBACGEN databases contain large families, sometimes composed of several thousand sequences. With such families, algorithms like CLUSTAL W and MULTALIN [7] are unable to deal with the problem of the realignment in times which are compatible with interactive use.

2 Implementation

We built a bioinformatic tool allowing the automatic identification of homologous sequences and their classification inside large sequence databases. The implemented program is composed of three stages. First, from an unknown sequence, we developed an identification algorithm using comparison and similarity search tools to find the gene family to which this sequence belongs. Second, it is necessary to use fast algorithms making it possible to align a large number of sequences or to gradually add a sequence to a preexisting alignment without recomputing all of the alignment. Lastly, the phylogenetic tree must be rebuilt by including the unknown sequence. This rebuilding must be fast and possible with a large number of sequences. Our tool thus contains several algorithms written in standard language ANSI C integrating different programs of similarity search, multiple alignments and phylogenetic tree rebuilding. These developments are integrated in a web application implemented in HTML-php, which makes it possible to have an interface with simple and fast use.

The identification algorithm uses BLAST to compare the unknown protein or nucleotide sequence with the sequences of the protein family database chosen by the user. We have chosen to work with protein databases because similarity search from a protein sequence is more sensitive than comparisons made from a nucleotide sequence. The BLAST result is analyzed in order to identify potential families of the unknown sequence. For each database sequence that matches with the unknown sequence at an e-value lower than a threshold (the best match e-value x 1e5), we determine its family and its BLAST score. For each family thus identified, a weighted average of the scores is calculated. The selected family is that with the highest average score. We also locate the non-overlapping matches of the sequences of potential families to the unknown sequence. All distinct families with non-overlapping matches are selected. Finally, if several families have the same average score, the families are proposed to the user. Then, the user can choose what family to use. The interface provides links to BLAST output and to information about proposed families in order to assist the user choices.

For each selected family, a set of multiple alignment programs is proposed to the user. The majority of these programs follow the progressive alignment heuristic approach. This approach is very fast, requires a moderate memory space and offers good performances with relatively well-conserved sequences. Several multiple alignment programs use this approach like CLUSTAL W, MULTALIN, MENTALIGN [8] and MUSCLE [9]. The latter proposes two more specific uses of the program, MUSCLE-prog and MUSCLE-fast. They make it possible to align a large number of sequences much more quickly than with other programs (MUSCLE-fast is faster than MUSCLE-prog but less accurate). Other programs such as MABIOS [10] use algorithms of alignment by blocks and obtain good results for homologous sequences. All these programs also make it possible to very quickly add a sequence to a preexisting alignment. For this, the majority of these programs use the principle of profile-profile alignment. According to the selected family, the proposed list of alignment programs varies. Indeed, problems (e.g., too slow execution for a web application, insufficient memory) may occur when some programs such as CLUSTAL W and MABIOS are used to add a sequence to an alignment of more than 500 sequences. In order to help the user in his/her choice, links to BLAST output and to information about each selected family are given.

The obtained alignment is used as input for the phylogenetic tree building program. In our case, we require a program which guarantees a great speed of execution even if the quality of tree is not optimal. Indeed, the aim of this tool is to allow the user to locate approximately the unknown sequence in the selected family. Thus we have chosen to use QuickTree [11]. This phylogenetic tree rebuilding program is a fast implementation of the Neighbor-Joining algorithm [12]. It allows a rapid phylogenetic rebuilding of large sequence families. This program provides an option to bootstrap which makes it possible to have a certain validation of the obtained tree. Moreover, this option is used only when the family contains less than 100 sequences to keep short running times. The tree obtained by QuickTree is then rooted at its center.

All results are presented in an HTML page. The user can visualize the BLAST output and information about each selected family. Obtained alignments and phylogenetic trees can be displayed by two Java applets

(Fig. 1): Jalview ([http://www2.ebi.ac.uk/~michele/jalview/]) and ATV [13].

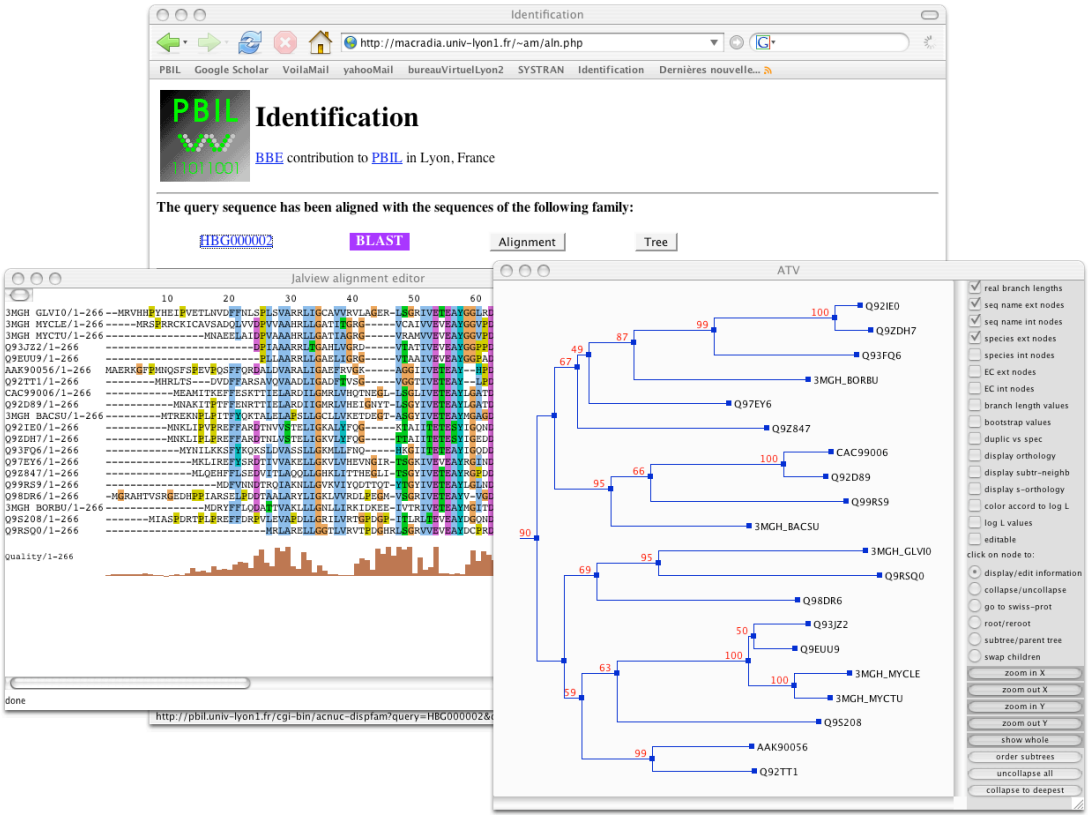

Figure 1. Screenshot of the web application.

3 Perspectives

In databases such as HOBACGEN and HOVERGEN, alignments and trees are precomputed for families of less than 500 sequences. However, in the case of a family of several thousand sequences, the family multiple alignment can take several tens of minutes. With a web application, such running times are not acceptable. We thus envisage to use the algorithms integrated into our identification tool to precompute alignments and trees of large families of these databases. Moreover, an algorithm to add a sequence to a preexisting tree could be integrated into our identification tool in order to avoid recomputing completely trees of large sequence families. Thus, the necessary time for the on-line identification could be improved.

References

[1] G. Devulder, G. Perrière, F. Baty and J.P. Flandrois, BIBI, a bioinformatic bacterial identification tool. *J. Clin. Microbiol.*, 41, 1785-1787, 2003.

[2] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J. Lipman, Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403-10, 1990.

[3] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller and D.J. Lipman, Gapped BLAST and PSI-BLAST : a new generation of protein database search programs. *Nucleic Acids Res.*, 25, 3389-3402, 1997.

[4] J.D. Thompson, D.G. Higgins and T.J. Gibson, CLUSTAL W : improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic*

Acids Res., 22, 4673-4680, 1994.
[5] L. Duret, G. Perrière and M. Gouy, HOVERGEN : database and software for comparative analysis of homologous vertebrate genes. *In Bioinformatics Databases and Systems*, Letovsky, S. (ed.), Kluwer Academic Publishers, Boston, MA, pp. 13-29, 1999.
[6] G. Perrière, L. Duret and M. Gouy, HOBACGEN : database system for comparative genomic s in bacteria. *Genome Res*., 10, 379-385, 2000.
[7] F. Corpet, Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res*., 16, 10881-10890, 1988.
[8] J.F. Dufayard, *Incremental algorithms for the alignment and the phylogeny of large homologous sequence families*, Ph.D., Joseph Fourier University – Grenoble, 2004.
[9] R.C. Edgar, MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics,* 5(1):113, 2004.
[10] S. Abdeddaïm, *Biological sequences comparison*, Ph.D., University of Paris, 1996.
[11] K. Howe, A. Bateman and R. Durbin, QuickTree: building huge Neighbour-Joining trees of protein sequences. *Bioinformatics* 18(11):1546-1547, 2002.
[12] N. Saitou and M. Nei, The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol*. 4(4):406-425, 1987.
[13] C.M. Zmasek and S.R. Eddy, ATV : display and manipulation of annotated phylogenetic trees. *Bioinformatics* 17 :383-384, 2001.

2 Article2 : HoSeqI : automated homologous sequence identification in gene family databases

Arigon AM, Perriere G, Gouy M. (2006) HoSeqI : automated homologous sequence identification in gene family databases. Bioinformatics. 22(14) :1786-7.

BIOINFORMATICS APPLICATIONS NOTE Vol. 22 no. 14 2006, pages 1786–1787
doi:10.1093/bioinformatics/btl179

Phylogenetics

HoSeqI: automated homologous sequence identification in gene family databases

Anne-Muriel Arigon*, Guy Perrière and Manolo Gouy

Laboratoire de Biométrie et Biologie Évolutive, UMR CNRS 5558, Université Claude-Bernard Lyon 1, 43 boulevard du 11 Novembre 1918, 69622 Villeurbanne Cedex, France

Received on February 17, 2006; revised on April 19, 2006; accepted on May 3, 2006
Advance Access publication May 8, 2006
Associate Editor: Joaquin Dopazo

ABSTRACT

Summary: We present a web service allowing to automatically assign sequences to homologous gene families from a set of databases. After identification of the most similar gene family to the query sequence, this sequence is added to the whole alignment and the phylogenetic tree of the family is rebuilt. Thus, the phylogenetic position of the query sequence in its gene family can be easily identified.

Availability: http://pbil.univ-lyon1.fr/software/HoSeqI/

Contact: arigon@biomserv.univ-lyon1.fr

Supplementary Information: Supplementary Data are available at *Bioinformatics* online.

In several contexts such as (1) species or taxon identification from molecular markers of environmental organisms, (2) confrontation of a new sequence to a database, or (3) update of homologous gene family sequence databases, the classification of a new sequence into a collection is needed. This classification allows the identification of which family the sequence belongs to and contributes to the assessment of its evolutionary relationships. Today, massive sequencing techniques are routinely used and the number of new available sequences grows up quickly. The identification tasks require the chaining of different programs (for similarity search, alignment and tree computation) that are sometimes complex to handle. Moreover, some results have to be manually checked. Doing these tasks sequentially makes the work of sequence identification tedious and time-consuming. Automated bioinformatic tools are thus necessary to carry out these operations in an accurate and fast way. Some tools exist to make sequence identification. For instance, BIBI (Bioinformatic Bacterial Identification) (Devulder *et al*., 2003) was specifically developed for bacterial identification. The European ribosomal RNA database (Wuyts *et al*., 2004) compiles complete or nearly complete ribosomal RNA (rRNA) sequences and uses the BIBI algorithm in order to allow a user to make rRNA quick phylogeny analyses. The Ribosomal Database Project (Cole *et al*., 2005) proposes a database with aligned and annotated rRNA gene sequences and provides analysis services such as the RDP classifier that places sequences in the RDP hierarchy in order to give an initial taxonomic placement for sequences.

We developed comprehensive sequence family databases (i.e. HOVERGEN and HOGENOM) (Duret *et al*., 1999) in which homologous protein gene sequences are clustered into families and aligned. These databases can be used for different purposes, among which are phylogenetic analyses. The addition of a single sequence to a given family from these databases can have many repercussions on the topology of the associated phylogenetic tree; these changes may be located near the introduced sequence, but they may also be located in deep nodes. In such case, the phylogenetic information brought by the whole family should be taken into account. Also, as HOVERGEN and HOGENOM contain large families, with several thousand sequences, powerful algorithms are required in order to quickly add a sequence to a large alignment. Currently available sequence identification tools such as those presented previously are developed to treat specific data such as rRNA sequences. BIBI algorithm limits comparisons to the most similar sequences and the RDP classifier uses a naïve Bayesian rRNA classifier. So they cannot be used effectively with large family databases such as HOVERGEN and HOGENOM.

We built a software environment—called HoSeqI (Homologous Sequence Identification)—allowing the automatic identification of homologous sequences and their classification into our sequence family databases. It integrates different programs of similarity search, multiple alignments and phylogenetic tree building, as well as specific tools we developed. This environment can be accessed through a web service implemented in HTML-PHP. It is divided into three parts. First, the identification procedure uses BLASTP (Altschul *et al*., 1997) to compare the query sequence with the entries of the family database chosen by the user. BLASTP outputs are parsed in order to identify to which families the submitted sequence belongs. All distinct families with non-overlapping matches are selected allowing to process sequences that contain non-overlapping regions from distinct homologous gene families. If several families are identified, they are all proposed to the user who can then choose which one to select. The interface provides links to BLASTP output and information about proposed families in order to assist user choices.

Second, for each identified family, a set of multiple alignment programs is proposed to the user CLUSTAL W (Thompson *et al*., 1994), MULTALIN (Corpet, 1988), MABIOS (Abdeddaïm, 1997), MENTALIGN (Dufayard, 2004) and MUSCLE (Edgar, 2004). MENTALIGN is an incremental algorithm that has been developed specifically by our group in order to manage very large alignments and trees containing thousands of sequences. MUSCLE proposes two specific uses of the program, MUSCLE-prog and MUSCLE-fast allowing to align a large number of sequences much more quickly than with other programs. All these programs also make

*To whom correspondence should be addressed.

it possible to very quickly add a sequence to a pre-existing alignment. The HOVERGEN and HOGENOM databases contain all multiple alignments and phylogenetic trees for families of <500 sequences. So, the query sequence can be easily added to alignments of these families. For other families (>500 sequences), the whole sequence alignment has to be computed. According to the identified family, the proposed list of alignment programs varies. Indeed, problems may occur when some programs such as CLUSTAL W and MABIOS are used to compute a multiple alignment containing >500 sequences (e.g. execution is too slow for a web application, memory allocation).

Lastly, the obtained alignment is used to build the phylogenetic tree. The user can choose among the following tree building programs: QuickTree (Howe *et al*., 2002), FastME (Desper and Gascuel, 2002), BIONJ (Gascuel, 1997) and PhyML (Guindon and Gascuel, 2003). QuickTree is a fast implementation of the neighbor-joining (NJ) algorithm (Saitou and Nei, 1987). It allows a rapid phylogenetic rebuilding for large sequence families. FastME is based on the minimum evolution method. BIONJ is an improved version of the NJ algorithm. PhyML is able to compute large phylogenies by maximum likelihood. When the input of the phylogenetic tree program has to be a distance matrix, we use PROTDIST [with Kimura's formula (Kimura, 1983)] to compute it (Felsenstein, 1989). For each program, the user can apply the bootstrap option. The tree is then automatically rooted at its mid-point.

For all programs used in HoSeqI (BLASTP, multiple alignment programs and phylogenetic tree building programs), the interface allows to choose non-default parameter values. All results are presented through web pages and can be downloaded. Resulting alignments and phylogenetic trees can also be displayed by two Java applets: Jalview (http://www2.ebi.ac.uk/~michele/jalview/) and ATV (Zmasek and Eddy, 2001). Some selected options can result in time-consuming alignment and phylogenetic tree building (e.g. if the user chooses PhyML with the bootstrap option). In these cases, computations are performed offline and the user receives an e-mail with links to the various results that are kept on the server for one month.

The usefulness of HoSeqI is to automate the identification process on large family databases and to contribute to the study of the evolutionary background of new sequences. HoSeqI proposes a user-friendly interface that allows a user to easily identify a query sequence and to visualize the obtained alignment and tree. The user can thus locate the sequence in the tree of its gene family and study the evolution of this new sequence. Computation times range between 30 s (for 143 sequences in the associated family) and 2 min 30 s (for 1132 sequences in the associated family).

Conflict of Interest: none declared.

REFERENCES

Abdeddaïm,S. (1997) Fast and sound two-step algorithms for multiple alignment of nucleic sequences. *Int. J. Artif. Intell. Tools*, **6**, 179–192.

Altschul,S.F. *et al*. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.

Cole,J.R. *et al*. (2005) The Ribosomal Database Project (RDP-II): sequences and tools for high-throughput rRNA analysis. *Nucleic Acids Res*., **33**, D294–D296.

Corpet,F. (1988) Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res*., **16**, 10881–10890.

Desper,R. and Gascuel,O. (2002) Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *J. Comput. Biol*., **19**, 687–705.

Devulder,G. *et al*. (2003) BIBI, a bioinformatic bacterial identification tool. *J. Clin. Microbiol*., **41**, 1785–1787.

Dufayard,J.F. (2004) Incremental algorithms for the alignment and the phylogeny of large homologous sequence families. Ph.D. Thesis, Joseph Fourier University, Grenoble, France.

Duret,L., Perrière,G. and Gouy,M. (1999) HOVERGEN: database and software for comparative analysis of homologous vertebrate genes. In Letovsky,S. (ed.), *Bioinformatics Databases and Systems*. Kluwer Academic Publishers, Boston, MA, pp. 13–29.

Edgar,R.C. (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, **5**, 113.

Gascuel,O. (1997) BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol. Biol. Evol*., **14**, 685–695.

Guindon,S. and Gascuel,O. (2003) A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol*., **52**, 696–704.

Howe,K. *et al*. (2002) QuickTree: building huge Neighbour-Joining trees of protein sequences. *Bioinformatics*, **18**, 1546–1547.

Felsenstein,J. (1989) PHYLIP—Phylogeny Inference Package (Version 3.2). *Cladistics*, **5**, 164–166.

Kimura,M. (1983) *The Neutral Theory of Molecular Evolution*. Cambridge University Press, Cambridge.

Saitou,N. and Nei,M. (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol*., **4**, 406–425.

Thompson,J.D. *et al*. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acids Res*., **22**, 4673–4680.

Wuyts,J. *et al*. (2004) The European ribosomal RNA database. *Nucleic Acids Res*., **32**, D101–D103.

Zmasek,C.M. and Eddy,S.R. (2001) ATV: display and manipulation of annotated phylogenetic trees. *Bioinformatics*, **17**, 383–384.

RESUME en français

Le nombre de séquences génomiques disponibles augmente très vite du fait du développement de méthodes de séquençage massif. La classification de ces séquences est nécessaire et permet l'étude de leurs relations évolutives. Des outils bioinformatiques automatisés sont indispensables pour effectuer ces opérations d'identification de façon précise et rapide. Nous avons développé HoSeqI (Homologous Sequence Identification), un système permettant d'automatiser l'identification de séquences dans de grandes banques de familles de gènes homologues. HoSeqI propose une interface accessible sur internet (http ://pbil.univ-lyon1.fr/software/HoSeqI/) afin d'identifier une séquence et de visualiser l'alignement et la phylogénie obtenus. Un autre programme, dérivé d'HoSeqI, a été implémenté pour l'ajout automatique de séquences génomiques aux banques de familles. Enfin, un travail sur l'identification automatique de séquences bactériennes d'ARN 16S et la détection de séquences chimères a été effectué.

TITRE en anglais

Development of automated identification tools using large gene family databases

RESUME en anglais

The number of available genomic sequences is growing very fast, due to the development of massive sequencing techniques. Sequence classification is needed and contributes to the assessment of their evolutionary relationships. Automated bioinformatics tools are thus necessary to carry out these identification operations in an accurate and fast way. We developed HoSeqI (Homologous Sequence Identification), a software environment allowing this kind of automated sequence identification using homologous gene family databases. HoSeqI is accessible through a Web interface (http ://pbil.univ-lyon1.fr/software/HoSeqI/) allowing to identify a sequence and to visualize obtained alignment and phylogeny. We also implemented another set of programs - derivated from the one used in HoSeqI - in order to automatically add genomic sequences to family databases. At last, some developments aimed at automated identification of 16S RNA bacterial sequences and detection of chimeric sequences.

DISCIPLINE

Bioinformatique

MOTS-CLES

Identification automatique, similarité, alignement, phylogénie, chimère.

INTITULE ET ADRESSE DE L'U.F.R. OU DU LABORATOIRE :

Laboratoire de Biométrie et Biologie Evolutive - UMR 5558 CNRS
Batiment Gregor Mendel - Université Claude Bernard Lyon1
43, bv du 11 novembre 1918 - 69622 Villeurbanne cedex