Travail d'expertise interdisciplinaire

Auteur : Mullender, Coralie
Promoteur(s) : Baurain, Denis; Lupo, Valérian
Faculté : Faculté des Sciences
Diplôme : Master en bioinformatique et modélisation, à finalité approfondie
Année académique : 2023-2024
URI/URL : http://hdl.handle.net/2268.2/20539

Université de Liège

Département des Sciences de la Vie - Faculté des Sciences

Phylogénie structurale des Mur ligases bactériennes et archéennes.

Mémoire en vue de l'obtention du grade de Master en Bioinformatique et Modélisation, à finalité approfondie (SMUBIF019901).

MULLENDER Coralie (S150470)
Promoteur: BAURAIN Denis
Co-Promoteur: LUPO Valérian

Laboratoire de Phylogénie des Eucaryotes
InBios

Juin 2024

Table des matières

1 Remerciements

2 Liste des abréviations

3 Résumé

4 Introduction **1**
4.1 L'arbre de la vie : Bactéries et Archées en confrontation . . . 1
4.2 Peptidoglycane et pseudomuréine : Diversité dans les parois cellulaires . . . 2
4.3 Phylogénie structurale . . . 8

5 Objectifs **9**

6 Matériels et Méthodes **10**
6.1 Environnement . . . 10
6.1.1 Durandal . . . 10
6.1.2 AIDA . . . 10
6.1.3 Ordinateur personnel . . . 10
6.2 Programmes . . . 10
6.3 Schéma récapitulatif des manipulations . . . 12
6.4 Protocole manuel (Distance RMSD) . . . 12
6.4.1 Sélection et prédiction des structures tridimensionnelles . . . 12
6.4.2 Extraction des coordonnées des domaines . . . 15
6.4.3 Prédiction des distances RMSD sur Pymol . . . 15
6.4.4 Obtention des arbres sur R studio . . . 16
6.4.4.1 Générer les données en bash . . . 16
6.4.4.2 Statistiques descriptives . . . 17
6.4.4.3 SDM et arbres phylogénétiques . . . 17
6.4.5 Jackknife . . . 18
6.5 FoldTree (lDDT, MT et Fident) . . . 18
6.5.1 Phylogénie structurale des archées et des bactéries . . . 19
6.5.2 Jackknife . . . 19
6.6 Consurf . . . 20

7 Résultats **20**
7.1 Protocole manuel (Distance RMSD) . . . 20
7.1.1 Sélection et prédiction des structures tridimensionnelles . . . 20
7.1.2 Matrice RMSD . . . 21
7.1.2.1 Statistiques descriptives . . . 21
7.1.2.2 Distributions des distances RMSD inter-familles. . . . 22
7.1.2.3 Distribution des distances RMSD intra-familles. . . . 23
7.1.2.4 Boxplots des valeurs RMSD. . . . 24
7.1.2.5 SDM et arbres phylogénétiques . . . 26
7.2 Phylogénie structurale des Mur ligases par FoldTree . . . 29

7.3 Comparaison des différentes approches phylogénétiques . . . 31
7.3.1 Phylogénie structurale des archées et des bactéries . . . 35
7.4 Jacknife de séquences . . . 38

8 Discussion **39**
8.1 Comparaison du protocole (RMSD) et de FoldTree . . . 39
8.2 Phylogénie des Mur ligases et voie de biosynthèse . . . 40
8.2.1 Vue globale des distributions inter-familles . . . 40
8.2.2 Analyses séparées des domaines . . . 41
8.2.2.1 Domaine 2 . . . 41
8.2.2.2 Domaine 1 . . . 41
8.2.2.3 Domaine 3 . . . 43
8.2.3 Voie de biosynthèse des peptides de la PM . . . 44

9 Perspectives **44**

10 Annexes
10.1 Images supplémentaires . . .
10.2 Programmes . . .
10.2.1 (2) hmms-parser-coord.pl . . .
10.2.2 (3) pymol-script.pl . . .
10.2.3 (5) pymol-script.pml . . .
10.2.4 (7) analyses-RMSD.R . . .
10.2.5 (8) analyses-SDM.R . . .
10.2.6 (9) jack-dist.R . . .
10.2.7 (10) SnakeM_myfam.pl . . .
10.2.8 (11) foltree.sh . . .
10.2.9 (14) compo_murligase.R . . .

Références

1 Remerciements

Ce travail, qui représente l'aboutissement d'un deuxième master, a pour moi été une opportunité immense de développer de nouvelles compétences et de continuer mon développement scientifique intellectuel. Il représente également une possibilité de modifier mon cap professionnel en ajoutant la bioinformatique à mon master de Biologie des Organismes et Écologie obtenu précédemment. Ce master ne m'a pas permis uniquement de développer mes compétences informatiques, il m'a également permis de compléter mes connaissances biologiques. En effet, lors de ce master, j'ai eu l'occasion de suivre des cours de génomique qui m'ont aidé à boucher (en partie) les lacunes présentes en biologie moléculaire et cellulaire. Ce nouveau cursus additionné au précédent m'offre une vision plus globale de la Biologie.

Tout d'abord, je tiens à remercier tout particulièrement le professeur D. Baurain pour la chance qu'il m'a donnée en acceptant ma demande de mémoire dans son laboratoire et pour son écoute. Je le remercie également lui, ainsi que V. Lupo pour leur encadrement tout au long de ce travail, leur gentillesse, leur aide, leur patience et leur bonne humeur. Au-delà de ce mémoire, pour les cours qu'il dispense ainsi que pour les autres, le professeur Baurain est toujours disponible et prêt à aider. Cette disponibilité mérite également des remerciements. Je remercie aussi F. Kerff pour ses précieux conseils.

Ensuite, mes remerciements vont également à Marie Harmel avec qui j'ai partagé le local et mon temps.

Finalement, je tiens à remercier mes proches, mes parents, Ilyass Fraterrigo et Sébastien Mirolo pour leur soutien sans faille.

2 Liste des abréviations

Abréviation	Nom complet	Nom Français
AAs	amino acids	acides aminés
LBCA	Last Bacterial Commun Ancestor	Dernier ancêtre commun bactérien
lDDT	local Distance Difference Test	Test local de Différence de Distance
LUCA	Last Universal Commun Ancestor	Dernier ancêtre commun universel
NAT	N-acetyltalosaminuronic acid	acide N-acétyltalosaminuronique
NJ	Neighbour joining	/
OTU	Operational Taxonomic Unit	Unité Taxonomique Opérationnelle
PFTE	Pairwise Fractional Topological Equivalence	équivalence topologique fractionnaire par paire
PG	Peptidoglycan - murein	Peptidoglycane - muréine
PM	Pseudomurein - Pseudopeptidoglycan	Pseudomuréine - Pseudopeptidoglycane
RMSD	Root-Mean-Square deviation	Déviation quadratique moyenne
SDM	structural dissimilarity metric	métrique de dissimilarité structurale
SRMS	Similarity Measure	mesure de similarité
TalNAc	N-acetyltalosaminurinic	N-acétyltalosaminurinique
UDP-GlcNAc	uridine diphosphate N-acetylglucosamine	uridine diphosphate N-acétylglucosamine
UDP-MurNAc	UDP-N-acetylmuramic acid	acide UDP-N-acétylmuramique

3 Résumé

Les archées et les bactéries possèdent parmi leurs différences leur paroi. Cette différence suggère une évolution indépendante de la synthèse de la paroi dans les deux domaines (Albers & Meyer, 2011 ; Kandler & Konig, 1993 ; Subedi et al., 2021a). Cependant, chez certaines archées méthanogènes, une structure analogue au peptidoglycane (PG) est retrouvée, le pseudopeptidoglycane (pseudomuréine ou PM) (Subedi et al., 2021b ; Visweswaran et al., 2011). Dans V. Lupo et al. 2022, les auteurs ont identifié des Muramyl ligases (*Mur*α, *Mur*β, *Mur*γ et Murδ) homologues aux ligases bactériennes (MurCDEF), se trouvant en cluster dans les génomes des archées méthanogènes (les méthanopyrales et les méthanobactériales) possédant de la PM. Ils ont pu identifier à quelles étapes de la biosynthèse de la PM certains de ces gènes interviennent (Albers & Meyer, 2011 ; Lupo et al., 2022). Les quatre Mur ligases bactériennes qui permettent la formation du PG sont composées de trois domaines qui sont homologues dans les différentes familles de protéines (Smith et al., 1997). Chez les Mur ligases archéennes, Subedi et al. 2022 retrouvent une structure similaire, chez Murδ, avec trois domaines distincts.

Ces analyses font suite aux études menées par Subedi et al. (2021), Subedi et al. (2022) et Lupo et al. (2022). Dans ce travail, un protocole permettant de construire des arbres phylogénétiques sur base de leur distance structurelle (RMSD) a été créé. De plus, l'outil FoldTree a également été utilisé pour comparer les résultats obtenus avec d'autres métriques que le RMSD (lDDT, MT et Fident). Les résultats de cette étude montrent une variation importante des valeurs RMSD dans la comparaison entre les différents domaines (1, 2 et 3). Un travail de caractérisation des domaines individuels pour 45 séquences protéiques provenant de quatre familles de Mur ligases bactériennes ainsi que quatre familles de Mur ligases archéennes a été réalisé à l'aide de la phylogénie structurale (RMSD, FoldTree). Une nouvelle hypothèse sur les rôles respectifs des différentes des Mur ligases dans la voie de biosynthèse de la PM est obtenue. Murδ serait capable de reconnaître un glucide et de relier la chaîne d'AAs à l'acide N-acétyltalosaminuronique (NAT). Les trois domaines possèdent des valeurs RMSD très différentes en fonction de ceux-ci, montrant la complexité de la phylogénie des Mur ligases et offrant de nouvelles perspectives.

4 Introduction

4.1 L'arbre de la vie : Bactéries et Archées en confrontation

L'arbre de la vie est composé de trois domaines : les bactéries, les archées et les eucaryotes. Malheureusement, à l'heure actuelle l'enracinement de l'arbre de la vie n'est pas encore résolu (Fig.1)(Eme & Tamarit, 2024). L'hypothèse mise en avant dans la littérature est l'enracinement de l'arbre de la vie entre le domaine des bactéries et le domaine des archées (Dagan et al., 2010 ; Gogarten et al., 2010 ; Gribaldo & Cammarano, 1998). Les eucaroytes sont apparus suite à une fusion entre les bactéries et les archées. Cependant, aucune étude menée jusqu'à présent ne donne des preuves suffisantes pour appuyer cette hypothèse (Eme & Tamarit, 2024). Deux hypothèses alternatives existent également, la première suggère un enracinement au sein même du domaine des bactéries (Devos, 2021). La seconde parle d'un enracinement entre les eucaryotes et les procaryotes (Brinkmann & Philippe, 1999 ; Lopez et al., 1999). La principale cause de cette incertitude est l'apparition de signaux non verticaux dans certaines phylogénies de gènes. Ces signaux peuvent être la conséquence d'erreurs dans les méthodes utilisées qui donnent lieu à des artéfacts (Steenwyk et al., 2023). Cependant, ces signaux non verticaux pourraient également être dus à des événements de transferts horizontaux de gènes. Ces transferts sont très présents chez les bactéries et les archées (Zhaxybayeva & Doolittle, 2011) (Eme & Tamarit, 2024). Une autre source d'erreurs peut être liée au biais de position de LUCA (le dernier ancêtre commun) dans l'arbre de la vie. La position hypothétique de LUCA dans l'arbre de la vie est déterminante pour nos compréhensions actuelles. Si sa position était différente, nos déductions et les informations que nous avons sur lui et sur les embranchements suivants pourraient également être différentes (Eme & Tamarit, 2024). Découvrir et déterminer les relations possibles entre des gènes similaires répartis dans les différents groupes de l'arbre du vivant augmente les possibilités de résoudre les liens entre ces groupes. La difficulté est d'autant plus importante qu'un nombre important de transferts entre les domaines de l'arbre de la vie se sont produits. Ce nombre important de transferts est également vrai pour les gènes qui composent les parois cellulaires des procaryotes (Lupo et al., 2022).

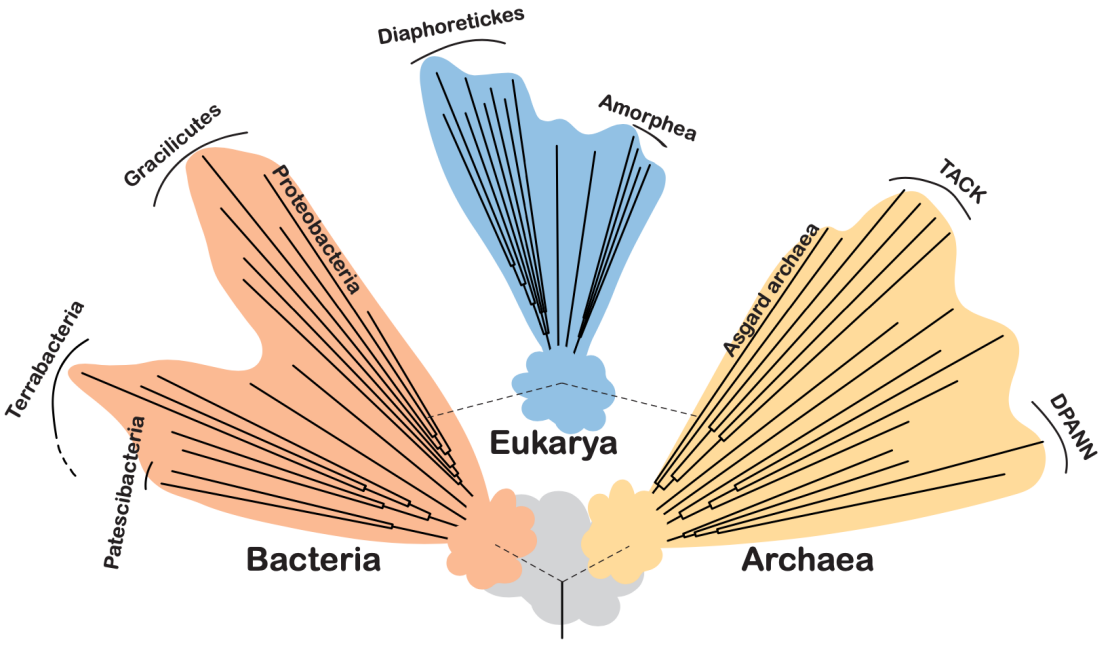

FIGURE 1 – Représentation schématique des incertitudes liées à certaines branches clés de l'arbre du vivant (Eme et Tamarit., 2024).

4.2 Peptidoglycane et pseudomuréine : Diversité dans les parois cellulaires

La paroi cellulaire entoure la majorité des cellules procaryotes. C'est une structure indispensable à la survie des organismes, elle procure une protection contre l'environnement et maintient la pression osmotique interne (Lupo et al., 2022 ; Meyer & Albers, 2020 ; Pazos & Peters, 2019). Le peptidoglycane (PG) est un polymère qui recouvre la membrane cytoplasmique chez la majorité des bactéries (Pazos & Peters, 2019). Le PG est composé d'acide N-acétylmuramique et de N-acétyl-D-glucosamine (GlcNAc) liés par des liaisons glycosidiques β-(1→4) (Visweswaran et al., 2011).

Les archées et les bactéries possèdent parmi leurs différences, leur paroi. Cette différence suggère une évolution indépendante de la synthèse de la paroi dans les deux branches (bactéries et archées) (Albers & Meyer, 2011 ; Kandler & Konig, 1993 ; Subedi et al., 2021). Cependant, chez certaines archées méthanogènes, une structure homologue au PG est retrouvée, le pseudopeptidoglycane (pseudomuréine ou PM) (Subedi et al., 2021 ; Visweswaran et al., 2011). La PM est constituée d'acide N-acétyltalosaminuronique (TalNAc) et de GlcNAc interconnectés par des liaisons glycosidiques β-(1→3) (Fig.2) (Visweswaran et al., 2011). À côté de la PM, on retrouve diverses autres parois chez les archées telles que les hétéropolysaccharides sulfatés, la méthanochondroïtine, le glutaminylglycane, les gaines protéiques, l'halamucine et les couches de surface glycoprotéiques (Albers & Meyer, 2011).

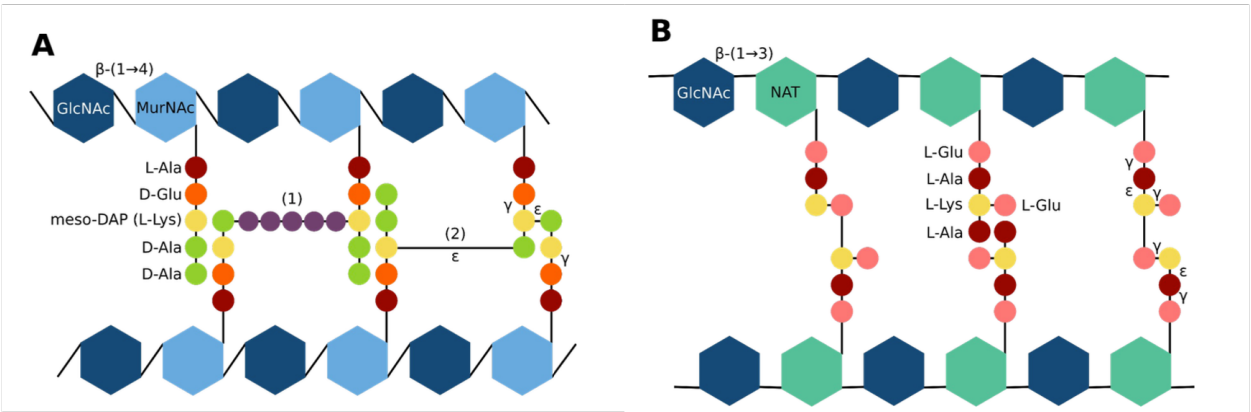

FIGURE 2 – Figure de comparaison de la structure du peptidoglycane et de la pseudomuréine. A. La chaîne de peptidoglycane composée des GlcNAc et des MurNAc, ainsi que ses acides aminés. B. La chaîne de pseudomuréine composée de GlcNAC et de NAT, ainsi que ses acides aminés (Lupo et al., 2022).

Bien que les chaînes de PG et de PM soient très similaires dans leur structure tridimensionnelle et leur fonction (Leps et al., 1984 ; Visweswaran et al., 2011), elles présentent des différences fondamentales dans leur biosynthèse et leur composition. Ces distinctions ont conduit Hartmann & Konig (1990) à émettre l'hypothèse (h0) selon laquelle le PG et la PM n'ont pas évolué à partir d'un ancêtre commun, mais plutôt que leurs similitudes sont le résultat de convergences évolutives (Hartmann & Konig, 1990).

L'intérêt particulier pour la paroi bactérienne découle de la lutte contre les bactéries pathogènes et la recherche de nouveaux antibiotiques (Bugg et al., 2011 ; Mandelstam & Rogers, 1959). La synthèse du PG se réalise grâce à l'intervention de plusieurs Muramyl ligases (Mur ligases). Ces enzymes permettent de catalyser la formation de liaisons peptidiques. Chez les bactéries, la biosynthèse du PG nécessite l'intervention de six Mur ligases (MurA/B, et MurC/D/E/F). MurA et MurB transforment UDP-GlcNAc en UDP-MurNAc. MurC, MurD MurE et MurF sont des peptides ligases qui vont ensuite ajouter les acides aminés (L-Ala, D-Glu, L-Lys et D-Ala-D-Ala) qui composent la chaîne peptidique. Le lipide I va être formé à l'aide de MraY qui transfère le phospho-N-acétylmuramoul-pentapeptide de l'UDP-MurNAc-pentapeptide ou transporteur lipidique undécaprényl-P (Und-P) (Kouidmi et al., 2014 ; Pazos & Peters, 2019). MurG va ensuite ajouter le GlcNAC de l'UDP-GlcNac pour former le lipide II qui sera ensuite transporté à l'extérieur de la membrane par une flippase (MurJ) (Sham et al., 2014). Le lipide II pourrait ensuite se lier au peptidoglycane en construction (Fig.3) (Pazos & Peters, 2019).

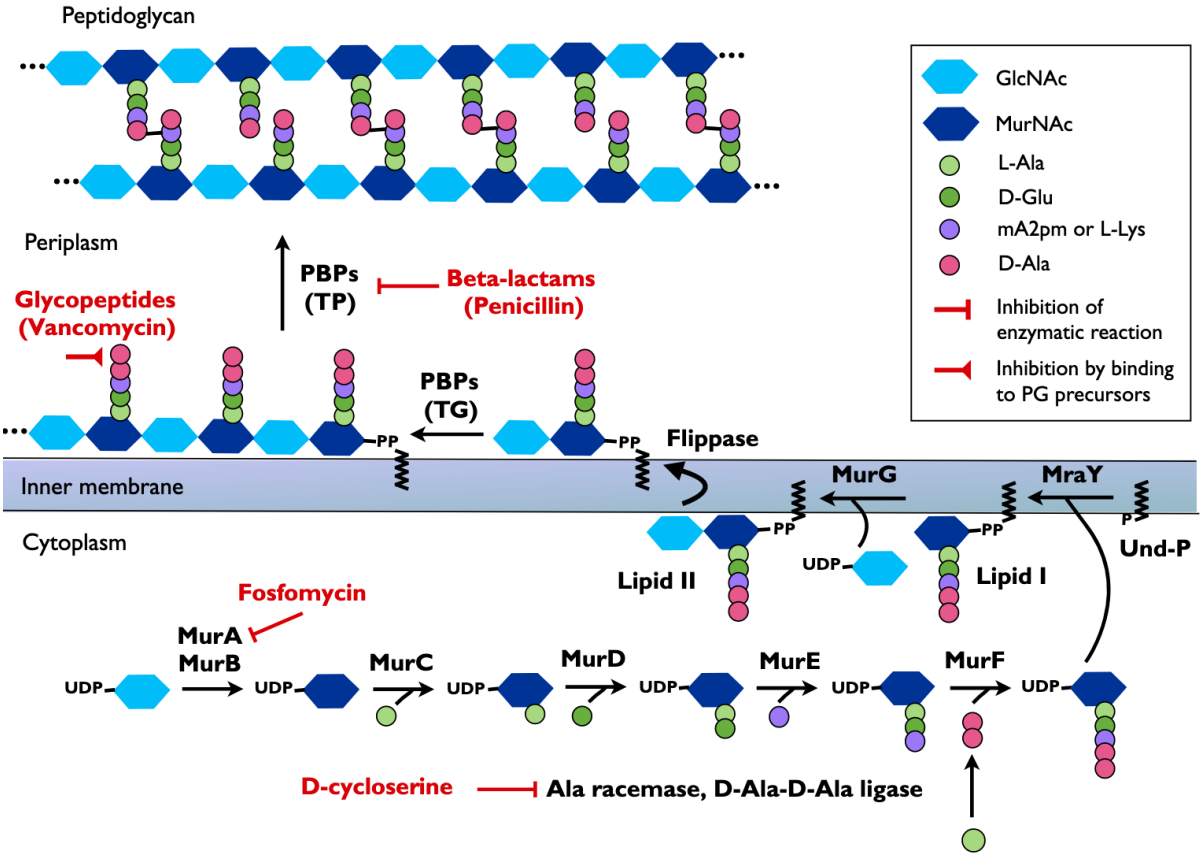

FIGURE 3 – Représentation schématique de la biosynthèse du peptidoglycane chez les bactéries. Le peptide GlcNAc subit une série de réactions à l'aide des différentes Mur ligases (A/B/C/D/E/F). MurA et MurB transforment UDP-GlcNAc en UDP-MurNAc. MurC, MurD, MurE et MurF ajoutent successivement les différents acides aminés. La molécule est ensuite liée à la membrane par MraY et forme le Lipide I. MurG ajoute un GlcNAc à la molécule afin de former le lipide II. Finalement, MurJ permet le passage du lipide II à travers la membrane et le lipide II se lie au peptidoglycane en construction (Kouidmi et al., 2014).

L'abréviation ***dcw*** vient de "Division and Cell Wall" (division et paroi cellulaire). Ce cluster contient un groupe de gènes impliqués dans la synthèse du peptidoglycane et dans la division cellulaire indispensable à la survie des bactéries (Ayala et al., 1994 ; Mingorance & Tamames, 2004). Les gènes impliqués dans la synthèse du PG se trouvent dans le cluster de gènes ***dcw***, qui est présent dans le génome de l'ancêtre commun des bactéries. Les gènes sont dans un ordre hautement conservé (Fig.4) (Léonard et al., 2022 ; Nikolaichik & Donachie, 2000). L'hypothèse la plus probable est la présence de ces gènes chez LBCA (le dernier ancêtre commun des bactéries). Une autre hypothèse suggère une acquisition légèrement plus tardive suivie d'un transfert latéral de gènes (Mingorance & Tamames, 2004). Le cluster ***dcw*** possède les gènes ***mur*** responsables de la formation du peptidoglycane.

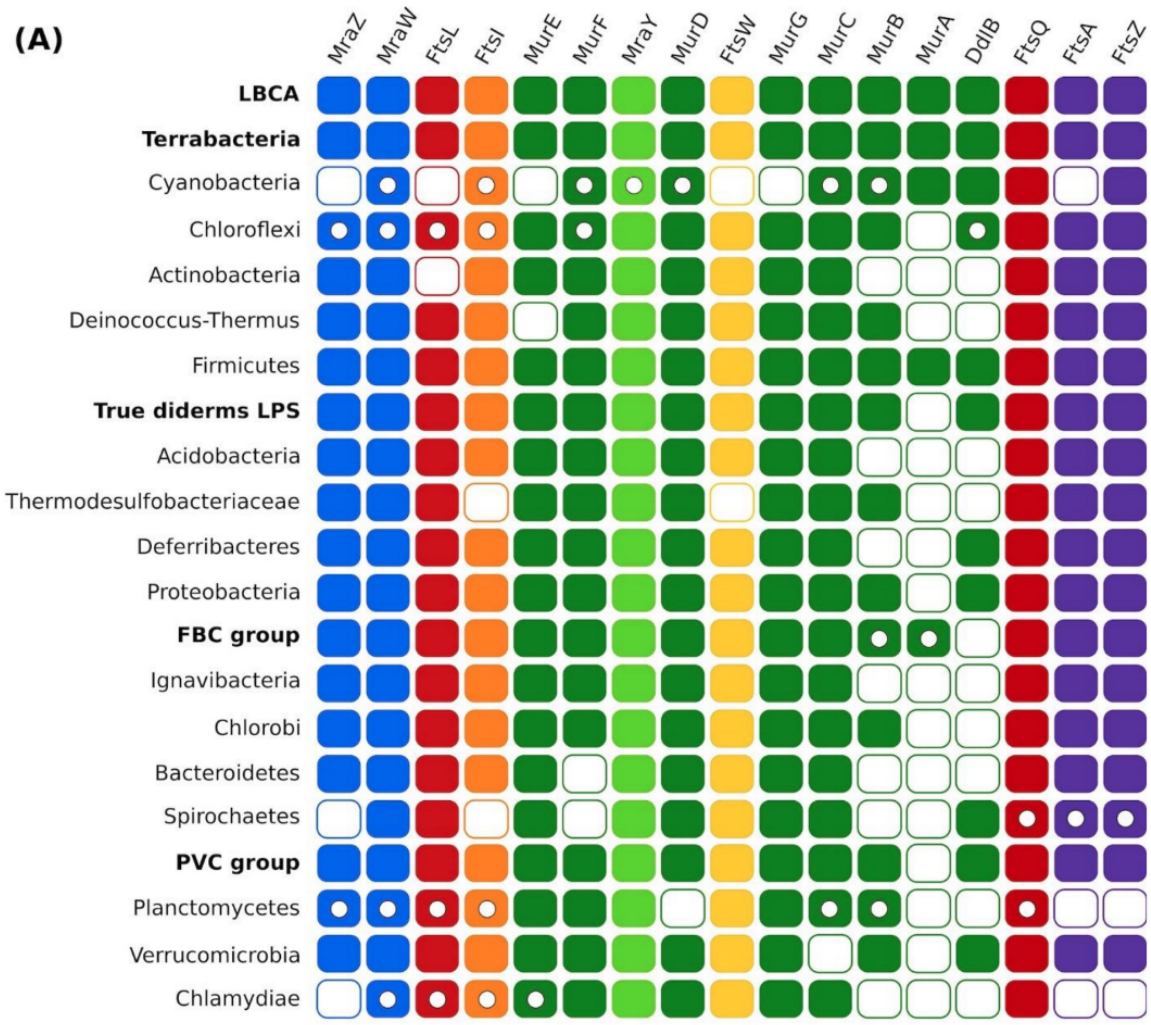

FIGURE 4 – Vue d'ensemble de la distribution des gènes et des analyses de synténie. (A) Organisation du cluster dcw dans certains ANC (derniers ancêtres communs) parmi les bactéries. Rectangle plein = gène présent dans le cluster principal ; cercle vide dans un rectangle = gène présent mais dans un sous-cluster ; rectangle vide = gène présent mais en dehors de tout cluster (Léonard et al., 20022).

Du côté des archées, la biosynthèse de la PM se résume en l'ajout de la chaîne d'acides aminés pour former le pentapeptide décrit par Hartmann & König (1994). Chaque acide aminé est ajouté au groupe N-amino de l'acide glutamique en trois étapes. Premièrement, le groupement phosphate de l'ATP est transféré sur le groupe amino de l'acide glutamique. Deuxièmement, UTP et le n-P-glutamique permettent de former le n-UDP-Glu (plus du pyrophosphate). L'acide aminé (L-Ala, L-Lys, L-Ala ou L-Glu) est ensuite ajouté (Hartmann & König, 1994).

La distribution phylogénétique des organismes contenant du PG et de la PM présente des différences marquées. En effet, le PG est largement présent chez les bactéries, tandis que la PM se retrouve uniquement chez certains membres spécifiques des archées méthanogènes, les méthanobactériales et méthanopyrus (Visweswaran et al., 2011). Malgré les différences fondamentales dans leurs voies de biosynthèse, l'évolution du PG et de la PM

semble liée par l'implication de gènes homologues aux gènes de la biosynthèse du PG (comme des Mur ligases) chez les méthanopyrales et les méthanobactériales (Leahy et al., 2010 ; Lupo et al., 2022 ; Smith et al., 1997).

Les précurseurs de la voie de biosynthèse de la PM sont connus. Cependant, la liste des gènes et leurs étapes précises d'implication dans la biosynthèse ne sont pas encore connues. Dans leur étude de 2022, V. Lupo et al. ont identifié des gènes codant pour des Mur ligases (Murα, Murβ,Murδ et Murγ) homologues aux Mur ligases bactériennes (MurCDEF), organisés en clusters dans les génomes des archées méthanogènes (les méthanopyrales et les méthanobactériales) possédant de la PM. Ils ont également déterminé les étapes spécifiques de la biosynthèse de la PM auxquelles certains de ces gènes participent (Albers & Meyer, 2011 ; Lupo et al., 2022).
L'hypothèse avancée sur la biosynthèse de la PM à l'aide de ces quatre Mur ligases est présentée dans la (Fig.5). Dans cette hypothèse, les Mur ligases archéennes joueraient un rôle similaire aux Mur ligases bactériennes et permettraient l'ajout des différents acides aminés.

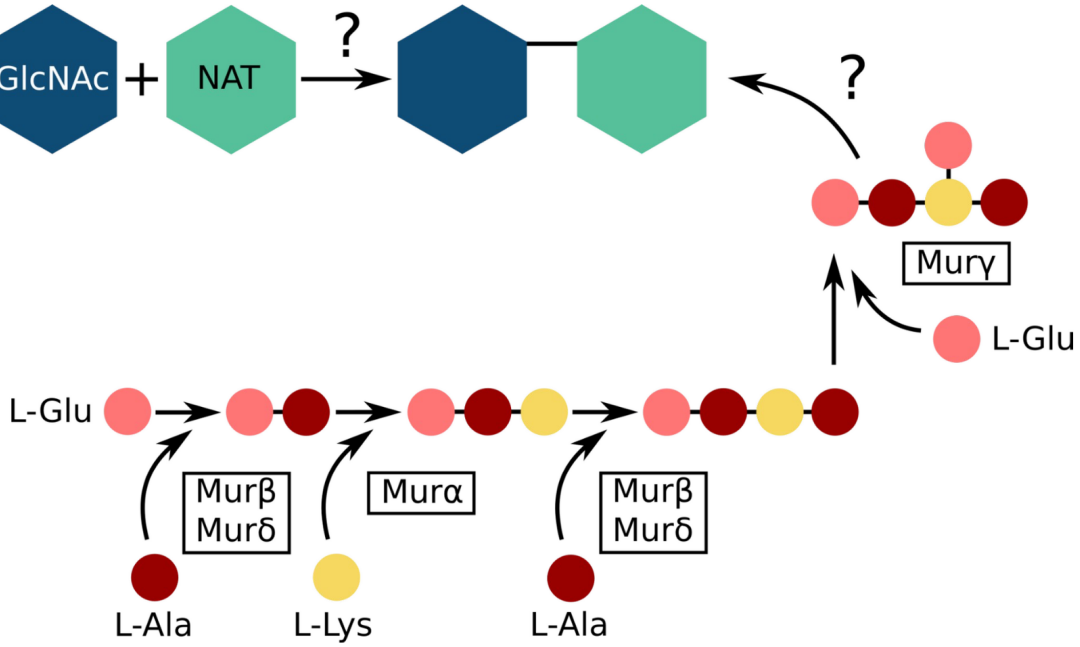

FIGURE 5 – Voie de synthèse hypothétique de la PM. Premièrement, Murβ et Murδ ajoutent la L-Alanine sur la L-Glutamine (cet ajout sera répétée ensuite). Deuxièmement, Murα ajoute la L-Lysine à la chaîne d'acides aminés. Troisièmement, après le deuxième ajout de la L-Alanine, Murγ va finalement ajouter la L-Glutamine à la chaîne d'acides aminés. Cette chaîne sera ensuite ajoutée au NAT dans une étape qui n'est pas encore déterminée (Lupo et al., 2022).

Il existe chez les archées méthanogènes deux clusters homologues au cluster ***dcw*** (Fig.6) (Hartmann & Konig, 1990 ; Hartmann & König, 1994 ; Lupo et al., 2022 ; Subedi et al., 2021). Ces clusters sont présents chez toutes les archées méthanogènes produisant de la PM et absents chez celles n'en produisant pas. Ces découvertes mettent en lumière une nouvelle hypothèse (h1), celle d'une évolution liée partant d'un ancêtre commun entre le PG et la PM (Subedi et al., 2021). Dans ces clusters on retrouve des gènes orthologues aux gènes des Mur ligases bactériennes, ***murC***, ***murD***, ***murE*** (Leahy et al., 2010 ; Subedi et al., 2021).

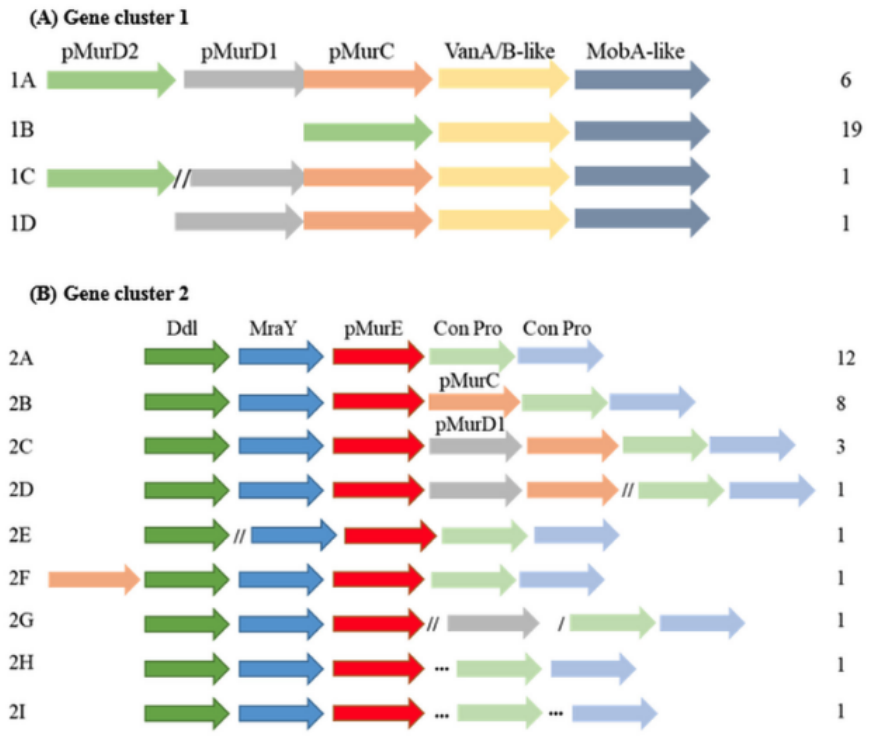

FIGURE 6 – Clusters Archées. (A) Cluster de gènes de la biosynthèse de la pseudomuréine 1. Les gènes sont identifiés par les flèches colorées ci-dessous, et le nombre d'occurrences dans un échantillon de 27 génomes archéens contenant de la pseudomuréine est indiqué à droite. Les types de clusters de gènes sont classés par lettres sur la marge de gauche. Les flèches grises/oranges chevauchantes indiquent que les gènes ont été annotés comme étant fusionnés/chevauchants ; /, un gène séparé ; //, deux gènes séparés. La pointe de flèche de chaque gène représente sa direction dans le génome. B) Cluster de gènes de la biosynthèse de la pseudomuréine 2. Les gènes sont montrés pour 29 génomes contenant de la pseudomuréine. Les types de clusters de gènes sont classés par lettres sur la marge de gauche et le nombre d'occurrences de ceux-ci est indiqué sur la marge droite (/, un gène séparé ; //, deux gènes séparés ; ..., non étroitement associés avec pMurE dans le génome) (Subedi et al., 2021).

Les quatre Mur ligases bactériennes qui permettent la formation du PG sont composées de trois domaines qui sont homologues dans les différentes familles de protéines (Smith, 2006), le domaine 1 (ou N-terminal), le domaine 2 (ou central) et le domaine 3 (ou C-terminal) (Kouidmi et al., 2014). Le domaine 1 permet de lier le substrat, c'est-à-dire l'UDP-MurNAc. Il est constitué d'un feuillet β et d'un nombre variable d'hélices α . Le domaine 2 permet de lier l'ATP et comporte une boucle très conservée à cet effet. Il est composé d'un feuillet β (chez MurC) et d'hélices α. Le domaine 3 comporte le Rossmann dinucleotide-binding fold qui permet de lier l'acide aminé. Il est composé d'un feuillet β et d'hélices α (Kouidmi et al., 2014 ; Smith, 2006).

Chez les Mur ligases archéennes, Subedi et al. (2022) retrouvent une structure similaire chez Murδ. Les trois domaines sont constituée chacun d'un feuillet β et d'un nombre d'hélices α variable en fonction de la famille. La présence d'un site de liaison de l'ATP a été observé ainsi que deux sites de liaison à l'UDP (Subedi et al., 2022).

Les travaux récents de Lupo et al. (2022) ont été réalisés sur base de phylogénies dites “classiques” qui utilisent les séquences d’AAs en Maximum de Vraisemblance. Ces méthodes ont été peu concluantes pour déterminer quelle hypothèse (h0 ou h1) est la plus probable (Fig.7). Une autre technique de phylogénie moins classique existe dans la littérature, la phylogénie structurale basée sur les structures tridimensionnelles des protéines. Cette méthode offre une perspective novatrice pour élucider les complexités de cette phylogénie.

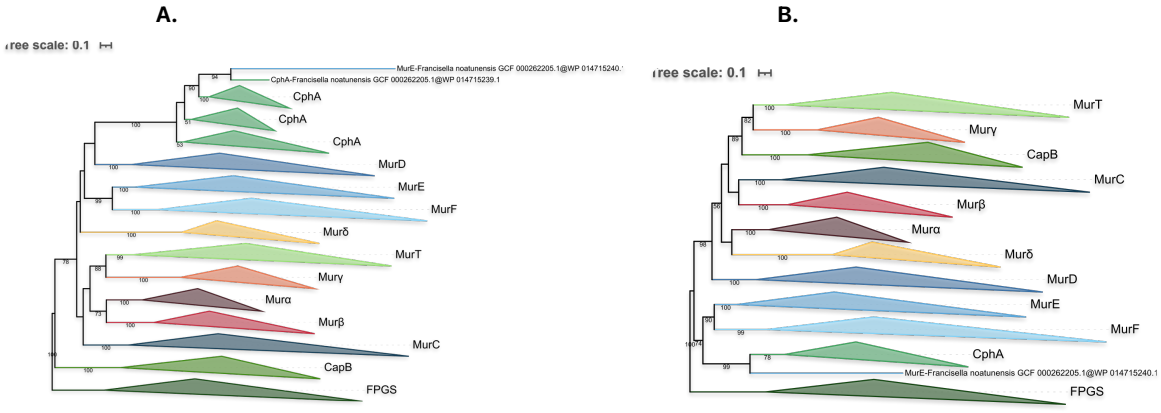

FIGURE 7 – A. Arbre phylogénétique des familles contenant le domaine Mur enraciné sur FPGS. Arbre inféré sur base d’une matrice de 3407 séquences utilisant IG-TREE sous le modèle C40+G4. B. Arbre phylogénétique des familles contenant le domaine Mur enraciné sur FPGS. Arbre inféré sur base d’une matrice de 3407 séquences utilisant IG-TREE sous le modèle C20+G4 (Lupo et al. 2022).

4.3 Phylogénie structurale

Les séquences d’acides aminés sont majoritairement utilisées pour étudier l’histoire évolutive des protéines. Cependant, une série d’études ont pu montrer que la structure tridimensionnelle favorise la compréhension des relations évolutives (Agarwal et al., 2009 ; Balaji & Srinivasan, 2007 ; Lakshmi et al., 2015). En effet, les structures tridimensionnelles des protéines et leur fonction sont mieux conservées que les séquences au cours de l’évolution (Johnson et al., 1990 ; Lakshmi et al., 2015) ce qui permet de résoudre des phylogénies plus complexes. La structure est trois à dix fois plus conservée que les séquences d’AAs (Illergård et al., 2009). En effet, Chothia et Lesk (1986) ont été les premiers à montrer la relation entre la distance entre les séquences et la divergence structurelle entre protéines homologues (Chothia & Lesk, 1986). De plus, il a été démontré que certaines protéines homologues conservent des structures tridimensionnelles fortement similaires, alors que leur identité de séquence est faible (Naveenkumar et al., 2022). La phylogénie structurale permet d’aider à résoudre des questions de relations évolutives lorsque la phylogénie basée sur les séquences ne donne pas de résultats clairs. Cependant, différentes méthodes sont utilisées dans la littérature afin de réaliser cette phylogénie structurale. La majorité des méthodes utilisées parte d’une matrice de distances entre les protéines, distance qui peut être représentée par différentes métriques (RMSD, lDDT, TMalign) (Agarwal et al., 2009 ; Johnson et al., 1990 ; Lakshmi et al., 2015 ; Mariani et al., 2013 ; Naveenkumar et al., 2022 ; Sala et al., 2023 ; van Kempen et al., 2024 ; Zhang & Skolnick, 2005). La majorité des travaux dans la littérature se basent sur la méthode de RMSD pour construire leurs arbres phylogénétiques structuraux.

$$RMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{r}_i^{(1)} - \mathbf{r}_i^{(2)}\right\|^2}$$

Avec ri(1)-ri(2) qui est la distance euclidienne entre les positions des atomes dans les deux structures (Riniker et al., 2012).

Cette distance de RMSD est ajustée afin de calculer le SDM, en tenant compte du nombre de carbones α constituant les protéines (ou les domaines protéiques).
La formule utilisée est la suivante :

$$SDM = -100 \times (w_1 \times PFTE + w_2 \times SRMS)$$

Avec,

$$PFTE = \frac{\text{nbr d'atomes équivalent carbone alpha}}{\text{nbr de résidus de la plus petite protéine}}$$

$$SRMS = 1 - RMSDdans\text{Å}/max\text{Å}$$

[1]

$$w_1 = \frac{(1 - PFTE) + (1 - SRMS)}{2}$$

[2]

$$w_2 = \frac{(PFTE + SRMS)}{2}$$

Et,

$$w_1 + w_2 = 1$$

(Johnson et al., 1990)

Dans ses travaux, Illergård et al. (2009) a montré que la structure des protéines (mesurée par RMSD ou d'autres méthodes similaires) change approximativement de manière linéaire avec la distance évolutive (Illergård et al., 2009). Cependant, David Moi et son équipe (2024) a conçu une nouvelle approche pour répondre à la question de la faisabilité de la phylogénie structurale, FoldTree (https ://github.com/DessimozLab/foldTree). FoldTree permet rapidement et facilement d'obtenir des arbres basés sur la structure tridimensionnelle des protéines. Trois métriques sont sélectionnables dans leur approche, lDDT, TMalign et Fident (méthode basée sur Foldseek). Dans ce travail, FoldTree est utilisé afin d'obtenir des arbres avec d'autres métriques que le RMSD.

5 Objectifs

Ce travail comporte deux objectifs principaux. Le premier est de mettre en place un protocole de phylogénie structurale (basée sur les structures tridimensionnelles des protéines) pour essayer de résoudre des phylogénies

1. le maxÅ est la valeur de distance maximum obtenue pour la comparaison entre protéines (ou domaines de protéines).
2. Les poids sont utilisés afin de modérer l'influence du PFTE en présence de petites distances pour lesquelles le SRMS donne une meilleure représentation de la relation entre les protéines (inversement).

complexes, difficiles à résoudre avec la phylogénie dite “classique”. Pour ce faire, une méthode “manuelle” sera développée afin de visualiser les différentes étapes nécessaires à la construction de tels arbres. Pour cette méthode, la métrique RMSD est utilisée car celle-ci est fortement documentée dans la littérature. Ensuite, une méthode développée par D. Moi et son équipe sera testée, FoldTree. Le deuxième objectif est d'étudier les relations phylogénétiques des mur ligases bactériennes et archéennes à l'aide de ces arbres phylogénétiques structuraux. Ces relations ont été au préalable étudiées par V. Lupo lors de sa thèse de doctorat, mais elles n'ont pas été résolues. Ce travail permet l'addition de la phylogénie sur base des structures tridimensionnelles à la phylogénie classique utilisée par V. Lupo.

6 Matériels et Méthodes

6.1 Environnement

6.1.1 Durandal

`Durandal` est un cluster de calculs géré par l'Unité de Phylogénomique des Eucaryotes et appartenant à l'unité de recherche InBioS-PhytoSYSTEMS de l'Université de Liège. Il est composé de 228 cœurs physiques contenus dans 12 nœuds. Il est composé de 2880 cœurs CUDA (GPU), 2.9 TB de RAM et 162 TB d'espace de stockage. Le système d'exploitation de durandal est le système Linux CentOS 6.6.

6.1.2 AIDA

`AIDA` est un cluster de calculs géré l'Unité de Phylogénomique des Eucaryotes et appartenant à InBioS-PhytoSYSTEMS de l'Université de Liège. Il est composé de 20 coeurs physiques. Il est composé de 6912 coeurs CUDA, 263 GB de RAM et 7 TB d'espace de stockage. Le système d'exploitation est Ubuntu Linux 22.04.

6.1.3 Ordinateur personnel

MacBook Air Retina contenant quatre coeurs, 1,1 Gz Intel Core i5 comme processeur et de 8 Go de mémoire (3733MHz LPDDR4X). La version du stystème est la Ventura 13.6.3.

6.2 Programmes

n° d'utilisation	Nom du programme	Version	Développeurs	Références
1	AlphaFold	V2.0 2022	DeepMind Technologies Limited	(Jumper et al., 2021 ; Varadi et al., 2024)

n° d'utilisation	Nom du programme	Version	Développeurs	Références
16	Consurf	2019	Barak Yariv, Elon Yariv, Amit Kessel, Gal Masrati, Adi Ben Chorin, Eric Martz, Itay Mayrose, Tal Pupko, and Nir Ben-Tal	(Ashkenazy et al., 2016 ; Yariv et al., 2023)
17	MAFFT	V7.453 Nov 2019	Kazutaka Katoh	(Rozewicki et al., 2019)
18	iTOL	V6.9 Avr 2024	Letunic I et Bork P	(*Interactive Tree of Life (iTOL) V6*, n.d.)
5	Pymol	2.5.4 Juil 2022	Schrödinger	
/	Rstudio	2023.12.1+402		
2	hmmsearch	HMMER 3.3 Nov 2019	SR Eddy	
5	Pymol	2.5.4 Juil 2022	Schrödinger	
13	parse_consense_out.pl	?	D. Baurain	
14	get-stat.pl		V. Lupo	
3	hmms-parser-coord.pl		cfr annexes	
4	pymol-script.pl		cfr annexes	
6	pymol-script.pml		cfr annexes	
7	SmallDom.sh		cfr résultats	
8	analyses_RMSD.R		cfr annexes	
9	analyses_SDM.R		cfr annexes	
10	jack_dist.R		cfr annexes	
11	SnakeM_myfam.pl		cfr annexes	
12	foldtree.sh		cfr annexes	

6.3 Schéma récapitulatif des manipulations

6.4 Protocole manuel (Distance RMSD)

6.4.1 Sélection et prédiction des structures tridimensionnelles

Une liste de protéines à étudier provenant de différents organismes a été compilée par V. Lupo :

N°	Identifiant GenBank	Organisme	Mur ligases
1	WP_011244563.1	Synechococcus elongatus	MurF
2	WP_034526537.1	Secundilactobacillus oryzae	MurF
3	WP_014731099.1	Mesotoga prima	MurF
4	NP_213563.1	Aquifex aeolicus VF5	MurF
5	WP_011361105.1	Chlorobium chlorochromatii	MurF
6	WP_039704550.1	Helicobacter pylori	MurF
7	WP_011378051.1	Synechococcus elongatus	MurE
8	WP_034525752.1	Secundilactobacillus oryzae	MurE
9	WP_014731098.1	Mesotoga	MurE
10	NP_214197.1	Aquifex aeolicus	MurE
11	WP_011361104.1	Chlorobium chlorochromatii	MurE
12	WP_039704973.1	Helicobacter pylori	MurE
13	WP_011954343.1	Methanobrevibacter	Murα
14	WP_084789739.1	Methanobacterium congolense	Murα
15	WP_088335895.1	Methanopyrus sp.	Murα
16	WP_112094175.1	Methanothermobacter tenebrarum	Murα
17	WP_013413417.1	Methanothermus fervidus	Murα
18	WP_011954344.1	Methanobrevibacter	Murβ
19	WP_071906152.1	Methanobacterium congolense	Murβ
20	WP_088336155.1	Methanopyrus sp.	Murβ
21	WP_112094176.1	Methanothermobacter tenebrarum	Murβ
22	WP_013413418.1	Methanothermus fervidus	Murβ
23	WP_011953694.1	Methanobrevibacter	Murγ
24	WP_071906075.1	Methanobacterium congolense	Murγ
25	WP_013413421.1	Methanothermus fervidus	Murγ
26	WP_112094177.1	Methanothermobacter tenebrarum	Murγ
27	WP_088336109.1	Methanopyrus sp.	Murγ
28	WP_011953843.1	Methanobrevibacter smithii	Murδ
29	WP_071906154.1	Methanobacterium congolense	Murδ
30	WP_013413839.1	Methanothermus fervidus	Murδ
31	WP_112093959.1	Methanothermobacter tenebrarum	Murδ
32	WP_088335892.1	Methanopyrus sp.	Murδ
33	WP_011244660.1	Synechococcus elongatus	MurC
34	WP_034525710.1	Secundilactobacillus oryzae	MurC
35	WP_014731103.1	Mesotoga	MurC
36	NP_213937.1	Aquifex aeolicus	MurC

N°	Identifiant GenBank	Organisme	Mur ligases
37	WP_039704479.1	Helicobacter pylori	MurC
38	WP_041466419.1	Chlorobium chlorochromatii	MurC
39	WP_011361110.1	Chlorobium chlorochromatii	MurC
40	WP_034526164.1	Secundilactobacillus oryzae	MurD
41	WP_041928127.1	Mesotoga prima	MurD
42	NP_214421.1	Aquifex aeolicus	MurD
43	WP_011361107.1	Chlorobium chlorochromatii	MurD
44	WP_011244732.1	Synechococcus elongatus	MurD
45	WP_039704394.1	Helicobacter pylori	MurD

Cette liste est stockée dans un fichier .list sur `durandal`. Les identifiants spécifiques ont été isolés avec la commande suivante :

```
$ head sequence_id.list | cut -d '@' -f 2 > ID_prot.txt
```

Ensuite, l'ensemble des prédictions protéiques des génomes complets sont téléchargés sur le cluster de calcul `durandal` :

```
$ for ID in $(more ID_prot.txt);
$ do efetch -db protein -id $ID -format fasta;
$ done > SeqMurT.fasta
```

Afin de pouvoir réaliser les arbres phylogénétiques structuraux, les fichiers PDB contenant les structures tridimensionnelles des protéines doivent être téléchargés. Pour ce faire, les prédictions des structures tridimensionnelles des différentes séquences protéiques sélectionnées ont été recherchées sur les bases de données **AlphaFold** (https ://alphafold.ebi.ac.uk/) et `Uniprot` (https ://www.uniprot.org/) à l'aide des identifiants des protéines. Les protéines qui n'étaient pas présentes dans les bases de données ont été prédites sur `AIDA` qui contient le programme `AlphaFold` (programme1) qui permet de faire des prédictions des structures tridimensionnelles des protéines via la ligne de commande. Ce programme utilise `Python`. Il demande en entrée un fichier contenant la séquence pour laquelle la structure tridimensionnelle doit être prédite et produit des fichiers PDB (`relaxed_model_*_pred_0.pdb`) qui contiennent ces structures.

```
$ python3 /home/cmullender/alphafold/docker/run_docker.py \
  --gpu_devices=0 --fasta_paths=/home/cmullender/.../ \
  sequence.seq --max_template_date=2023-01-01 \
  --data_dir=/data/af \
  --output_dir=/home/cmullender/.../ \
  --model_preset=monomere \
  --num_multimer_predictions_per_model=5 \
  --use_precomputed_msas=false & \
```

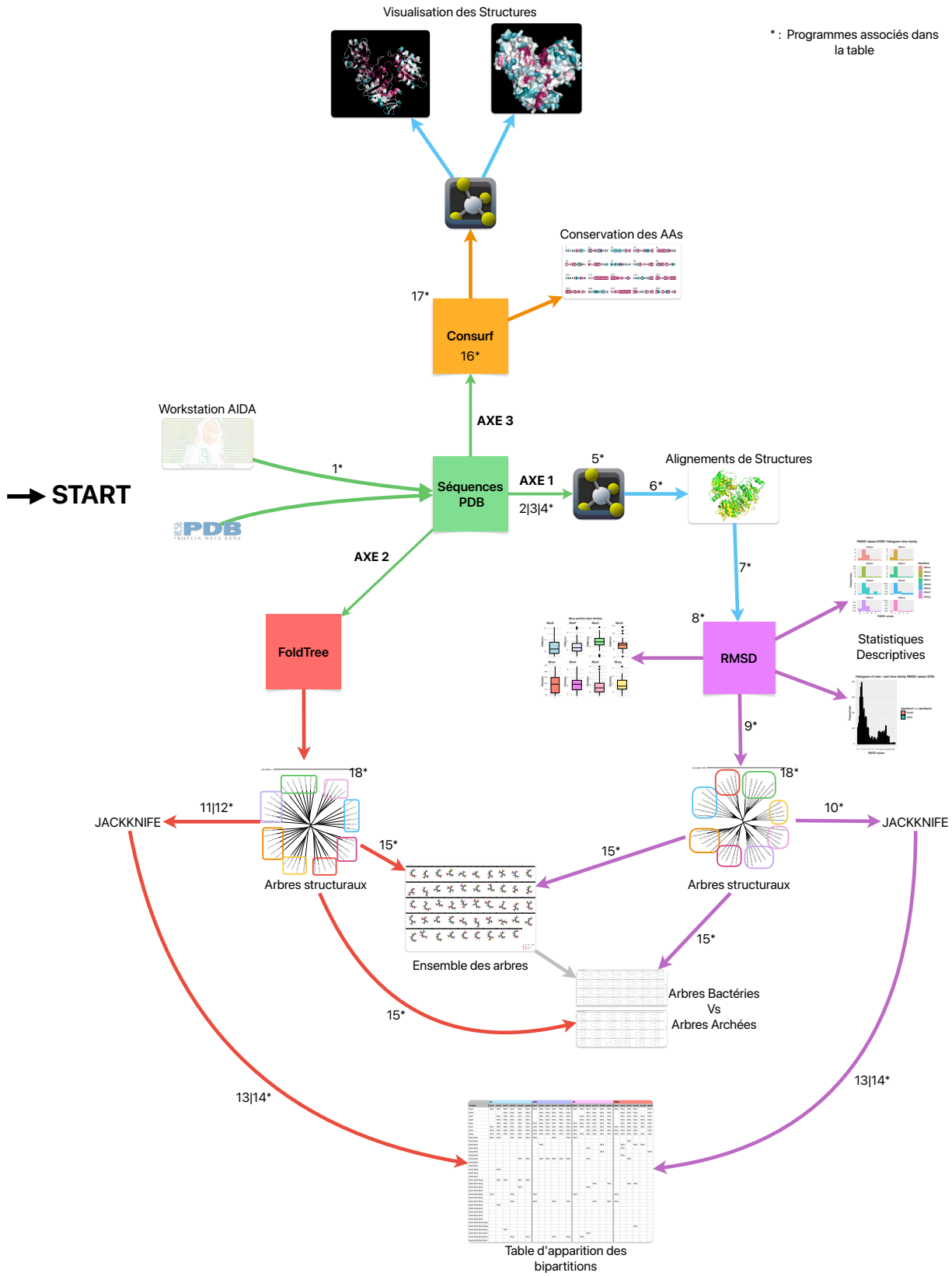

FIGURE 8 – Manipulations

6.4.2 Extraction des coordonnées des domaines

Cette étude étant basée sur la prédiction d'arbres structuraux à partir des différents domaines des protéines Mur ligases, l'ensemble des coordonnées de ces domaines doit être extrait. Pour ce faire, les positions des acides aminés qui délimitent le domaine central sont obtenues. La position exacte du domaine central permettra ensuite de déterminer la position du domaine N-terminal et C-terminal qui se trouvent de part et d'autre du domaine central. Afin d'obtenir les coordonnées du domaine central, le programme `hmmsearch` (programme2) est utilisé. Ce programme demande un fichier .hmm et un fichier fasta qui contient les séquences. Il délivre un fichier .hmms (`mur_central_domain_structural_subset.hmms`) contenant les coordonnées du domaine central.

```
$ hmmsearch --domtblout mur_central_domain_structural_subset.hmms \
  data/mur_central_domain.hmm Seq_modif-names.fasta
```

Après avoir obtenu la position du domaine central, le script `hmms-parser-coord.pl` (programme3) génère comme fichier de sortie les coordonnées des trois domaines de la protéine. Ce programme demande en entrée le profil pHMM (profile Hidden Markov Model) qui est le résultat de la recherche pHMM contenant les coordonnées du domaine central (.hmms) et la liste des séquences en format fasta. Il exporte les coordonnées des trois domaines dans un fichier texte. Ce fichier est ensuite modifié afin d'obtenir les coordonnées des trois domaines dans trois fichiers distincts.

```
$ ./hmms-parser-coord.pl| EtudeStructure/Seq_modif-names.fasta \
  EtudeStructure/mur_central_domain_structural_subset.hmms
```

6.4.3 Prédiction des distances RMSD sur Pymol

Maintenant que les fichiers PDB et les coordonnées des différents domaines sont prêts, la distance entre les différentes protéines comparées deux à deux peut être obtenue. Cette distance est calculée par le programme `Pymol` (programme5). Afin d'augmenter la rapidité des analyses, des scripts lançant automatiquement les commandes `Pymol` ont été créés. Ces scripts sont eux-mêmes écrits par le programme `pymol-script.pl` (programme4). Ce programme demande deux fichiers textes, l'un contenant la liste des noms des protéines présentes dans les fichiers PDB et le fichier de sortie contenant les coordonnées du domaine étudié obtenu précédemment. Le programme `pymol-script.pl` (programme4) exporte en sortie un fichier .pml qui contient un script python destiné à `Pymol`.

Ce script a ensuite été modifié à plusieurs reprises (cfr Programmes) afin de :

— sortir les informations de la protéine complète (non séparée en domaines).

— sortir les informations des domaines regroupés (combinaison des domaines 1 et 2, et combinaison des domaines 2 et 3).

— sortir les informations relatives au nombre de carbones α qui composent les différentes protéines (nécessaire à la formule pour l'obtention du SDM).

Le fichier `pymol-script.pml` (programme6) est fourni à `Pymol` (programme5) afin d'aligner structuralement

les domaines des protéines deux à deux. Cet alignement permet de calculer la distance de RMSD qui représente la somme des distances inter-atomes entre les différentes paires de protéines comparées. La sortie de cette commande est un fichier texte qui renferme l'ensemble des sorties `Pymol`.

```
$ pymol -c pymol-script.pml > pymol-output.txt
```

Le fichier `pymol-output.txt` est ensuite modifié afin de ne garder que les informations utiles à la suite de l'analyse : la distance RMSD, le nombre de carbones α, la taille de la plus petite protéine alignée pour chaque aligement structural.

6.4.4 Obtention des arbres sur R studio

À ce stade, deux étapes essentielles sont réalisées. La première est de récupérer les valeurs RMSD et de les associer aux protéines correspondantes dans une première matrice. La deuxième est de modifier les valeurs RMSD en valeurs SDM et de former une deuxième matrice qui combine les valeurs SDM et les protéines associées.

6.4.4.1 Générer les données en bash Afin de faire fonctionner ce code, différents fichiers ont dû être créés sur base de la sortie du code donné précédemment à `Pymol`. Ce code génère un fichier contenant l'intégralité des sorties de `Pymol` (programme5), il est donc nécessaire de traiter cette sortie afin d'en extraire les informations nécessaires suivantes :

— RMSD

— nbrCA (nombre de carbones α)

— nbrSmall (la taille de la plus petite protéine alignée)

— liste_prot (la liste complète des protéines utilisées)

Différentes lignes de commandes ont été utilisées :

RMSD

```
$ grep "Executive: RMSD" testcoord1.txt | \
  tr -s " " ";" | cut -d";" -f5  > RMSD.txt
```

NbrCA

```
#Permet d'avoir le nombre de carbones alpha et donc de résidus alignés
$ grep "Executive: RMSD" DOM*CA_SP.txt | tr -s " " ";" | \
  cut -d";" -f6 | sed 's/[()]//g' > DOM*CA_Aligned.txt
```

NbrSmall

```
#Permet d'avoir la taille deux à deux des protéines alignées
$ grep "MatchAlign: aligning" DOM*CA_SP.txt | cut -d"(" -f2 | \
  cut -d" " -f1 -f3 | sed 's/[)]//g' | sed 's/[...]//g' | \
  tr " " ";" > DOM13CA_Small.txt
```

SmallDom.Sh

```
#Permet d'avoir la taille de la plus petite protéine alignée
#!/bin/bash
#ouvrir le fichier
fichier=$1

#trouver la plus petite des deux
while IFS=';' read -r num1 num2 || [ -n "$num1" ]; do
  if [ "$num1" -lt "$num2" ]; then
     echo "$num1"
  else
     echo "$num2"
  fi
done < "$fichier"
```

```
$ ./SmallDom.sh Nbr_residu1.txt > Small_residu1.txt
```

6.4.4.2 Statistiques descriptives La matrice contenant les valeurs RMSD des Mur ligases est nécessaire afin de réaliser les statistiques descriptives des données. Elle est obtenue grâce à Rstudio. Le script `analyses_RMSD.r` (programme8) combine la création de cette matrice RMSD et l'obtention de différentes représentations de données sous forme d'histogrammes de distributions et de boxplots. Ce script demande en entrée uniquement les valeurs RMSD et les noms des protéines dans l'ordre d'obtention des valeurs RMSD sur `Pymol`. Il utilise les packages `ggplot2` (Wickham, 2016), `gridExtra` (Auguie, 2010) et `reshape2` (Wickham, 2007).

6.4.4.3 SDM et arbres phylogénétiques La matrice contenant les valeurs SDM est produite ainsi que les arbres NJ à l'aide du script `analyses_SDM.r` (programme9) sur `Rstudio`. Ce script demande plusieurs informations : les valeurs RMSD, le nombre de carbones α, la taille de la plus petite protéine alignée, ainsi que les noms des protéines dans l'ordre d'obtention des valeurs RMSD sur `Pymol`. Il produit des fichiers textes contenant l'arbre phylogénétique structurale attendu au format Newick. Ce format est le format standard d'écriture des arbres phylogénétiques. Les arbres sont visualiser sur `iTOL` (programme18) (https ://itol.embl.de/).

Cette méthodologie a été appliquée pour l'ensemble des domaines ou groupes de domaines analysés :

— La combinaison des 3 domaines
— La combinaison du domaine 1 avec le domaine 2

— La combinaison du domaine 2 avec le domaine 3 [3]
— Le domaine 1
— Le domaine 2
— Le domaine 3

6.4.5 Jackknife

Afin de vérifier la robustesse des arbres phylogénétiques obtenus pour chaque domaine ou combinaisons de domaines, les organismes utilisés ont été rééchantillonnés au hasard afin de générer 100 arbres composés d'identifiants de séquences différents. Pour ce faire, un script `jack_dist.r` (programme10) est créé. Ce script prend la matrice de SDM créée précédemment. Il enlève aléatoirement un identifiant de séquence pour chaque famille de la matrice et crée un arbre avec la nouvelle matrice qui contient huit identifiants en moins. Ce script réalise ces étapes 100 fois et stocke l'ensemble des arbres obtenus en format Newick dans un seul fichier texte.

Le programme `parse_consense_out.pl` (programme13) est utilisé afin de déterminer la présence ou l'absence de bipartitions [4] dans un arbre phylogénétique (newick). Le programme a besoin de l'arbre que l'on souhaite analyser au format newick et d'un fichier contenant les OTU qui composent l'arbre fourni. C'est-à-dire, le nom de la famille et l'ensemble des identifiants des séquences qui composent la famille. Enfin, l'ensemble des résultats de `parse_consense_out.pl` (programme13) est donné à `get-stat.pl` (programme14) qui va calculer le pourcentage d'apparition de chaque clan présent dans nos 100 arbres.

6.5 FoldTree (lDDT, MT et Fident)

Moi et al. (2023) ont conçu une nouvelle approche pour répondre à la question de la phylogénie basée sur les structures tridimensionnelles des protéines : `FoldTree` disponible sur github (https ://github.com/DessimozLab/fold_tree). L'utilisation de ce programme nécessite mamba afin d'utiliser `Snakemake`. Le protocole pour télécharger Snakemake se trouve à l'addresse suivante : https ://snakemake.readthedocs.io/en/stable/getting_started/installation.html. Une fois `Snakemake` installé, et `FoldTree` installés, il ne reste plus qu'à créer un dossier `myfam` contenant un dossier structs et un fichier `identifiers.txt`. Le fichier `identifiers.txt` doit contenir les identifiants UniProt des protéines utilisées. Le programme va rechercher les fichiers PDB associés sur **AlphaFold** (protein database). Si les fichiers PDB sont déjà présents sur la machine, il suffit alors de les placer dans le dossier structs et de laisser le fichier `identifiers.txt` vide.

La commande pour lancer FoldTree est la suivante :

```
snakemake --cores 4 --use-conda -s ./workflow/fold_tree \
  --config folder=./myfam filter=False custom_structs=True
```

3. La combinaison du domaine 1 et 3 a été exclue en raison de la coupure présente par le domaine 2 entre les deux. Cette coupure pose des questions sur l'agencement et les mouvements des domaines 1 et 3 autour du domaine 2.
4. Division d'un ensemble de séquences ou de taxons en deux sous-ensembles distincts qui représente une séparation des taxons en deux groupes au niveau d'une branche spécifique de l'arbre.

Fold_tree
FoldTree est un programme conçu par Moi et al. (2023) dans le but de proposer une méthode simple et efficace pour créer des arbres phylogénétiques sur base de distances entre les structures tridimensionnelles des protéines. FoldTree propose trois métriques différentes afin de fournir trois arbres phylogénétiques différents. Ce programme aligne et compare l'ensemble des structures, protéine contre protéine, en utilisant l'alphabet structural de Foldseek en neighbour joining (Moi et al., 2023). Les trois métriques sont les suivantes :
-LDDT (local distance difference test) : Calcule des distances locales entre toutes les paires d'atomes de la structure d'intérêt et de la structure de référence, en utilisant un rayon d'inclusion qui définit jusqu'à quelle distance les différences sont prises en compte. La mesure de lDDT vérifie si la structure prédite est fidèle au modèle de référence (Mariani et al., 2013).
-TM score (alignement de structure rigide) : Évalue la similarité globale entre deux structures protéiques. Cette mesure est moins affectée que la lDDT par les différences locales et se concentre sur un alignement local. Elle ne prend pas en compte la taille des protéines (Zhang & Skolnick, 2004).
-Fident (distance dérivée à partir des similarités par rapport à un alphabet structurel) : Mesure utilisant Foldseek qui utilise une alphabet structurel pour représenter les éléments de structure secondaire. Cette mesure permet de mieux capturer les similarités complexes entre les structures qui sont difficilement capturées par les méthodes basées sur les coordonnées atomiques (Moi et al., 2023).
L'étude menée par David Moi montre que la métrique Fident donne les meilleurs résultats en terme de congruence avec les arbres phylogénétiques basés sur les séquences d'AAs.
(Moi et al., 2023)

6.5.1 Phylogénie structurale des archées et des bactéries

Des phylogénies comportant uniquement les familles bactériennes ou les familles archéennes ont été construites.L'objectif est de déterminer l'importance relative de chaque domaine influençant la topologie dans les phylogénies. Cette analyse sera menée sur la combinaison des trois domaines, ainsi que sur les domaines indépendants et les combinaisons de domaines. Les arbres obtenus par la méthode privilégiée par FoldTree ont été gardés et affichés sur `Itol` (programme18) afin de trouver l'enracinement le plus probable et d'éventuels clans composés de familles différentes. Deux approches d'enracinnement ont été considérées : un enracinement séparant l'arbre en deux clans de tailles égales et un enracinement au niveau de la plus longue branche de l'arbre.

6.5.2 Jackknife

Comme précédemment, 100 arbres vont être générés, afin de réaliser un Jackknife sur les arbres créés par `FoldTree`. `FoldTree` a besoin d'un dossier adéquat de départ pour fonctionner, c'est pourquoi il fallait créer pour chaque arbre un dossier de base pour le logiciel. Le programme `SnakeM_myfam.pl` (programme11) génère 100 dossiers myfam contenant les fichiers PDB et le fichier `identifiers.txt`. `SnakeM_myfam.pl` (programme11) demande en entrée le nom du domaine que l'on souhaite traiter (domaine 1, domaine 2, domaine 3, combinaison du domaine 1 et du domaine 2, combinaison du domaine 2 et du domaine 3 ou protéine complète). Il va ensuite aller rechercher le dossier correspondant contenant l'ensemble des fichiers PDB et `identifiers.txt`. Il va recopier 100 fois le contenu de ce dossier dans des dossiers nouvellement

créés. Pour chaque dossier il va supprimer aléatoirement un identifiant de séquence de chaque famille. Une fois, l'ensemble des dossiers créés, le script `foldtree.sh` (programme12) va permettre de lancer le programme `FoldTree` pour l'ensemble de ces dossiers à l'aide d'une boucle.

6.6 Consurf

La conservation des AAs au sein de chaque famille de Mur ligases a été déterminée par `Consurf` (programme16). Ce dernier est un programme open source (https ://consurf.tau.ac.il/consurf_index.php) qui prend en entrée un fichier PDB, un identifiant de la base de données **UniProt** ou une séquence d'acides aminés. Dans cette analyse, un fichier PDB d'une protéine sélectionnée parmi celles présentes dans la liste (cfr liste) est utilisé. Plusieurs paramètres sont ensuite sélectionnés manuellement. Un fichier comportant un alignement multiple de séquences, est téléchargé sur `Consurf`. Ce fichier est créé au préalable en alignant par `MAFFT` (programme17) une série de séquences de Mur ligases (cfr Seq_Consurf.xlsx). On spécifie dans l'alignement la séquence associée à la structure PDB. Le reste des paramètres est laissé par défaut. `Consurf` (programme16) donne une série de fichiers de sortie qui permettent d'analyser le niveau de conservation des acides aminés présents dans la structure. Ce niveau de conservation est représenté par un code couleur, soit sur la séquence d'acides aminés, soit sur un fichier PDB qui contient la séquence donnée colorée avec les couleurs correspondantes. Ce fichier PDB est visualisé sur `Pymol`.

7 Résultats

7.1 Protocole manuel (Distance RMSD)

7.1.1 Sélection et prédiction des structures tridimensionnelles

À la suite des recherches des fichiers PDB sur les bases de données (**AlphaFold** et **Uniprot**), sept protéines (WP_039704973.1, WP_088335895.1, WP_088336155.1, WP_088336109.1, WP_088335892.1, WP_039704394.1, WP_039704550.1) sur 45 (cfr liste) n'avaient pas de fichier associé dans les bases de données. Nous avons donc dû prédire la structure de ces protéines via la version en ligne de commandes d'**AlphaFold**. **AlphaFold** donne en fichiers de sortie plusieurs modèles. Les modèles relaxés [5](relaxed_model_*_pred.pdb) ont été sélectionnés pour les 7 structures prédites.

Les structures nous ont permis de réaliser un alignement tridimensionnel entre séquences protéiques afin d'obtenir des valeurs RMSD. Ces valeurs ont été calculées via `Pymol` par alignement des différents domaines et combinaisons de domaines entre les protéines sélectionnées (Fig.9). Une valeur de RMSD supérieure à 5 Å montre que les protéines ne sont pas alignées entre elles (communication personnelle : F. Kerff). La distance RMSD des protéines de séquences identiques est inférieure à 0.5 Å (tous les atomes sont alignés)(Illergård et al., 2009). Plus la valeur s'approche de 5 Å, plus le nombre d'atomes alignés diminue. Au-dessus de 5 Å aucun atome ou presque n'est aligné, les protéines ne sont donc pas alignables.

5. Les modèles relaxés sont obtenus par une minimisation d'énergie avec le champ de force Amber à partir du meilleur modèle non relaxé initialement obtenu. Cette minimisation permet de corriger les erreurs géométriques et énergétiques, afin que la structure soit conforme aux contraintes physiques et chimiques qui existent à l'état naturel (Varadi et al., 2024).

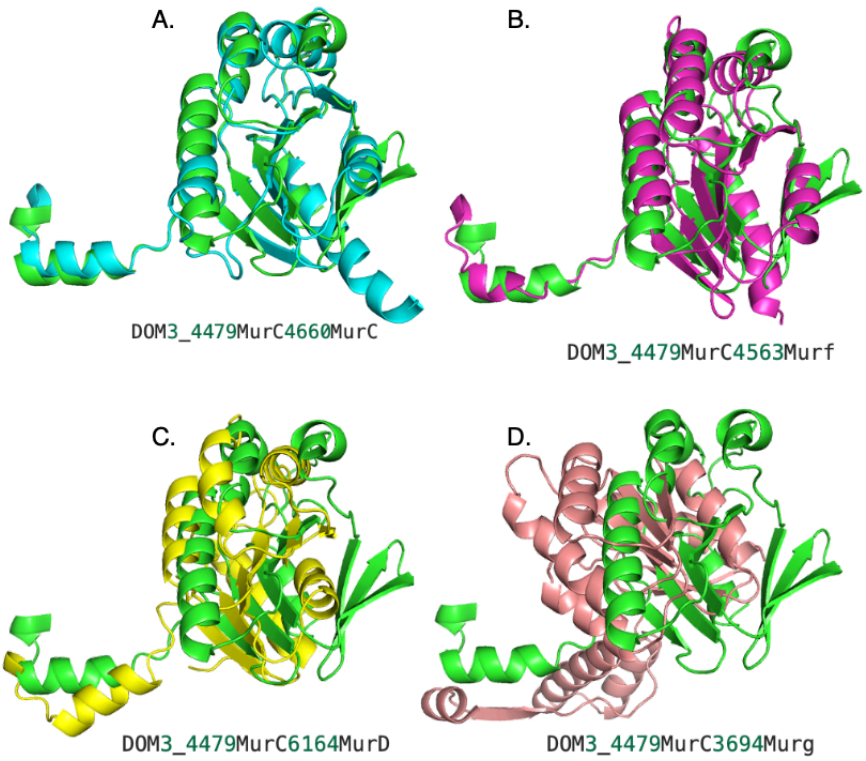

FIGURE 9 – Alignement des structures du domaine 3. Afin de visualiser ce que signifie les valeurs de RMSD, voici différents alignements qui correspondent à des valeurs de RSMD précises. WP_039704479MurC est représenté en vert. A) Comparaison entra famille, WP_039704479MurC et WP_011244660MurC en blue (0.898). B) WP_039704479MurC et WP_011244563MurF en magenta (4.914). C) WP_039704479MurC et WP_034526164MurD en jaune (7.601). D) WP_039704479MurC et WP_011953694Murg en rose clair (13.075)

WP_039704479@MurC a donc été manuellement alignée avec les protéines WP_011244660@MurC, WP_011244563@MurF, WP_034526164@MurD et WP_011953694@Murg sur Pymol comme exemple d'alignement. Bien que la limite a été fixée à 5 Å (RMSD), les valeurs de distance supérieures gardent tout de même des informations importantes pour la formation des arbres.

7.1.2 Matrice RMSD

7.1.2.1 Statistiques descriptives Les données, obtenues suite à l'alignement des fichiers PDB (RMSD brut), ont été analysées sous différentes formes afin de visualiser les distributions des valeurs de RMSD brutes. Les distances au sein des familles, les distances entre les familles, et les distances à l'intérieur de chaque famille ont été examinés. Ces analyses ont été effectuées pour différentes combinaisons de domaines, allant d'un domaine unique jusqu'aux trois domaines combinés.

7.1.2.2 Distributions des distances RMSD inter-familles. Les distributions des valeurs de RMSD intra et inter-familles pour la protéine complète sont montrées dans la figure 2 (Fig.11). Les valeurs extrêmes (considérées comme étant des valeurs supérieures à 9 dans la comparaison entre familles) sont de moins de 1% pour tous les domaines et combinaisons de domaines à l'exception du domaine 1 qui compte 18% de valeurs extrêmes. Le domaine 2 comporte un maximum de RMSD de 4.666 Å et les valeurs de RMSD entre 1.25 et 1.5 Å représentent les valeurs les plus fréquentes dans la distribution de données. Cette distribution se retrouve dans l'ensemble des domaines (et combinaisons de domaines) à l'exception du domaine 1 (Fig.10). Le domaine 1 montre une distribution avec un premier maximum aux alentours de 2 Å et un deuxième maximum (moins important) entre 8 et 13 Å de RMSD pour les valeurs entre les familles.

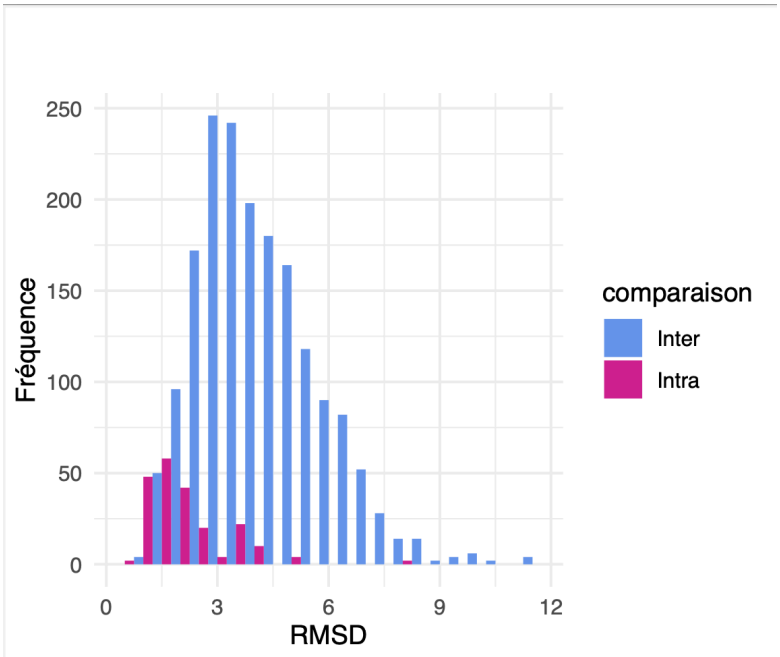

FIGURE 10 – Histogramme de distribution de l'ensemble des valeurs RMSD séparés inter et intra famille pour la protéine complète. Les valeurs RMSD entre les familles sont représentées en bleu. Les valeurs RSMD à l'intérieur des familles sont représentées en rose.

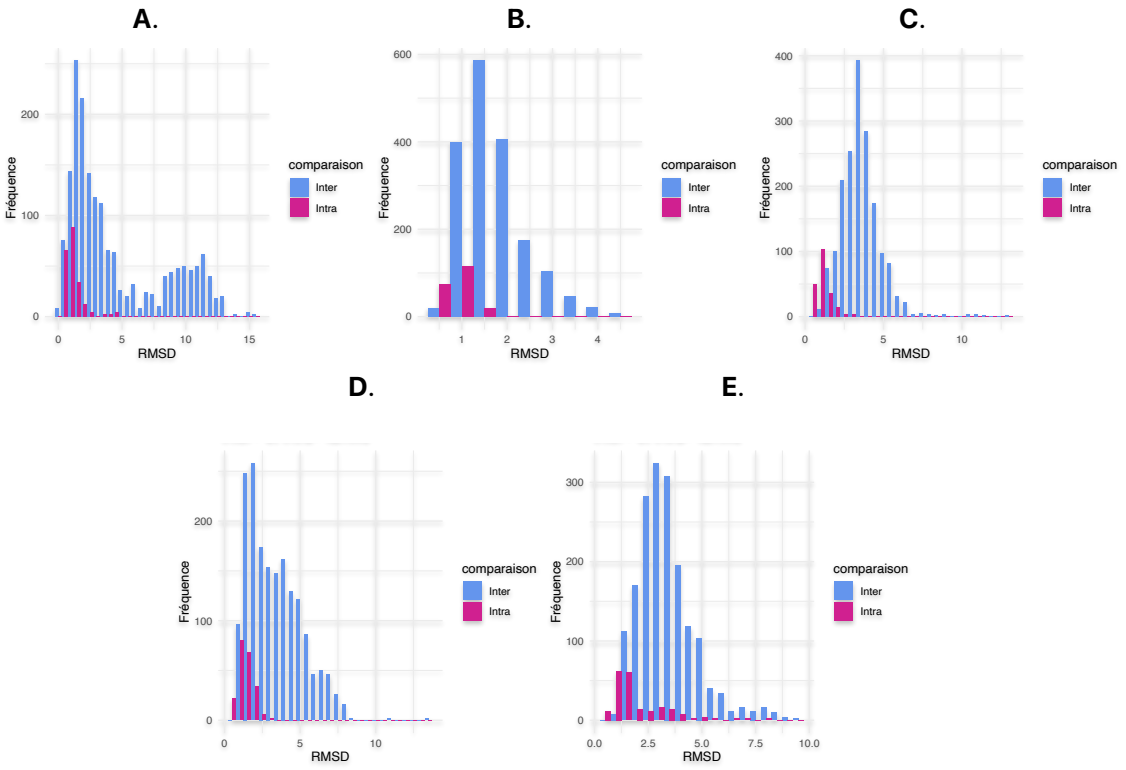

FIGURE 11 – A. Histogramme de distribution des valeurs RMSD (intra et inter familles) pour le domaine1. B. Histogramme de distribution des valeurs RMSD (intra et inter familles) pour le domaine 2. C. Histogramme de distribution des valeurs RMSD (intra et inter familles) pour le domaine 3. D. Histogramme de distribution des valeurs RMSD (intra et inter familles) pour le domaine 1 et 2. E. Histogramme de distribution des valeurs RMSD (intra et inter familles) pour le domaine 2 et 3.

7.1.2.3 Distribution des distances RMSD intra-familles. Pour les valeurs intra-familles, les valeurs entre 1 et 1.25 Å correspondent aux valeurs les plus fréquentes, et la distribution est condensée au niveau des valeurs faibles avec une médiane de 0.8385 Å. Des valeurs supérieures à 5 Å de RMSD sont retrouvées dans les distances RMSD intra-familles (cfr section matériel et méthode), ce qui signifie que tous les domaines 2 ne s'alignent pas. Ces exceptions sont de l'ordre de moins d'1% pour les comparaisons intra-familles de l'ensemble de la protéine et de l'ordre de 3.8% pour la combinaison du domaine 2 et du domaine 3 (les autres domaines sont à 0%). Toutefois, si on calcule les RMSD domaine par domaine, on obtient des résultats très différents (Fig.12). La famille MurD possède les valeurs les plus extrêmes. Effectivement, pour la protéine globale ainsi que pour la combinaison des domaines 2 et 3, les valeurs de RMSD pour la famille MurD atteignent jusqu'à 8.119 Å (WP_039704394 vs WP_041928127). Ces valeurs montrent qu'il existe déjà des différences importantes entre les protéines de la famille MurD. Le domaine 2 montre pour l'ensemble des familles des valeurs qui s'arrêtent aux alentours de 2 Å (RMSD) (Fig.13), ce qui suggère un haut taux de conservation entre les protéines de mêmes familles au niveau de ce domaine.

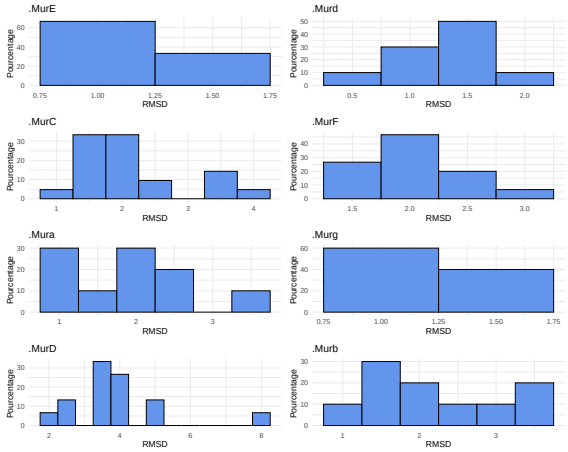

FIGURE 12 – Histogramme de distribution des valeurs RMSD intra famille pour la protéine complète (domaine1/2/3). Zoom sur les valeurs intra familles et sur les valeurs extrêmes.

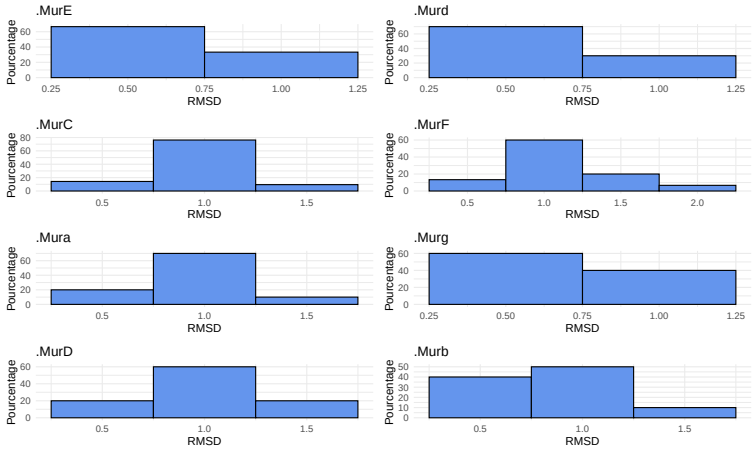

FIGURE 13 – Histogramme de distribution des valeurs RMSD intra famille pour le domaine 2.

7.1.2.4 Boxplots des valeurs RMSD. Les boxplots réalisés avec les valeurs RMSD sur l'ensemble des protéines montrent que MurE est la protéine qui est la plus proche structurellement de Murα, Murβ et Murγ. MurE est plus éloignée de MurF et MurD qui sont des Mur ligases bactériennes, que des protéines achéennes (Fig.14). Pour le domaine 1, le RMSD entre MurE et MurF est inférieur à 2 Å. Les RMSD entre MurE et Murδ est supérieur à 3 Å. L'ensemble des autres comparaisons montre des valeurs au-dessus de 4 Å avec comme valeurs extrêmes le RMSD avec Murβ qui est supérieur à 9 Å. Le domaine 1 de la protéine MurE est donc fortement similaire au domaine 1 de MurF et totalement différent du domaine 1 de la protéine Murβ (Fig.15).

Le domaine 2 est le domaine le plus conservé. Les boxplots corroborent cette hypothèse avec des valeurs de RMSD globalement inférieures à 2 Å à l'exception de MurF qui est légèrement au-dessus de 2 Å. Le domaine 3 est le domaine le plus différent entre les différentes familles de protéines. Comme pour les protéines globales, MurE semble plus proche des protéines archéennes sur les boxplots (Fig.15).

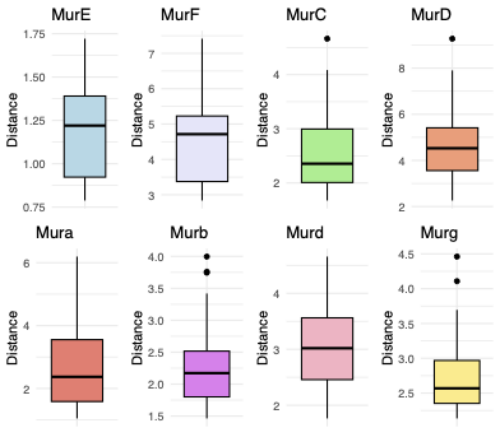

FIGURE 14 – Boxplots de distribution des valeurs RMSD entre la famille MurE et le reste des familles pour la combinaison des trois domaines.

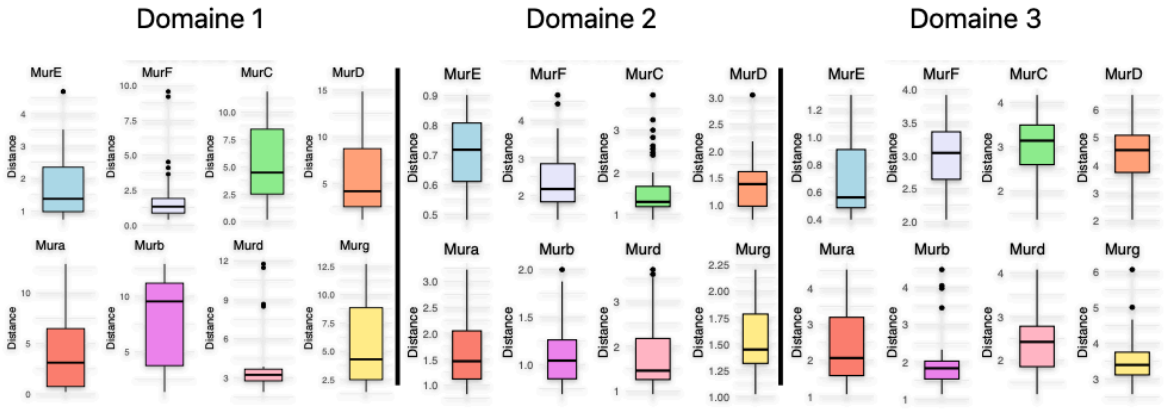

FIGURE 15 – Boxplots de distributions des valeurs RMSD entre la famille MurE et le reste des familles pour les domaines 1, 2 et 3.

MurD est un cas particulier. En effet, les boxplots qui représentent les valeurs RMSD pour l'ensemble des

protéines montrent que MurD est très éloigné de toutes les autres familles mais également de sa propre famille de protéines. Les protéines MurD se différencient fortement entre elles en fonction des espèces. Ces différences sont également visibles entre le mélange du domaine 2 et 3. Le domaine 2 montre toujours qu'il est le plus conservé de tous les domaines. Le domaine 2 étant le plus conservé, afin d'approfondir l'analyse, la niveau de conservation de la combinaison des trois domaines a été réalisé sur Consurf. L'analyse sur Consurf montre un haut taux de conservation au niveau de la poche réactionnelle de la protéine, et ce pour l'ensemble des familles (ex (Fig.16). De plus, Consurf met en évidence le domaine 2 comme le domaine le plus conservé avec le motif GXXGKT/S(Kouidmi et al., 2014) présent dans l'ensemble des familles également.

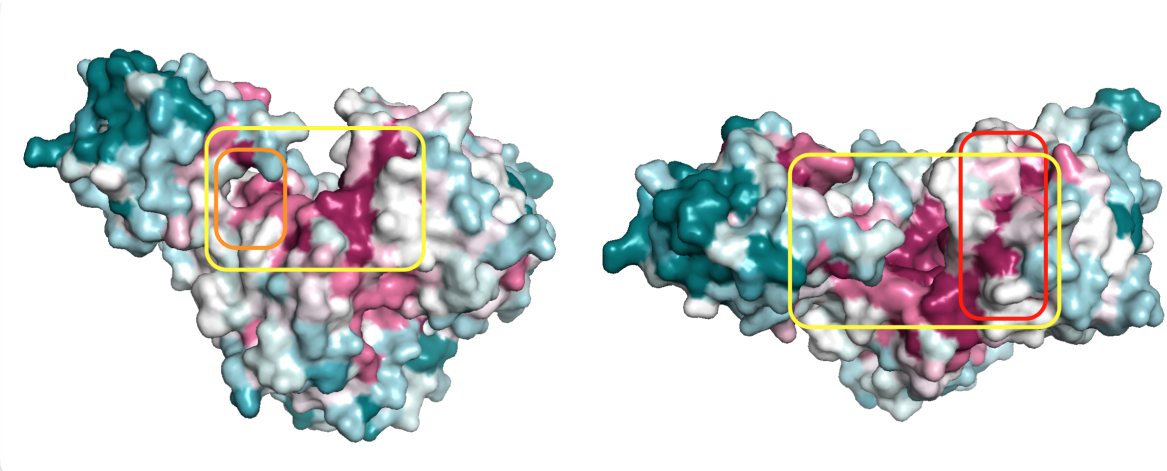

FIGURE 16 – Consurf pour la protéine MurE. En rose foncé, les AAs les plus conservés. En bleu foncé, les AAs les moins conservés. Le cadre orange représente la zone de liaison à l'ATP. Le cadre en rouge représente la zone potentielle de liaison à l'UDP. Le cadre en jaune représente la poche réactionnelle.

7.1.2.5 SDM et arbres phylogénétiques Les données ajustées (SDM), obtenues par alignement des fichiers PDB, ont permis de créer des arbres phylogénétiques. La méthode de phylogénie structurale, basée sur le SDM, donne des arbres qui se différencient en fonction du domaine ou de la combinaison de domaines qui est étudié (Fig.17). La combinaison des trois domaines, la combinaison du domaine 1 et du domaine 2, ainsi que les domaines 2 et 3 comportent des bipartitions qui regroupent les séquences de même famille entre elles. Cependant, les distances entre les différentes familles diffèrent d'un arbre à l'autre et donc d'un domaine (ou groupement de domaines) à l'autre. En ce qui concerne le domaine 1 (Fig.18), les familles Murα, MurD, MurF et MurE sont despersées à travers l'arbre et ne forment pas de groupe distinct. Le groupement du domaine 2 et du domaine 3 montre également des familles divisées, telles que MurC, MurD et Murβ qui ne forment pas de groupes monophylétiques indépendants des autres familles.

FIGURE 17 – Phylogénie structurale basée sur la distance SDM de la combinaison des trois domaines. En rouge, la famille des Murα. En fushcia, la famille des Murβ. En rose, la famille Murδ. En jaune, la gamille Murγ. En vert, la famille MurC. En bleu, la famille MurE. En mauvre, la famille MurF. En orange, la famille MurD.

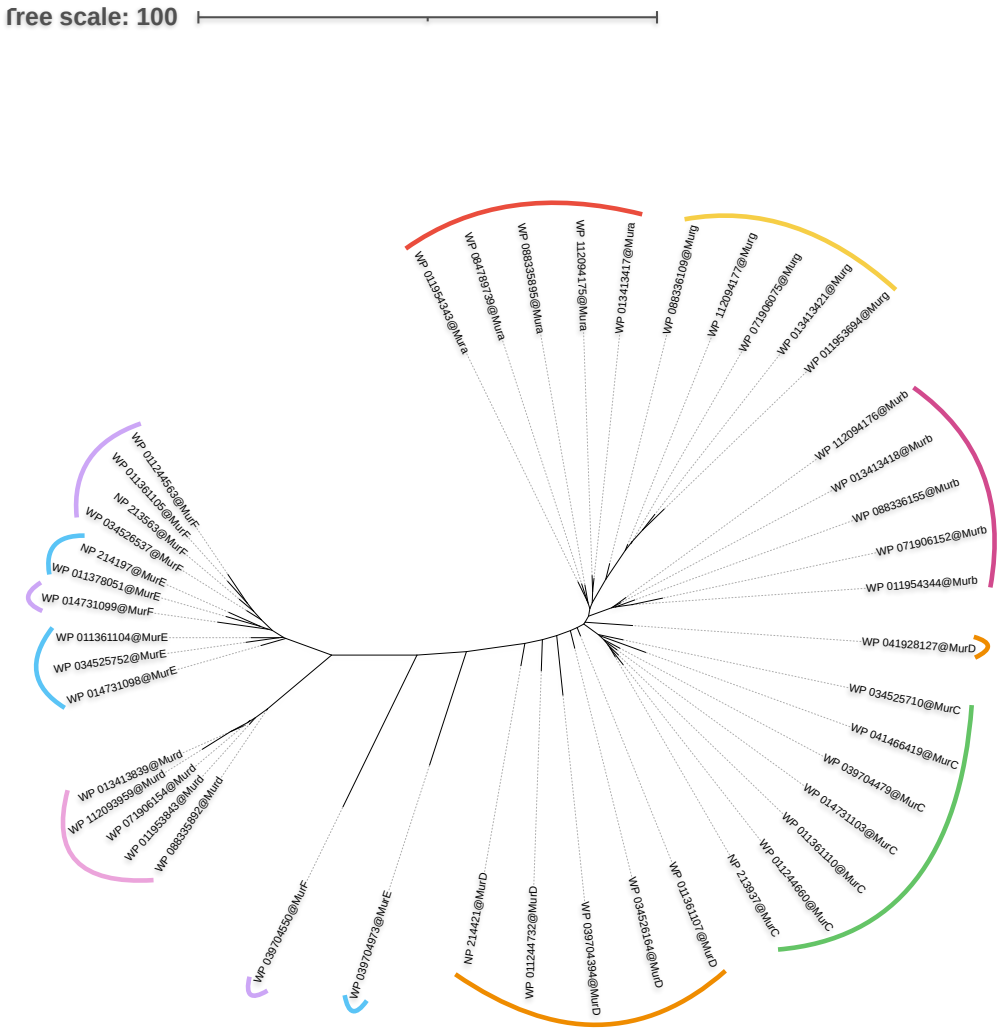

FIGURE 18 – Phylogénie structurale basée sur la distance SDM du domaine 1. En rouge, la famille des Murα. En fushcia, la famille des Murβ. En rose, la famille Murδ. En jaune, la gamille Murγ. En vert, la famille MurC. En bleu, la famille MurE. En mauvre, la famille MurF. En orange, la famille MurD.

Les clans qui se composent de plus d'une famille de protéines que l'on retrouve dans les différents arbres phylogénétiques sont les suivante :

	Mur ligases qui composent le clan	Domaines pour lesquels on retrouve le clan
1	Murα/Murδ	3
2	Murα/MurF	2, combinaison 1 et 2, et combinaison 2 et 3
3	Murβ/MurE	2
4	Murγ/Murδ	combinaison des trois domaines
5	Murγ/MurC	2

	Mur ligases qui composent le clan	Domaines pour lesquels on retrouve le clan
6	Murγ/MurD	3
7	MurD/Murβ/Murγ	3 et combinaison 1 et 2
8	Murα/Murβ/Murγ	1
9	MurE/MurF/Murδ	1
10	MurD/MurE/Murβ/Murγ	combinaison 1 et 2

L'ensemble des arbres obtenus (ensemble des trois domaines, domaine 1, domaine 2, domaine 3, domaines 1/2 et domaines 2/3) se trouvent en annexes.

7.2 Phylogénie structurale des Mur ligases par FoldTree

Les méthodes de phylogénie structurale, proposées par le programme `FoldTree` donnent également des arbres qui se différencient en fonction du domaine ou de la combinaison de domaines qui est étudié (Fig.19).Pour les trois types de distances utilisées par FoldTree (Fident, lDDT et MT), la combinaison des trois domaines, la combinaison des domaines 2 et 3, la combinaison des domaines 1 et 2, ainsi que le domaine 3 présentent les différentes familles de protéines regroupées entre elles. Cependant, pour le domaine 1 et le domaine 2 des différences sont visibles entre les méthodes (cfr Fig19).

Les clans retrouvés composés de plus d'une famille de protéines que l'on retrouve sont les suivantes :

	Mur ligases qui composent le clan	Présent dans le domaine 1	Présent dans le domaine 2	Présent dans le domaine 3	Présent dans la combinaison des domaines 1 et 2	Présent dans la combinaison des domaines 2 et 3	Présent dans la combinaison des domaines 1, 2 et 3
1	Murα/Murβ	FT - lDDT - MT	FT	FT	FT - lDDT	FT	FT - lDDT
2	Murα/MurF	/	lDDT - MT - RMSD	MT	RMSD	RMSd	RMSD
3	Murβ/MurE	/	RMSD	FT - lDDT - MT	RMSD	RMSD	RMSD
4	Murγ/Murδ	/	/	FT	/	MT - RMSD	RMSD
5	Murγ/MurD	/	lDDT	FT - lDDT - MT - RMSD	lDDT - MT	FT - lDDT	FT - lDDT
6	MurD/MurE	FT	/	/	/	/	/

	Mur ligases qui composent le clan	Présent dans le domaine 1	Présent dans le domaine 2	Présent dans le domaine 3	Présent dans la combinaison des domaines 1 et 2	Présent dans la combinaison des domaines 2 et 3	Présent dans la combinaison des domaines 1, 2 et 3
7	MurC/MurF/Murδ	/	FT	FT	/	FT	FT
8	MurD/Murβ/Murγ	/	lDDT - MT	RMSD	MT - RMSD	MT	
9	MurE/Murα/Murβ	/	/	FT - lDDT - MT	/	FT - lDDT	/
10	Murα/Murβ/Murγ	FT - lDDT - MT - RMSD	FT	/	FT	/	/
11	MurE/MurF/Murδ	lDDT - MT - RMSD	/	/	FT - lDDT - MT	/	lDDT - MT
12	MurD/Murγ/MurE	/	/	/	/	FT	/
13	MurD/Murγ/Murα/Murβ	/	MT	RMSD	FT - lDDT	/	/

L'apparition de ces différents clans est liée au domaine étudié et à la méthode utilisée.

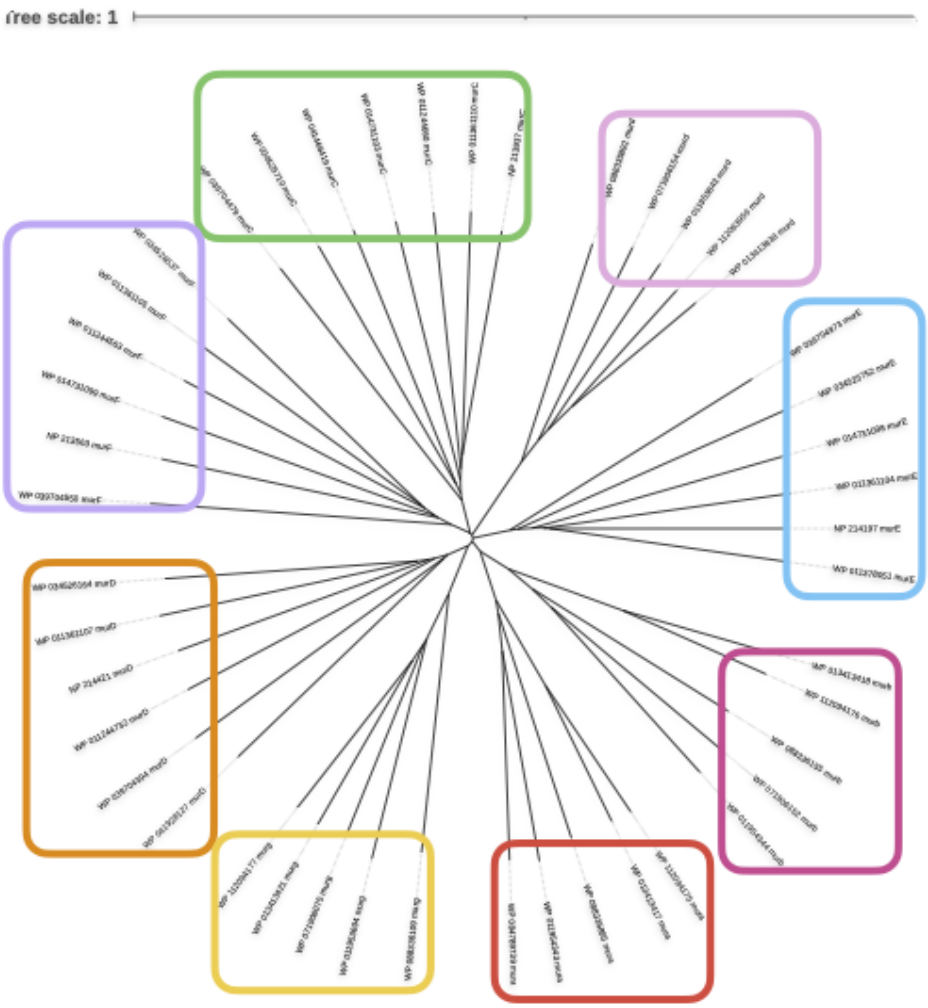

FIGURE 19 – Phylogénie structurale basée sur le Fident de l'ensemble des protéines.FoldTree permet de créer des arbres sur base de la lDDT, MT ou Fident. Lors de cette étape 18 arbres sont donc calculés.En rouge, la famille des Murα. En fuscia, la famille des Murβ. En rose, la famille Murδ. En jaune, la gamille Murγ. En vert, la famille MurC. En bleu, la famille MurE. En mauvre, la famille MurF. En orange, la famille MurD.

7.3 Comparaison des différentes approches phylogénétiques

V.Lupo, durant son doctorat a analysé la phylogénie des Mur ligases par différentes méthodes. Les méthodes réalisées par Valérian et les méthodes réalisées dans le cadre de ce mémoire ont été rassemblées dans une seule et même figure (Fig.21). La comparaison de l'ensemble des arbres obtenus avec les différentes méthodes utilisées lors de ce travail, ainsi que les différentes méthodes utilisées par V. Lupo lors de sa thèse a permis de mettre en évidence plusieurs points :

1. dans 82% [6] des cas, les séquences protéiques se regroupent par famille (domaines et méthodes confondus).
2. dans 44% des cas, les familles ne se regroupent pas entre elles dans le domaine 1.
3. dans 37.5% des cas, le clan Murα/Murβ est présent (domaines et méthodes confondus).
4. dans 37.5% des cas, le clan Murγ/MurD est présent (domaines et méthodes confondus).
5. dans 29.1% des cas, le clan MurE/MurF/Murδ est présent (domaines et méthodes confondus).
6. Aucun clan, en dehors des regroupements des familles, n'est présent dans plus de 50% des cas.

Type de données	Programme	Méthodes	Utilisateur
Séquences AAs	BAli-Phy	Baliphy	V. Lupo
Séquences AAs	IQ-TREE	C20	V. Lupo
Séquences AAs	IQ-TREE	C40	V. Lupo
Séquences AAs	IQ-TREE	LG4X	V. Lupo
Structures protéiques	FoldTree	Distance Fident	C. Mullender
Structures protéiques	FoldTree	Distance lDDT	C. Mullender
Structures protéiques	FoldTree	Distance MT	C. Mullender
Structures protéiques	Pipeline C. Mullender	Distance RMSD	C. Mullender

6. 100% étant lorsque l'ensemble des arbres obtenus (domaines, combinaison de domaines et méthodes confondues) possèdent le clan.

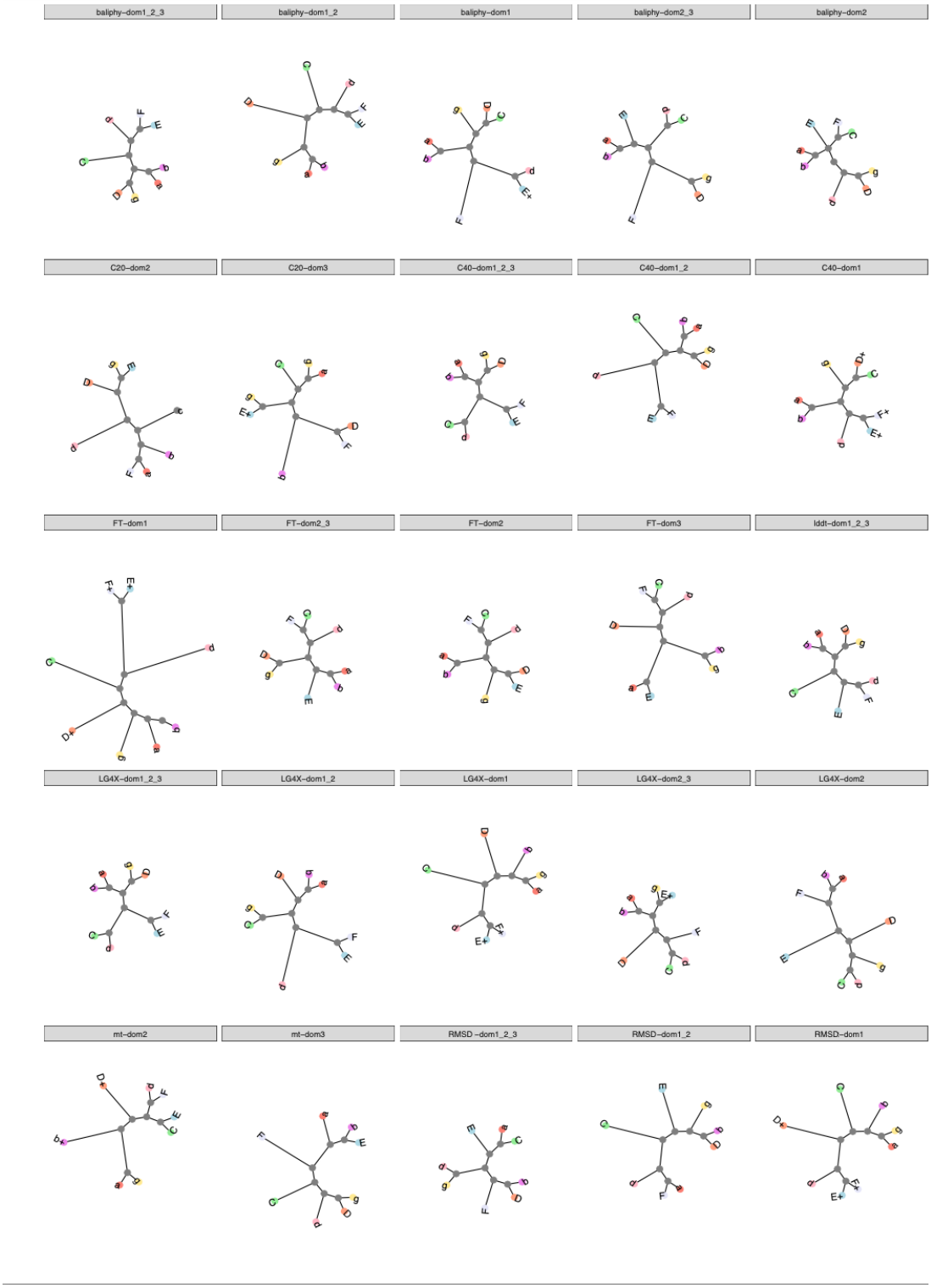

FIGURE 20 – Figure (part 1) rassemblant l'ensemble des arbres obtenus pour V. Lupo et C. Mullender. Les séquences ont été regroupés par famille en une seule branche unique afin de faciliter la lecture. Les arbres ont ensuite été triés par méthode et par domaine.

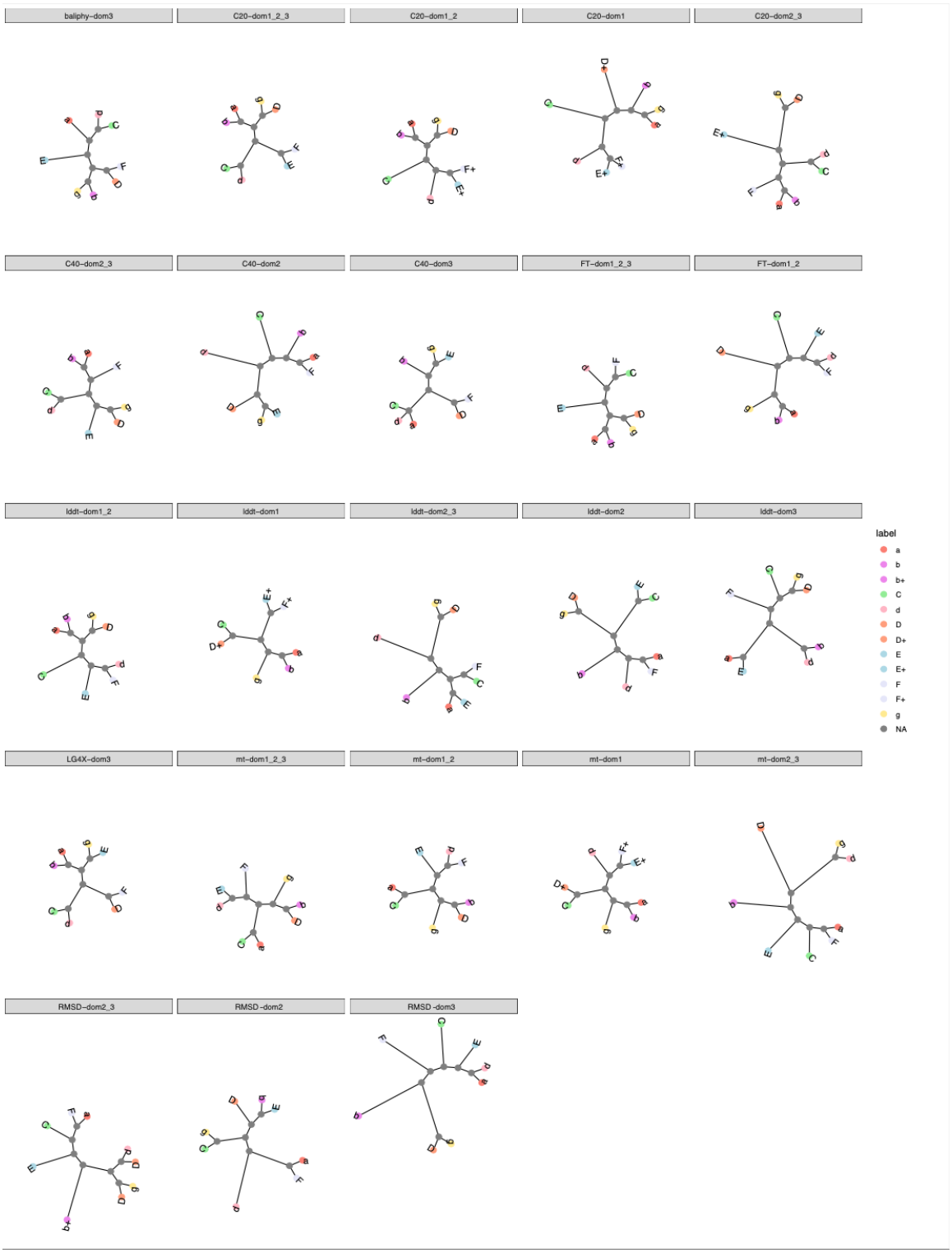

FIGURE 21 – Figure (part 2) rassemblant l'ensemble des arbres obtenus pour V. Lupo et C. Mullender. Les séquences ont été regroupés par famille en une seule branche unique afin de faciliter la lecture. Les arbres ont ensuite été triés par méthode et par domaine.

7.3.1 Phylogénie structurale des archées et des bactéries

En raison de l'absence de consensus parmi les différentes méthodes de phylogénie lorsque les familles archéennes et bactériennes sont considérées simultanément, nous avons décidé de concentrer notre étude sur les phylogénies des Mur ligases bactériennes et archéennes prises séparément. Deux approches d'enracinnement ont été appliquées : un enracinement séparant l'arbre en deux clans de tailles égales et un enracinement au niveau de la plus longue branche de l'arbre. La deuxième méthode a été rejetée, car les longueurs de branches n'étaient généralement pas suffisamment différentes pour déterminer laquelle était la plus longue parmi l'ensemble.

Suite à la visualisation de l'ensemble des arbres, deux figures représentant les séquences correspondant aux bactéries et les séquences correspondantes aux archées ont été créées.
Lorsque des arbres uniquement composés des séquences archéennes sont réalisés, deux bipartitions ressortent (Fig.23). La première est composée d'un groupe avec Murα/Murβ et un groupe avec Murδ/Murγ. La deuxième bipartition comporte un premier groupe avec Murβ/Murγ et un deuxième avec Murδ/Murα. Les associations Murα/Murγ, ainsi que Murβ/Murδ ne sont jamais retrouvées.

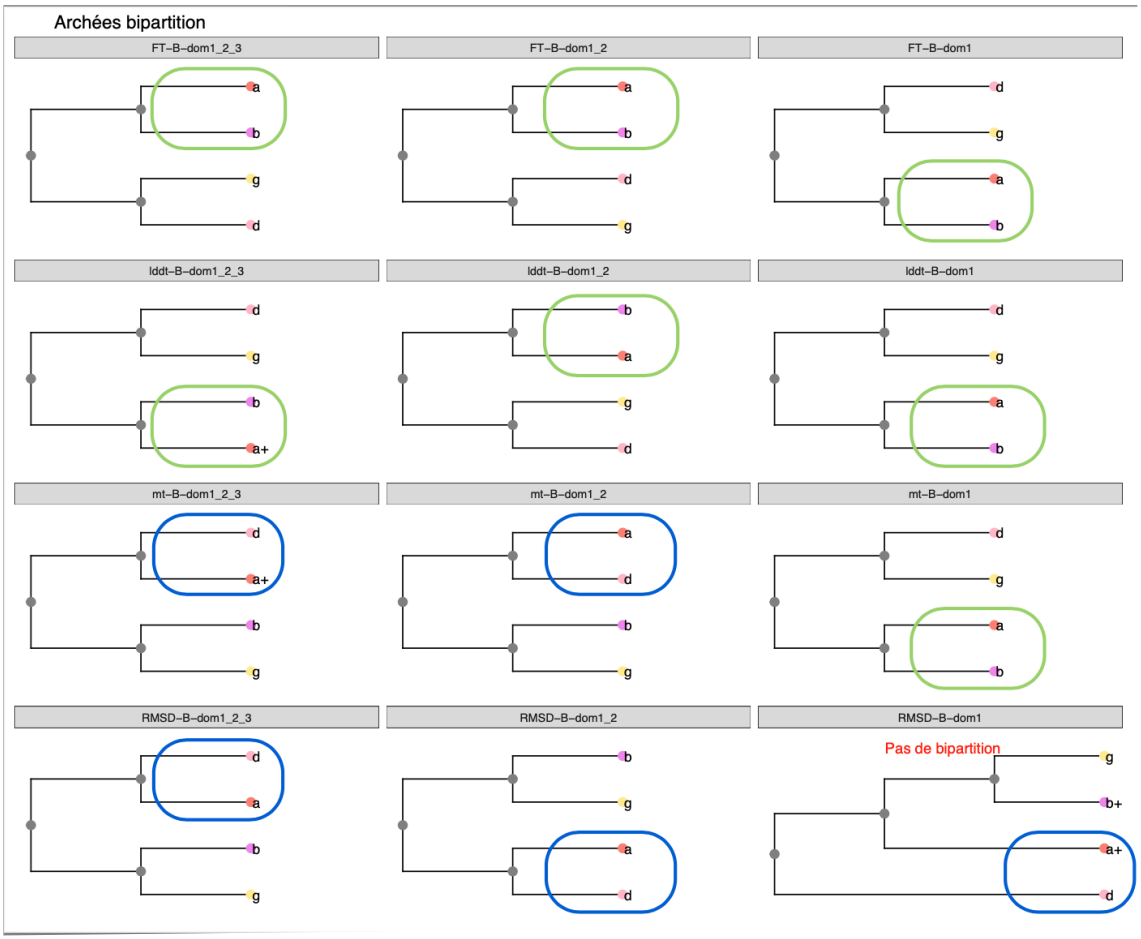

FIGURE 22 – Phylogénie archéenne avec le RMSD, MT, lDDT et Fident (part 1). Le premier clan (Murα/Murβ) retrouvé est en vert dans 54,16% des arbres. Le deuxième clan (Murα/Murδ) trouvé est en bleu dans 37,5% dans les arbres. Les arbres ont été réécrits afin de fusionner les séquences d'une même famille en une seule feuille pour faciliter la lecture.

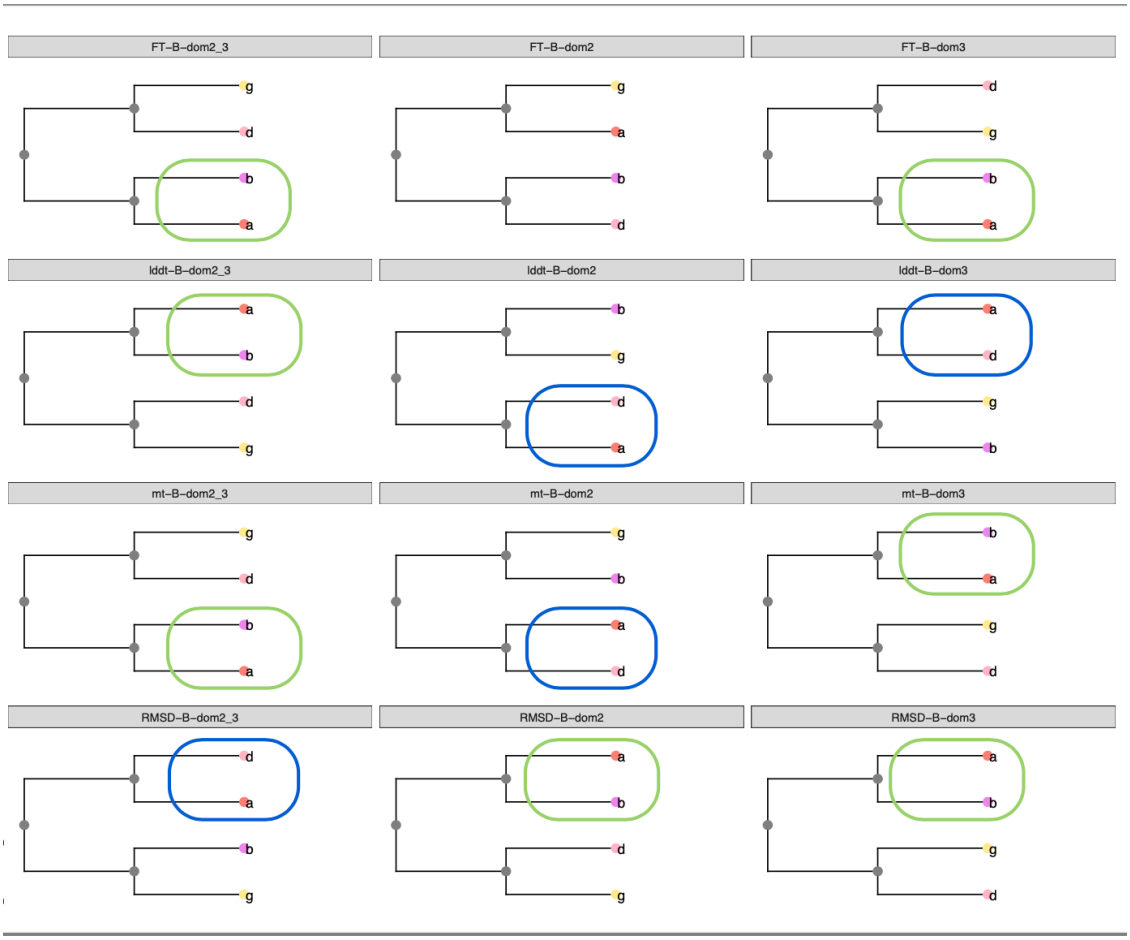

FIGURE 23 – Phylogénie archéenne avec le RMSD, MT, lDDT et Fident (part 2). Le premier clan (Murα/Murβ) retrouvé est en vert dans 54,16% des arbres. Le deuxième clan (Murα/Murδ) trouvé est en bleu dans 37,5% dans les arbres. Les arbres ont été réécrits afin de fusionner les séquences d'une même famille en une seule feuille pour faciliter la lecture.

Pour les phylogénies bactériennes les trois bipartitions possibles sont retrouvées, MurE/MurF et MurC/MurD, MurF/MurC et MurE/MurD, et MurF/MurD et MurE/MurC. Ces associations semblent être en lien avec le(s) domaine(s) des protéines considéré(s) (Fig.25). Le domaine 1 montre une bipartition de type MurF/MurE et MurC/MurD. Le domaine 2, une bipartion MurC/MurF et MurE/MurD. Le domaine 3, une bipartition MurE/MurC et MurD/MurF. Les groupes du domaine 1 semblent l'emporter sur les groupes du domaine 2 lorsqu'on les combine. Les groupes du domaine 2 semblent l'emporter sur les groupes du domaine 3 lorsqu'on les combine.

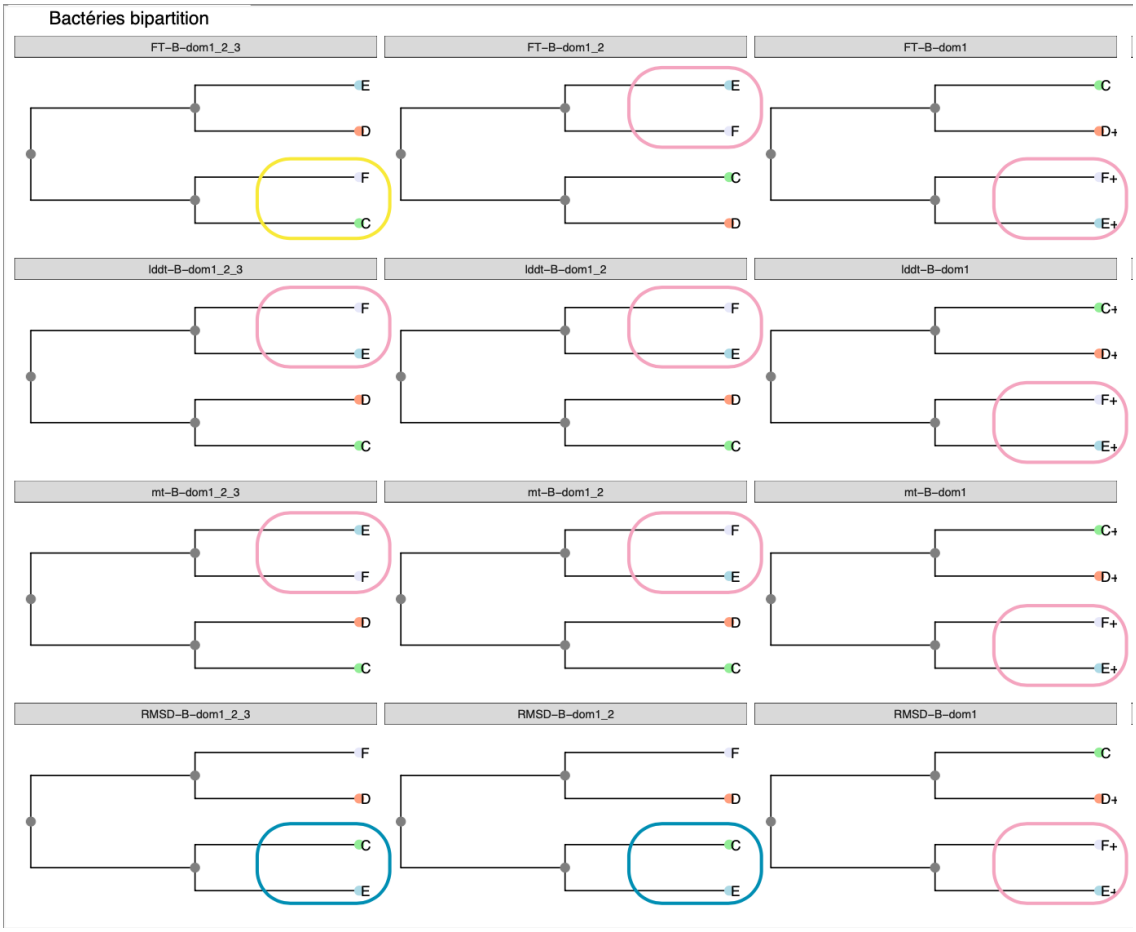

FIGURE 24 – Phylogénie bactérienne avec le RMSD, MT, lDDT et Fident (part 1). Le premier clan (MurF/MurC) est en jaune dans 33,34% des arbres. Le deuxième clan (MurE/MurF) est en rose dans 37,5% des arbres. Le troisième clan (MurE/MurC) est en bleu dans 29,167% des arbres.

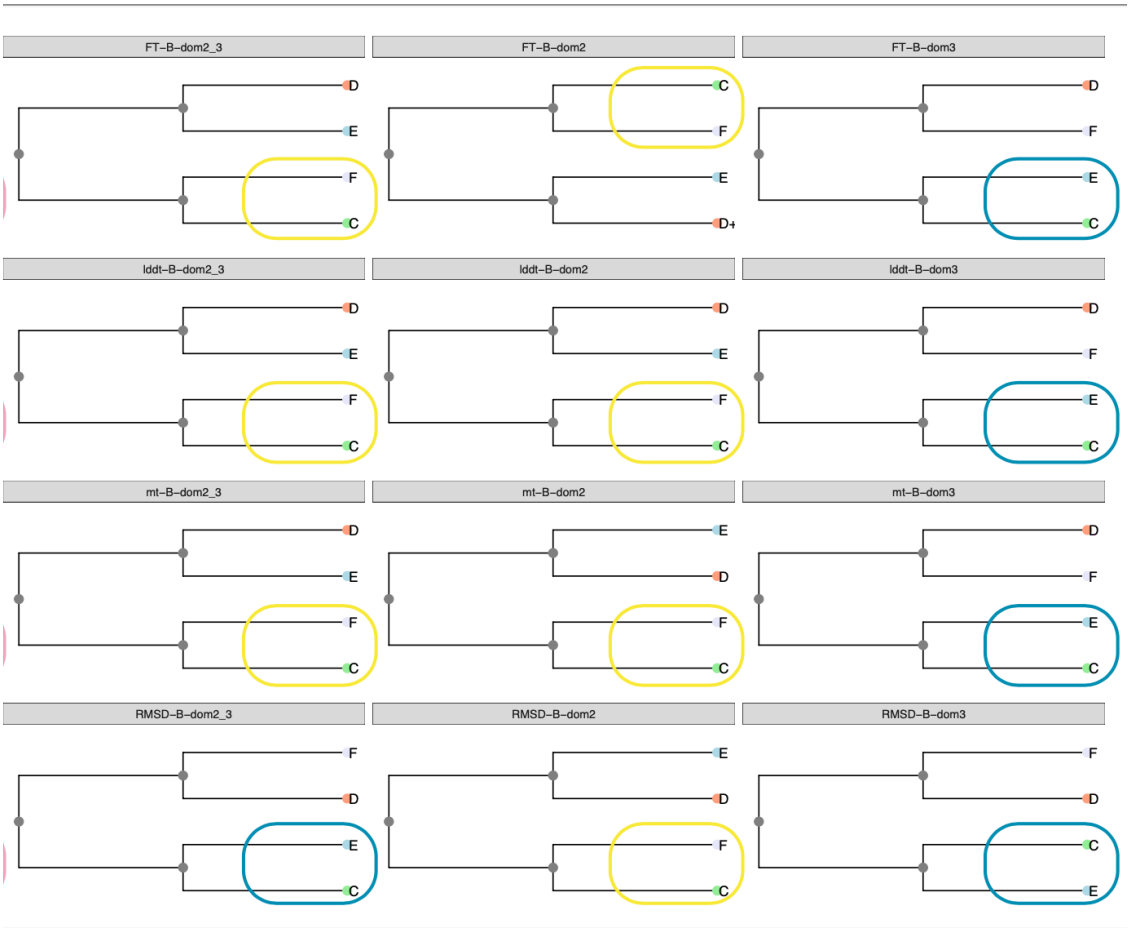

FIGURE 25 – Phylogénie bactérienne avec le RMSD, MT, lDDT et Fident (part 2). Le premier clan (MurF/MurC) est en jaune dans 33,34% des arbres. Le deuxième clan (MurE/MurF) est en rose dans 37,5% des arbres. Le troisième clan (MurE/MurC) est en bleu dans 29,167% des arbres.

7.4 Jacknife de séquences

La robustesse des différentes méthodes utilisées dans le cadre de ce travail a été testée par une méthode de Jacknife (rééchantillonnage des séquences utilisées pour créer les arbres phylogénétiques). Les méthodes Fident et lDDT semblent les plus robustes. La méthode RMSD semble la moins robuste (les valeurs inférieures à 10% sont considérées comme des topologies qui ne sont pas présentes dans les arbres ; les valeurs supérieures à 10% et inférieures à 90% sont considérées comme des topologies présentes mais pas de façon homogène dans les abres et les valeurs supérieures à 90% sont considérées comme des topologies présentes uniformément dans les arbres).

Les observations en termes de bipartitions sont similaires aux observations décrites dans la section “Comparaison des différentes approches phylogénétiques”. Les différentes familles de Mur ligases se séparent bien en huit clans pour le domaine 2, le domaine 3, la combinaison du domaine 1 et du domaine 2, la combinaison du domaine 2 et le domaine 3 et la combinaison des trois domaines. D’autres clans sont également formés, Murα/Murβ, Murα/MurF, Murβ/MurE, Murγ/Murδ, Murγ/MurD, MurC/MurF/Murδ , MurD/Murβ/Murγ, MurE/Murα/Murβ, Murα/Murβ/Murγ, MurE/MurF/Murδ, MurD/Murγ/MurE et

MurD/Murγ/Murα/Murβ (cfr section Comparaison des différentes approches phylogénétiques).

	FT						LDDT						MT						RMSD					
Familles	dom1	dom2	dom3	dom12	dom23	global	dom1	dom2	dom3	dom12	dom23	global	dom1	dom2	dom3	dom12	dom23	global	dom1	dom2	dom3	dom12	dom23	global
MurC	100	100	100	100	100	100	100	100	100	100	100	100	93	100	100	100	100	100	85	100	100	100	28	67
MurD	1	14	100	48	100	100	0	85	100	100	100	100	1	21	100	81	95	100	1	89	100	100	13	20
MurF	0	100	57	100	100	100	0	99	100	100	100	100	3	83	100	100	100	100	0	97	93	99	100	100
MurE	0	100	100	100	100	100	13	100	100	100	100	100	7	100	100	100	100	100	1	99	100	100	100	100
Mura	18	100	100	100	100	100	76	100	100	100	100	100	74	100	100	100	100	100	0	97	100	100	96	96
Murd	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Murb	93	100	100	100	100	100	100	100	100	100	100	100	100	20	100	100	80	100	93	70	96	100	27	27
Murg	100	100	100	100	100	100	100	100	100	100	100	100	99	100	100	100	100	100	100	100	100	100	100	100
Mura-Murb	97	26	25	78	100	100	88	0	1	100	2	100	84	0	0	0	0	0	0	0	0	0	0	0
Mura-Murd	0	0	5	0	0	0	0	0	12	0	0	0	0	6	0	0	0	0	0	0	10	0	16	16
Mura-MurF	0	1	2	0	0	0	0	100	0	0	50	0	0	32	36	5	89	0	0	93	0	60	44	51
Murb-MurE	0	0	27	0	0	0	0	0	15	0	38	0	0	0	71	0	0	0	0	54	0	11	14	16
Murg-Murd	0	5	13	0	0	0	0	0	0	0	0	0	0	0	0	0	85	0	0	0	0	22	34	40
Murg-MurC	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	88	0	0	0	0
Murg-MurD	0	0	20	0	87	100	0	80	100	100	100	100	0	0	100	35	5	0	0	0	96	0	5	10
Murg-MurE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	10
Murd-MurC	0	1	36	0	38	0	0	0	1	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0
MurD-MurE	0	44	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0
MurD-MurF	0	0	19	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MurE-MurF	6	0	0	24	0	0	4	0	0	13	0	0	4	0	0	0	0	0	2	0	7	0	0	0
MurC-MurF-Murd	0	43	16	0	70	100	0	0	0	0	0	0	0	4	0	0	0	0	0	0	8	0	0	0
MurD-Murb-Murg	0	0	9	0	0	0	0	27	0	0	0	0	0	11	0	100	0	100	0	0	98	36	2	2
MurE-Mura-Murb	0	0	38	0	40	0	0	0	32	0	24	0	0	0	41	0	0	0	0	1	0	0	0	0
MurF-Mura-Murb	0	0	0	0	0	0	0	2	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
Mura-Murb-Murg	70	26	0	84	0	0	67	0	0	0	0	0	60	0	0	0	0	0	66	0	0	0	0	0
MurC-Murd-Murg	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MurE-MurF-Murd	6	0	0	100	0	0	100	0	0	100	0	100	45	0	0	93	0	100	100	0	6	0	0	0
MurD-Murg-MurE	0	6	0	0	52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MurD-Murg-MurF	0	0	15	0	24	0	0	0	56	0	0	0	0	0	28	0	0	0	0	0	0	0	0	0
Mura-Murd-MurC	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0
Mura-Murg-MurC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Murb-Murg-MurE	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MurC-Mura-Murb-Murc	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MurD-MurE-Murb-Murg	0	0	3	0	0	0	0	0	0	0	0	0	0	1	1	2	0	0	0	0	2	28	2	5
MurE-Mura-Murb-Murg	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MurE-MurF-Mura-Murb	0	0	5	0	0	0	0	0	1	0	11	0	0	0	30	0	0	0	0	0	0	0	0	0
MurD-Murg-Mura-Murb	3	0	0	96	0	100	0	0	0	71	0	100	0	36	0	0	0	0	2	0	19	0	0	0
Murg-MurD-MurE-MurF	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIGURE 26 – Table récapitulative des Jackniffs.Les résultats Jackniff réalisés sur les arbres basés sur le RMSD et les arbres de FoldTree (lDDT, MT et Fident) ont été regroupés sous forme de tableau.

8 Discussion

8.1 Comparaison du protocole (RMSD) et de FoldTree

Ces analyses font suite aux études menées par Subedi et al. (2021), Subedi et al. (2022) et Lupo et al. (2022). Dans ces études, Subedi a présenté les toutes premières structures expérimentales de Mur ligases archéennes, Murα (Mfer336) et Murδ (Mfer762, Mfer734). Quant à l'étude de V. Lupo, ce dernier a essayé de résoudre la phylogénie des Mur ligases sur base des séquences primaires en AAs. Cependant, ces méthodes plus "traditionnelles" de phylogénie n'ont pas mené à un consensus pour résoudre l'histoire évolutive de la famille des Mur ligases, bien que certains clans, comme le clan MurE/F et le clan Murα/β, sont réccurents dans les phylogénies obtenues. Durant cette étude, nous avons mis en place un protocole de phylogénie structurale basé sur des matrices de distances entre les structures tridimensionnelles des différents domaines et différentes combinaisons de domaines des Mur ligases sur 45 séquences protéiques.

Le protocole développé sur base des mesures RMSD est la méthode la plus couramment utilisée (Breitling

et al., 2001 ; Bujnicki, 2000 ; Illergård et al., 2009 ; Johnson et al., 1990 ; Lakshmi et al., 2015 ; Loughran et al., 2008 ; Naveenkumar et al., 2022). Le RMSD est une mesure de distances qui permet de comparer des structures protéiques. Cette mesure augmente de façon linéaire avec la distance d'évolution entre les deux protéines (Illergård et al., 2009). Contrairement à certaines études, nous avons ajusté la valeur de RMSD afin d'obtenir des valeurs plus adéquates (Johnson et al., 1990). Le SDM ajoute à la mesure de RMSD la valeur de PFTE (qui représente le ratio du nombre d'équivalences sur le nombre total de résidus dans la plus petite protéine). Cette information supplémentaire permet de mieux caractériser les protéines de structures éloignées en considérant le repliement de la protéine plutôt que l'ensemble des atomes(Johnson et al., 1990 ; Naveenkumar et al., 2022). Pendant la réalisation de mon mémoire, FoldTree a été rendu disponible, mais les analyses ne fournissent pas la matrice de valeurs de distances en fichier de sortie. Notre protocole conserve et fournit la matrice utilisée pour générer l'arbre phylogénétique. Ces matrices permettent de réaliser toutes sortes d'analyses supplémentaires comme des analyses statistiques que FoldTree ne permet pas.

8.2 Phylogénie des Mur ligases et voie de biosynthèse

Dans l'étude précédente réalisée par Lupo et al. (2022) et communication personnelle, bien que la phylogénie des Mur ligases soit complexe à résoudre, les auteurs ont montré que les Mur ligases n'étaient pas présentes chez LUCA mais déjà présentes chez LBCA. De plus, ils ont également montré que les Mur ligases archéennes proviennent de deux ou trois transferts depuis les bactéries, puis se seraient diversifiées par duplications de gènes. Leurs arbres phylogénétiques indiquent que deux évènements de transfert horizontaux est l'hypothèse la plus probable. En effet, Murα, murβ et Murγ semblent avoir une origine commune avec MurD, tandis que Murδ est plus proche de MurC.

8.2.1 Vue globale des distributions inter-familles

Dans ce travail, des variations importantes des valeurs RMSD dans la comparaison entre les domaines inter-familles sont mises en avant (Fig.11). En effet, lorsqu'on compare les valeurs RMSD pour la protéine complète et les domaines individuels on remarque que les distributions montrent un mode aux alentours de 3 Å (RMSD) (Fig.10), alors que, lorsqu'on regarde les histogrammes de ces mêmes distributions mais en fonction des domaines individuels, on remarque des variations dans les distributions. Le domaine 1 comporte un mode principal de fréquences en dessous de 2.5 Å (RMSD) ce qui est inférieur à la protéine globale. De plus, ce domaine comporte un deuxième mode moindre aux alentours de 10 Å (RMSD), ce qui suggère la présence de paires de protéines dont la structure diffère beaucoup au niveau du domaine 1. Le domaine 2 possèdent une distribution de valeurs nettement plus compacte avec un mode en dessous de 1.5 Å (RMSD), ce qui montre la grande conservation de ce domaine entre les familles. Enfin, le domaine 3 possède un mode entre 3 et 4 Å (RMSD) (Fig.11).

Les résultats de ce travail ont été obtenus avec 45 séquences, ce qui ne représente qu'une petite partie de la diversité des Mur ligases. Un nombre réduit de séquences permet d'avoir des temps de calcul raisonnables pour les prédictions. La famille MurD possède de grandes variations de RMSD dans la comparaison des protéines complètes, allant d'une valeur inférieure à 2 Å jusqu'à une valeur supérieure à 8 Å (Fig.28). La comparaison des différentes familles de Mur ligases sur base de seulement quelques séquences par famille risque de ce fait d'induire un biais. Une solution aurait été de reprendre la liste non exhaustive mais conséquente (+/- 3000 séquences) obtenue dans les précédents travaux de V. Lupo afin de se limiter aux séquences dont les structures étaient déjà calculées dans la base de données **AlphaFold** ou de prédire les structures des séquences qui

n'étaient pas présentes dans **AlphaFold**. Cependant, les deux alternatives au protocole actuel auraient pris un temps de calcul considérable.

8.2.2 Analyses séparées des domaines

Étant donné la différence de conservation de structure à travers les différents domaines, nous avons analysé les domaines indépendamment. Ces analyses ont montré la grande complexité des Mur ligases. La phylogénie de ces protéines varie en fonction du domaine ou de la combinaison de domaines étudiés. De plus, certaines familles présentent des variations structurelles au niveau intra-familial (Table : résumé des résultats (Fig.32).

8.2.2.1 Domaine 2 Le domaine 2 est le domaine le plus conservé, il est d'ailleurs hautement conservé avec des valeurs de RMSD égales ou inférieures à 2.17 Å. Cette conservation provient probablement du motif (TGTNGKSTT) compris dans la ATP-binding P-loop (Subedi et al., 2022), qui permet la liaison à l'ATP, qui est logé au coeur de ce domaine (Fig.27).

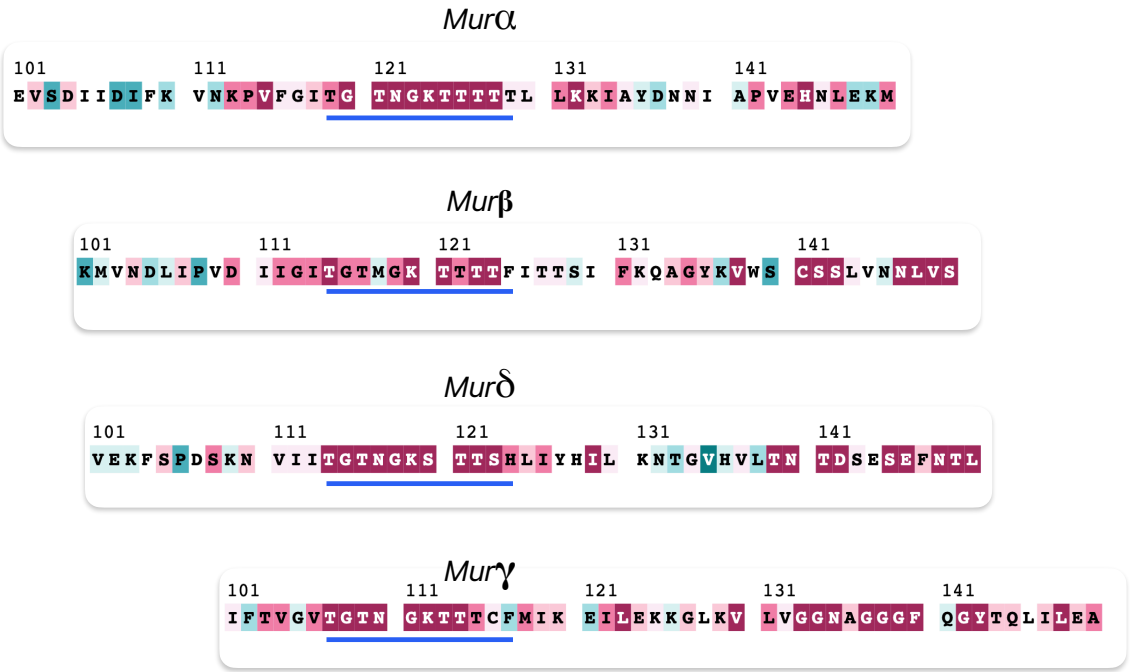

FIGURE 27 – Portion 101-151 du domaine 2 des Mur ligases archéennes. Souligné en bleu, le motif de liaison à l'ATP. Plus le rose est foncé, plus l'acide aminé est conservé et plus le bleu est foncé, moins l'acide aminé est conservé. Le motif TGTNGKSTT est fortement conservé chez les quatre familles de Mur ligases archéennes. Murα correspond à WP_011954343, Murβ correspond à WP_011954344, Murδ correspond à WP_011953843 et Murγ correspond à WP_011953694.

8.2.2.2 Domaine 1 Chez les Mur ligases bactériennes, le domaine 1 est le domaine qui reconnait la partie glucidique (UDP-MurNAc) du précurseur du PG, tandis que chez les archées, il reconnait certainement

le UDP-N-Glu-γ. Lorsqu'on compare les valeurs de RMSD entre les différentes familles pour ce domaine, on remarque un regroupement des MurE et MurF qui se différencient des autres (Fig.29). MurE et MurF sont d'ailleurs majoritairement mélangées entre elles dans les arbres phylogénétiques obtenus pour le domaine 1. Ce regroupement est probablement le résultat de changement de d'orientation entre MurC/D (repliement ouvert) et MurE/F (repliement fermé) pour permettre la liaison du substrat au niveau du domaine 1 (Smith, 2006). Les comparaisons inter-familles par paires de protéines pour Murα, Murβ et Murγ ont des RMSD faibles pour le domaine 1 (inférieur à 2 Å), ce qui signifie que le domaine 1 de ces protéines est très similaire. Les clans Murα, Murβ sont retrouvés dans dans la majorité des arbres construits en Jackknife (Fident : 97%, LDDT : 88%, MT : 84%, RMSD : 0%) et Murα, Murβ, Murγ également (Fident : 70%, LDDT : 67%, MT : 60%, RMSD : 66%) (Fig.28). Ces trois protéines archéennes, ainsi que Murδ, sont également proches de MurC et MurD qui comportent des valeurs de RMSD inférieures à 3 Å. Cependant, fait intéressant, la médiane de Murδ a une valeur de RMSD de 11.059 Å quand on le compare avec les autres protéines archéennes (Fig.29). De plus le domaine 1 de Murδ semble plus proche de celui de toutes les familles de bactéries que des domaines 1 des archées (Fig.28). Suite à cette observation, Murδ interviendrait dans l'étape où le pentapeptide serait ajouté sur le disacharide. Par conséquent, le L-Ala serait ajouté deux fois par la même enzyme. Le domaine 1 de Murδ reconnaitrait ainsi un sucre tout comme les domaines 1 des protéines bactériennes, ce qui le différencierait des domaines 1 des autres Mur archéennes, qui, elles, reconnaitraient un acide aminé (en excluant murδ) (Fig.29). La protéine archéenne Murδ a une mediane inférieure à 3 Å avec la famille MurD (2.8035 Å RMSD), ce qui contredit Subedi et al. (2022) pour lequel Murδ ne s'aligne pas avec MurD (Subedi et al., 2022).

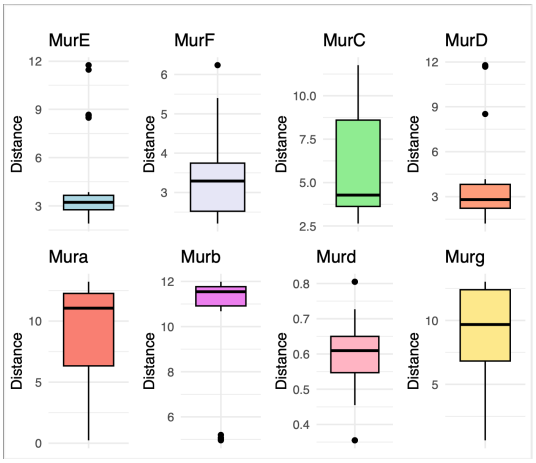

FIGURE 28 – Boxplots du domaine 1 de la famille archéenne Murδ avec l'ensemble des autres familles de protéines. Les médianes des boxplots avec les familles Murα, Murβ et Murγ (familles archéennes) présentent des valeurs de RMSD largement supérieures à la limite de 5.

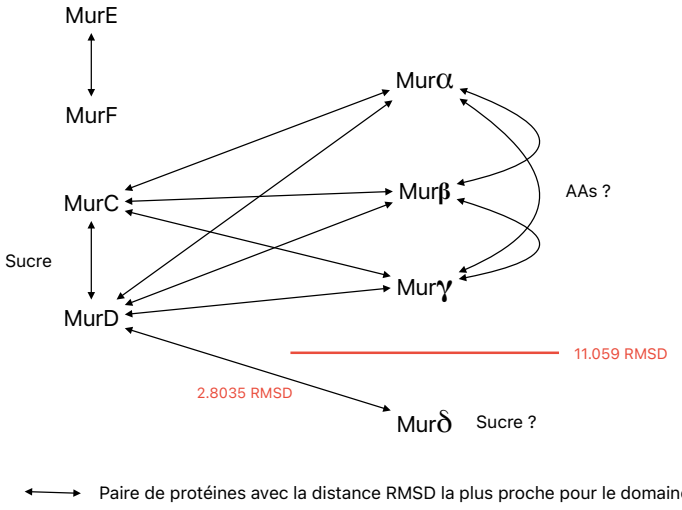

FIGURE 29 – Schéma des relations entre le domaine 1 des différentes familles de protéines.

8.2.2.3 Domaine 3 En ce qui concerne le domaine 3, qui reconnaitrait les acides aminés à ajouter à la chaîne, on remarque que celui de la protéine MurE est légèrement plus similaire (RMSD inférieur à 3 Å) aux domaines 3 archéens de Murδ, Murα et Murβ (Murβ étant également proche de MurC). Le domaine 3 de MurE est d'ailleurs légèrement plus similaire aux domaines 3 archéen (2.2415 Å) que des domaines 3 bactériens (3.317 Å). Les domaines 3 des protéines MurF, MurD et Murγ sont chacun séparées de l'ensemble des autres protéines. Les protéines Murα, Murβ et MurD ont des structures similaires au niveau du domaine 3 (Subedi et al., 2021). Une explication plausible réside dans la conformation des acides aminés reconnus et l'acide aminé lui-même. En effet, MurF et MurD reconnaissent des acides aminés en conformation D (D-ala et D-Glu) alors que l'ensemble des domaines 3 des autres protéines reconnaissent des acides aminés en conformation L (L-Ala, L-Lys et L-Glu). Murγ reconnait l'acide aminé L-Glu, or la glutamine n'est reconnue que par Murγ et par MurD. Cependant, MurD le reconnait en conformation D et pas L (Fig.30).

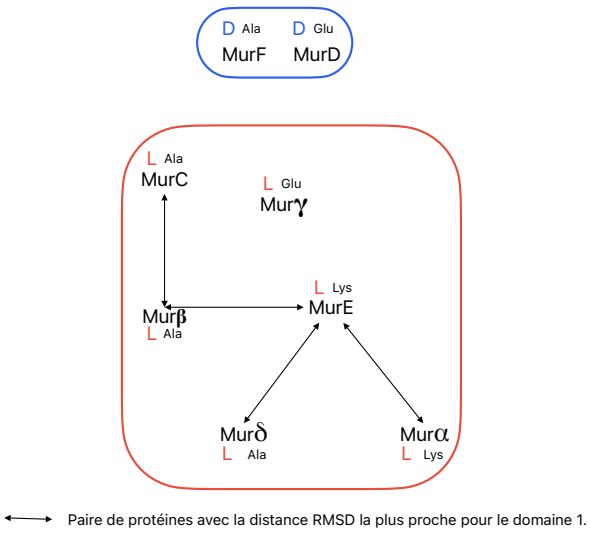

FIGURE 30 – Schéma des relations du domaine 3 des différentes familles de protéines

8.2.3 Voie de biosynthèse des peptides de la PM

L'hypothèse de la biosynthèse des Mur ligases archéennes présentée pour Lupo et al. (2022) (cfr introduction) est donc modifiée au vu de nos résultats. Le domaine 1 de Murβ reconnaitrait le L-Glu, le domaine 2 de l'ensemble des Mur ligases permettrait l'utilisation de l'ATP pour permettre la liaison de la L-Ala par le domaine 3 à la chaîne d'acides aminés en devenir. Le domaine 1 de Murα reconnaitrait le L-Ala activé sur la chaîne d'acides aminés, et le domaine 3 reconnaitrait le L-Lys pour le lier à la chaîne d'acides aminés. Murβ ajouterait un nouveau L-Ala à la chaîne. Le L-Glu serait ajouté à la chaîne d'acides aminés grâce au domaine 3 de Murγ qui aurait au préalable reconnu le L-Ala par son domaine 1. Enfin, le domaine 1 de Murδ reconnaitrait le sucre (NAT). Le domaine 3 reconnaitrait la chaîne d'acides aminés pour le L-Glu et la lierait au sucre.

9 Perspectives

Au vu de l'incongruence entre les phylogénies de domaines/protéines complètes, ainsi que la conservation de ces structures, notamment pour Murδ, on peut exclure l'hypothèse d'un ancêtre commun à toutes les Mur ligases. On peut se demander si les Mur ligases actuelles sont issues de recombinaisons entre plusieurs ancêtres en plus de duplications de gènes (chez les archées), ou si on assiste à des évènements de convergence évolutive localisés au niveau de certains motifs essentiels au bon fonctionnement des protéines. L'analyse approfondie de la conservation des différents domaines permettrait de déterminer si on a affaire à de la convergence évolutive ou à de la conservation plus ciblée des régions structurelles des protéines. En effet, les deux hypothèses donneraient des prédictions différentes, la recombinaison donnerait lieu à de grands morceaux de séquences similaires entre les familles de Mur ligases alors que la convergence évolutive donnerait la conservation d'acides aminés particuliers le long de la séquence (Fig.31) La simulation des liaisons entre les domaines et différents ligands (sucres ou acides aminés) permettrait de répondre à la question de la

différenciation de Murδ (reconnaissance de sucre) et de MurD, MurF et Murγ (conformation des acides aminés reconnus). Une étude plus approfonndie sur les zones conservées grâce au programme Consurf permettrait de mieux comprendre la répartition des morceaux conservés dans les différents domaines.

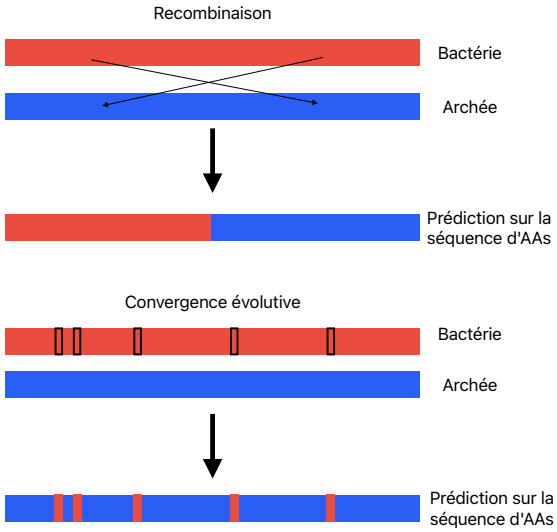

FIGURE 31 – A. Représentation schématique de la prédiction d'un évènements de recombinaison. B. Représentation schématique de la prédiction d'un évènement de convergence évolutive.

				Domaine 1				Domaine 3			
	Famille	AA	L/D	Distance intra famille	Distance inter famille		Valeurs inférieures à 3 de RMSD	Distance intra famille	Distance inter famille		Valeurs inférieures à 3 de RMSD
					Archées	Bactéries			Archées	Bactéries	
UDP-MurNAc	MurE	LYS	L	1.373	3.75525	4.158	MurF (1.3215)	0.557	2.2415	3.137	Mura (2.0625) Murb (1.824) Murd (2.4205)
	MurF	ALA	D	0.941	6.76275	3.5925	MurE (1.3215)	1.442	3.761	3.088	/
	MurC	ALA	L	0.898	2.3335	3.5925	Mura (1.944) Murb (2.448) Murg (2.219) MurD (1.7775)	1.136	3.3185	3.137	Murb (2.997)
	MurD	GLU	D	1.233	1.79425	4.158	Mura (1.598) Murb (1.989) Murd (2.8035) Murg (1.5995) MurC (1.7775)	0.834	4.062	4.5395	/
Acide animé activé	Mura	LYS	L	1.379	1.565	2.51825	Murb (1.565) Murg (1.28) MurC (1.944) MurD (1.598)	1.4105	3.486	3.66575	MurE (2.0625)
	Murb	ALA	L	0.786	1.565	5.98975	Mura (1.565) Murg (1.378) MurC (2.449) MurD (1.989)	1.082	3.486	3.33425	MurC (2.997) MurE (1.824)
	Murd	ALA	L	0.6095	11.059	3.25525	MurD (2.8035)	0.4815	3.267	3.24575	MurE (2.4205)
	Murg	GLU	L	1.03	1.378	3.25475	Mura (1.28) Murb (1.378) MurC (2.219) MurD (1.5995)	1.0125	3.674	3.984	/

	Famille	Domaine2		
		Distance intra famille	Distance inter famille	
			Archées	Bactéries
UDP-MurNAc	MurE	0.717	1.45025	1.382
	MurF	0.971	1.71925	2.17
	MurC	0.994	1.4205	1.7985
	MurD	0.841	1.69925	1.7985
Acide animé activé	Mura	0.9815	1.409	1.525
	Murb	0.9635	1.462	1.42725
	Murd	0.6925	1.462	1.62875
	Murg	0.5575	1.631	1.4995

FIGURE 32 – Table récapitulative des médianes des boxplots des valeurs de RMSD avec les valeurs extrêmes (de plus de 3 RMSD) associés.

10 Annexes

10.1 Images supplémentaires

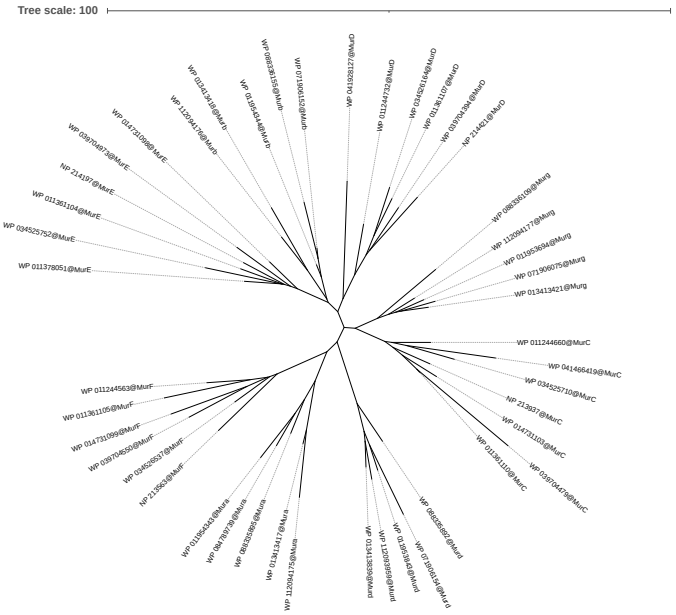

FIGURE 33 – Arbre SDM domaine 2

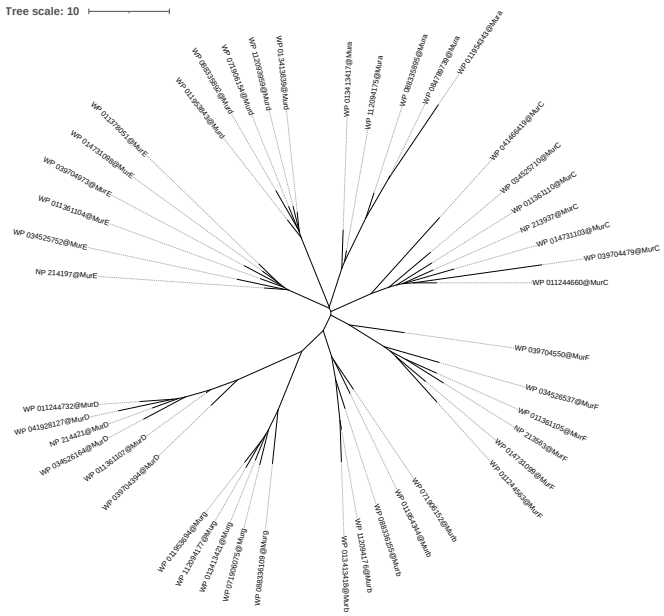

FIGURE 34 – Arbre SDM domaine 3

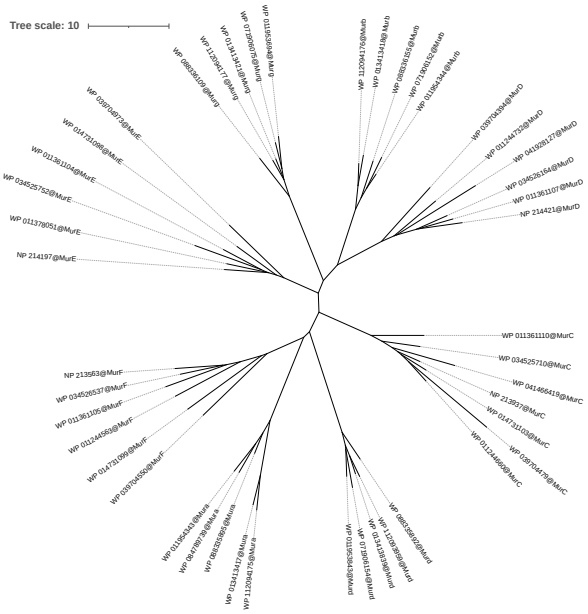

FIGURE 35 – Arbre SDM combinaison domaines 1 et 2

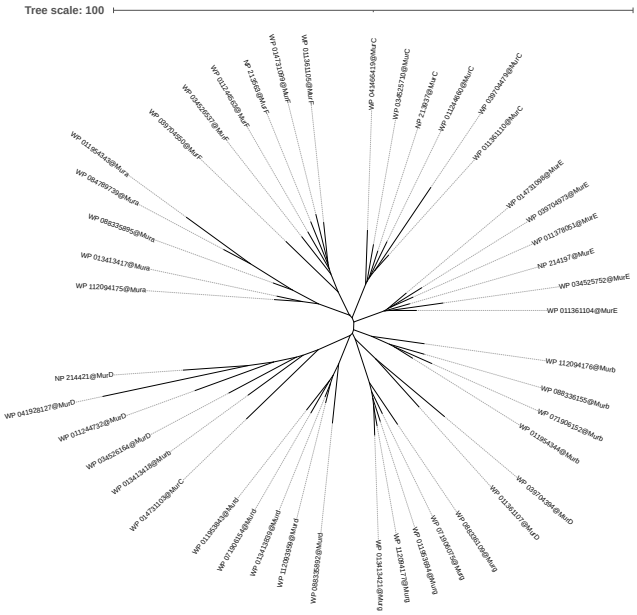

FIGURE 36 – Arbre SDM combinaison domaines 2 et 3

10.2 Programmes

10.2.1 (2) hmms-parser-coord.pl

```
#!/usr/bin/env perl
# PODNAME:
# CONTRIBUTOR:

use Modern::Perl '2011';
use autodie;
```

```perl
#use Getopt::Euclid qw(:vars);
use Smart::Comments '###';

use Tie::IxHash;

use Bio::FastParsers;
use aliased 'Bio::FastParsers::Hmmer::DomTable';
use Bio::MUST::Core;
use Bio::MUST::Core::Utils qw(secure_outfile);
use aliased 'Bio::MUST::Core::Ali';
use aliased 'Bio::MUST::Core::Seq';

my $infile = shift;
my $hmms   = shift;

my $ali = Ali- load($infile);
my $report = DomTable->new( file => $hmms);

my $ali_dom1 = Ali->new();
my $ali_dom2 = Ali->new();
my $ali_dom3 = Ali->new();

my $previous_hit = q{};
my $previous_len;

my @coordonnees_dom1;
my @coordonnees_dom2;
my @coordonnees_dom3;

HIT:
while (my $hit = $report->next_hit) {

    # extract full id and length of hit's sequence
    # ... and coordinates of central domain
    my ($full_id, $s_len, $d_start, $d_end) = ($hit->target_name, $hit->tlen, $hit->ali_from, $hit->ali_to);
    my $dom_len = $d_end - $d_start;

    # skip hit if...
    next HIT if $dom_len < 100;
#    next HIT if $d_start < 80;
    next HIT if $d_start < 70;
    next HIT if $full_id eq $previous_hit && $dom_len < $previous_len;

    $previous_hit = $full_id;
    $previous_len = $dom_len;

    # get current sequence from ALI
    $full_id =~ s/_/ /;
    my $seq = $ali->get_seq_with_id($full_id);

    # extract domain 1
    my @dom1_coord = [ 1, $d_start - 1];
    my $dom1 = $seq->spliced_seq( \@dom1_coord );
    $ali_dom1->add_seq($dom1);
    push @coordonnees_dom1, @dom1_coord;

    # extract domains 2
    my @dom2_coord = [ $d_start, $d_end ];
```

```
    my $dom2 = $seq->spliced_seq( \@dom2_coord );
    $ali_dom2->add_seq($dom2);
    push @coordonnees_dom2, @dom2_coord;

    # extract domain 3
    my @dom3_coord = [ $d_end + 1, $s_len ];
    $dom3_coord[1] = $s_len; #NEW
    my $dom3 = $seq->spliced_seq( \@dom3_coord );
    $ali_dom3->add_seq($dom3);
    push @coordonnees_dom3, @dom3_coord;

}

#$ali_dom1->store_fasta(   secure_outfile($infile, '-dom1'   ) );
#$ali_dom2->store_fasta(   secure_outfile($infile, '-dom2'   ) );
#$ali_dom3->store_fasta(   secure_outfile($infile, '-dom3'   ) );

my $outfile1 = 'output_dom1_coord.txt';
open (my $out1, '>', $outfile1);

for my $coord_ref1 (@coordonnees_dom1) {
    my ($one, $end) = @{$coord_ref1};
    print $out1 "$full_id; $one; $end \n";
}

close $out1;

my $outfile2 = 'output_dom2_coord.txt';
open (my $out2, '>', $outfile2);

for my $coord_ref2 (@coordonnees_dom2) {
    my ($end, $len) = @{$coord_ref2};
    print $out2 "$full_id; $end; $len \n";
}

close $out2;

my $outfile3 = 'output_dom3_coord.txt';
open (my $out3, '>', $outfile3);

for my $coord_ref3 (@coordonnees_dom3) {
    my ($start, $len) = @{$coord_ref3};
    print $out3 "$full_id; $start; $end \n";
}

close $out3;
```

10.2.2 (3) pymol-script.pl

```
#!/usr/bin/env perl

#avoid boilerplate
use Modern::Perl '2011';
use Smart::Comments;

use strict;
use warnings;
```

```
use feature 'say';

unless (@ARGV == 3) {
    die <<"EOT";
Usage= $0 <infile.txt> <outfile2.txt> <outfile.pml>
This tool receives a TXT file containing the coordiantes of protein domain 1 and
an other TXT file containing the name of the proteins.
It creates a pymol script using domain 1 coordinates that calculates the
structural distance.
Exemple: $0 outfile_dom1.txt PDB_file.txt pymol_script1.pml
EOT
}

# Fichier d'entrée pour les coordonnées
my $infile1 = shift;
my $infile2 = shift;
my $outfile1 = shift;


# Tableau pour stocker les lignes lues
my @lines;
my @lines_proteines;
# Ouvrir le fichier en lecture
open my $in, '<', $infile1;



# Lire et stocker les lignes dans @lines
while (my $line = <$in>) {
    chomp $line;
    push @lines, $line;
}
### @lines

# Fermer le fichier après lecture
close $in;
open my $in2, '<', $infile2;

while (my $line = <$in2>) {
    chomp $line;
    push @lines_proteines, $line;
}
### @lines_proteines
close $in2;
###test: $#lines
open my $out, '>', $outfile1;
say {$out} "# Script Pymol";
say {$out} "# scriptpymol.pml";
say {$out} "#-----------------------------------------\n";
my $i=0;
# Parcourir chaque ligne pour les calculs
for my $prot1 (@lines_proteines) {
    my ($dom1_coord1, $dom1_coord2) = split "; ", $lines[$i];
#    my $prot1 = $lines_proteines[$i];

    #$prot1 =~ s/\.pdb$//;
    ### $prot1

    say {$out} "load '/Users/coralie/Desktop/Memoire/PDB_files/$prot1.pdb'";
```

```
    say {$out} "load '/Users/coralie/Desktop/Memoire/PDB_files/$prot1.pdb', SameProt";
    say {$out} "\nsuper /$prot1//A/$dom1_coord1-$dom1_coord2, /SameProt//A/$dom1_coord1-$dom1_coord2" ;

    say {$out} "delete SameProt";

    # Parcourir les lignes une deuxième fois pour le calcul
    for my $j ($i+1 .. $#lines) {
       # next if $line eq $line; # Ignorer la même ligne

        my ($dom1_coord3, $dom1_coord4) = split "; ", $lines[$j];
        my $prot2 = $lines_proteines[$j];

        say {$out} "load '/Users/coralie/Desktop/Memoire/PDB_files/$prot2.pdb'";
        $prot2 =~ s/\.pdb$//;

        say {$out} "\nsuper /$prot1//A/$dom1_coord1-$dom1_coord2, /$prot2//A/$dom1_coord3-$dom1_coord4" ;

        say {$out} "delete $prot2\n";

    }

    say {$out} "delete $prot1";
    $i++;
}

# Afficher les résultats
say {$out} "quit()";
close $out;
```

10.2.3 (5) pymol-script.pml

```
# Script Pymol
# scriptpymol.pml
#-----------------------------------------

load '/Users/coralie/Desktop/Memoire/PDB_files/WP_011244563.pdb'

sel /WP_011244563//A/117-317

save DOM2_WP_011244563.pdb, sele

delete sele

delete WP_011244563
load '/Users/coralie/Desktop/Memoire/PDB_files/WP_034526537.pdb'

sel /WP_034526537//A/117-317

save DOM2_WP_034526537.pdb, sele

delete sele

delete WP_034526537
load '/Users/coralie/Desktop/Memoire/PDB_files/WP_014731099.pdb'

sel /WP_014731099//A/117-317
```

```
save DOM2_WP_014731099.pdb, sele

delete sele

delete WP_014731099

...

load '/Users/coralie/Desktop/Memoire/PDB_files/NP_214421.pdb'

sel /NP_214421//A/117-317

save DOM2_NP_214421.pdb, sele

delete sele

delete NP_214421
load '/Users/coralie/Desktop/Memoire/PDB_files/WP_011361107.pdb'

sel /WP_011361107//A/117-317

save DOM2_WP_011361107.pdb, sele

delete sele

delete WP_011361107
load '/Users/coralie/Desktop/Memoire/PDB_files/WP_011244732.pdb'

sel /WP_011244732//A/117-317

save DOM2_WP_011244732.pdb, sele

delete sele

delete WP_011244732
load '/Users/coralie/Desktop/Memoire/PDB_files/WP_039704394.pdb'

sel /WP_039704394//A/117-317

save DOM2_WP_039704394.pdb, sele

delete sele

delete WP_039704394
quit()
```

10.2.4 (7) analyses-RMSD.R

```r
library(ggplot2)
library(gridExtra)
library(reshape2)

# Data loading
rmsd <- as.numeric(readLines("Global/RMSDGlobal.txt"))
|_prot <- readLines("PDB_files_Ordre.txt") #Proteins ID

# Changing names to include the type of ligase wall
```

```
new_list_prot <-readLines("PDB_files.txt") #List with ID and wall type
names_mapping <- data.frame(anciens_noms = |_prot, nouveaux_noms = new_list_prot)

for (i in 1:length(|_prot)) { #remplace the names
  |_prot[|_prot == names_mapping$anciens_noms[i]] <- names_mapping$nouveaux_noms[i]
}

# Populating the matrix.
prot1 <- c() # Creating vectors to store the names
prot2 <- c()

# For loop to create protein pairs
for (i in 1:(length(|_prot))) {
  for (j in (i):length(|_prot)) {
    prot1 <- c(prot1, |_prot[i])
    prot2 <- c(prot2, |_prot[j])
  }
}

df <- data.frame( # Creating a data frame that will associate names with distances
  Proteine1 = prot1,
  Proteine2 = prot2,
  Distance = rmsd #To make the boxplots
)

dfbis <- data.frame(
  Proteine1 = prot2,
  Proteine2 = prot1,
  Distance = rmsd #To make the boxplots
)

dfcomp <- rbind(df, dfbis)
dfcomp$Distance <- as.numeric(dfcomp$Distance)
matrice <- xtabs(Distance ~ Proteine2 + Proteine1, data = dfcomp) #Creating a matrix based on the dataframe

################## histogramme inter- intra- familles

# Définir la fonction calculer_moyenne_double_identifiant avec les points (.)
calculer_moyenne_double_identifiant <- function(matrice, identifiant1, identifiant2) {
  lignes_identifiant1 <- grep(identifiant1, rownames(matrice))
  colonnes_identifiant2 <- grep(identifiant2, colnames(matrice))

  valeurs <- matrice[lignes_identifiant1, colonnes_identifiant2]
  valeurs <- valeurs[valeurs != 0]

  if (length(valeurs) > 0) {
    # Créer un data frame avec les identifiants et les valeurs
    resultats_temp <- data.frame(identifiant1 = identifiant1, identifiant2 = identifiant2, valeurs = valeurs)
    return(resultats_temp)
  } else {
    return(NULL)
  }
}

# Définir les identifiants
identifiants <- c('MurE', 'Murd', 'MurC', 'MurF', 'Mura', 'Murg', 'MurD', 'Murb')

# Initialiser le data frame des résultats
resultats <- data.frame(identifiant1 = character(), identifiant2 = character(), valeurs = numeric(), \
```

```
  stringsAsFactors = FALSE)

# Calculer les moyennes pour chaque paire d'identifiants
for (id1 in identifiants) {
  for (id2 in identifiants) {
    resultats_temp <- calculer_moyenne_double_identifiant(matrice, paste('.', id1, sep = ''), \
      paste('.', id2, sep = ''))
    if (!is.null(resultats_temp)) {
      resultats <- rbind(resultats, resultats_temp)
    }
  }
}

# Afficher les premiers résultats
print(head(resultats))

# Créer une nouvelle colonne pour différencier les comparaisons intra et inter
resultats$comparaison <- ifelse(resultats$identifiant1 == resultats$identifiant2, 'Intra', 'Inter')

# Créer l'histogramme avec deux couleurs
p <- ggplot(resultats, aes(x = valeurs, fill = comparaison)) +
  geom_histogram(binwidth = 0.5, position = 'dodge') +
  labs(title = 'Histogramme des valeurs RMSD \n inter- et intra- famille', x = 'RMSD', y = 'Fréquence') +
  scale_fill_manual(values = c('Intra' = 'violetred', 'Inter' = 'cornflowerblue')) +
  theme_minimal()

# Afficher l'histogramme
print(p)

################## histogramme intra famille

# Filtrer les résultats pour ne garder que les comparaisons intra-familles
resultats_intra <- subset(resultats, comparaison == 'Intra')
color <- c("")
# Créer une | de graphiques pour chaque famille
plots <- list()

for (famille in unique(resultats_intra$identifiant1)) {
  p <- ggplot(subset(resultats_intra, identifiant1 == famille), aes(x = valeurs)) +
    geom_histogram(binwidth = 0.5, fill = 'cornflowerblue', color = 'black') +
    labs(title = paste(famille), x = 'RMSD', y = 'Fréquence') +
    theme_minimal()
  plots[[famille]] <- p
}

# Afficher les graphiques
library(gridExtra)
do.call(grid.arrange, c(plots, ncol = 2))
########################################################## CREATING THE BOXPLOTS
# Select the mur ligase to represent in boxplots
dfcomp_filtered_a_F <- dfcomp[grepl("[0-9A-Za-z]*@Mura[0-9A-Za-z]*", dfcomp$Proteine1) & \
 grepl("[0-9A-Za-z]*Murg[0-9A-Za-z]*", dfcomp$Proteine2) & dfcomp$Proteine1 != dfcomp$Proteine2, ]
dfcomp_filtered_b_F <- dfcomp[grepl("[0-9A-Za-z]*@Murb[0-9A-Za-z]*", dfcomp$Proteine1) & \
grepl("[0-9A-Za-z]*Murg[0-9A-Za-z]*", dfcomp$Proteine2) & dfcomp$Proteine1 != dfcomp$Proteine2, ]
dfcomp_filtered_g_F <- dfcomp[grepl("[0-9A-Za-z]*@Murg[0-9A-Za-z]*", dfcomp$Proteine1) & \
grepl("[0-9A-Za-z]*Murg[0-9A-Za-z]*", dfcomp$Proteine2) & dfcomp$Proteine1 != dfcomp$Proteine2, ]
dfcomp_filtered_d_F <- dfcomp[grepl("[0-9A-Za-z]*@Murd[0-9A-Za-z]*", dfcomp$Proteine1) & \
grepl("[0-9A-Za-z]*Murg[0-9A-Za-z]*", dfcomp$Proteine2) & dfcomp$Proteine1 != dfcomp$Proteine2, ]
dfcomp_filtered_E_F <- dfcomp[grepl("[0-9A-Za-z]*@MurE[0-9A-Za-z]*", dfcomp$Proteine1) & \
```

```
grepl("[0-9A-Za-z]*Murg[0-9A-Za-z]*", dfcomp$Proteine2) & dfcomp$Proteine1 != dfcomp$Proteine2, ]
dfcomp_filtered_F_F <- dfcomp[grepl("[0-9A-Za-z]*@MurF[0-9A-Za-z]*", dfcomp$Proteine1) & \
grepl("[0-9A-Za-z]*Murg[0-9A-Za-z]*", dfcomp$Proteine2) & dfcomp$Proteine1 != dfcomp$Proteine2, ]
dfcomp_filtered_C_F <- dfcomp[grepl("[0-9A-Za-z]*@MurC[0-9A-Za-z]*", dfcomp$Proteine1) & \
grepl("[0-9A-Za-z]*Murg[0-9A-Za-z]*", dfcomp$Proteine2) & dfcomp$Proteine1 != dfcomp$Proteine2, ]
dfcomp_filtered_D_F <- dfcomp[grepl("[0-9A-Za-z]*@MurD[0-9A-Za-z]*", dfcomp$Proteine1) & \
grepl("[0-9A-Za-z]*Murg[0-9A-Za-z]*", dfcomp$Proteine2) & dfcomp$Proteine1 != dfcomp$Proteine2, ]

########################################################## CREATING THE BOXPLOTS
#library(ggplot2)
#library(gridExtra)

# Define custom color palette
my_colors <- c("lightblue", "lavender", "lightgreen", "lightsalmon", "salmon", "violet", "pink1", \
 "lightgoldenrod1")

# Create individual boxplots with custom colors and aesthetics
plot1 <- ggplot(dfcomp_filtered_E_F, aes(x = "E", y = Distance)) +
  geom_boxplot(fill = my_colors[1], color = "black") +
  labs(title = "MurE", y = "Distance") +
  theme_minimal() +
  #scale_y_continuous(limits = c(0, 6)) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank(), axis.ticks.x = element_blank())

plot2 <- ggplot(dfcomp_filtered_F_F, aes(x = "F", y = Distance)) +
  geom_boxplot(fill = my_colors[2], color = "black") +
  labs(title = "MurF", y = "Distance") +
  theme_minimal() +
  #scale_y_continuous(limits = c(0, 6)) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank(), axis.ticks.x = element_blank())

# Combine the remaining plots with similar adjustments
plot3 <- ggplot(dfcomp_filtered_C_F, aes(x = "C", y = Distance)) +
  geom_boxplot(fill = my_colors[3], color = "black") +
  labs(title = "MurC", y = "Distance") +
  theme_minimal() +
  #scale_y_continuous(limits = c(0, 6)) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank(), axis.ticks.x = element_blank())

plot4 <- ggplot(dfcomp_filtered_D_F, aes(x = "D", y = Distance)) +
  geom_boxplot(fill = my_colors[4], color = "black") +
  labs(title = "MurD", y = "Distance") +
  theme_minimal() +
  #scale_y_continuous(limits = c(0, 6)) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank(), axis.ticks.x = element_blank())

plot5 <- ggplot(dfcomp_filtered_a_F, aes(x = "a", y = Distance)) +
  geom_boxplot(fill = my_colors[5], color = "black") +
  labs(title = "Mura", y = "Distance") +
  theme_minimal() +
  #scale_y_continuous(limits = c(0, 6)) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank(), axis.ticks.x = element_blank())

plot6 <- ggplot(dfcomp_filtered_b_F, aes(x = "b", y = Distance)) +
  geom_boxplot(fill = my_colors[6], color = "black") +
  labs(title = "Murb", y = "Distance") +
  theme_minimal() +
  # scale_y_continuous(limits = c(0, 6)) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank(), axis.ticks.x = element_blank())
```

```
plot7 <- ggplot(dfcomp_filtered_d_F, aes(x = "d", y = Distance)) +
  geom_boxplot(fill = my_colors[7], color = "black") +
  labs(title = "Murd", y = "Distance") +
  theme_minimal() +
  # scale_y_continuous(limits = c(0, 6)) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank(), axis.ticks.x = element_blank())

plot8 <- ggplot(dfcomp_filtered_g_F, aes(x = "g", y = Distance)) +
  geom_boxplot(fill = my_colors[8], color = "black") +
  labs(title = "Murg", y = "Distance") +
  theme_minimal() +
  #scale_y_continuous(limits = c(0, 6)) +
  theme(axis.title.x = element_blank(), axis.text.x = element_blank(), axis.ticks.x = element_blank())

combined_plots <- grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, ncol = 4, \
  top = "Murg and the other families")

####### Heatmap
# Créer la heatmap avec une nouvelle palette de couleurs (bleu au jaune)
# Plot the heatmap
GlobalCA_matrice <- melt(Global_matrice_base)
colnames(GlobalCA_matrice)[1] <- "Var1"
colnames(GlobalCA_matrice)[2] <- "Var2"

ggplot(data = GlobalCA_matrice, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) +
  scale_fill_gradient(low = "white", high = "violetred")  # Changez "blue" et "red" par les
      couleurs que vous préférez
```

10.2.5 (8) analyses-SDM.R

```
library(phangorn)

# Lecture des donnees
rmsd <- as.numeric(readLines("RMSD12.txt"))
nbrCA <- as.numeric(readLines("DOM12CA_Aligned.txt"))
nbrSmall <- as.numeric(readLines("DOM12_Small.txt"))
|_prot <- readLines("PDB_files_Ordre.txt") #ID des proteines

#Creation des vecteurs vides (srms, pfte, w1, w2, sdm)
srms <- numeric(length(rmsd))
pfte <- numeric(length(rmsd))
w1 <- numeric(length(rmsd))
w2 <- numeric(length(rmsd))
sdm <-numeric(length(rmsd))
test <-numeric(length(rmsd))

# calcul des differentes valeurs (srms, pfte, w1, w2, sdm)
for (i in 1:length(rmsd)) {
  srms[i] <- 1 - (rmsd[i]/13.415)
}

for (i in 1:length(rmsd)) {
  pfte[i] <- (nbrCA[i]/nbrSmall[i])
}
```

```r
for (i in 1:length(rmsd)) {
  w1[i] <- ((1 - pfte[i])+(1 - srms[i]))/2
  w2[i] <- (srms[i] + pfte[i])/2
  test[i] <- w1[i] + w2[i]
}

etap1 <-numeric(length(rmsd))
etap2 <-numeric(length(rmsd))
etap3 <-numeric(length(rmsd))
for (i in 1:length(rmsd)) {
      etap1[i] <- (w1[i] * pfte[i])
      etap2[i] <- (w2[i] * srms[i])
      etap3[i] <- log(etap1[i]+etap2[i]) #ln car article de base

      sdm[i] <- -100 * etap3[i]
}

############################################################
#Creation de la matrice

#Changer de noms pour avoir le type de Mur ligase
new_list_prot <-readLines("PDB_files.txt") #| avec ID et le type de mur
names_mapping <- data.frame(anciens_noms = |_prot, nouveaux_noms = new_list_prot)

for (i in 1:length(|_prot)) { #remplacer les noms
  |_prot[|_prot == names_mapping$anciens_noms[i]] <- names_mapping$nouveaux_noms[i]
}

#Remplir la matrice

prot1 <- c() #creation des vecteurs pour stocker les noms
prot2 <- c()

# Boucle for pour créer les paires de protéines
for (i in 1:(length(|_prot))) { #for (i in 1:(length(|_prot)-1)) {
  for (j in (i):length(|_prot)) { # for (j in (i +1 ):length(|_prot)) {
    prot1 <- c(prot1, |_prot[i])
    prot2 <- c(prot2, |_prot[j])
  }
}

df <- data.frame( #creation d un data frame qui va associer les noms avec les distances
  Proteine1 = prot1,
  Proteine2 = prot2,
  Distance = sdm
)

dfbis <- data.frame(
  Proteine1 = prot2,
  Proteine2 = prot1,
  Distance = sdm
)

dfcomp <- rbind(df, dfbis)
dfcomp$Distance <- as.numeric(dfcomp$Distance)
matrice <- xtabs(Distance ~ Proteine2 + Proteine1, data = dfcomp) #creation d une matrice sur base du dataframe

#Sorties (arbre NJ, CSV et Newick)
```

```
library(ape)

arbre_nj_SDM <- nj(matrice) #Creation d un arbre sous NJ
plot(arbre_nj_SDM, "phylogram", main="DOM12CA_final_arbre", cex = 0.4)# visualisation de l'arbre
dev.off()
write.csv(matrice, file = "DOM12CA_Cutoff_matrice.csv", row.names = TRUE)#format CSV

write.tree(arbre_nj_SDM, file = "DOM12CA_final_arbre.txt")
```

10.2.6 (9) jack-dist.R

```
library(phangorn)

# Lecture des donnees
rmsd <- as.numeric(readLines("RMSD1.txt"))
nbrCA <- as.numeric(readLines("DOM1CA_Aligned.txt"))
nbrSmall <- as.numeric(readLines("DOM1_Small.txt"))
|_prot <- readLines("PDB_files_Ordre.txt") #ID des proteines

#Creation des vecteurs vides (srms, pfte, w1, w2, sdm)
srms <- numeric(length(rmsd))
pfte <- numeric(length(rmsd))
w1 <- numeric(length(rmsd))
w2 <- numeric(length(rmsd))
sdm <-numeric(length(rmsd))
test <-numeric(length(rmsd))

# calcul des differentes valeurs (srms, pfte, w1, w2, sdm)
for (i in 1:length(rmsd)) {
  srms[i] <- 1 - (rmsd[i]/15.387)
}

for (i in 1:length(rmsd)) {
  pfte[i] <- (nbrCA[i]/nbrSmall[i])
}

for (i in 1:length(rmsd)) {
  w1[i] <- ((1 - pfte[i])+(1 - srms[i]))/2
  w2[i] <- (srms[i] + pfte[i])/2
  test[i] <- w1[i] + w2[i]
}

etap1 <-numeric(length(rmsd))
etap2 <-numeric(length(rmsd))
etap3 <-numeric(length(rmsd))
for (i in 1:length(rmsd)) {
  etap1[i] <- (w1[i] * pfte[i])
  etap2[i] <- (w2[i] * srms[i])
  etap3[i] <- log(etap1[i]+etap2[i]) #ln car article de base

  sdm[i] <- -100 * etap3[i]
}

############################################################
#Creation de la matrice

#Changer de noms pour avoir le type de Mur ligase
```

```
new_list_prot <-readLines("PDB_files.txt") #| avec ID et le type de mur
names_mapping <- data.frame(anciens_noms = |_prot, nouveaux_noms = new_list_prot)

for (i in 1:length(|_prot)) { #remplacer les noms
  |_prot[|_prot == names_mapping$anciens_noms[i]] <- names_mapping$nouveaux_noms[i]
}

#Remplir la matrice

prot1 <- c() #creation des vecteurs pour stocker les noms
prot2 <- c()

# Boucle for pour créer les paires de protéines
for (i in 1:(length(|_prot))) { #for (i in 1:(length(|_prot)-1)) {
  for (j in (i):length(|_prot)) { # for (j in (i +1 ):length(|_prot)) {
    prot1 <- c(prot1, |_prot[i])
    prot2 <- c(prot2, |_prot[j])
  }
}

df <- data.frame( #creation d un data frame qui va associer les noms avec les distances
  Proteine1 = prot1,
  Proteine2 = prot2,
  Distance = sdm
)

dfbis <- data.frame(
  Proteine1 = prot2,
  Proteine2 = prot1,
  Distance = sdm
)

dfcomp <- rbind(df, dfbis)
dfcomp$Distance <- as.numeric(dfcomp$Distance)
matrice <- xtabs(Distance ~ Proteine2 + Proteine1, data = dfcomp) #creation d une matrice sur base du dataframe

#Sorties (arbre NJ, CSV et Newick)
library(ape)

arbre_nj_SDM <- nj(matrice_sans_identifiant) #Creation d un arbre sous NJ
plot(arbre_nj_SDM, "phylogram", main="DOM23CA_NJ_Cutoff2", cex = 0.4)# visualisation de l'arbre
dev.off()
write.csv(matrice, file = "DOM23_matrice.csv", row.names = TRUE)#format CSV

write.tree(arbre_nj_SDM, file = "DOM23CA_final_arbre3.txt")

#######################################

# | des identifiants
identifiants <- c(
  "NP_213563@MurF", "NP_213937@MurC", "NP_214197@MurE", "NP_214421@MurD",
  "WP_011244563@MurF", "WP_011244660@MurC", "WP_011244732@MurD", "WP_011361104@MurE",
  "WP_011361105@MurF", "WP_011361107@MurD", "WP_011361110@MurC", "WP_011378051@MurE",
  "WP_011953694@Murg", "WP_011953843@Murd", "WP_011954343@Mura", "WP_011954344@Murb",
  "WP_013413417@Mura", "WP_013413418@Murb", "WP_013413421@Murg", "WP_013413839@Murd",
  "WP_014731098@MurE", "WP_014731099@MurF", "WP_014731103@MurC", "WP_034525710@MurC",
  "WP_034525752@MurE", "WP_034526164@MurD", "WP_034526537@MurF", "WP_039704394@MurD",
  "WP_039704479@MurC", "WP_039704550@MurF", "WP_039704973@MurE", "WP_041466419@MurC",
  "WP_041928127@MurD", "WP_071906075@Murg", "WP_071906152@Murb", "WP_071906154@Murd",
```

```
  "WP_084789739@Mura", "WP_088335892@Murd", "WP_088335895@Mura", "WP_088336109@Murg",
  "WP_088336155@Murb", "WP_112093959@Murd", "WP_112094175@Mura", "WP_112094176@Murb",
  "WP_112094177@Murg"
)
identifiants<-rownames(matrice)
##################################
# Définir une fonction pour retirer les identifiants de la matrice
retirer_identifiants <- function(matrice, identifiants_a_retirer) {
  indices_a_garder <- !rownames(matrice) %in% identifiants_a_retirer
  matrice_sans_identifiants <- matrice[indices_a_garder, indices_a_garder]
  return(matrice_sans_identifiants)
}
familles <- c("MurC", "MurD", "MurE", "MurF", "Murd", "Mura", "Murb", "Murg")
# Répéter le processus pour générer 100 matrices différentes
resultats <- list()
set.seed(42)  # Pour reproductibilité de l'aléatoire
for (i in 1:100) {
  # Sélectionner aléatoirement un identifiant de chaque famille
  identifiants_aleatoires <- character(0)
  for (famille in familles) {
    identifiants_famille <- grep(paste0("@", famille), identifiants, value = TRUE)
    if (length(identifiants_famille) > 0) {
      identifiant_aleatoire <- sample(identifiants_famille, 1)
      identifiants_aleatoires <- c(identifiants_aleatoires, identifiant_aleatoire)
    }
  }
 # print(identifiants_aleatoires)

  # Retirer les identifiants de la matrice de base
  matrice_sans_identifiants <- retirer_identifiants(matrice, identifiants_aleatoires)

  # Ajouter la matrice sans identifiants à la | des résultats
  resultats[[i]] <- matrice_sans_identifiants
}

# Afficher un exemple de matrice
print(resultats[[1]])

# Construire les arbres Neighbor Joining (NJ)
arbres_nj <- lapply(resultats, nj)
plot(arbres_nj[[1]], "phylogram", main = "Arbre Neighbor Joining")
#Transformation des arbres en formats Newick
arbres_newick <- lapply(arbres_nj, function(arbre) write.tree(arbre))
#Formation du fichier contenant tous les arbres
chemin_fichier <- "arbres_newick_DOM1.tree"
con <- file(chemin_fichier, "w")
for (arbre in arbres_newick) {
  writeLines(arbre, con)
}
close(con)
```

10.2.7 (10) SnakeM_myfam.pl

```
#!/usr/bin/env perl

use Modern::Perl '2011';
use autodie;
```

```
use Smart::Comments '###';
use File::Copy::Recursive qw(fcopy rcopy dircopy fmove rmove dirmove);

print <<'EOT';
The SnakeM_myfam program generates necessary files for Fold_Tree usage. Fold_tree requires a myfam folder
containing a struct folder and an identifiers.txt file. It locates the pdb files for the domain we wish to
analyze. Firstly, you need to create files named struct_{DOMAIN}containing all pdf files selected by the
domain. This directory (struct_{DOMAIN}) must be within a directory named after the domain.
|
|
Give the domaine name. EX: DOM1, DOM2, DOM3, DOM12, DOM23, Global >
EOT

#Receive the domain name
my $domaine = <>;
chomp $domaine;

print "|\n|\n| Creation du fichier myfam_{domaine} en cours... |\n|\n| Création du dossier relatif au \
domaine choisi en cours...\n|\n|";

my $name_dir = "myfam_$domaine";
mkdir $name_dir;

# Create de identifier file
my $fichier = "/Users/coralie/fold_tree/$name_dir/identifiers.txt";
open my $fh, '>', $fichier or die "Impossible d'ouvrir le fichier '$fichier' en écriture : $!";
close $fh;

#print "Struct$domaine\n $name_dir ";

my $fichier_DOM = "/Users/coralie/fold_tree/$domaine/Struct$domaine";
my $repertoire_DOM = "/Users/coralie/fold_tree/$name_dir/structs";
dircopy($fichier_DOM, $repertoire_DOM) or die "Copy failed: $!";

#Boostrap 100
print "Création des dossiers et suppression des individus en cours...\n";

for my $i (1..100) {
#copy the myfam folder 100 times
my $fichier   = "/Users/coralie/fold_tree/$name_dir";
my $repertoire_cp = "/Users/coralie/fold_tree/${name_dir}_$i";
dircopy( $fichier, $repertoire_cp) or die "Copy failed: $!";

#Remove randomly one individual of each group
my $repertoire = "/Users/coralie/fold_tree/${name_dir}_$i/structs";
opendir(my $dh, $repertoire) or die "Impossible d'ouvrir le répertoire '$repertoire': $!"; #open the directory

my %familles; # hash family

while (my $fichier = readdir($dh)) { #loop for find de mur* in the pdb file name
   # extract mur* name
    if ($fichier =~ /mur(\w)\.pdb/) { #regax
        my $famille = "mur$1";

        push @{$familles{$famille}}, $fichier; #push de new family
    }
}
```

```
closedir($dh); # close dir

# check the nbr of pdb file before
#print "Nombre de fichiers pour chaque famille avant la suppression :\n";
#foreach my $famille (sort keys %familles) {
#    my $nb_fichiers = scalar @{$familles{$famille}};
#    print "Famille $famille : $nb_fichiers fichiers\n";
#}

# delete one individual for each family
foreach my $famille (keys %familles) {
    my @fichiers = @{$familles{$famille}};
    my $indice = int(rand(scalar @fichiers)); # choose one random pdb file
    my $fichier_supprime = splice(@fichiers, $indice, 1); # delete
    @{$familles{$famille}} = @fichiers;

# Delete the real file in the dir
    my $chemin_fichier = "$repertoire/$fichier_supprime";
    unlink $chemin_fichier or warn "Impossible de supprimer le fichier '$fichier_supprime': $!";
    #print "Fichier supprimé de la famille $famille : $fichier_supprime\n";
}

}

print "|\n|\n|";
print "Dossiers dupliqués\n";
print "|\n|\n|";
print "Individus supprimés\n";
```

10.2.8 (11) foltree.sh

```
#!/bin/bash

# Activer l'environnement conda avec Mamba
#mamba activate foldtree

# Vérifier si le nombre d'arguments est correct
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 <global_value>"
    exit 1
fi

global_value="$1"
# Utiliser une boucle pour exécuter Snakemake pour chaque valeur de $j
for j in $(seq 1 100); do
    snakemake --cores 4 --use-conda -s ./workflow/fold_tree --config folder=./myfam_${global_value}_$j \
      filter=False custom_structs=True
done
```

10.2.9 (14) compo_murligase.R

```
#V.Lupo and C.Mullender
# packages loading
library(ggtree)
library(tidyverse)
```

```
#Files loading
tree_folder <- "Trees-topo-A/"
tree_files <- list.files(path = tree_folder, full.names = TRUE)
trees <- lapply(tree_files, read.tree)

#Names loading
basenames<-unlist(lapply(tree_files, basename))
cleaned_basenames <- gsub("topo-|\\.tre", "", basenames)

trees <- lapply(seq_along(tree_files), function(i) {
  tree <- read.tree(tree_files[i])
  tree_name <- tools::file_path_sans_ext(cleaned_basenames[i])
  attr(tree, "tree_name") <- tree_name
  tree$tip.label <- gsub("^[Mm][uU][rR]", "", tree$tip.label)
  return(tree)
})

# Change the trees' names in the trees object
names(trees) <- cleaned_basenames

# Define some colours for specific prefixes
colors <- c('a'='salmon', 'C'='lightgreen', 'E'='lightblue', 'g'='lightgoldenrod1', 'd'='pink1', \
  'D'='lightsalmon', 'F'='lavender', 'b'='violet')

# Duplicate the colors for the prefixes with '+' to match those without the '+'.
for (prefix in c('a', 'C', 'E', 'g', 'd', 'D', 'F', 'b')) {
  colors[paste0(prefix, '+')] <- colors[prefix]
  colors[paste0(prefix, '*')] <- colors[prefix]
}

# Change the classe top multiPhylo
class(trees) <- "multiPhylo"

#Create the figure with all the trees.
#ggtree(trees, layout = "unrooted") + facet_wrap(~.id, ncol = 10) + geom_label2(aes(label = label, \
  hjust = ifelse(isTip, 1, 0)), size = 1)
gg.tree <-ggtree(trees, layout = "unrooted") + facet_wrap(~.id, ncol = 10)
# Add branch labels without gray boxes
#gg.tree <- gg.tree + geom_tippoint(color = 'white', size = 3) + geom_tiplab()
gg.tree <- gg.tree + geom_point(aes(color = label), size = 3) + scale_color_manual(values = colors) \
  + geom_tiplab()
gg.tree <- gg.tree + theme(legend.position = "bottom", legend.justification = "right", legend.box = "horizontal")
gg.tree <- gg.tree + theme(strip.background = ifelse(gg.tree$data$.id %in% c("arbre1", "arbre3"), \
  element_rect(fill = "red"), element_blank()))

#gg.tree <- gg.tree + geom_hilight(            # highlights the nodes 6 and 7 in yellow
#  node = c(6,7),
#  fill = "yellow")

print(gg.tree)

#Save the figure
ggsave("trees-topo2.pdf", width = 29.7, height = 20)

################################################################## BACTERIA (or ARCHEA)
# load topology tree files
tree_folder <- "Length-branche/" #or Bipartition
tree_files <- list.files(path = tree_folder, full.names = TRUE)
trees <- lapply(tree_files, read.tree)
```

```r
#Names loading
basenames<-unlist(lapply(tree_files, basename))
cleaned_basenames <- gsub("topo-|\\.tre", "", basenames)

trees <- lapply(seq_along(tree_files), function(i) {
  tree <- read.tree(tree_files[i])
  tree_name <- tools::file_path_sans_ext(cleaned_basenames[i])
  attr(tree, "tree_name") <- tree_name
  tree$tip.label <- gsub("^[Mm][uU][rR]", "", tree$tip.label)
  return(tree)
})

# Change the trees' names in the trees object
names(trees) <- cleaned_basenames

# Define some colours for specific prefixes
colors <- c('C'='lightgreen', 'E'='lightblue', 'D'='lightsalmon', 'F'='lavender')
#colors <- c('a'='salmon', 'g'='lightgoldenrod1', 'd'='pink1', 'b'='violet') #if we do it for archea

# Duplicate the colors for the prefixes with '+' to match those without the '+'.
for (prefix in c( 'C', 'E', 'D', 'F','a','b','d','g')) {
  colors[paste0(prefix, '+')] <- colors[prefix]
}

# Change the classe top multiPhylo
class(trees) <- "multiPhylo"

#Create the figure with all the trees.
gg.tree <-ggtree(trees) + facet_wrap(~.id, ncol = 6) + ggtitle("Bactéries plus grande branche")
gg.tree <- gg.tree + geom_point(aes(color = label), size = 3) + scale_color_manual(values = colors) \
  + geom_tiplab()
print(gg.tree)

#Save the figure
ggsave("trees-topo-bacteries-L.pdf", width = 25, height = 10)
```

Références

Agarwal, G., Rajavel, M., Gopal, B., & Srinivasan, N. (2009). Structure-Based Phylogeny as a Diagnostic for Functional Characterization of Proteins with a Cupin Fold. *PLoS ONE*, *4*(5), e5736. https://doi.org/10.1371/journal.pone.0005736

Albers, S.-V., & Meyer, B. H. (2011). The archaeal cell envelope. *Nature Reviews. Microbiology*, *9*(6), 414–426. https://doi.org/10.1038/nrmicro2576

Ashkenazy, H., Abadi, S., Martz, E., Chay, O., Mayrose, I., Pupko, T., & Ben-Tal, N. (2016). ConSurf 2016 : An improved methodology to estimate and visualize evolutionary conservation in macromolecules. *Nucleic Acids Research*, *44*(Web Server issue), W344–W350. https://doi.org/10.1093/nar/gkw408

Auguie, B. (2010). *gridExtra : Miscellaneous Functions for "Grid" Graphics* (p. 2.3). Comprehensive R Archive Network. https://doi.org/10.32614/CRAN.package.gridExtra

Ayala, F. J., Escalante, A., O'Huigin, C., & Klein, J. (1994). Molecular genetics of speciation and human origins. *Proceedings of the National Academy of Sciences*, *91*(15), 6787–6794. https://doi.org/10.1073/pnas.91.15.6787

Balaji, S., & Srinivasan, N. (2007). Comparison of sequence-based and structure-based phylogenetic trees of homologous proteins : Inferences on protein evolution. *Journal of Biosciences*, *32*(1), 83–96. https://doi.org/10.1007/s12038-007-0008-1

Breitling, R., Laubner, D., & Adamski, J. (2001). Structure-based Phylogenetic Analysis of Short-chain Alcohol Dehydrogenases and Reclassification of the 17beta-Hydroxysteroid Dehydrogenase Family. *Molecular Biology and Evolution*, *18*(12), 2154–2161. https://doi.org/10.1093/oxfordjournals.molbev.a003761

Brinkmann, H., & Philippe, H. (1999). Archaea sister group of Bacteria ? Indications from tree reconstruction artifacts in ancient phylogenies. *Molecular Biology and Evolution*, *16*(6), 817–825. https://doi.org/10.1093/oxfordjournals.molbev.a026166

Bugg, D, B., Cg, D., & Di, R. (2011). Bacterial cell wall assembly : Still an attractive antibacterial target. *Trends in Biotechnology*, *29*(4). https://doi.org/10.1016/j.tibtech.2010.12.006

Bujnicki, J. M. (2000). Phylogeny of the Restriction Endonuclease-Like Superfamily Inferred from Comparison of Protein Structures. *Journal of Molecular Evolution*, *50*(1), 39–44. https://doi.org/10.1007/s002399910005

Chothia, C., & Lesk, A. M. (1986). The relation between the divergence of sequence and structure in proteins. *The EMBO Journal*, *5*(4), 823–826.

Dagan, T., Roettger, M., Bryant, D., & Martin, W. (2010). Genome networks root the tree of life between prokaryotic domains. *Genome Biology and Evolution*, *2*, 379–392. https://doi.org/10.1093/gbe/evq025

Devos, D. P. (2021). Reconciling Asgardarchaeota Phylogenetic Proximity to Eukaryotes and Planctomycetes Cellular Features in the Evolution of Life. *Molecular Biology and Evolution*, *38*(9), 3531–3542. https://doi.org/10.1093/molbev/msab186

Eme, L., & Tamarit, D. (2024). Microbial Diversity and Open Questions about the Deep Tree of Life. *Genome Biology and Evolution*, *16*(4), evae053. https://doi.org/10.1093/gbe/evae053

Gogarten, W., Vandermeulen, E., Van Aken, H., Kozek, S., Llau, J. V., Samama, C. M., & European Scoeity of Anaesthesiology. (2010). Regional anaesthesia and antithrombotic agents : Recommendations of the European Society of Anaesthesiology. *European Journal of Anaesthesiology*, *27*(12), 999–1015. https://doi.org/10.1097/EJA.0b013e32833f6f6f

Gribaldo, S., & Cammarano, P. (1998). The root of the universal tree of life inferred from anciently duplicated genes encoding components of the protein-targeting machinery. *Journal of Molecular Evolution*, *47*(5), 508–516. https://doi.org/10.1007/pl00006407

Hartmann, E., & Konig, H. (1990). Comparison of the biosynthesis of the methanobacterial pseudomurein and the eubacterial murein. *Naturwissenschaften*, *77*(10), 472–475. https://doi.org/10.1007/BF01135923

Hartmann, E., & König, H. (1994). A novel pathway of peptide biosynthesis found in methanogenic Archaea. *Archives of Microbiology*, *162*(6), 430–432. https://doi.org/10.1007/BF00282108

Illergård, K., Ardell, D. H., & Elofsson, A. (2009). Structure is three to ten times more conserved than sequence—A study of structural response in protein cores. *Proteins : Structure, Function, and Bioinformatics*, *77*(3), 499–508. https://doi.org/10.1002/prot.22458

Interactive Tree of Life (iTOL) v6 : Recent updates to the phylogenetic tree display and annotation tool | Nucleic Acids Research | Oxford Academic. (n.d.). https ://academic.oup.com/nar/advance-article/doi/10.1093/nar/gkae268/7645242 ?login=true.

Johnson, M. S., Sutcliffe, M. J., & Blundell, T. L. (1990). Molecular anatomy : Phyletic relationships derived from three-dimensional structures of proteins. *Journal of Molecular Evolution*, *30*(1), 43–59. https://doi.org/10.1007/BF02102452

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., … Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2

Kandler, O., & Konig, H. (1993). Chapter 8 Cell envelopes of archaea : Structure and chemistry. In M. Kates, D. J. Kushner, & A. T. Matheson (Eds.), *New Comprehensive Biochemistry* (Vol. 26, pp. 223–259). Elsevier. https://doi.org/10.1016/S0167-7306(08)60257-4

Kouidmi, I., Levesque, R. C., & Paradis-Bleau, C. (2014). The biology of Mur ligases as an antibacterial target. *Molecular Microbiology*, *94*(2), 242–253. https://doi.org/10.1111/mmi.12758

Lakshmi, B., Mishra, M., Srinivasan, N., & Archunan, G. (2015). Structure-Based Phylogenetic Analysis of the Lipocalin Superfamily. *PLOS ONE*, *10*(8), e0135507. https://doi.org/10.1371/journal.pone.0135507

Leahy, S. C., Kelly, W. J., Altermann, E., Ronimus, R. S., Yeoman, C. J., Pacheco, D. M., Li, D., Kong, Z., McTavish, S., Sang, C., Lambie, S. C., Janssen, P. H., Dey, D., & Attwood, G. T. (2010). The Genome Sequence of the Rumen Methanogen Methanobrevibacter ruminantium Reveals New Possibilities for Controlling Ruminant Methane Emissions. *PLOS ONE*, *5*(1), e8926. https://doi.org/10.1371/journal.pone.0008926

Léonard, R. R., Sauvage, E., Lupo, V., Perrin, A., Sirjacobs, D., Charlier, P., Kerff, F., & Baurain, D. (2022). Was the Last Bacterial Common Ancestor a Monoderm after All ? *Genes*, *13*(2), 376. https://doi.org/10.3390/genes13020376

Leps, B., Labischinski, H., Barnickel, G., Bradaczek, H., & Giesbrecht, P. (1984). A new proposal for the primary and secondary structure of the glycan moiety of pseudomurein. *European Journal of Biochemistry*, *144*(2), 279–286. https://doi.org/10.1111/j.1432-1033.1984.tb08461.x

Lopez, P., Forterre, P., & Philippe, H. (1999). The root of the tree of life in the light of the covarion model. *Journal of Molecular Evolution*, *49*(4), 496–508. https://doi.org/10.1007/pl00006572

Loughran, N. B., O'Connor, B., Ó'Fágáin, C., & O'Connell, M. J. (2008). The phylogeny of the mammalian heme peroxidases and the evolution of their diverse functions. *BMC Evolutionary Biology*, *8*(1), 101. https://doi.org/10.1186/1471-2148-8-101

Lupo, V., Roomans, C., Royen, E., Ongena, L., Jacquemin, O., Kerff, F., & Baurain, D. (2022). *Origin and Evolution of Pseudomurein Biosynthetic Gene Clusters* (p. 2022.11.30.518518). bioRxiv. https://doi.org/10.1101/2022.11.30.518518

Mandelstam, J., & Rogers, H. J. (1959). The incorporation of amino acids into the cell-wall mucopeptide of staphylococci and the effect of antibiotics on the process. *Biochemical Journal*, *72*(4), 654–662.

Mariani, V., Biasini, M., Barbato, A., & Schwede, T. (2013). lDDT : A local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, *29*(21), 2722–2728. https://doi.org/10.1093/bioinformatics/btt473

Meyer, B. H., & Albers, S.-V. (2020). Archaeal Cell Walls. In *eLS* (pp. 1–14). John Wiley & Sons, Ltd. https://doi.org/10.1002/9780470015902.a0000384.pub3

Mingorance, J., & Tamames, J. (2004). The bacterial dcw gene cluster : An island in the genome ? In M. Vicente, J. Tamames, A. Valencia, & J. Mingorance (Eds.), *Molecules in Time and Space : Bacterial Shape, Division and Phylogeny* (pp. 249–271). Springer US. https://doi.org/10.1007/0-306-48579-6_13

Moi, D., Bernard, C., Steinegger, M., Nevers, Y., Langleib, M., & Dessimoz, C. (2023). Structural phylogenetics unravels the evolutionary diversification of communication systems in gram-positive bacteria and their viruses. *bioRxiv*, 2023.09.19.558401. https://doi.org/10.1101/2023.09.19.558401

Naveenkumar, N., Prabantu, V. M., Vishwanath, S., Sowdhamini, R., & Srinivasan, N. (2022). Structures of distantly related interacting protein homologs are less divergent than non-interacting homologs. *FEBS Open Bio*, *12*(12), 2147–2153. https://doi.org/10.1002/2211-5463.13492

Nikolaichik, Y. A., & Donachie, W. D. (2000). Conservation of gene order amongst cell wall and cell division genes in Eubacteria, and ribosomal genes in Eubacteria and Eukaryotic organelles. *Genetica*, *108*(1), 1–7. https://doi.org/10.1023/a:1004077806910

Pazos, M., & Peters, K. (2019). Peptidoglycan. In A. Kuhn (Ed.), *Bacterial Cell Walls and Membranes* (pp. 127–168). Springer International Publishing. https://doi.org/10.1007/978-3-030-18768-2_5

Riniker, S., Eichenberger, A. P., & van Gunsteren, W. F. (2012). Solvating atomic level fine-grained proteins in supra-molecular level coarse-grained water for molecular dynamics simulations. *European Biophysics Journal*, *41*(8), 647–661. https://doi.org/10.1007/s00249-012-0837-1

Rozewicki, J., Li, S., Amada, K. M., Standley, D. M., & Katoh, K. (2019). MAFFT-DASH : Integrated protein sequence and structural alignment. *Nucleic Acids Research*, *47*(W1), W5–W10. https://doi.org/10.1093/nar/gkz342

Sala, D., Engelberger, F., Mchaourab, H. S., & Meiler, J. (2023). Modeling conformational states of proteins with AlphaFold. *Current Opinion in Structural Biology*, *81*, 102645. https://doi.org/10.1016/j.sbi.2023.102645

Sham, L.-T., Butler, E. K., Lebar, M. D., Kahne, D., Bernhardt, T. G., & Ruiz, N. (2014). MurJ is the flippase of lipid-linked precursors for peptidoglycan biogenesis. *Science (New York, N.Y.)*, *345*(6193), 220–222. https://doi.org/10.1126/science.1254522

Smith. (2006). Structure, Function and Dynamics in the *Mur* Family of Bacterial Cell Wall Ligases. *Journal of Molecular Biology*, *362*(4), 640–655. https://doi.org/10.1016/j.jmb.2006.07.066

Smith, Doucette-Stamm, L. A., Deloughery, C., Lee, H., Dubois, J., Aldredge, T., Bashirzadeh, R., Blakely, D., Cook, R., Gilbert, K., Harrison, D., Hoang, L., Keagle, P., Lumm, W., Pothier, B., Qiu, D., Spadafora, R., Vicaire, R., Wang, Y., … Reeve, J. N. (1997). Complete genome sequence of Methanobacterium thermoautotrophicum deltaH: Functional analysis and comparative genomics. *Journal of Bacteriology*, *179*(22), 7135–7155.

Steenwyk, J. L., Li, Y., Zhou, X., Shen, X.-X., & Rokas, A. (2023). Incongruence in the phylogenomics era. *Nature Reviews Genetics*, *24*(12), 834–850. https://doi.org/10.1038/s41576-023-00620-x

Subedi, B. P., Martin, W. F., Carbone, V., Duin, E. C., Cronin, B., Sauter, J., Schofield, L. R., Sutherland-Smith, A. J., & Ronimus, R. S. (2021). Archaeal pseudomurein and bacterial murein cell wall biosynthesis share a common evolutionary ancestry. *FEMS Microbes*, *2*, xtab012. https://doi.org/10.1093/femsmc/xtab012

Subedi, B. P., Schofield, L. R., Carbone, V., Wolf, M., Martin, W. F., Ronimus, R. S., & Sutherland-Smith, A. J. (2022). Structural characterisation of methanogen pseudomurein cell wall peptide ligases homologous to bacterial MurE/F murein peptide ligases. *Microbiology*, *168*(9), 001235. https://doi.org/10.1099/mic.0.001235

van Kempen, M., Kim, S. S., Tumescheit, C., Mirdita, M., Lee, J., Gilchrist, C. L. M., Söding, J., & Steinegger, M. (2024). Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, *42*(2), 243–246. https://doi.org/10.1038/s41587-023-01773-0

Varadi, M., Bertoni, D., Magana, P., Paramval, U., Pidruchna, I., Radhakrishnan, M., Tsenkov, M., Nair, S., Mirdita, M., Yeo, J., Kovalevskiy, O., Tunyasuvunakool, K., Laydon, A., Žídek, A., Tomlinson, H., Hariharan, D., Abrahamson, J., Green, T., Jumper, J., … Velankar, S. (2024). AlphaFold Protein Structure Database in 2024 : Providing structure coverage for over 214 million protein sequences. *Nucleic Acids Research*, *52*(D1), D368–D375. https://doi.org/10.1093/nar/gkad1011

Visweswaran, G. R. R., Dijkstra, B. W., & Kok, J. (2011). Murein and pseudomurein cell wall binding domains of bacteria and archaea—a comparative view. *Applied Microbiology and Biotechnology*, *92*(5), 921–928. https://doi.org/10.1007/s00253-011-3637-0

Wickham, H. (2007). Reshaping Data with the reshape Package. *Journal Of Statistical Software*, *21*, 1–20. https://doi.org/10.18637/jss.v021.i12

Wickham, H. (2016). *Ggplot2*. Springer International Publishing. https://doi.org/10.1007/978-3-319-24277-4

Yariv, B., Yariv, E., Kessel, A., Masrati, G., Chorin, A. B., Martz, E., Mayrose, I., Pupko, T., & Ben-Tal, N. (2023). Using evolutionary data to make sense of macromolecules with a “face-lifted” ConSurf. *Protein Science*, *32*(3), e4582. https://doi.org/10.1002/pro.4582

Zhang, Y., & Skolnick, J. (2004). Scoring function for automated assessment of protein structure template quality. *Proteins : Structure, Function, and Bioinformatics*, *57*(4), 702–710. https://doi.org/10.1002/prot.20264

Zhang, Y., & Skolnick, J. (2005). TM-align : A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, *33*(7), 2302–2309. https://doi.org/10.1093/nar/gki524

Zhaxybayeva, O., & Doolittle, W. F. (2011). Lateral gene transfer. *Current Biology*, *21*(7), R242–R246. https://doi.org/10.1016/j.cub.2011.01.045