Université de Montréal

Développement de méthodes d'assemblage de génomes *de novo* adaptées aux bactéries endosymbiotes

par Jean-François Théroux

Département de biochimie et médecine moléculaire
Faculté de médecine

Mémoire présenté à la Faculté de médecine en vue de l'obtention du grade de maîtrise en bio-informatique

30 avril 2015

Résumé

Le but de ce projet était de développer des méthodes d'assemblage *de novo* dans le but d'assembler de petits génomes, principalement bactériens, à partir de données de séquençage de nouvelle-génération. Éventuellement, ces méthodes pourraient être appliquées à l'assemblage du génome de StachEndo, une Alpha-Protéobactérie inconnue endosymbiote de l'amibe *Stachyamoeba lipophora*. Suite à plusieurs analyses préliminaires, il fut observé que l'utilisation de lectures Illumina avec des assembleurs par graphe DeBruijn produisait les meilleurs résultats. Ces expériences ont également montré que les contigs produits à partir de différentes tailles de k-mères étaient complémentaires pour la finition des génomes. L'ajout de longues paires de lectures chevauchantes se montra essentiel pour la finition complète des grandes répétitions génomiques. Ces méthodes permirent d'assembler le génome de StachEndo (1,7 Mb). L'annotation de ce génome permis de montrer que StachEndo possède plusieurs caractéristiques inhabituelles chez les endosymbiotes. StachEndo constitue une espèce d'intérêt pour l'étude du développement endosymbiotique.

Mots clés : *assemblage de génome de novo, séquençage nouvelle-génération, qualité d'assemblage, graphe DeBruijn, k-mère, endosymbiote, Rickettsiales*

Summary

The goal of this project was to develop *de novo* genome assembly methods adapted to small genomes, especially bacterial, using next-generation sequencing data. Eventually, these methods could be used to assemble the genome of StachEndo, an unknown Alpha-Proteobacteria ensymbiont of the *Stachyamoeba lipophora* amoeba. Preliminary findings showed that the use of Illumina reads with DeBruijn graph assemblers yielded the best results. These experiments also showed that contigs produced with k-mers of various sizes were complementary in genome finishing assays. The addition of long-range paired-end reads proved necessary to fully close genomic assembly gaps. These methods made the assembly of StachEndo's genome (1.7 Mb) possible. Through the annotation of StachEndo's genes, several features that are unusal for endosymbionts were identified. StachEndo seems to be an interesting species for the study of endosymbiotic evolution.

Keywords : *de novo genome assembly, next-generation sequencing,assembly quality, DeBruijn graph , k-mer, endosymbiont, Rickettsiales*

Table des matières

Table des tableaux

Table des figures

Abbréviations

ADN - Acide désoxyribonucléique

ADNc - Acide désoxyribonucléique complémentaire

ARN - Acide ribonucléique

ARNr - Acide ribonucléique ribosomal

ARNt - Acide ribonucléique de transfert

DBG - Graphe DeBruijn

Mb - Méga base (un million de base)

Mpb - Méga paire de base (un million de paires de bases)

nt - nucléotide

OLC - *Overlap-Layout-Consensus* (type d'assembleur)

pb - paire de bases

PCR - Réaction en chaîne par polymérase

TCA - Acide tricarboxylique/acide citrique

Dédicace

A mes parents, ma sœur et mon beau-frère,

pour m'avoir encouragé et soutenu constamment tout au long de cette aventure.

Vous pouvez enfin arrêter de vous inquiéter et vous reposer.

A mes amis de toujours,

pour être restés présents malgré toutes les fois où je vous ai laissé tombés à la dernière minute parce que j'avais trop de travail.

Je n'aurai plus d'excuses maintenant.

A mes professeurs et collègues,

pour avoir fait de moi une personne plus confiante et un meilleur scientifique.

A bientôt, sûrement.

A mes étudiants,

pour avoir égayé mes journées et m'avoir permis de mûrir et de vaincre ma timidité.

J'espère que je vous aurai autant appris que vous ne m'en avez appris.

Puis,

Au futur

et à ce qu'il réserve...

Remerciements

Je tiens à remercier mon directeur de maîtrise, **Franz Lang**
pour m'avoir accueilli dans son laboratoire et m'avoir guidé au cours de ces années,

Mon parrain de maîtrise, **Nicolas Lartillot**, pour tous ses conseils,

Gertraud Burger, **Sandrine Moreira**, **Tizziana Cambiotti**, **Lise Forget** et **Nicolas Schweiger** pour leur collaboration au cours de ce projet,

Toute l'équipe du **Centre Robert-Cedergren**,
mais plus particulièrement à tous mes chers amis et bio-informaticiens (présents et passés):
Gabrielle Thauvette, **Natacha Beck**, **Matt Sarrasin**, **Simon Laurin-Lemay**, **Sahar Parto**, **Raphaël Poujol**, **Benjamin Matala**, **Eric Fournier** et **Lili-Anh Le Minh**,

Sans oublier,
Hervé Philippe, **Henner Brinkmann**, **Yaoqing Shen** et **Sophie Breton**,

Ainsi que des remerciements spéciaux à
Elaine Meunier et **Marie Pageau**.

Encore une fois,
Merci à tous.

1- INTRODUCTION

Le développement du séquençage fut une révolution technologique importante, permettant pour la première fois de connaître la séquence d'une molécule d'ADN. Cette percée donna naissance au domaine de la génomique telle que nous la connaissons. L'arrivée des méthodes de séquençage de nouvelle génération amorça une autre révolution.

A - Contexte théorique du projet

A.1 - Théorie du séquençage de nouvelle-génération

Les méthodes de séquençage de nouvelle-génération permettent de séquencer un grand nombre de molécules d'ADN en parallèle (1). Ces techniques ont comme avantage d'être beaucoup plus rapides et moins dispendieuses que les anciennes méthodes (1, 2) (comme Sanger (3)). Cependant, ces techniques ne peuvent déterminer la séquence d'un chromosome entier d'un seul coup. Il est nécessaire de fragmenter l'ADN (génomique ou complémentaire) qui nous intéresse et de séquencer ces fragments en parallèle (1). La séquence d'un fragment d'ADN généré par le séquençage est appelé lecture.

Différents protocoles de séquençage produisent différents types de lectures. On distingue généralement deux types de lectures de séquençage, les longues et les courtes. Le terme « courtes lectures » est habituellement utilisé pour décrire les lectures produites avec des techniques de séquençage comme Illumina (4) et Ion Torrent (5) qui, initialement, généraient des lectures de tailles allant de 50 à 100 paires de bases (pb). Avec l'amélioration de ces méthodes, Illumina et Ion Torrent peuvent maintenant produire des lectures allant jusqu'à environ 300 pb (6, 7). Pour ce qui est du terme «longues lectures », il décrit les lectures plus longues que quelques centaines de paires de bases. Pensons entre-autres aux lectures 454 (8) pouvant atteindre une taille de 700 pb et aux lectures PacBio (9) qui dépassent même la taille des lectures produites par les méthodes traditionnelles de Sanger (environ 900 pb) en atteignant les 10 000 pb et plus.

Compte tenu de différences technologiques significatives entre ces méthodes de séquençage,

chacune possède ses propres artefacts (erreurs) systématiques compliquant leurs utilisations. De plus la séquence de la molécule d'ADN ou d'ARN initiale devra être reconstruite (assemblée) en se fiant à la séquence des lectures que nous avons générées.

A.2 - Théorie de l'assemblage de nouvelle-génération

Il existe deux grands types d'assemblages, les assemblages avec référence génomique et les assemblages *de novo* (10). Les assemblages avec référence consistent à aligner les lectures de séquençage par rapport à un génome de référence en se basant sur leur similarité avec ce dernier (10). Cette méthode nécessite évidemment une séquence de référence très proche du génome que nous avons séquencé. Dans les cas où aucune référence n'est disponible, il faut utiliser les assemblages *de novo* (10). Ce problème est beaucoup plus complexe, mais il est nécessaire d'utiliser les approches *de novo* si nous voulons étudier des organismes totalement inconnus ou trop divergents des références disponibles.

L'assemblage *de novo* de certains génomes pose de nombreuses difficultés ; notamment, la formation de «trous» dans l'assemblage, causé par des régions répétées plus longues que la taille des fragments séquencés (10). Par rapport aux régions génomiques uniques, qui sont relativement uniformes, les régions répétées ont une grande «couverture» car toutes leurs copies sont assemblées ensembles sans prendre en compte les régions uniques qui les séparent (10). Également, certaines régions peuvent être difficiles à séquencer pour des raisons biochimiques (11-13), sans compter les erreurs systématiques associées aux diverses techniques de séquençage utilisées. Les sources d'erreurs potentielles sont donc nombreuses, particulièrement dans les expériences d'assemblage *de novo*.

Dans ce projet, nous avons voulu développer des techniques d'assemblage *de novo* appropriées aux génomes bactériens. Compte tenu des nombreuses embûches associées au séquençage et à l'assemblage de génome, notre but premier fut d'arriver à obtenir une marche à suivre quasi-automatique, facile à utiliser et donnant des résultats fiables. L'obtention de résultats fiables est essentielle si nous voulons utiliser ces assemblages pour accomplir diverses analyses dont les résultats dépendent en grande partie de la fiabilité des données utilisées, comme l'étude de

l'organisation du génome, l'annotation de gènes, l'inférence d'arbres phylogénétique, ainsi que de nombreuses autres méthodes de génomique comparative.

B - Caractéristiques des méthodes de séquençage de nouvelle-génération

Pour accomplir cette tâche, il est nécessaire de pouvoir comparer l'assemblage d'un même génome sous différentes conditions avec un standard fiable. À cette fin, un séquençage Sanger (standard fiable, incluant analyses et contrôles manuels par un expert), 454 et Illumina du génome mitochondrial de l'amibe terrestre *Stachyamoeba lipophora* fut réalisé.

B.1 – Séquençage 454

Tout d'abord, le protocole 454, tel que mentionné précédemment, génère des « longues » lectures. Il s'agit d'une réaction chimique de pyroséquençage lors de laquelle l'intégration d'un nucléotide est détectée par le biais d'une réaction enzymatique photoémettrice (8). Une erreur systématique plutôt difficile à contourner est associée à cette réaction. Cette erreur, probablement la mieux caractérisée de 454, est le compte erroné du nombre de nucléotides composant un homopolymère (8). Un homopolymère est une répétition d'un même nucléotide plusieurs fois de suite (ex : 5'-AAAAAAA-3'). En raison d'une limitation enzymatique dans la réaction de séquençage, il est impossible de déterminer avec certitude la longueur de ces répétitions si elles sont plus longues que 6 à 10 nucléotides (8). 454 présente également différents artefacts se reflétant par l'insertion ou la délétion de positions dans la séquence nucléotidique (0,4-1%) (14-16).

B.2 – Séquençage Illumina

Dans un deuxième temps, les lectures Illumina sont considérées comme de « courtes » lectures bien que certaines avancées dans la technologie de séquençage leur permettent maintenant de rivaliser avec les lectures 454 en terme de taille (présentement 300 nt) (6, 7). Le séquençage Illumina est réalisé par amplification locale sur un support de verre, de façon à produire des fagots d'ADN (4). Ce procédé sert à augmenter localement la force du signal associé à un

fragment (4). Plusieurs artéfacts plus ou moins fréquents ont été associés à ce protocole. La présence de motifs spécifiques ou de structures secondaires possibles sur les fragments séquencés semble en être la cause, induisant un déphasage du séquençage parmi les fragments amplifiés (17). Les erreurs de type déphasage affligent ~0,015% des lectures Illumina (14). On retrouve également un taux de substitution de nucléotides plus élevé qu'avec la technologie 454 (13, 14, 18). La formation de lectures chimériques et la présence de nucléotides provenant d'amorces et d'adapteurs de séquençage limitent également l'utilisabilité des lectures de séquençage (19).

C - Théorie des assembleurs *de novo*

Pour assembler des lectures de séquençage, nous utilisons des logiciels appelés assembleurs. Les assembleurs, à l'aide d'algorithmes variés, utilisent les lectures et tentent de les agencer de façon à obtenir la séquence complète de la molécule séquencée. Lorsque plusieurs lectures sont assemblées, elles forment ce qu'on appelle un « contig ». Il s'agit essentiellement d'une portion de la séquence finale que nous cherchons à assembler, et donc de la molécule d'ADN ou du génome séquencé.

La plupart des algorithmes d'assemblage *de novo* de génomes couramment utilisés appartiennent à deux grandes familles.

C.1 - Assembleur de type *Overlap-Layout-Consensus*

La première famille, celle des algorithmes de type *Overlap-Layout-Consensus* (OLC), a été élaborée pour la première fois en 1980 (20). Elle est mieux adaptée à l'assemblage de longues lectures (ex : Sanger et 454). Ces algorithmes commencent par calculer tous les chevauchements possibles entre toutes les paires de lectures fournies (10, 21). Ces informations servent à créer un graphe dont les nœuds représentent chacune des lectures. Deux nœuds reliés entre eux correspondent à deux lectures qui se chevauchent. L'assembleur cherche ensuite un chemin qui traverse le graphe en passant par chaque nœud une seule fois (10). Les contigs sont construits au fur et à mesure que les nœuds sont sélectionnés et que

leurs lectures respectives sont ajoutées à la séquence consensus de l'assemblage. En ne sélectionnant que les chemins qui passent une seule fois par chaque nœud, on s'assure de ne pas intégrer la même lecture dans plusieurs contigs (10). Cette étape correspond à chercher un chemin Hamiltonien dans un graphe (10). Les chemins divergents peu couverts (engendrés par des lectures erronées) sont retirés pour simplifier la résolution de l'assemblage et le risque de commettre des erreurs (10).

Dans les cas où l'assemblage semble comporter des régions répétées, les portions du graphe correspondant à ces régions sont masquées et l'assembleur se concentre sur les portions uniques de l'assemblage. On considère donc des chemins correspondant à des sous-graphes du graphe entier excluant les zones répétées. On obtiendra alors plusieurs contigs (10). Par la suite, l'assembleur tente de résoudre les répétitions en comblant les trous présents entre les différents contigs à l'aide de lectures plus longues ou de jeux de données pairées, lorsque disponibles (10).

Les premières générations d'algorithmes OLC étaient plus susceptibles de faire des erreurs lors de la phase de résolution des zones répétées. Par exemple, imaginons des régions uniques A, C et D séparées par une région répétée B (voir **Figure 1.1**). Il serait possible lors de la phase d'assemblage que les deux répétitions soient écrasées et que l'on obtienne un assemblage erroné (voir **Figure 1.2 et 1.2'**) (10). Cependant, les implémentations actuelles des algorithmes OLC sont plus prudentes et vont généralement briser les contigs aux limites des zones répétitives pour éviter la création de séquences chimériques (10). Par conséquent, il est commun de retrouver des régions répétées hautement couvertes aux extrémités de contigs (10).

Les algorithmes OLC utilisent une approche assez simple et instinctive qui est très modulaire. De plus, ils peuvent considérer des chevauchements de taille variable entre les lectures à assembler, permettant une discrimination plus facile des meilleurs chevauchements. Cependant, pour ce faire, il est nécessaire de calculer le meilleur chevauchement possible entre toutes les paires de lectures ce qui peut devenir très coûteux en temps et en mémoire, surtout si la quantité de lectures utilisée est très importante. Par conséquent, l'utilisation des

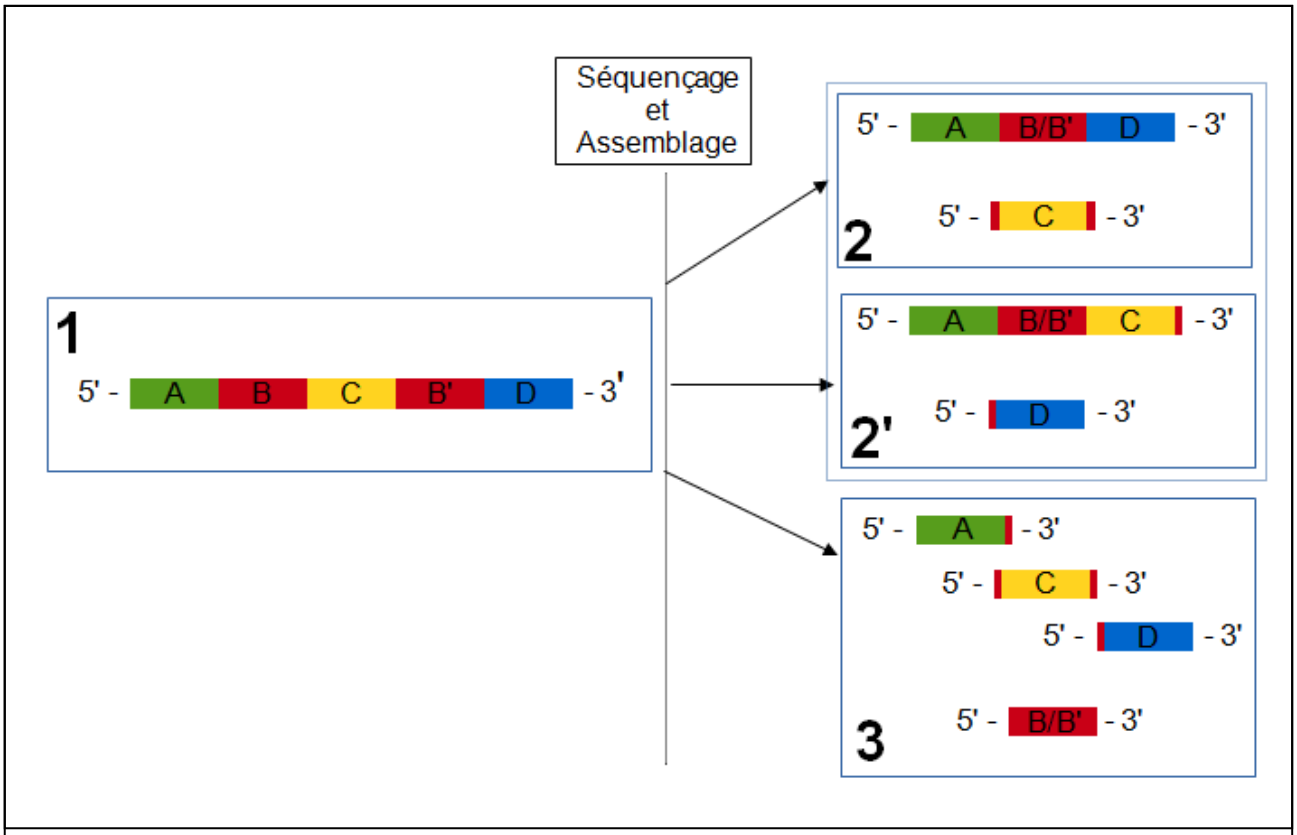


Figure 1. Représentation schématisée de l'affaissement d'une région répétée sur elle-même lors de l'assemblage de séquences. (1) Séquence contenant une région répétée deux fois (la région B et la répétition de cette région B') encadrée par trois régions uniques A, C et D. (2) Assemblage où les régions B et B' se retrouvent écrasées menant à la formation d'un trou et à l'ordre erronés des régions uniques A-B-D. (2') Assemblage où les régions B et B' se retrouvent écrasées menant à la formation d'un trou. (3) Assemblage où les régions répétées sont affaissée indépendamment des régions uniques, créant plus de trous mais ne bloquant pas l'assemblage.

chevauchements entre les lectures comme base de l'assemblage fait en sorte que ces algorithmes sont plus appropriés pour l'assemblage des longues lectures d'ADN de haute qualité (22). En effet, puisque plus un chevauchement est long, plus il est statistiquement significatif, les chevauchements entre de plus longues lectures ont le potentiel d'être plus uniques et significatifs que ceux entre des séquences plus courtes. De plus, trouver un chemin Hamiltonien à l'intérieur d'un graphe est un problème NP-complet, rendant la résolution du problème d'assemblage assez difficile (10). **Newbler (8), Celera (23), Mira (24) et Arachne**

(25) sont des exemples d'assembleurs de type OLC.

C.2 - Assembleur de type graphe DeBruijn

La deuxième famille d'algorithmes d'assemblage est celle des assembleurs de type graphe DeBruijn (DBG). Cette méthode, qui est mieux adaptée aux courtes lectures, a été présentée en 2001 avec le logiciel Euler. Le principe derrière les assembleurs DBG vise à simplifier la recherche d'un chemin Hamiltonien dans le graphe en le transformant en recherche d'un chemin Eulérien (10, 26-28). Un chemin Eulérien est un chemin qui passe à travers tous les arcs d'un graphe une seule fois. Contrairement à la recherche d'un chemin Hamiltonien qui est NP-complet, la recherche d'un chemin Eulérien peut se faire en temps polynomial (28).

Pour ce faire, on construit un graphe représentant le chevauchement entre k-mères plutôt qu'entre lectures (26). Un k-mère est une sous-séquence d'une longueur donnée k provenant d'une lecture. Dans les méthodes DBG, on commence par recenser tous les k-mères d'une taille donnée présents dans nos lectures (10, 26). On construit ensuite un graphe dont les nœuds correspondent à chaque séquence unique de k-mère (10, 26). Les arcs entre deux nœuds indiquent un chevauchement entre ces deux k-mères, tel que les k-1 derniers nucléotides du premier k-mère sont identiques aux k-1 premiers nucléotides du deuxième (21, 26).

Bien qu'il semble non-intuitif de briser nos lectures en plus petits fragments alors que le but de l'assemblage est de combiner nos lectures, cette transposition de problème a pour effet de simplifier le graphe considéré (26, 28). De cette façon toutes les séquences identiques de taille k présentes dans le jeu de données ne se trouvent représentées qu'une seule fois dans le graphe plutôt que chacune des lectures (26, 28). L'assemblage est ensuite déduit en sélectionnant le chemin correspondant aux nœuds du graphe traversés par ces lectures. De plus, en considérant les lectures de façon partielle, il est possible d'utiliser des portions d'une lecture pour construire le graphe, malgré le fait qu'elle puisse contenir des erreurs de séquençage ou des restants d'amorces.

L'utilisation de graphes DeBruijn permet l'utilisation de nombreux algorithmes pour faciliter

la recherche de chemins Eulériens ce qui simplifie grandement plusieurs étapes de résolution du graphe (26). Par contre, un des paramètres essentiels à une bonne utilisation des graphes DeBruijn, la taille de k-mère, n'est pas aussi facile à manier. La taille de k-mère choisie change grandement l'allure du graphe produit. En supposant qu'il existe une taille optimale de k-mère que l'on décrira comme « k » qui est un nombre impair (pour éviter qu'un k-mère soit également son propre complément inverse (27)) positif forcément plus petit que la taille de nos lectures, choisir une valeur de k-mère très inférieure ou très supérieure à k pourrait devenir nuisible à l'assemblage. Si une valeur de k-mère trop petite est choisie, le graphe résultant sera difficile à résoudre (27-29), Si nous prenons l'exemple de la **Figure 2**, une taille de k-mère petite signifiera que les k-mères générés seront probablement très redondants, par conséquent de nombreux arcs quitteront et arriveront de chacun des nœuds du graphe, produisant un nombre important de cycles dans le graphe (27). Tout cycle à l'intérieur du graphe DeBruijn est le signe de la présence d'une répétition, puisqu'il existe un arc partant du nœud courant vers un nœud déjà visité.

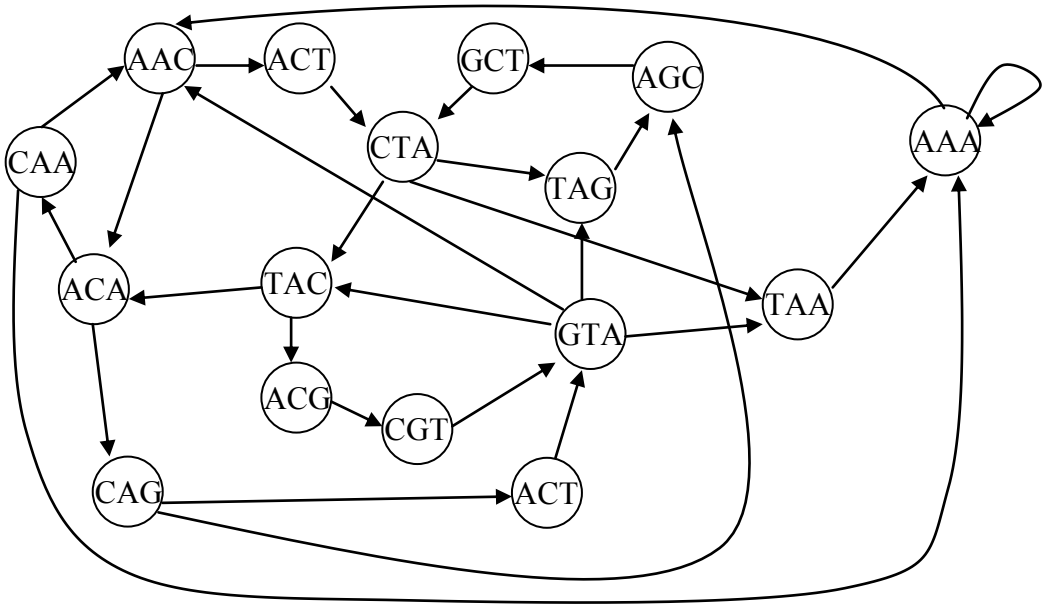
Tel que mentionné précédemment, les répétitions constituent un des éléments les plus difficiles à résoudre de l'assemblage. La présence de cycles dans le graphe DeBruijn aurait pour effet de perturber l'assemblage, puisque passer par un cycle signifie que l'on choisirait un arc plus d'une fois ce qui ne respecte pas les contraintes d'un chemin Eulérien (27, 29, 30). Imaginons une situation où on essaye d'assembler des lectures provenant d'une région similaire à celle présentée à la **Figure 1.1** (A-B-C-B'-D). L'assemblage qui en résulterait serait donc divisé en cinq parties, un contig représentant la région précédent la première répétition, un autre contig représentant la région suivant la deuxième répétition, un troisième contig situé entre les deux répétions et un dernier contig représentant la région répétée (voir **Figure 1.3**). La couverture de ce contig sera plus élevée que celle des autres contigs puisqu'il sera composé des lectures appartenant aux deux régions répétées sans discrimination. Les graphes DeBruijn n'ont pas de problème avec les répétions si elles sont plus courtes que la longueur du k-mère, puisque la répétition ne se reflète pas dans l'organisation du graphe (29, 30). Par conséquent, cette propriété nous inciterait à choisir la taille de k-mer la plus grande possible.

Séquence d'origine: ACTAGCTACAACAGTACGTACGTAAAC

Lectures de séquençage:

```
ACTAGCTA
  TAGCTACA
    GCTACAAC
        AACAGTA
         ACAGTACG
            GTACGTAC
               CGTACGTA
                  CGTAAAC
```

longueur de k-mère = 3



longueur de k-mère = 5

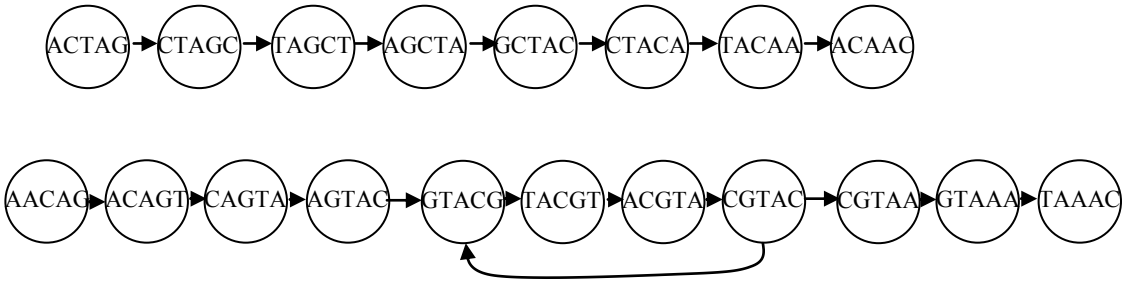


Figure 2. Graphes DeBruijn du même jeu de données théoriques à partir de k-mères de tailles différentes. (A) Avec un k-mère de taille 3, le graphe DeBruijn comporte de multiples cycles rendant sa résolution difficile. (B) Avec un k-mère de taille 5, il n'existe qu'un seul cycle dans le graphe. Cependant, la couverture inégale de la molécule séquencée empêche l'observation du k-mère «CAACA» détruisant la connexité du graphe.

Choisir une taille de k-mère égale à la taille de nos lectures revient à utiliser une approche très proche des algorithmes OLC, puisqu'on considère au final l'alignement entre les séquences entières de nos lectures. Choisir une taille de k-mère trop grande, c'est-à-dire beaucoup plus grande que la taille k optimale n'est pas non plus sans risque. En effet, les graphes DeBruijn représentent la superposition exacte entre les k-mères. L'introduction d'erreurs dans les lectures lors du séquençage a donc pour effet de créer des lectures dont la séquence diverge de celle des autres et qui entraînent l'apparition de nouveaux k-mères qui complexifient la structure du graphe (27, 29). Il est également possible qu'à cause d'une faible couverture dans certaines régions séquencées, ou d'une densité d'erreur trop importante, qu' il n'y ait pas assez de lectures pour générer tous les k-mères nécessaires à l'assemblage de lectures qui se chevauchent sur une longueur plus courte que la taille du k-mère (voir **Figure 2**) (27-29). Ce phénomène crée littéralement un trou dans le graphe et dans l'assemblage (27-29). Les séquences plus courtes que la taille de k-mères seront également exclues. Des tailles de k-mères trop élevés rendent donc l'assemblage vulnérable aux erreurs de séquençage et aux régions de couvertures inégales (29).

Cependant, à l'inverse, utiliser des k-mères de grandes tailles, permet de ne conserver que les régions très bien soutenues de notre assemblage (29). Puisqu'elles sont composées de plus de lectures, elles résistent mieux à la perte de séquences causée par la présence d'erreurs.

En résumé, trouver la taille de k-mère optimal est un problème difficile puisqu'il s'agit d'une valeur qui dépend à la fois de la longueur des lectures disponibles, de la présence de répétions, de la densité d'erreurs dans les lectures et de la couverture de séquençage. **Velvet (27), SOAP DeNovo (31) et ABySS (32) sont des exemples d'assembleur DBG.**

C.3 – Le rôle de la couverture dans les assembleurs de type Graphe DeBruijn

Avec le développement des assembleurs, certains algorithmes ont introduit le paramètre de seuil de couverture qui permet d'établir la couverture de séquençage attendue et donc de faire la distinction à l'intérieur du graphe entre les zones répétées (couverture élevée), les zones erronées (couverture faible) et les régions justes (couverture moyenne attendue) (33, 34). Tout

de même, bien que ces améliorations facilitent l'assemblage, la résolution des répétions plus longues que la taille des lectures reste un problème non-résolu (27-29). Cependant, le traitement des répétitions par les graphes DeBruijn rend cette méthode plus intéressante que les méthodes OLC; car au lieu de masquer les régions répétées, elles sont représentées dans le graphe DeBruijn. Bien que de multiples permutations soient possibles autour des zones répétées, leur présence est directement modélisée par le graphe qui se trouve à être une source d'informations potentiellement très utile pour la finition de l'assemblage.

D - Finition d'assemblages partiels

La finition consiste à (essayer de) combler les trous qui séparent notre assemblage en plusieurs contigs (35). À cette étape, notre assemblage peut se présenter en un mélange de deux formes : sous la forme des contigs ou sous la forme de *scaffolds* (échafaudages) (35, 36). Un *scaffold* est une structure composée de plusieurs contigs dont on connaît l'orientation relative, ainsi que la distance approximative les séparant (35-37). Les *scaffolds* peuvent être construits par les assembleurs s'ils ont à leur disposition des lectures pairées (*paired-end*). Les lectures pairées sont des paires de lectures séquencées aux deux extrémités d'un même fragment d'ADN. De cette façon en connaissant la taille du fragment séquencé et la taille des lectures générées, nous pouvons déduire les nombres de nucléotides qui se trouvent entre ces deux lectures. L'utilisation de ces types de lectures permet la formation de *scaffolds* qui peuvent être utilisés comme squelettes pour la finition de l'assemblage, et surtout à la résolution de répétitions (36, 37). Il existe un autre type de lectures pairées appelé *paired-end overlap*. Lorsque la taille du fragment séquencé est plus petite que la somme des tailles des lectures, il n'y a pas d'espace séparant les lectures d'une paire. Les deux lectures se chevauchent donc. Ces lectures peuvent être fusionnées pour obtenir de plus longues lectures.

Les outils de finitions actuels utilisent généralement une des deux approches suivantes : l'extension des extrémités des contigs (38-40) ou le remplissage de *scaffolds* (38, 41-44). L'extension des extrémités de contigs consiste à chercher des lectures dont la séquence est similaire à celle de l'extrémité d'un contig (40). Les lectures qui correspondent à cette séquence sont ajoutées aux contigs pour l'allonger progressivement (40). Éventuellement ces

contigs allongés pourraient se chevaucher, menant ainsi à la création d'un nouveau contig plus gros. Le remplissage de *scaffolds* va plutôt utiliser des lectures pairées pour essayer de former des *scaffolds* et de trouver s'il existe déjà des contigs qui pourraient combler ces *scaffolds* (41, 44). En utilisant l'information de paire, il est possible de connaître la distance séparant deux lectures. De cette façon on peut déduire l'emplacement de certains contigs. Dans certains cas cette information peut être utilisée pour résoudre le positionnement des régions répétitives de l'assemblage.

Cependant, les approches ne se concentrant que sur un seul trou à la fois, ou une seule extrémité à la fois, risquent fortement de commettre des erreurs d'assemblage. Une lecture ne peut être placée qu'à un seul endroit dans l'assemblage. Lorsqu'on considère des régions répétitives qui sont la cause principale de trous dans l'assemblage, particulièrement avec les approches DBG, on retrouve souvent des lectures qui pourraient être placées à plus d'un endroit. En n'observant qu'une seule partie du problème à la fois on risque de ne pas considérer la présence d'un conflit potentiel et de procéder à des ajouts erronés qui pourraient bloquer le processus d'assemblage et de finition en créant un contig chimérique.

L'avantage des méthodes d'assemblage par graphe est qu'en considérant toutes les données en même temps, elles peuvent détecter ce genre de conflits et éviter de construire de mauvais assemblages. Une méthode de finition intéressante devrait essayer d'utiliser une approche similaire pour obtenir des résultats optimaux.

E – Erreurs de séquençage

E.1 – Identification et correction des erreurs de séquençage

Il est important de comprendre que la qualité d'un assemblage dépend directement de la qualité des lectures assemblées. Il sera extrêmement difficile d'obtenir un assemblage de qualité si notre séquençage contient un taux d'erreurs élevé. C'est pourquoi plusieurs articles de la littérature portent sur la détection d'erreurs de séquençage (45-54). La marche à suivre idéale serait de corriger ces lectures de façon à faciliter l'assemblage et à exclure de potentielles erreurs dans l'assemblage, plutôt que de les tronquer ou de les exclure complètement. De cette

façon, on peut également minimiser la perte de données potentiellement associée à ces erreurs.

De façon générale, les méthodes de correction des lectures de séquençage supposent que les erreurs de séquençage sont relativement rares (aléatoires), et que par conséquent on retrouvera plus de lectures justes que de lectures aléatoirement erronées. Bien qu'il existe plusieurs approches pour la correction de lectures, un grand nombre d'entre elles utilisent l'abondance de k-mères, soit directement (46, 51, 53), soit par l'entremise d'arbres à suffixes ou d'alignements multiples (47, 54). Ces outils vont identifier les k-mères rares et considérer qu'ils sont le résultat d'erreurs de séquençage. Ces séquences seront ensuite corrigées en utilisant la séquence du k-mère juste le plus similaire.

Cependant, ces approches pourraient introduire des erreurs si la couverture de la molécule séquencée n'est pas uniforme ou si la procédure n'utilise pas de seuil de couverture approprié à ces données. En effet, si nous essayions de séquencer un génome possédant plusieurs régions difficiles à séquencer ou si la fragmentation du génome avant le séquençage est produite par une méthode biaisée (comme la sonication par exemple), la répartition des lectures ne serait pas uniforme sur tout le génome. De cette façon, il serait possible d'identifier un k-mère rare comme étant erroné, et donc de modifier sa séquence en voulant le corriger, alors qu'il appartient simplement à une région possédant une faible couverture.

Plus récemment, on voit apparaître des approches différentes, par exemple BayesHammer (48) compris dans l'assembleur SPADES (55). Cette méthode consiste à utiliser le *clustering* Bayésien pour définir une liste de k-mères sûrs qui seront utilisés pour corriger les k-mères divergents (48). BayesHammer fonctionne bien même avec les jeux de données dont la couverture n'est pas uniforme (48).

La grande limite de toutes ces méthodes est qu'il faut quand même que plusieurs lectures justes soient présentes dans les jeux de données pour faire une correction adéquate d'une erreur donnée. C'est pourquoi, les erreurs de séquençage systématiques restent plus difficiles à corriger. L'identification des lectures justes constitue donc un défi de taille.

La présence d'erreurs systématiques au niveau du compte des homopolymères dans le protocole 454 et la longueur maximale plutôt réduite des lectures 454 (500 pb) expliquent d'ailleurs la perte de popularité du séquençage 454 comparativement à des méthodes comme Illumina (11, 14, 21).

E.2 - Erreurs de séquençage en paires

Les protocoles de séquençage de lectures pairées peuvent, eux aussi, engendrer des erreurs qui compliqueront l'assemblage, comme la production de lectures chimériques (56). La production de paires dont la taille d'insert ne correspond pas à la distance attendue est également possible, ce qui peut entraîner une confusion durant la phase de *scaffolding* (56).

F - Application pratique du projet : StachEndo

En étant conscient des limitations des différentes méthodes de séquençage et des erreurs pouvant en découler, ainsi que des différents types d'assembleurs, nous espérons pouvoir obtenir une compréhension suffisante du domaine de l'assemblage *de novo* de génomes pour développer une marche à suivre quasi-automatique, permettant l'assemblage *de novo* de génomes bactériens et eucaryotes de taille modeste (jusqu'à ~100 MB, comme la plupart des champignons).

Or, notre laboratoire a détecté la présence d'une bactérie endosymbiote dans l'amibe terrestre *Stachyamoeba lipophora,* en plus d'une mitochondrie. Cette bactérie endosymbiote est, à notre connaissance, inconnue. Nous ferons référence à cette bactérie au cours de ce mémoire par le nom StachEndo (***Stach****yamoeba lipophora* **endo**symbiont). Le génome de StachEndo, ainsi que de la mitochondrie de *Stachyamoeba lipophora* furent séquencés en utilisant les méthodes Sanger (seulement pour le génome mitochondrial), 454 et Illumina.

Des analyses réalisées sur des séquences partielles du génome de StachEndo ont révélé de fortes similarités entre ce dernier et les bactéries endosymbiotes Rickettsia. D'autres analyses ont également montrées que contrairement à toutes les espèces connues de Rickettsiales,

StachEndo et son hôte sont co-dépendants et donc qu'aucun des deux ne peut survivre sans l'autre. Cette relation n'est pas sans rappeler le lien qui unit les organismes eucaryotes à leurs organelles, comme les mitochondries et les chloroplastes.

F.1 - Théorie endosymbiotique de l'origine des organelles

Selon la théorie endosymbiotique de l'origine des organelles, il y a environ 1 milliard d'années, des bactéries aérobiques auraient pénétré à l'intérieur d'organismes proto-eucaryotes par phagocytose (57-59). Le passé bactérien de l'organelle expliquerait la présence de son ADN qui n'est pas apparenté à celui de son hôte et la seconde membrane l'entourant serait un vestige de l'invagination de la membrane cellulaire de cet hôte lors de la phagocytose de la bactérie aérobique (58).

Ces bactéries, plutôt que d'être éliminées par leur nouvel hôte proto-eucaryote, développèrent une relation symbiotique mutuellement bénéfique avec leur hôte (60). En devenant un endosymbiote (symbiote vivant à l'intérieur de son hôte), ces bactéries obtiennent la protection de leur hôte et peuvent profiter des nutriments de ce dernier. En échange, la cellule eucaryote acquiert la capacité de générer de l'ATP de façon aérobique, facilitant ainsi sa production d'énergie (60).

Ces bactéries aérobiques, au fils du temps, se seraient transformées, pour devenir les mitochondries avec leur génome hautement réduit que nous connaissons actuellement. De façon analogue, des bactéries photosynthétiques auraient donné naissance aux chloroplastes des cellules végétales (60, 61).

Un des processus qui explique la réduction du génome de ces endosymbiotes en organelles, telles que nous les connaissons, est l'évolution réductive (60). Dans les scénarios où l'hôte et son endosymbiote développent une relation dépendante et la bactérie devient un endosymbiote « obligatoire » (ni l'hôte ni l'endosymbiote ne peut survivre seul), l'endosymbiote vit exclusivement à l'intérieur de son hôte, par conséquent son environnement est particulièrement différent de celui d'une bactérie vivant librement (60, 62).

La redondance de fonctions entre l'hôte et son endosymbiote est un des facteurs importants de cette diminution. Par exemple, puisque l'organisme hôte peut fournir de nombreux métabolites et protéines à son endosymbiote, les gènes qui permettent à la bactérie d'accomplir la biosynthèse de ses composés ne sont plus essentiels à sa survie (62). De cette façon, la pression sélective empêchant la fixation de mutation sur ces gènes de façon à conserver leur intégrité est grandement diminuée (60, 62). Ultimement, ces gènes, étant devenus non-fonctionnels, sont purgés du génome (62). Des événements de recombinaison à l'intérieur du génome de l'endosymbiote peuvent aussi prendre place, menant au réarrangement ou même à la délétion de régions sans fonction du génome (62).

La bactérie endosymbiote vit relativement isolée à l'intérieur de son hôte, elle n'a donc pas l'opportunité de corriger ce genre de pertes lors d'événements de reproduction sexuée ou par transformation bactérienne. Les bactéries endosymbiotes obligatoires fixent donc ces mutations plus rapidement que les bactéries vivant librement (60, 62).

La perte des gènes associés à la biosynthèse des nutriments est souvent observée chez les bactéries endosymbiotes, laissant penser qu'il s'agit d'une des premières phases de pertes lors de la réduction du génome chez les endosymbiotes obligatoires (60, 62, 63). De son côté, l'organisme hôte peut également perdre certains de ces gènes si les fonctions qui leurs sont associées sont prises en charge par l'endosymbiote. Ces transformations ont pour effet de renforcer progressivement la co-dépendance qui existe entre l'hôte et son endosymbiote (60).

F.2 - Phylogénie des endosymbiotes et des mitochondries

Avec le développement des méthodes d'analyse bio-informatiques, il a été possible de renforcer la théorie endosymbiotique. En effet, plusieurs études utilisant des séquences génomiques, des séquences de protéines encodées par la mitochondrie, ou des séquences d'ARNr, ont montré par inférence d'arbres phylogénétiques qu'il existait un lien de parenté entre les mitochondries et la grande classe des Alpha-Protéobactéries, notamment des bactéries endosymbiotes, ainsi que des parasites intracellulaires (61, 64-69) .

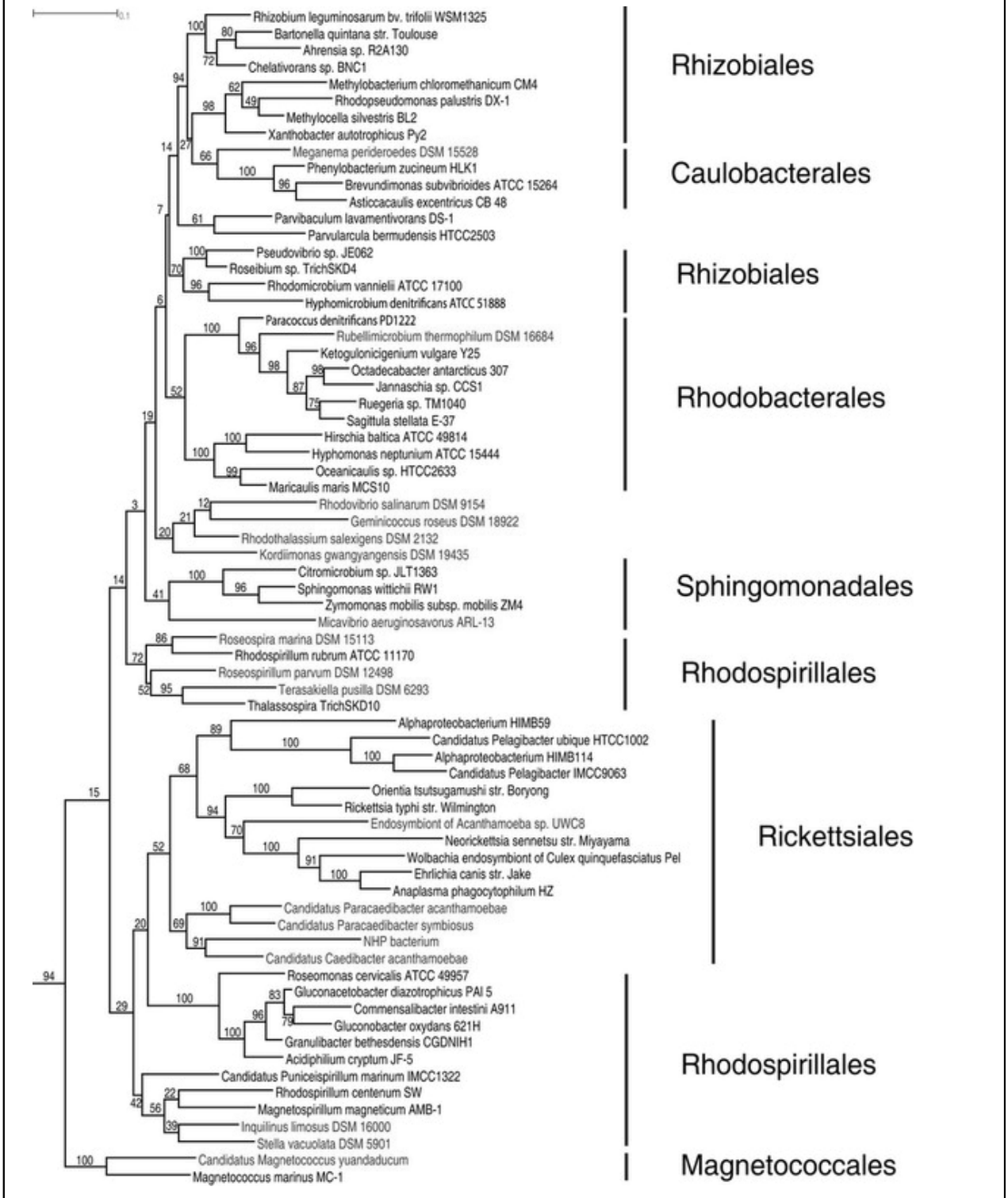


Figure 3. Arbre phylogénétique des Alpha-Protéobactéries, réalisé à partir des séquences des ARN 16S par maximum de vraisemblance. Figure tirée de Wang & Wu (2015) (70).

Le positionnement exact de la proto-mitochondrie ne fait pas consensus (61, 64-68), l'emplacement phylogénétique le plus souvent mentionné dans la littérature se trouve dans l'ordre des Rickettsiales. L'ordre des Rickettsiales est composé exclusivement d'endosymbiotes obligatoires, contrairement aux autres Alpha-Protéobactéries qui vivent librement. Les Rickettsia sont des bactéries Gram-négative menant un mode de vie de parasite endosymbiote obligatoire (70). Plusieurs maladies, dont certaines peuvent affecter l'humain, leurs sont associées (71). En raison de leur mode de vie, les Rickettsia ont connu d'importants événements de réduction de génome sous l'influence de l'évolution réductive, rappelant certaines caractéristiques mitochondriales (62).

F.3 - Caractéristiques génétiques de StachEndo

Le fait que StachEndo ait développé une co-dépendence avec son hôte indique qu'elle pourrait devenir une sorte de future organelle pour *Stachyamoeba lipophora*. Ce constat nous indique qu'au cours de leur vie commune, les deux espèces ont perdu des gènes, sous l'effet de l'évolution réductive, renforçant ainsi l'interdépendance entre ces deux organismes.

Par contre, l'analyse réalisée sur des séquences préliminaires (disponibles au début du projet) de StachEndo a également permis de détecter la présence de divers gènes (dont des gènes flagellaires) qui ne sont habituellement pas associées aux endosymbiotes (72). Ces indices nous amènent à croire que StachEndo est probablement un jeune endosymbiote dont le génome est à un stade relativement précoce de réduction, puisque ces gènes, qui sont a priori obsolètes, sont toujours présents.

L'analyse du génome de StachEndo nous permettrait d'en apprendre plus sur les mécanismes de la transition de la vie libre à la vie d'endosymbiote. L'ajout d'un nouveau génome de Rickettsia, qui, d'après les séquences partielles obtenues, serait potentiellement moins divergent, pourrait également aider à consolider certaines phylogénies dont celles s'attardant à l'organisation de l'ordre des Rickettsiales (ayant connu un influx de nouvelles espèces candidates) et également à l'emplacement de l'origine des mitochondries.

C'est pour toutes ces raisons que nous chercherons à assembler complètement le génome de StachEndo à l'aide des méthodes que nous aurons développées.

En résumé, nous avons fait séquencer le génome de StachEndo, ainsi que le génome mitochondrial de *Stachyamoeba lipophora* avec les technologies 454 et Illumina. Nous avons donc eu accès à de longues et de courtes lectures. De plus le séquençage Illumina était de type Mate-pair et a produit des lectures pairées. Dans le but de trouver une technique d'assemblage efficace, nous avons comparé divers algorithmes publiques d'assemblage de génome *de novo* appartenant aux deux grandes familles d'assembleurs actuels (OLC et graphes DeBruijn), ainsi que plusieurs outils publiques de finitions de génome. Plus précisément, nous avons évalué plus en détails les assembleurs suivants:

- Mira (24)
- Newbler (8)
- Celera (23)
- Velvet (27)
- SOAP DeNovo (31)

et les outils de finitions suivants:

- SSPACE (37)/GapFiller (41)
- Minimus2 (73)
- CONSED (74)

En nous inspirant des approches d'assemblages de transcriptomes utilisant de multiples valeurs de k-mères simultanément, nous avons développé un algorithme de finition de génomes appelé "addition préférentielle de k-mères". Cette méthode permet une combinaison dirigée de plusieurs assemblages préexistants d'un même jeu de données. Cette idée, que nous avons d'abord essayée de recrée à partir d'outils de finitions publiés, fut implémentée sous forme d'un script Perl réutilisant uniquement l'aligneur de séquences Nucmer. Les détails algorithmiques de cette nouvelle méthode sont exposés dans ce mémoire. Nous proposons également une approche utilisant l'assembleur Velvet permettant l'assemblage de lectures pairées provenant de multiples librairies possédant des tailles d'inserts variées.

Toutes ces expériences furent menées dans le but d'obtenir une méthode automatisée (pour limiter le risque d'introduire des erreurs à cause de manipulations humaines) nous permettant d'assembler le génome de StachEndo, l'endosymbiote bactérien inconnu de *Stachyamoeba lipophora*, mais assez versatile pour être utilisée avec différents types de génomes.

2- ASSEMBLAGES ET PROPRIÉTÉS DES K-MÈRES

Le projet de ce mémoire peut être divisé en deux grands thèmes. Premièrement, les travaux en lien avec l'assemblage de génomes et, deuxièmement, l'analyse du génome nouvellement assemblé de StachEndo.

A - Données utilisées

Quatre jeux de lectures du génome de StachEndo et du génome mitochondrial de *Stachyamoeba lipophora* furent utilisés pour ce projet. Tout d'abord, un séquençage Sanger du génome mitochondrial, suivi de son assemblage fut réalisé. Ces données seront utilisées comme référence pour évaluer la qualité des assemblages mitochondriaux.
Pour les données de nouvelle-génération, un séquençage fut réalisé selon le protocole 454, ce qui généra 805 532 lectures d'une taille moyenne de 370 pb. Dans un deuxième temps, un premier séquençage HiSEQ Illumina *paired-end* fut réalisé selon le protocole Mate-Pair, produisant 72 812 370 lectures, ou 36 406 185 paires de lectures de 108 pb. Le troisième jeu de données correspond à un deuxième séquençage Illumina *paired-end* (MiSEQ cette fois-ci). Ce deuxième séquençage a produit 14 660 224 lectures du type pairées chevauchantes d'une longueur moyenne de 250 pb. Ce troisième jeu ne fut utilisé que lors de la finition du génome.

B – Méthodes d'assemblage

Dans cette partie du mémoire, nous nous intéresserons aux diverses méthodes d'assemblage de génome. Pour ce projet, nous avons cherché à comparer plusieurs assembleurs de façon à trouver lequel, ou la combinaison desquels, serait le plus efficace pour assembler des génomes bactériens. Nous avons décidé d'utiliser des assembleurs facilement disponibles (préférablement dans le domaine public), plutôt que d'en créer un nouveau de toute pièce, car il existe un nombre impressionnant de logiciels publiés qui font déjà bien le travail (actuellement, on peut facilement recenser près d'une cinquantaine d'assembleurs). Puisque tous les assembleurs présentent des défauts et engendrent différents types d'erreurs, et qu'il nous semble difficile de faire mieux en créant notre propre assembleur, il nous apparaissait

donc plus pertinent de concentrer notre développement sur la finition des génomes pour contourner ces limitations.

B.1 – Sélection de logiciels d'assemblage de base

Nous avons donc sélectionné 5 assembleurs aux propriétés différentes en nous basant sur les noms les plus souvent retrouvés dans la littérature. Nous nous sommes assurés d'avoir des assembleurs appartenant aux deux grandes familles d'assembleurs. Ces assembleurs peuvent tous produire des assemblages hybrides, c'est-à-dire un assemblage produit à partir de lectures de plus d'une technologie de séquençage.

Tableau I – Logiciel d'assemblage *de novo* de séquences et leurs propriétés

Assembleurs	Mira	Celera	Newbler	Velvet	SOAP DeNovo
Algorithme	OLC	OLC	OLC	De Bruijn	DeBruijn
Type de séquençage	Sanger, 454, Illumina, ABI SOLiD PacBio, ...	Sanger, 454, Illumina, PacBio	Sanger, 454	Illumina, ABI SOLiD	Illumina, ABI SOLiD
Types de séquences	L, C, PE	L, C, PE	L, PE	C, PE	C, PE
Assemblages hybrides	Oui	Oui	Oui	Oui	Oui

Légende : L = longues lectures, C= courtes lectures, PE = paired-end

Dans la famille des assembleurs OLC, nous avons sélectionné Newbler, Celera et MIRA et dans celle des assembleurs par graphe DeBruijn, Velvet et SOAP DeNovo. Ces deux derniers ont été conçus pour l'assemblage de courtes lectures, mais peuvent utiliser des lectures plus longues en échange d'utilisation de mémoire supérieure (en utilisant plus de mémoire).

D'autres assembleurs ont été également testés, mais écartés des étapes suivantes du projet. Mentionnons entre autres, ABYSS et Ray (75), dont les résultats préliminaires étaient très similaires à ceux de Velvet et SOAP DeNovo, et ALLPATHS-LG (76) qui n'était pas tout à

fait adapté au type de données dont nous disposions.

B.2 – Identification d'une méthode de comparaison

Maintenant que nous avons sélectionné cinq assembleurs, il faut trouver une façon de comparer leurs assemblages. Pour ce faire, on doit comparer le résultat de l'assemblage d'une même molécule d'ADN dont la séquence est connue de façon à identifier le nombre d'erreurs présent dans chaque assemblage. Idéalement, la molécule d'ADN avec laquelle nous conduirons ces tests devrait posséder certains points communs avec une bactérie endosymbiote, puisque le but ultime reste d'étudier des bactéries de ce type.

Sachant que le cytoplasme de l'amibe *Stachyamoeba lipophora* (obtenu en brisant sa membrane nucléaire) contient l'ADN de son noyau, de ses mitochondries, de StachEndo et potentiellement des débris provenant des bactéries composant sa diète, les lectures produites par les séquençages de StachEndo contiennent donc, elles-aussi, tous ces génomes. Bien que le noyau soit facile à extraire de ce mélange, il n'est pas aussi facile de séparer StachEndo des mitochondries puisque les deux possèdent des caractéristiques semblables (gènes et pourcentage de nucléotides A+T).

Heureusement, la séquence du génome mitochondriale de *Stachyamoeba* avait déjà été déterminée dans notre laboratoire grâce au séquençage Sanger. Le séquençage Sanger, bien que beaucoup plus coûteux et long à réaliser, produit des lectures longues en faisant très peu d'erreurs. Ces lectures avaient été assemblées avec la série d'outils Phred/Phrap/Consed (74, 77, 78) sans difficulté, donnant **un seul contig circulaire de 49 685 pb**. En raison de la méthode utilisée pour obtenir cette séquence, nous considérons que cette séquence du génome mitochondrial de *Stachyamoeba lipophora* est exacte et pourrait être utilisée comme standard de comparaison des méthodes d'assemblage. De plus, plusieurs analyses ont été menées sur ce génome. Elles révèlent qu'il contient tous les gènes mitochondriaux attendus et qu'aucun d'eux ne possède de changement de cadre de lecture. Nous considérons donc que la séquence de ce petit génome est assez fiable pour nous servir de génome de référence.

En alignant toutes les lectures de séquençage sur ce génome de référence, on a extrait les lectures provenant du séquençage de la mitochondrie. Au final, 32 635 sur 805 532 lectures 454 et 130 968 sur 72 812 370 lectures Illumina Mate-Pair appartiennent à la mitochondrie.

B.3 – Comparaison des logiciels d'assemblage

Tableau II – Comparaison de différents protocoles d'assemblage de séquence sur le génome mitochondrial de *Stachyamoeba lipophora*

Assemblage	454Mira [1]	454 Celera [2]	454 Newbler [3]	Mira (Illumina Velvet) [4]	Mira (Illumina SOAP DeNovo) [5]	454 backbone + Illumina (Mira) [6]	(454 Mira + 454 Newbler +Illumina SOAP DeNovo) Mira [7]
# reads	33 374	33 374	33 374	130 986	130 986	164 360	197 734
Taille (pb)	52 215	50 075	49 709	49 105	48 889	49 174	51 483
Insertions	4	1	0	0	0	1	1
Délétions	0	2	0	0	0	0	0
Substitutions	2	0	0	0	0	2	0
Homopolymères (uniques)	67(43)	130 (99)	78 (56)	12 (12)	11(11)	20 (18)	70 (53)

Légende : - Homopolymères = nombre d'erreurs attribuables aux erreurs d'homopolymères
- uniques = nombre d'homopolymères d'une longueur erronée
- Insertion/délétion = nucléotide de plus/de moins à une position précise par rapport à la séquence de référence

B.3.1 – Newbler, Celera et Mira sur les lectures 454

On a procédé à un assemblage des lectures 454 avec les assembleurs Newbler, Celera et Mira.

Dans les trois cas, l'assemblage résultant, formant un seul contig, est plus grand que le génome mitochondrial attendu. Cette observation s'explique par le fait que l'on retrouve une zone répétée aux extrémités du contig. On peut expliquer cette répétition par le caractère circulaire du génome mitochondrial (vérifié par alignement contre référence). En conclusion, le génome mitochondrial forme un seul contig.

On remarque que lors des assemblages Mira [1] et Celera [2] seulement quelques positions de l'assemblage furent soient insérées ou perdues (causé par le taux d'insertion/délétion de nucléotide plus élevé du séquençage 454). Par contre, on remarque, dans les trois assemblages utilisant uniquement les lectures 454, un nombre très important d'erreurs au niveau du compte des homopolymères. Cette constatation était attendue puisque la détermination du nombre de nucléotides dans un homopolymère est la principale source d'erreurs lors du séquençage 454. Les lectures 454 étant systématiquement erronées, il est donc attendu que leur assemblage contienne des erreurs à ces positions également. On remarque que l'assembleur Celera est particulièrement mauvais pour estimer le nombre correct de nucléotides dans un homopolymère, générant un nombre d'erreurs presque deux fois plus grand que Mira et Newbler.

B.3.2 – Velvet, SOAP DeNovo et Mira sur les lectures Illumina

Nous avons ensuite procédé à un assemblage des lectures Illumina avec les assembleurs Velvet et SOAP DeNovo. Cependant, l'utilisation de ces outils ne nous a pas permis d'obtenir le génome mitochondrial de *Stachyamoeba lipophora* en un seul contig.

C'est pourquoi nous avons utilisé la dizaine de contigs produits par les assemblages Velvet et SOAP DeNovo et les avons donnés en entrée à Mira en les faisant passer pour de longues lectures de types Illumina. Pour contourner des exigences de Mira, on a attribué une qualité uniforme de 30 à toutes les positions des contigs.

Suite à ces modifications, Mira réussit finalement à concilier ces contigs de façon à n'en produire qu'un seul, d'abord à partir des contigs de Velvet et ensuite à partir de ceux de SOAP

DeNovo. Ce faisant, on a pu obtenir les assemblages [4] et [5] du **Tableau II**. On remarque que ces deux assemblages sont plus courts. En effet, ils ne couvrent pas complètement le génome mitochondrial de *Stachyamoeba lipophora.* Une région d'environ 500 pb est absente des deux assemblages. Cette région est très riche en homopolymères T et A. Un réalignement des lectures Illumina mitochondriales que nous avons extraites pour faire l'assemblage nous indique que la couverture dans cette région est beaucoup plus faible qu'ailleurs dans le génome, ce qui expliquerait nos difficultés d'assemblage (11-13). La faible couverture pourrait être interprétée comme un signe de la présence d'erreurs de séquençage. Un nombre important de lectures associées à cette région serait donc exclue de l'assemblage. Cependant, il est très important de remarquer la très faible quantité d'erreurs associée à ces deux méthodes. On ne dénote qu'un petit nombre d'erreur d'homopolymères.

B.3.3 – Assemblages hybrides

Nous avons ensuite essayé de produire des assemblages hybrides. Un assemblage hybride consiste à combiner des lectures provenant de différentes technologies de séquençage (10). Notre supposition initiale était de combiner nos données 454 et Illumina de façon à tirer profit des avantages des deux, tout en minimisant leurs inconvénients respectifs.

La première approche utilisée [7] était plutôt naïve. Elle consistait simplement à combiner les contigs de trois assemblages précédents (Illumina avec SOAP DeNovo et 454 avec Newbler et Mira) à l'aide de Mira. L'assemblage résultant ne fut pas concluant.

Nous avons ensuite essayé une approche plus structurée [6]. Puisque les données Illumina semblent contenir moins d'erreurs que les 454 qui sont plus longues, nous avons décidé d'utiliser l'option « *mapping/backbone* » de Mira (24). Cette option permet de définir des lectures qui serviront de carcasse (*backbone*) sur laquelle les autres lectures seront alignées (*mapping*) (24). De cette façon, les données 454 sont utilisées comme squelette, puisqu'elles s'assemblent facilement pour former le chromosome mitochondrial complet. Ensuite, les données Illumina plus précises servent à corriger l'assemblage.

L'assemblage résultant a produit le résultat espéré. On a obtenu un assemblage couvrant tout le génome mitochondrial et possédant un nombre d'erreurs bien inférieur à celui des assemblages n'utilisant que des données 454. Par contre, il nécessite de faire séquencer un génome plusieurs fois. Dans certaines situations, il peut être difficile d'extraire suffisamment d'ADN pour plusieurs séquençages. Le coût de multiples séquençages peut également être prohibitif.

Nous avons également essayé d'assembler nos données Illumina à l'aide des assembleurs OLC, ainsi que nos données 454 à l'aide d'assembleurs par graphes DeBruijn. Aucun de ces assemblages n'a donné de résultats pertinents.

La principale observation tirée de ces résultats est que les lectures Illumina, bien que plus difficiles à assembler que les lectures 454 qui sont plus longues, produisent des assemblages de meilleure qualité. Seuls les assembleurs de type graphe DeBruijn ont permis d'assembler nos données Illumina. Cependant, ces assemblages sont incomplets. Il est donc nécessaire de trouver des façons efficaces de combiner les contigs produits, et ainsi combler les trous présents dans ces assemblages. L'utilisation d'autres assembleurs comme Mira semble une piste de solution.

Toutefois, avant de commencer à imaginer de nombreuses techniques de finition de génome, il est essentiel de bien comprendre les limites des assembleurs DBG.

B.4 – Limites des graphes DeBruijn et propriétés de l'assemblage

Puisque nous avons décidé d'approfondir la piste des assembleurs de type graphe DeBruijn, il est nécessaire de bien comprendre comment fonctionnent ces derniers et particulièrement comment choisir la taille de k-mère optimale pour un assemblage.

Un k-mère est une sous-séquence d'une taille donnée *k*. Dans les assembleurs DBG, on commence par observer tous les k-mères existant dans les jeux de données. Ces k-mères sont ensuite reliés les uns aux autres pour former le graphe de Bruijn duquel l'assemblage serait inféré (10).

Théoriquement, plus un k-mère est petit, moins il sera rare dans les jeux de données. Cela aura pour effet de créer un graphe très complexe contenant de nombreuses arêtes et donc difficile à résoudre (27-29). Inversement plus un k-mère est grand, plus il est unique, ce qui réduira considérablement le nombre d'arêtes du graphe. Également, des k-mères plus longs permettent de résoudre des répétitions plus grandes (27-29). Cependant, la diminution du nombre d'arêtes peut également rendre le graphe plus vulnérable aux k-mères provenant de lectures erronées (27-29).

Dans l'absolu, un k-mère doit toujours être impair (pour éviter d'obtenir une séquence palindromique ambiguë) et plus petit que la taille de lectures.

Le but de cette expérience était d'observer comment la variation de la taille de k-mère influence les assemblages produits par les assembleurs de Bruijn. Velvet a été utilisé pour ces tests plutôt que SOAP DeNovo car Velvet ne possède pas de limite sur la taille maximale de ses k-mères, alors que SOAP DeNovo, pour des raisons de simplification et d'économie de mémoire dans le code, ne peut dépasser 63 pb (31).

On a utilisé quatre jeux de données relativement différents : le génome mitochondrial de *Stachyamoeba lipophora*, utilisé précédemment comme génome gabarit pour l'évaluation des assembleurs, le génome de la bactérie d'intérêt StachEndo, les génomes nucléaires et mitochondriaux de *Scutellospora heterogama* (un champignon mycorhize vivant une relation symbiotique avec des racines de plantes) (données non-publiées) et le transcriptome de cellules musculaires squelettiques de *Cyclorana alboguttata (79)*, une espèce de grenouille originaire de l'Australie. On a observé les effets de différentes tailles de k-mère dans un intervalle allant d'environ la moitié jusqu'à un peu plus court que la longueur moyenne des lectures de séquençage.

B.4.1 – Application au premier jeu (*Stachyamoeba lipophora* mitochondrial)

On observera dans cette section les effets des différents k-mères sur la variation de la taille

totale de l'assemblage, du nombre de contigs, de la taille moyenne de ces contigs et de la couverture médiane de l'assemblage.

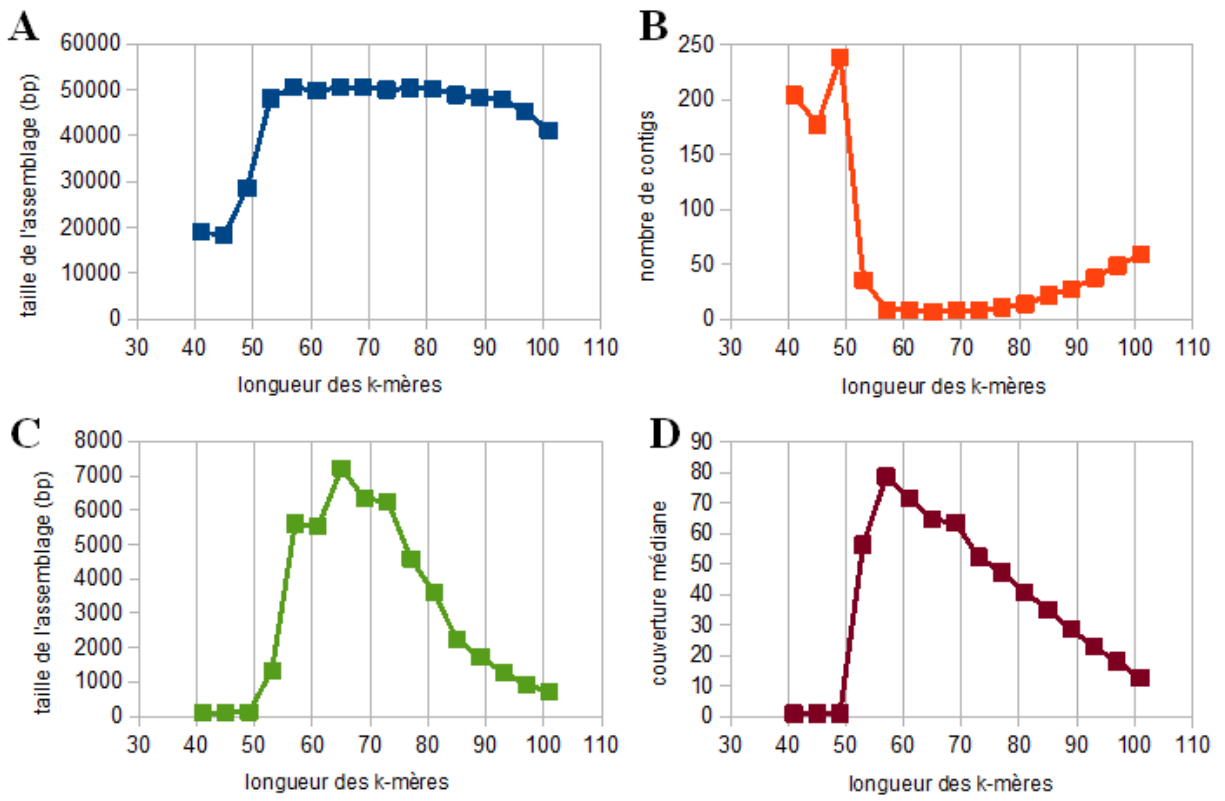


Figure 4. Impact de la variation de la longueur des k-mères sur l'assemblage du génome mitochondrial de *Stachyamoeba lipophora*. (**A**) Variation de la taille de l'assemblage (pb). (**B**) Variation sur le nombre de contigs. (**C**) Variation de la taille moyenne des contigs (pb). (**D**) Variation de la couverture médiane des contigs de l'assemblage.

Ainsi, on peut mieux comprendre l'effet de la taille des k-mères. Cette combinaison de caractéristiques nous permet d'observer en détail l'évolution de l'assemblage. Par exemple, une diminution de la taille de l'assemblage associée à la diminution du nombre de contigs et

l'augmentation de la couverture médiane nous indique que la taille de l'assemblage diminue, car l'assemblage génère moins de contigs contenant en général plus de lectures. L'assemblage devient donc meilleur. Inversement, si nous observions une diminution de la taille de l'assemblage associée à la diminution du nombre de contigs et la diminution de la couverture médiane, cela signifierait que nous perdons des contigs et que le nombre de lectures utilisées dans les contigs restant diminue. L'assemblage devient donc moins bon.

La **Figure 4** montre l'effet des différentes tailles de k-mères (de 41 à 101 par intervalle de 4) sur l'assemblage du génome mitochondrial de *Stachyamoeba lipophora*. La sous-**figure 4.A** révèle qu'initialement la taille de l'assemblage est assez faible et qu'elle augmente avec la taille de k-mère. Elle atteint un plateau entre les tailles de k-mère 53 à 81, avant de diminuer tranquillement. La **Figure 4.B** montre que de façon générale le nombre de contigs diminue avec l'augmentation de la taille des k-mères jusqu'à l'atteinte d'un palier entre les k-mères 57 et 77, pour ensuite augmenter une fois ce palier dépassé. Sur la **Figure 4.C** on observe que la taille moyenne des contigs augmente progressivement avec la taille des k-mères avant d'atteindre un maximum au k-mère 73, puis redescend. La même observation peut être faite à la **Figure 4.D** en observant la couverture médiane des contigs augmenter, puis diminuer après le k-mère 73, avec la taille des k-mères.

En résumé, on remarque, avec ces quatre figures, que les lectures semblent mieux s'assembler au fur et à mesure que la taille des k-mers augmente et ce jusqu'à l'atteinte de certaines valeurs, que nous décrirons ici comme optimales, à partir desquelles l'assemblage diminue tranquillement en efficacité.

B.4.2 – Application au deuxième jeu (StachEndo)

Avec des variations de taille de k-mères identiques à celles du jeu précédent, il est possible d'observer un phénomène similaire lors de l'assemblage du génome de StachEndo à la **Figure 5**. Dans la **Figure 5.A**, on remarque que la taille de l'assemblage semble atteindre un certain plateau autour du k-mère 67. Sur la sous-**figure 5.B**, on voit le nombre de contigs diminuer avec l'augmentation de la longueur des k-mères. Le nombre de contigs présents dans

l'assemblage atteint un plateau entre les valeurs de k-mère 73 et 91. Une augmentation du nombre de contigs semble se produire après le k-mère 91 (non visible à cause de l'échelle) alors que le nombre de contigs passe de 55 à 72. Les sous-**figures 5.C et 5.D** montrent que la taille moyenne des contigs et leur couverture augmentent avec la taille des k-mères utilisés.

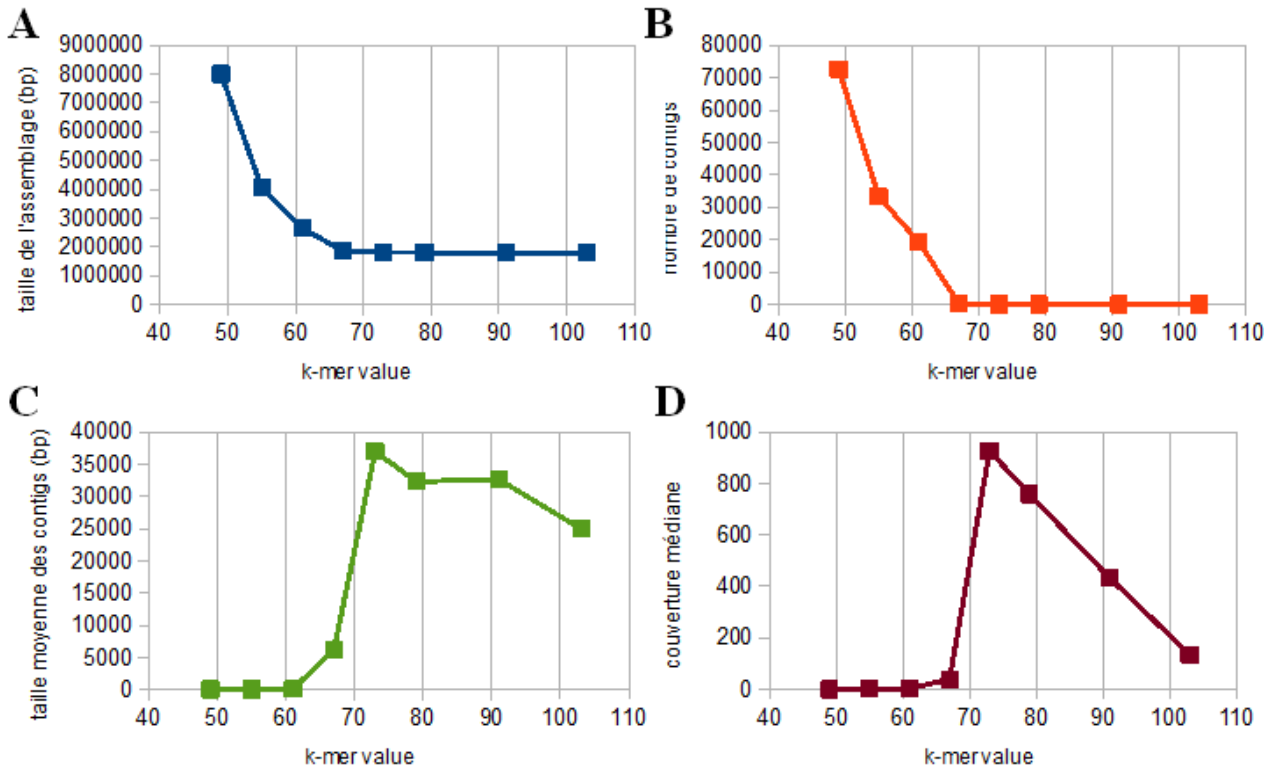


Figure 5. Impact de la variation de la longueur des k-mères sur l'assemblage du génome de la bactérie endosymbiote StachEndo. (**A**) Variation de la taille de l'assemblage (pb). (**B**) Variation sur le nombre de contigs. (**C**) Variation de la taille moyenne des contigs (pb). (**D**) Variation de la couverture médiane des contigs de l'assemblage.

De la même façon qu'avec le génome mitochondrial de *Stachyamoeba lipophora*, on remarque que l'augmentation de la taille des k-mères du graphe DeBruijn semble améliorer les résultats de l'assemblage jusqu'à l'atteinte d'une valeur limite (ici environ 73bp) à partir de laquelle l'assemblage se dégrade. On observe aussi la présence d'un plateau pour le nombre de contigs.

B.4.3 – Application au troisième jeu (*Scutellospora heterogama*)

Avec le jeu de données provenant de *Scutellospora heterogama*, on observe des comportements différents de nos variables. Au lieu d'observer deux régions distinctes mais continues, l'une où l'assemblage semble s'améliorer et l'autre où il semble se dégrader, on voit plutôt un dédoublement de ces régions.

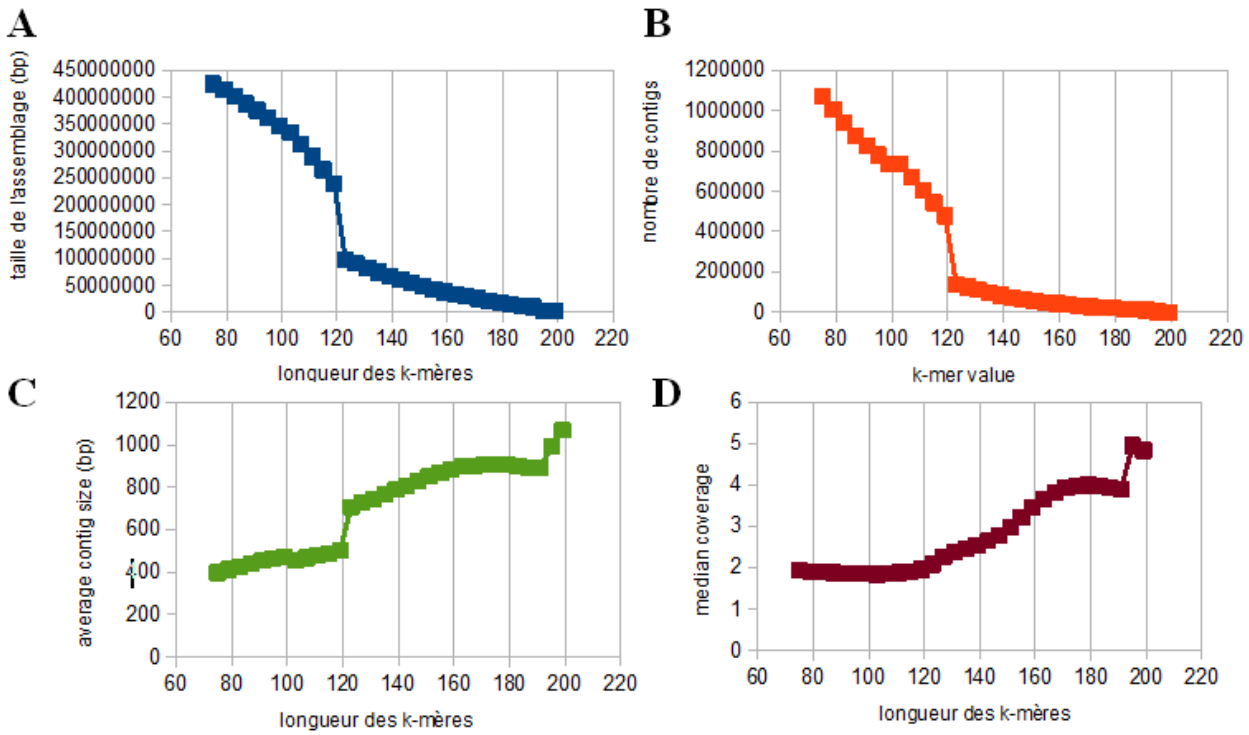


Figure 6. Impact de la variation de la longueur des k-mères sur l'assemblage du génome de *Scutellospora heterogama.* (**A**) Variation de la taille de l'assemblage (pb). (**B**) Variation sur le nombre de contigs. (**C**) Variation de la taille moyenne des contigs (pb). (**D**) Variation de la couverture médiane des contigs de l'assemblage.

Ces observations sont liées à la présence d'un « contaminant » dans notre jeu de données. En effet, les lectures de séquençage générées ne couvrent pas seulement le génome nucléaire de *S. heterogama*, mais également son génome mitochondrial. On est donc en présence de deux assemblages en parallèle dans la même expérience. Puisque le génome mitochondrial est plus petit que le génome nucléaire et qu'il est plus couvert que le génome nucléaire (une cellule possède plusieurs mitochondries), il devrait être plus facile à assembler. De plus, puisque le génome nucléaire est plus long et qu'il possède possiblement plus de régions répétées, son assemblage est plus efficace avec de plus grandes valeurs de k-mers.

Selon nous, la première moitié des courbes correspondrait à l'assemblage du génome mitochondrial et la deuxième moitié au génome nucléaire de *Scutellospora heterogama*.

Cette hypothèse semble confirmer par le fait que la grande dénivellation entre les deux régions de nos courbes correspond sensiblement à la région où l'assemblage mitochondrial semble le plus performant. En effet, autour des k-mers 123-127, on retrouve la totalité du génome mitochondrial en 6 contigs. Par la suite, l'assemblage du génome nucléaire devient de plus en plus efficace. Cependant, les lectures mitochondriales et les contigs qu'ils forment ne disparaissent pas pour autant. Bien que leur taille varie un peu, ces contigs continuent d'exister même lorsque la taille de k-mère augmente. La principale différence entre ces contigs à différents k-mères est la profondeur de leur couverture. En utilisant des k-mères de taille 127, les contigs mitochondriaux formés possèdent une couverture d'environ 55 à 60x ce qui est 4 à 5 fois plus important que les contigs nucléaires de taille comparable. Lorsque la taille des k-mères atteint 191, la couverture de ces contigs chute à environ 22 lectures. Les quelques contigs nucléaires de taille comparable (moins de 1% de l'assemblage) possèdent une couverture comparable, largement supérieure à la couverture médiane de l'assemblage (~ 4).

Avec nos trois premiers jeux de données, on remarque tout de même la présence d'une taille de k-mère particulière ou d'un intervalle de taille, que l'on décrira comme point de rupture. Ce point de rupture marque la distinction entre deux comportements de l'assemblage. Plus la taille de k-mère choisie se rapprochera du point de rupture, plus la taille des contigs augmentera et leur couverture également. Une fois le point de rupture dépassé, la taille des contigs diminuera

plus ou moins rapidement. Il en va de même pour leur couverture. La stringence de l'assemblage atteignant un niveau critique, la perte de lectures aura a priori un effet négatif sur l'assemblage. Les contigs perdront progressivement des lectures. On verra la couverture des contigs diminuer jusqu'à ce qu'ils se brisent dans leurs régions de plus faibles couverture et que certains contigs disparaissent totalement.

B.4.4 – Application au quatrième jeu (transcriptome de *Cyclorana alboguttata*)

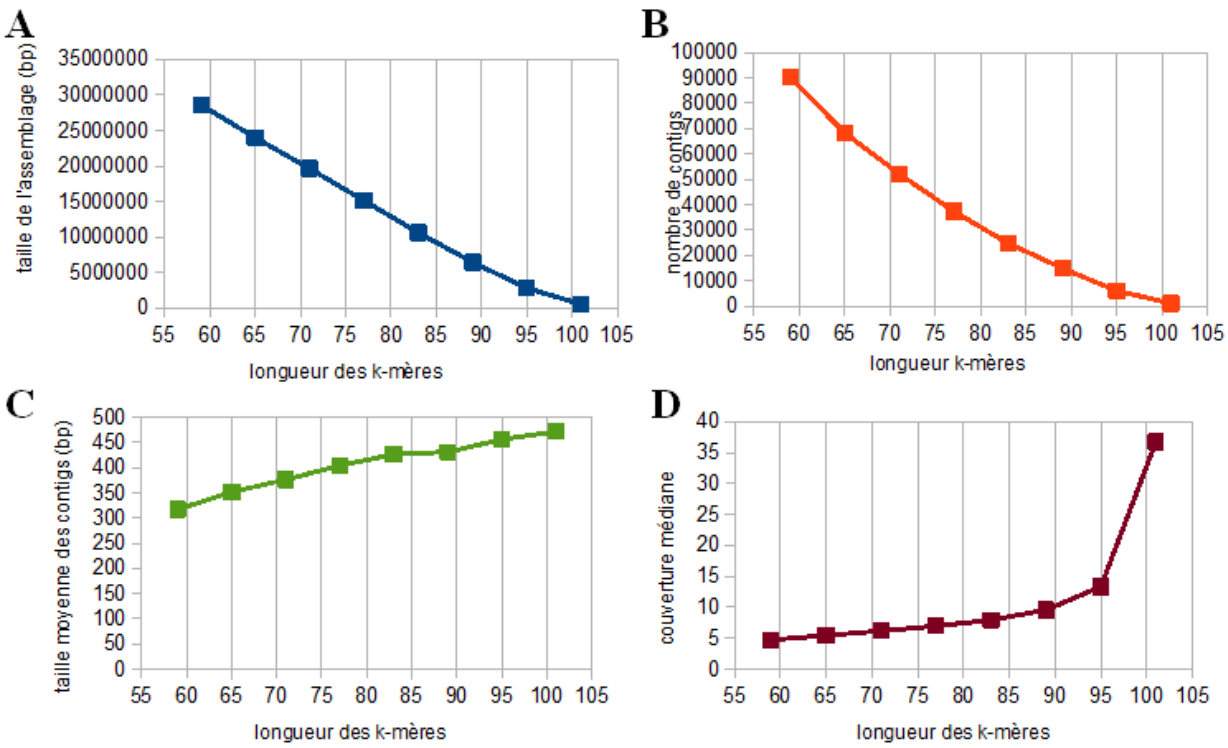


Figure 7. Impact de la variation de la longueur des k-mères sur l'assemblage du transcriptome de cellules musculaires squelettiques de la grenouille *Cyclorana alboguttata.* (**A**) Variation de la taille de l'assemblage (pb). (**B**) Variation sur le nombre de contigs. (**C**) Variation de la taille moyenne des contigs (pb). (**D**) Variation de la couverture médiane des contigs de l'assemblage.

Le quatrième jeu de données est très différent des trois précédents. Il s'agit de lectures provenant du séquençage Illumina du transcriptome de cellules musculaires squelettiques de la grenouille *Cyclorana alboguttata*. La **Figure 7** représente la variation de différentes caractéristiques de l'assemblage selon la taille de k-mère utilisée pour construire le graphe DeBruijn de l'assemblage. Contrairement aux autres jeux de données, on ne remarque pas de pics particuliers ou de paliers sur nos courbes. La taille de l'assemblage et le nombre de contigs semblent constamment diminuer avec l'augmentation de la taille de k-mères, alors que la taille moyenne des contigs et leur couverture augmentent. À aucun moment, on observe l'inversion de ces tendances.

Les assemblages de transcriptome représentent une situation particulière qui s'apparente un peu à ce que nous avons observé avec le jeu de donnée précédent. Le jeu de données de *S. heterogama* était composé de lectures appartenant à au moins deux molécules distinctes, le noyau de *S. heterogama* et son génome mitochondrial. Dans le cas d'un assemblage de transcriptomes, toutes les molécules d'ARN d'un échantillon sont retro-transcrites en ADNc, puis séquencées. Les lectures proviennent de multiples molécules d'ARN transcrites. Nous avons donc plusieurs assemblages en parallèle. Tel qu'observé avec le jeu de données de *S. heterogama,* ainsi qu'à la **Figure 7.D**, l'augmentation de la taille de k-mère favorise l'assemblage des régions plus couvertes. Dans le cadre du séquençage d'ADNc, un ADNc particulier sera plus séquencé que les autres s'il correspond à un ARN présent en plus grande quantité dans les tissus utilisés. Les ARNs qui sont plus exprimés par les tissus étudiés formeront donc des contigs avec une couverture plus grande. L'augmentation de la taille de k-mère, en favorisant la formation de contigs très couverts, peut diminuer la sensibilité avec laquelle nous pouvons détecter des ARNs peu exprimés et dont les contigs sont donc peu couverts. De façon générale, l'assemblage d'un transcrit est plus simple que l'assemblage d'un génome puisqu'il est plus court, mais une expérience de transcriptomique peut difficilement être assemblée avec une seule taille de k-mère sans perdre une partie des transcrits séquencés. Les transcrits peu couverts ne peuvent être assemblés correctement qu'avec des k-mères de petites tailles. Certaines approches, comme OASES (80), essayent de combiner les résultats d'assemblages générés à partir de k-mères plus petits (plus sensibles) et plus grands (plus stringents) pour obtenir des assemblages plus complets.

Dans le cas des assemblages *de novo* de transcriptomes, nous n'avons pas observé de point de rupture. Nous observons cependant la forte résistance des régions de forte couverture aux critères stringents des assemblages avec grand k-mères.

B.4.5 – Lien entre la taille des k-mères et la qualité des résultats

En résumé, nous avons observé les liens unissant la taille de k-mères choisie avec le résultat de l'assemblage par graphe DeBruijn. Le choix de k-mères plus longs permet de résoudre des répétitions de taille comparable. En choisissant un k-mère plus long, il est possible d'assembler de plus longs contigs. De plus, pour qu'une arrête relie deux k-mères dans un graphe DeBruijn, il faut que les k-1 derniers caractères du k-mère de départ soient identiques aux k-1 premiers caractères du k-mère d'arrivée de l'arrête. En raison de cette contrainte, le choix de longs k-mères impose donc une stringence plus importante lors de l'assemblage. Les lectures erronées sont donc exclues de l'assemblage puisqu'elles conduisent à des impasses dans le graphe d'assemblage. On remarque donc la diminution du nombre de lectures utilisées dans les contigs et la disparition des contigs moins couverts, puisqu'ils possèdent moins de lectures utilisables. En contrepartie, l'utilisation de k-mères plus petits permet de capter ces portions moins bien couvertes de l'assemblage. Par contre, la variabilité induite par les lectures erronées et une résolution plus faible pour la résolution des répétitions rendent l'obtention d'assemblages complets plus difficile. Les deux types de k-mères possèdent leurs propres avantages et inconvénients et l'utilisation de k-mères de taille moyenne nous donnent simplement des résultats intermédiaires qui ne sont pas pleinement satisfaisants.

Il serait intéressant d'appliquer une approche utilisant plusieurs tailles de k-mères à l'assemblage de génomes. En produisant différents assemblages à partir de tailles de k-mères différentes, il serait possible de combiner leurs différents contigs de façon à obtenir une séquence plus complète. Les grands k-mères nous permettraient d'obtenir moins de contigs qui seraient plus couverts et plus fiables. Des contigs provenant d'expériences utilisant des k-mères plus petits pourraient ensuite être ajoutés pour combler les trous restant dans l'assemblage.

B.5 – Évaluation de la qualité de l'assemblage

Maintenant que nous avons une meilleure idée du fonctionnement des assembleurs de type graphe DeBruijn, il nous faut trouver une façon d'évaluer la qualité des assemblages. Sans génome de référence fiable, il devient difficile de savoir si un assemblage donné est juste ou non.

La méthode classique d'analyse de la qualité des assemblages *de novo* est le N50. Le N50 est une mesure calculée en additionnant les contigs générés du plus grand au plus petit jusqu'à ce que cette somme atteigne 50 % de la taille totale de l'assemblage (somme de la taille de tous les contigs) (81, 82). Le N50 est la taille du dernier contig ajouté. De cette façon, on sait que 50 % de l'assemblage est composé de contigs d'une longueur plus grande ou égale à celle du N50. En résumé, plus un N50 est grand, meilleur est l'assemblage puisque ses contigs sont longs (81, 82). Certaines variations de ce calcul existent. Le N90 ou N70 utilisent la même logique mais avec des pourcentages de 90 et 70, respectivement (82).

La notion de N50 est imparfaite à plusieurs niveaux. Premièrement, c'est une notion qui ne permet pas de comparer plusieurs assemblages du même génome à partir de différents assembleurs ou de différents jeux de données (82). En effet, certains assembleurs auront tendance, par peur d'exclure des zones de faible couverture, à créer plus de contigs. D'autres assembleurs par graphes DeBruijn produisent des micro-contigs à partir de cul-de-sac dans l'assemblage, et le dédoublement de contigs très similaires dans le cas de génomes hétérozygotes (82, 83). Il est évident que tous ces contigs ne font pas réellement parti de l'assemblage complet visé, mais ils grossiront tout de même la taille de l'assemblage qui est utilisée pour calculer le N50, faussant ainsi la mesure.

Certains groupes de recherche ont donc essayé d'introduire la notion de NG50 qui utilise la taille attendue du génome plutôt que la taille totale de l'assemblage généré. De cette façon, lorsqu'on compare différents assemblages du même génome, une seule taille de référence est considérée, rendant le calcul plus facilement comparable (82). Par contre, cette modification

ne corrige pas la principale imperfection de la notion de N50 (82). La valeur du N50 dépend entièrement de la taille des contigs, mais ne prend pas réellement en compte leur exactitude et donc la qualité de l'assemblage (84). En effet, un long contig complètement erroné provoquera une inflation du N50 (84). D'autres méthodes d'évaluation vont simplement favoriser les assemblages possédant le moins de contigs, ou alternativement les assemblages possédant la meilleure couverture. Mais en ne se fiant qu'à un seul paramètre, il est impossible d'observer adéquatement la dynamique et la qualité de l'assemblage (82, 84).

Il est donc important de trouver une façon plus réaliste d'évaluer la qualité de l'assemblage. La méthode idéale serait de pouvoir annoter le génome pour vérifier que tous les gènes sont complets et qu'entre autres, ils ne contiennent pas de décalage de cadre de lecture. Idéalement, des connaissances sur la synthénie du génome pourraient aussi être utiles.

Pour ce qui est des génomes comme celui de StachEndo, le génome bactérien qui nous intéresse, la présence d'une mitochondrie à l'intérieur de la cellule hôte pourrait se révéler utile. En supposant que, comme pour le jeu de données de *S. heterogama* (**Figure 6**), notre jeu de données contient à la fois le génome qui nous intéresse et un génome connu séquencé en même temps, il serait possible d'utiliser le génome connu comme baromètre de qualité. Tel que nous l'avons observé, il est possible d'assembler des génomes distincts à l'intérieur d'un jeu de données. Notre hypothèse serait que si l'assembleur produit un assemblage fiable du génome de référence, il est probable que les autres contigs appartenant à l'autre génome soient de qualité comparable.

Nous avons besoin d'une méthode de comparaison simple. Nous nous sommes donc inspiré des caractéristiques des assemblages que nous avons observés lors de notre étude des effets de la taille de k-mère sur l'assemblage (voir section **B.4 – Limites des graphes DeBruijn et propriétés de l'assemblage**). Nous avons donc décidé d'observer plusieurs paramètres classiques qui sont faciles à calculer : le N50, la taille de l'assemblage, le nombre de contigs, la distribution des tailles des contigs, la couverture médiane de l'assemblage, la taille des 4 plus grands contigs, ainsi que le nombre de lectures utilisées pour réaliser l'assemblage et le nombre de nucléotides indéterminés (N). Pris individuellement, ces caractéristiques nous

donnent une vision très partielle de l'assemblage. En les combinant, nous pouvons observer la dynamique par laquelle l'assemblage forme les contigs. En utilisant la couverture des contigs et la façon dont l'utilisation de lectures est modulée entre différents assemblages, nous pouvons identifier les assemblages les plus fiables (dont les contigs sont les mieux supportés). En ajoutant les notions de taille de contigs et de taille de l'assemblage, le tout comparé à la taille attendue de l'assemblage, il devient maintenant plus facile d'observer les assemblages contenant les plus grands contigs fiables.

Cependant, l'annotation et l'analyse en taille du contenu génique de l'assemblage reste la meilleure façon d'analyser un assemblage. Il faut donc procéder à cette dernière analyse une fois que l'on a obtenu un assemblage quasi-final.

C – Assemblage du génome de StachEndo et « finishing »

Avec tous les outils en main pour attaquer l'assemblage du génome de la bactérie endosymbiote StachEndo, nous avons d'abord tenté d'obtenir un assemblage préliminaire de bonne qualité, et ensuite essayé diverses méthodes de finition pour obtenir une séquence génomique complète.

C.1 – Assemblage préliminaire

Des assemblages du génome de StachEndo ont été réalisés simultanément aux assemblages mitochondriaux présentés au **Tableau II**. Puisque plusieurs erreurs ont été observées dans les assemblages mitochondriaux, les assemblages de StachEndo conjoints comportent vraisemblablement les mêmes erreurs. Ces expériences ont tout de même permis d'estimer la taille du génome complet de StachEndo (environ 1,8 Mb). L'assemblage de StachEndo correspondant à l'assemblage hybride naïf (assemblage [7] **Tableau II**) était composé de 11 contigs de 740 à 967 364 pb, pour un total d'environ 1,74 Mb. Ces contigs regroupent donc la plupart du génome.

Malgré les erreurs trouvées dans son assemblage mitochondrial (au moins 8 décalages de cadres de lectures en plus des erreurs d'homopolymères), la structure générale du génome de StachEndo devrait être suffisamment conservée pour permettre une annotation préliminaire. Pour ce faire, le logiciel RAST (*Rapid Annotation using Subsystem Technology*) (85) a été utilisé. Ces résultats sont présentés et analysés dans la section **3-A – Annotation préliminaire RAST**.

C.2 – Couverture attendue et seuil de couverture dans les assembleurs DBG

La comparaison des assemblages du génome mitochondrial de *Stachyamoeba lipophora*, avait montré que l'utilisation des données Illumina produit moins d'erreurs. Les lectures courtes de type Illumina produisent de bons résultats lorsque utilisées avec des assembleurs de type graphe DeBruijn. Cependant, malgré le choix de tailles de k-mères optimales, les contigs qui sont produits restent plus courts que les contigs composés de lectures 454. Il est donc important de trouver comment utiliser ces lectures Illumina plus efficacement.

Le meilleur assemblage de StachEndo à partir d'Illumina était composé de 53 contigs (taille totale 1,78Mb). La taille de k-mère optimale retenue pour cet assemblage était de 79. L'assemblage est toujours loin d'être complet.

Pour améliorer l'assemblage, nous avons donc mené certaines expériences en utilisant deux autres paramètres des assembleurs de type graphe DeBruijn que nous avions brièvement mentionnés lors des sections précédentes. Ces paramètres sont la couverture attendue de l'assemblage (*expected coverage*) et le seuil de couverture (*coverage cutoff*) (27, 33). Fournir une valeur de couverture attendu est très utile pour l'assemblage. Les contigs possédants une couverture proche de cette valeur attendue seront considérés comme des contigs « normaux », alors que les contigs possédant une couverture fortement supérieure à cette valeur seront considérés comme appartenant à des répétitions (27, 33). Pour sa part, le seuil de couverture permet de simplifier le graphe DeBruijn de l'assemblage (27, 33). On considère que les nœuds du graphe possédant une couverture inférieure à ce seuil sont erronés. Ils sont donc retirés du graphe, réduisant ainsi le nombre de chemins dans le graphe et simplifiant donc sa résolution.

Si un seuil est trop élevé, il pénalisera des chemins viables qui sont moins couverts ce qui amputera l'assemblage.

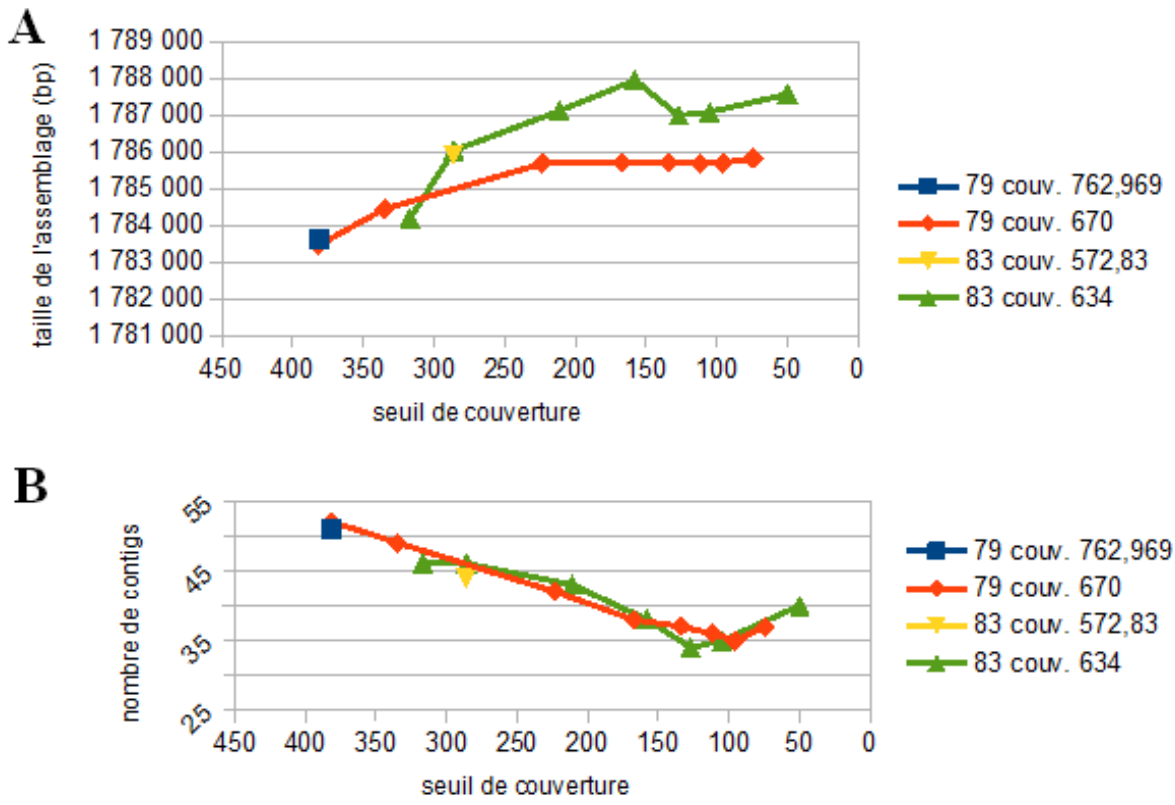


Figure 8. Variation de la taille totale de l'assemblage (**A**) et du nombre de contigs (**B**) en fonction du seuil de couverture de l'assemblage. *On observe quatre assemblages Velvet du génome de StachEndo à partir de données Illumina ; (**bleu**) assemblage avec k-mères d'une longueur de 79 par défaut de Velvet (couverture attendue (762,962) et seuil de couverture (381,485)), (**orange**) assemblages avec k-mères d'une longueur de 79 utilisant une couverture attendue de 670 et un seuil de couverture variable, (**jaune**) assemblage avec k-mères d'une longueur de 83 par défaut de Velvet (couverture attendue (572,83) et seuil de couverture (286,41)) et (**vert**) assemblages avec k-mères d'une longueur de 83 utilisant une couverture attendue de 634 et un seuil de couverture variable.*

Lors de notre assemblage Illumina-Velvet, on a laissé Velvet estimer lui-même les valeurs de couverture à utiliser. La couverture attendue calculée était de 762,97 et le seuil de couverture

était de 381,45. On a ensuite utilisé cet assemblage pour calculer la couverture médiane de cette expérience. On a obtenu une valeur de 670 qui a été utilisée comme nouvelle couverture attendue de l'assemblage. Avec cette dernière valeur de couverture attendue, et la même taille de k-mère (79), on a testé de multiples valeurs de couverture. Le premier essai utilisait le seuil calculé par Velvet soit 381,45. L'assemblage résultant était composé de 52 contigs donc quasi-identique à notre assemblage utilisant les paramètres calculés automatiquement.

Puis, on a utilisé systématiquement différentes fractions de la couverture attendue comme valeur de seuil afin d'explorer l'impact de la variation du seuil de couverture sur l'assemblage résultant. Rappelons que les nœuds situés sous le seuil de couverture sont considérées comme erronées et que la variation du seuil rend donc l'identification des régions erronées plus ou moins stringente. On a utilisé des seuils allant de la moitié jusqu'au dixième de la couverture attendue.

Avec un seuil égal à 50 % de la couverture, on suppose que les nœuds possédant une couverture deux fois plus petite que la couverture attendue sont erronés. Inversement, avec un seuil égal à 10 % de la couverture, on commence à considérer des nœuds comme erronés à partir d'une couverture 10 fois plus petite que la couverture attendue. La **Figure 8** donne un aperçu des différents essais effectués et des résultats obtenus.

L'analyse des résultats pour des k-mères de taille 79 a permis de constater que :
- avec un seuil de 50 % de la couverture attendue (50 % X 670 = 335), il y a moins de contigs (49 contigs) qu'avec les paramètres automatiques de Velvet et la taille totale de l'assemblage et la couverture médiane et le nombre de lectures utilisées augmentent ;
- avec un seuil de 223,3 (1/3 de la couverture attendue), le nombre continue de diminuer (42 contigs) et les trois autres variables augmentent ;
- entre 223,3 et 95,7, la diminution de la valeur du seuil continue à s'accompagner d'une diminution du nombre de contigs mais la taille totale de l'assemblage se stabilise autour de 1,785Mb. ;
- au seuil de 95,7 (1/7 la couverture attendue), on obtient l'assemblage composé du plus petit nombre de contigs (35) ;

- à partir d'un seuil de 74,44 (1/9 de la couverture attendue), l'assemblage se dégrade, c'est-à-dire que le nombre de contigs augmente, puisque les grands contigs commencent à éclater en contigs plus petits et que la taille totale de l'assemblage augmente de façon beaucoup plus rapide.

Les paramètres optimaux de cet assemblage semblent donc être une couverture attendue de 670 et un seuil de couverture de 95.7 (septième de la couverture). Ce qui est loin de la valeur par défaut de Velvet.

On a ensuite répété les essais avec les autres valeurs de k-mères en commençant avec une taille de k-mère de 83 qui a produit notre deuxième meilleur assemblage. Cet assemblage était composé de 44 contigs et Velvet a utilisé une couverture attendue de 572,83 et un seuil de couverture de 286,41. La valeur de couverture médiane calculée pour cet assemblage est de 634. On remarque que la taille médiane de cet assemblage est différente de celle calculée pour l'assemblage réalisé avec des k-mères de taille 79. Cette observation est normale puisque les deux assemblages sont de tailles différentes et sont composés de contigs différents. Ils n'utilisent pas les mêmes lectures et leur couverture est forcément différente.

En utilisant cette couverture médiane comme couverture attendue de l'assemblage, on a fait varier la valeur du seuil de couverture. L'analyse des résultats permet de constater que (voir **Figure 8**) :
- la diminution du seuil s'accompagne d'une diminution du nombre de contigs formés et d'une augmentation de la taille totale de l'assemblage ;
- le seuil optimal pour l'assemblage se situe au cinquième de la couverture attendue (126,8), et produit une taille totale de 1,785Mb composé de 34 contigs ;
- pour des seuils plus petits, le nombre de contigs commence à augmenter et les plus grands contigs commencent à éclater.

À la lumière de ces deux groupes d'expériences, on constate que les valeurs optimales des paramètres de couverture de Velvet semblent être dépendantes de l'assemblage. Les paramètres optimaux d'un assemblage réalisé à une valeur de k-mère donnée ne seront donc

pas forcément les paramètres optimaux d'un autre assemblage réalisé à partir d'une autre valeur de k-mère, même s'il s'agit d'assemblages du même jeu de données. Des observations faites précédemment nous portent à croire qu'il serait avantageux pour la finition d'assemblage d'avoir accès à plusieurs assemblages d'un même jeu de données obtenues à partir de différentes valeurs de k-mères. Toutefois, la production de tous les assemblages nécessaires à l'identification des paramètres optimaux correspondant à chaque valeur de k-mère peut s'avérer longue et coûteuse en espace disque.

Dans le but d'éviter ce problème, deux autres expériences ont été menées. La première consistait à utiliser les paramètres optimaux de l'assemblage du k-mère 79 sur l'assemblage du k-mère 83. L'assemblage résultant est d'une taille de 1,787Mb composé de 36 contigs. Bien que cet assemblage semble un peu moins bon que l'assemblage optimal précédemment identifié, il semble tout de même supérieur à l'assemblage par défaut du k-mère 83. La seconde expérience consistait à utiliser les paramètres optimaux de l'assemblage du k-mère 83 sur l'assemblage du k-mère 79. L'assemblage résultant est d'une taille de 1,784Mb composé de 38 contigs. On remarque que dans les deux expériences, bien que non optimaux les valeurs de couverture attendue que nous utilisons sont assez proches de celles utilisées automatiquement par Velvet, mais que les valeurs de seuil de couverture que nous utilisons sont beaucoup plus petites que les valeurs par défaut.

Considérant que nous désirons disposer de plusieurs assemblages issus du même jeu de données mais générés avec des tailles de k-mères différentes, mais qu'il est très exigeant d'identifier la combinaison optimale de paramètres pour chaque assemblage, nous avons opté pour une solution plus simple. Cette solution consiste dans un premier temps à utiliser les paramètres automatiques (par défaut) de Velvet sur le meilleur assemblage obtenue, c'est-à-dire correspondant à une certaine longueur de k-mère. On fait ensuite le seuil de couverture dans le but d'identifier le seuil optimal, et finalement à utiliser ces paramètres avec chacun des autres k-mères.

Dans le cas de l'assemblage du génome de StachEndo, nous avons donc optimisé les paramètres avec l'assemblage des k-mères de longueur 79. On a obtenu 9 assemblages avec

des tailles de k-mères différentes en utilisant 670 comme valeur de couverture attendue de l'assemblage, et 95,7 comme seuil de couverture. Avec ces réglages, les assemblages obtenus du génome mitochondrial de *Stachyamoeba lipophora* comprennent deux contigs. Les contigs résultants de ces assemblages seront utilisés pour faire la finition du génome (voir **Figure 9** pour un aperçu des assemblages qui seront utilisés pour la finition du génome de StachEndo).

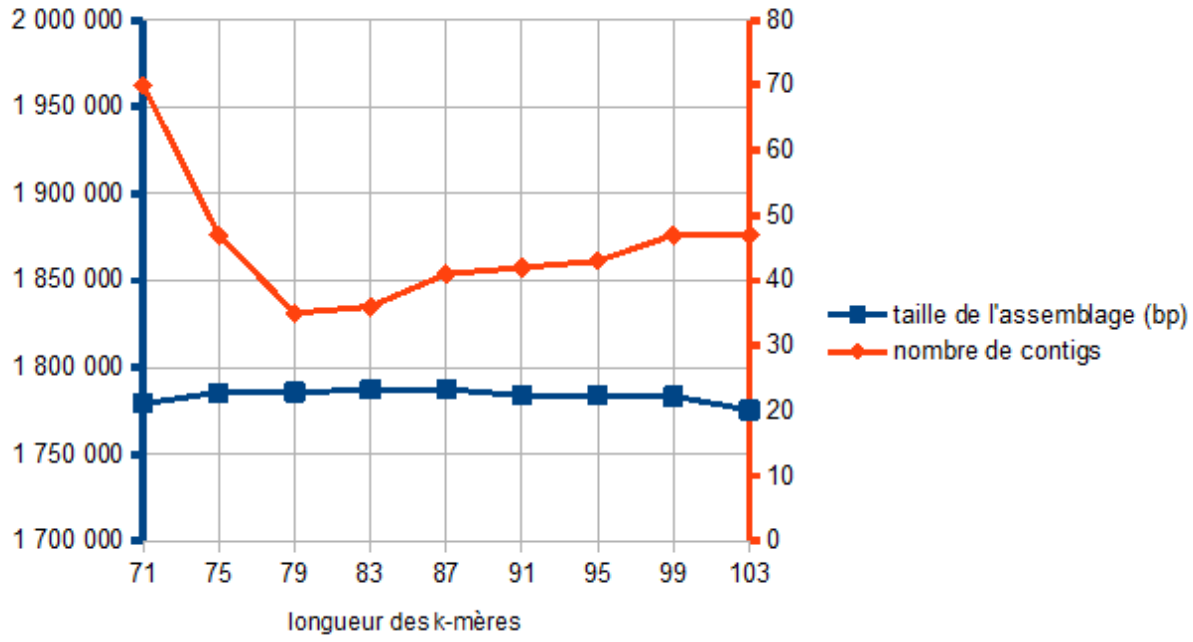


Figure 9. Propriétés des assemblages qui seront utilisés pour faire la finition du génome de StachEndo. Ces assemblages ont été réalisés avec des lectures Illumina paired-end et le logiciel Velvet en utilisant une couverture attendue de 670 et un seuil de couverture de 95,7.

C.3 - Finition de génomes bactériens à l'aide de méthodes connues

La finition d'assemblage est le procédé par lequel on essaye de fermer les trous dans un assemblage de façon à obtenir autant de contigs que de molécules séquencées. À la base, on essaye de combiner des contigs préexistants pour obtenir de plus grand contigs. Les trous dans les assemblages sont généralement associés à la présence d'une région répétée dans

l'assemblage ou d'une région faiblement couverte par le séquençage (10).

Dans certaines situations, si les contigs ne peuvent pas être superposés pour former un nouveau contig, il peut être possible de les utiliser pour former des *scaffolds*. Cela consiste à créer une structure comprenant deux ou plusieurs contigs séparés par un nombre approximatif de nucléotides inconnus (10, 29). Les *scaffolds* peuvent être créés en utilisant des lectures pairées. En connaissant l'espace entre deux lectures d'une paire, chacune se trouvant sur un contig différent, on peut inférer la distance entre ces deux contigs et créer un *scaffold* (10, 29). En connaissant la distance séparant deux séquences, il peut être plus facile de trouver ce qui pourrait se situer entre elles. La facilité et la fiabilité avec laquelle nous pouvons créer des *scaffolds* dépend directement de la précision avec laquelle la librairie de séquençage pairée a été créée (10, 19, 29). Si l'espacement entre les lectures des paires d'une même expérience de séquençage n'est pas constant, il peut devenir difficile d'utiliser ce jeu de données.

Pour essayer de finir le génome de StachEndo, nous avons d'abord testé plusieurs méthodes de finition de génome disponibles. La premier outil que nous avons testé est le logiciel de *scaffolding* SSPACE (37).

C.3.1 – SSPACE

L'approche de SSPACE consiste à aligner des lectures *paired-end* sur des contigs fournis par l'utilisateur (37). La distance attendue entre les lectures d'une même paire est ensuite utilisée pour positionner les contigs les uns par rapport aux autres (37). De plus, en combinant SSPACE avec GapFiller (41), il est possible de combler les régions séparant les contigs en ajoutant des lectures aux extrémités des contigs existants. De cette façon, les trous dans les *scaffolds* peuvent être réduits ou éliminés, produisant un assemblage plus complet (37, 41).

Nous avons donc d'utilisé SSPACE pour créer des *scaffolds* à l'aide des contigs Velvet optimaux (voir section précédente) et de l'information apportée par les lectures Illumina *paired-end*. Au départ, notre assemblage comptait 35 contigs totalisant 1 785 608 pb. Il est passé à 5 *scaffolds* et 22 contigs totalisant 1 791 117 pb. Ces 5 *scaffolds* comprenaient 13

contigs. De ces 5 *scaffolds*, un était complètement refermé et sans aucun trou et se composait de trois anciens contigs. Deux autres sont composés de trois régions de séquence connue séparées par deux trous. Les 2 autres *scaffolds* ne possèdent qu'un seul trou chacun. Il y a donc six trous à combler au total. La taille de ces trous est en moyenne de 913 pb. Il s'agit de trous assez petits qui devraient être faciles à combler.

Nous avons ensuite utilisé GapFiller avec les mêmes lectures Illumina *paired-end* pour refermer ces trous. Cette manipulation n'a pas permis de modifier le nombre de *scaffolds* et de contigs mais a fait légèrement augmenter la taille totale de l'assemblage (1 791 791 pb). Trois des six trous précédemment identifiés sont essentiellement fermés. On détecte simplement la présence d'un nucléotide indéterminé à la position de l'ancien trou. Les trois trous restant sont répartis sur deux *scaffolds* et sont d'une taille moyenne de 260 pb.

Un des *scaffolds* créé et complètement fermé correspond au génome mitochondrial de StachEndo. Le génome circulaire complet est donc présent à la fin de nos manipulations sous la forme d'un seul contig. Il s'agit d'une nette amélioration par rapport aux contigs de notre assemblage où le génome mitochondrial complet représentait deux contigs.

Au final, le *scaffolding* et le remplissage des trous nous a permis de simplifier l'assemblage original, nous permettant de réduire de 8 le nombre de morceaux le composant. Cependant, même après la formation de *scaffolds*, l'assemblage est composé d'une vingtaine de contigs. Le *scaffolding* ne semble pas nous permettre de pousser la finition plus loin.

Une des grandes limites du *scaffolding* de contigs à l'aide de lectures *paired-end* est que la formation de *scaffolds* dépend en grande partie de la qualité du séquençage *paired-end* effectué (19, 33). Si la production de la librairie *paired-end* utilisée pour le séquençage est mal faite, il est possible que la distance réelle séparant les deux lectures d'une paire ne soit pas la distance attendue. Cette information erronée peut perturber le fonctionnement du logiciel de *scaffolding* qui utilise la distance entre les lectures d'une paire utilisée dans l'assemblage. La formation de *scaffolds* peut en être affectée. La formation de *scaffolds* erronés est également possible (19).

Nous avons donc aligné les paires de lectures sur l'assemblage de référence Sanger du génome mitochondrial de *Stachyamoeba lipophora* pour avoir une idée de la distribution de leur espacement. Selon cette analyse, 30 % des paires ne sont pas correctement espacées (**Figure 10**). Il est possible que ces erreurs expliquent en partie le peu de *scaffolds* que nous avons réussi à créer.

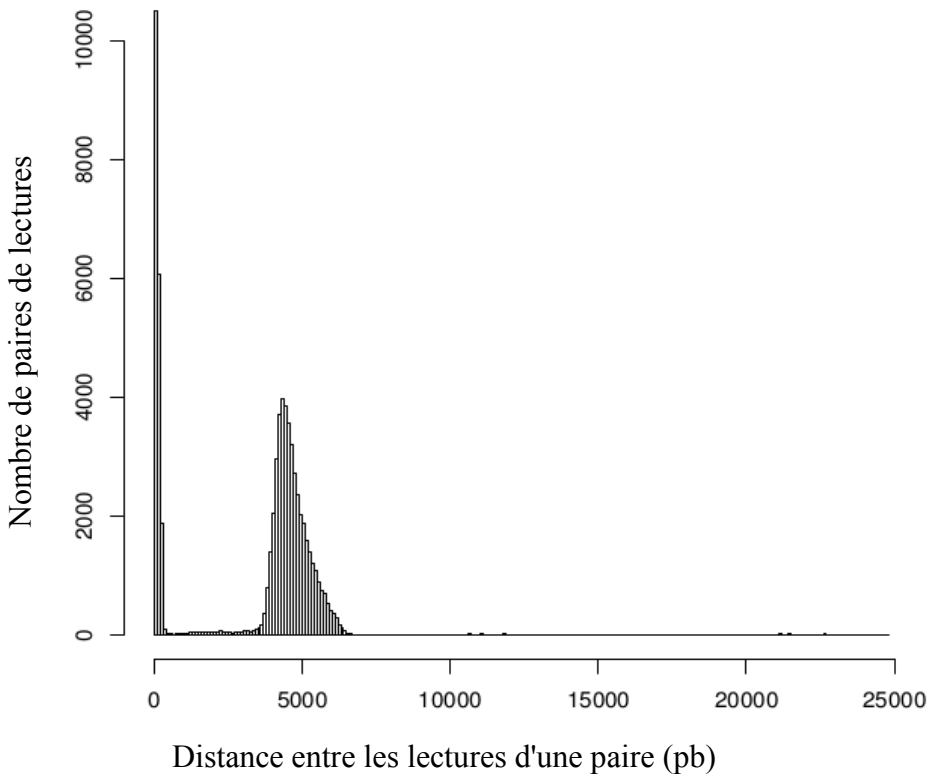


Figure 10. Distribution de la distance entre les lectures Illumina pairées de types Mate-pair dans le génome mitochondrial de *Stachyamoeba lipophora.* La distance attendue étai d'environ 5000 pb. On remarque qu'environ 30 % des lectures de se trouvent à des distances plus rapprochées.

Il est également important de ne pas minimiser la difficulté que représente la résolution des répétitions. Si une répétition est plus longue que l'espace séparant les lectures de nos paires, ou si nous avons trop peu de paires valides, nous ne pourrons pas utiliser correctement l'information de paires pour les résoudre.

C.3.2 – Minimus2

Compte tenu des résultats obtenus, nous avons poursuivi nos efforts. Le deuxième outil que nous avons testé est Minimus2 (73), spécialisé pour la finition d'assemblages et la fusion de contigs. Minimus2 nécessite la définition de deux groupes de séquences qui seront comparés à l'aide du logiciel d'alignement Nucmer (86, 87). Puis, si possible, ils seront combinés pour former de nouveaux contigs. Dans un premier temps, nous avons essayé de simplifier le meilleur assemblage. Nous avons donc défini nos deux jeux de données comme étant les contigs de l'assemblage du k-mère 79. Suite à l'assemblage par Minimus2, nous sommes passés d'un assemblage initial de 35 contigs seulement à un assemblage de 33 contigs d'une taille quasiment inchangée. A priori, il y a peu de chevauchements possibles dans les contigs générés. Les trous restants sont sûrement produits par la présence de répétitions et par une couverture inégale. Une approche intéressante pour essayer de fermer ces trous serait d'utiliser les contigs variés des assemblages d'un même jeu de données à partir de k-mères de tailles différentes.

En effet, tel que mentionné précédemment, l'assemblage d'un même jeu de données à l'aide de k-mères de tailles différentes présente plusieurs avantages. Les contigs qui sont générés ne sont pas exactement les mêmes. De plus, les k-mères de plus grande taille favorisent la formation des contigs appartenant aux régions plus couvertes de l'assemblage. Inversement, l'utilisation de k-mères plus courts permet de mieux capter les régions de faible couverture. C'est pour ces raisons qu'on a essayé de combiner les contigs de nos deux meilleurs assemblages générés à partir de k-mères de tailles différentes. Le premier groupe donné à Minimus2 était donc composé des contigs du k-mère 79 et le deuxième groupe, des contigs du k-mère 83. Le résultat de cette tentative de finition fut beaucoup plus fructueux. Le résultat final de cet assemblage de contigs nous a permis d'obtenir un nouvel assemblage composé de 18 contigs pour un total de 1,784Mpb. Malgré tous les contigs qui furent combinés lors de la finition de cet assemblage, nous retrouvons toujours le génome mitochondrial de *Stachyamoeba lipophora* en deux contigs distincts. En moyenne les contigs sont d'une longueur de presque 100 000 pb, ce qui est près du double de la taille moyenne des contigs

avant la finition. Il semblerait que, conformément à notre hypothèse, l'utilisation de contigs générés à l'aide de k-mères de tailles différentes présente une bonne source de contigs pour la finition de génome.

Pour une troisième tentative, on a décidé de fournir plus de contigs d'assemblages différents à Minimus2. On a donc ajouté un deuxième assemblage au deuxième groupe de contigs. Notre premier groupe était donc composé des contigs du k-mère 79 et notre deuxième groupe, des contigs des k-mères 83 et 87. Les résultats de cette expérience ne sont pas très bons. On a obtenu 24 contigs, ce qui est plus qu'aux deux autres essais, et une taille totale de l'assemblage de 2.1Mpb, ce qui est largement supérieur à la taille attendue.

Finalement, une dernière tentative fut réalisée en ajoutant les résultats de l'assemblage du génome de StachEndo généré avec les k-mères 91 et 75. Cependant, peu importe le nombre d'assemblages ajoutés ou les combinaisons possibles entre-elles, l'utilisation d'un deuxième groupe de contigs composé de plus d'un assemblage est nuisible pour la finition. Plus on fournit de séquences à Minimus2 et plus il essaiera de les ajouter au premier groupe, ce qui fera gonfler l'assemblage, mais ne permettra pas de combiner aussi efficacement les contigs du premier groupe.

Parallèlement, on a également voulu voir si Minimus2 pouvait être utilisé pour combler les gaps dans les *scaffolds*, de cette façon, nous pourrions utiliser Minimus2 comme substitut de GapFiller. Nous avons donc utilisé les 27 *scaffolds* créés par SSPACE à partir de l'assemblage du k-mère 79. Ces *scaffolds* composèrent le premier groupe de séquences données à Minimus2. Le second groupe correspondait aux contigs de l'assemblage réalisé avec le k-mère 71. Cet assemblage et celui disposant du k-mère le plus petit parmi les assemblages du génome de StachEndo que nous avons réalisé. Par conséquent, il s'agit de l'assemblage qui devrait contenir le plus de diversité et qui est donc plus susceptible de contenir des contigs pouvant remplir les trous entre les contigs des *scaffolds* créés. Le résultat de Minimus2 ne fut pas convaincant puisqu'il fut incapable de fermer, même partiellement, les trous des *scaffolds* fournis.

En conclusion, la principale utilité de Minimus2 semble être de pouvoir combiner des jeux de séquences différents. Cette propriété en fait un bon candidat pour la finition de génomes. Cependant, il faut faire attention quant aux lectures que nous utilisons lors de la finition. L'utilisation de Minimus2 nécessite la définition de deux groupes de séquences. Le premier groupe est composé des séquences de référence, le deuxième groupe est composé des séquences qui seront comparées et ajoutées au premier. Selon nos observations, cette caractéristique peut être utilisée pour combiner des assemblages différents du même jeu de données. Cependant, si le deuxième groupe contient beaucoup de séquences, il semblerait que la réduction de l'assemblage ne fonctionne pas de façon aussi efficace qu'avec moins de séquences.

À ce stade-ci nous avons été capables de réduire l'assemblage du génome de StachEndo à seulement 16 contigs. Par contre nous n'avons pas réussi à obtenir l'assemblage complet du génome mitochondrial de *S. lipophora* en un seul contig, tel qu'attendu. Nous allons donc continuer à essayer d'autres approches de finition. Cependant, nous remarquons que les approches de réassemblage pourraient être une avenue intéressante pour réunir les contigs de différentes expériences d'assemblage d'une même séquence.

C.3.3 – CONSED

Les tentatives suivantes ont fait appel à Consed, un logiciel de visualisation et de finition d'assemblage publié en 1998 (74). Il fait partie de la suite d'assemblage Phred (évaluation de la qualité des lectures), Phrap (assemblage), et Consed. Initialement, cette suite d'outils avait été développée pour le projet de séquençage complet du génome humain dans les années 90 (77). C'est un des assembleurs les plus anciens et les plus utilisés. La popularité de cette suite d'outils a diminué avec l'arrivée des méthodes de séquençage de nouvelles génération, car les lectures produites par ces méthodes étaient beaucoup plus courtes que les lectures de type Sanger couramment utilisées à l'époque de sa création. Par contre, les nouvelles versions de Consed supportent l'utilisation de lectures de nouvelle génération (88).

Le module Autofinish (89) de Consed sert à prédire des amorces de séquençage pour générer

les lectures permettant de compléter les trous dans l'assemblage (89). On a donc utilisé les modules d'assemblage normaux de Consed pour combiner les contigs. Après de multiples calculs, Consed fut incapable de réduire davantage l'assemblage que nous lui avions fourni. Il ne semble donc pas y avoir de chevauchement assez significatif parmi les contigs pour que Phred/Phrap/Consed puisse procéder à leur assemblage.

Pour utiliser ces modules, il est impératif de fournir à Consed les séquences des contigs que l'on désire réassembler en plus des positions des différentes lectures de séquençage les composant (89). De tels fichiers sont très lourds et leur traitement est long et coûteux en mémoire. Même si la suite de Phrap est un classique de l'assemblage, il serait probablement plus approprié de se pencher sur des approches mieux adaptées aux techniques de séquençage de nouvelle-génération.

C.3.4 – Mira

Finalement, nous avons essayé d'utiliser Mira pour la finition du génome. Mira avait donné de bons résultats lors de l'assemblage du génome mitochondrial de StachEndo avec les lectures 454, ainsi que lors de l'assemblage hybride du même génome par alignement des lectures Illumina sur les lectures 454. De plus Mira est un assembleur qui accepte un large éventail de type de données, que ce soit des lectures de type Sanger ou bien de type « nouvelle génération ». Il serait donc intéressant d'utiliser Mira pour combiner les contigs de différentes expériences.

Par contre, certains facteurs limitent l'utilisation de Mira pour faire de la finition d'assemblage. Premièrement, Mira n'accepte pas les lectures plus longues que 29 900 paires de bases. Tous les contigs plus longs que 29 900 nucléotides devront donc être coupés. Pour s'assurer que les fragments de contigs pourront toujours être assemblés pour reformer leur contig original, nous produirons des fragments chevauchants. Les contigs ont donc été divisés en fragments de 29 900 nucléotides débutant à toutes les 10 000 positions. En décalant les point de fragmentation de 10 000 nucléotides, on s'assure que des fragments adjacents se chevaucheront sur près du deux-tiers de leur séquence. Deuxièmement, Mira utilise la qualité des lectures qui lui sont

données pour calculer plusieurs statistiques qui orienteront certaines de ses décisions. De façon générale, peu d'assembleurs DBG attribuent des valeurs de qualité aux positions de leurs assemblages. Les contigs que nous avons générés ne possèdent donc pas de score de qualité. Pour faciliter le bon fonctionnement de l'assemblage, nous attribuons une qualité uniforme de 30 à toutes les positions de l'assemblage. Il s'agit d'une qualité moyenne qui n'avantage ou ne désavantage aucun contig, mais permet à Mira de fonctionner.

Le premier essai consista à confier à Mira les fragments des 35 contigs produits à partir de k-mères de longueur 79 et un fichier de qualité (uniforme à 30). L'assemblage obtenu était composé de 26 contigs et avait une taille totale de 2,1 Mpb. La taille moyenne des contigs est 81 500 pb et leur couverture médiane de 2. En matière du nombre de contigs, il s'agit du plus petit assemblage que nous ayons produit en n'utilisant que les contigs de l'assemblage de k-mères 79. Cependant, on remarque que la taille totale de l'assemblage est assez grande.

Nous avions déjà mentionné que l'assembleur Mira avait tendance à produire beaucoup de contigs. Un certain nombre d'entre eux sont habituellement des débris de l'assemblage qui peuvent être retirés par filtration des contigs peu couverts. Dans cette situation, il est difficile de distinguer les véritables contigs des débris d'assemblage, car tous les contigs ne sont composés que de 2 à 4 séquences. Par conséquent, il est impossible de distinguer les contigs très couverts des contigs peu couverts. Cette approche de finition n'a donc pas été suffisante pour obtenir le génome mitochondrial de *Stachyamoeba lipophora* en un seul contig. Il est possible qu'en ajoutant plus de contigs à l'assemblage, les différences de couverture entre les bons contigs et les débris deviendront plus évidentes.

Nous avons ensuite essayé de finir le génome de StachEndo en utilisant les contigs des assemblages Velvet aux k-mères 79 et 83 puisqu'il s'agit des deux meilleurs assemblages. L'assemblage résultant est composé de 40 contigs et d'une taille de 3,9 Mpb.

Puisque Mira, contrairement à Minimus2, essaye de réassembler les contigs qui lui sont fournis, plutôt que de combiner deux jeux de séquences, tous les contigs sont considérés pêle-mêle. Par conséquent, l'assemblage résultant est composé de plus de contigs que chacun des

assemblages que nous lui avons donnés en entrée, mais de moins de contigs que ces deux assemblages mis ensemble. Pour cet assemblage la gamme de couverture est un peu plus étendue, allant de 2 à 6 avec une médiane de 3. Les contigs représentant le génome mitochondrial de *Stachyamoeba lipophora* ont une couverture de 4. En filtrant notre assemblage pour ne conserver que les contigs ayant une couverture de 4 ou plus, on conserve 14 contigs équivalent à 2,1 Mpb de l'assemblage. En ne conservant que les contigs possédant une couverture de 5 ou plus, il resterait 5 contigs représentant 1,1Mpb. Les écarts de couverture entre les contigs ne semblent pas très importants. Le choix arbitraire de seuil de couverture arbitraire pour filtrer les contigs risque de créer des trous dans l'assemblage.

Nous avons poursuivi les expériences avec des contigs provenant des assemblages du génome de StachEndo obtenues avec les k-mères 79, 83 et 87. Le résultat de cet assemblage était composé de 55 contigs d'une taille totale de 5.5Mpb. Il semble donc que plus on ajoute de contigs d'assemblages différents, plus l'assemblage lui-même devient gros et la taille moyenne des contigs augmente. On a obtenu deux très grands contigs de 781 981 et 696 131 pb qui à eux deux sont de la taille attendue du génome de StachEndo. Cependant, ces contigs étaient identiques sur 99 % des positions en commun, ne déviant l'une de l'autre que pour un seul nucléotide.

Mira semble donc avoir des difficultés à assembler les contigs. Une mise en garde est apparue lors de l'exécution nous informant de la détection de *megahub* dans l'assemblage. Les *megahubs*, selon Mira, sont des lectures de l'assemblage qui sont extrêmement répétitives et qui sont donc des centres (*hubs*) autour desquels un nombre important de combinaison de lectures pourrait prendre place. Puisque que les contigs provenant de différents assemblages sont partiellement redondants, cela le dédoublement de l'assemblage par Mira. On a donc procédé à un *clustering* des contigs produits, pour éliminer les contigs quasi-dédoublés. Résultat : l'assemblage ne contenait réellement que 19 contigs d'une taille totale de 2 292 819 pb (donc trop grand). Deux de ces contigs représentent le génome mitochondrial de *Stachyamoeba lipophora*.

On a tenté une dernière expérience combinant les contigs des k-mères 79, 83, 87 et 75. Le

résultat était composé de 78 contigs d'une taille totale de 7.4 Mpb. Dans ce cas-ci, on remarque que la taille moyenne des contigs n'est pas aussi grande. Ce constat reste le même suite au *clustering* des contigs. Après le *clustering*, l'assemblage est d'une taille de 2 831 475 pb et composé de 25 contigs. Cette fois-ci, le génome mitochondrial de *Stachyamoeba lipophora* est présent en un seul contig. Bien que la taille totale de l'assemblage ait énormément diminuée suite au *clustering*, la taille totale de cet assemblage est toujours très supérieure à la taille attendue du génome de StachEndo (1,8 Mpb).

En conclusion, Mira paraît être un outil intéressant pour la finition de génome à partir d'assemblages à divers k-mères. Cependant, son utilisation nécessite de nombreuses manipulations préliminaires des données. Du découpage des contigs trop longs, jusqu'au *clustering* des contigs, plusieurs analyses doivent être exécutées avant d'obtenir l'assemblage final. Aucun de ces traitements n'est, en soi, très complexe et le tout peut facilement être automatisé. La grande limite de Mira a toujours été sa tendance à produire un nombre très important de contigs, spécifiquement de très courts contigs. Lors des tests, nous avons remarqué que le nombre de contigs produits pouvait également être gonflé artificiellement par la présence de contigs dupliqués. L'utilisation de *clustering* semble permettre d'enrayer ou, au moins, de limiter ce phénomène. Il pourrait être intéressant de procéder à un *clustering* des contigs des assemblages à finir avant de les réassembler pour voir si cela pourrait réduire le phénomène des *megahubs* et de la duplication de contigs observée et, ainsi, améliorer l'assemblage. Il faudrait cependant être prudent, car ce faisant, la couverture de Mira risque de devenir encore plus basse qu'elle ne l'était déjà. Ceci pourrait augmenter l'accumulation d'erreurs dans l'assemblage.

C.4 – Finition par addition préférentielle de k-mères

Lors des expériences précédentes, nous avons observé l'importance du choix de la taille de k-mères utilisée pour l'assemblage de génome par graphes DeBruijn. Nous avons introduit le concept de point de rupture, la taille de k-mère optimale pour l'assemblage après laquelle l'assemblage se dégrade progressivement. L'utilisation de k-mères plus grand que cette valeur mène à la disparition de contigs de faible couverture de l'assemblage. Ce phénomène a pour

avantage de favoriser l'assemblage des régions plus couvertes, puisque la présence d'une plus grande quantité de lectures dans une région lui permet de mieux survivre aux critères plus stringents des assemblages à grands k-mères. Cependant, la disparition des régions peu couvertes fragilise la formation de contigs qui deviennent plus vulnérables à la distribution inégale des lectures séquencées, ainsi qu'aux zones très divergentes causées par la présence d'erreurs de séquençage ou de polymorphisme. Inversement, l'utilisation de k-mères plus petit que le point de rupture permet de mieux capter ces zones plus complexes qui sont perdues avec des k-mères de grande taille. Indépendamment du concept de point de rupture, la résolution d'une répétition d'une taille donnée dans un graphe DeBruijn ne peut être accomplie qu'en utilisant des k-mères qui sont plus longs que la répétition en question. L'utilisation de grand k-mères permet donc d'assembler plus facilement les répétitions.

Essentiellement, on observe que tous les assemblages et tous les k-mères ne sont pas équivalents. Il ne semble donc pas logique de considérer tous les assemblages au même niveau lorsque nous essayons de finir un génome. Lorsque nous fournissons à Mira des jeux de contigs, ces contigs sont simplement comparés les uns aux autres. Dans le cas de Minimus2, au lieu de comparer tous les contigs entre eux, on compare plutôt deux jeux de contigs l'un à l'autre. Compte tenu de ce que nous avons observé, il semblerait plus logique d'essayer de combler le meilleur assemblage avec nos autres assemblages de bonne qualité, puis, seulement après, d'aller chercher dans les assemblages plus biaisés/spécialisés produits avec de très grandes ou de très petites tailles de k-mères.

C.4.1 – Méthode

Nous introduisons donc l'idée de finir un génome par addition préférentielle de k-mères. En résumé, chaque assemblage se trouve attribué un ordre hiérarchique de préférence. On essaye donc de finir le génome en utilisant en priorité les contigs appartenant à un assemblage d'un niveau hiérarchique supérieur avant de passer à un assemblage de rang inférieur. On positionnera au sommet de cette hiérarchie les assemblages qui se situent autour du point de rupture. Il s'agit des meilleurs assemblages dont nous disposons. Plus on descendra dans cette hiérarchie plus on se rapprochera des assemblages générés avec des tailles de k-mères

éloignées du point de rupture. Puisque ces assemblages sont plus vulnérables aux erreurs de séquençage et d'assemblage, ils ne seront utilisés qu'en dernier recours. De cette façon, on peut tirer profit des avantages des assemblages réalisés en amoindrissant les risques d'introduire des erreurs lors de la finition, ou de bloquer la progression de l'assemblage en raison de signaux contradictoires provenant des assemblages différents.

Description algorithmique de l'approche :

Soit A_k , un assemblage qui est un ensemble composé de contigs $\{c_{k1}...c_{kn}\}$

- *$k \in N^0$ identifie un assemblage en fonction de sa priorité lors de la finition*
- *A_0 est l'assemblage au sommet de la hiérarchie, c'est l'assemblage que l'on cherche à finir, par addition de contigs*

La première étape de l'algorithme consiste à simplifier chacun des assemblages qui seront utilisés pour faire la finition. Le but de cette étape est de vérifier qu'aucun des contigs ne pourraient être combinés pour former de nouveau contig. En procédant à cette vérification, on s'assure que l'assemblage sera minimal. Suite à cette étape tout chevauchement multiple d'un contig est forcément attribuable à une répétition puisque toutes les simplifications évidentes de contigs ont été accomplies.

Simplification d'un assemblage :

Pour un assemblage donné A_k , tel que $k \in N^0$:

- *Comparaison des contigs $\{c_{k1}...c_{kn}\}$ entre eux à l'aide de l'aligneur Nucmer pour détecter chevauchements possibles.*
- *Sélection des chevauchements valides. Un chevauchement est valide si :*
  - *> Un contig est complètement inclus dans l'autre*
  - *> Les contigs se chevauchent sur une de leurs extrémités respectives*
- *Fusion des contigs en fonction des chevauchements valides.*
  - *> Les contigs inchangés restent dans A_k , les contigs utilisés lors de*

fusions sont retirés et remplacés par les nouveaux contigs créés

Toutes les comparaisons utilisent Nucmer pour aligner les contigs à analyser. L'utilisation de Nucmer est simple et le format de ses alignements est facile à interpréter. Nucmer est un outil fiable et également très rapide.

Les chevauchements que nous recherchons doivent se conformer à certains critères. Les contigs doivent obligatoirement se chevaucher au niveau de leurs extrémités pour qu'ils puissent être fusionnés. La seule exception permise est lorsqu'un contig est entièrement englobé par un autre contig. Puisque les extrémités des contigs sont généralement moins bien couvertes que le reste du contig, il est possible que des erreurs bloquant l'assemblage se trouvent dans ces régions. Nous avons donc inclus un paramètre définissant une distance maximale des extrémités des contigs en dessous de laquelle les contigs peuvent être coupés pour permettre la formation d'un chevauchement valide. La taille minimale et le nombre maximal d'erreur permis dans un chevauchement valide sont également ajustables.

Après simplification de chaque assemblage, on commencer à comparer l'assemblage à finir avec les autres assemblages.

<u>Addition préférentielle des assemblages :</u>

Soit K = 1, tant que K est plus petit que la valeur maximale de k :

- *Comparaison des contigs* $\{c_{01}...c_{0n}\}$ *de l'assemblage* A_o *avec les contigs* $\{c_{K1}...c_{Kn}\}$ *de l'assemblage* A_K *pour détecter des chevauchements*
- *Sélection des chevauchements valides.*
  - *> Si des chevauchements valides existent :*
    - *Fusion des contigs en fonction des chevauchements valides.*
    - *Les contigs inchangés restent dans leur assemblage respectif, les contigs utilisés lors de fusions sont effacés de leur assemblage respectif et les nouveaux contigs créés sont placé dans* A_0

- On essaye de simplifier l'assemblage A_0 pour voir si les nouveaux contigs permettent de simplifier l'assemblage davantage.

- On réinitialise K = 1

On retourne au début de la hiérarchie des assemblages avec A_0 pour voir si, suite aux modifications, il est possible d'insérer des contigs provenant des assemblages possédant une priorité supérieure.

> Si aucun chevauchement valide n'est détecté, on passe à l'assemblage suivant ; K++

Le processus de finition se déroule de la façon suivante. On commence par comparer les contigs de notre meilleur assemblage (A_0) à ceux de notre second assemblage (A_1) dans l'ordre prioritaire que nous avons établi. S'il nous est possible de détecter des chevauchement entre les contigs de A_0 et A_1, nous procédons à la fusion de ces contigs qui appartiennent maintenant à A_0. On essaye ensuite de voir s'il est possible de simplifier l'assemblage A_0. On réessaye ensuite d'ajouter des contigs de A_1 à A_0. S'il nous est impossible d'ajouter d'autres contigs de A_1 à A_0, on passe à l'assemblage suivant selon la liste, soit A_2. De façon générale, dès qu'il est impossible d'ajouter des contigs de l'assemblage A_n à A_0, on passe à l'assemblage suivant et on essaye d'ajouter les contigs de A_{n+1} à A_0. À l'inverse, après chaque addition de contigs à A_0, on retourne à l'assemblage A_1 pour voir si les nouveaux ajouts à A_0 permettent d'intégrer des contigs d'assemblages mieux classés à l'assemblage. Le processus de finition ne se termine que lorsque plus aucun contig d'aucun des assemblages fournis ne peut être ajouté.

C.4.2 – Application de la méthode et résultats

Pour évaluer la pertinence de notre méthode on a effectué plusieurs expériences lors de

finition du génome de StachEndo à partir des divers assemblages Velvet-Illumina présentés à la **Figure 9**.

Puisque l'assemblage de StachEndo utilisant des k-mères de longueur 79 est celui composé du moins du contigs, et étant d'une taille très près de la taille attendue du génome, c'est cet assemblage qui est utilisé comme A_0 dans nos expériences. L'assemblage issu des k-mères de longueur 83 est notre deuxième plus petit assemblage, il sera donc utilisé comme A_1 . L'ordre des autres assemblages que nous avons est plus difficile à établir. A priori, si nous nous fions d'abord au nombre de contigs, puis à la taille de totale de l'assemblage, l'ordre des k-mères serait : 87, 91, 95, 75, 99, 103 et 71. Nous avons donc procédé à plusieurs tentatives de finition en utilisant un nombre plus ou moins important d'assemblages.

Nous avons d'abord décidé de faire la finition en n'utilisant que deux assemblages, A_0 = 79 et A_1 = 83. En utilisant 23 contigs de A_1 , nous arrivons à réduire le nombre de contigs dans A_0 de 34 à 26. Ce nouvel assemblage est de 1,83 Mpb et ses contigs font en moyenne 70 400 pb.

En ajoutant à notre calcul, le troisième assemblage de notre liste, soit A_2 = 87, plus de la moitié des contigs de A_1 et A_2 sont ajoutés A_0 dont le nombre de contigs reste 26, mais dont la taille passe à 1,9 Mpb. L'ajout de A_3 = 91, réduit légèrement le nombre de contigs de 26 à 25. L'ajout de A_4 = 95 nous permet de réduire l'assemblage à 23 contigs.

L'ajout des k-mères 75 et 99 nous permet d'atteindre un assemblage de 21 contigs. L'ajout des autres assemblages ne fait qu'augmenter la taille de l'assemblage. Notre assemblage final est d'une taille de 2,8 Mpb, ce qui est beaucoup plus grand que les 1,83 Mpb attendus.

Pour essayer de réduire d'avantage le nombre de contigs de notre assemblage, nous avons utilisé notre méthode de simplification de contigs en permettant aux régions de chevauchement d'être au minimum identiques à 90 %. Cette manipulation nous permet de réduire notre assemblage à 19 contigs d'un total de 2,6 Mpb. Cette méthode, tout comme la

finition par SSPACE et MIRA, permet d'obtenir le génome mitochondrial de *Stachyamoeba lipophora* en un seul contig. En termes de nombre de contigs, cette méthode de finition produit des résultats comparables à ceux des autres méthodes que nous avons testées précédemment. Avec 19 contigs, l'addition préférentielle de k-mères se situe au milieu des autres approches.

Cependant, une analyse nous montre que les contigs produits par notre approche pourraient être simplifiés d'avantage. Nous avons détecté au moins trois chevauchements à l'aide de l'algorithme BLAST qui, n'étant pas sur des séquences identiques, n'étaient pas identifiés par notre approche très stricte. Nous avons donc ré-exécuté l'addition préférentielle de k-mères mais en acceptant des chevauchements entre contigs possédant au moins 90 % d'identité (par opposition à 100 % lors de notre autre essai).

Cette nouvelle finition a généré un assemblage de 12 contigs d'un total de 2,6 Mpb. Il s'agit d'une réduction importante des 19 contigs que nous avions précédemment obtenus. Cependant, nous observons toujours certaines superpositions non résolues entre nos contigs. Ces superpositions mesurent environ 250 000 pb et sont identiques à 99 %. Cependant, ces chevauchements de contigs ne sont pas détectés par Nucmer, donc il est impossible pour notre logiciel de les considérer lors de la simplification de l'assemblage. Ces chevauchements se produisent sur des segments de séquences quasi-identiques à l'exception d'une ou deux régions d'une centaine de nucléotides où le pourcentage d'identité diminue autour de 90-85 %. Ces régions de plus faible identité semblent empêcher leur détection par Nucmer.

Ces superpositions ne sont pas ambiguës et sont de grande taille. En procédant à leur simplification, on obtient un assemblage de 1,9 Mpb en 10 contigs. La taille totale de l'assemblage est très proche du 1,8 Mpb attendu. De plus, cette approche est celle qui nous a permis de réduire le plus possible le nombre de contigs composant l'assemblage. Ces observations semblent nous indiquer que l'addition préférentielle de k-mères est l'approche la plus efficace pour la finition de génome, parmi celles que nous avons mentionné jusqu'à présent.

C.4.3 – Critiques de la méthode

Tel que montré par la réduction du nombre de contigs lors de la finition, l'addition préférentielle de k-mères est une approche conceptuellement valide. Pour nous assurer de sa réelle efficacité, nous avons comparé les 10 contigs, obtenus avec cette méthode, au génome fini de StachEndo obtenu grâce à l'approche décrite dans la section suivante (**B.4 Assemblage complet du génome de la bactérie endosymbiote StachEndo**). Cette analyse nous indique que les contigs possèdent un taux d'identité de 99,94 % avec le génome complet que nous supposons être juste. Malgré le fait que nos contigs contiennent un certain nombre d'erreurs potentielles, la structure générale de l'assemblage est conforme. La précision de la finition pourrait probablement être améliorée en procédant à un mappage des lectures de séquençage sur nos contigs après addition préférentielle pour corriger les erreurs pouvant être introduites par des erreurs dans les contigs.

Cependant, des erreurs systématiques sont observables dans les contigs post-finition. Le génome de StachEndo possède, d'après notre assemblage plusieurs régions hautement répétitives d'environ 1000 pb. Notre méthode a tendance à écraser partiellement ces répétitions si elles se trouvent aux extrémités des contigs, produisant des régions plus courtes. Puisque nous cherchons toujours les chevauchements de plus grandes tailles entre les contigs, si les contigs se chevauchent sur une région répétitive la région de chevauchement peut sembler plus longue. La fusion des contigs mène donc à la formation d'une séquence plus courte qu'attendue.

Le concept de graphe, en prenant en compte la couverture des k-mères, permet trouver la longueur correcte de telles répétitions, évitant d'écraser ces répétitions. Il serait donc intéressant de trouver une façon d'intégrer la représentation en graphe de l'assemblage à l'addition préférentielle de k-mères. De cette façon, il nous serait possible de tirer profit d'information relative aux lectures de séquençage et à la couverture de l'assemblage pour résoudre certaines ambiguïtés de nucléotides, ainsi que la taille et l'agencement des répétitions.

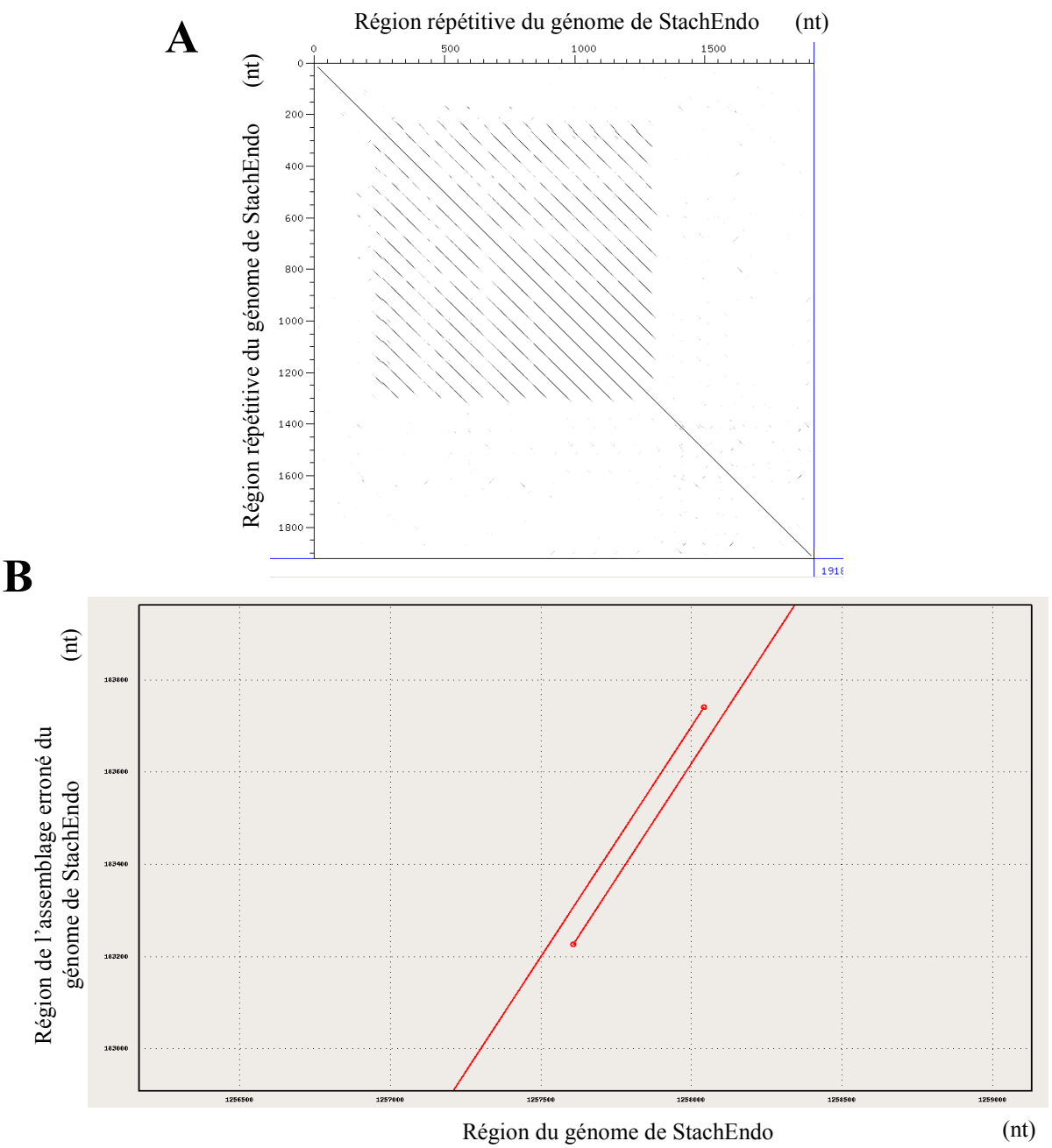


Figure 11. Erreurs d'assemblages associées à la présence de régions hautement répétitives dans le génome de StachEndo. (**A**) Diagramme à points (*dotplot*) du génome assemblé complet de StachEndo contre lui-même dans une région hautement répétée (environ 100 nt des positions 1 257 500 à 1 259 499). (**B**) Alignement de l'assemblage erroné réalisé par addition préférentielle des k-mères contre le génome complet de StachEndo présentant, en ordonnées, aux positions 163 200 à 163 600, l'écrasement d'une région répétitive.

Au niveau de notre implémentation actuelle de la méthode d'addition préférentielle de k-mères, l'utilisation de Nucmer pour la détection des chevauchements de contigs ne semble pas idéale puisque nous ratons certains chevauchements observables avec d'autres outils. Il serait donc nécessaire de trouver d'autres outils aussi rapides et simple à analyser que Nucmer, mais possédant une sensibilité plus élevée. La re-correction des lectures utilisées pour ces assemblages pourrait peut-être aider à diminuer ce type d'erreurs.

C.5 - Assemblage complet du génome de la bactérie endosymbiote StachEndo

Pour aider la finition de l'assemblage du génome de StachEndo, un second séquençage Illumina, utilisant le protocole MiSEQ *paired-end overlap*, fut effectué. 7 330 112 paires de lectures de 250 pb furent produites par ce séquençage. Ces lectures sont pairées chevauchantes, elles peuvent donc être fusionnées de façon à obtenir des lectures plus longues. De cette façon, on tire profit de lectures d'une taille comparable à 4 mais avec le taux d'erreur plus faible d'Illumina et ne possédant pas d'erreurs systématiques du compte des homopolymères. L'ajout de ces nouvelles lectures nous permis de compléter l'assemblage du génome de StachEndo.

C.5.1 – Description de l'approche

La première étape de l'assemblage consiste à identifier les amorces de séquençage présentes sur les lectures et de les retirer. Pour cette étape, les logiciels Trimmomatic (90) et Cutadapt (91) sont utilisés. Par la suite les paires de lectures MiSEQ chevauchantes sont fusionnées, lorsque possible, avec le logiciel FLASH (92). Cette étape produit des contigs Illumina qui sont d'une longueur comparable à des lectures 454. Cette situation s'apparente à avoir de longue lectures de séquençage de type Illumina. Toute queue poly-A ou poly-T restant sur les contigs sont également retirées en utilisant Prinseq (93) pour ne pas nuire à l'assemblage. Ces contigs sont ensuite fournis à Velvet en plus des lectures ne se chevauchant pas et de celles devenues orphelines suites à nos diverses filtrations.

Plusieurs assemblages Velvet ont ensuite été effectués. Comme lors des expériences

précédentes, on commence par déterminer des valeurs de couverture attendue et de seuil de couverture raisonnables. On réalise ensuite plusieurs assemblages pour identifier une taille de k-mère optimale.

La deuxième phase de cette méthode consiste à prendre les contigs du meilleur assemblage généré à partir des données MiSEQ et d'y ajouter les lectures HiSEQ mate-pair sur de longs fragments d'ADN (environ 5 kpb). La même approche, décrite précédemment, avec Velvet est utilisée (calcul de paramètres sensés et identification du k-mère optimal).

La troisième et dernière phase de cet assemblage consiste à reprendre le meilleur assemblage généré à l'étape précédente et à essayer de refermer les trous formants les contigs. Pour ce faire, les contigs du meilleur assemblage sont multipliés dix fois, pour créer une illusion de couverture, et retraités par Velvet. En utilisant Velvet pour faire la finition, il est possible d'utiliser la couverture des séquences, ainsi que l'information des paires de séquences pour orienter la fermeture des trous de l'assemblage. Ces informations n'étaient pas prises en compte dans notre implémentation de l'addition préférentielle des k-mères. L'utilisation de ces informations devraient permettre de résoudre diverses situations ambiguës, ainsi que d'éviter de commettre des erreurs lors de la finition.

C.5.2 – Obtention de l'assemblage final

Des 7 330 112 paires de lectures MiSEQ chevauchantes, 6 887 146 purent être assemblées. La taille moyenne de ces nouveaux fragments était d'environ 340 nt.

L'assemblage Velvet des lectures MiSEQ a produit un assemblage de 1,8 Mpb ce qui correspond parfaitement à nos estimations précédentes. Par contre, cet assemblage était composé de 29 contigs différents tous plus courts qu'un dixième de l'assemblage total, à l'exception d'un seul contig mesurant un peu plus de 0,8 Mbp. L'ajout des lectures HiSEQ à l'assemblage n'a que très légèrement réduit le nombre de contigs produits tout en faisant augmenter la taille totale de l'assemblage.

Cependant, en réinjectant les contigs du meilleur assemblage obtenu, ainsi que les lectures de HiSEQ et MiSEQ, dans Velvet, l'assemblage résultant est composé de 22 contigs. Le plus grands de ces contigs est un contig circulaire de 1 737 891 pb. En soustrayant la région chevauchante permettant de fermer le cercle du contig, le contig aurait une taille de 1 731 915 pb.

Compte tenu de sa taille, cohérente avec les 1,8 Mpb attendus, et de sa forme circulaire, concordant avec la forme des chromosomes bactériens d'espèces apparentées, nous supposons donc que ce contigs correspond au génome complet de la bactérie endosymbiote StachEndo. L'annotation de ce contig nous permettra de confirmer son identité et d'en apprendre plus sur les propriétés de StachEndo.

Tableau III - Comparaison des assemblages du génome de StachEndo et de la mitochondrie de Stachyamoeba lipophora produits par les 5 méthodes de finition testées à partir de contigs Velvet-Illumina (Original).

	Original (Velvet)	SSPACE + GapFiller	Minimus2	Mira	Addition préférentielle des k-mères	Velvet avec paires chevauchantes
Nb. de contigs	35	22	18	19	10	2
Nb. de *scaffolds*	0	5	0	0	0	0
Taille totale (Mpb)	1,786	1,792	1,784	2,293	1,9	1,732

Une observation importante est que l'utilisation de plusieurs librairies pairées Illumina possédant des tailles d'inserts différentes s'est montré essentielle pour la finition du génome de

StachEndo. L'utilisation d'une librairie très espacée (ici 5 kb avec la technologie HiSEQ Mate-Pair de Illumina) permet de résoudre plus de répétitions génomiques. L'utilisation de librairie plus rapprochées, voire même chevauchantes (ici chevauchement d'environ 90 nt avec la technologie MiSEQ Paired-End), permet la formation de lectures Illumina artificiellement longues. Puisque le séquençage Illumina produit des lectures très fiables, ce type de séquençage pairé permet de remplacer les méthodes séquençage comme 454, dont le seul avantage réel était la plus grande longueur de leurs lectures. Notre recommandation, pour d'autres projets de séquençage *de novo* et de finition de petits génomes, est donc de privilégier la combinaison de ces différents types de librairies pairées. De cette façon, on peut bénéficier de la précision supérieure des techniques de séquençage de courtes lectures et quand même obtenir des lectures de grande taille à des coûts réduits.

3 – ANNOTATION ET ANALYSE DU GÉNOME DE STACHENDO

Grâce à l'utilisation d'une combinaison de lectures HiSEQ et MiSEQ, nous avons été capable d'assembler le génome de StachEndo. Nous savons donc maintenant que son génome est composé d'un seul chromosome circulaire mesurant 1 731 915 pb composé à 67% d'A/T. Puisque nous avons maintenant accès à un génome complet, il nous est possible de prédire les gènes qui y sont encodés et de les utiliser pour réaliser une multitude d'analyses.

A – Annotation préliminaire RAST

Regardons les résultats de l'annotation RAST réalisée sur notre assemblage préliminaire du génome de StachEndo (voir section **B.1 – Assemblage préliminaire**). RAST (Rapid Annotation using Subsystem Technology) est un outil d'annotation de génomes. Lors de l'annotation, le logiciel commence par prédire les ARNt et les ARNr présents dans la séquence fournie (85). Ensuite, il utilise un autre logiciel, GLIMMER2 (94, 95), pour prédire rapidement les gènes. Ces gènes sont comparés à une base de données de gènes essentiels à la vie qui sont présents chez toutes les espèces. En fonction des résultats de cette comparaison, RAST défini les espèces qui sont les plus proches phylogénétiquement parlant du génome fourni (85). L'annotation est ensuite complétée en utilisant ces espèces comme références pour la recherche, puis pour l'attribution de la fonction des autres gènes (85). Les fonctions, étant déjà réparties dans des sous-systèmes selon leur rôle biochimique, sont utilisées pour reconstituer les voies métaboliques et les éléments de machinerie cellulaire présents dans le génome annoté (85).

La **Figure 12** présente la liste d'espèces que RAST a identifiées comme étant les plus proches de StachEndo. On remarque que toutes les espèces identifiées sont des Alpha-Protéobactéries ce qui est conforme à nos attentes. Il est possible d'observer deux grands groupes de bactéries parmi ces résultats. Le premier groupe (Azospirillum, Magnetospirillum et Paracoccus*)* est composé de bactéries vivant toutes librement. Azospirillum et Magnetospirillum appartiennent à la famille des Rhodospirillaceae. Paracoccus appartient à la famille des Rhodobacteraceae. Le deuxième groupe, qui occupe une part plus importante de cette table de résultat, est

composé de bactéries endosymbiotes de l'ordre des Rickettsiales. On remarque que les bactéries de la famille des Rickettsiaceae (les 18 Rickettsia) sont plus présentes que les bactéries de la famille des Anaplasmataceae (les 5 Ehrlichia et la Wolbachia). Il semble donc raisonnable de penser que StachEndo présente de forte similarité avec les bactéries du genre des Rickettsia ce qui corrobore nos impressions initiales. Il est tout de même intéressant de noter que l'espèce prédite comme étant la plus proche est une bactérie libre (*Azospirillum sp. B510*) plutôt qu'une endosymbiote. Notre hypothèse concernant cette observation est que StachEndo se trouve, phylogénétiquement parlant, plus près de la division entre les bactéries endosymbiotes et les bactéries vivant librement que beaucoup de Rickettsiaceae. Nous chercherons donc à étudier ce qui différencie StachEndo des Rickettsiaceae et des Rhodospirillaceae.

Genome ID	Score	Genome Name
137722.3	516	Azospirillum sp. B510
336407.4	391	Rickettsia bellii RML369-C
293614.3	384	Rickettsia akari str. Hartford
293614.5	372	Rickettsia akari str. Hartford
257363.1	365	Rickettsia typhi str. Wilmington
336407.7	350	Rickettsia bellii RML369-C
391896.3	338	Rickettsia bellii OSU 85-389
391896.4	327	Rickettsia bellii OSU 85-389
293613.4	323	Rickettsia canadensis str. McKiel
342108.9	314	Magnetospirillum magneticum AMB-1
342108.5	296	Magnetospirillum magneticum AMB-1
272947.1	283	Rickettsia prowazekii str. Madrid E
315456.3	283	Rickettsia felis URRWXCal2
293613.3	282	Rickettsia canadensis str. McKiel
257363.4	280	Rickettsia typhi str. Wilmington
272947.5	274	Rickettsia prowazekii str. Madrid E
315456.7	273	Rickettsia felis URRWXCal2
431944.4	260	Magnetospirillum gryphiswaldense MSR-1
272944.1	258	Rickettsia conorii str. Malish 7
269484.4	257	Ehrlichia canis str. Jake
66084.3	256	Wolbachia sp. wRi
302409.3	251	Ehrlichia ruminantium str. Gardel
269484.6	249	Ehrlichia canis str. Jake
449216.3	249	Rickettsia prowazekii Rp22
272944.4	248	Rickettsia conorii str. Malish 7
302409.5	243	Ehrlichia ruminantium str. Gardel
254945.26	242	Ehrlichia ruminantium str. Welgevonden
444612.3	238	Rickettsia endosymbiont of Ixodes scapularis
318586.4	230	Paracoccus denitrificans PD1222
163164.1	222	Wolbachia sp. endosymbiont of Drosophila melanogaster

Figure 12. Liste des espèces présentant les sous-systèmes fonctionnels les plus similaires à ceux de StachEndo selon logiciel RAST.

B – Annotation du génome final avec PROKKA

Pour réaliser l'annotation finale du génome de StachEndo, nous avons utilisé le logiciel Prokka (96) spécialisé dans l'annotation de bactéries, d'archéobactéries et de virus. Prokka utilise une combinaison d'outils pour identifier le plus de gènes possibles puis combinent ces différents résultats pour produire une annotation cohérente (96). En utilisant Prodigal (97) pour prédire les gènes, RNAmmer (98) pour les ARN ribosomaux (ARNr), Aragorn (99) pour les ARN de transfert (ARNt) et Infernal (100) pour les ARN non-codants, Prokka arrive à avoir une bonne vue d'ensemble du génome. L'annotation est également raffinée par l'utilisation de bases de données comme Pfam (101), TIGRFAM (102) et les RefSeq du NCBI (103).

B.1 – Organisation du génome de StachEndo

Le résultat final de l'annotation nous rapporte que le génome de StachEndo contiendrait 1 580 gènes. On compte au total 1 536 gènes protéiques et 44 codants pour des molécules d'ARN. En termes de gènes non-redondants, on compte 940 gènes protéiques et 23 gènes d'ARN. Lorsqu'on compare le génome de StachEndo à celui d'autres Alpha-Protéobactéries, endosymbiotes et vivant librement, on remarque que StachEndo se trouve généralement entre ces deux groupes. Au niveau de la taille du génome, la taille du génome de StachEndo est plus petite que celle des génomes de Rhodospirillaceae (104-111), mais se classe parmi les plus grands Rickettsiales (112-171). Similairement, StachEndo possède un nombre de gènes non-redondants plus important que les autres Rickettsiales, mais moindre que les bactéries vivant librement. Au total 87,6% du génome de StachEndo est codant, le tout regroupé en 932 opérons/unités transcriptionnelles (prédits par l'outil PathwayTools (172-174)).

Les ARN ribosomiques, ou ARNr, sont des composantes importantes des cellules de tout organisme puisqu'ils forment la partie ARN des ribosomes, ribonucléoprotéines responsables de la traduction de l'ARN en protéines. Les ribosomes sont composés de deux sous-unités ; la grande sous-unité 50S et la petite sous-unité 30S. La grande sous-unité regroupe les ARNr 5S et 23S, alors que la petite contient l'ARNr 16S. Puisque la présence des trois ARNr est

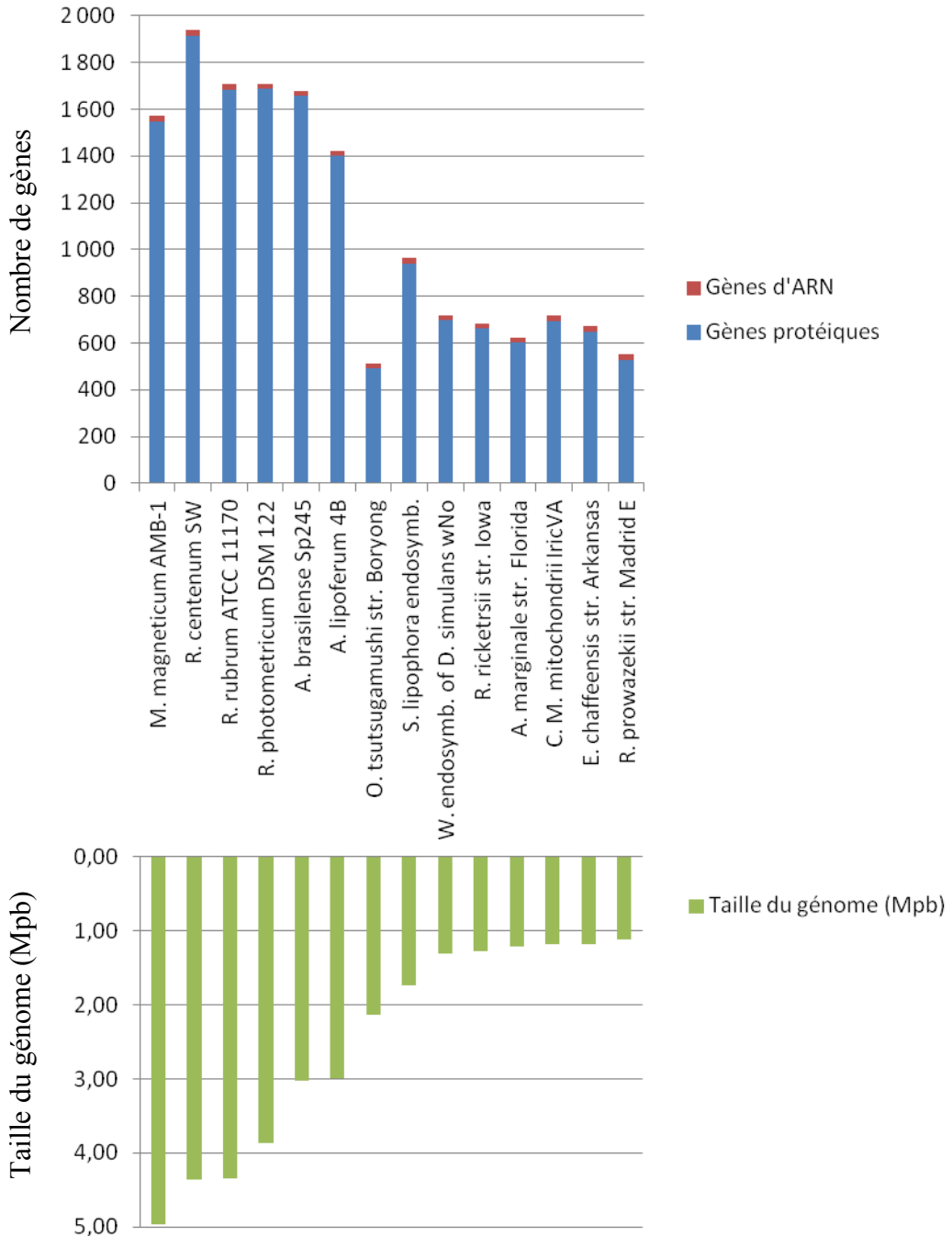


Figure 13. Comparaison de taille des génomes (bas) et du nombre de gènes (haut) de diverses espèces d'Alpha-Protéobactéries. (*Magnetospirillum magneticum* AMB-1 ref :(111); *Rhodospirillum centenum* SW ref :(104); *Rhodospirillum rubrum* ATCC 11170 ref :(106); *Rhodospirillum photometricum* DSM 122 ref :(105); *Azospirillum brasilense* Sp245 ref :(108); *Azospirillum lipoferum* 4B ref :(108); *Orientia tustsugamushi* Boryong ref :(152); Bactérie endosymbiote de *Stachyamoeba lipophora* (StachEndo); Wolbachia endosymbiote de *Drosophila simulans* wNo ref :(156); *Rickettsia rickettsii* Iowa ref :(145); *Anaplasma marginale* Florida ref :(167); *Midichloria mitochondrii* IricVa ref :(175); *Ehrlichia chaffeensis* Arkansas ref :(161); *Rickettsia prowazekii* Madrid E ref :(135))

nécessaire pour la formation d'un ribosome, les trois gènes appartiennent au même opéron. Ce qui assure que les trois gènes seront co-transcrits. Dans les opérons ribosomaux, les ARNr 5S et 23S sont côte à côte et sont séparés de l'ARNr 16S par deux ARNt. On retrouve cette conformation chez les espèces appartenant à la famille des Rhodospirillaceae (104-111). Il est également possible de retrouver des copies des gènes des ARNr ailleurs dans le génome et non sous forme d'opéron.

Par contre chez StachEndo, nous n'observons pas d'opéron ribosomal unique regroupant les trois ARNr. Nous observons plutôt un opéron regroupant les ARNr 5S et 23S et un autre opéron contenant l'ARNr 16S. Cette organisation est la même que celle retrouvée chez les Rickettsiales (112-171) (à l'exception des mitochondries, si incluses dans cet ordre, qui possèdent un seul opéron regroupant l'ARNr 16S et un ARNr 12S (176)).

Une autre caractéristique du génome de StachEndo qui semble intéressante est la répartition de ses ARNt. En théorie, le code génétique possède 61 codons représentant, de façon redondante, les 20 acides-aminés biologiques. Dans la réalité, la cellule n'a pas besoin d'avoir autant d'ARNt différents. L'ARN a la faculté de s'apparier de façon non-canonique permettant à un ARNt de s'apparier à un codon qui lui est complémentaire à l'exception du nucléotide à sa dernière position qui pourrait être différent du nucléotide à la première position de son anticodon. Un ARNt d'un même acide-aminé, mais d'un anticodon semblable, peut donc remplacer l'ARNt possédant l'anticodon exact. En théorie, compte tenu de la répartition des acides-aminés par rapport aux codons dans le code génétique bactérien, il faut un minimum de 24 ARNt (comme dans la plupart des mitochondries au travers des eucaryotes) pour pouvoir effectuer une traduction fidèle des ARN messagers (60).

Une analyse rapide nous a indiqué que les Alpha-Protéobactéries de la famille des Rhodospirillaceae possèdent en moyenne plus d'une quarantaine d'ARNt (104-111). Cependant, dans les bactéries endosymbiotes, sous l'effet de l'évolution réductive, un nombre important de gènes d'ARNt peut disparaître. Les endosymbiotes de la famille des Rickettsiales en possèdent plus d'une trentaine (112-171). Selon notre annotation, le génome de StachEndo contiendrait 37 ARNt, ce qui est similaire à ce que nous avons observé chez les Rickettsiales.

Tableau IV - Répartition des gènes d'ARNt annotés dans le génome de la bactérie endosymbiote StachEndo.

Acide-aminé	Nombre d'ARNt	Anticodons
Sérine	5	GCT, CAG, CGA, TGA, GGA
Leucine	4	TAA, TAG, CAA, GAG
Arginine	4	TCT, ACG, CCT, CCG
Méthionine	3	CAT
Thréonine	3	GGT, CGT, TGT
Alanine	2	GGC, TGC
Glycine	2	GCC, TCC
Valine	2	TAC, GAC
Asparagine	1	GTT
Acide aspartique	1	GTC
Cystéine	1	GCA
Glutamine	1	TTG
Acide glutamique	1	TTC
Histidine	1	GTG
Isoleucine	1	GAT
Lysine	1	TTT
Phénylalanine	1	GAA
Proline	1	TGG
Tryptophane	1	CCA
Tyrosine	1	GTA

Il y a au moins un ARNt pour chaque acide-aminé. Il est à noter que la sérine, la leucine et l'arginine font partie des acides-aminés possédant le plus d'ARNt d'anticodons différents (60). Ce phénomène peut être expliqué par le fait que dans le code génétique, plus de codons sont dédiés à ces acides-aminés qu'aux autres (6 codons contre 2 à 4 codons pour les autres (exceptés méthionine et tryptophane qui n'ont qu'un codon chacun)). En effet, la traduction de 6 codons ne peut pas être assurée par un seul anticodon (60).

Il est également intéressant de noter que StachEndo possède trois copies du même ARNt avec anticodon CAU, l'une pour la méthionine initiatrice, l'une pour l'élongation et la troisième qui est modifiée pour reconnaître les codons ATA (isoleucine) (177).

Nous remarquons que pour les acides-aminés ne possédant plus qu'un seul ARNt, l'anticodon conservé se termine toujours par G ou T. Ces nucléotides sont ceux qui permettent d'établir le plus d'appariements non-Watson-Crick (60). Les deux seules exceptions sont les ARNt de la méthionine et du tryptophane. Cependant, ces deux acides-aminés ne possèdent qu'un seul codon possible, il n'y a donc pas de sélection de codon possible.

B.2 – Analyse phylogénétique de StachEndo

Les gènes de StachEndo étant maintenant annotés, il devient possible de réaliser une analyse phylogénétique. De cette façon, nous serons en mesure de positionner StachEndo à l'intérieur de l'arbre évolutif des Alpha-Protéobactéries.

Pour ce faire, nous avons sélectionné 22 gènes protéiques d'origine mitochondriale largement présents chez la plupart des Alpha-Protéobactéries. Le choix de ces gènes nous permet d'inclure les mitochondries dans l'arbre. Ainsi, nous avons choisis 68 espèces différentes. Quarante-et-un sont des Alpha-Protéobactéries, dont 27, incluant StachEndo, sont des endosymbiotes. Les 27 autres espèces sont des mitochondries de divers eucaryotes. Le **Tableau V** présente la répartition des protéines choisies chez ces espèces.

Nous avons ensuite utilisé le modèle CAT-GTR (178) pour inférer la phylogénie de ces espèces. Nous avons laissé rouler le modèle jusqu'à la convergence de la probabilité de sa prior. L'arbre résultant est présenté à la **Figure 14**.

L'arbre inféré, possède un embranchement divisant les espèces présentées en deux grands groupes : les bactéries endosymbiotes de l'ordre des Rickettsiales et les bactéries vivant librement qui proviennent d'ordres variés. Chez les bactéries vivant librement, nous avons

souligné l'ordre des Rhodosprillales. C'est à cet ordre qu'appartiennent les espèces d'Azospirillum et de Magnetospirillum identifiées par RAST en raison de leurs similarités fonctionnelles avec StachEndo.

Tableau V - Distribution des gènes utilisés pour inféré l'arbre phylogénétique (Figure 14)

Nom des protéines	Symbole du gène associé	Nombre d'espèces possédant cette protéine (sur 68)
Sous-unité α de l'ATP synthéase	atp1	58
Sous-unité β de l'ATP synthéase	atp2	58
Sous-unité γ de l'ATP synthéase	atp3	50
Sous-unité 6 de l'ATP synthéase	atp6	66
Cytochrome b	cob	65
Sous-unité I de l'oxydase du cytochrome c	cox1	61
Sous-unité II de l'oxydase du cytochrome c	cox2	60
Sous-unité III de l'oxydase du cytochrome c	cox3	61
Protéine d'assemblage du cytochrome c	cox11	44
Sous-unité 1 de la déshydrogénase du NADH	nad1	64
Sous-unité 3 de la déshydrogénase du NADH	nad3	64
Sous-unité 4 de la déshydrogénase du NADH	nad4	66
Sous-unité 4L de la déshydrogénase du NADH	nad4L	33
Sous-unité 5 de la déshydrogénase du NADH	nad5	67
Sous-unité 7 de la déshydrogénase du NADH	nad7	58
Sous-unité 8 de la déshydrogénase du NADH	nad8	57
Sous-unité 9 de la déshydrogénase du NADH	nad9	60
Sous-unité 10 de la déshydrogénase du NADH	nad10	58
Sous-unité 11 de la déshydrogénase du NADH	nad11	56
Sous-unité 1 de la déshydrogénase du succinate	sdh1	56
Sous-unité 2 de la déshydrogénase du succinate	sdh2	61
Facteur d'élongation Tu	tufA	58

StachEndo, pour sa part, se positionne, sans surprise, dans la portion Rickettsiales/endosymbiotes de l'arbre. Les Rickettsiales sont composés de plusieurs familles dont les Rickettsiacea (comprenant les genus Rickettsia et Orientia) et les Anaplasmataceae (comprenant les genus Ehrlichia, Wolbachia, Anaplasma et Neorickettsia). C'est également au sein de ce groupe que se trouverait l'origine des mitochondries. Plusieurs espèces appartenant aux Rickettsiales n'ont pas encore de classification arrêtée (les Caedibacter, Paracoccus, Hepatobacter et Midichloria). L'arbre inféré positionne l'endosymbiote d'intérêt, StachEndo, à la base de la famille des Rickettsiaceae. La famille des Rickettsiaceae est cohérente avec la forte similarité fonctionnelle détectée entre StachEndo et le genus Rickettsia par RAST. Selon la topologie de l'arbre, StachEndo semble être le membre des Rickettsiaceae le plus près de leur ancêtre commun. StachEndo semble également plus proche de l'origine des Rickettsiales que beaucoup d'autres espèces, ce qui explique probablement les similarités observées avec les Anaplasmataceae et les Rhodospirillales à l'aide de RAST.

Pour ce qui est du reste de notre arbre, il respecte les phylogénies établies des Alpha-Protéobactéries, à l'exception des espèces appartenant à la famille des Rhodospirillales qui ne forment pas un groupe monophylétique dans notre arbre. Cet artéfact est probablement causé par le fait que notre arbre est basé uniquement sur les gènes d'origine mitochondriale, ce qui limite sa résolution. Il est à noter que les branches des sous-arbres sont assez courtes et que ces espèces sont malgré tout présentées comme étant phylogénétiquement rapprochées dans notre arbre.

Puisque le clade composé des génomes mitochondriaux forme une longue branche, il est possible que la divergence trop importante de ces séquences biaise l'arbre qui est inféré par attraction des longues branches. Nous avons donc retiré les mitochondries de notre jeu de données et procédé à une nouvelle inférence d'arbre. L'arbre sans mitochondrie présente la même disposition des espèces restantes. Il nous est donc possible de confirmer que l'attraction possible de la longue branche mitochondriale ne semble pas avoir biaisé de façon visible notre arbre.

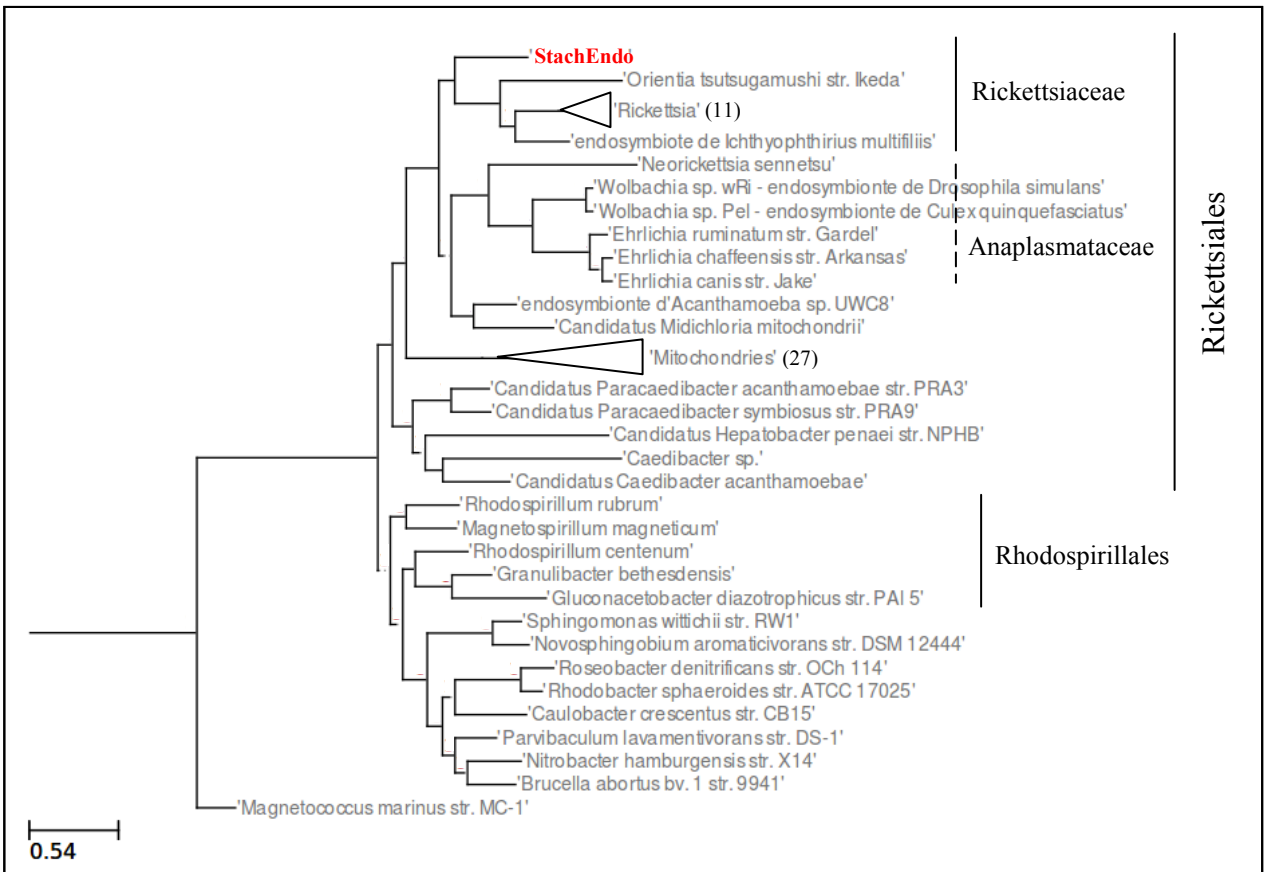


Figure 14. Arbre phylogénétique de 41 espèces d'Alpha-Protéobactéries généré à partir de 22 gènes protéiques d'origine mitochondriale (la bactérie endosymbiote d'intérêt StachEndo est indiquée en rouge).

B.3 – Différences métaboliques et fonctionnelles de StachEndo

Une fois l'annotation en main, nous avons utilisé PathwayTools pour assigner les protéines et les enzymes, prédites avec Prokka, à des voies métaboliques connues (172-174, 179). Grâce à cette analyse, il nous a été possible d'étudier, à un niveau métabolique, les différences importantes entre StachEndo et les autres Alpha-Protéobactéries qui en sont évolutivement proches (104-171). Puisque l'étude des Rickettsiales des genus Caedibacter, Paracoccus, Hepatobacter et Midichloria est plus récentes, PathwayTools ne disposait pas d'informations

métaboliques à leur égard. La comparaison de StachEndo avec les autres Rickettsiales se limite donc aux deux familles les mieux étudiés, Rickettsiaceae et Anaplasmataceae. Compte tenu de leur emplacement phylogénétique, ce sont les deux groupes les plus proches de StachEndo.

B.3.1 – Appareil flagellaire

L'appareil flagellaire est un mécanisme de propulsion présent sur les cellules de certaines espèces. Les flagelles commencent habituellement dans le cytoplasme où se trouve un complexe protéique servant de moteur (180). Elles se prolongent jusqu'à l'extérieur de la membrane externe de la cellule, formant le filament flagellaire, une sorte de queue mobile (180). Des flagelles ont été observés chez les Alpha-Protéobactéries et notamment dans l'ordre des Rhodospirillaceae (104-111, 180). La grande exception semble être les bactéries endosymbiotes (181). Jusqu'à maintenant des flagelles n'ont été observés chez aucune espèce de Rickettisa, d'Orientia, de Wolbachia ou d'Anaplasma (112-171). Cette observation est souvent expliquée par le fait qu'une bactérie endosymbiote obligatoire, comme le sont les membres des Rickettsiales, n'a plus besoin de se déplacer puisqu'elle vit exclusivement à l'intérieur de son hôte (181). Il n'y a donc plus d'avantages à posséder une telle machinerie. Sous la pression de l'évolution réductive, ces gènes inutiles devraient être éliminés.

En utilisant Pathway Tools, ainsi que des profils HMM provenant de PFAM spécifiques aux protéines flagellaires, nous avons pu identifier 35 gènes codant pour des protéines composant la structure du flagelle et son module d'exportation. Ce module permet d'exporter et d'assembler les protéines composant le flagelle à l'intérieur et au travers de la membrane cellulaire de la bactérie. Une étude de Liu et Ochman (2007) a observé que seuls 21 gènes de la cinquantaine de gènes flagellaires connus sont réellement essentiels à la formation de flagelles (180). Ces 21 gènes sont tous présents dans StachEndo.

A priori, il semblerait donc que StachEndo pourrait posséder des flagelles fonctionnels. StachEndo n'est pas le premier membre des Rickettsiales, groupe souvent défini comme composé de bactérie Gram négative sans flagelle, chez lequel un nombre important de gènes

flagellaires a été identifié (181). La bactérie *Midichloria mitochondrii,* par exemple, possède 26 gènes flagellaires mais n'a pas de flagelles (175). Selon notre analyse, il lui manquerait au minimum les gènes essentielles FlgB (tige proximale, élément du moteur) et FliQ (complexe d'exportation) (175).

Plus récemment, au moins cinq espèces de Rickettsiales possédant des flagelles ont été identifiées. Des flagelles immobiles ont été observés par microscopie chez *Lyticum flagellatum* et *Lyticum sinuosum,* deux endosymbiotes obligatoires de paramécies (182). Par contre, des flagelles mobiles ont été observés chez *Trichorickettisia mobilis* et *Gigarickettsia flagellata (183)*. Des expériences ont montré que ces endosymbiotes étaient incapables de se déplacer à l'intérieur du cytoplasme de leur hôte, mais étaient capables de le faire à l'intérieur du nucléoplasme dans des régions où la chromatine était moins dense (183). Puisqu'il s'agit de recherches récentes, les génomes de ces espèces n'ont pas été publiés, il nous est donc impossible de comparer les gènes flagellaires de ces espèces avec ceux de StachEndo.

Il semble donc fort possible que StachEndo possède des flagelles fonctionnels même s'il est un endosymbiote obligatoire. La seule façon de confirmer ou d'infirmer définitivement leur présence serait d'aller observer StachEndo *in vivo*. L'utilité de ces potentiels flagelles nous est inconnue. Aucune des études effectuées sur les autres endosymbiotes flagellés mentionnés précédemment n'a détecté d'autres fonctions que la motilité de l'endosymbiote. D'un point de vue évolutif, il s'agit d'une certaine confirmation du positionnement de StachEndo et de ces autres endosymbiotes. Ils se retrouvent entre les Alpha-Protéobactéries qui possèdent et utilisent toujours leurs flagelles pour se déplacer, et les Rickettsiales qui n'en possèdent pas, les ayant sûrement perdus sous l'influence de l'évolution réductive, puisqu'ils n'ont plus besoin de se déplacer.

B.3.2 – Voies métaboliques de production d'énergie

La production d'énergie chez les bactéries implique l'interaction de multiples voies métaboliques pour permettre la libération d'ATP. D'abord, la glycolyse permet la dégradation du glucose en pyruvate. En parallèle, la néoglucogenèse resynthétise du pyruvate à partir de glucose. Le pyruvate prend ensuite part au cycle de l'acide citrique (ou cycle de Krebs, cycle

de l'acide tricarboxylique (TCA cycle)) Ce cycle mène à la production de de NADH/NAPDH et de quinole utilisés comme donneurs d'électrons lors de la phosphorylation oxydative, menant à la phosphorylation d'ADP en ATP. Compte tenu du rôle primordial de ces voies, leurs enzymes sont présentes chez la plupart des bactéries et des eucaryotes (184, 185). Les Alpha-Protéobactéries vivant librement possèdent ces voies métaboliques de façon fonctionnelle (185). Cependant, puisque certaines bactéries endosymbiotes ont la faculté d'importer des composés chimiques tels que le pyruvate, le glucose ou même de l'ATP directement de leur hôte, il ne leur est plus essentiel de conserver les gènes des enzymes permettant de former ces voies (184-186).

Dans la famille des Rickettsiaceae, plusieurs espèces ont perdu partiellement ou totalement les enzymes permettant la glycolyse et la néoglucogenèse (112-153). Du côté des Anaplasmataceae, la plupart des enzymes requises sont présentes. La seule déviation semble être la substitution du glucose pour du fructose-1,6-bisphosphate comme substrat de la glycolyse et produit final de la néoglucogenèse (154-171).

La situation de StachEndo est identique à ce qui est observé chez les Anaplasmataceae. En effet, la presque totalité des enzymes nécessaires à la glycolyse et à néoglucogenèse sont présentes. La seule enzyme manquante est la 6-phosphofructokinase forçant probablement StachEndo à utiliser le fructose-1,6-bisphosphate plutôt que le glucose pour produire du pyruvate.

Au niveau du cycle de l'acide citrique, plusieurs variantes de ce dernier ont été observées chez les bactéries. La forme classique procaryote du cycle (voir **Figure 15.A**), telle que retrouvée chez les Rhodospirillaceae, a été prédite chez StachEndo. Elle serait également présente de façon dispersée chez les Rickettsiales et en particulier chez Wolbachia (117, 118, 121, 122, 124, 146, 147, 153, 154, 156-159, 163, 167). Les autres Rickettsiales étudiés possèdent une forme incomplète du cycle qui est tout de même capable de produire de l'ATP, de réduire des molécules de NADP+ en NADPH nécessaire à la phosphorylation oxydative. (voir **Figure 15.B**)

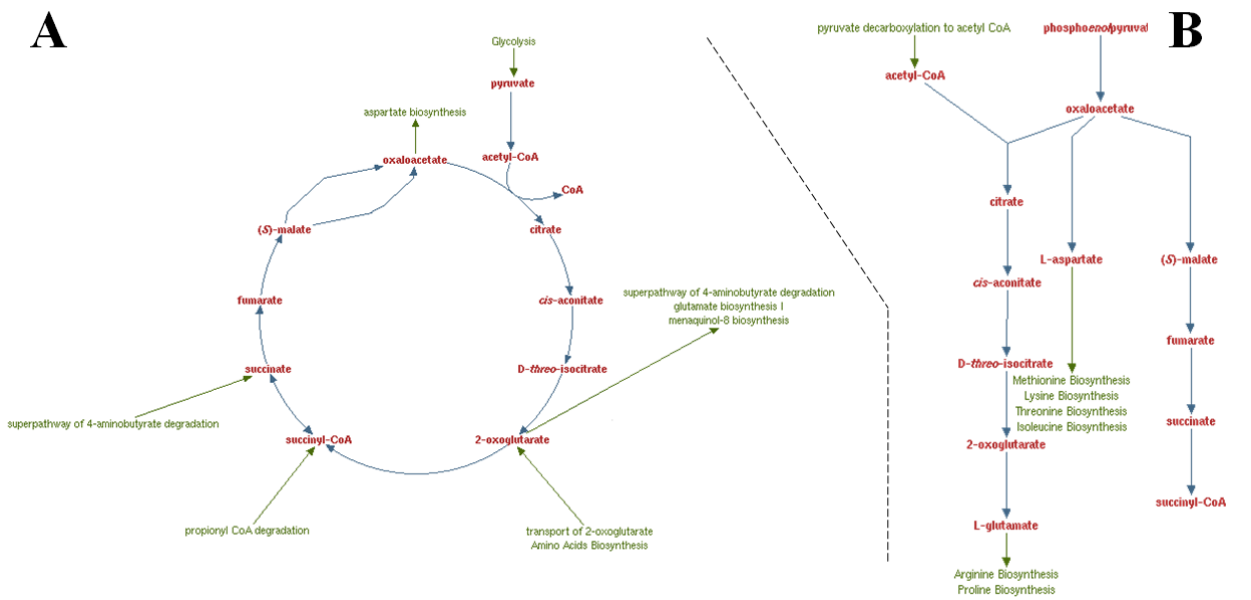


Figure 15. Schéma Pathway Tools du cycle de l'acide citrique bactérien classique (A) (*TCA cycle I: prokaryotic*) et de la cascade présente chez les endosymbiotes Rickettsiales (B) (*TCA cycle VI: obligate autotrophs*) (172-174, 179)**.**

Finalement, la chaîne de transport des électrons de phosphorylation oxydative est composée de cinq complexes protéiques. Ces cinq complexes semblent être présents chez toutes les espèces d'Alpha-Protéobactéries que nous avons analysées (104-171). Cette analyse a également montré que StachEndo possédait deux enzymes pouvant servir d'oxydase alternative à la chaîne de transport. D'abord l'oxydase alternative AOX, capable de transférer directement les électrons du complexe I, via l'ubiquinole, jusqu'aux molécules d'O2 sans passer par les complexes III et IV. Cette enzyme semble absente des Rhodospirillaceae et des Rickettsiales observés (104-171). Cependant, elle fut observée chez d'autres Alpha-Protéobactéries et Protéobactéries (187, 188). La distribution très irrégulière de AOX au travers de l'évolution laisse croire qu'il pourrait s'agirait d'un gène qui serait apparu chez plusieurs espèces de bactéries par transfert horizontal (187, 188).

La deuxième oxydase alternative est l'oxydase cytochrome bd-II d'ubiquinole capable de

transférer directement les électrons de molécules de quinole aux molécules d'O2 (présente également chez les Rhodospirillaceae et les Rickettsiaceae). Ces réactions chimiques alternatives offriraient une source d'ATP continue à la bactérie en cas d'inhibition de différents complexes de la chaîne de transport. StachEndo et les autres Rickettsiales possèdent également des translocases d'ATP leur permettant d'importer de l'ATP directement de leur hôte. De cette façon, il ne leur est pas nécessaire de le produire eux-mêmes.

Selon cette annotation, StachEndo possède des capacités de production d'énergie qui sont beaucoup plus complètes que les autres Rickettsiaceae, se rapprochant des capacités des Anaplasmataceae. Ces similarités pourraient être expliqués par le positionnement de StachEndo au sommet des Rickettsiaceae et donc à proximité des Anaplasmataceae.

B.3.3 - Biosynthèse des composés

B.3.3.1 - Synthèse des acides-aminés

La synthèse des acides-aminés est une voie métabolique importante pour un organisme. Les bactéries vivant librement comme les Alpha-Protéobactéries de la famille des Rhodospirillaceae sont capables de synthétiser tous les acides-aminés canoniques à partir de produits rejetés par les voies de production d'ATP. Cependant, les bactéries endosymbiotes, ayant potentiellement perdues certaines de ces voies et étant capables d'importer les acides-aminés de leur hôte, peuvent perdre les gènes leur permettant de synthétiser *de novo* ces molécules, devenant ainsi dépendantes de leur hôte.

Les synthèses de la proline, de l'asparagine, de la méthionine, de la thréonine, de la leucine, de l'isoleucine, de la cystéine, de la valine et de l'histidine semblent impossibles chez tous les Rickettsiales étudiés (112-171). Inversement, les synthèses de la lysine et de l'acide aspartique semblent possibles chez la plupart des Rickettsiales. Les conversions de cystéine en alanine, de sérine en glycine et de glutamate en acide glutamique sont également possible chez tous les Rickettsiales étudiés. Par contre, la production de glutamate ne semble possible que chez les Rickettsia et StachEndo (112-151). La synthèse de l'arginine à partir du glutamate est présente chez des espèces d'Ehrlichia et un nombre important des enzymes impliquées est conservé au

travers des Anaplasmataceae (154-171). Elle est cependant complètement absente chez les Rickettsiaceae, incluant StachEndo (112-153).

La voie métabolique, qui permet la synthèse des acide-aminés aromatiques, est l'endroit ou StachEndo se démarque complètement des autres espèces de Rickettsiales étudiés. La synthèse de ces acides-aminés se fait en passant par la voie métabolique du shikimate puis par la voie du chorismate (189). C'est à partir du chorismate que la voie métabolique se divise pour synthétiser d'un côté la phénylalanine et la tyrosine et de l'autre le tryptophane en présence de sérine (189). Chez tous les Rickettsiales que nous avons observés, ces voies métaboliques sont absentes. La seule exception est StachEndo qui possède l'entièreté des voies métaboliques du shikimate et du chorismate. Bien que StachEndo ne possède pas les enzymes permettant la synthèse de la phénylalanine, de la tyrosine et du tryptophane, certaines enzymes permettant de convertir le chorismate en précurseurs du tryptophane sont présentes (différentes composantes de la synthase de l'anthranilate (EC 4.1.3.27/EC 2.4.2.18)).

En résumé, l'analyse effectuée montre que StachEndo a conservé un certain nombre de gènes lui permettant de synthétiser des acides-aminés. Cependant, plus des deux-tiers des acide-aminés classiques devront être importés de *Stachyamoeba lipophora.* StachEndo possède une gamme assez importante de gènes de synthèse des acides-aminés lorsque comparée aux autres espèces de Rickettsiales. Il est également intéressant de remarquer qu'au niveau des Rickettsiales, les voies précurseurs du shikimate et du chorismate ne semblent présentes que chez StachEndo.

Toutes ces analyses montrent que StachEndo a perdu la majorité de gènes associés à la biosynthèse *de novo* des acides-aminés, ce qui confirme la dépendance nécessaire et le caractère obligatoire de la symbiose existant entre StachEndo et *Stachyamoeba lipophora.* Cependant, la présence de plusieurs enzymes, disparues chez les autres Rickettsiales, appartenant à des voies métaboliques devenues non-fonctionnelles, concorde avec le positionnement de StachEndo dans l'arbre phylogénétique. Sa position au sommet des Rickettsiaceae est reflétée par la présence de vestiges importants de voies métaboliques provenant de ces ancêtres vivant librement, alors que les autres endosymbiotes les ont perdus.

B.3.3.2 - Synthèse des acides nucléiques

La synthèse des acides nucléiques est une autre voie déterminante pour une espèce. Les acides nucléiques composent l'ADN et l'ARN. La capacité d'une espèce endosymbiote de synthétiser elle-même les molécules composant sont matériel génétique, ainsi que toutes les machineries à base d'ARN nous renseigne énormément sur son autonomie vis-à-vis de son hôte.

Chez les Anaplasmataceae, la synthèse des pyrimidines et des purines est possible (154-171). Chez les Rickettsiaceae, malgré la présence de diverses enzymes permettant la phosphorylation des nucléotides monophosphates en nucléotides triphosphates, la synthèse des nucléotides est impossible (112-153).

StachEndo se démarque des autres Rickettsiaceae en possédant, comme les Anaplasmataceae, une voie de synthèse des pyrimidines complète. Par contre, même s'il possède plus d'enzymes appartenant à cette voie que les autres Rickettsiaceae, la synthèse des purines reste impossible pour StachEndo. Il peut être intéressant de mentionner que StachEndo possède les gènes permettant la conversion des ribonucléosides en désoxyribonucléosides. La conversion de dCTP en TTP est également possible.

En résumé, StachEndo semble parfaitement capable de synthétiser ses propres pyrimidines, mais semble incapable de synthétiser ses propres purines. StachEndo dépend donc de *Stachyamoeba lipophora* duquel il peut importer des purines à l'aide de translocases. Bien qu'en comparaison des Rickettsiales de la famille des Anaplasmataceae, StachEndo possède des capacités de synthèse des acides nucléiques plus restreintes, sa capacité à synthétiser des pyrimidines lui permet de se démarquer des autres Rickettsiaceae.

B.3.3.3 - Synthèse de la paroi cellulaire

La paroi cellulaire des Alpha-Protéobactéries vivant librement est composée de plusieurs couches successives : d'abord, la membrane cytoplasmique composée de deux couches de

phospholipides, ensuite, une couche de peptidoglycanes et finalement, la membrane externe, composée d'une couche de lipopolysaccharides à l'intérieure et d'une couche de phospholipides à l'extérieur (190).

Au niveau de la synthèse des phospholipides et des peptidoglycanes, la plupart des Rickettsiaceae que nous avons analysés, incluant StachEndo semblent capables de produire plusieurs type de phospholipides. Par contre, chez les Anaplasmataceae, la synthèse de peptidoglycanes est impossible chez les Ehrlichia et plusieurs espèces d'Anaplasma (160-164, 169-171). Les Wolbachia semblent posséder des voies fonctionnelles (154-159). La synthèse de lipopolysaccharides leur est également impossible possible (154-171).

Chez les Rickettsiaceae, la présence des voies permettant la synthèse des lipopolysaccharides varie d'une espèce à une autre (112-153). La plupart des enzymes de cette voie présentes chez les Rhodospirillaceae sont également chez StachEndo. Il semble donc probable que StachEndo soit toujours capable de synthétiser des lipopolysaccharides.

Nos observations semblent nous indiquer que StachEndo est toujours en mesure de synthétiser les biomolécules composant les différentes couches de la paroi cellulaire classique des bactéries Gram négative, ce qui, d'après nos observations, n'est pas conservé chez tous les Rickettsiales (112-171).

Finalement, nous avons observé, chez StachEndo, la présence d'enzymes composant une voie métabolique qui semble complètement absente chez les autres Rickettsiales, ainsi que chez les Rhodospirillaceae. Il s'agit de la voie métabolique de résistance aux polymyxines. Les polymyxines sont des antibiotiques de bactéries Gram négative. Les polymyxines interagissent avec les lipides A formant les lipopolysaccharides ce qui interfère avec la structure de la membrane externe des bactéries, la rendant plus vulnérable. Ultimement, cette déstabilisation de la membrane mène à la mort de la bactérie.

StachEndo possèdent une série de gènes (arnA, arnBm arnC et arnT. une formyltransférase/décarboxylase, une transaminase, et deux transférases respectivement) qui

permet de modifier la structure des lipides A, empêchant la fixation de polymyxines, mais leur permettant toujours de former la membrane externe. L'apparition de cette résistance est fort intéressante. Puisqu'elle ne semble pas présente chez les espèces proches d'Alpha-Protéobactéries vivant librement, il est possible que StachEndo l'ait obtenue par transfert horizontal.

En conclusion, on remarque que les voies métaboliques de StachEndo semblent, dans l'ensemble, plus complètes que celles des Anaplasmataceae et des autres Rickettsiaceae. Cette observation, ainsi que la présence de certaines enzymes absentes chez les autres Rickettsiales étudiés (mais présentes chez les bactéries vivant librement) ajoute de la crédibilité à l'hypothèse selon laquelle StachEndo pourrait être un jeune endosymbiote puisqu'il présente des caractéristiques habituellement perdues chez les endosymbiotes. De plus l'emplacement de StachEndo près de la base des Rickettsiaceae semble également ajouter de la crédence à cette hypothèse.

4 - CONCLUSION

Ce projet avait pour but le développement de méthodes d'assemblage *de novo* adaptées aux génomes bactériens, à l'aide de données de séquençage de nouvelle génération. Après avoir testé de multiples combinaisons d'approches et de jeux de données, l'utilisation de lectures de types Illumina (qui se sont montrées plus justes que les données de type 454) avec des assembleurs de type graphe DeBruijn (les mieux adaptés aux lectures courtes comme Illumina) s'est révélée l'approche la plus efficace, formant les contigs les plus justes.

Le second but de ce projet était d'utiliser les approches que nous aurions développées pour assembler, puis analyser le génome d'une Alpha-Protéobactérie endosymbiote inconnue, surnommé StachEndo, vivant de façon co-dépendante avec son hôte, l'amibe terrestre *Stachyamoeba lipophora*. En utilisant, l'assembleur DBG Velvet avec des lectures Illumina HiSEQ, il avait été possible d'obtenir un assemblage du génome de StachEndo et de la mitochondrie de *Stachyamoeba lipophora* composé de 35 contigs. L'assemblage était probablement perturbé par la présence de répétitions plus longues que les lectures Illumina. L'ajout de lectures Illumina MiSEQ *paired-end overlap*, qui grâce à leur chevauchement forment des lectures plus longues, a permis de simplifier cet assemblage. La réinjection dans l'assembleur de contigs de bonne qualité aux côtés de lectures plus longues fut essentiel pour compléter l'assemblage du génome de StachEndo.

Il est difficile d'évaluer de façon précise la qualité de cet assemblage de StachEndo. Cependant, l'intégrité du génome mitochondrial assemblé en parallèle, ainsi que l'annotation de gènes complets semble indiquer qu'il s'agit, au minimum, d'une ébauche assez complète. Pour s'assurer de la structure du génome, il pourrait être intéressant de refaire séquencer les portions qui possèdent une faible couverture ou qui flanquent de grandes répétitions (ou de les valider par PCR). De cette façon, il serait possible de vérifier la justesse de l'assemblage à ces positions sensibles.

L'utilisation des deux librairies pairées permit l'obtention du génome de StachEndo en un seul contig circulaire de 1 731 915 pb. En termes de gènes non-redondants, on y retrouve 940

gènes protéiques et 23 gènes d'ARN, faisant de StachEndo un des génomes de Rickettsiales observés les plus riches. Cette richesse (se reflétant, également, par la présence prédite de voies métaboliques absentes chez d'autres Rickettsiales) et son positionnement phylogénétique près du sommet des familles les mieux caractérisées des Rickettsiales semblent indiquer un passé non-endosymbiote relativement récent (lorsque comparé aux autres Rickettsiaceae par exemple).

Avec ce génome en main, il est maintenant possible d'affirmer que StachEndo est une bactérie jusqu'à présent non répertoriée. Nous proposons donc de la nommer ***Endostachyamoeba necessaria***. Le préfixe grec "*endo*-", signifiant à l'intérieur, combiné au nom "*stachyamoeba*", indique que cette bactérie est un endosymbiote d'un organisme du genus Stachyamoeba. Le nom d'espèce "*necessaria*", est un adjectif latin signifiant "indispensable". Il est utilisé ici pour décrire le caractère obligatoire de la relation unissant cette bactérie à son hôte, puisqu'aucun des deux ne peux survivre sans l'autre.

5 - RÉFÉRENCES

1. Metzker ML (2010) Sequencing technologies - the next generation. (Translated from eng) *Nat Rev Genet* 11(1):31-46 (in eng).
2. Schuster SC (2008) Next-generation sequencing transforms today's biology. (Translated from eng) *Nat Methods* 5(1):16-18 (in eng).
3. Sanger F, Nicklen S, & Coulson AR (1977) DNA sequencing with chain-terminating inhibitors. (Translated from eng) *Proc Natl Acad Sci U S A* 74(12):5463-5467 (in eng).
4. Bentley DR*, et al.* (2008) Accurate whole human genome sequencing using reversible terminator chemistry. (Translated from eng) *Nature* 456(7218):53-59 (in eng).
5. Rothberg JM*, et al.* (2011) An integrated semiconductor device enabling non-optical genome sequencing. (Translated from eng) *Nature* 475(7356):348-352 (in eng).
6. IonTorrent (2013) The Ion PGM System, with 400-base read length chemistry, enables routine high-quality de novo assembly of small genomes. in *APPLICATION NOTE - Ion PGM small genome sequencing*, ed Corporation LT, pp 1-6.
7. Illumina (2015) MiSeq System, Focused power. Speed and simplicity for targeted and small-genome sequencing. in *System Specification Sheet : Sequencing*, ed Illumina I, pp 1-4.
8. Margulies M*, et al.* (2005) Genome sequencing in microfabricated high-density picolitre reactors. (Translated from eng) *Nature* 437(7057):376-380 (in eng).
9. Eid J*, et al.* (2009) Real-time DNA sequencing from single polymerase molecules. (Translated from eng) *Science* 323(5910):133-138 (in eng).
10. Pop M (2009) Genome assembly reborn: recent computational challenges. (Translated from eng) *Brief Bioinform* 10(4):354-366 (in eng).
11. Chen YC, Liu T, Yu CH, Chiang TY, & Hwang CC (2013) Effects of GC bias in next-generation-sequencing data on de novo genome assembly. (Translated from eng) *PLoS One* 8(4):e62856 (in eng).
12. Hillier LW*, et al.* (2008) Whole-genome sequencing and variant discovery in C. elegans. (Translated from eng) *Nat Methods* 5(2):183-188 (in eng).
13. Dohm JC, Lottaz C, Borodina T, & Himmelbauer H (2008) Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. (Translated from eng) *Nucleic Acids Res* 36(16):e105 (in eng).
14. Luo C, Tsementzi D, Kyrpides N, Read T, & Konstantinidis KT (2012) Direct comparisons of Illumina vs. Roche 454 sequencing technologies on the same microbial community DNA sample. (Translated from eng) *PLoS One* 7(2):e30087 (in eng).
15. Gilles A*, et al.* (2011) Accuracy and quality assessment of 454 GS-FLX Titanium pyrosequencing. (Translated from eng) *BMC Genomics* 12:245 (in eng).
16. Huse SM, Huber JA, Morrison HG, Sogin ML, & Welch DM (2007) Accuracy and quality of massively parallel DNA pyrosequencing. (Translated from eng) *Genome Biol* 8(7):R143 (in eng).
17. Nakamura K*, et al.* (2011) Sequence-specific error profile of Illumina sequencers. (Translated from eng) *Nucleic Acids Res* 39(13):e90 (in eng).
18. Hoffmann S*, et al.* (2009) Fast mapping of short sequences with mismatches, insertions and deletions using index structures. (Translated from eng) *PLoS Comput Biol* 5(9):e1000502 (in eng).

19. Tsai IJ, *et al.* (2014) Summarizing specific profiles in Illumina sequencing from whole-genome amplified DNA. (Translated from eng) *DNA Res* 21(3):243-254 (in eng).
20. Staden R (1979) A strategy of DNA sequencing employing computer programs. (Translated from eng) *Nucleic Acids Res* 6(7):2601-2610 (in eng).
21. Nagarajan N & Pop M (2013) Sequence assembly demystified. (Translated from eng) *Nat Rev Genet* 14(3):157-167 (in eng).
22. Paszkiewicz K & Studholme DJ (2010) De novo assembly of short sequence reads. (Translated from eng) *Brief Bioinform* 11(5):457-472 (in eng).
23. Myers EW, *et al.* (2000) A whole-genome assembly of Drosophila. (Translated from eng) *Science* 287(5461):2196-2204 (in eng).
24. Chevreux B, Wetter, T. and Suhai,S. (1999) Genome Sequence Assembly Using Trace Signals and Additional Sequence Information. *Computer Science and Biology: Proceedings of the German Conference on Bioinformatics* (99):11.
25. Batzoglou S, *et al.* (2002) ARACHNE: a whole-genome shotgun assembler. (Translated from eng) *Genome Res* 12(1):177-189 (in eng).
26. Compeau PE, Pevzner PA, & Tesler G (2011) How to apply de Bruijn graphs to genome assembly. (Translated from eng) *Nat Biotechnol* 29(11):987-991 (in eng).
27. Zerbino DR & Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. (Translated from eng) *Genome Res* 18(5):821-829 (in eng).
28. Pevzner PA, Tang H, & Waterman MS (2001) An Eulerian path approach to DNA fragment assembly. (Translated from eng) *Proc Natl Acad Sci U S A* 98(17):9748-9753 (in eng).
29. Miller JR, Koren S, & Sutton G (2010) Assembly algorithms for next-generation sequencing data. (Translated from eng) *Genomics* 95(6):315-327 (in eng).
30. Pevzner PA (1989) 1-Tuple DNA sequencing: computer analysis. (Translated from eng) *J Biomol Struct Dyn* 7(1):63-73 (in eng).
31. Luo R, *et al.* (2012) SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. (Translated from eng) *Gigascience* 1(1):18 (in eng).
32. Simpson JT, *et al.* (2009) ABySS: a parallel assembler for short read sequence data. (Translated from eng) *Genome Res* 19(6):1117-1123 (in eng).
33. Zerbino DR, McEwen GK, Margulies EH, & Birney E (2009) Pebble and rock band: heuristic resolution of repeats and scaffolding in the velvet short-read de novo assembler. (Translated from eng) *PLoS One* 4(12):e8407 (in eng).
34. Li R, *et al.* (2010) De novo assembly of human genomes with massively parallel short read sequencing. (Translated from eng) *Genome Res* 20(2):265-272 (in eng).
35. Mardis E, McPherson J, Martienssen R, Wilson RK, & McCombie WR (2002) What is finished, and why does it matter. (Translated from eng) *Genome Res* 12(5):669-671 (in eng).
36. Nagarajan N, *et al.* (2010) Finishing genomes with limited resources: lessons from an ensemble of microbial genomes. (Translated from eng) *BMC Genomics* 11:242 (in eng).
37. Boetzer M, Henkel CV, Jansen HJ, Butler D, & Pirovano W (2011) Scaffolding pre-assembled contigs using SSPACE. (Translated from eng) *Bioinformatics* 27(4):578-579 (in eng).
38. Tsai IJ, Otto TD, & Berriman M (2010) Improving draft assemblies by iterative

mapping and assembly of short reads to eliminate gaps. (Translated from eng) *Genome Biol* 11(4):R41 (in eng).
39. Tang S, Gong Y, & Edwards EA (2012) Semi-automatic in silico gap closure enabled de novo assembly of two Dehalobacter genomes from metagenomic data. (Translated from eng) *PLoS One* 7(12):e52038 (in eng).
40. Fondi M*, et al.* (2014) Enly: Improving Draft Genomes through Reads Recycling. (Translated from eng) *J Genomics* 2:89-93 (in eng).
41. Boetzer M & Pirovano W (2012) Toward almost closed genomes with GapFiller. (Translated from eng) *Genome Biol* 13(6):R56 (in eng).
42. Galardini M, Biondi EG, Bazzicalupo M, & Mengoni A (2011) CONTIGuator: a bacterial genomes finishing tool for structural insights on draft genomes. (Translated from eng) *Source Code Biol Med* 6:11 (in eng).
43. Piro VC*, et al.* (2014) FGAP: an automated gap closing tool. (Translated from eng) *BMC Res Notes* 7:371 (in eng).
44. Koren S, Miller JR, Walenz BP, & Sutton G (2010) An algorithm for automated closure during assembly. (Translated from eng) *BMC Bioinformatics* 11:457 (in eng).
45. Salzberg SL*, et al.* (2012) GAGE: A critical evaluation of genome assemblies and assembly algorithms. (Translated from eng) *Genome Res* 22(3):557-567 (in eng).
46. Zhao X*, et al.* (2010) EDAR: an efficient error detection and removal algorithm for next generation sequencing data. (Translated from eng) *J Comput Biol* 17(11):1549-1560 (in eng).
47. Salmela L & Schroder J (2011) Correcting errors in short reads by multiple alignments. (Translated from eng) *Bioinformatics* 27(11):1455-1461 (in eng).
48. Nikolenko SI, Korobeynikov AI, & Alekseyev MA (2013) BayesHammer: Bayesian clustering for error correction in single-cell sequencing. (Translated from eng) *BMC Genomics* 14 Suppl 1:S7 (in eng).
49. Lim EC*, et al.* (2014) Trowel: a fast and accurate error correction module for Illumina sequencing reads. (Translated from eng) *Bioinformatics* 30(22):3264-3265 (in eng).
50. Greenfield P, Duesing K, Papanicolaou A, & Bauer DC (2014) Blue: correcting sequencing errors using consensus and context. (Translated from eng) *Bioinformatics* 30(19):2723-2732 (in eng).
51. Song L, Florea L, & Langmead B (2014) Lighter: fast and memory-efficient sequencing error correction without counting. (Translated from eng) *Genome Biol* 15(11):509 (in eng).
52. Medvedev P, Scott E, Kakaradov B, & Pevzner P (2011) Error correction of high-throughput sequencing datasets with non-uniform coverage. (Translated from eng) *Bioinformatics* 27(13):i137-141 (in eng).
53. Kelley DR, Schatz MC, & Salzberg SL (2010) Quake: quality-aware detection and correction of sequencing errors. (Translated from eng) *Genome Biol* 11(11):R116 (in eng).
54. Kao WC, Chan AH, & Song YS (2011) ECHO: a reference-free short-read error correction algorithm. (Translated from eng) *Genome Res* 21(7):1181-1192 (in eng).
55. Bankevich A*, et al.* (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. (Translated from eng) *J Comput Biol* 19(5):455-477 (in eng).
56. Quail MA*, et al.* (2008) A large genome center's improvements to the Illumina

sequencing system. (Translated from eng) *Nat Methods* 5(12):1005-1010 (in eng).
57. Lang BF (2014) Mitochondria and the origin of eukaryotes. *Endosymbiosis*, ed Löffelhardt W (Springer, Vienna), pp 3-18.
58. Margulis L & Bermudes D (1985) Symbiosis as a mechanism of evolution: status of cell symbiosis theory. (Translated from eng) *Symbiosis* 1:101-124 (in eng).
59. Khachane AN, Timmis KN, & Martins dos Santos VA (2007) Dynamics of reductive genome evolution in mitochondria and obligate intracellular microbes. (Translated from eng) *Mol Biol Evol* 24(2):449-456 (in eng).
60. Andersson SG & Kurland CG (1998) Reductive evolution of resident genomes. (Translated from eng) *Trends Microbiol* 6(7):263-268 (in eng).
61. Degli Esposti M*, et al.* (2014) Evolution of mitochondria reconstructed from the energy metabolism of living bacteria. (Translated from eng) *PLoS One* 9(5):e96566 (in eng).
62. Wixon J (2001) Featured organism: reductive evolution in bacteria: Buchnera sp., Rickettsia prowazekii and Mycobacterium leprae. (Translated from eng) *Comp Funct Genomics* 2(1):44-48 (in eng).
63. Gray MW, Burger G, & Lang BF (1999) Mitochondrial evolution. (Translated from eng) *Science* 283(5407):1476-1481 (in eng).
64. Williams KP, Sobral BW, & Dickerman AW (2007) A robust species tree for the alphaproteobacteria. (Translated from eng) *J Bacteriol* 189(13):4578-4586 (in eng).
65. Abhishek A, Bavishi A, & Choudhary M (2011) Bacterial genome chimaerism and the origin of mitochondria. (Translated from eng) *Can J Microbiol* 57(1):49-61 (in eng).
66. Thiergart T, Landan G, Schenk M, Dagan T, & Martin WF (2012) An evolutionary network of genes present in the eukaryote common ancestor polls genomes on eukaryotic and mitochondrial origin. (Translated from eng) *Genome Biol Evol* 4(4):466-485 (in eng).
67. Chang X, Wang Z, Hao P, Li YY, & Li YX (2010) Exploring mitochondrial evolution and metabolism organization principles by comparative analysis of metabolic networks. (Translated from eng) *Genomics* 95(6):339-344 (in eng).
68. Andersson SG, Karlberg O, Canback B, & Kurland CG (2003) On the origin of mitochondria: a genomics perspective. (Translated from eng) *Philos Trans R Soc Lond B Biol Sci* 358(1429):165-177; discussion 177-169 (in eng).
69. Gray MW, Burger G, & Lang BF (2001) The origin and early evolution of mitochondria. (Translated from eng) *Genome Biol* 2(6):REVIEWS1018 (in eng).
70. Wang Z & Wu M (2015) An integrated phylogenomic approach toward pinpointing the origin of mitochondria. (Translated from eng) *Sci Rep* 5:7949 (in eng).
71. Allen AC & Spitz S (1945) A Comparative Study of the Pathology of Scrub Typhus (Tsutsugamushi Disease) and Other Rickettsial Diseases. (Translated from eng) *Am J Pathol* 21(4):603-681 (in eng).
72. Mariconti M*, et al.* (2012) A study on the presence of flagella in the order Rickettsiales: the case of 'Candidatus Midichloria mitochondrii'. *Microbiology* 158(Pt 7):1677-1683.
73. Sommer DD, Delcher AL, Salzberg SL, & Pop M (2007) Minimus: a fast, lightweight genome assembler. (Translated from eng) *BMC Bioinformatics* 8:64 (in eng).
74. Gordon D, Abajian C, & Green P (1998) Consed: a graphical tool for sequence finishing. (Translated from eng) *Genome Res* 8(3):195-202 (in eng).

75. Boisvert S, Laviolette F, & Corbeil J (2010) Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies. (Translated from eng) *J Comput Biol* 17(11):1519-1533 (in eng).
76. Butler J*, et al.* (2008) ALLPATHS: de novo assembly of whole-genome shotgun microreads. (Translated from eng) *Genome Res* 18(5):810-820 (in eng).
77. Ewing B, Hillier L, Wendl MC, & Green P (1998) Base-calling of automated sequencer traces using phred. I. Accuracy assessment. (Translated from eng) *Genome Res* 8(3):175-185 (in eng).
78. Ewing B & Green P (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. (Translated from eng) *Genome Res* 8(3):186-194 (in eng).
79. Reilly BD, Schlipalius DI, Cramp RL, Ebert PR, & Franklin CE (2013) Frogs and estivation: transcriptional insights into metabolism and cell survival in a natural model of extended muscle disuse. (Translated from eng) *Physiol Genomics* 45(10):377-388 (in eng).
80. Schulz MH, Zerbino DR, Vingron M, & Birney E (2012) Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels. (Translated from eng) *Bioinformatics* 28(8):1086-1092 (in eng).
81. Earl D*, et al.* (2011) Assemblathon 1: a competitive assessment of de novo short read assembly methods. (Translated from eng) *Genome Res* 21(12):2224-2241 (in eng).
82. Bradnam KR*, et al.* (2013) Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. (Translated from eng) *Gigascience* 2(1):10 (in eng).
83. Kajitani R*, et al.* (2014) Efficient de novo assembly of highly heterozygous genomes from whole-genome shotgun short reads. (Translated from eng) *Genome Res* 24(8):1384-1395 (in eng).
84. Narzisi G & Mishra B (2011) Comparing de novo genome assembly: the long and short of it. (Translated from eng) *PLoS One* 6(4):e19175 (in eng).
85. Aziz RK*, et al.* (2008) The RAST Server: rapid annotations using subsystems technology. (Translated from eng) *BMC Genomics* 9:75 (in eng).
86. Delcher AL*, et al.* (1999) Alignment of whole genomes. (Translated from eng) *Nucleic Acids Res* 27(11):2369-2376 (in eng).
87. Kurtz S*, et al.* (2004) Versatile and open software for comparing large genomes. (Translated from eng) *Genome Biol* 5(2):R12 (in eng).
88. Gordon D & Green P (2013) Consed: a graphical editor for next-generation sequencing. (Translated from eng) *Bioinformatics* 29(22):2936-2937 (in eng).
89. Gordon D, Desmarais C, & Green P (2001) Automated finishing with autofinish. (Translated from eng) *Genome Res* 11(4):614-625 (in eng).
90. Bolger AM, Lohse M, & Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. (Translated from eng) *Bioinformatics* 30(15):2114-2120 (in eng).
91. Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. *Journal EMBnet* 17(1).
92. Magoc T & Salzberg SL (2011) FLASH: fast length adjustment of short reads to improve genome assemblies. (Translated from eng) *Bioinformatics* 27(21):2957-2963 (in eng).
93. Schmieder R & Edwards R (2011) Quality control and preprocessing of metagenomic

datasets. (Translated from eng) *Bioinformatics* 27(6):863-864 (in eng).
94. Delcher AL, Harmon D, Kasif S, White O, & Salzberg SL (1999) Improved microbial gene identification with GLIMMER. (Translated from eng) *Nucleic Acids Res* 27(23):4636-4641 (in eng).
95. Salzberg SL, Delcher AL, Kasif S, & White O (1998) Microbial gene identification using interpolated Markov models. (Translated from eng) *Nucleic Acids Res* 26(2):544-548 (in eng).
96. Seemann T (2014) Prokka: rapid prokaryotic genome annotation. (Translated from eng) *Bioinformatics* 30(14):2068-2069 (in eng).
97. Hyatt D*, et al.* (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. (Translated from eng) *BMC Bioinformatics* 11:119 (in eng).
98. Lagesen K*, et al.* (2007) RNAmmer: consistent and rapid annotation of ribosomal RNA genes. (Translated from eng) *Nucleic Acids Res* 35(9):3100-3108 (in eng).
99. Laslett D & Canback B (2004) ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. (Translated from eng) *Nucleic Acids Res* 32(1):11-16 (in eng).
100. Nawrocki EP & Eddy SR (2013) Infernal 1.1: 100-fold faster RNA homology searches. (Translated from eng) *Bioinformatics* 29(22):2933-2935 (in eng).
101. Finn RD*, et al.* (2010) The Pfam protein families database. (Translated from eng) *Nucleic Acids Res* 38(Database issue):D211-222 (in eng).
102. Haft DH*, et al.* (2013) TIGRFAMs and Genome Properties in 2013. (Translated from eng) *Nucleic Acids Res* 41(Database issue):D387-395 (in eng).
103. Tatusova T, Ciufo S, Fedorov B, O'Neill K, & Tolstoy I (2015) RefSeq microbial genomes database: new representation and annotation strategy. (Translated from eng) *Nucleic Acids Res* 43(7):3872 (in eng).
104. Lu YK*, et al.* (2010) Metabolic flexibility revealed in the genome of the cyst-forming alpha-1 proteobacterium Rhodospirillum centenum. *BMC genomics* 11:325.
105. Duquesne KaS, J. (2012) Shotgun genome sequence of Phaeospirillum photometricum DSM 122. (CNRS, Marseille).
106. Munk AC*, et al.* (2011) Complete genome sequence of Rhodospirillum rubrum type strain (S1). *Standards in genomic sciences* 4(3):293-302.
107. Lonjers ZT*, et al.* (2012) Identification of a new gene required for the biosynthesis of rhodoquinone in Rhodospirillum rubrum. (Translated from eng) *J Bacteriol* 194(5):965-971 (in eng).
108. Wisniewski-Dye F*, et al.* (2011) Azospirillum genomes reveal transition of bacteria from aquatic to terrestrial environments. *PLoS genetics* 7(12):e1002430.
109. Kaneko T*, et al.* (2010) Complete genomic structure of the cultivated rice endophyte Azospirillum sp. B510. (Translated from eng) *DNA Res* 17(1):37-50 (in eng).
110. Wang X*, et al.* (2014) Complete Genome Sequence of Magnetospirillum gryphiswaldense MSR-1. (Translated from eng) *Genome Announc* 2(2) (in eng).
111. Matsunaga T*, et al.* (2005) Complete genome sequence of the facultative anaerobic magnetotactic bacterium Magnetospirillum sp. strain AMB-1. *DNA research : an international journal for rapid publication of reports on genes and genomes* 12(3):157-166.
112. Fournier PE*, et al.* (2009) Analysis of the Rickettsia africae genome reveals that virulence acquisition in Rickettsia species may be explained by genome reduction.

(Translated from eng) *BMC Genomics* 10:166 (in eng).
113. Madan A, Fahey,J., Helton,E., Ketteman,M., Madan,A., Rodrigues,S., Sanchez,A., Whiting,M., Dasch,G. and Eremeeva,M. (2007) Complete Genome Sequence of Rickettsia akari. (University of Iowa, Iowa City).
114. Johnson SL, Munk,A.C., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia australis str. Cutlack. (Los Alamos National Laboratory, Los Alamos).
115. Madan A, Lee,H., Madan,A., Yoon,J.-G., Ryu,G.-Y., Dasch,G. and Ereemeva,M. (2007) Complete genome sequencing of Rickettsia bellii OSU 85-389. (University of Iowa, Iowa City).
116. Ogata H*, et al.* (2006) Genome sequence of Rickettsia bellii illuminates the role of amoebae in gene exchanges between intracellular pathogens. (Translated from eng) *PLoS Genet* 2(5):e76 (in eng).
117. Johnson SL, Munk,A.C., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia canadensis str. CA410. (Los Alamos National Laboratory, Los Alamos).
118. Madan A, Fahey,J., Helton,E., Ketteman,M., Madan,A., Rodrigues,S., Sanchez,A., Whiting,M., Dasch,G. and Eremeeva,M. (2007) Complete Genome Sequence of Rickettsia canadensis str. McKiel. (University of Iowa, Iowa City).
119. Ogata H*, et al.* (2001) Mechanisms of evolution in Rickettsia conorii and R. prowazekii. (Translated from eng) *Science* 293(5537):2093-2098 (in eng).
120. Ogata H*, et al.* (2005) The genome sequence of Rickettsia felis identifies the first putative conjugative plasmid in an obligate intracellular parasite. (Translated from eng) *PLoS Biol* 3(8):e248 (in eng).
121. Duan C*, et al.* (2011) Complete genome sequence of Rickettsia heilongjiangensis, an emerging tick-transmitted human pathogen. (Translated from eng) *J Bacteriol* 193(19):5564-5565 (in eng).
122. Matsutani M*, et al.* (2013) Complete genomic DNA sequence of the East Asian spotted fever disease agent Rickettsia japonica. (Translated from eng) *PLoS One* 8(9):e71861 (in eng).
123. Johnson SL, Munk,A.C., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia massiliae str. AZT80. (Los Alamos National Laboratory, Los Alamos).
124. Blanc G*, et al.* (2007) Lateral gene transfer between obligate intracellular bacteria: evidence from the Rickettsia massiliae genome. (Translated from eng) *Genome Res* 17(11):1657-1664 (in eng).
125. Johnson SL, Munk,A.C., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia montanensis str. OSU 85-930. (Los Alamos National Laboratory, Los Alamos).
126. Johnson SL, Munk,A.C., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia parkeri str. Portsmouth. (Los Alamos National Laboratory, Los Alamos).
127. Felsheim RF, Kurtti TJ, & Munderloh UG (2009) Genome sequence of the endosymbiont Rickettsia peacockii and comparison with virulent Rickettsia rickettsii: identification of virulence factors. (Translated from eng) *PLoS One* 4(12):e8361 (in eng).

128. Johnson SL, Munk,A.C., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia philipii str. 364D. (Los Alamos National Laboratory, Los Alamos).
129. Bishop-Lilly KA, Ge,H., Butani,A., Osborne,B., Verratti,K., Mokashi,V., Nagarajan,N., Pop,M., Read,T.D. and Richards,A.L. (2013) Genome Sequencing of Four Strains of Rickettsia prowazekii, the Causative Agent of Epidemic Typhus, Including One Flying Squirrel Isolate. *Genome Announc* 1(3).
130. Johnson SL, Sims,D., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia prowazekii str. BuV67-CWPP. (Los Alamos National Laboratory, Los Alamos).
131. Johnson SL, Munk,A.C., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia prowazekii str. Chernikova. (Los Alamos National Institue Laboratory, Los Alamos).
132. Johnson SL, Sims,D., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia prowazekii str. Dachau. (Los Alamos National Laboratory, Los Alamos).
133. Johnson SL, Sims,D., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia prowazekii str. GvV257. (Los Alamos National Laboratory, Los Alamos).
134. Johnson SL, Sims,D., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia prowazekii str. Katsinyian. (Los Alamos National Laboratory, Los Alamos).
135. Andersson SG*, et al.* (1998) The genome sequence of Rickettsia prowazekii and the origin of mitochondria. *Nature* 396(6707):133-140.
136. Bechah Y*, et al.* (2010) Genomic, proteomic, and transcriptomic analysis of virulent and avirulent Rickettsia prowazekii reveals its adaptive mutation capabilities. (Translated from eng) *Genome Res* 20(5):655-663 (in eng).
137. Johnson SL, Sims,D., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia prowazekii str. RpGvF24. (Los Alamos National Laboratory, Los Alamos).
138. Johnson SL, Munk,A.C., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia rhipicephali str. 3-7-female6-CWPP. in *Los Alamos* (Los Alamos National Laboratory).
139. Johnson SL, Davenport,K.W., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia rickettsii str. Arizona. (Los Alamos National Laboratory, Los Alamos).
140. Johnson SL, Davenport,K.W., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia rickettsii str. Brazil. (Los Alamos National Laboratory, Los Alamos).
141. Johnson SL, Munk,A.C., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia rickettsii str. Colombia. (Los Alamos National Laboratory, Los Alamos).
142. Johnson SL, Munk,A.C., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia rickettsii str. Hauke. (Los Alamos National Laboratory, Los Alamos).
143. Johnson SL, Sims,D., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome

Sequence of Rickettsia rickettsii str. Hino. (Los Alamos National Laboratory, Los Alamos).
144. Johnson SL, Davenport,K.W., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia rickettsii str. Hlp#2. (Los Alamos National Laboratory, Los Alamos).
145. Ellison DW*, et al.* (2008) Genomic comparison of virulent Rickettsia rickettsii Sheila Smith and avirulent Rickettsia rickettsii Iowa. *Infection and immunity* 76(2):542-550.
146. Madan A, Fahey,J., Helton,E., Ketteman,M., Madan,A., Rodrigues,S., Sanchez,A., Dasch,G. and Eremeeva,M. (2007) Complete Genome Sequence of Rickettsia rickettsii str. 'Sheila Smith'. (University of Iowa, Iowa City).
147. Fournier PE, El Karkouri K, Robert C, Medigue C, & Raoult D (2012) Complete genome sequence of Rickettsia slovaca, the agent of tick-borne lymphadenitis. (Translated from eng) *J Bacteriol* 194(6):1612 (in eng).
148. Johnson SL, Munk,A.C., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia slovaca str. D-CWPP. (Los Alamos National Laboratory, Los Alamos).
149. Johnson SL, Sims,D., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia typhi str. B9991CWPP. (Los Alamos National Laboratory, Los Alamos).
150. Johnson SL, Sims,D., Han,S., Bruce,D.C. and Dasch,G.A. (2012) Complete Genome Sequence of Rickettsia typhi str. TH1527. (Los Alamos National Laboratory, Los Alamos).
151. McLeod MP*, et al.* (2004) Complete genome sequence of Rickettsia typhi and comparison with sequences of other rickettsiae. (Translated from eng) *J Bacteriol* 186(17):5842-5855 (in eng).
152. Cho NH*, et al.* (2007) The Orientia tsutsugamushi genome reveals massive proliferation of conjugative type IV secretion system and host-cell interaction genes. (Translated from eng) *Proc Natl Acad Sci U S A* 104(19):7981-7986 (in eng).
153. Nakayama K*, et al.* (2008) The Whole-genome sequencing of the obligate intracellular bacterium Orientia tsutsugamushi revealed massive gene amplification during reductive genome evolution. (Translated from eng) *DNA Res* 15(4):185-199 (in eng).
154. Klasson L*, et al.* (2008) Genome evolution of Wolbachia strain wPip from the Culex pipiens group. (Translated from eng) *Mol Biol Evol* 25(9):1877-1887 (in eng).
155. Wu M*, et al.* (2004) Phylogenomics of the reproductive parasite Wolbachia pipientis wMel: a streamlined genome overrun by mobile genetic elements. (Translated from eng) *PLoS Biol* 2(3):E69 (in eng).
156. Ellegaard KM, Klasson L, Naslund K, Bourtzis K, & Andersson SG (2013) Comparative genomics of Wolbachia and the bacterial species concept. *PLoS genetics* 9(4):e1003381.
157. Darby AC*, et al.* (2012) Analysis of gene expression from the Wolbachia genome of a filarial nematode supports both metabolic and defensive roles within the symbiosis. (Translated from eng) *Genome Res* 22(12):2467-2477 (in eng).
158. Foster J*, et al.* (2005) The Wolbachia genome of Brugia malayi: endosymbiont evolution within a human pathogenic nematode. (Translated from eng) *PLoS Biol* 3(4):e121 (in eng).
159. Klasson L*, et al.* (2009) The mosaic genome structure of the Wolbachia wRi strain

infecting Drosophila simulans. (Translated from eng) *Proc Natl Acad Sci U S A* 106(14):5725-5730 (in eng).
160. Palenik B, Copeland,A., Lucas,S., Lapidus,A., Barry,K., Detter,J.C., Glavina,T., Hammon,N., Israni,S., Pitluck,S., Chain,P., Malfatti,S., Shin,M., Vergez,L., Schmutz,J., Larimer,F., Land,M., Mavrommatis,K. and Richardson,P. (2005) Complete sequence of Ehrlichia canis str. Jake. (US DOE Joint Genome Institute, Walnut Creek).
161. Dunning Hotopp JC*, et al.* (2006) Comparative genomics of emerging human ehrlichiosis agents. *PLoS genetics* 2(2):e21.
162. Thirumalapura NR, Qin X, Kuriakose JA, & Walker DH (2014) Complete Genome Sequence of Ehrlichia muris Strain AS145T, a Model Monocytotropic Ehrlichia Strain. (Translated from eng) *Genome Announc* 2(1) (in eng).
163. Frutos R*, et al.* (2006) Comparative genomic analysis of three strains of Ehrlichia ruminantium reveals an active process of genome size plasticity. (Translated from eng) *J Bacteriol* 188(7):2533-2542 (in eng).
164. Collins NE*, et al.* (2005) The genome of the heartwater agent Ehrlichia ruminantium contains multiple tandem repeats of actively variable copy number. (Translated from eng) *Proc Natl Acad Sci U S A* 102(3):838-843 (in eng).
165. Herndon DR, Palmer GH, Shkap V, Knowles DP, Jr., & Brayton KA (2010) Complete genome sequence of Anaplasma marginale subsp. centrale. (Translated from eng) *J Bacteriol* 192(1):379-380 (in eng).
166. Pierle SA*, et al.* (2014) Genetic Diversity of Tick-Borne Rickettsial Pathogens; Insights Gained from Distant Strains. (Translated from Eng) *Pathogens* 3(1):57-72 (in Eng).
167. Dark MJ*, et al.* (2009) Conservation in the face of diversity: multistrain analysis of an intracellular bacterium. *BMC genomics* 10:16.
168. Brayton KA*, et al.* (2005) Complete genome sequencing of Anaplasma marginale reveals that the surface is skewed to two superfamilies of outer membrane proteins. (Translated from eng) *Proc Natl Acad Sci U S A* 102(3):844-849 (in eng).
169. Al-Khedery BaB, A.F. (2013) Complete Genome Sequence of Anaplasma phagocytophilum str. Dog2. (University of Florida, Gainesville).
170. Al-Khedery BaB, A.F. (2013) Complete Genome Sequence of Anaplasma phagocytophilum str. HZ2. (University of Florida, Gainesville).
171. Barbet T (2013) Complete Genome Sequence of Anaplasma phagocytophilum str. JM. (University of Florida, Gainesville).
172. Karp PD, Paley S, & Romero P (2002) The Pathway Tools software. (Translated from eng) *Bioinformatics* 18 Suppl 1:S225-232 (in eng).
173. Karp PD*, et al.* (2010) Pathway Tools version 13.0: integrated software for pathway/genome informatics and systems biology. (Translated from eng) *Brief Bioinform* 11(1):40-79 (in eng).
174. Dale JM, Popescu L, & Karp PD (2010) Machine learning methods for metabolic pathway prediction. (Translated from eng) *BMC Bioinformatics* 11:15 (in eng).
175. Sassera D*, et al.* (2011) Phylogenomic evidence for the presence of a flagellum and cbb(3) oxidase in the free-living mitochondrial ancestor. *Molecular biology and evolution* 28(12):3285-3296.
176. Yang L*, et al.* (2014) Species identification through mitochondrial rRNA genetic

analysis. (Translated from eng) *Sci Rep* 4:4089 (in eng).
177. Lang BF, Lavrov, D., Beck, N. et Steinberg, S.V. (2012) Mitochondrial tRNA Structure, Identity, and Evolution of the Genetic Code. *Organelle Genetics - Evolution of Organelle Genomes and Gene Expression*, ed Bullerwell CE (Springer Berlin Heidelberg, Berlin), pp 431-474.
178. Quang le S, Gascuel O, & Lartillot N (2008) Empirical profile mixture models for phylogenetic reconstruction. (Translated from eng) *Bioinformatics* 24(20):2317-2323 (in eng).
179. Caspi R*, et al.* (2012) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. (Translated from eng) *Nucleic Acids Res* 40(Database issue):D742-753 (in eng).
180. Liu R & Ochman H (2007) Stepwise formation of the bacterial flagellar system. (Translated from eng) *Proc Natl Acad Sci U S A* 104(17):7116-7121 (in eng).
181. Sanglier S, Atmanene C, Chevreux G, & Dorsselaer AV (2008) Nondenaturing mass spectrometry to study noncovalent protein/protein and protein/ligand complexes: technical aspects and application to the determination of binding stoichiometries. (Translated from eng) *Methods Mol Biol* 484:217-243 (in eng).
182. Boscaro V*, et al.* (2013) Rediscovering the genus Lyticum, multiflagellated symbionts of the order Rickettsiales. (Translated from eng) *Sci Rep* 3:3305 (in eng).
183. Vannini C*, et al.* (2014) Flagellar movement in two bacteria of the family rickettsiaceae: a re-evaluation of motility in an evolutionary perspective. (Translated from eng) *PLoS One* 9(2):e87718 (in eng).
184. Min CK*, et al.* (2008) Genome-based construction of the metabolic pathways of Orientia tsutsugamushi and comparative analysis within the Rickettsiales order. *Comparative and functional genomics*:623145.
185. Mertens E (1993) ATP versus pyrophosphate: glycolysis revisited in parasitic protists. *Parasitology today* 9(4):122-126.
186. Zientz E, Dandekar T, & Gross R (2004) Metabolic interdependence of obligate intracellular bacteria and their insect hosts. *Microbiology and molecular biology reviews : MMBR* 68(4):745-770.
187. Stenmark P & Nordlund P (2003) A prokaryotic alternative oxidase present in the bacterium Novosphingobium aromaticivorans. (Translated from eng) *FEBS Lett* 552(2-3):189-192 (in eng).
188. McDonald AE & Vanlerberghe GC (2006) Origins, evolutionary history, and taxonomic distribution of alternative oxidase and plastoquinol terminal oxidase. (Translated from eng) *Comp Biochem Physiol Part D Genomics Proteomics* 1(3):357-364 (in eng).
189. Herrmann KM & Weaver LM (1999) The Shikimate Pathway. *Annual review of plant physiology and plant molecular biology* 50:473-503.
190. Beveridge TJ (1999) Structures of gram-negative cell walls and their derived membrane vesicles. *Journal of bacteriology* 181(16):4725-4733.