

Génotypage différentiel au niveau de l'ADN
mitochondrial entre reine et ouvrière issues de
3 colonies de *M. quadrifasciata*

Deschuyteneer Audry

Bachelier en biotechnique Bloc3

HEH Département des Sciences et technologies

HEPH – Condorcet

Année académique 2023-2024

Génotypage différentiel au niveau de l'ADN mitochondrial entre reine et ouvrière issues de 3 colonies de *M. quadrifasciata*

Deschuyteneer Audry

Enseignant suiveur ;David Coornaert

Natalia De Souza Araujo

Bachelier en biotechnique Bloc3

HEH Département des Sciences et technologies

HEPH – Condorcet

Année académique 2023-2024

Remerciements

Tout d'abord, j'aimerais remercier Natalia De Souza Araujo pour la confiance qu'elle m'a donnée pour mener à bien ces recherches. Ensuite, Monsieur Cornaert qui m'a encadré durant mon stage et m'a permis d'approfondir mes recherches. Enfin, j'aimerais remercier Alessia Bellanca pour le soutien qu'elle m'a fourni durant l'élaboration de mon travail.

Résumé

Ce travail de recherche a exploré la diversité génétique dans l'ADN mitochondrial des abeilles *Melipona quadrifasciata* afin de mieux comprendre les différences inter colonies et les mécanismes de différenciation des castes. L'analyse a révélé des variants génétiques spécifiques à la colonie 1, incluant des SNPs, des insertions et des délétions, présents à la fois chez les reines et les ouvrières. Aucun variant n'a été observé au sein des autres colonies, ce qui reflète la stabilité de l'ADN mitochondrial transmis par la mère à sa descendance. Bien que ces variants semblent avoir un impact limité sur la survie des abeilles, cette étude souligne la nécessité d'analyser l'ensemble du génome pour obtenir une vision plus complète des mécanismes génétiques en jeu. Les résultats obtenus fournissent une base pour des recherches futures visant à explorer la différenciation des castes et l'adaptation des colonies chez *Melipona quadrifasciata*.

Abstract

This research explored the genetic diversity in the mitochondrial DNA of *Melipona quadrifasciata* bees to better understand inter-colony differences and caste differentiation mechanisms. The analysis revealed colony-specific genetic variants in colony 1, including SNPs, insertions, and deletions, present in both queens and workers. No variants were observed within the other colonies, reflecting the stability of maternally inherited mitochondrial DNA. Although these variants appear to have a limited impact on bee survival, this study highlights the need to analyze the entire genome to obtain a more comprehensive view of the underlying genetic mechanisms. The results provide a foundation for future research aimed at exploring caste differentiation and colony adaptation in *Melipona quadrifasciata*.

Table des figures

Figure 1: <i>Melipona quadrifasciata</i>	2
Figure 2: Aire de répartition de <i>M. quadrifasciata</i> au brésil.....	2
Figure 3: Gardien de l'entrée du nid	3
Figure 4: Exemple de nymphe d'ouvrière et de reine	5
Figure 5: Dotmatcher basique de la référence contre O2N1.....	10
Figure 6: Dotplot du fichier biplot.fasta contre la référence	10
Figure 7: Dotplot de O1N1 contre biplot.fasta en ajustant les paramètres de seuil et de fenêtre	11
Figure 8: Exemple de fichier avec le début décallé par rapport à la référence	12
Figure 9: Capture d'écran du fichier de sortie de merger	13
Figure 10: dotplot du fichier O1N1 corrigé par le script put_start.sh.....	13
Figure 11: Exemple de fichier ayant une duplication	14
Figure 12: Dotplot du fichier corrigé par le script deduplicate.sh.....	14
Figure 13: Dotplot de fichier à inverser.....	15
Figure 14: Dotplot du fichier corrigé avec revseq	15
Figure 15: Comparaison des 2 génomes de référence	19
Figure 16: Dotplot des fichiers ayant les séquences très répétitives aux extrémités	20
Figure 17: Capture d'écran d'une insertion dans la colonie 1	21
Figure 18: Capture d'écran d'une délétion et d'un SNP dans la colonie1.....	21
Figure 19: Capture d'écran d'un biais de séquencage.....	21
Figure 20: Capture d'écran du nombre de variants du fichier snpEff_summary.html	23
Figure 21: Capture d'écran des codons modifiés par les variants contenu dans le fichier snpEff_summary.html	23
Figure 22: Démonstration de la réorganisation des fichiers par Clustal	24
Figure 23: Capture d'écran du fichier snpEff.config	31
Figure 24: Capture d'écran du script auto_dotmatcher.sh	31
Figure 25: Capture d'écran du script deduplicate.sh	32
Figure 26: Capture d'écran du script put_start.sh.....	32
Figure 27: Capture d'écran du script auto_snpEff.sh	33
Figure 28: Bases modifiées par les variants dans toutes les séquences	34
Figure 29: Codons modifiés par les variants.....	34
Figure 30: Acides aminés modifié par les variants	35
Figure 31: Bases modifiées par les variants dans toutes les séquences à partir du fichier vcf modifié	36
Figure 32: Captue d'écran de la matrice EDNAFULL utilisée par dotmatcher	36
Figure 33: Dotplot de biplot.fasta contre O1N1	37
Figure 34: Dotplot de biplot.fasta contre O2N1	37
Figure 35: Dotplot de biplot.fasta contre O3N1	38
Figure 36: Dotplot de biplot.fasta contre O4N1	38
Figure 37: Dotplot de biplot.fasta contre Q1N1	39
Figure 38: Dotplot de biplot.fasta contre Q2N1	39
Figure 39: Dotplot de biplot.fasta contre Q3N1	40
Figure 40: Dotplot de biplot.fasta contre O1N2	40
Figure 41: Dotplot de biplot.fasta contre O2N2	41
Figure 42: Dotplot de biplot.fasta contre O3N2	41
Figure 43: Dotplot de biplot.fasta contre O4N2	42
Figure 44: Dotplot de biplot.fasta contre Q1N2	42

Figure 45: Dotplot de biplot.fasta contre Q2N2	43
Figure 46: Dotplot de biplot.fasta contre Q3N2	43
Figure 47: Dotplot de biplot.fasta contre O1N3	44
Figure 48: Dotplot de biplot.fasta contre O2N3	44
Figure 49: Dotplot de biplot.fasta contre O3N3	45
Figure 50: Dotplot de biplot.fasta contre O4N3	45
Figure 51: Dotplot de biplot.fasta contre Q1N3	46
Figure 52: Dotplot de biplot.fasta contre Q2N3	46
Figure 53: Dotplot de biplot.fasta contre Q3N3	47
Figure 54: Délétion et SNP dans la colonie 1	48
Figure 55: Délétion de TA dans la colonie 1	48
Figure 56: Insertion de TTTA dans la colonie 1	48
Figure 57: Délétion de ATAT dans la colonie 1	48
Figure 58: Délétion de GAA dans la colonie 1	49
Figure 59: SNP de A vers T dans la colonie 1	49
Figure 60: SNP de A vers G dans la colonie 1	49
Figure 61: SNP de C vers T dans la colonie 1	49
Figure 62: Insertion et SNP dans la colonie 1	50
Figure 63: SNP de A vers G dans la colonie 1	50

Table des tableaux

Tableau 1: Récapitulatif des modifications à apporter aux fichiers	12
Tableau 2: Récapitulatif des différences entre les fichiers sam et sorted.bam	16
Tableau 3: Récapitulatif des variants identifiés via ClustalX	25
Tableau 4: Récapitulatif des gènes modifiés par les variants	26

Table des matières

Remerciements.....	4
Résumé	5
Abstract	5
Table des figures.....	6
Table des tableaux.....	7
Introduction	1
Objectifs	4
Matériel et Méthode	5
Introduction.....	5
Matériel.....	5
Echantillons biologiques utilisés.....	5
Données reçues	6
Prétraitement des données.....	6
Transformation des fichiers.....	7
Alignment des séquences.....	7
Mise en évidence des variants.....	8
Méthode.....	9
Extraction et séquençage	9
Première analyse.....	9
Seconde analyse.....	19
Résultats et interprétation	23
Première analyse.....	23
Seconde analyse.....	24
Discussion	27
Conclusion	28
Bibliographie.....	29
Annexes	31

Introduction

Le travail réalisé se porte sur l'analyse du génome mitochondrial des abeilles *Melipona quadrifasciata* afin de détecter de potentiels variants. L'ADN mitochondrial est couramment utilisé comme marqueur génétique chez les vertébrés et invertébrés car il possède des caractéristiques intéressantes : un haut taux de mutation, une transmission maternelle, une absence de recombinaison et une petite taille moléculaire (G.S. Barni, 2007).

La mitochondrie est l'organite cellulaire produisant l'énergie pour toute la cellule. Une des caractéristiques des mitochondries est sa contenance en ADN. Le génome mitochondrial est confiné à l'intérieur des mitochondries et est distinct de l'ADN nucléaire. Comme cet ADN est indépendant de l'ADN nucléaire, il n'est transmis que par la mère à sa descendance. Une particularité du génome mitochondrial des arthropodes est la présence d'une région non codante très répétitive contenant beaucoup d'adénine et de thymine (Aragão, 2019) (Slzbg, 2024).

Appartenant au phylum des arthropodes, dans la classe des insectes, dans l'ordre des hyménoptères et dans la famille des apidae, les abeilles du genre *Melipona* sont un genre d'abeilles sans dard répandues dans les zones chaudes tropicales et subtropicales de presque tous les continents (absent en Europe) car celles-ci ne résistent pas à un climat tempéré. Il existe environ 50 espèces de *Melipona*. Au Brésil et au Mexique, ces abeilles sont utilisées dans la production de miel (Jean.claude, 2013).

Les melipones ont une importance considérable dans la reproduction des plantes des zones tropicales. Par exemple, la vanille dépend d'une seule espèce de *Trigone*. Sur le continent Américain, la meliponiculture est présente depuis les civilisations précolombiennes mais est de plus en plus remplacée par la production de miel de l'abeille *Apis mellifera* (production de miel plus importante). Cependant, la meliponiculture est davantage mise en avant par les gouvernements car le miel produit par les melipones est plus qualitatif que le miel produit par les *Apis mellifera*. Etant donné l'absence de dard chez ces abeilles, la meliponiculture est plus aisée et plus abordable (França, 2011).

L'abeille *Melipona quadrifasciata* est une espèce sociale, originaire des états côtiers du sud-est du Brésil. Ces abeilles vivent dans des ruches de boue, construites dans le creux des arbres, créant des passages étroits afin de ne laisser passer qu'une abeille à la fois. Les *Melipona quadrifasciata* sont souvent utilisées comme pollinisatrices dans les serres car celles-ci sont plus efficaces que les autres abeilles mellifères. En effet, les melipones permettent un rendement de fruits plus lourds, plus gros et contenant plus de graines (FAYET, 2014).



Figure 2: Aire de répartition de *M. quadrifasciata* au brésil



Figure 1: *Melipona quadrifasciata*

La morphologie de ces abeilles se distingue par un corps arrondi de couleur noir foncé, des antennes légèrement incurvées et des ailes translucides. Mesurant entre 10 et 11 mm, elles sont plus lourdes que les abeilles communes. Elles peuvent être identifiées par les rayures jaune vif sur les tergites¹ abdominaux (du troisième au sixième). Les reines et les ouvrières sont différentes sur plusieurs aspects. L'abdomen des reines se dilate avec le développement ovarien, rendant les reines plus âgées, plus grandes que les ouvrières (un trait typique de la plupart des abeilles sociales). Les reines présentent également de légères variations de coloration, avec des yeux et des poils bruns, par rapport aux yeux et aux poils noirs des ouvrières.

Les melipones sont une espèce d'abeilles hautement eusociales, caractérisée par des colonies dynamiques généralement dirigées par une reine issue d'un seul accouplement. Une colonie typique compte en moyenne 300 à 400 ouvrières adultes et une reine. Lorsque le nombre d'ouvrières dépasse 500 ou 600 dans la colonie mère, certaines ouvrières commencent à construire un nouveau nid dans une cavité d'arbre adaptée, y stockant du miel et du pollen. Une fois le nouveau nid prêt, une « abeille princesse » (gyne² accouplée) rejoint les ouvrières. Si elle est acceptée, elle commence à pondre des œufs et devient la nouvelle reine. Comme chez les autres melipones, son abdomen se dilate avec le temps jusqu'à trois fois ou plus (un phénomène appelé physogastrisme³), la rendant incapable de voler et ne quittant plus jamais le nid (Wdsieling, 2024).

Lorsque la nouvelle reine est définitivement installée dans son futur nid, les ouvrières agrandissent celui-ci en y ajoutant des cellules. Elles sont construites et approvisionnées par les ouvrières. Plusieurs ouvrières peuvent construire une même cellule. Lorsqu'une cellule est terminée, la reine y pénètre et inspecte son contenu. Si la cellule est acceptée par la reine, alors celle-ci y reste et une ouvrière lui apporte de la nourriture. La reine nourrie peut alors pondre un œuf dans la cellule, qui est ensuite fermée jusqu'à l'écllosion.

La différenciation des ouvrières et des reines se fait de manière génétique et environnementale. Les cellules qui produisent les ouvrières et les reines sont indiscernables et mélangées. Les

¹ Partie dorsale de chaque anneau des insectes.

² Femelle fécondée fondatrice

³ Relatif à une physogastrie, dilatation de l'abdomen.

cellules qui reçoivent une petite quantité de nourriture donnent des ouvrières tandis que les cellules bien nourries donnent des gynes et des ouvrières. Mais ce n'est pas la seule raison de différenciation. En effet, pour qu'une gyne se développe, celle-ci doit être hétérozygote. Si ce n'est pas le cas, l'individu sera une ouvrière.



Figure 3: Gardien de l'entrée du nid

Objectifs

Ce travail s'inscrit dans une étude plus large, visant à démontrer les mécanismes génétiques sous-jacent à la différenciation des castes chez les abeilles *Melipona quadrifasciata*, une espèce d'abeilles sans aiguillon particulièrement intéressante en raison de sa structure sociale complexe. Les castes au sein d'une colonie, telles que les reines et les ouvrières, sont différencierées par des facteurs génétiques et environnementaux mais les mutations spécifiques responsables de ces différences demeurent mal comprises. Ce projet vise à éclaircir ces mécanismes en se concentrant sur l'ADN mitochondrial, une composante génomique souvent négligée mais cruciale pour la compréhension des adaptations énergétiques et fonctionnelles de ces insectes.

Le travail réalisé ne porte que sur une partie de l'ADN total séquencé, à savoir l'ADN mitochondrial, qui a été séquencé au préalable par la méthode Oxford Nanopore. Cette méthode de séquençage de nouvelle génération est connue pour sa capacité à générer de longues lectures de séquences, facilitant ainsi l'étude des variants structurels et des mutations difficiles à détecter par d'autres technologies.

Le premier objectif de ce travail est de détecter les différences génétiques entre trois colonies distinctes de *Melipona quadrifasciata*, en utilisant les génomes mitochondriaux de 21 abeilles, incluant 3 reines et 4 ouvrières par colonie. Ces comparaisons inter colonies permettront de mettre en évidence des variants spécifiques, tels que des SNPs (Single Nucleotide Polymorphisms) ainsi que des insertions ou délétions (indels), pouvant être associés à l'adaptation locale ou à des traits spécifiques des colonies.

Le second objectif consiste à explorer les variations génétiques entre les reines et les ouvrières au sein d'une même colonie. Ces variants génétiques, en particulier ceux localisés dans des gènes codants ou des régions régulatrices, pourraient influencer des processus biologiques essentiels, comme la production d'énergie cellulaire, la régulation du cycle de vie et la résistance au stress.

Un autre aspect crucial de l'analyse est d'évaluer l'impact fonctionnel de ces variants. L'identification des mutations non synonymes, c'est-à-dire celles qui entraînent un changement dans la séquence d'acides aminés des protéines mitochondrielles, est primordiale pour comprendre comment ces modifications peuvent affecter la structure, la fonction des protéines et la physiologie des abeilles. Une attention particulière sera portée aux gènes impliqués dans la chaîne respiratoire mitochondriale, étant donné leur rôle central dans la production d'ATP et la régulation métabolique, qui sont essentiels à la survie et à la spécialisation des castes au sein des colonies.

Ce projet a pour but de contribuer à la compréhension des bases génétiques de la différenciation des castes chez les *Melipona quadrifasciata*, mais aussi d'apporter des éléments de réflexion sur l'évolution des systèmes sociaux complexes.

Matériel et Méthode

Introduction

Cette section détaille les approches bioinformatiques et les logiciels utilisés pour l'analyse des données de séquençage de l'ADN mitochondrial des abeilles *Melipona quadrifasciata*. La collecte des échantillons biologiques et le séquençage de l'ADN total ont été réalisés par Natalia De souza Araujo au laboratoire de l'ULB.

L'isolation du génome mitochondrial a également été réalisé par Natalia. Dès lors, le travail présenté repose sur le traitement des données, l'identification des single nucleotid polymorphism (SNPs) et l'analyse des variations génétiques afin de garantir une analyse rigoureuse et reproductible des données.

Matériel

Echantillons biologiques utilisés

Les analyses ont été réalisées sur 21 abeilles de l'espèce *Melipona quadrifasciata* provenant de 3 colonies. Les abeilles proviennent du Brésil à Ribeirão Preto, São Paulo et ont été récoltées en mars 2023. 3 reines et 4 ouvrières ont été prélevées par colonie, au stade de nymphe. C'est le premier stade où on peut distinguer les reines des ouvrières par rapport à la taille de leurs yeux et de leur tête.



Figure 4: Exemple de nymphes de reine et d'ouvrières

Ci-dessus, une reine et une ouvrière au stade de nymphe. L'ouvrière est plus petite que la reine.

Données reçues

Les données obtenues comprennent 21 dossiers d'ADN mitochondrial. Chaque dossier comporte :

- L'ADN mitochondrial au format fasta
- Un fichier d'informations, contenant la taille du génome et le pourcentage de GC
- Les gènes présents dans le génome sous forme de nucléotides
- Les gènes présents dans le génome sous forme d'acides aminés
- Un fichier GB contenant les informations des gènes et la séquence
- Un fichier d'annotations au format gff
- Un fichier TBL des gènes présents

Chaque fichier contenant l'ADN au format fasta a une taille d'environ 22Ko.

Prétraitement des données

1. Dotmatcher (Longden, 1999)
 - EMBOSS version : 6.6.0.0
 - Description : Dotmatcher est un outil utilisé pour créer des graphiques en points (dot plots) afin de visualiser les similarités entre 2 séquences d'ADN ou de protéines. Il fait partie de la suite EMBOSS (European Molecular Biology Open Software Suite) et est utilisé pour identifier les régions de similarité et les motifs répétitifs entre 2 séquences. Dotmatcher génère des matrices de points où les correspondances entre les séquences sont représentées graphiquement, ce qui permet une analyse visuelle rapide des alignements et des structures répétitives.
2. Merger (Williams, emboss merger, 1999)
 - EMBOSS version : 6.6.0.0
 - Description : Merger est un outil utilisé pour comparer et aligner 2 séquences d'ADN ou de protéines. Il permet de réaliser des alignements locaux ou globaux et est particulièrement utile pour identifier les différences précises entre les séquences, telles que les insertions, délétions ou substitutions. En utilisant Merger, il est possible de déterminer l'emplacement exact des anomalies ou des discordances dans les séquences d'ADN, ce qui permet ensuite de corriger ces erreurs pour assurer une meilleure correspondance avec une séquence de référence.

3. Revseq (Williams, emboss revseq, 1999)

- EMBOSS version : 6.6.0.0
- Description : Revseq est un outil de la suite EMBOSS utilisé pour produire la séquence réverse complémentaire d'une séquence d'ADN donnée. En spécifiant une séquence d'ADN en entrée, Revseq inverse l'ordre des nucléotides et remplace chaque nucléotide par son complément : A par T, T par A, C par G, et G par C. Cet outil est utile dans les analyses où il est nécessaire de travailler avec la séquence réverse complémentaire, comme pour aligner correctement des séquences complémentaires inversées par rapport à une séquence de référence.

Transformation des fichiers

1. Samtools (whitwham, 2021)

- Version : 1.18-25-g1f96fa8
- Description : Samtools est un ensemble de programmes utilisés pour interagir avec les fichiers SAM (Sequence Alignment/Map) et BAM (Binary Alignment/Map), qui sont des formats standardisés pour représenter les alignements de séquences de lecture sur une séquence de référence. Samtools permet de trier, d'indexer, de convertir et de filtrer les fichiers SAM/BAM, ainsi que d'effectuer des opérations de base telles que l'extraction de sous-ensembles de données, la fusion de fichiers et la génération de statistiques de couverture. Il est un outil essentiel pour le traitement post-alignement des données de séquençage.

Alignement des séquences

1. BWA (Burrows-Wheeler Aligner) (R., 2013)

- Version : 0.7.18-r1243-dirty
- Description : BWA est un outil bioinformatique utilisé pour aligner des séquences de lecture courtes sur une séquence de référence. Il utilise l'algorithme de Burrows-Wheeler pour permettre une recherche rapide et efficace des correspondances dans la séquence de référence. BWA est particulièrement adapté aux lectures de séquençage de nouvelle génération (NGS) et prend en charge différents modes d'alignement, notamment BWA-MEM, qui est recommandé pour les lectures longues et courtes. Le programme est connu pour sa rapidité, son efficacité et sa capacité à gérer les insertions et les délétions.

2. ClustalX (John Wiley & Sons, 2003)

- Version : 2.1
- Description : ClustalX est un logiciel d'alignement multiple de séquences doté d'une interface graphique, basé sur l'algorithme de ClustalW2. Il est utilisé pour aligner des séquences nucléotidiques ou protéiques de manière progressive, en construisant d'abord une matrice de distances, puis un guide tree qui sert de base à l'alignement multiple. Les alignements produits peuvent être visualisés directement dans le programme, avec une mise en couleur des résidus selon leur conservation, facilitant ainsi l'analyse des résultats. Les fichiers d'alignement générés sont exportables aux formats standards (FASTA, Clustal) pour une utilisation ultérieure dans d'autres analyses bioinformatiques.

Mise en évidence des variants

1. Bcftools (Li, 2024)

- Version : 1.18-31-g40f7e2
- Description : Bcftools est un ensemble de programmes utilisé pour manipuler les fichiers VCF (Variant Call Format) et BCF (Binary Call Format), qui sont des formats standardisés pour stocker les données de variations génétiques. Bcftools permet de convertir, trier, filtrer, annoter et indexer les fichiers VCF/BCF, ainsi que d'effectuer des opérations comme l'appel de variantes et la comparaison entre ensembles de variantes. Il est couramment utilisé en conjonction avec Samtools pour un traitement complet des données de séquençage.

2. snpEff (Cingolani, 2024)

- Version : 5.2c
- Description : snpEff est un logiciel de prédiction et d'annotation des effets des variantes génétiques. Il permet d'annoter les SNPs et autres variantes en fonction de leurs impacts sur les gènes et les protéines, en utilisant une base de données de génomes et d'annotations génomiques. snpEff classe les effets des variantes en fonction de leur impact génétique probable, allant des mutations silencieuses aux mutations non-sens, et fournit des informations détaillées sur les conséquences biologiques des variations détectées.

Méthode

Extraction et séquençage

Natalia et son équipe ont échantillonné des nymphes d'ouvrières et de reines de 2 espèces (*Melipona quadrifasciata* et *Melipona scutellaris*) aux yeux allant du rose au brun, directement à partir des colonies en ouvrant les cellules reproductrices et en récupérant les échantillons au stade de nymphe. Il s'agit de l'un des premiers stades de développement au cours duquel les reines et les ouvrières peuvent être clairement différenciées en fonction des différences morphologiques, notamment la taille et la forme des yeux et de la tête.

L'ADN total a été extrait du corps entier de la nymphe à l'aide du kit universel Qiagen AllPrep DNA/RNA/miRNA. Le séquençage du génome entier des reines et des ouvrières a été réalisé à l'aide de la technologie Oxford Nanopore (mhc1 et P2Solo), avec des cellules en flux version 10.4.1. Pour les échantillons séquencés à l'aide du mhc1, nous avons utilisé le kit de préparation de banque LSK114 et pour le séquençage P2Solo, le kit de bibliothèque SQK-NBD114-96. Le séquençage de basecalling a été effectué à l'aide de Dorado v0.5.3 avec le mode d'appel duplex supérieur (modes de basecalling dna_r10.4.1_e8.2_400bps_sup@v4.1.0 ou [dna_r10.4.1_e8.2_400bps_sup@v4.3.0](#)).

Première analyse

La première analyse effectuée est basée sur le pipeline trouvé sur le site Bioinfo-fr.net et a été adapté pour être utilisé avec les données reçues (ClemBuntu, 2016).

Correction des fichiers

Une brève analyse préliminaire des données reçues a permis de mettre en évidence certains problèmes. Le premier problème rencontré est la présence de noms de fichiers incorrects. Cela est dû au fait que le programme utilisé pour assembler les génomes, donne des noms automatiques aux fichiers et aux séquences. Une correction a donc été apportée manuellement pour faciliter la compréhension des résultats.

Le second problème est un fichier manquant. Le fichier contenant l'ADN de l'ouvrière 2 de la colonie 1 n'est pas présent dans le dossier contenant toutes les séquences fasta. Ce fichier a par la suite été retrouvé dans un autre dossier.

Pour garantir la qualité des séquences, un examen a été effectué à l'aide de **Dotmatcher**. Afin de réaliser cet examen, une référence a été choisie parmi les abeilles du groupe. Suivant les conseils de Natalia, c'est l'ouvrière numéro 2 de la colonie 2 qui a été sélectionnée.

Dotmatcher produit des graphiques constitués de barres, qui décrivent l'alignement. Le programme nécessite la taille de la fenêtre utilisée en entrée ainsi que le score seuil à atteindre. Ce programme fonctionne selon un modèle de fenêtre glissante. Cela signifie qu'une fenêtre d'une taille prédéfinie de nucléotides de la première séquence est alignée à la seconde séquence. Ensuite, un score est attribué à l'alignement suivant une matrice (présente dans EMBOSS). Si le score atteint le seuil prédéfini, alors le programme trace une ligne à la position de l'alignement. Sinon, rien n'est tracé.

Il est donc essentiel de trouver les bonnes valeurs de fenêtre et de seuil pour l'alignement. L'analyse préliminaire de la matrice de score est obligatoire pour savoir quel seuil attribuer. En

effet, si une fenêtre est définie avec 10 nucléotides et que la matrice de score définit un match comme +5 points, alors, le score maximal ne peut être que de 50. Avec ces paramètres, Dotmatcher recherche un alignement parfait. Ce n'est pas le but de l'analyse étant donné qu'on s'attend à avoir des variants. La fenêtre utilisée a donc été fixée à 20 et le score seuil a été fixé à 100. Cela donne la commande suivante :

```
dotmatcher O1N1.fasta O2N2.fasta
```

Une fois ce premier alignement effectué, la fenêtre glisse de 1 nucléotide dans une des séquence et un autre alignement est effectué. De cette manière, tous les alignements possibles sont réalisés sur une longueur égale à la taille de la fenêtre.

Les premières analyses, réalisées sur les fichiers, ont aligné la séquence de l'abeille à analyser, contre la référence. Un dotplot est donc produit :

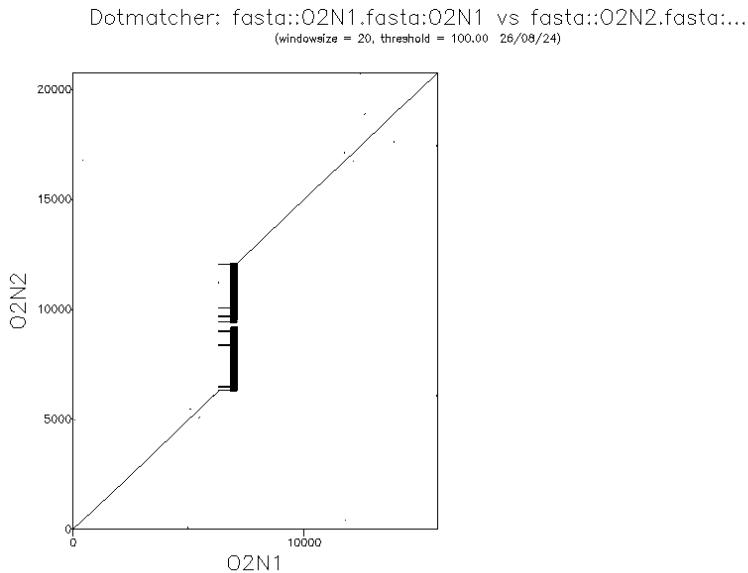


Figure 5: Dotmatcher basique de la référence contre O2N1

La limite de cette approche est que les séquences s'alignent contre la séquence réverse complémentaire à la référence ne peuvent pas être identifiées. Pour contrer ce problème, un fichier de référence particulier a été créé (biplot.fasta). Celui-ci contient la séquence de référence ainsi que sa séquence réverse complémentaire, mises bout à bout. De cette manière, l'alignement réalisé par Dotmatcher permettra de voir si la séquence s'aligne avec la référence ou avec sa séquence reverse complémentaire. Grâce à cela, le nombre de graphiques ainsi que le temps nécessaire à les examiner est divisé par deux.

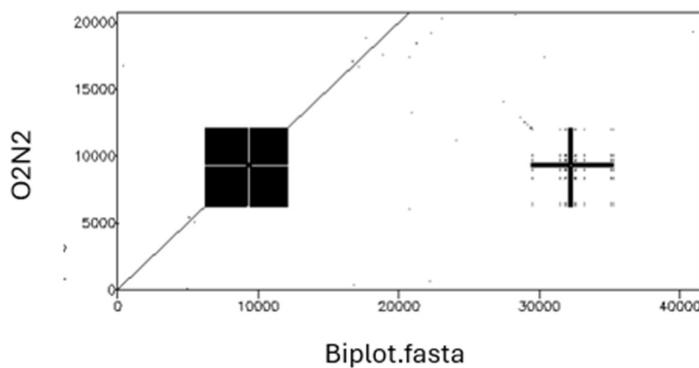


Figure 6: Dotplot du fichier biplot.fasta contre la référence

Ces étapes permettent de rendre les séquences les plus similaires possible en éliminant les duplications, les inversions et les séquences qui ont un début différent par rapport à la référence. Grâce à cela, le programme d'alignement BWA sera plus rapide et plus efficace.

Réorganisation des fichiers

Une des particularités des mitochondries des arthropodes est qu'il contient une région très répétitive d'adénine et de thymine. Le problème dans ce genre de région est que le séquençage est peu qualitatif. Cela entraîne des problèmes dans les séquences. Par conséquent, certains fichiers ont une région répétitive beaucoup plus petite que les autres.

Dans la référence choisie, cette région est présente de la position 6289 à 12111. Cette région très répétitive fait qu'un « bloc » apparaît au milieu du graphe. Etant donné la répétitivité de cette région, une « croix » est obtenue dans la séquence réverse complémentaire. En effet, la séquence contient énormément de AT successifs, il est donc normal de retrouver cette succession dans la séquence réverse complémentaire. Pour supprimer cette « croix » présente dans la séquence reverse, une adaptation de la fenêtre choisie et du score seuil doit être effectuée. La fenêtre a été fixée à 200 et le seuil à 1000.

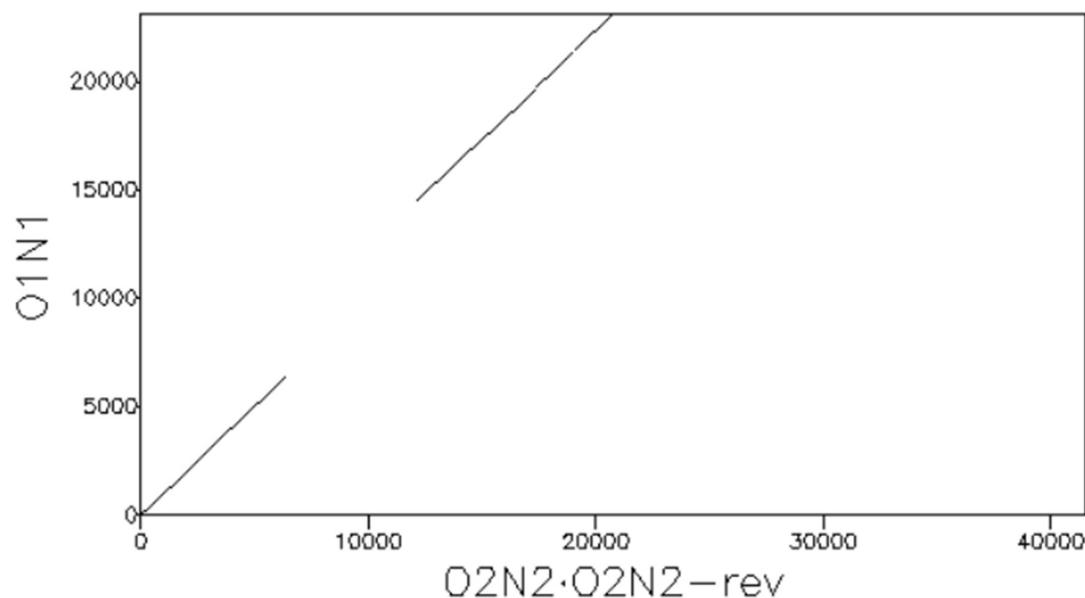


Figure 7: Dotplot de O1N1 contre biplot.fasta en ajustant les paramètres de seuil et de fenêtre

Une seconde constatation est que mis à part la région répétitive au centre (bloc noir), l'alignement est une ligne partant du zéro et allant jusqu'à la fin de la séquence. C'est ce type de morphologie qui est recherchée dans tous les graphes.

Une fois les paramètres optimaux trouvés, un script bash permet de lancer des Dotmatchers consécutifs pour qu'un graphe soit généré par fichier en comparant chaque séquence à biplot.fasta (voir annexe point 8). Cela permet de trouver les séquences problématiques. Trois types de problèmes ont été mis en évidence dans les fichiers :

- Morceaux de séquence dupliqués.
- L'ADN circulaire fait que le début de la séquence de référence n'est pas le même que celui de la séquence du fichier.
- Certaines séquences sont les réverses compléments de la référence.

Voici un tableau récapitulatif des séquences qui ont dû être corrigées :

Tableau 1: Récapitulatif des modifications à apporter aux fichiers

Dédupliquer	Inverser	Décaler
O4N1, Q2N1, O1N2	Q1N3	O1N1, Q2N2, Q3N2, Q1N3

Le problème le plus fréquent est un fichier ayant un début de séquence différent de celui de la référence (décalé). En effet, l'ADN mitochondrial est circulaire. Ce qui signifie que le début de la séquence de référence n'est pas forcément le même que la séquence analysée.

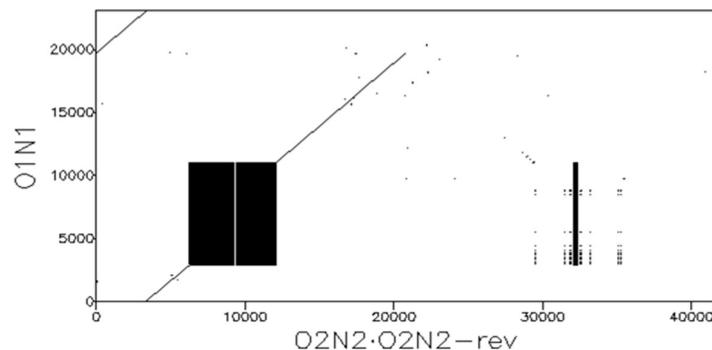


Figure 8: Exemple de fichier avec le début décalé par rapport à la référence

Afin de trouver la position exacte à laquelle la séquence doit-être décalée, le programme **merger** de la suite EMBOSS a été utilisé. En précisant quel morceau de la référence doit être aligné à la séquence analysée, une position précise de coupure peut être déterminée.

O2N2	3351	aatctttcaaaaataataaaaaattataaataataatctatttattgaaat 	3400
O1N1	1	-----ataaaaattataaataataatctatttattgaaat	35
O2N2	3401	tattgatcctaaagatgaaatataatttcaacaataatatgaatctggat 	3450
O1N1	36	tattgatcctaaagatgaaatataatttcaacaataatatgaatctggat	85
O2N2	3451	aatctgaatatcgctgttgttattcctattaatcctaaaaatgttgtgga 	3500
O1N1	86	aatctgaatatcgctgttgttattcctattaatcctaaaaatgttgtgga	135
O2N2	3501	aaaaatgttaattcacccaataaatataaaaaaaaaattgaaattttaa 	3550
O1N1	136	aaaaatgttaattcacccaataaatataaaaaaaaaattgaaattttaa	185

Figure 9: Capture d'écran du fichier de sortie de merger

Une première approche consistait à corriger manuellement les fichiers. Comme cette manipulation prenait beaucoup de temps, le script put_start.sh a été créé (voir annexe figure 23). En spécifiant le nom du fichier d'entrée, la position de coupure et le nom du fichier de sortie, le script corrige les fichiers automatiquement.

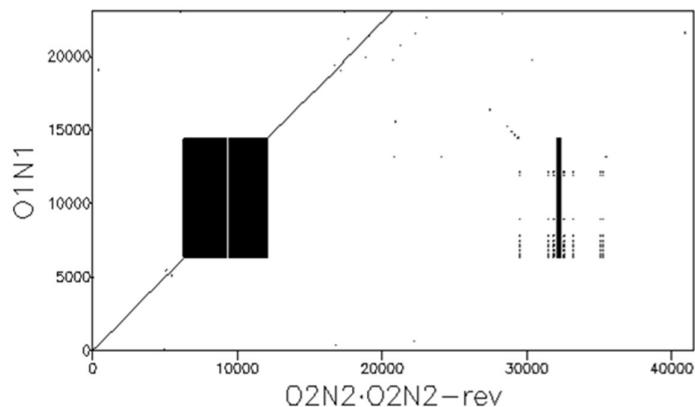


Figure 10: Dotplot du fichier O1N1 corrigé par le script put_start.sh

L'image ci-dessus montre que le morceau de la séquence a bien été décalé. Le graphe suit un tracé sur toute la diagonale.

Le second problème le plus fréquent est une duplication de séquence.

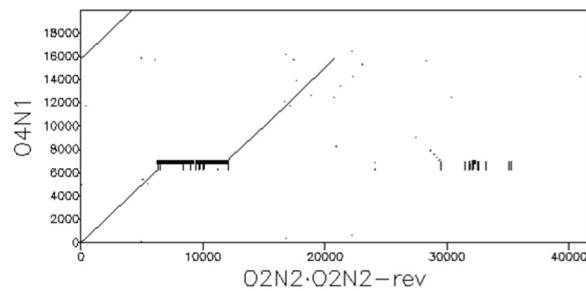


Figure 11: Exemple de fichier ayant une duplication

Afin de corriger ces fichiers, **merger** a encore été utilisé pour trouver les positions précises de duplication. Comme précédemment, la correction manuelle de ces problèmes étant très fastidieuse, un script `deduplicate.sh` (voir annexe figure 22) a permis de régler ces problèmes.

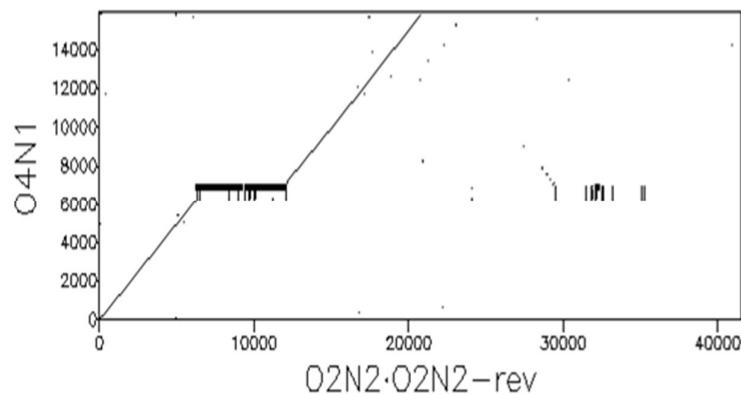


Figure 12: Dotplot du fichier corrigé par le script `deduplicate.sh`

Pour le fichier à inverser, l'outil **Revseq** d'EMBOSS a été utilisé. Puis, la séquence a été décalée avec le script `put_start.sh`.

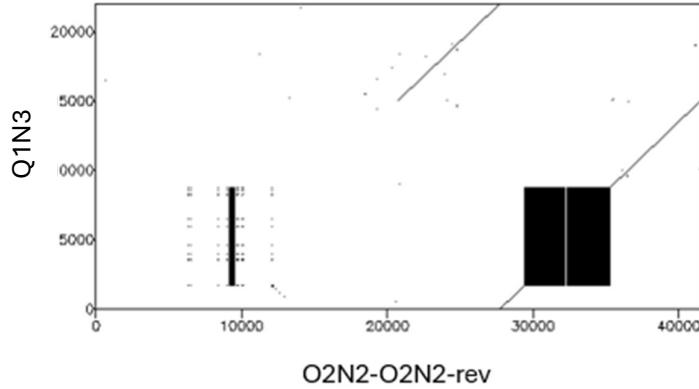


Figure 13: Dotplot de fichier à inverser

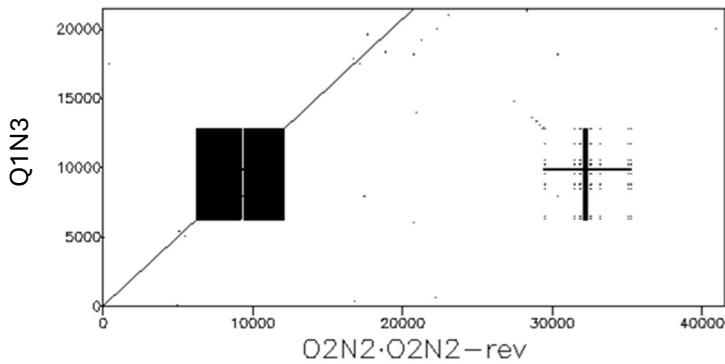


Figure 14: Dotplot du fichier corrigé avec revseq

Après toutes ces corrections, les fichiers ont pu être alignés.

Alignement des séquences

L'alignement réalisé utilise l'outil **BWA** pour produire un fichier sam utilisable dans **Tablet**. Avant de procéder à l'alignement proprement dit, il est nécessaire de créer un index de la séquence de référence à l'aide de la commande suivante :

```
bwa index reference.fasta
```

L'index de la référence est indispensable car il permet à BWA de fonctionner de manière beaucoup plus rapide, efficace et précise. Cet index optimise la recherche des régions homologues lors de l'alignement des séquences sur la séquence de référence.

Plusieurs alignement ont été réalisés. Le premier correspond à l'alignement de tous les fichiers contre la référence afin de montrer les différences globales qu'il y a entre les fichiers.

Ensuite, un second alignement a été réalisé entre la colonie n°1 et la référence dans le but de mettre en évidence les différences spécifiques à cette colonie. Enfin, le dernier alignement a été réalisé entre les reines et la référence pour déterminer si des variants spécifiques sont présents entre reines et ouvrières.

Voici un exemple pour la création du fichier contenant toutes les séquences :

```
cat *.fasta > allseq.fasta
```

Ce fichier contient toutes les séquences séparées par leur nom. Lorsque ce fichier est créé, l'alignement peut être effectué.

```
/opt/bwa/bwa mem reference.fasta allseq.fasta > allseq.sam
```

Une fois l'alignement réalisé, un fichier avec l'extension « .sam » (Sequence Alignment/Map) est obtenu. Ce fichier contient les résultats de l'alignement, incluant des informations détaillées comme :

- Le nom de la séquence de lecture,
- La position de l'alignement sur la séquence de référence,
- La qualité de l'alignement,
- D'autres données fournissant des informations supplémentaires sur l'alignement.

Une fois le fichier sam obtenu, il doit être converti en un fichier « .bam », puis trié pour obtenir un fichier « .s.bam » (sorted BAM). Cette étape est obligatoire car certains outils, comme **bcftools mpileup**, nécessitent des fichiers sorted BAM pour fonctionner correctement.

Les grandes différences entre les fichiers sam et s.bam sont :

Tableau 2: Récapitulatif de différences entre les fichiers sam et sorted.bam

Caractéristiques	Fichier SAM	Fichier sorted BAM
Format	Texte lisible, non compressé	Binaire, compressé
Taille du fichier	Grande (en fonction de la taille des alignements)	Plus petite en raison de la compression
Lisibilité	Lisible par l'homme (texte brut)	Non lisible directement (format binaire)
Vitesse de traitement	Lente à traiter	Plus rapide à traiter
Tri des alignements	Non trié	Alignements triés par position sur la séquence de référence
Utilisation en analyse	Utilisé pour l'examen initial des alignements	Préparé pour des analyses en aval, comme la détection de variants
Compatibilité avec les outils	Lisible par les outils compatibles avec SAM	Requis pour de nombreux outils bioinformatiques, comme Bcftools
Indexation possible	Non indexé	Peut être indexé (génération d'un fichier .bai) pour un accès rapide

La conversion du fichier « .sam » en un fichier sorted BAM « .s.bam » se fait en deux étapes à l'aide de l'outil **samtools** :

La première étape consiste à convertir le fichier SAM en un fichier BAM, qui est un format binaire plus compact. La commande utilisée est :

```
/opt/samtools/samtools view -S -b allseq.sam > allseq.bam
```

Ensuite, le fichier BAM obtenu est trié par position sur la séquence de référence pour produire un fichier sorted BAM, à l'aide de la commande suivante :

```
/opt/samtools/samtools sort allseq.bam -o allseq.s.bam
```

Cette méthode assure que les fichiers d'alignement soient prêts pour les étapes suivantes, comme la détection de variants, où un tri et une compression des données sont requis pour un fonctionnement optimal.

Identification des variants

L'identification des variants (SNPs et indels) a été réalisée en deux étapes principales en utilisant la suite **BCFtools**.

La première étape consiste à utiliser **bcftools mpileup** pour générer un fichier de pileup à partir des fichiers alignés au format s.bam. Cette étape est essentielle pour établir une base de données complète sur les positions potentiellement variables dans le génome analysé.

La commande utilisée pour cette étape est la suivante :

```
bcftools mpileup -Ou -f reference.fasta input.s.bam > output.mpileup.bcf
```

- -Ou indique que la sortie doit être non compressée en format BCF.
- -f reference.fasta spécifie la séquence de référence utilisée pour l'alignement.

Ensuite, le fichier de pileup généré est passé à la commande **bcftools call** pour identifier et appeler les variants présents dans les séquences analysées. **bcftools call** interprète les données de pileup pour distinguer les positions où des variations existent par rapport à la séquence de référence, en générant un fichier au format **VCF** (Variant Call Format).

La commande utilisée pour cette étape est la suivante :

```
bcftools call -mv -Ov -o variants.vcf output.mpileup.bcf
```

- -mv indique que l'outil doit appeler les SNPs et les indels (modes multialleliques et variants).
- -Ov spécifie que la sortie doit être en format VCF texte.
- -o variants.vcf désigne le fichier de sortie contenant les variants appelés.

Cette combinaison de **bcftools mpileup** et **bcftools call** permet une détection des variants.

Détection des impacts des variants sur les gènes

Une fois les variants identifiés avec Tablet, leur impact sur le génome a été identifié grâce à snpEff. Ce programme utilise une base de données préexistante ou personnalisée afin de trouver l'impact potentiel des variants sur le génome.

Etant donné que *Melipona quadrifasciata* n'est pas une espèce modèle, une base de données personnalisée a dû être créée. Pour cela, un fichier de configuration particulier a été créé (voir annexe figure 20) forçant le programme à utiliser les fichiers stockés localement pour créer la base de données.

Pour utiliser une base de données personnalisée, un dossier data contenant un dossier portant le nom de notre base de données doit être créé dans le répertoire de snpEff. Le dossier créé dans le répertoire data a été appelé Melipona_O2N2. Ensuite, la base de données personnalisée est créée à partir d'un génome de référence et d'un fichier d'annotations au format gff. Ces deux fichiers doivent se trouver dans le dossier data/Melipona_O2N2 contenu dans le répertoire de snpEff. De plus, ces fichiers doivent être renommés en séquence.fasta pour le génome de référence et genome.gff pour le fichier d'annotations.

Une fois les fichiers correctement formatés et le fichier de configuration créé, la base de données peut être fabriquée. Pour cela, la commande suivante est effectuée :

```
java -jar /opt/snpEff/snpEff.jar build -gff3 -noCheckCds -noCheckProtein -v Melipona_O2N2
```

- -gff3 spécifie le type de fichier utilisé pour l'annotation des gènes (gff).
- -noCheckCds empêche snpEff de vérifier les CDS car les CDS contenus dans le fichier gff ne sont pas toujours complets.
- -noCheckProtein empêche snpEff de vérifier la pertinence des CDS traduits en protéines car les CDS contenus dans le gff ne sont pas toujours complets.

Une autre particularité du génome mitochondrial est que son code génétique est différent du code génétique du noyau. Ce code génétique particulier a donc été ajouté au fichier de configuration. Le programme tient donc compte de ces différences lors de l'évaluation des impacts.

Une fois toutes ces étapes réalisées, snpEff a été lancé avec la commande suivante :

```
java -jar /opt/snpEff/snpEff.jar ann -v Melipona_O2N2 allseq.vcf > allseq_ann.vcf
```

Une des particularités de snpEff est qu'il produit un fichier html contenant un rapport complet de tous les variants ainsi que leurs impacts. Grâce à cela, les variants ainsi que leurs impacts ont pu être mis en évidence.

La limitation de cette analyse est la présence de la région très répétitive. Cette région très répétitive induit beaucoup d'erreurs de séquençage. Ce qui donne donc un nombre élevé de variants dans ces régions, sans pour autant que ces variants soient corrects.

Afin de palier à ce problème, les variants identifiés dans la région très répétitive ont été supprimés du fichier vcf manuellement et une nouvelle analyse snpEff a été lancée. Cela a considérablement réduit le nombre de variants présents (de 144 à 14).

Visualisation de l'alignement

L'étape suivante consiste à utiliser **Tablet** pour visualiser l'alignement contenu dans le fichier sam combiné au fichier vcf créé par bcftools. Cela permet de montrer les endroits contenant des variants. De plus, le fichier gff de la référence permet de montrer les gènes affectés par un variant. Comme cela, les impacts des variants détectés par snpEff peuvent être visibles directement dans une interface graphique.

Afin d'automatiser tout le processus, un script est écrit (voir annexe point 4). Ce script permet d'exécuter toutes les étapes de l'analyse allant de l'alignement à la détection des impacts avec snpEff. Cela permet un gain de temps important.

Seconde analyse

Après réflexion avec Monsieur Coornaert, il a été démontré que **BWA** et **Bcftools** ne sont pas conçus pour fonctionner de manière optimale avec des séquences assemblées mais avec des reads (morceaux de séquences issus directement du séquençage). L'alignement qui est obtenu ainsi que les variants mis en évidence sont donc peu fiables.

Un autre outil a donc été suggéré ; il s'agit de **ClustalX**. Ce programme permet de réaliser une multiplicité d'alignements entre toutes les séquences afin de voir les séquences superposées et de trouver les potentiels variants.

De plus, une discussion avec Natalia a mis en évidence que le génome de référence utilisé n'est pas celui de l'ouvrière 2 de la colonie 2 mais celui de l'ouvrière 2 de la colonie 1.

Prétraitement des données

Les corrections appliquées aux fichiers sont basées sur le génome de l'ouvrière 2 de la colonie 2. Ce génome n'est pas la référence utilisée par Natalia dans ses analyses. Une nouvelle correction a été réalisée en utilisant le génome de l'ouvrière 2 de la colonie 1. Les 2 génomes étant assez similaires, les fichiers à modifier sont les mêmes. La seule différence réside dans le fait que l'ouvrière 2 de la colonie 1 possède une partie répétitive plus petite.

Dotmatcher: fasta::O2N1.fasta;O2N1 vs fasta::O2N2.fasta;...
(windowsize = 20, threshold = 100.00 26/08/24)

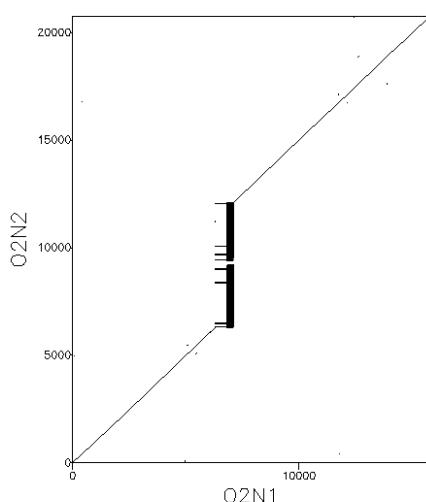


Figure 15: Comparaison des 2 génomes de référence

Etant donné que la région très répétitive ne contient aucun gène, une autre correction peut être réalisée afin de faciliter l'alignement. Cela consiste à placer les morceaux très répétitifs aux extrémités des séquences. En organisant les séquences de cette manière, la séquence d'intérêt se trouve au milieu et la séquence très répétitive est coupée en 2 et ne gêne pas l'alignement. La position 7000 a été utilisée comme site de coupure pour chaque fichier avec le script `put_start.sh`. Les nouveaux fichiers fasta ont donc une structure totalement différente car la région très répétitive n'est plus au centre mais aux extrémités.

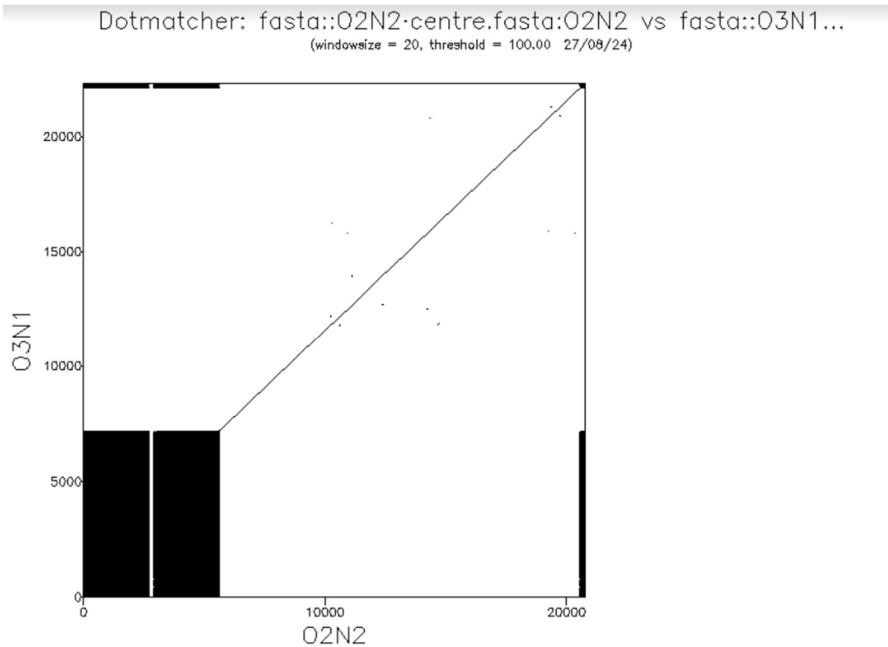


Figure 16: Dotplot des fichiers ayant les séquences très répétitives aux extrémités

Alignement et identification des variants

Une fois les corrections appliquées, ClustalX est utilisé. Il s'agit d'un programme permettant d'aligner plusieurs séquences entre elles. Cela permet de visualiser directement les variants potentiels entre les séquences.

Un fichier contenant toutes les séquences groupées en colonies a été créé avec :

```
cat allcolonie1.fasta allcolonie2.fasta allcolonie3.fasta > allseq.fasta
```

Ensuite, ce fichier a été envoyé dans **ClustalX** pour l'alignement. ClustalX est doté d'un affichage graphique ce qui permet de directement visualiser l'alignement.

Celui-ci permet de trouver facilement les variants entre les séquences. En effet, les séquences parfaitement alignées entre elles ont une étoile sur le nucléotide associé. En cherchant les positions où les étoiles sont absentes, il est assez facile de trouver les variants.

16 variants ont été identifiés de cette manière⁴. Ces variants sont de 3 types ; SNP, insertion ou délétion.

⁴ Toutes les captures d'écran réalisées sont en annexe point 9

Figure 17: Capture d'écran d'une insertion dans la colonie 1

Q2N2	A	T	A	T	A	T	A	A	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q3N3	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q2N3	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q1N2	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q1N3	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q2N3	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q4N2	T	A	T	A	T	A	T	T	A	A	T	G	A	A	G	T	A	T	A	A	T	G	T	A
Q3N3	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q4N3	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q2N2	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q1N3	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q1N2	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q3N2	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q3N2	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q3N2	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q2N2	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q4N2	T	A	A	T	A	T	A	T	A	T	T	A	A	T	G	A	A	T	A	A	T	G	T	A
Q2N1	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q1N2	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q1N1	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q3N1	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	
Q3N1	A	T	A	T	A	T	A	T	A	T	G	A	A	G	T	A	T	A	A	G	T	A	T	

Figure 18: Capture d'écran d'une délétion et d'un SNP dans la colonie 1

ClustalX permet également de mettre en évidence les erreurs de séquençage qui ont eu lieu. Lorsqu'une région contient un grand nombre de répétitions d'un même nucléotide ou d'un même groupe de nucléotides, le séquenceur peut omettre un nucléotide. Cela entraîne des biais de séquençage dans les résultats.

Figure 19: Capture d'écran d'un biais de séquencage

La limitation de cette approche est qu'aucune référence n'est utilisée. Cela a comme conséquence que les gènes impliqués ne peuvent pas être identifiés comme avec Tablet. Donc les impacts génétiques des variants sont plus compliqués à déterminer.

Détermination de l'impact génétique

Afin de trouver les impacts potentiels des variants sur le génome, les fichiers contenant les séquences nucléotidiques des gènes ont été utilisés. En alignant le génome d'une abeille avec ses gènes, il est possible de déterminer les positions de début et de fin des gènes. Ensuite, prettyplot permet de visualiser l'alignement produit par Clustalx avec les positions et de trouver les endroits où un SNP est dans un gène.

Résultats et interprétation

Première analyse

Les résultats obtenus sont basés sur le fichier vcf modifié pour enlever les variants présents dans la région très répétitive. Les résultats obtenus ont montré la présence de 14 variants. Ces variants sont des SNPs, des insertions ou des délétions.

Variants rate details			
Chromosome	Length	Variants	Variants rate
O2N2	20,772	14	1,483
Total	20,772	14	1,483

Number variants by type	
Type	Total
SNP	7
MNP	0
INS	3
DEL	4
MIXED	0
INV	0
DUP	0
BND	0
INTERVAL	0
Total	14

Figure 20: Capture d'écran du nombre de variants du fichier `snpEff_summary.html`

Grâce à la visualisation du fichier sam combiné au fichier vcf dans Tablet, il a été possible de voir directement à quels endroits sont présent ces variants. De plus, en y ajoutant le fichier gff de la référence, il est possible de voir dans quels gènes ces variants apparaissent. Parmi ces variants, 7 provoquent un changement de nucléotide. Le fichier html créé par snpEff permet de voir dans quelle mesure ces variants affectent les gènes.

	-	AAA	AAT	ATT	GTT	TAA	TAT	TCT	TTC	TTT
-						1				
AAA										
AAT	1									
ATT					1					
GTT										
TAA										
TAT							1			
TCT										1
TTC										
TTT	1									

Figure 21: Capture d'écran des codons modifiés par les variants contenu dans le fichier `snpEff_summary.html`

Ce tableau présent dans le fichier html produit par snpEff récapitule les codons qui ont été modifiés.

Bien que le premier pipeline utilisé ait été considéré partiellement incorrect, il a tout de même permis de mettre en évidence la présence de variants. En effet, 14 variants ont pu être mis en

évidence, tous provenant de la colonie 1. De plus, les scripts créés pour corriger les fichiers ont été réutilisés lors de la seconde analyse.

Seconde analyse

ClustalX aligne les séquences entre elles et regroupe les séquences similaires. Comme le fichier d'entrée est ordonné par colonies, l'alignement en sortie devrait être ordonné de la même manière. En effet, les individus d'une même colonie, c'est-à-dire issus d'une même reine, sont censés avoir le même génome mitochondrial.

L'alignement en sortie de ClustalX montre que les fichiers ont été réordonnés.

Q2N2	AAAAATTTATTTAT
O3N3	AAAAATTTATTTAT
O2N3	AAAAATTTATTTAT
O1N2	AAAAATTTATTTAT
Q1N3	AAAAATTTATTTAT
Q2N3	AAAAATTTATTTAT
O4N2	AAAAATTTATTTAT
Q3N3	AAAAATTTATTTAT
O4N3	AAAAATTTATTTAT
O2N2	AAAAATTTATTTAT
O1N3	AAAAATTTATTTAT
Q1N2	AAAAATTTATTTAT
Q3N2	AAAAATTTATTTAT
O3N2	AAAAATTTATTTAT
O2N1	AAAAATTTATTTAT
O4N1	AAAAATTTATTTAT
Q2N1	AAAAATTTATTTAT
O1N1	AAAAATTTATTTAT
Q1N1	AAAAATTTATTTAT
Q3N1	AAAAATTTATTTAT
O3N1	AAAAATTTATTTAT

Figure 22: Démonstration de la réorganisation des fichiers par Clustal

La figure ci-dessus permet de montrer que les génomes des colonies 2 et 3 sont assez similaires comparé à la colonie 1 car les génomes de la colonie 2 et 3 sont entremêlés tandis que les génomes de la colonie 1 sont regroupés.

Etant donné que les séquences ont été corrigées de telle manière à ce que les morceaux très répétitifs soient placés aux extrémités, le début et la fin des alignements doivent-être ignorés. La séquence d'intérêt commence donc après le morceau très répétitif du début et se termine avant celui présent à la fin.

L'examen de l'alignement permet de mettre en évidence 16 variants, tous présents dans la colonie 1. Ces variants sont soit des insertions, soit des délétions, soit des SNPs.

Tableau 3: Récapitulatif des variants identifiés via ClustalX

/		IN : A(T)ATAATTA	DEL : T	SNP : T->A	SNP : T->C	DEL : TA	IN : A	SNP : A->G	IN : TTTA
Col 1	O1	V	V	V	V	V	V	V	V
	O2	V	V	V	V	V	V	V	V
	O3	V	V	V	V	V	V	V	V
	O4	V	V	V	V	V	V	V	V
	R1	V	V	V	V	V	V	V	V
	R2	V	V	V	V	V	V	V	V
	R3	V	V	V	V	V	V	V	V
Col 2	O1	F	F	F	F	F	F	F	F
	O2	F	F	F	F	F	F	F	F
	O3	F	F	F	F	F	F	F	F
	O4	F	F	F	F	F	F	F	F
	R1	F	F	F	F	F	F	F	F
	R2	F	F	F	F	F	F	F	F
	R3	F	F	F	F	F	F	F	F
Col 3	O1	F	F	F	F	F	F	F	F
	O2	F	F	F	F	F	F	F	F
	O3	F	F	F	F	F	F	F	F
	O4	F	F	F	F	F	F	F	F
	R1	F	F	F	F	F	F	F	F
	R2	F	F	F	F	F	F	F	F
	R3	F	F	F	F	F	F	F	F
/		DEL : ATAT	DEL : GAA	SNP : A->T	SNP : A->G	SNP : C->T	IN : A	SNP : A->G	SNP : A->G
Col 1	O1	V	V	V	V	V	V	V	V
	O2	V	V	V	V	V	V	V	V
	O3	V	V	V	V	V	V	V	V
	O4	V	V	V	V	V	V	V	V
	R1	V	V	V	V	V	V	V	V
	R2	V	V	V	V	V	V	V	V
	R3	V	V	V	V	V	V	V	V
Col 2	O1	F	F	F	F	F	F	F	F
	O2	F	F	F	F	F	F	F	F
	O3	F	F	F	F	F	F	F	F
	O4	F	F	F	F	F	F	F	F
	R1	F	F	F	F	F	F	F	F
	R2	F	F	F	F	F	F	F	F
	R3	F	F	F	F	F	F	F	F
Col 3	O1	F	F	F	F	F	F	F	F
	O2	F	F	F	F	F	F	F	F
	O3	F	F	F	F	F	F	F	F
	O4	F	F	F	F	F	F	F	F
	R1	F	F	F	F	F	F	F	F
	R2	F	F	F	F	F	F	F	F
	R3	F	F	F	F	F	F	F	F

Ce tableau montre que tous les variants trouvés sont présents dans la colonie 1. Aucun variant n'a été trouvé entre la colonie 2 et 3. Cela implique que la colonie 1 a un génotype particulier par rapport aux 2 autres.

De plus, aucun variant spécifique à une caste n'a été détecté. En effet, à l'exception des variants spécifiques à la colonie 1, aucun autre variant n'a été trouvé spécifiquement chez les reines par rapport aux ouvrières.

L'impact de ces variants ne peut pas être mesuré avec snpEff étant donné qu'aucune référence n'est utilisée lors de l'alignement ClustalX. Cependant, une recherche manuelle a mis en évidence le fait que 3 SNPs, une insertion et une délétion sont présents dans les gènes ND2, ND3, ND4 et ATP8.

Tableau 4: Récapitulatif des gènes modifiés par les variants

Gènes impliqués	Variants
ND4	SNP : A -> G SNP : C -> T
ND3	IN : TTTA
ATP8	DEL : GAA
ND2	SNP : A -> G

Ce tableau récapitule les gènes qui ont subi une variation.

La fonction des gènes qui contiennent une variation est cruciale pour savoir dans quelle mesure ces variations sont importantes.

- ND4, ND3, ND2 : Les protéines ND sont des sous-unités de la NADH déshydrogénase. C'est le plus grand des 5 complexes de la chaîne de transports d'électrons. Ce complexe, aussi appelé complexe I, est crucial pour la production d'énergie cellulaire. Une variation au sein de ces gènes peut donc avoir de graves conséquences. (Arcadian, MT-ND4, 2008)
- ATP8 : Gène codant pour une petite sous-unité de l'ATP synthase. Celui-ci étant responsable de la production d'ATP. Une variation dans cette sous-unité peut entraîner un mauvais fonctionnement de l'ATP synthase. Cela peut avoir de graves conséquences. (Arcadian, MT-ATP8, 2008)

Discussion

L'objectif de ce travail était de chercher des différences génétiques dans l'ADN mitochondrial des abeilles *Melipona quadrifasciata*, afin de détecter d'éventuels variants responsables de la différenciation en castes, ainsi que de comprendre la diversité génétique entre les colonies.

Des variants ont été mis en évidence entre la colonie 1 et les deux autres colonies, indiquant que ces variants sont spécifiques à cette colonie car ils sont présents tant chez les reines que chez les ouvrières. Ces variants se présentent sous trois formes : SNPs, insertions et délétions. L'ADN mitochondrial, en tant que molécule génétique distincte du noyau, transmise exclusivement par la mère, s'est révélé un outil efficace pour identifier ces différences inter colonies. Cependant, aucune différence intra coloniale n'a été constatée, ce qui corrobore le fait que tous les individus d'une même colonie partagent un génome mitochondrial identique, hérité de leur mère commune.

Les variants détectés ont potentiellement un impact sur les gènes où ils se trouvent. Néanmoins, le fait que ces variants soient présents chez tous les individus d'une même colonie, qui continue de prospérer, suggère qu'ils n'ont pas d'effet délétère significatif sur la survie ou la fonction des abeilles. Cette constatation souligne l'importance de ces variants dans l'adaptation ou la stabilité génétique des colonies, bien que leur impact fonctionnel exact sur les protéines mitochondrielles reste à explorer.

Même si cette étude se concentre exclusivement sur l'ADN mitochondrial, l'une de ses limites majeures est de ne pas inclure l'ensemble du génome, qui pourrait révéler davantage de variations responsables de la différenciation des castes. En outre, les données analysées provenaient d'un ADN mitochondrial déjà assemblé, limitant ainsi la profondeur de l'analyse. Une étude future incluant l'assemblage des reads séquencés et l'analyse du génome nucléaire offrirait une vue plus complète de la variation génétique et des mécanismes de différenciation des castes.

Pour affiner ces analyses, il serait intéressant de réaliser une étude complète à partir des reads séquencés, en incluant l'assemblage du génome entier et l'identification des variants. Cela permettrait de confirmer les résultats obtenus et d'explorer des aspects encore inexploités du génome de *Melipona quadrifasciata*, notamment la variation nucléaire et son rôle potentiel dans la différenciation des castes.

En conclusion, l'objectif principal de cette étude, qui était d'identifier des variants inter colonies et intra colonies, a été partiellement atteint. Des variants inter colonies ont été identifiés, mais aucune variation intra coloniale n'a été détectée. Ces résultats posent les bases pour des recherches futures qui pourraient intégrer des analyses plus approfondies du génome nucléaire et des études fonctionnelles des variants identifiés, afin de mieux comprendre les mécanismes génétiques à l'œuvre dans la différenciation des castes chez les abeilles *Melipona quadrifasciata*.

Conclusion

Cette étude avait pour objectif d'identifier des différences génétiques dans l'ADN mitochondrial des abeilles *Melipona quadrifasciata*, en se concentrant sur la diversité inter colonie et intra colonies. Les analyses ont révélé des variants spécifiques à la colonie 1, comprenant des SNPs, des insertions et des délétions, présents tant chez les reines que chez les ouvrières de cette colonie. Aucune variation génétique n'a été observée au sein des autres colonies, ce qui confirme la transmission maternelle de l'ADN mitochondrial et souligne la stabilité génétique intra coloniale.

Les variants détectés dans les gènes mitochondriaux, notamment ND4 et ATP8, semblent n'avoir qu'un impact limité sur la survie et la fonctionnalité des abeilles, suggérant qu'ils pourraient être neutres ou légèrement bénéfiques pour la colonie 1. Cette observation indique que ces variants ne compromettent pas la viabilité des colonies, mais pourraient potentiellement jouer un rôle dans l'adaptation locale ou la spécialisation fonctionnelle des colonies.

Cependant, cette étude présente certaines limites, notamment l'analyse restreinte à l'ADN mitochondrial, qui ne reflète qu'une partie de la diversité génomique totale. De plus, l'utilisation de données d'ADN mitochondrial déjà assemblées a limité la profondeur des analyses, empêchant une exploration plus détaillée des variations présentes. Pour une compréhension plus complète des mécanismes génétiques sous-jacents à la différenciation des castes, il est essentiel d'examiner le génome nucléaire dans son intégralité et de réaliser des analyses fonctionnelles des variants identifiés.

En perspective, des recherches futures devraient intégrer le séquençage du génome entier afin de capturer une image complète de la variation génétique chez *Melipona quadrifasciata*. Une telle approche permettrait non seulement de confirmer les variants mitochondriaux identifiés, mais aussi de découvrir des variants nucléaires potentiellement impliqués dans la différenciation des castes. De plus, des études fonctionnelles sur les variants mitochondriaux pourraient élucider leur rôle exact dans la biologie et l'adaptation des abeilles, offrant des informations précieuses pour la compréhension du fonctionnement des insectes eusociaux.

En conclusion, cette étude a réussi à identifier des variants génétiques spécifiques à une colonie, enrichissant ainsi la connaissance de la diversité génétique chez *Melipona quadrifasciata*. Ces résultats sont un point de départ pour des investigations futures visant à explorer les aspects génétiques et fonctionnels de la différenciation des castes et de l'adaptation des colonies, contribuant ainsi à une meilleure compréhension des mécanismes évolutifs impliqués dans la structure sociale des abeilles sans aiguillon.

Bibliographie

- Aragão, A. d. (2019, juillet). Description and phylogeny of the mitochondrial genome of Sabethes. *ScienceDirect*, pp. 607-611.
- Arcadian. (2008, septembre 26). *MT-ATP8*. Récupéré sur Wikipedia:
<https://en.wikipedia.org/wiki/MT-ATP8>
- Arcadian. (2008, septembre 26). *MT-ND4*. Récupéré sur Wikipedia:
<https://en.wikipedia.org/wiki/MT-ND4>
- Bioinfoxpert (Réalisateur). (2020). (*Bioinformatics*) Dotmacher practical of bioinformatics; A Way To Qualitative Compare Two Sequences [Film].
- Cingolani, P. (2024, 9 4). *pcingola.github.io*. Récupéré sur github:
<https://pcingola.github.io/SnpEff/snpeff/introduction/>
- ClemBuntu. (2016, mai 26). *Introduction à l'analyse des SNPs*. Récupéré sur Bioinfo-fr.net:
<https://bioinfo-fr.net/introduction-a-lanalyse-des-snps>
- FAYET, A. (2014, mai). Les abeilles mélipones, Melipona beebei, Meliponiculture. *Fiche pédagogique*.
- França, K. P. (2011, août 2). *forca-da-urucu-verdadeira-melipona*. Récupéré sur Mélipaire du Sertão: <https://meliponariodosertao.blogspot.com/2011/08/forca-da-urucu-verdadeira-melipona.html>
- G.S. Barni, R. S. (2007, Février). Mitochondrial genome differences between the. *ResearchGate*, p. 9.
- Jean.claude. (2013, juin 28). *Melipona*. Récupéré sur Wikipedia:
<https://fr.wikipedia.org/wiki/Melipona>
- John Wiley & Sons, I. (2003). *icb.usp.br*. Récupéré sur
MultipleSequenceAlignmentUsingClustalwAndClustalx:
http://www.icb.usp.br/~biocomp/praticas/aula_07/MultipleSequenceAlignmentUsingClustalwAndClustalx.pdf
- Li, H. (2024, avril 29). *bctools*. Récupéré sur samtools.github.io:
<https://samtools.github.io/bcftools/bcftools.html>
- Longden, I. (1999, juin 1). *dotmatcher*. Récupéré sur emboss.bioinformatics:
<https://emboss.bioinformatics.nl/cgi-bin/emboss/help/dotmatcher>
- Manuela Moreno-Carmona, P. M.-L. (2023, avril 5). Comparative analysis of mitochondrial genomes reveals family-specific architectures and molecular features in scorpions (Arthropoda: Arachnida: Scorpiones). *ScienceDirect*, p. volume 859.
- R., L. H. (2013). *bwa*. Récupéré sur manpages.org: <https://manpages.org/bwa>
- Slzbg. (2024, juillet 31). *Génome mitochondrial*. Récupéré sur Wikipedia:
https://fr.wikipedia.org/wiki/G%C3%A9nome_mitochondrial

Sterling-Montealegre, R. A. (2024, janvier 20). Variability and evolution of gene order rearrangement in mitochondrial genomes of arthropods (except Hexapoda). *ScienceDirect*, p. volume 892.

Wdsieling. (2024, février 6). *Melipona quadrifasciata*. Récupéré sur Wikipedia: https://en.wikipedia.org/wiki/Melipona_quadrifasciata

whitwham, j. e. (2021, février 2). *samtools*. Récupéré sur github: <https://github.com/samtools/samtools?tab=readme-ov-file>

Williams, G. (1999). *emboss merger*. Récupéré sur galaxy-iuc.github: <https://galaxy-iuc.github.io/emboss-5.0-docs/merger.html>

Williams, G. (1999, janvier 26). *emboss revseq*. Récupéré sur emboss.open-bio.org: <http://emboss.open-bio.org/rel/rel6/apps/revseq.html>

Annexes

1) Contenu du fichier.snpEff.config

```
1 codon.Invertebrate_Mitochondrial      : TTT/F , TTC/F , TTA/L , TTG/L+, TCT/S , T
2
3 genomes : Melipona_O2N2
4 Melipona_O2N2.genome : Melipona_O2N2
5 Melipona.codonTable : Invertebrate_Mitochondrial|
```

Figure 23: Capture d'écran du fichier.snpEff.config

2) Script pour exécuter dotmatcher automatiquement

```
fasta_dir="/data/TFE/allseq"
output_dir="/data/TFE/allseq/new_ref/imgs"

default_output="dotmatcher.1.png"

for fasta_file in "$fasta_dir"/*.fasta; do
    base_name=$(basename "$fasta_file" .fasta)
    output_file="$output_dir/${base_name}_dotplot.png"
    dotmatcher biplot.fasta "$fasta_file" -window 20 -threshold 100 -graph png
    generated_file=$(ls | grep -E '\.png$')
    if [ -f "$default_output" ]; then
        mv "$default_output" "$output_file"
    fi
done|
```

Figure 24: Capture d'écran du script auto_dotmatcher.sh

3) Script pour dédupliquer les fichiers

```
# Usage : rearange_fasta.sh input.fasta output.fasta start_cut end_cut
input_file=$1
output_file=$2
start_position=$3
end_position=$4

header=$(head -n 1 "$input_file")
sequence=$(tail -n +2 "$input_file" | tr -d '\n')

part1=${sequence:0:$start_position}
part2=${sequence:$end_position}

new_sequence="$part1$part2"
{
echo "$header"
echo "$new_sequence" | fold -w 60
} > "$output_file"
```

Figure 25: Capture d'écran du script deduplicate.sh

4) Script put_start.sh

```
input_fasta=$1
cut_pos=$2
out_fasta=$3

seq_name=$(grep ">" $input_fasta | sed 's/>///')
sequence=$(grep -v ">" $input_fasta | tr -d '\n')

end_part=${sequence:$cut_pos-1}
start_part=${sequence:0:$cut_pos-1}

#end_part=$(grep -v ">" $input_fasta | cut -c $cut_pos-)
#start_part=$(grep -v ">" $input_fasta | cut -c 1-$(( ${cut_pos} - 1 )))

echo ">$seq_name" > $out_fasta
echo "$end_part$start_part" >> $out_fasta
```

Figure 26: Capture d'écran du script put_start.sh

4) Script pour réaliser toute l'analyse 1

```

BWA="/opt/bwa/bwa"
SAMTOOLS_PATH="/opt/samtools"
BCFTOOLS_PATH="/opt/bcftools"
SNPEFF_PATH="/opt/snpEff"

# Dossier contenant les fichiers FASTA
FASTA_DIR=$1
ref=$3

# Créer un dossier de sortie si nécessaire
OUTPUT_DIR=$2
mkdir -p "${OUTPUT_DIR}"

# nom de la BDD.snpEff à utiliser
BDDSNP=$4

# Boucle sur chaque fichier FASTA dans le dossier
for FASTA_FILE in ${FASTA_DIR}/*.fasta; do
    BASENAME=$(basename "${FASTA_FILE}" .fasta)
    OUTPUT_SUBDIR="${OUTPUT_DIR}/${BASENAME}"

    # Créer un sous-dossier pour ce fichier FASTA
    mkdir -p "${OUTPUT_SUBDIR}"

    # Étape 1 : Alignement avec Bowtie2
    echo "Aligning ${BASENAME} with Bowtie2..."
    "${BWA}" mem "$ref" "${FASTA_FILE}" > "${OUTPUT_SUBDIR}/${BASENAME}.sam"

    # Étape 2 : Conversion SAM à BAM
    echo "Converting SAM to BAM for ${BASENAME}..."
    "${SAMTOOLS_PATH}/samtools" view -Sb "${OUTPUT_SUBDIR}/${BASENAME}.sam" > "${OUTPUT_SUBDIR}/${BASENAME}.bam"

    # Étape 3 : Tri du fichier BAM
    echo "Sorting BAM for ${BASENAME}..."
    "${SAMTOOLS_PATH}/samtools" sort "${OUTPUT_SUBDIR}/${BASENAME}.bam" -o "${OUTPUT_SUBDIR}/${BASENAME}.s.bam"

    # Étape 5 : mpileup et appel de variants avec bcftools
    echo "Calling variants for ${BASENAME}..."
    "${BCFTOOLS_PATH}/bcftools" mpileup -f "$ref" "${OUTPUT_SUBDIR}/${BASENAME}.s.bam" | \
    "${BCFTOOLS_PATH}/bcftools" call -mv -Ob -o "${OUTPUT_SUBDIR}/${BASENAME}.bcf"

    # Étape 6 : Conversion BCF en VCF
    echo "Converting BCF to VCF for ${BASENAME}..."
    "${BCFTOOLS_PATH}/bcftools" view "${OUTPUT_SUBDIR}/${BASENAME}.bcf" -Ov -o "${OUTPUT_SUBDIR}/${BASENAME}.vcf"

    # Étape 7 : Annotation avec SnpEff
    echo "Annotating variants for ${BASENAME} with SnpEff..."
    java -jar /opt/snpEff/snpEff.jar ann -v "$BDDSNP" "${OUTPUT_SUBDIR}/${BASENAME}.vcf" > "${OUTPUT_SUBDIR}/${BASENAME}_annotated.vcf"

    # Génération de rapport HTML (optionnel)
    java -jar "${SNPEFF_PATH}/snpEff.jar" ann -v "$BDDSNP" -stats "${OUTPUT_SUBDIR}/${BASENAME}_snpEff.html" "${OUTPUT_SUBDIR}/${BASENAME}.vcf"

    echo "Finished processing ${BASENAME}.""
done

echo "All files processed."

```

Figure 27: Capture d'écran du script auto_snpEff.sh

5) Capture d'écran de la page html générée avec snpEff avant épuration du fichier vcf

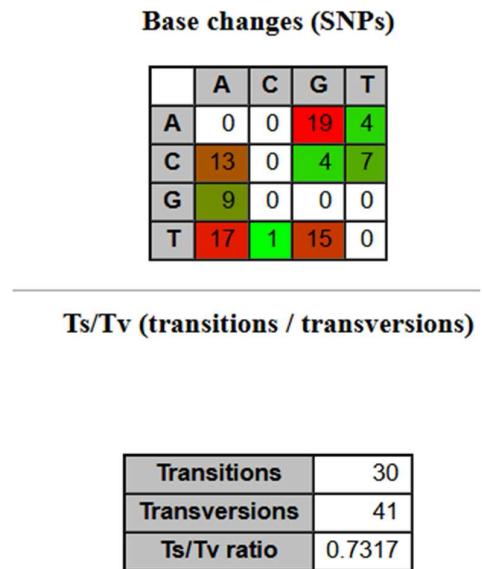


Figure 28: Bases modifiées par les variants dans toutes les séquences

	-	AAA	AAT	ATT	GTT	TAA	TAT	TCT	TTC	TTT
-						1				
AAA										
AAT			1							
ATT						1				
GTT										
TAA										
TAT							1			
TCT										1
TTC										
TTT	1									1

Figure 29: Codons modifiés par les variants

	*	-	?	F	I	K	N	S	V	Y
*										
-	1		1							
?										
F		1		1						
I										1
K										
N						1				
S			1							
V										
Y										1

Figure 30: Acides aminés modifiés par les variants

6) Capture d'écran de la page html générée avec snpEff après épuration du fichier vcf

Variants rate details			
Chromosome	Length	Variants	Variants rate
O2N2	20,772	14	1,483
Total	20,772	14	1,483

Number variants by type	
Type	Total
SNP	7
MNP	0
INS	3
DEL	4
MIXED	0
INV	0
DUP	0
BND	0
INTERVAL	0
Total	14

Figure 31: Bases modifiées par les variants dans toutes les séquences à partir du fichier vcf modifié

7) Matrice de score EDNAFULL utilisée par dotmatcher

A	T	G	C	S	W	R	Y	K	M	B	V	H	D	N	
A	5	-4	-4	-4	-4	1	1	-4	-4	1	-4	-1	-1	-1	-2
T	-4	5	-4	-4	-4	1	-4	1	1	-4	-1	-4	-1	-1	-2
G	-4	-4	5	-4	1	-4	1	-4	1	-4	-1	-1	-4	-1	-2
C	-4	-4	-4	5	1	-4	-4	1	-4	1	-1	-1	-1	-4	-2
S	-4	-4	1	1	-1	-4	-2	-2	-2	-2	-1	-1	-3	-3	-1
W	1	1	-4	-4	-4	-1	-2	-2	-2	-2	-3	-3	-1	-1	-1
R	1	-4	1	-4	-2	-2	-1	-4	-2	-2	-3	-1	-3	-1	-1
Y	-4	1	-4	1	-2	-2	-4	-1	-2	-2	-1	-3	-1	-3	-1
K	-4	1	1	-4	-2	-2	-2	-1	-4	-1	-3	-3	-1	-1	-1
M	1	-4	-4	1	-2	-2	-2	-4	-1	-3	-1	-1	-3	-1	-1
B	-4	-1	-1	-1	-1	-3	-3	-1	-1	-3	-1	-2	-2	-2	-1
V	-1	-4	-1	-1	-1	-3	-1	-3	-3	-1	-2	-1	-2	-2	-1
H	-1	-1	-4	-1	-3	-1	-3	-1	-3	-1	-2	-2	-1	-2	-1
D	-1	-1	-1	-4	-3	-1	-1	-3	-1	-3	-2	-2	-2	-1	-1
N	-2	-2	-2	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Figure 32: Captue d'écran de la matrice EDNAFULL utilisée par dotmatcher

8) Images produites par le script auto_dotmatcher.sh

O1N1

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

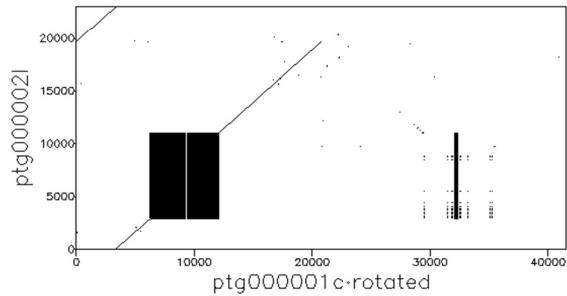


Figure 33: dotplot de biplot.fasta contre O1N1

O2N1

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

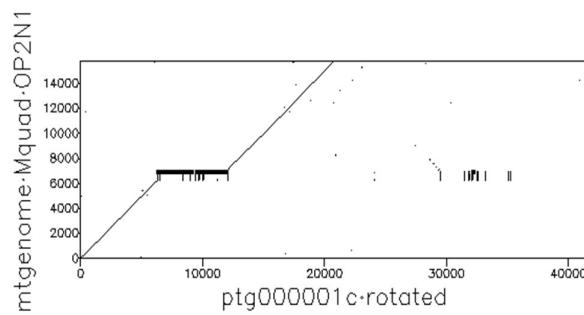


Figure 34: dotplot de biplot.fasta contre O2N1

O3N1

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

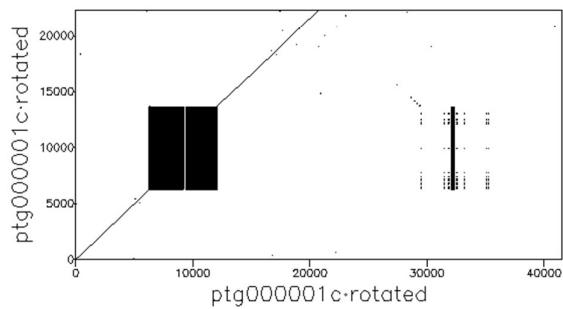


Figure 35: dotplot de biplot.fasta contre O3N1

O4N1

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

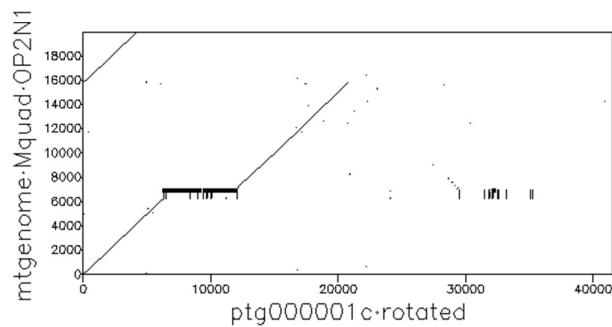


Figure 36: dotplot de biplot.fasta contre O4N1

Q1N1

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

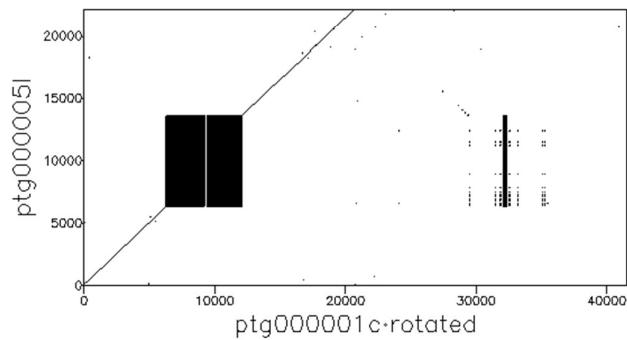


Figure 37: dotplot de biplot.fasta contre Q1N1

Q2N1

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

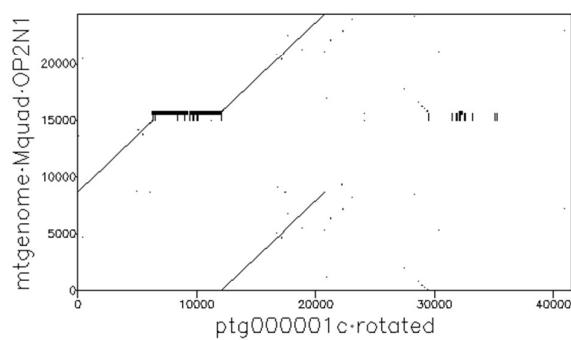


Figure 38: dotplot de biplot.fasta contre Q2N1

Q3N1

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

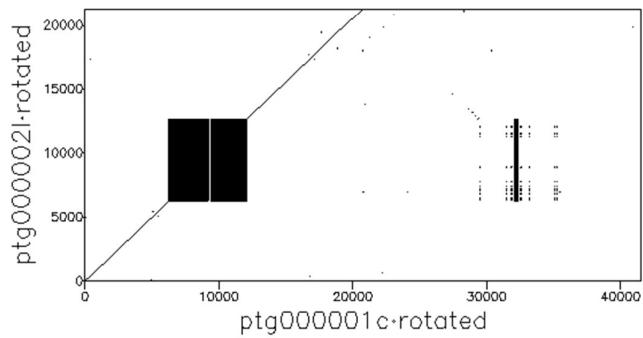


Figure 39: dotplot de biplot.fasta contre Q3N1

O1N2

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

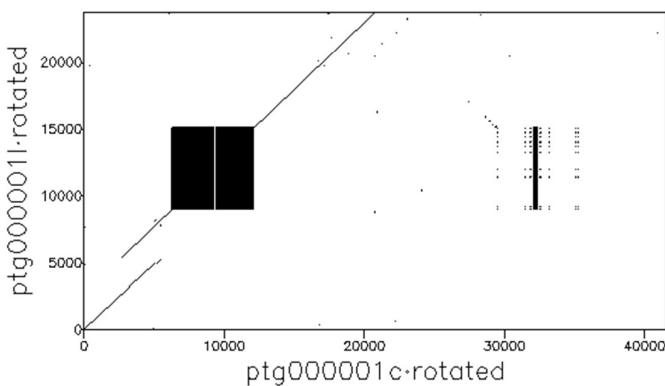


Figure 40: dotplot de biplot.fasta contre O1N2

O2N2

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c·rot...
(windowsize = 20, threshold = 100.00 01/07/24)

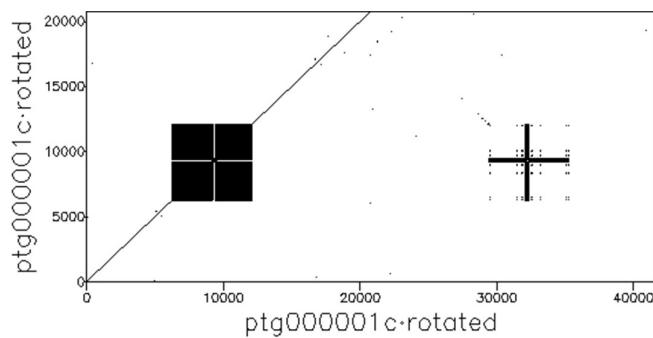


Figure 41: dotplot de biplot.fasta contre O2N2

O3N2

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c·rot...
(windowsize = 20, threshold = 100.00 01/07/24)

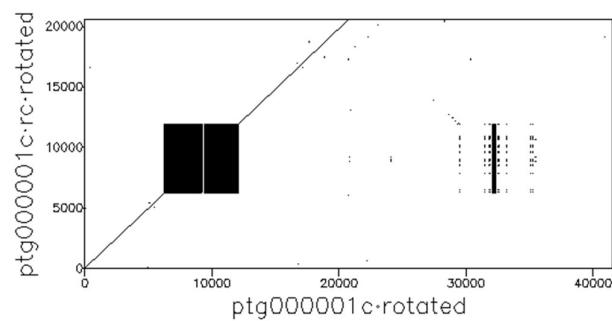


Figure 42: dotplot de biplot.fasta contre O3N2

O4N2

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c·rot...
(windowsize = 20, threshold = 100.00 01/07/24)

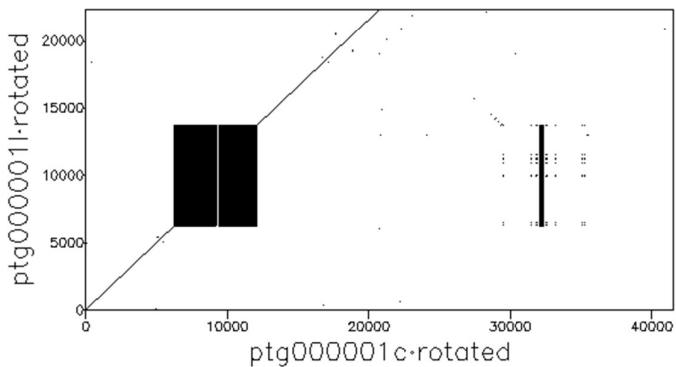


Figure 43: dotplot de biplot.fasta contre O4N2

Q1N2

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c·rot...
(windowsize = 20, threshold = 100.00 01/07/24)

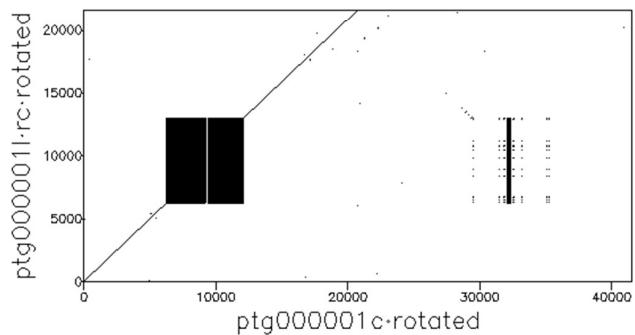


Figure 44: dotplot de biplot.fasta contre Q1N2

Q2N2

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c·rot...
(windowsize = 20, threshold = 100.00 01/07/24)

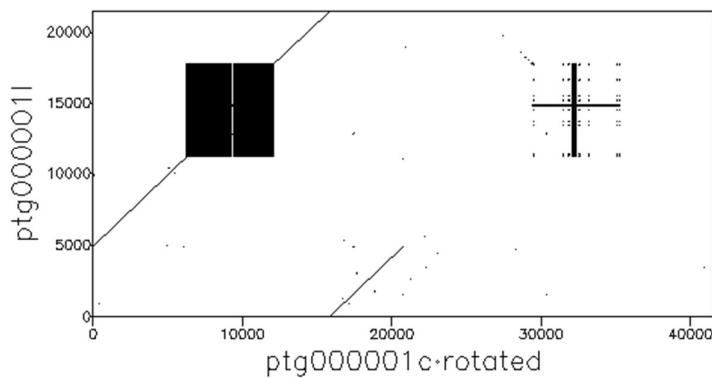


Figure 45: dotplot de biplot.fasta contre Q2N2

Q3N2

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c·rot...
(windowsize = 20, threshold = 100.00 01/07/24)

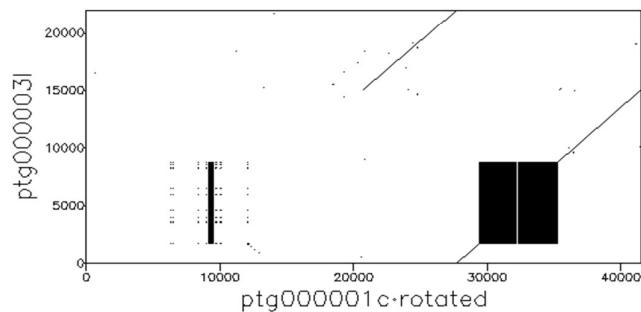


Figure 46: dotplot de biplot.fasta contre Q3N2

O1N3

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c·rot...
(windowsize = 20, threshold = 100.00 01/07/24)

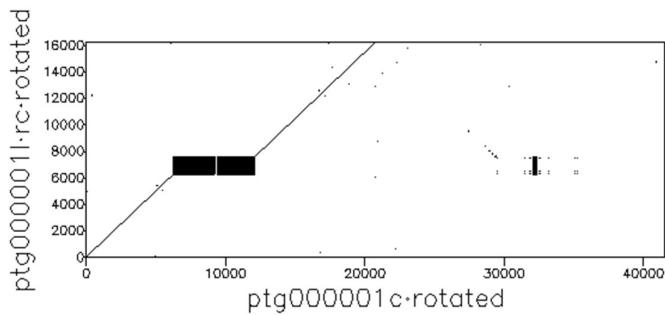


Figure 47: dotplot de biplot.fasta contre O1N3

O2N3

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c·rot...
(windowsize = 20, threshold = 100.00 01/07/24)

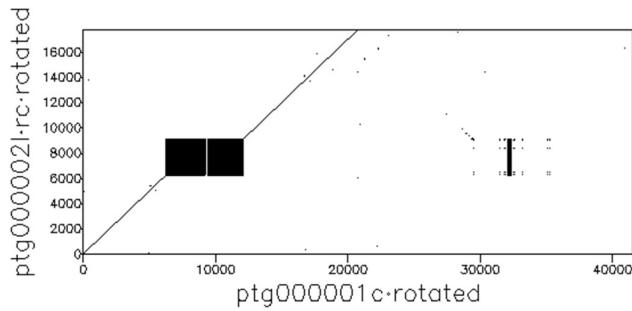


Figure 48: dotplot de biplot.fasta contre O2N3

O3N3

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

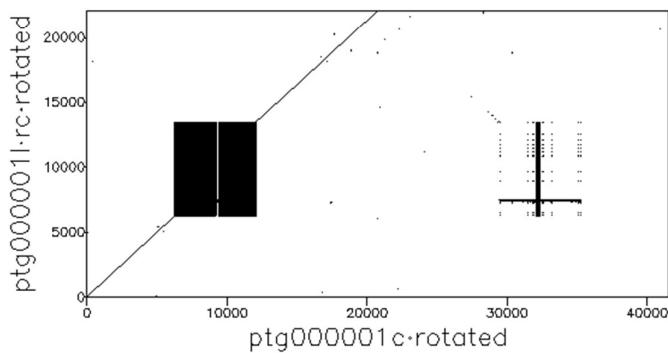


Figure 49: dotplot de biplot.fasta contre O3N3

O4N3

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

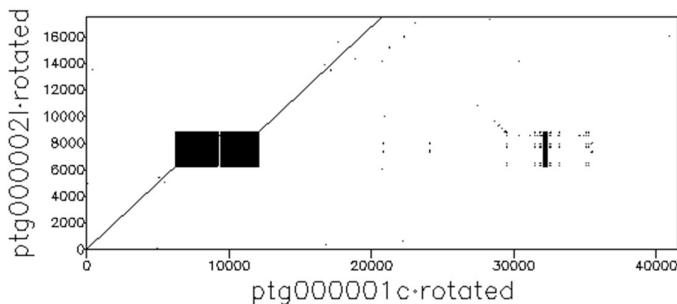


Figure 50: dotplot de biplot.fasta contre O4N3

Q1N3

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

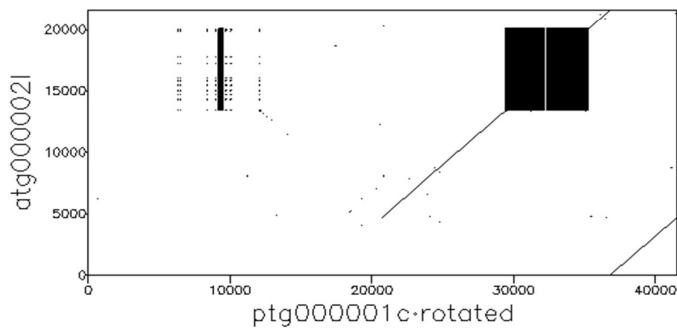


Figure 51: dotplot de `biplot.fasta` contre Q1N3

Q2N3

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

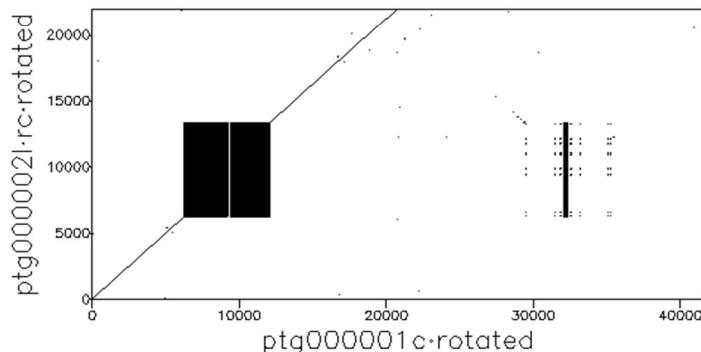


Figure 52: dotplot de `biplot.fasta` contre Q2N3

Q3N3

Dotmatcher: fasta:::/data/TFE/biplot.fasta:ptg000001c.rot...
(windowsize = 20, threshold = 100.00 01/07/24)

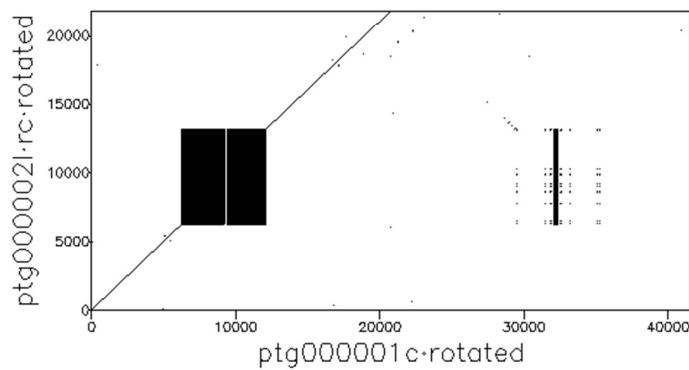


Figure 53: dotplot de biplot.fasta contre Q3N3

9) Captures d'écran de clustalX

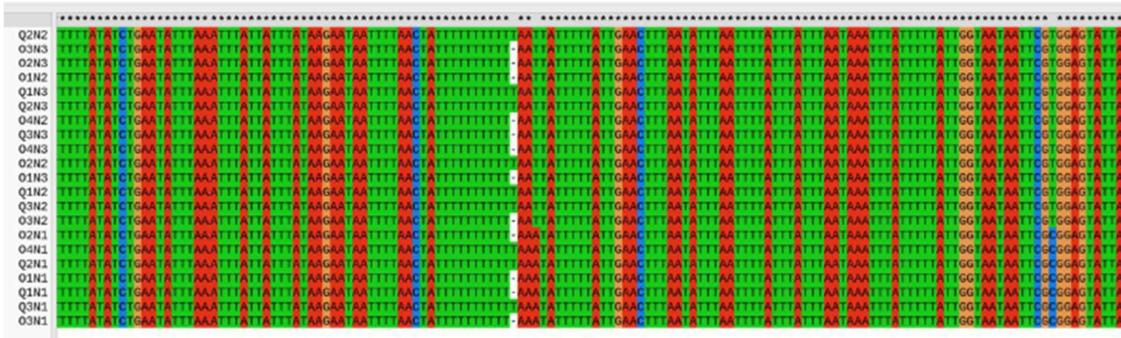


Figure 54: Déletion et SNP dans la colonie 1

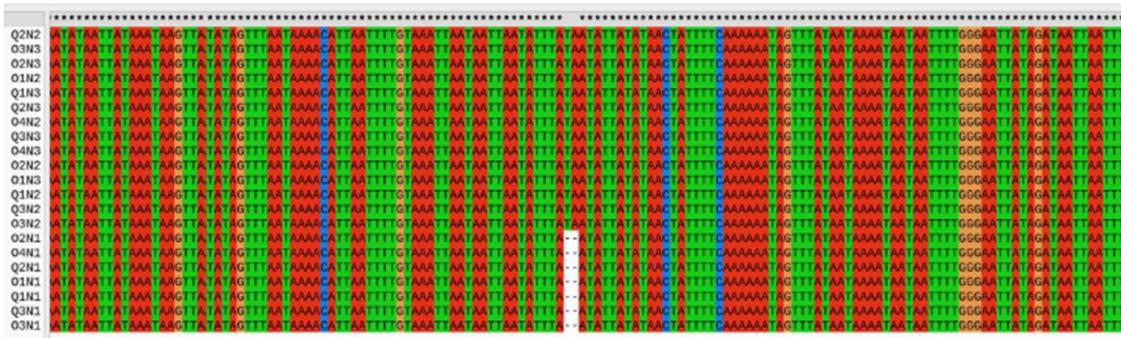


Figure 55: Déletion de TA dans la colonie 1

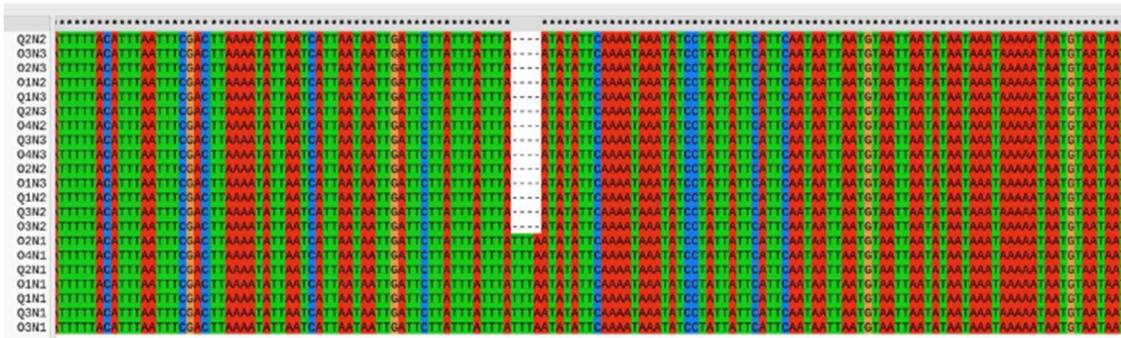


Figure 56: Insertion de TTTA dans la colonie 1

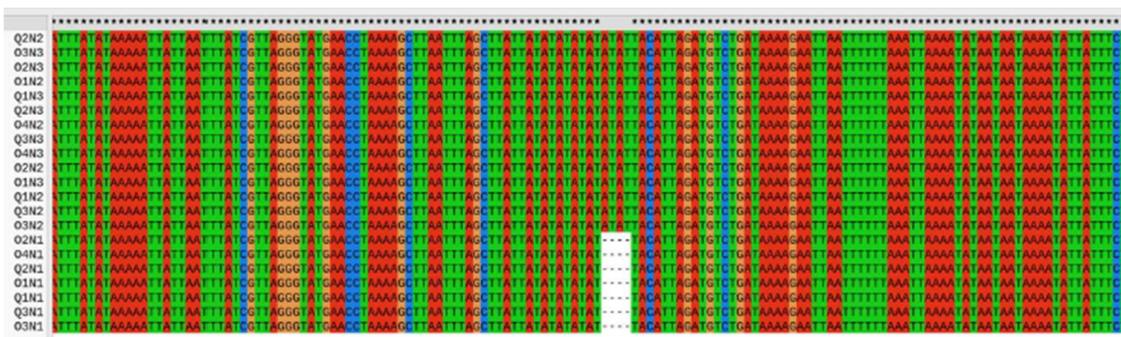


Figure 57: Déletion de ATAT dans la colonie 1

Figure 58: Délétion de GAA dans la colonie 1

Figure 59: SNP de A vers T dans la colonie 1

Figure 60: SNP de A vers G dans la colonie 1

Figure 61: SNP de C vers T dans la colonie 1

Figure 62: Insertion et SNP dans la colonie 1

Figure 63: SNP de A vers G dans la colonie 1