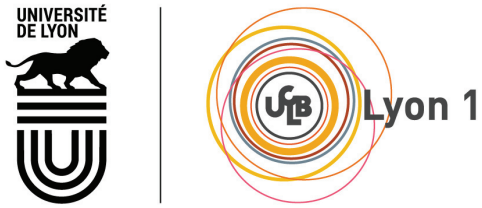
Reconstruction conjointe de l'ordre des gènes de génomes actuels et ancestraux et de leur évolution structurale dans un cadre phylogénétique

Yoann Anselmetti



N° d'ordre NNT : 2017LYSE1242

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON

opérée au sein de

l'Université Claude Bernard Lyon 1

École Doctorale ED341

Évolution, Écosystèmes, Microbiologie, Modélisation (E2M2)

Spécialité de doctorat : Doctorat de Bioinformatique

Discipline : Biologie évolutive

Soutenue publiquement le 29/11/2017, par :

Yoann Louis ANSELMETTI

Reconstruction conjointe de l'ordre des gènes de génomes actuels et ancestraux et de leur évolution structurale dans un cadre phylogénétique

---

Devant le jury composé de :

OUANGRAOUA Aïda, Professeure adjointe, Université de Sherbrooke	Rapporteure
FISCHER Gilles, Directeur de recherche, UPMC	Rapporteur
DESSIMOZ Christophe, Professeur *SNSF*, Université de Lausanne	Examinateur
MOUCHIROUD Dominique, Professeure, Université Lyon 1	Examinatrice
BÉRARD Sèverine, Maître de conférences, Université de Montpellier	Directrice
TANNIER Éric, Chargé de recherche, INRIA Rhône-Alpes	Directeur
CHATEAU Annie, Maître de conférences, Université de Montpellier	Invitée

UNIVERSITE CLAUDE BERNARD - LYON 1

Président de l'Université	**M. le Professeur Frédéric FLEURY**
Président du Conseil Académique	M. le Professeur Hamda BEN HADID
Vice-président du Conseil d'Administration	M. le Professeur Didier REVEL
Vice-président du Conseil Formation et Vie Universitaire	M. le Professeur Philippe CHEVALIER
Vice-président de la Commission Recherche	M. Fabrice VALLÉE
Directrice Générale des Services	Mme Dominique MARCHAND

COMPOSANTES SANTE

Faculté de Médecine Lyon Est – Claude Bernard	Directeur : M. le Professeur G.RODE
Faculté de Médecine et de Maïeutique Lyon Sud – Charles Mérieux	Directeur : Mme la Professeure C. BURILLON
Faculté d'Odontologie	Directeur : M. le Professeur D. BOURGEOIS
Institut des Sciences Pharmaceutiques et Biologiques	Directeur : Mme la Professeure C. VINCIGUERRA
Institut des Sciences et Techniques de la Réadaptation	Directeur : M. X. PERROT
Département de formation et Centre de Recherche en Biologie Humaine	Directeur : Mme la Professeure A-M. SCHOTT

COMPOSANTES ET DEPARTEMENTS DE SCIENCES ET TECHNOLOGIE

Faculté des Sciences et Technologies	Directeur : M. F. DE MARCHI
Département Biologie	Directeur : M. le Professeur F. THEVENARD
Département Chimie Biochimie	Directeur : Mme C. FELIX
Département GEP	Directeur : M. Hassan HAMMOURI
Département Informatique	Directeur : M. le Professeur S. AKKOUCHE
Département Mathématiques	Directeur : M. le Professeur G. TOMANOV
Département Mécanique	Directeur : M. le Professeur H. BEN HADID
Département Physique	Directeur : M. le Professeur J-C PLENET
UFR Sciences et Techniques des Activités Physiques et Sportives	Directeur : M. Y.VANPOULLE
Observatoire des Sciences de l'Univers de Lyon	Directeur : M. B. GUIDERDONI
Polytech Lyon	Directeur : M. le Professeur E.PERRIN
Ecole Supérieure de Chimie Physique Electronique	Directeur : M. G. PIGNAULT
Institut Universitaire de Technologie de Lyon 1	Directeur : M. le Professeur C. VITON
Ecole Supérieure du Professorat et de l'Education	Directeur : M. le Professeur A. MOUGNIOTTE
Institut de Science Financière et d'Assurances	Directeur : M. N. LEBOISNE

Nothing in biology makes sense except in the light of evolution.

Theodosius Dobzhansky (1972)

Toutes les feuilles mortes se marrent entre elles.

Extrait de "Si rien ne bouge" de Noir Désir (1991)

Résumé

L'émergence des technologies de séquençage haut-débit a permis, au cours des années 2000, l'augmentation d'exponentielle du nombre de génomes pour lesquels la séquence d'ADN complète est disponible. L'accès à cette multitude de génomes a ouvert la voie à l'étude de l'évolution de la structure linéaire des génomes, sous la forme d'un ordre de marqueurs génétiques. La nécessité pour les technologies de séquençage haut-débit de fragmenter les génomes avant de les séquencer a nécessité le développement d'approches algorithmiques pour l'assemblage de génomes qui consiste à reconstituer l'enchaînement complet des nucléotides le long des chromosomes. En plus de la complexité algorithmique de ce problème, la présence de séquences répétées au sein des génomes accentue la difficulté d'assemblage des génomes aboutissant à une reconstitution incomplète de la structure linéaire. Cette fragmentation des génomes implique une reconstruction incomplète de l'évolution structurale des génomes.

Dans ce contexte, nous avons développé un outil algorithmique qui permet de conjointement reconstruire l'ordre de gènes d'espèces ancestrales et d'améliorer la reconstitution de l'ordre des gènes chez les espèces actuelles de la phylogénie considérée. La méthode permet de considérer les duplications, pertes et transferts de gènes dans l'inférence de l'évolution de l'ordre des gènes.

Mots-clés : bioinformatique, reconstruction de génomes ancestraux, *scaffolding*, évolution de l'ordre des gènes, adjacences de gènes, phylogénie, *Anopheles*

Abstract

The emergence of high throughput sequencing technologies allowed, in the 2000s, an exponential increase of the number of genomes for which the complete DNA sequence is available. Access to this multitude of genomes has opened the way for studying the evolution of the linear structure of genomes, in the form of an order of genetic markers. The need for high-throughput sequencing technologies to fragment genomes before sequencing has required the development of algorithmic approaches to genome assembly that consists of reconstructing the complete sequence of nucleotides along chromosomes. In addition to the algorithmic complexity of this problem, the presence of repeated sequences within genomes accentuates the difficulty of assembling genomes resulting in an incomplete reconstruction of the linear structure. This fragmentation of the genomes implies an incomplete reconstruction of the structural evolution of the genomes.

In this context, we have developed an algorithmic tool that allows to jointly reconstruct the order of genes of ancestral species and to improve the reconstruction of gene order in extant species of the considered phylogeny. The method allows to consider the duplications, losses and transfers of genes in the inference of the evolution of the gene order.

Keywords : bioinformatics, ancestral genome reconstruction, scaffolding, gene order evolution, gene adjacencies, phylogeny, *Anopheles*

Remerciements

Au cours de ces 3 années de thèse, j'ai eu la grande chance de pouvoir effectuer mes travaux dans un environnement de recherche épanouissant dans lequel j'ai rencontré un grand nombre de personnes qui ont été impliquées de près ou de loin dans mes travaux thèse par leur aide, leur soutien et leur passion pour la science.

Je voudrais tout d'abord remercier mes deux codirecteurs de thèse, Sèverine Bérard et Éric Tannier, qui m'ont permis au cours de cette thèse d'avoir une grande autonomie tout en ayant été toujours présent lors des difficultés rencontrées. J'ai donc pu au cours de ces trois années de thèse me construire en tant que chercheur dans les meilleures conditions possibles. J'ai été le premier "thésard" de Sèverine et je tiens à la remercier très chaleureusement pour sa bienveillance, sa disponibilité (malgré les heures d'enseignement) et son soutien. Travailler avec toi a été un réel plaisir, merci encore et pardon pour les longues journées et le stress que je t'ai causé à la fin de ma thèse. Merci à Éric pour sa supervision depuis Lyon, pour ses conseils avisés lors de la rédaction des articles et la préparation des présentations orales et également de m'avoir permis de combler mes lacunes dans la connaissance du monde de la recherche scientifique.

J'aimerai aussi remercier Vincent Berry qui m'a permis de rencontrer Sèverine dans le cadre de mon stage de Master 2 et pour sa participation à l'encadrement de ce stage qui a abouti à une version préliminaire de l'algorithme ARt-DeCo développé au cours de la thèse. Merci également à Annie Chateau qui a également participé à l'encadrement de mon stage de Master 2 et été impliquée dans le suivi de mes travaux de thèse notamment dans le cadre de mon suivi de thèse. Je suis très heureux que tu es pu participer à ma soutenance de thèse. J'espère dans le futur avoir d'autres occasions de collaborer avec vous dans les années à venir.

Merci à Cédric Chauve avec qui j'ai eu l'occasion d'effectuer une partie de mes travaux de thèse dans son laboratoire à l'Université Simon Fraser à Vancouver, ayant abouti au développement de l'algorithme ADseq et à la participation d'une collaboration avec les membres du *Anopheles Genome Cluster Consortium*. Merci pour tes précieux conseils de bonnes pratiques de codage et de reproductibilité des résultats qui m'ont beaucoup servi au cours de la thèse et me serviront dans mes recherches futures. Dans le cadre de cette collaboration, je tiens à remercier le programme de bourse de recherche de

Mitacs Globalink - Campus France et le NSERC qui ont permis de financer mon voyage au Canada.

Je remercie chaleureusement tous les membres de mon jury de thèse qui ont accepté de participer à l'évaluation des travaux effectués au cours de mes trois années de thèse. Merci à mes deux rapporteurs Aïda Ouangraoua et Gilles Fischer d'avoir bien voulu prendre le temps de lire mon manuscrit de thèse. Merci à tous les deux et à Christophe Dessimoz et Dominique Mouchiroud pour la discussion enrichissante au cours des questions de la soutenance de thèse.

J'aimerai aussi remercier Raluca Uricaru, Thomas Faraut et Sam Meyer, les membres de mon comité de suivi de thèse, qui ont bien voulu participer à celui-ci et qui m'ont permis au cours de la thèse d'effectuer des points d'étapes et de débattre avec des chercheurs extérieurs au projet et de faire un point sur l'avancée de mes travaux de recherche.

Cette thèse ne serait également pas ce qu'elle est sans l'équipe de recherche dans laquelle j'ai évolué et je suis très heureux d'avoir pu l'effectuer au sein de l'équipe "Phylogénie et Évolution Moléculaire" de l'ISEM. Merci aux co-bureaux avec qui j'ai passé de très bons moments. À Marjo pour ses petits sauts d'humeurs qui mettent de l'ambiance positive dans le bureau 🙂. À Clémentine pour son espièglerie et sa mauvaise foi légendaire 🤪. À Andrea pour tout l'énergie positive méditerranéenne qu'elle apporte dans le bureau 😁. À Paul pour ses intrusions dans notre bureau à l'heure du goûter pour déguster une tartine et débattre des sujets d'actualités 😉. À tous les doctorants et post-docs de l'ISEM avec qui j'ai partagé de merveilleux moments aussi bien au laboratoire qu'en dehors. Merci à Myriam, Sam, Quentin, Émeric, Yoann, Maëva, Alain, Manon, Sergio, Alexis, Maud, Maxime, Quentin, Mine (et j'en oublie bien d'autres sûrement). Un remerciement tout particulier à nos délégués doctorants Yoann et Maëva qui ont magnifiquement joué leur rôle et ont mis en place les apéros doctorants et le week-end d'intégration qui ont permis à l'ensemble des doctorants et post-doctorants de l'ISEM de se rencontrer et de se soutenir les uns les autres.

Merci également à tous les membres de l'équipe "Bioinformatique, Phylogénie et Génomique Évolutive" du LBBE mon deuxième laboratoire que j'ai eu l'occasion de côtoyer lors des meetings de l'ANR Ancestrome, au cours de mes séjours sur Lyon et à l'occasion de JOBIM 2016. Merci pour vos précieux conseils et votre accueil au sein du laboratoire. Un remerciement tout particulier à Wandrille, Magali et Priscilla avec qui nous étions co-supervisés par Éric et avec qui j'ai participé à de nombreuses conférences. Merci pour

ces bons moments passés avec vous.

Un énorme merci à mes amis hors de Montpellier que j'ai toujours eu à cœur de revoir lors de mes escapades en région Rhône-Alpes, en Bourgogne et à Toulouse. Merci à Achraf, Baba, Nicolas pour leur compagnie lors de nos retrouvailles à Lyon, à nos nuits dans les bars et à nos discussions de tout et de rien. Merci à Chachou, Nico et à leurs 2 petits loups Nolan et Maëlys avec qui l'on passe toujours de bons moments en Haute-Savoie surtout autour d'un bon plateau de jeux. Merci à Alex et Flora, nous avons peu l'occasion de nous revoir ces dernières années mais c'est toujours avec plaisir. Merci à tous les deux ce merveilleux nouvel an à Bazoches et à la prochaine dégustation de bière. Merci à Manon et Matthieu pour les bons moments passés à Toulouse avec les membres de la cocholoc' et leur brin de folie.

Merci enfin à toute ma famille de m'avoir permis de faire mes études dans les meilleurs conditions possibles. Les week-end/vacances à Grenoble, à Machilly, en Suisse et dans le Jura ont été de grandes bouffées d'oxygène dans ce long périple qu'est la thèse.

Merci à Aude d'avoir toujours été présente à mes côtés.

Table des matières

Table des figures

Liste des tableaux

Introduction

Les premiers travaux connus émettant l'hypothèse d'une parenté des espèces sous la forme d'un arbre, nommé *arbre phylogénétique*, remontent au 19e siècle. La figure 1 représente le célèbre croquis de Charles Darwin considéré comme l'un des tous premiers schémas d'arbre évolutif. L'*arbre phylogénétique du vivant* sera par la suite popularisé à travers l'œuvre majeure de Charles Darwin, *De l'Origine des Espèces* [1] (1859), par l'unique illustration de ce livre représentant un schéma de l'arbre du vivant basé sur le principe de descendance des espèces par modification (*cf.* figure 2). Dans *De l'Origine des Es-*

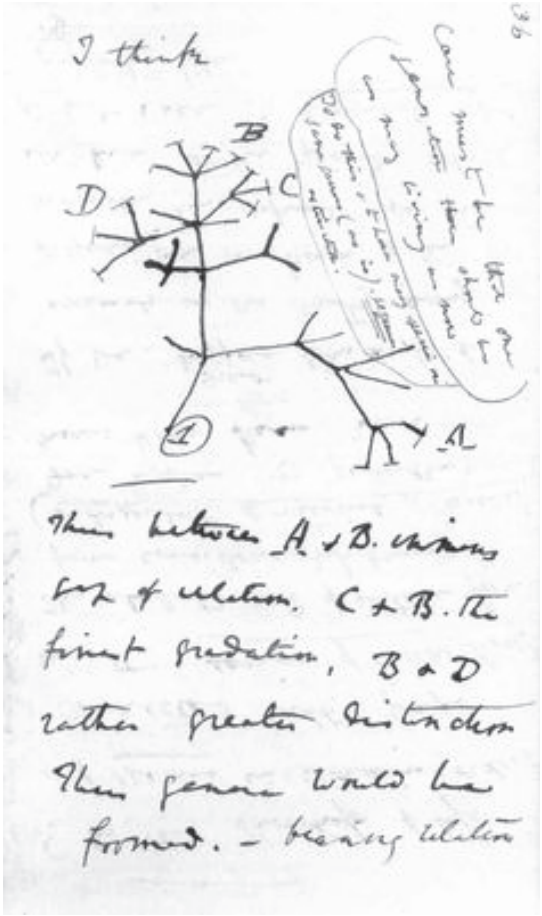


FIGURE 1 – Premier croquis d'un arbre évolutif de Charles Darwin (tiré de *First Notebook on Transmutation of Species*, 1837).

1. titre original : *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*.

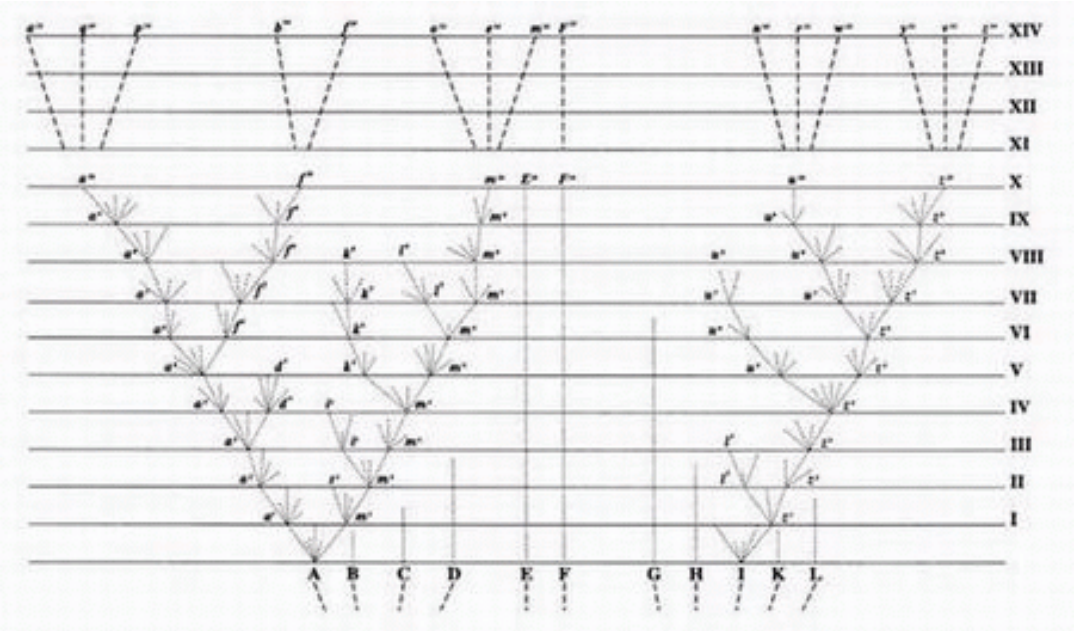


FIGURE 2 – Schéma de l'arbre phylogénétique du vivant présent dans *De l'Origine des Espèces* de Charles Darwin (1859).

pèces, Charles Darwin décrit le mécanisme de sélection naturelle expliquant l'évolution et l'apparition d'espèces au cours du temps. La co-découverte de ce mécanisme est souvent attribuée à Alfred Russell Wallace et Charles Darwin. Il est cependant à noter qu'une première description succincte de ce mécanisme a été faite par Patrick Matthew dans son livre *On Naval Timber and Arboriculture* publié en 1831 soit 28 ans avant la publication de *De l'Origine des Espèces* !

Par la suite, Ernst Haeckel s'est inspiré des travaux publiés dans *De l'Origine des Espèces* pour établir les premières phylogénies d'espèces [Dayrat, 2003] par une approche d'anatomie comparée. Cette approche consiste à établir les relations de parenté entre les espèces par l'analyse des similarités et dissimilarités anatomiques entre les espèces. Il est d'ailleurs l'inventeur du terme *phylogénie* qui correspond à l'étude des relations de parenté entre les espèces, souvent représentées sous la forme d'un arbre. En 1866, Ernst Haeckel établit une phylogénie regroupant l'ensemble des espèces connues à cette époque (*cf.* figure 3).

Au cours du 20^{e} siècle, la redécouverte des lois de l'hérédité, ou *lois de Mendel*, par Carl Correns, Erich von Tschermak-Seysenegg et Hugo De Vries en 1900 [2] et la découverte de l'ADN comme support de l'hérédité par [Avery et al., 1944], confirmée par les expériences de Hershey and Chase [1952], vont permettre le développement d'une nouvelle discipline : la *phylogénie*

2. initialement découverte par Mendel [1865].

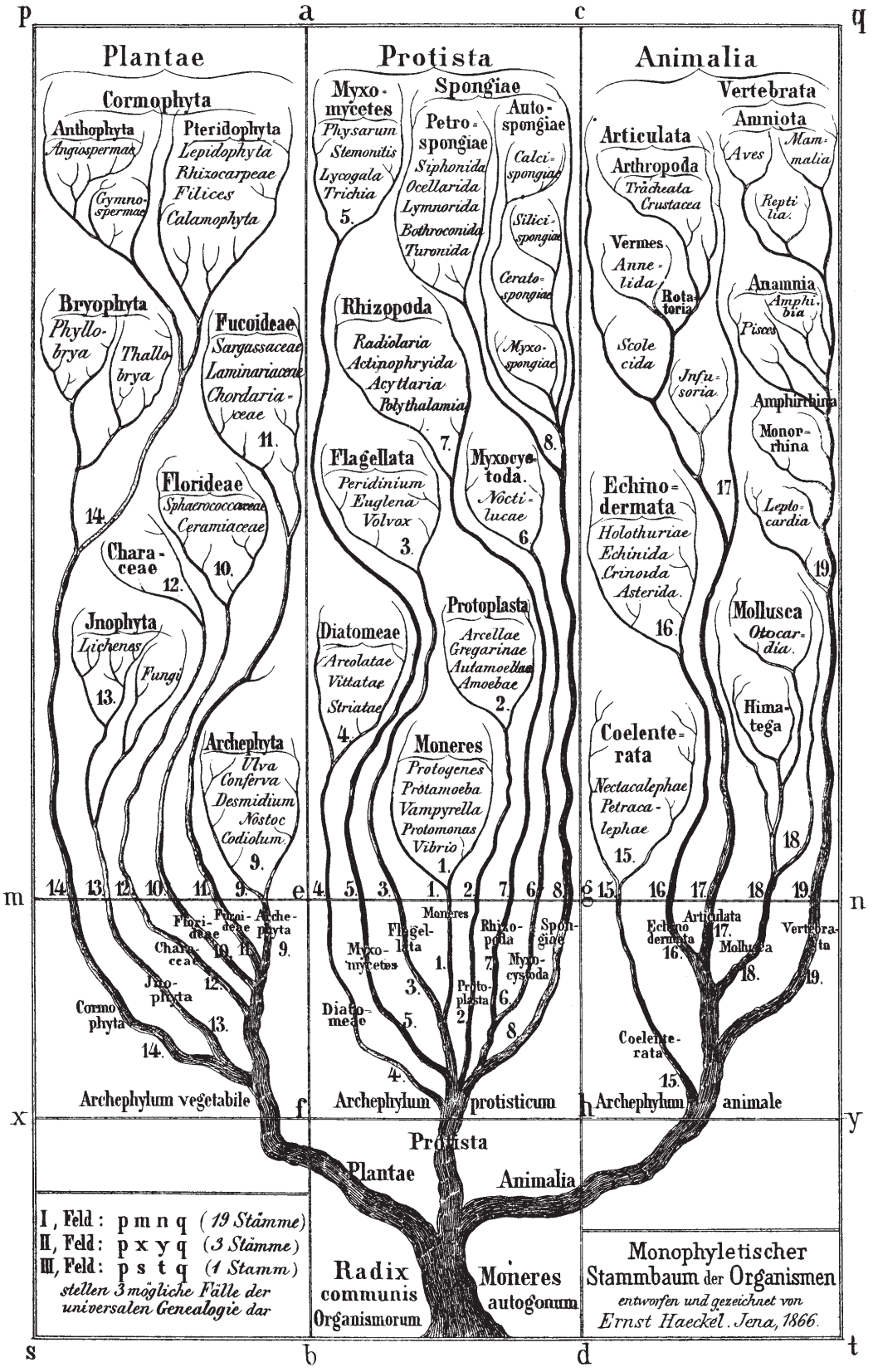


FIGURE 3 – Phylogénie du vivant d'Ernst Haeckel (tiré de *Generelle Morphologie der Organismen*, 1866).

moléculaire. La phylogénie moléculaire consiste à établir les relations de parenté entre les espèces non plus en comparant leurs caractéristiques morpho-anatomiques mais leurs caractéristiques moléculaires et plus particulièrement leurs séquences nucléotidiques (ARN [3] ou ADN [4]). La molécule d'ADN a été découverte par Friedrich Miescher en 1869 par l'identification d'une substance phosphorée dans le noyau de cellules qu'il avait initialement nommée *nucléine* [Dahm, 2005, 2008].

Les années 1950 voient l'apparition de méthodes permetttant de déterminer la séquence en acides aminés de molécules protéiques. Dans les années 1960, Emile Zuckerckandl et Linus Pauling se servent de séquences protéiques de molécules d'hémoglobines de différentes espèces afin d'établir l'une des premières phylogénies basées sur des caractéristiques moléculaires [Zuckerkandl and Pauling, 1965] par alignement de ces séquences et énumération des similarités/dissimilarités entre elles. Les années 1960 voient également l'apparition des premières méthodes de séquençage de molécules d'ARN. Il a, par la suite, fallu attendre une dizaine d'années pour obtenir les premières phylogénies de l'arbre du vivant, dont celle inférée par Woese and Fox [1977] basée sur la comparaison de séquences d'ARN ribosomiques 16S et 18S. Ces séquences ont été utilisées car elles sont très conservées dans l'ensemble du vivant et permettent ainsi de déterminer la topologie globale de l'arbre phylogénétique du vivant jusqu'au niveau de l'hypothétique ancêtre commun universel du vivant (couramment appelé *LUCA* [5]). Avant cette publication, l'arbre du vivant était divisé en deux domaines : les *eucaryotes* [6] et les *procaryotes* [7]. La publication de Woese and Fox [1977] met en évidence l'existence d'un troisième domaine qui scinde le domaine des procaryotes en deux : le domaine des *archéobactéries* [8] et le domaine des *bactéries*. La partition de l'arbre du vivant en trois grands domaines est confirmée par de nouvelles expériences de Woese et al. [1990]. Cette représentation du vivant en trois domaines est aujourd'hui reconnue par l'ensemble de la communauté scientifique (*cf.* figure 4).

Pour inférer l'arbre du vivant, les premières phylogénies ne considéraient qu'un faible nombre de marqueurs génomiques. Cette limitation provenait du fait que les méthodes de séquençage des années 1980 et 1990 étaient trop

3. Acide RiboNucléique.
4. Acide DésoxyriboNucléique.
5. *Last Universal Common Ancestror*.
6. regroupant les espèces dont les cellules sont composées d'un noyau.
7. regroupant les espèces dont les cellules n'ont pas de noyau.
8. plus tard renommé *archées*.

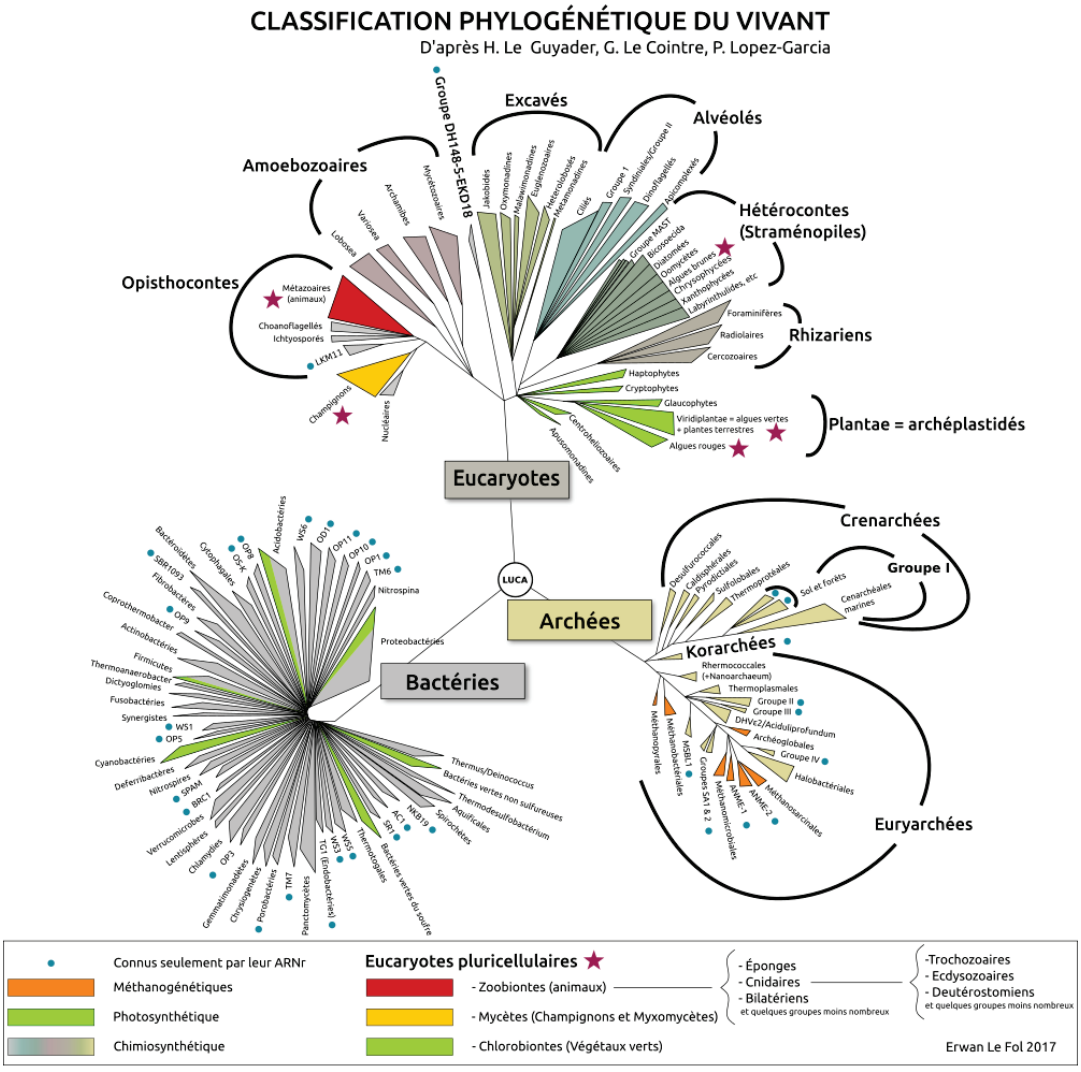


FIGURE 4 – Représentation de l'arbre du vivant actuellement reconnu par la communauté scientifique. La racine de cet arbre correspond au dernier ancêtre universel commun (*LUCA*) et est partitionné en trois domaines du vivant : les eucaryotes, les archées et les bactéries (figuré tirée de `http://www.svt-monde.org/IMG/png/arbre-Phylogenetique.png`).

coûteuses en temps et en argent pour obtenir la séquence d'ADN complète d'un large éventail d'espèces [Carlson, 2003]. Au cours des années 2000, le développement de technologies et méthodologies de séquençage haut-débit a permis l'obtention de la séquence nucléotidique complète de très nombreux génomes. Cette explosion du nombre de génomes disponibles dans les bases de données a ouvert la perspective de pouvoir effectuer la reconstruction de l'histoire évolutive des espèces non plus sur quelques marqueurs génomiques mais à l'échelle de la séquence entière des génomes en se basant sur des marqueurs orthologues [9] le long des génomes.

Cet accès à un nombre croissant de génomes entiers a également permis de reconstruire l'histoire évolutive, non plus à partir du contenu nucléotidique de la séquence des génomes, mais de la structure linéaire des génomes, au sens de l'ordre linéaire de marqueurs le long d'un génome. Des méthodes permettant de reconstruire l'histoire évolutive de cette organisation des génomes ont été développées en amont et en parallèle du développement du séquençage haut-débit. Les premières méthodes développées étaient basées sur l'analyse des points de cassures génomiques provoqués lors de réarrangements chromosomiques des génomes. Une anecdote intéressante est que la première phylogénie d'espèces établie sur des critères génétiques/génomiques a été inférée par Sturtevant and Dobzhansky [1936]; Dobzhansky and Sturtevant [1938] à partir de l'analyse d'un type de réarrangements chromosomiques, les inversions chromosomiques, chez des mouches du genre *Drosophila*. Ces inversions ont été découvertes par Sturtevant [1921, 1926] à partir de l'analyse du caryotype de ces espèces.

Cependant, les méthodes de reconstruction de l'histoire évolutive de l'organisation des génomes sont, à l'heure actuelle, peu exploitées car la complexité algorithmique de la plupart de ces méthodes permet leur application uniquement sur des jeux de données de petites tailles (de l'ordre de dix espèces maximum). De plus, elles nécessitent de connaître l'ordre et l'orientation complets des marqueurs génomiques le long du génome des espèces actuelles considérées. Si les génomes des espèces actuelles sont fragmentés,

9. ensemble de marqueurs ayant une origine ancestrale commune et dont la diversification provient d'événements de spéciation (transformation d'une espèce ancestrale en deux espèces filles).

alors on a des informations de synténies manquantes entre les marqueurs génomiques rendant difficile, voire impossible, la résolution complète de l'histoire évolutive de l'organisation de ces marqueurs. Or, actuellement la majorité des génomes séquencés ne sont pas dans un format adéquat pour étudier l'évolution de leur structure. En effet, les technologies de séquençage haut-débit nécessitent le découpage des génomes en petites portions génomiques. La séquence nucléotidique du génome séquencé étant présente en plusieurs copies, il est possible d'aligner ces petits fragments d'ADN obtenus les uns avec les autres sur leurs séquences communes afin d'obtenir des fragments génomiques de taille plus grande. Cette approche d'assemblage des génomes est néanmoins insuffisante car pour la majorité des projets de séquençage entrepris de grands génomes d'eucaryotes ceux-ci demeurent incomplètement assemblés (c.-à-d. que les génomes sont sous la forme de sections génomiques plus ou moins longues, appelées *contigs*, dont l'ordre et l'orientation relative dans les chromosomes ne sont pas connus).

Pour pallier ce problème de résolution incomplète de la structure de la séquence d'ADN des génomes séquencés, des approches de *génomique comparative* ont été développées, permettant de se servir de la séquence d'ADN complètement assemblée de génomes proches comme référence pour guider l'assemblage.

Dans ce contexte, mon projet de thèse a eu pour objectif de développer une méthode qui permette de reconstruire l'évolution de l'ordre de marqueurs génomiques le long d'une phylogénie des espèces. Au début du projet, nous avons fait le constat qu'un nombre important de grands génomes d'espèces eucaryotes disponibles dans les bases de données étaient incomplètement assemblés. Nous avons adapté notre stratégie pour pouvoir considérer cette fragmentation des génomes dans notre approche de reconstruction de l'histoire évolutive de leur organisation. En effectuant cette démarche, nous avons établi que cette reconstruction le long de la phylogénie des espèces considérées permettaient de transmettre des signaux de synténies de marqueurs génomiques d'une espèce à une autre, pondérés par le degré de parenté entre ces espèces. L'approche que nous avons développée au cours de cette thèse permet de conjointement reconstruire l'histoire évolutive de l'ordre de marqueurs génomiques le long de la phylogénie des espèces considérées et améliorer l'assemblage de génomes actuels incomplètement assemblés par génomique comparative.

Ce manuscrit est découpé en deux parties, une première partie présente les différentes notions de phylogénétique moléculaire et génomique nécessaire à la compréhension de nos travaux et effectue un état de l'art sur les méthodes pour le séquençage et l'assemblage de génomes ainsi que les méthodes pour la reconstruction de l'évolution de l'ordre de marqueurs génomiques le long d'une phylogénie d'espèces. La deuxième partie de ce manuscrit présente les travaux effectués au cours de cette thèse dont les trois premiers chapitres détaillent les méthodes, algorithmes et logiciels développées, et les deux derniers chapitres présentent l'application de ces méthodes sur un jeu de données de 18 espèces de moustiques. Enfin, un dernier chapitre est consacré à la conclusion et aux perspectives de ce projet de thèse.

Première partie

État de l'art

La première partie de ce manuscrit de thèse a pour but de donner aux lecteurs les notions de base nécessaires à sa compréhension. Elle présente également l'état de l'art des approches algorithmiques existantes pour l'assemblage de génomes et celles pour la reconstruction de l'évolution de l'ordre et de l'orientation de marqueurs génomiques le long d'une phylogénie d'espèces. Le projet de thèse ayant eu pour but de développer une approche qui reconstruise conjointement l'ordre de marqueurs génomiques d'espèces actuelles et ancestrales d'une même phylogénie d'espèces, nos travaux font appel à des notions de génomique, d'évolution, de phylogénétique moléculaire, de séquençage et d'assemblage de génomes. Il est donc nécessaire d'introduire ces notions et le contexte scientifique de ces différents domaines à l'heure actuelle pour comprendre l'originalité de l'approche que nous avons développée au cours de cette thèse.

La première partie de ce manuscrit est divisée en trois chapitres :

1. dans le chapitre 1, nous présentons les notions de génomique et phylogénétique moléculaire nécessaires à la compréhension de ce manuscrit ;
2. le chapitre 2 détaille différentes méthodes développées par la communauté scientifique afin de reconstituer la structure complète de la séquence nucléotidique des génomes actuels. Nous y présentons en particulier un ensemble de méthodes consistant à résoudre le problème d'assemblage de génomes par génomique comparative. Puis, nous comparons les propriétés de ces méthodes à l'approche que nous avons développée au cours de cette thèse pour ce problème ;
3. Le chapitre 3 présente un ensemble représentatif des différentes approches méthodologiques et algorithmiques permettant de reconstruire l'histoire évolutive de l'ordre et de l'orientation de marqueurs génomiques le long d'une phylogénie d'espèces. Ces méthodes permettent de déterminer l'organisation relative de ces marqueurs le long du génome d'espèces ancestrales disparues. Dans ce même chapitre, nous discutons de la similitude entre l'inférence de l'ordre et de l'orientation de fragments génomiques d'un génome actuel et la reconstruction de l'histoire évolutive de l'ordre et de l'orientation de marqueurs génomiques d'espèces ancestrales. Enfin, nous exploitons la possibilité d'utiliser la phylogénie des espèces pour résoudre conjointement ces deux problèmes. L'approche développée au cours de cette thèse est présentée dans la partie II de ce manuscrit.

Chapitre 1

La phylogénétique moléculaire à l'échelle des génomes

1.1 Notions de génomique

1.1.1 Génome, chromosomes et gènes

La modification du matériel génétique joue un rôle important dans l'évolution des espèces, il est donc primordial de comprendre les éléments structurels de l'information génétique.

Le *génome* d'un organisme correspond à la séquence nucléotidique complète de celui-ci. Dans la majorité des cas, cette séquence nucléotidique est une molécule d'ADN[1] composés de deux brins d'ADN anti-parallèles correspondant chacun à une chaîne de *nucléotides*. Un nucléotide est composé d'un groupement phosphate lié à un désoxyribose lui-même lié à un nucléoside, ou nucléobase. Pour l'ADN, il y a quatre nucléosides différents regroupés dans deux ensembles de molécules :

— les purines : l'Adénine (A) et la Guanine (G) ;
— les pyrimidines : la Cytosine (C) et la Thymine (T).

Les deux chaînes d'un ADN double brin sont associées par la complémentarité des bases : $A - T$ et $C - G$.

La liaison des groupements phosphates aux désoxyriboses permet de donner une orientation à un brin d'ADN suivant la position du carbone auquel un groupement phosphate se lie à un désoxyribose (*cf.* figure 1.1) donnant une orientation $5' \rightarrow 3'$ pour le brin d'ADN codant/non-transcrit et $3' \rightarrow 5'$ pour le brin d'ADN non-codant/transcrit. La figure 1.2 illustre la structure et le contenu d'un ADN double brin illustrant la description de ce paragraphe.

1. Acide DésoxyriboNucléique.

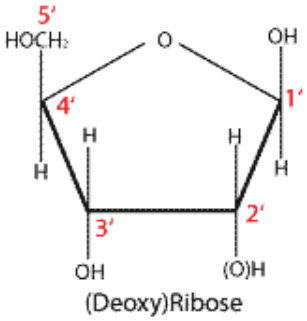


FIGURE 1.1 – Schéma représentant la numérotation des carbones d'une molécule de (désoxy)ribose (figure tirée de https://commons.wikimedia.org/wiki/File:DeoxyriboseLabeled.png).

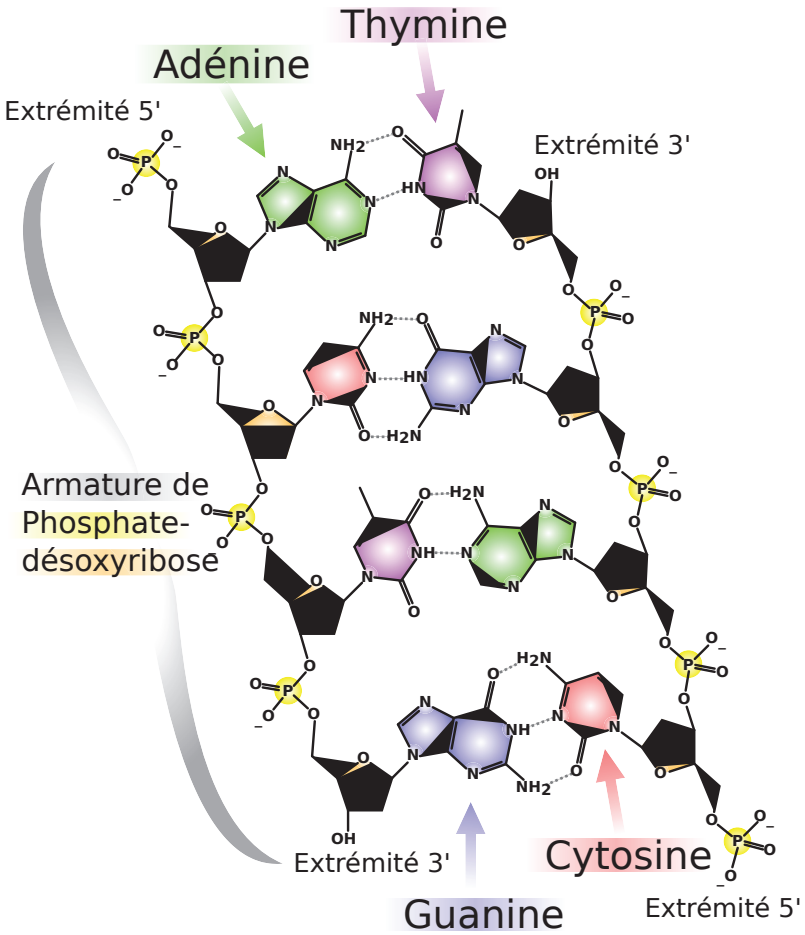


FIGURE 1.2 – Schéma représentant une section d'ADN double brin (figure tirée de https://commons.wikimedia.org/wiki/File:DNA_chemical_structure-1-.fr.svg).

La séquence d'ADN d'un génome est répartie sur un ensemble de *chromosomes* linéaires ou circulaires dont le nombre dépend de l'espèce à laquelle appartient l'organisme (bien que chez certaines espèces le nombre de chromosomes peut varier entre deux individus Lukhtanov et al. [2011]). Chaque chromosome est constitué d'une unique molécule d'ADN sur laquelle sont positionnés des *gènes*. Un gène est un segment chromosomique, localisé par une position et une orientation unique sur le chromosome, correspondant à une séquence d'ADN qui peut être transcrit en molécule fonctionnelle appelée ARN[2] (à l'exception des virus à ARN dont les gènes sont déjà présents sous la forme de séquences d'ARN).

L'orientation de la séquence d'ADN est utilisée pour orienter les gènes sur les chromosomes. Un gène localisé sur le brin $5' \rightarrow 3'$ a une orientation "+" tandis qu'un gène sur le brin $3' \rightarrow 5'$ a une orientation "-".

1.1.2 L'adjacence de gènes

Le projet de thèse s'intéressant à l'évolution de l'ordre des gènes le long d'une phylogénie d'espèces. Le type de synténies que nous utilisons pour reconstruire l'ordre des gènes est l'*adjacence de gènes*.

Étant donnés n gènes échantillonnés dans un génome, ordonnés linéairement le long des séquences[3] de ce génome, deux gènes g_1 et g_2 sont adjacents, noté $g_1 \sim g_2$ ou $g_2 \sim g_1$, s'ils sont situés sur la même séquence et s'il n'y a pas d'autres gènes entre g_1 et g_2 (*cf.* figure 1.3).

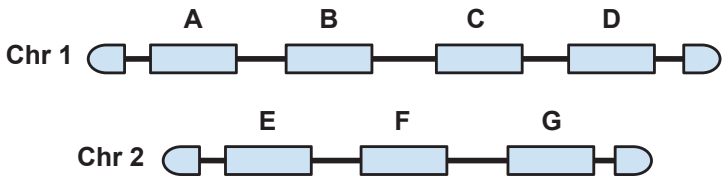


FIGURE 1.3 – Schéma représentant des adjacences de gènes sur des chromosomes. On a $B \sim A$ et $B \sim C$. A a une adjacence (avec B), tandis que B en a deux (avec A et C).

Ainsi, un gène a au plus deux adjacences sur les séquences (format linéaire ou circulaire). Au vu de la quantité de gènes contenus dans un génome composé de chromosomes linéaires, la proportion de gènes ayant une seule adjacence est minime (*cf.* exemple 1).

2. Acide RiboNucléique.
3. correspondant aux chromosomes si le génome est complètement assemblé.

Exemple 1

Chez *Homo sapiens*, le génome est composé d'approximativement 20.000 gènes répartis sur 22 paires de chromosomes homologues linéaires, deux chromosomes sexuels linéaires, X et Y, et le génome mitochondrial circulaire. Il y a donc $24 * 2 = 48$ gènes n'ayant qu'une seule adjacence ce qui représente $0{,}24\ \%$ du nombre de gènes total (en considérant que l'emplacement de l'ensemble des gènes s'organise sous la forme de structures linéaires ou circulaires).

Il est nécessaire de préciser que dans les données réelles la conformation de l'ordre des gènes n'est pas complètement linéaire. En effet, la présence de trois cadres de lecture de gènes sur la séquence d'un brin d'ADN et la structure de l'ADN en double brin impliquent des chevauchements de séquences entre les gènes. La séquence de certains gènes est parfois même entièrement incluse entre les extrémités de début et de fin d'autres gènes. Un filtrage des données génomiques est donc nécessaire avant d'obtenir une conformation entièrement linéaire des gènes le long des génomes. Nous décrivons le protocole de filtre que nous avons utilisé pour traiter les gènes inclus dans d'autres dans la section 5.2.1 (p. 102).

1.2 Les modifications génétiques

Pour déterminer l'histoire évolutive des espèces, il a d'abord fallu comprendre les mécanismes moléculaires permettant le processus de l'évolution moléculaire. Les travaux sur les pneumocoques par Avery et al. [1944] ont permis d'établir que l'ADN est le support de l'hérédité décrite par Mendel [1865]. L'évolution des espèces est en grande partie le résultat de modifications de l'ADN. Nous les avons classées dans trois catégories représentant différentes échelles de modification du génome, celles-ci pouvant être simultanées :

— les **mutations ponctuelles** qui affectent la séquence d'ADN à une petite échelle en modifiant le contenu nucléotide par nucléotide ;
— les **modifications du contenu en gènes** qui modifient la séquence nucléotidique à une échelle intermédiaire en modifiant la quantité et la diversité génique d'un génome ;

— les **réarrangements chromosomiques** qui altèrent la structure des chromosomes.

1.2.1 Les mutations ponctuelles

Ces mutations affectent les **nucléotides** et son majoritairement produites par des erreurs de la machinerie moléculaire, par exemple lors de la réplication ou de la réparation de la séquence d'ADN. Les mutations ponctuelles sont communément classées dans trois catégories :

— la **substitution** qui remplace un nucléotide par un autre ;
— l'**insertion** qui ajoute un nucléotide dans la séquence d'ADN ;
— la **délétion** qui enlève un nucléotide de la séquence d'ADN.

Il est à noter que pour les insertions et délétions, des successions de nucléotides plus ou moins longues peuvent être concernées sans modifier le contenu en gènes.

Ces mutations ponctuelles modifient la séquence nucléotidique et lorsque celles-ci offrent un avantage dans la survie ou l'adaptation de l'espèce dans son environnement, elles sont conservées par le processus de sélection naturelle. Cependant, ces événements ne suffisent pas pour expliquer l'histoire évolutive des gènomes. D'autres mécanismes biologiques sont responsables de l'évolution des génomes.

1.2.2 Les modifications du contenu en gènes

Parmi l'ensemble des modifications pouvant modifier le contenu en gènes d'un génome, nous en considérons trois qui sont utilisées dans nos modèles :

— la **duplication de gène** qui est un processus évolutif qui duplique et transpose un gène sur le génome, résultant souvent en la production d'une nouvelle fonctionnalité génique, néofonctionnalisation [4] (*cf.* référence [Magadum et al., 2013] pour une revue sur la duplication de gène) ;
— la **perte de gène** qui conduit à la fin de l'histoire évolutive d'un gène. Ce phénomène peut être causé par la pseudogénisation du gène qui implique une accumulation de mutations ponctuelles sur sa séquence

4. génération d'un gène codant pour une nouvelle protéine avec une fonction nouvelle.

ou celle de sa zone régulatrice entraînant l'inactivation ou l'inutilisation du gène. La perte d'un gène peut aussi provenir de modifications de plus grande ampleur telles que les réarrangements chromosomiques qui peuvent entraîner la coupure ou la suppression de grands segments de la séquence du gène le rendant inutilisable (*cf.* référence [Albalat and Cañestro, 2016] pour une revue sur le rôle de la perte de gène dans l'évolution des espèces) ;

— le **transfert de gène** est le mécanisme par lequel un gène est transféré horizontalement d'une espèce à une autre contemporaine et dans une même zone géographique. Ce mécanisme est très utilisé chez les bactéries dans le cadre d'échange de plasmides (petits chromosomes circulaires) souvent illustré par l'acquisition de gènes de résistance aux antibiotiques présents sur ces plasmides et apportant un avantage sélectif [Robicsek et al., 2006]. Des transferts horizontaux par *introgression* ont également été décrits chez des espèces pluricellulaires [Harrison and Larson, 2014; Fontaine et al., 2015; Li et al., 2016].

Définition 1 (Introgression)

L'introgression est un transfert horizontal de matériel génétique d'une espèce *donneuse* A au pool génétique d'une autre espèce *receveuse* B qui soient suffisamment proches, génétiquement, pour donner naissance à des hybrides fertiles. Le transfert de gènes s'implante dans l'espèce B par croisements successifs majoritaires entre des individus de l'espèce B et des hybrides d'espèces A et B.

1.2.3 Les réarrangements chromosomiques

Ce que nous considérons comme réarrangements chromosomiques, dans ce manuscrit, ce sont des modifications du génome résultant d'une combinaison de créations et cassures d'adjacences entre des marqueurs le long des chromosomes. Ces réarrangements s'effectuent sur de larges portions chromosomiques contenant souvent plusieurs gènes dont voici certains exemples :

— l'**inversion**, qui implique un chromosome et correspond à une inversion d'une partie génomique de ce chromosome ;

— la **transposition**, qui implique un chromosome et correspond au déplacement d'une section chromosomique à une autre position dans le génome ;

— la **translocation**, qui est un réarrangement entre deux chromosomes homologues qui entraîne un échange de portion chromosomique entre ces deux chromosomes ;

— la **fusion**, qui entraîne la jonction de deux chromosomes par leurs extrémités pour n'en former plus qu'un ;

— la **fission**, qui correspond au clivage d'un chromosome donnant naissance à deux chromosomes.

Les points du génome où le génome se fracture lors de réarrangements chromosomiques sont appelés *points de cassure*. Certaines méthodes de reconstruction de l'histoire évolutive de l'ordre de marqueurs génomiques le long d'une phylogénie sont basées sur la détection ces points de cassure pour retrouver l'enchaînement des réarrangements au cours de l'évolution des espèces de la phylogénie considérée.

1.3 Les arbres phylogénétiques

Pour représenter les liens de parentés entre des espèces ou des gènes, la représentation la plus courante est celle d'un arbre, appelé *arbre phylogénétique*.

1.3.1 Modélisation mathématique

Un arbre phylogénétique est un graphe raciné, non cyclique, connexe, considéré ici comme orienté de la racine aux feuilles. La *racine* correspond à l'ancêtre commun le plus récent de toutes les *feuilles* de l'arbre qui elles, correspondent aux *espèces/gènes/adjacences* présents. Un *nœud* interne dans un arbre phylogénétique correspond à l'ancêtre commun le plus récent des feuilles descendantes de ce nœud. Suivant les contextes, un nœud peut également être étiqueté avec un *événement évolutif*. Lorsque l'on considère un ensemble d'arbres de même type, on parle de *forêt d'arbres*.

1.3.2 Inférence d'un arbre phylogénétique

Pour inférer un arbre phylogénétique, on se base sur l'alignement de séquences nucléotidiques ou protéiques des espèces étudiées. On utilise les ressemblances et dissemblances locales entre ces séquences, parfois résumées dans un score de similarité global. Plus les séquences sont similaires, plus

leur ancêtre commun est supposé récent par rapport aux autres séquences étudiées.

Les méthodes les plus utilisées pour l'inférence d'arbres phylogénétiques sont basées sur des approches probabilistes. La méthode probabiliste la plus utilisée est l'inférence par maximum de vraisemblance [Stamatakis, 2014; Guindon et al., 2010]. Cette approche prend en paramètres deux types de données :

— l'alignement multiple des séquences nucléotidiques ou protéiques du jeu de données ;

— un modèle d'évolution des séquences (ex : JC69, K2P, HKY85, GTR, CAT) qui permet de prendre en compte les différents taux de permutations des nucléotides Arenas [2015] ;

L'inférence par maximum de vraisemblance est une approche qui donne la topologie pour laquelle la probabilité que le modèle d'évolution génère le jeu de données d'entrée (alignement multiple des séquences) est maximale. Cette approche est implémentée dans le logiciel RAxML [Stamatakis, 2014] qui a été utilisé dans le cadre d'expériences effectuées au cours de cette thèse.

L'*arbre d'espèces* représente les relations de parentés entre différentes espèces plus ou moins proches. De tels arbres ont d'abord été inférés à partir d'observations morphologiques et ont par la suite été affinés par inférence à partir d'arbres de gènes. Cependant les topologies des arbres de gènes pouvant être fortement divergentes de celle de l'arbre des espèces associé, l'inférence d'un arbre des espèces nécessite une grande quantité d'arbres de gènes pour obtenir une topologie consensus relativement sûre de l'arbre des espèces [Maddison, 1997; Ma et al., 2000].

Dans un arbre d'espèces, tous les nœuds ancestraux correspondent à des événements de *spéciation*. La spéciation est le phénomène évolutif par lequel un groupe d'individus d'une espèce évolue distinctement des autres individus, c'est-à-dire subit une grande quantité de modifications génétiques conservées par les descendants. La différence entre ces individus et les autres est telle qu'elle donne naissance à une nouvelle espèce.

Pour inférer la complexité de l'histoire évolutive à l'échelle des gènes, l'inférence par maximum de vraisemblance à partir des séquences nucléotidiques est insuffisante, car elle ne prend pas en compte le changement de contenu en gènes au cours de l'évolution.

Pour l'inférence d'arbres phylogénétiques moléculaires, d'autres méthodes permettent de considérer conjointement les événements de mutations ponctuelles et de modifications du contenu en gènes des séquences d'ADN, comme la méthode PHYLDOG [Boussau et al., 2013]. Dans une approche similaire, la méthode PROFILENJ [Noutahi et al., 2016], utilisée dans ce manuscrit, permet à partir d'un arbre de gènes inféré par maximum de vraisemblance de corriger la topologie de ces nœuds internes, dont le support est inférieur à un seuil, de telle sorte que le nombre d'événements modifiant le contenu en gènes est minimal.

1.3.3 La réconciliation d'arbres de gènes

L'*arbre de gènes* correspond à l'histoire évolutive d'une famille de gènes homologues. C'est-à-dire un ensemble de gènes descendant d'un gène ancestral commun et qui souvent code pour des protéines ou ARN fonctionnels ayant des fonctions similaires. L'histoire des gènes peut être affectée par des modifications du contenu en gènes et être en conflit topologique avec l'histoire évolutive des espèces. La *réconciliation* permet d'expliquer les différences de topologie entre un arbre de gènes et l'arbre des espèces associé (*cf.* figure 1.4). Ces différences s'expliquent par le fait que les gènes peuvent subir des événements évolutifs sans pour autant résulter en la spéciation d'une espèce en deux autres. La réconciliation permet d'annoter les nœuds des arbres de gènes avec ces événements évolutifs, pour permettre de résoudre les conflits entre la topologie de l'arbre de gènes avec celle de l'arbre d'espèces.

La *réconciliation parcimonieuse* est la première à avoir été décrite et consiste à *minimiser* le nombre d'événements évolutifs pour réconcilier un arbre de gènes avec un arbre des espèces. Elle a été décrite par Goodman et al. [1979] qui s'intéressaient à inférer l'arbre de gènes de la famille des globines dans l'arbre des espèces des vertébrés. Dans cet article, les auteurs ont défini un modèle de duplication-perte de gènes (*DL model*) qui a largement été réutilisé par la communauté scientifique [Guigó et al., 1996; Maddison, 1997; Doyon, 2010; Chauve et al., 2013].

Cependant, il ne prend pas en compte les événements de transfert de gène. Pour prendre en compte ce type de réarrangement, un nouveau modèle a été développé basé sur le *DL model* auquel on intègre les transferts de gènes. Ce modèle est appelé modèle de duplication-transfert-perte de gènes (*DTL model*) [Hallett and Lagergren, 2001; Doyon et al., 2010; Patterson et al., 2013;

Szöllősi et al., 2013]. Dans ce manuscrit nous utilisons l'algorithme de réconciliation ecceTERA [Jacox et al., 2016] qui permet de réconcilier un arbre de gène avec l'arbre d'espèces en considérant avec le modèle *DL* ou le modèle *DTL*.

Dans un arbre de gènes, les événements évolutifs aux nœuds ancestraux peuvent être de différente nature. Dans le modèle que nous considérons, un nœud ancestral peut être annoté par un événement de spéciation, duplication de gène, perte de gène ou transfert de gène.

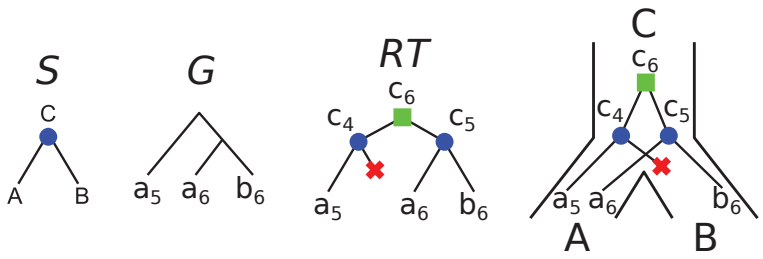


FIGURE 1.4 – Représentation de la réconciliation d'un arbre de gènes G avec un arbre des espèces S, où a_5 et a_6 sont des gènes de l'espèce A et b_6 un gène de l'espèce B. Au milieu, la représentation RT (*Reconciled Tree*) de la réconciliation de G avec la topologie de S. À droite une représentation de cette réconciliation à l'intérieur de l'arbre des espèces. (*cf.* table 1.1 pour la signification des symboles).

Pour la suite de ce manuscrit, il est nécessaire d'introduire un autre type d'arbre phylogénétique, l'**arbre d'adjacences de gènes**. Ce type d'arbre est similaire à l'arbre de gènes excepté que les nœuds de l'arbre ne correspondent plus à des gènes mais à des adjacences de gènes (*cf.* sous-section 1.1.2) et permet de représenter la coévolution d'une paire de gènes. Dans les modèles considérés dans ce manuscrit de thèse, il est nécessaire d'associer de nouveaux événements évolutifs aux nœuds de ce type d'arbres, en plus de ceux présents chez les arbres de gènes :

— la *duplication d'adjacences de gènes* correspondant à une duplication simultanée de deux gènes adjacents ;
— la *perte d'adjacence de gènes* correspondant à une perte simultanée de deux gènes adjacents ;
— le *transfert d'adjacence de gènes* correspondant à un transfert simultané de deux gènes adjacents ;
— la *cassure d'adjacence de gènes* correspondant à un rupture de l'adjacence entre deux gènes résultant d'un réarrangement chromosomique ;
— le *gain d'adjacence de gènes* correspondant à la formation d'une adjacence entre deux gènes à la suite d'un réarrangement chromosomique ou d'un changement du contenu en gène (la perte d'un gène entre

deux autres gènes par exemple).

Les tables 1.1 et 1.2 offrent un récapitulatif des événements évolutifs considérés dans les arbres phylogénétiques et la figure 1.5 donne une illustration des différents arbres phylogénétiques considérés dans ce manuscrit.

Événements évolutifs	Notation	Symbole graphique
Spéciation	Spec	●
Duplication de gène	GDup	■
Duplication d'adjacence de gènes	ADup	□
Perte de gène	GLos	✖
Perte d'adjacence de gènes	ALos	⨻
Transfert de gène	GTra	pas de symbole
Transfert d'adjacence de gènes	ATra	pas de symbole
Création d'adjacence de gènes	Ga	▲
Cassure d'adjacence de gènes	Br	↯

TABLE 1.1 – Liste des événements évolutifs dans les arbres phylogénétiques considérés.

Type d'arbre	Événements évolutifs associés
Arbre des espèces	●
Arbre de gènes	●, ■, ✖, GTra
Arbre d'adjacences	●, ■, ✖, GTra, □, ⨻, ATra, ▲, ↯

TABLE 1.2 – Liste des arbres phylogénétiques considérés et des événements évolutifs associés.

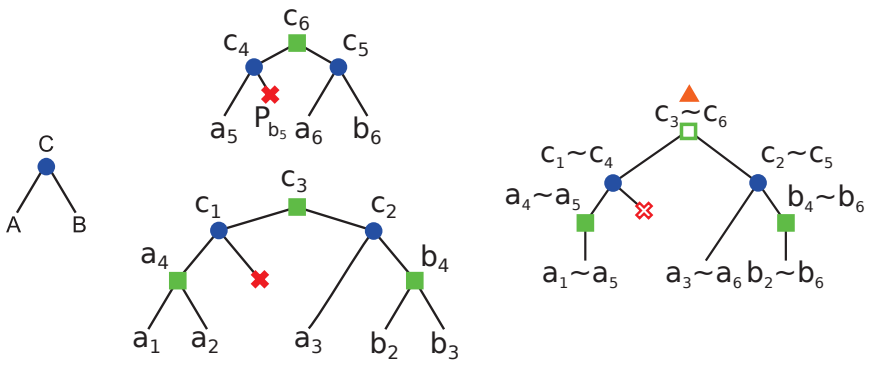


FIGURE 1.5 – Représentation des différents arbres phylogénétiques. Sur la gauche un arbre des espèces, au milieu deux arbres de gènes et à droite un arbre d'adjacences de gènes reconstruits à partir des arbres de gènes.

Chapitre 2

Séquençage et assemblage *de novo* de génomes

Dans cette thèse, nous nous intéressons à l'histoire évolutive de la structure des génomes. Pour reconstruire cette histoire évolutive, il est nécessaire d'avoir accès à l'enchaînement linéaire complet de marqueurs génomiques le long des chromosomes des espèces actuelles de la phylogénie étudiée. Pour cela, il est donc nécessaire de déterminer la séquence d'ADN complète ainsi que la structure des génomes sous la forme de chromosomes linéaires ou circulaires par séquençage et assemblage de la séquence d'ADN. Les chromosomes obtenus sont ensuite annotés par la position et l'orientation de marqueurs génomiques, comme les gènes.

Les méthodes de séquençage, couramment et actuellement utilisées, clivent la molécule d'ADN du génome à séquencer en fragments génomiques de quelques dizaines à centaines de bases pour les technologies haut-débit de 2^{e} génération et plusieurs milliers à dizaines de milliers de bases pour celles de 3^{e} génération. Il faut ensuite assembler et ordonner ces fragments génomiques séquencés afin d'obtenir la structure complète du génome qui peut ensuite être annotée par l'identification de marqueurs tels que les gènes. La figure 2.1 illustre le fait qu'il es possible d'avoir la séquence complète d'une portion d'ADN sans en connaître sa structure totale, ce qui arrive fréquemment dans les bases de données génomiques.

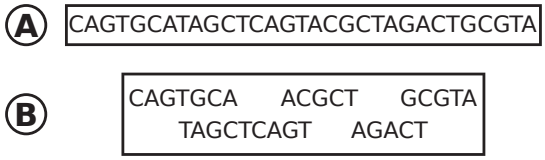


FIGURE 2.1 – **A** représente une portion d'ADN dont la séquence et la structure sont complètes. **B** représente cette même portion d'ADN pour laquelle la séquence est complète mais pas la structure.

Dans la section 2.1, nous présentons différentes méthodes et technologies de séquençage permettant l'obtention de la séquence nucléotidique complète de génomes. Cette séquence se présente sous la forme de fragments communément appelés *reads*. Dans la section 2.2, nous présentons différentes approches permettant d'assembler les *reads* en sortie du séquençage d'un génome afin d'obtenir la séquence linéaire complète de celui-ci. Dans un premier temps, nous présentons des méthodes d'assemblage *de novo* qui à partir des *reads* permettent d'obtenir des fragments génomiques plus longs, appelés *contigs*. Dans un deuxième temps, nous décrivons des méthodes permettant de résoudre l'ordre relatif des contigs le long de la séquence du génome à partir de différentes sources d'informations synténiques, appelées méthodes de *scaffolding*.

2.1 Séquençage des génomes

Dans le courant des années 1970, les deux premières méthodes de séquençage de l'ADN voient le jour, la méthode introduite par Gilbert et Maxam, appelée *méthode de séquençage chimique* [Gilbert and Maxam, 1973] et la méthode développée par Sanger et ses collègues, appelée communément *méthode Sanger* [Sanger and Coulson, 1975; Sanger et al., 1977] [1]. La méthode Sanger est devenue par la suite la méthode de séquençage de référence pour le séquençage des génomes pour trois raisons majeures :

— celle-ci utilise moins de molécules radioactives, plus tard remplacées par des sondes fluorescentes, pour déterminer l'enchaînement des nucléotides ;
— la méthode coûte également moins cher que sa concurrente ;
— l'approche de la méthode Sanger nécessite moins d'analyses pour déterminer la séquence d'ADN et est donc plus facilement automatisable que la méthode de séquençage chimique.

La méthode Sanger se sert du mécanisme naturel de synthèse de l'ADN par les molécules d'ADN polymérase pour permettre de déterminer la séquence d'ADN du génome séquencé.

1. *cf.* référence [Heather and Chain, 2016] pour une revue sur les méthodes de séquençage.

2.1.1 Technologies de séquençage haut-débit (*NGS*) de 1re génération

Au cours des années 2000, des nouvelles technologies de séquençage haut-débit, communément appelées *NGS* [2], ont vu le jour et progressivement remplacées la technologie de séquençage Sanger, celle-ci étant trop coûteuse en temps et en argent pour le séquençage de grands génomes [3]. Par exemple, la méthode Sanger a été utilisée dans le projet international de séquençage du génome humain (*Human Genome Project*) qui a eu lieu entre 1990 et 2003 et a coûté aux alentours de $3.000.000.000\ US\$$ [Human Genome Sequencing Consortium, 2004; Hutchison, 2007]. Aujourd'hui, il est possible de séquencer le génome complet d'un être humain en l'espace de quelques heures pour un coût d'environ $1.000\ US\$$. Cette amélioration est due aux technologies de séquençage de nouvelles générations dont une part importante sont basées sur le principe de la méthode Sanger, améliorée pour permettre un séquençage automatique et haut-débit [Metzker, 2010; Goodwin et al., 2016]. La figure 2.2 représente le coût de séquençage pour un génome humain et par millions de bases (Mb) de 2001 à 2015. La chute des coûts au cours de l'année 2007 coïncide avec la standardisation de l'utilisation des nouvelles technologies de séquençage haut-débit dans les projets de séquençage de génomes entiers.

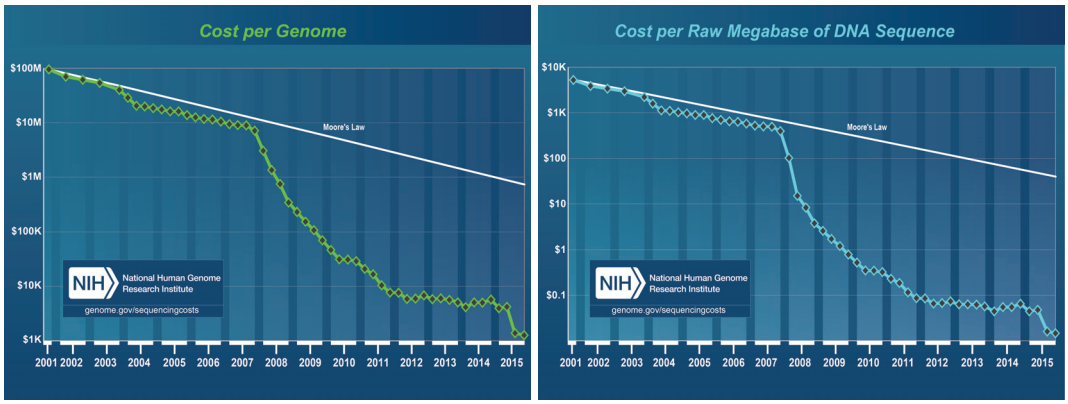


FIGURE 2.2 – Évolution du coût du séquençage de génomes au cours des années 2000. Graphique de gauche : évolution du coût de séquençage d'un génome humain. Graphique de droite : évolution du coût de séquençage par millions de bases (Mb). (figures tirées de `https://www.genome.gov/sequencingcostsdata/`)

2. *Next-Generation Sequencing*.
3. taille de génome supérieur à 100 Mb.

La première technologie de séquençage nouvelle génération a été développée par *Lynx Therapeutics Inc.* en 2000 [Brenner et al., 2000], cette innovation se détachait de la méthode Sanger par le haut débit de séquençage. Cependant, du fait de sa complexité et du fort taux d'erreurs de séquençage de la méthode, celle-ci n'a jamais été commercialisée.

Actuellement, deux technologies de séquençage dominent le marché des séquenceurs de nouvelles générations :

— la technologie Illumina de la compagnie *Illumina Inc.* (*NGS* de 2e génération) ;
— la technologie PacBio de la compagnie *Pacific Biosciences of California, Inc.* (*NGS* de 3e génération).

2.1.2 *NGS* de 2e génération

La majorité des méthodes de séquençage haut-débit de 2e génération suivent le protocole suivant :

1. extraction de l'ADN que l'on souhaite séquencer ;
2. fragmentation de l'ADN en fragments génomiques dont la taille est dépendante du protocole utilisé ;
3. ajout des adaptateurs en extrémité des fragments génomiques. Pour cette étape, les protocoles varient selon que l'on fasse le choix de faire un séquençage simple ou apparié ;
4. dépôt des fragments d'ADN simple brin sur des *clusters* de séquençage [4] où les fragments seront amplifiés et séquencés ;
5. amplification des fragments d'ADN par *PCR* [5] permettant l'obtention d'un signal de fluorescence plus intense ;
6. séquençage des fragments génomiques par synthèse du brin complémentaire des fragments d'ADN simples brins, par des molécules d'ADN polymérases, et lecture des nucléotides séquencés.

Pour la technologie Illumina et les autres technologies *NGS* de 2e génération, la taille des *reads* varient entre quelques dizaines et quelques centaines de paires de bases. Cette courte longueur des séquences d'ADN ne permet pas de localiser les zones répétées dont la longueur dépasse celle des *reads* [Treangen and Salzberg, 2012].

4. de telle sorte, qu'il n'y ait qu'un fragment par cluster.
5. *Polymerase-Chain Reaction*

Pour pallier ce problème, une stratégie de séquençage a été développée et permet d'apparier des *reads* entre lesquels la distance approximative est dépendante de la longueur des fragments d'ADN obtenus lors de l'étape 2 de fragmentation du protocole de séquençage.

Séquençage de *reads* appariés Le choix de la stratégie de séquençage avec des *reads* appariés s'effectue lors de la fragmentation de l'ADN et de l'ajout d'adaptateurs aux extrémités des fragments génomiques. Tout d'abord, le choix de la distance entre les deux *reads* appariés est déterminée lors de l'étape de fragmentation. La taille des fragments, que l'on nomme *taille d'insert*, détermine la distance approximative entre les *reads* d'une même paire.

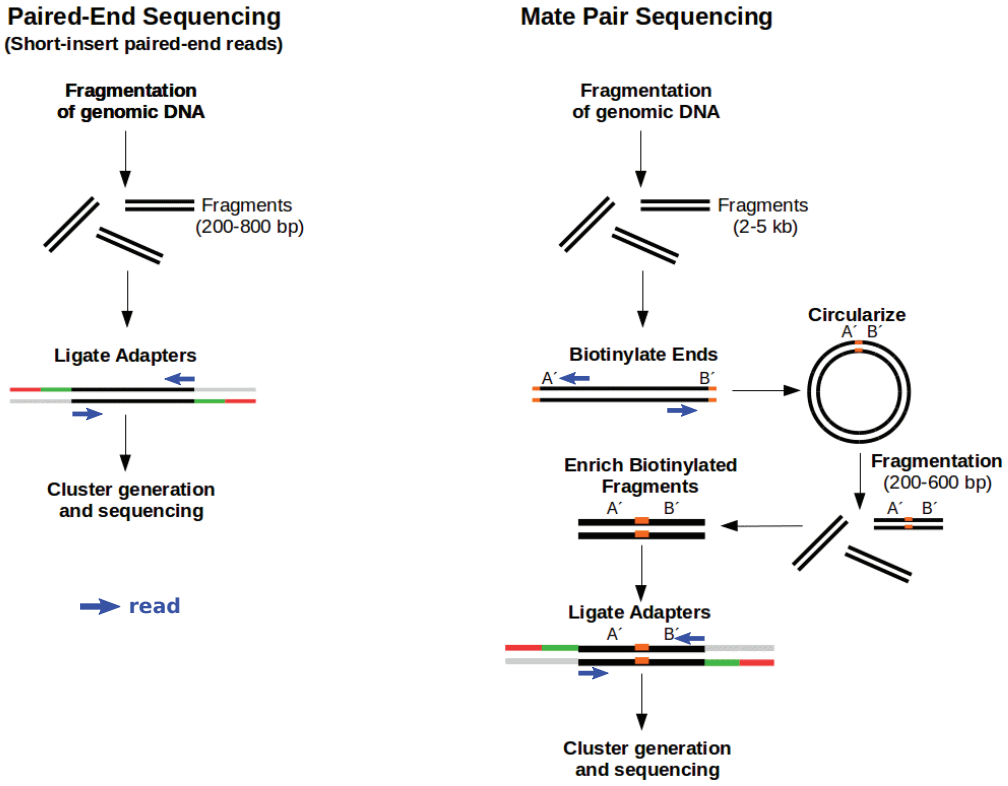


FIGURE 2.3 – Séquençage de *reads* appariés, *paired-end* à gauche et *mate pair* à droite. La portion rouge représente la séquence d'ancrage du fragment d'un brin d'ADN sur la surface de séquençage des séquenceurs et la portion verte représente la séquence de recrutement de l'ADN polymérase. Les flèches bleues indiquent les positions et sens de séquençage des *reads* (image tirée de `https://www.ecseq.com/support/ngs/what-is-mate-pair-sequencing-useful-for`).

Si la taille est inférieure à 1 *kb* (*cf.* partie gauche de la figure 2.3), le séquençage apparié est appelé *paired-end*. Dans ce cas, on ajoute des amorces pour fixer les fragments sur la surface de séquençage et pour recruter l'ADN polymérase. Lors du dépôt sur le support de séquençage, les deux brins sont

séparés et amplifiés séparément. Ainsi pour le brin d'ADN matrice, on séquence l'extrémité $5'$ dans le sens $5' \rightarrow 3'$ (*Forward*) et l'extrémité $3'$ dans le sens $3' \rightarrow 5'$ (*Reverse*). Cette orientation des paires de *reads* est communément noté *FR* ou $\rightarrow\leftarrow$.

Si la taille des fragments génomiques est supérieure à $1\ kb$ (*cf.* partie droite de la figure 2.3), les technologies ne permettent pas de séquencer les *reads* en *paired-end* car ces séquences sont trop longues pour les plateformes de séquençage de 2e génération. Il est donc nécessaire d'ajouter une étape de circularisation des fragments. Celle-ci consiste, dans un premier temps, à ajouter de la biotine aux extrémités des fragments d'ADN. Les fragments génomiques sont ensuite circularisés. Puis, les ADN circulaires sont fragmentés en fragments de l'ordre de quelques centaines de paires de base et des billes magnétiques de streptavidine sont utilisées pour sélectionner les fragments d'ADN contenant de la biotine. Ensuite, on suit le même protocole que pour les *reads paired-end*. Ainsi, pour le brin d'ADN matrice, on séquence l'extrémité 5′ dans le sens $3' \rightarrow 5'$ (*Reverse*) et l'extrémité 3′ dans le sens $5' \rightarrow 3'$ (*Forward*). Cette orientation des paires de *reads* est communément noté *RF* ou $\leftarrow\rightarrow$. Ce séquençage apparié pour les *reads* supérieur à $1\ kb$ est appelé *mate pair*.

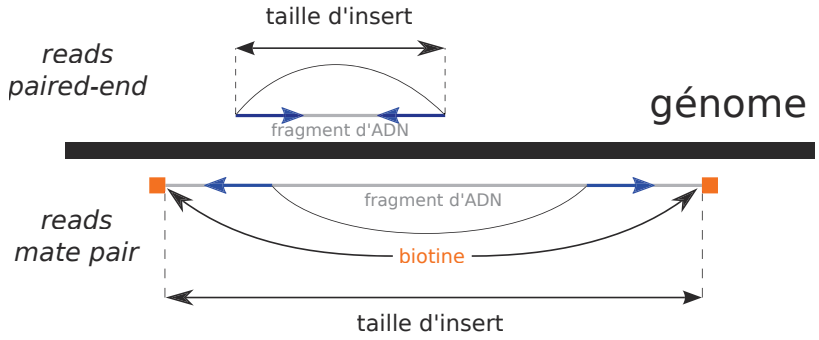


FIGURE 2.4 – Schéma illustrant la taille d'insert des *reads* appariés (*paired-end* et *mate-pair*) correspondant à la taille des fragments d'ADN obtenus lors de l'étape de fragmentation. Les fragments à partir desquels les *reads* ont été séquencées sont schématisées en **gris** ainsi que les biotines (en **oranges**) pour les *reads mate pair*. Ce schéma doit être mis en perspective avec le schéma de la figure 2.3.

L'ensemble des *reads* appariés produit lors d'un "*run*" de séquençage sont regroupés sous le terme de *librairie* caractérisée par sa taille d'insert [6] et l'orientation de séquençage des *reads* appariés (FR ou RF). Pour chaque librairie, on a donc une collection de paires de séquences (les *reads*) dont on connaît la distance approximative qui les sépare sur le génome (la taille d'insert). La

6. égale à la taille des fragments génomiques après la première étape de fragmentation.

figure 2.4 illustre la taille d'insert pour des *reads* appariés de *library paired-end* et *mate pair*.

Cette stratégie de séquençage permet d'obtenir des informations supplémentaires sur la structure linaire du génome séquencé qui seront exploitées lors de l'étape d'assemblage des *reads* sous forme de contigs puis de *scaffolds* (*cf.* section 2.2, p. 33). Ces données de séquençage appariées sont utilisées pour résoudre en partie la localisation de zones répétées. La résolution de la localisation des zones répétées reste incomplète à cause de la présence d'artefacts lors de la génération des *library paired-end* et *mate pair* entraînant une orientation incorrecte des paires de *reads*. Cela introduit des signaux contradictoires qui génèrent des erreurs. Même sans ces artefacts, le problème algorithmique qui consiste à ordonner et orienter des contigs à partir de données de séquençage appariées, appelé *Scaffolding Problem* [Huson et al., 2002] est un problème difficile qui nécessite l'utilisation d'heuristiques pour le résoudre.

2.1.3 *NGS* de 3e génération

Afin de pallier les problèmes de génération d'artefacts et de localisation des zones répétées des génomes, une nouvelle génération de technologies de séquençage a été développée, les *NGS* de 3e génération. Bien que les approches des *NGS* de 3e génération soient très différentes entre elles [Schadt et al., 2010], ces technologies ont deux points communs, apportant une forte amélioration du séquençage par rapport aux *NGS* de 2e génération :

1. elles ne nécessitent pas l'amplification de l'ADN, ce qui permet d'éviter l'introduction de mutations dans la séquence d'ADN lors de l'amplification par PCR ;
2. elles génèrent des *reads* d'une taille de plusieurs milliers à quelques centaines de milliers de paires de bases.

Pour illustrer les technologies de séquençage de 3e génération, nous présentons la technologie de séquençage *PacBio* qui est l'une des premières technologies de séquençage de 3e génération à avoir été commercialisée et est actuellement la technologie *NGS* de 3e génération la plus utilisée pour le séquençage de génomes entiers.

Protocole de séquençage de la technologie PacBio La technologie PacBio [Eid et al., 2009], comme les technologies de séquençage de 2e génération, se sert de la capacité des ADN polymérases à pouvoir synthétiser le brin

complémentaire d'une séquence d'ADN simple brin pour déterminer la composition en nucléotides de celui-ci. Pour améliorer le processus, la technologie PacBio utilise une surface de séquençage contenant des puits calibrés de $100\ nm$ de profondeur, appelés *Zero-Mode Waveguides* (*ZMW*) [Levene et al., 2003], au fond desquels est fixée une ADN polymérase. Des nucléotides modifiés sont utilisés et contiennent une sonde fluorescente (une couleur pour chaque type de nucléotides) qui lors du recrutement par l'ADN polymérase est clivée et libérée dans la solution de séquençage. Le protocole général de la méthode PacBio est le suivant :

1. extraction et purification de l'ADN de l'individu à séquencer ;
2. fragmentation de l'ADN en fragments de taille de plusieurs dizaines de milliers de bases ;
3. dépôt des fragments génomiques sur la surface de séquençage, contenant les *ZMW* avec une ADN polymérase, et des nucléotides fluorescents ;
4. recrutement des fragments génomiques simple brin par l'ADN polymérase et des nucléotides dont le signal de fluorescence est détecté uniquement au niveau de l'ADN polymérase fixée au fond du *ZMW*.

La lecture du séquençage du brin complémentaire est effectuée en temps réel dans l'ensemble des *ZMW* de la plaque de séquençage sans étape de nettoyage et clivage de terminateur en extrémité des nucléotides, ce qui permet une économie en temps et en ressources matérielles.

Cependant, ces méthodes ont à l'heure actuelle un taux d'erreurs de séquençage beaucoup plus élevé que les technologies de 2e génération [Rhoads and Au, 2015] ce qui nécessite une couverture de séquençage plus importante ou une combinaison avec des données de séquençage de technologies de 2e génération.

Conclusion sur le séquençage L'apport des technologies de séquençage haut-débit de 2e et 3e génération a permis de diminuer drastiquement le coût et le temps de séquençage de génomes entiers (*cf.* figure 2.2, p. 27). L'argent et le temps dépensés pour obtenir la séquence d'ADN du premier génome humain, rapporté au million de bases séquencées, n'aurait pas permis d'effectuer des études sur l'évolution moléculaire des espèces à l'échelle du génome entier. La nécessité pour les technologies de séquençage haut-débit de

fragmenter l'ADN en petites portions de génomes a nécessité le développement d'outils algorithmiques afin d'assembler l'ensemble de ces fragments pour retrouver la structure linéaire complète des génomes. Cette étape est importante pour permettre la détection de marqueurs génomiques, tels que les gènes, sur les génomes, et essentielle pour permettre de reconstruire l'histoire évolutive complète de l'ordre des gènes le long d'une phylogénie d'espèces.

2.2 Assemblage *de novo* de génomes entiers

Avec l'avènement des technologies de séquençage nouvelle génération (NGS), des méthodes algorithmiques ont du être développées pour assembler les millions de reads générés, afin de déterminer la structure linéaire complète de la séquence du génome d'intérêt (c.-à-d. l'enchaînement complet des nucléotides de la séquence d'ADN du génome séquencé).

L'assemblage des génomes ne se limite pas à une étape algorithmique mais est une succession d'étapes. L'implémentation de ces étapes est spécifique aux données générées par chaque technologie. Les étapes de l'assemblage des génomes sont :

1. le prétraitement des données de séquençage ;
2. l'assemblage *de novo* des *reads* sous forme de séquences nucléotidiques continues, appelées contigs ;
3. le *scaffolding* des contigs permettant d'ordonner et d'orienter les contigs sous forme de *scaffolds* (l'ensemble des *scaffolds* représentant dans le meilleur des cas l'ensemble des chromosomes).

Ekblom and Wolf [2014] passent en revue l'ensemble des étapes effectuées lors du séquençage et de l'assemblage *de novo*.

2.2.1 Prétraitement des données de séquençage

Avant d'effectuer l'assemblage des *reads* en sortie de séquençage, ceux-ci doivent être analysés et traités afin de retirer les séquences de mauvaise qualité ou ne correspondant pas à l'espèce dont on souhaite reconstruire la séquence d'ADN.

Cette étape consiste à effectuer l'élagage (*trimming*) des *reads*. Pour effectuer cette étape, il est possible d'utiliser l'outil TRIMMOMATIC [Bolger et al.,

2014]. TRIMMOMATIC retire les séquences d'amorces utilisées lors du séquençage et qui peuvent être présentes dans la séquence des *reads*. Il permet également de retirer une partie de l'extrémité 3' de la séquence des *reads*, qui est souvent de plus mauvaise qualité pour les données de séquençage obtenues avec une technologie de 2e génération.

2.2.2 Méthode assemblage *de novo*

Après élagage des *reads*, ceux-ci sont alignés les uns avec les autres afin d'obtenir la séquence de nucléotides contiguë la plus longue, appelée *contig*. Cette étape est appelée *assemblage de novo* de génome. Au cours des années 2000, un très grand nombre de méthodes d'assemblage *de novo* ont été développées pour permettre d'assembler les *reads* en sortie des différentes technologies de séquençage [Sohn and Nam, 2016].

L'approche algorithmique utilisée pour l'assemblage *de novo* des génomes est dépendante de la taille des *reads* disponibles. Pour des *reads* courts, la stratégie algorithmique la plus couramment utilisée est celle basée sur le graphe de *de Bruijn* [Idury and Waterman, 1995; Compeau et al., 2011]. Pour les *reads* longs, la plupart des méthodes utilisent une approche d'assemblage connue sous le nom de *Overlap-Layout-Consensus* [Staden, 1979; Li et al., 2012] qui fut initialement développée pour l'assemblage de données de séquençage de la méthode Sanger.

Ces approches algorithmiques sont implémentées dans les deux outils d'assemblage couramment utilisés pour l'assemblage *de novo* de génomes et auxquelles nous faisons allusion dans ce manuscrit :

— ALLPATHS-LG : une des méthodes les plus utilisées pour d'assemblage *de novo* de *reads* courts générés à partir de la plateforme Illumina et qui est basée sur une implémentation de l'algorithme de graphe de *de Bruijn* [Gnerre et al., 2011] ;

— CANU : une méthode développée pour l'assemblage de longs *reads* de technologies haut débit de 3e génération PacBio ou Oxford Nanopore sur la stratégie algorithmique d'assemblage *Overlap-Layout-Consensus* [Koren et al., 2017].

Pour plus d'informations sur les algorithmes et méthodes pour l'assemblage de génomes voir les références [Miller et al., 2010; Li et al., 2012; Wajid and Serpedin, 2012; Nagarajan and Pop, 2013].

2.2.3 *Scaffolding* de génomes

L'assemblage des génomes en sortie des méthodes d'assemblage *de novo* étant souvent fortement fragmenté (dû à la problématique de résolution de la localisation des zones répétées ou à la faible couverture de séquençage de certaines zones du génome), des méthodes dites de *scaffolding* ont été développées.

Le *scaffolding* génomique est une approche qui consiste à ordonner et orienter des marqueurs génomiques [7] le long de la séquence du génome d'une espèce afin d'améliorer la reconstruction de sa structure linéaire. Plusieurs approches sont possibles pour effectuer le *scaffolding* de contigs :

— se servir des données de séquençage appariées ;
— utiliser une carte génomique pour assigner les marqueurs/contigs aux chromosomes de l'espèce d'intérêt ;
— utiliser des approches de génomique comparative pour ordonner les marqueurs/contigs avec un ou plusieurs génomes de référence.

Scaffolding à partir de données de séquençage appariées

Une première étape pour effectuer le *scaffolding* de contigs à partir de données de séquençage appariées consiste à aligner les paires de *reads* sur les contigs produits lors de l'assemblage *de novo*. Pour cela des méthodes dites de *mapping* [Trapnell and Salzberg, 2009], comme l'outils d'alignement BOWTIE2 [Langmead and Salzberg, 2012], ont été développées permettant d'aligner des *reads* contenant des mutations ou erreurs dans leur séquence et de considérer de multiple positions sur l'assemblage du génome obtenu à partir d'une méthode d'assemblage *de novo*.

Le *Scaffolding Problem* a initialement été formalisé par Huson et al. [2002] qui ont permis de mettre en évidence la grande complexité de ce problème [8]. Par la suite, diverses méthodes de *scaffolding* à partir de données de séquençage ont été développées (*cf.* [Hunt et al., 2014] pour une revue d'un éventail des méthodes existantes).

Dans la suite de cette section, nous détaillons l'approche développée dans l'algorithme BESST [Sahlin et al., 2014, 2016] permettant de résoudre ce problème. Les prédictions d'adjacences de *scaffolding* pour lesquelles BESST calcule un score de *scaffolding* sont utilisées en entrée de notre méthode pour l'amélioration du *scaffolding* de génomes actuels. Notre méthode permettant

7. des contigs dans le cadre de l'assemblage des génomes.
8. problème NP-difficile.

l'intégration des adjacences de *scaffolding* pondérées produites par BESST est présentée dans le chapitre 5.

La méthode de *scaffolding* BESST Pour résoudre le *Scaffolding Problem*, qui consiste à ordonner et orienter un jeu de contigs à partir de données de séquençage appariés, BESST considère une librairie de *reads* appariés avec une taille moyenne d'insert μ et un écart-type σ. Les *reads* appariés sont alignés sur les contigs à ordonner et orienter avec un outil de *mapping*, comme BOWTIE2 [Langmead and Salzberg, 2012]. Cette étape de *mapping* génère un *graphe de scaffolding* G où chaque contig c_x est composé de deux sommets $c_{x,H}$ et $c_{x,T}$ correspondant resp. à l'extrémité 5′ et 3′. Pour un couple de *reads* appariés (x_1,x_2), si x_1 s'aligne exclusivement sur le contig c_k et x_2 sur le contig c_m, avec $k \neq m$, alors cette paire induit une orientation relative et une distance approximative entre les contigs c_k et c_m. Ce lien est représenté par un lien l.

Dans un premier temps BESST considère exclusivement les longs contigs dont la taille est supérieure à $\mu + 4\sigma$ correspondant à une valeur pour laquelle il est improbable qu'un long contig soit localisé entre les alignements de *reads* d'une même paire provenant d'une librairie. BESST estime ensuite la distance entre les paires de longs contigs, associés par un lien cohérent, par maximum de vraisemblance avec l'outil GAPEST [Sahlin et al., 2012]. Pour chaque paire de longs contigs c_x et c_y liés par une branche cohérente dans le graphe de *scaffolding* la méthode calcule deux scores :

1. le score de variation du lien, noté π_σ, indique le degré de similitude entre les distances observées pour l'ensemble des paires de *reads* et la distance estimée par GAPEST entre les contigs c_x et c_y. Où $\pi_\sigma = 1$ indique que les distances observées pour l'ensemble des paires de *reads* sont similaires à la distance estimée par GAPEST ;
2. le score de dispersion du lien, noté π_ζ, indique le degré de similitude entre la distribution des *reads* sur le contig c_x et la distribution des *reads* sur le contig c_y, pour les *reads* appariés soutenant le lien entre les contigs c_x et c_y. Où $\pi_\zeta = 1$ si pour les paires de *reads* liant les contigs c_x et c_y, la distribution des *reads* sur le contig c_x est similaire à celle des *reads* sur le contig c_y.

Pour plus de détails sur le calcul des scores π_σ et π_ζ voir la référence [Sahlin et al., 2014].

Le score global de chaque lien, noté π, est obtenu avec l'équation suivante :

$$\pi = \begin{cases} \pi_\sigma + \pi_\zeta & \text{si } \pi_\sigma, \pi_\zeta \geq 0{,}5 \\ 0 & \text{sinon} \end{cases}$$

Le *scaffolding* des longs contigs consiste à trouver un chemin sur le graphe G de poids maximal. Si il y a le choix entre deux liens avec chacun un score $\pi \geq 0{,}9$ alors les deux chemins sont conservés.

Pour compléter le graphe de *scaffolding* G, les contigs courts, avec une taille inférieure à $\mu + 4\sigma$ sont ajoutés et utilisés pour compléter les espaces entre les longs contigs. L'approche consiste à trouver un chemin qui maximise le nombre de paires de *reads* liant les contigs courts entre eux et avec les contigs longs tout en conservant le chemin global produit par le *scaffolding* des longs contigs sur le graphe G [Sahlin et al., 2014].

Scaffolding à partir de carte chromosomique

Une carte chromosomique correspond à l'assignation de marqueurs génétiques le long des chromosomes du génome d'une espèce. Ces marqueurs peuvent correspondre à des contigs permettant d'assigner de longues séquences d'ADN à des chromosomes et d'améliorer le *scaffolding* de génomes en donnant l'ordre relatif des contigs assignés sur la carte chromosomique.

Dans cette section, nous présentons la méthode employée pour la génération d'une carte chromosomique par la technique *FISH* (*Fluorescence In Situ Hybridisation*).

Génération de carte chromosomique par *FISH* Pour générer une carte génomique d'une espèce par *FISH*, une première étape consiste à produire des sondes fluorescentes à partir de portions de la séquence des contigs ou *scaffolds* de l'assemblage de référence de l'espèce. Ces sondes sont ensuite hybridées *in situ* sur les chromosomes pour localiser la position de ces séquences, et par extension des contigs/*scaffolds* auxquels ces séquences appartiennent. Pour pouvoir orienter un contig sur un chromosome, il est nécessaire de produire deux sondes, une à chaque extrémité du contig. Une fois les deux sondes hybridées sur un chromosome, une première étape pour le cytogénéticien consiste à identifier le chromosome sur lequel elles sont localisées. Les deux couleurs d'émission des deux sondes fluorescentes permettent de déterminer l'orientation sur le chromosome après identification des extrémités 5' et 3' du chromosome par le cytogénéticien. La localisation des séquences

hybridées sur le chromosome est déterminée à partir des bandes chromosomiques afin de pouvoir définir leur position relative à d'autres marqueurs assignés sur ce chromosome. Un exemple de *FISH* est présenté dans la figure 2.5, représentant pour l'espèce *An. albimanus* la localisation *in situ* par fluorescence de quatre gènes en extrémités de deux contigs permettant leur orientation sur les chromosomes. Sur les photos de cette figure, on observe sur le chromosome des parties claires (vertes) et des parties sombres. Ces différents segments correspondent à des bandes chromosomiques permettant au cytogénéticien de pouvoir définir plus précisément la position des séquences hybridées le long d'un chromosome. Un exemple de carte cytogénétique de l'espèce *An. atroparvus* est disponible dans l'article [Artemov et al., 2015] et de l'espèce *An. albimanus* dans l'article [Artemov et al., 2017].

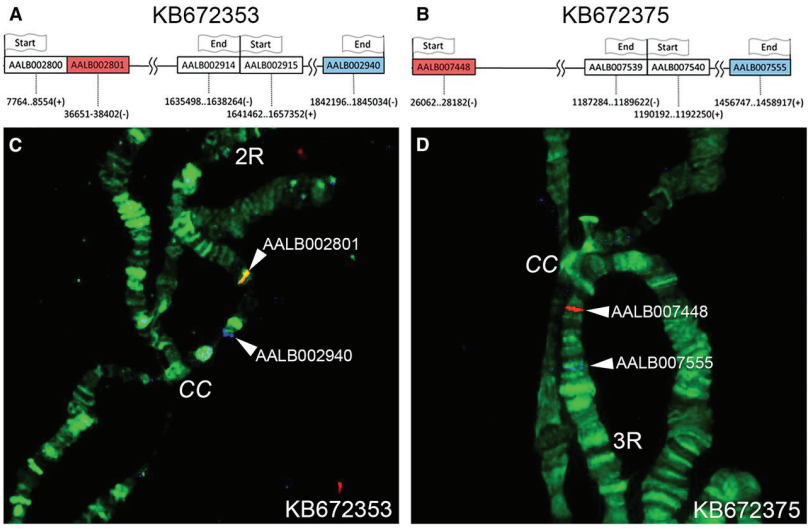


FIGURE 2.5 – Exemple de localisation des contigs KB672353 et KB672375 respectivement placés sur les bras chromosomiques $2R$ et $3R$ du génome d'*An. albimanus* par *FISH* (figure extraite de [Artemov et al., 2017]).

Scaffolding par génomique comparative

Une autre approche pour effectuer le *scaffolding* de contigs d'un assemblage de génome consiste à utiliser la séquence génomique d'un ou plusieurs autres génomes de référence. Notre travail s'appuyant sur la comparaison des assemblages de plusieurs génomes, nous nous attarderons plus sur la description des méthodes existantes dans cette catégorie.

Des méthodes utilisant des outils d'alignements deux à deux de génomes entiers sont parfois utilisées pour effectuer l'assemblage *de novo* de génomes d'intérêt en utilisant directement les *reads* appariés, produits lors du séquençage, qui sont ensuite alignés sur un ou plusieurs génomes de référence afin

de générer les contigs puis les *scaffolds*. C'est le cas pour la méthode RECORD [Buza et al., 2015] et dans [Schneeberger et al., 2011].

Dans cette section, nous nous concentrons sur les méthodes de *scaffolding* permettant d'ordonner et orienter les *contigs*. Pour la majorité de ces méthodes, une première étape consiste à effectuer un alignement deux à deux entre les contigs que l'on souhaite ordonner et orienter et le ou les génomes de référence, afin d'identifier les zones homologues entre les deux génomes et de définir les adjacences manquantes dans les contigs. Les outils d'alignements deux à deux les plus utilisés sont BLAT [Kent, 2002], BLAST [Altschup et al., 1990], LASTZ [Blanchette et al., 2004a], PROGRESSIVECACTUS [9] [Paten et al., 2011] et MUMMER [Delcher et al., 1999; Kurtz et al., 2004].

On peut classer les méthodes de *scaffolding* de génomes guidées par référence en deux grandes catégories :

— les méthodes utilisant un seul et unique génome de référence ;
— les méthodes utilisant plusieurs génomes de référence.

Cependant d'autres caractéristiques de ces méthodes sont importantes à prendre en compte. Comme l'utilisation de données de séquençage appariées pour guider le *scaffolding*, l'utilisation de la phylogénie des espèces pour pondérer les adjacences en fonction du degré de parenté des espèces ou encore la possibilité de pouvoir considérer des génomes de grandes tailles. En tout sept critères ont été identifiés et sont présentés dans la figure 2.6 qui classe 16 méthodes de *scaffolding* par génomique comparative dans différents ensembles caractérisant ces méthodes. La liste des 16 méthodes est la suivante :

— PROJECTOR2 [van Hijum et al., 2005] ;
— R2CAT [Husemann and Stoye, 2009] ;
— ABACAS [Assefa et al., 2009] ;
— MAUVE ALIGNER [Rissman et al., 2009] ;
— FILLSCAFFOLDS [Muñoz et al., 2010] ;
— TREECAT [Husemann and Stoye, 2010] ;
— RAGOUT [Kolmogorov et al., 2016] ;
— CHROMOSOMER [Tamazian et al., 2016] ;
— CAR [Lu et al., 2014] ;
— MEDUSA [Bosi et al., 2015] ;
— RACA [Kim et al., 2013] ;
— SIS [Dias et al., 2012] ;
— ALIGNGRAPH [Bao et al., 2014] ;
— OSLAY [Richter et al., 2007] ;
— CLA [Shaik et al., 2016] ;
— MULTI-CAR [Chen et al., 2016].

Nous présentons ci-après les méthodes de *scaffolding* par génomique comparative les plus répandues, les trois premières utilisent un seul génome de référence tandis que les trois dernières en utilisent plusieurs.

9. https://github.com/glennhickey/progressiveCactus.

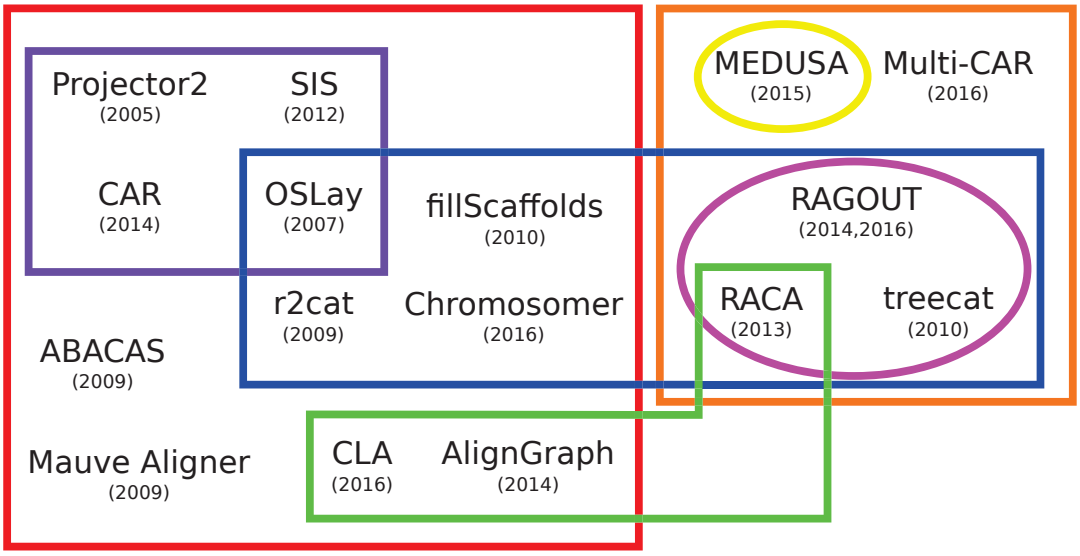


FIGURE 2.6 – Schéma de classification des différentes méthodes de *scaffolding* par génomique comparative. Les cadres représentent des ensembles qui caractérisent les différentes méthodes. Cadre **rouge** : utilisent un seul et unique génome de référence. Cadre **orange** : utilisent de multiple génomes de référence. Cadre **violet** : considèrent uniquement des génomes avec un seul et unique chromosome. Cadre **bleu** : considèrent des génomes de grandes tailles. Cadre **magenta** : utilisent des signaux phylogénétiques. Cadre **jaune** : considère des génomes de référence incomplètement assemblés. Cadre vert : nécessitent et utilisent des données de séquençage appariées pour guider le *scaffolding*.

CAR (*Contig Assembly using Rearrangements*) L'outil CAR est uniquement applicable pour le *scaffolding* de génomes de procaryotes composés d'un seul chromosome. CAR utilise l'outil d'alignement MUMMER pour identifier et orienter les marqueurs homologues entre les contigs à assembler et le génome de référence, dont l'assemblage doit être complet. À partir de ces marqueurs homologues, CAR applique un algorithme basé sur les groupes de permutations algébriques [Li et al., 2013] qui consiste à déterminer un ensemble de fusions des contigs qui minimise la distance de réarrangement entre le génome à assembler et le génome de référence. À partir de cette fusion des contigs, on déduit l'ordre et l'orientation des contigs inférés.

CHROMOSOMER L'outil CHROMOSOMER [Tamazian et al., 2016] a été développé pour permettre le *scaffolding* de contigs de grands génomes eucaryotes à partir d'un génome de référence complet. La méthode utilise l'outil d'alignement global BLAST pour définir la localisation des contigs sur le génome de référence (*cf.* référence [Tamazian et al., 2016] pour l'approche de sélection des alignements BLAST). L'ordre et l'orientation relative des contigs sont déduits des alignements BLAST des contigs sélectionnés sur les chromosomes

de référence.

ALIGNGRAPH L'outil ALIGNGRAPH [Bao et al., 2014] requiert la présence de données de séquençage appariées pour pouvoir effectuer le *scaffolding* assisté par un génome de référence complet. Dans une première étape, les *reads* appariés sont alignés sur les contigs à ordonner et le génome de référence. Puis, les contigs sont alignés sur le génome de référence guidés par l'alignement des *reads* appariés sur les contigs et le génome de référence. L'alignement global des contigs sur le génome de référence est effectué avec BLAT et les *reads* sont localisés sur les contigs (à assembler) et le génome de référence avec l'outil de *mapping* BOWTIE2 [Langmead and Salzberg, 2012]. Dans un troisième temps, ALIGNGRAPH construit un graphe dérivé du graphe *de Bruijn* appelé *PE multipositional de Bruijn graph* [Bao et al., 2014] combinant le *PE de Bruijn graph* [Medvedev et al., 2011] et le *Positional de Bruijn graph* [Ronen et al., 2012] permettant d'associer dans un même graphe les informations de *reads* appariés et d'alignements des contigs sur le génome de référence. Le *PE multipositional de Bruijn graph* est utilisé pour étendre la séquence des contigs ordonnés sur le génome de référence grâce aux paires de *reads* alignées sur le génome de référence et localisées dans l'espace entre les contigs ordonnés et orientés sur le génome de référence.

MEDUSA (*Multi-Draft based Scaffolder*) L'approche implémentée dans MEDUSA [Bosi et al., 2015] consiste à établir un graphe de *scaffolding* à partir de l'alignement des contigs à assembler sur les génomes de référence considérés, pour lesquels l'assemblage complet du génome n'est pas nécessaire. Dans ce graphe, les sommets représentent les contigs du génome à assembler et les arêtes sont pondérées par un poids représentant le nombre de génomes de référence qui supportent cette adjacence. MEDUSA établit l'ordre des contigs avec un algorithme déterminant un chemin qui maximise le poids. Puis, la méthode applique une règle majoritaire pour déterminer l'orientation des contigs. L'algorithme MEDUSA a été développée dans l'optique de *scaffolding* de procaryotes (génomes de petites tailles).

RAGOUT (*Reference-Assisted Genome Ordering UTility*) L'approche implémentée dans l'outil RAGOUT [Kolmogorov et al., 2014, 2016] suit le protocole suivant :

1. détection de blocs synténiques [10] à plusieurs échelles avec l'outil SIBELIA [Minkin et al., 2013], adapté pour la construction de blocs synténiques à partir d'un alignement multiple produit par l'outil PROGRESSIVECACTUS [Paten et al., 2011] (*cf.* référence [Kolmogorov et al., 2014; Ghiurcuta and Moret, 2014] pour plus d'informations sur l'obtention de blocs synténiques à différentes échelles) ;
2. établissement d'un *incomplete multi-color breakpoint graph* [Kolmogorov et al., 2014] permettant de synthétiser dans un même graphe les informations d'adjacences des blocs synténiques homologues dans l'ensemble des génomes (où chaque génome est représenté par une couleur). Le génome cible étant fragmenté en contigs, les informations d'adjacences aux sommets du graphe correspondant aux extrémités des contigs du génome à assembler, sont manquantes ;
3. les adjacences manquantes sont inférées en résolvant le *half-breakpoint pasimony problem*, défini dans [Kolmogorov et al., 2014], en utilisant les adjacences des génomes de référence pondérées par la proximité phylogénétique de la référence à la cible grâce à la phylogénie des espèces donnée en entrée de RAGOUT.

RACA (*Reference-Assisted Chromosome Assembly*) Pour effectuer le *scaffolding* de contigs, la méthode RACA [Kim et al., 2013] nécessite un génome de référence, un ou plusieurs génomes *outgroup* et des données de séquençage appariées produites à partir du séquençage des contigs à assembler. Dans un premier temps, RACA effectue l'alignement deux à deux, avec l'outil LASTZ, des contigs avec le génome de référence et les génomes *outgroup*. Les alignements colinéaires obtenus sont fusionnés sous forme de fragments synténiques. Puis, pour chaque paire de fragments synténiques, RACA calcule un score d'adjacence, indiquant à quel point les deux fragments sont adjacents dans le génome à assembler. Ce score est calculé en combinant :

1. la probabilité *a posteriori* de la présence de cette adjacence dans le génome cible, déterminée à partir de la présence ou non d'adjacences homologues chez le génome de référence et les génomes *outgroup* où le poids des adjacences est pondéré par les relations phylogénétiques ;
2. la quantité de *reads* appariés en accord avec cette adjacence.

10. ensemble de marqueurs

À partir de l'alignement des fragments synténiques du génome cible et des scores d'adjacences calculés, RACA construit un graphe de fragments synténiques où les sommets correspondent aux extrémités de ces fragments. Un fragment synténique correspond à une arête localisée entre les deux sommets du graphe correspondant aux extrémités de ce fragment et les fragments synténiques sont liés par des arêtes correspondant aux adjacences pondérées par leurs scores d'adjacences. RACA utilise ensuite un algorithme glouton qui permet de résoudre les incohérences synténiques et permet de générer des chaînes de fragments synténiques. Dans une dernière étape, en utilisant l'ordre et l'orientation des fragments synténiques inférés dans les chaînes, RACA ordonne et oriente les contigs du génome cible auxquels les fragments synténiques appartiennent.

Conclusion sur les méthodes de *scaffolding* par génomique comparative Parmi les 16 méthodes de *scaffolding* assistées par un ou plusieurs génomes de référence, un grand nombre ont été développées pour assister l'assemblage de génomes de procaryotes. Ces méthodes seront, si elles ne le sont pas déjà, rendues obsolètes par les technologies de 3^{e} génération. Car bien que le coût de séquençage de génomes avec la technologie PacBio soit plus cher qu'avec la technologie Illumina [Goodwin et al., 2016], celui-ci permet désormais d'obtenir la structure et la séquence complète de génomes de petites tailles tels que les procaryotes [Koren and Phillippy, 2015] et ceci sans l'utilisation de méthode de *scaffolding* par génomique comparative. Parmi les 16 méthodes de la figure 2.6 (p. 40), trois se détachent des autres : RAGOUT, RACA et TREECAT, car elles effectuent le *scaffolding* de grands génomes et considèrent plusieurs génomes de référence avec leurs relations phylogénétiques au génome cible. Cela permet leur application à un large éventail de jeux de données et l'utilisation de signaux de synténies pondérés par la proximité phylogénétique du génome cible à un génome de référence. Une caractéristique intéressante de MEDUSA, mais absente chez les trois précédentes, consiste à pouvoir considérer des génomes de référence incomplètement assemblés. Cela est nécessaire pour certains domaines du vivant pour lesquels aucun génome de référence n'est disponible et en accord avec l'état des bases de données génomiques qui indiquent qu'une très grande majorité des assemblages de génomes ont le statut de *permanent draft* (*cf.* base de données GOLD : `https://gold.jgi.doe.gov/statistics`).

La méthode développée au cours de ce projet de thèse, décrite dans la partie II (p. 67) de ce manuscrit,combine les avantages de ces quatre méthodes : elle permet d'effectuer le *scaffolding*, dans un cadre phylogénétique, de l'ensemble des génomes [11] d'un jeu de données composé de grands génomes d'eucaryotes assemblés à divers degrés de complétude, et il est possible, mais pas nécessaire, d'utiliser des données de séquençage appariées pour guider le *scaffolding*.

11. innovation par rapport aux quatre méthodes précédentes

Chapitre 3

Reconstruction de génomes ancestraux et de leur évolution structurale

Dans ce chapitre, nous nous concentrons sur la reconstruction de l'organisation des génomes ancestraux en chromosomes, au sens de reconstruction de l'ordre et de l'orientation de marqueurs génomiques le long des chromosomes d'espèces ancestrales. Nous examinons différentes approches méthodologiques et logicielles, appliquées à un large éventail d'ensemble de données génomiques provenant de différents royaumes de la vie et à différentes profondeurs évolutives. Ce chapitre a pour but d'effectuer un état de l'art de ces approches méthodologiques afin de les mettre en perspective avec l'approche implémentée dans l'algorithme DeCo [1] [Bérard et al., 2012]. Nous focalisons notre attention sur l'algorithme DeCo car l'approche développée au cours de cette thèse se base sur cet algorithme. Nous illustrons également la similitude entre la problématique de reconstruction de l'ordre de marqueurs génomiques chez des génomes ancestraux et le problème de *scaffolding*. Cette similitude est le sujet principal de ma thèse qui a abouti à l'implémentation de deux algorithmes permettant conjointement de reconstruire l'organisation de génomes ancestraux. À la fin de ce chapitre, nous présenterons les avantages et inconvénients de nos méthodes par rapport aux deux seules autres méthodes existantes et qui ont été toutes deux été développées très récemment (au cours de cette thèse). Cela montre que ce domaine de recherche est très actif.

Dans la section 3.1, nous effectuons un historique des études sur les réarrangements chromosomiques. La section 3.2 a pour objectif de définir les

1. *Detection of Coevolution*

données d'entrée nécessaires et propres à l'ensemble des méthodes et algorithmes permettant la reconstruction de l'organisation de génomes ancestraux et de leurs histoires évolutives. La section 3.3 présente un panel représentatif des méthodes disponibles pour l'inférence de l'ordre et l'orientation de marqueurs génomiques d'espèces ancestrales à partir de marqueurs présents sur les assemblages de génomes actuels. Enfin dans la section 3.4, nous discutons des relations entre la reconstruction de l'organisation de génomes ancestraux et l'assemblage de génomes actuels descendants. Puis nous présentons deux méthodes permettant conjointement de reconstruire l'ordre de marqueurs génomiques d'espèces ancestrales et d'effectuer le *scaffolding* de génomes actuels.

Une part importante du contenu de ce chapitre provient du chapitre de livre *"Ancient Genome Reconstruction"* de *"Comparative Genomics : Methods and Protocols"* de la série *"Methods in Molecular Biology"* de l'éditeur scientifique *"Springer"*, auquel nous avons participé à la rédaction et qui devrait paraître en 2018 [Anselmetti et al., 2018].

3.1 Historique des études sur les réarrangements chromosomiques

Les réarrangements ont été les premières mutations du génome découvertes [Sturtevant, 1921], bien avant la découverte de la structure moléculaire de l'ADN. Des études d'évolution moléculaire ont débuté avec la reconstruction de l'organisation des chromosomes de génomes ancestraux du genre *Drosophila*, à partir de la comparaison des génomes actuels [Sturtevant and Dobzhansky, 1936; Dobzhansky and Sturtevant, 1938]. Cependant, il a fallu près de 30 ans de plus avant la publication de la première étude sur la reconstruction de gènes ancestraux [Pauling and Zuckerkandl, 1963]. Le développement des technologies de séquençage haut-débit et des méthodes d'assemblage au cours des années 2000 a permis l'accès à un grand nombre de génomes séquencés plus ou moins bien assemblés (*cf.* chapitre 2). La disponibilité croissante de génomes séquencés a permis, au cours des quinze dernières années, le développement de méthodes reconstruisant l'organisation des génomes ancestraux à partir de la séquence génomique d'espèces descendantes [Muffato and Roest Crollius, 2008].

Alors que l'évolution par mutations nucléotidiques (substitutions, insertions, délétions) a été largement étudiée à partir des années 1960, l'évolution

par des événements évolutifs à l'échelle du génome, comme les réarrangements chromosomiques, est à l'heure actuelle faiblement étudiée comparée à l'évolution par mutations ponctuelles des séquences. Deux raisons peuvent être invoquées :

1. les études de réarrangement nécessitent d'avoir des génomes dont l'assemblage de la séquence d'ADN est entièrement résolu, or nous avons vu dans le chapitre 2 que l'assemblage du génome reste un problème difficile, ce qui entraîne un faible nombre de génomes complets disponibles dans les bases de données (*cf.* base de données GOLD [Mukherjee et al., 2017]) ;
2. l'espace d'états de l'évolution de la séquence génomique d'une espèce est très petit (4 nucléotides ou 20 acides aminés possibles par locus ancestral), ce qui entraîne des problèmes algorithmiques plus faciles que ceux des réarrangements génomiques, qui fonctionnent sur l'espace discret potentiellement infini d'éventuelles organisations chromosomiques (ordre de gènes par exemple).

Cependant, aucune de ces raisons n'est biologique, et de récents progrès technologiques et méthodologiques sont susceptibles de changer rapidement cette situation.

Dans la suite de ce chapitre, nous examinons une partie des méthodes de reconstruction de l'ordre de marqueurs génomiques chez des génomes d'espèces ancestrales, en se concentrant sur leurs principes méthodologiques, leurs forces et leurs faiblesses. Nous détaillons également les étapes de prétraitement des données qui sont nécessaires pour les utiliser.

3.2 Matériel génomique

Pour mener des études reconstruisant l'histoire évolutive de l'organisation des génomes, il est nécessaire d'avoir accès ou de générer les séquences et assemblages de génomes d'espèces actuelles (*cf.* chapitre 2). Ceux-ci sont souvent disponibles dans des bases de données publiques telles que les navigateurs génomiques Ensembl [Aken et al., 2017], UCSC [Tyner et al., 2017] et la base de données VectorBase [Giraldo-Calderón et al., 2015]. En fonction de la combinaison des propriétés des génomes séquencés (zones répétées en particulier), de la technologie de séquençage et de l'algorithme d'assemblage utilisés, les séquences assemblées peuvent être à différents niveaux d'achèvement :

— génome complet : la séquence et la structure des chromosomes sont complètes ;
— génome fragmenté : la séquence des chromosomes est fragmentée sous forme de contigs ou de *scaffolds*.

La fragmentation des assemblages de génomes actuels a un impact significatif sur la qualité des génomes ancestraux reconstruits, nous en discuterons dans la section 3.4 (p. 61).

Pour reconstituer l'organisation de génomes ancestraux à partir de la comparaison de génomes actuels, il faut d'abord définir un ensemble de marqueurs pour chaque génome actuel considéré, c.-à-d. des segments d'ADN définis par leurs coordonnées sur les génomes (chromosome, *scaffold* ou contig, position de départ, position de fin, sens de lecture). Les marqueurs sont regroupés en familles où deux marqueurs appartenant à la même famille sont homologues sur toute leur longueur.

Les familles de gènes, disponibles dans certaines bases de données [Aken et al., 2017; Penel et al., 2009; Waterhouse et al., 2013], sont de très bons candidats pour servir de marqueurs, bien que le chevauchement et le partage d'homologies partielles entre les gènes puissent poser problème à certaines méthodes.

Les marqueurs peuvent également être obtenus en construisant des blocs synténiques à partir d'alignements multiples de génomes entiers [Sankoff and Nadeau, 2003], en segmentant les génomes à partir d'alignements deux à deux [Višňovská et al., 2013], en recherchant des éléments ultra-conservés (*UCE*) [Dousse et al., 2016] ou des sondes virtuelles [Belcaid et al., 2007]. Ces méthodes sont utiles lorsqu'on considère des génomes avec une faible densité en gènes pour lesquels l'utilisation de familles de gènes est trop restrictive pour reconstruire l'histoire évolutive de l'organisation de génomes ancestraux.

Finalement, une approche comparative nécessite une information phylogénétique reliant une ou plusieurs espèces ancestrales d'intérêt à un ensemble d'espèces actuelles dont les données génomiques sont disponibles.

3.3 Méthodes et algorithmes pour la reconstruction de l'organisation de génomes ancestraux

Toutes les méthodes considèrent un génome comme un ensemble d'ordres circulaires ou linéaires de marqueurs, représentant des chromosomes ou des segments chromosomiques (contigs ou *scaffolds*). Cela implique que les coordonnées physiques exactes des marqueurs se transforment en ordre relatif des marqueurs et induit une perte d'information qui peut influencer la reconstruction de l'organisation de génomes ancestraux [Biller et al., 2016]. Les méthodes diffèrent dans leurs stratégies :

— soit elles modélisent l'évolution des réarrangements de marqueurs par des événements évolutifs tels que des duplications, des pertes, des réarrangements ;
— soit elles modélisent l'évolution de caractères synténiques plus locaux tels que la proximité physique des ensembles de marqueurs, comme les adjacences (*cf.* section 1.1.2, p. 15) ou des intervalles de marqueurs.

Nous appelons *intervalle* un ensemble d'au moins trois marqueurs contigus le long d'un génome actuel ou ancestral.

La première stratégie (évolution de génomes entiers) conduit rapidement à des problèmes de complexité algorithmique. La deuxième stratégie (évolution de caractères synténiques locaux) bénéficie d'une boîte à outils évolutive standard qui modélise l'évolution de la présence ou de l'absence d'un caractère et les problèmes de complexité sont reportés à l'étape finale de linéarisation. Les procédures de linéarisation bénéficient d'algorithmes issus du calcul de cartes physiques de génomes actuels [Alizadeh et al., 1995].

3.3.1 Évolution de génomes entiers

Nous décrivons d'abord l'approche qui considère l'évolution des génomes comme des ensembles d'ordres linéaires ou circulaires de marqueurs orientés. Ces ensembles peuvent être assimilés à des permutations. Cette approche consiste en la détection d'événements évolutifs modifiant l'organisation, d'un génome, la permutation, pouvant impliquer un ou plusieurs chromosomes :

— inversions ;
— translocations ;
— transpositions ;
— fissions ;
— fusions.

L'ensemble de ces événements évolutifs est inclus dans le modèle DOUBLE-CUT-AND-JOIN (DCJ) [Yancopoulos et al., 2005].

La reconstruction des génomes ancestraux consiste à inférer l'ordre des marqueurs pour tous les nœuds ancestraux, à partir de l'ordre des marqueurs représentant les génomes actuels de la phylogénie d'espèces, en maximisant un critère mathématique selon le modèle évolutif choisi. La plupart du temps, ce critère est un score de parcimonie, qui correspond au nombre minimal d'événements transformant une permutation en une autre [Fertin et al., 2009], également appelé distance. D'autres méthodes ont une approche consistant à maximiser un critère de vraisemblance.

Pour la plupart des modèles de réarrangements qui n'incluent pas les événements de duplications, la distance entre deux génomes peut être calculée efficacement. Cependant, même le problème de reconstruction du génome ancestral le plus simple, le *problème de la médiane*, qui consiste à reconstruire un génome ancestral minimisant la distance d'un arbre composé de trois espèces actuelles, est déjà *NP-difficile* [Tannier et al., 2009]. L'ajout de duplications rend tous les problèmes difficiles même pour la comparaison de deux génomes [Fertin et al., 2009] et rend la reconstruction d'événements de réarrangement le long d'une phylogénie des espèces insoluble. Les heuristiques pour le problème de reconstruction de génomes ancestraux suivent généralement la stratégie consistant à définir une permutation initiale à l'ensemble des nœuds internes de l'arbre, correspondant aux génomes ancestraux, puis à raffiner itérativement la solution en résolvant le problème de la médiane pour l'ensemble des nœuds internes jusqu'à ce que l'on ne puisse obtenir aucune amélioration supplémentaire de la distance globale de l'arbre.

Nous détaillons dans la suite un panel de méthodes représentatives implémentant l'approche de reconstruction de l'organisation de génomes sous forme de permutations.

GASTS (*Generalized Adequate Subtree Tree Scoring*) L'approche de GASTS [Xu and Moret, 2011] améliore cette stratégie en essayant de trouver une permutation initiale permettant d'éviter l'optimisation locale. En utilisant des sous-graphes adéquats pour l'affectation heuristique de la médiane, cette méthode peut gérer des données multi-chromosomiques avec des marqueurs uniques et universels, c.-à-d. présents dans l'ensemble des génomes en une seule et unique copie.

PATHGROUPS Une autre approche, basée sur la structure des données, PATHGROUPS [Zheng, 2010; Zheng and Sankoff, 2011], consiste à stocker des cycles partiellement achevés dans un graphe de point de cassures (plus communément appelée *breakpoint graph* [Bafna and Pevzner, 1996; Fertin et al., 2009]) pour chaque branche de la phylogénie. Les graphes sont complétés, par une approche gloutonne, et finissent par former des génomes à tous les nœuds internes. Cette solution peut être utilisée comme une initialisation avant les améliorations itératives locales basées sur la résolution du problème de la médiane à nouveau en utilisant l'approche PATHGROUPS. Une propriété intéressante de PATHGROUPS est qu'elle permet de considérer des duplications de génomes entiers.

MGRA (*Multiple Genome Rearrangements and Ancestors*) La méthode MGRA [Alekseyev and Pevzner, 2009] repose sur un *multiple breakpoint graph* combinant les permutations de l'ensemble des génomes actuels dans une seule structure. L'approche MGRA recherche les cassures en accord avec la structure de l'arbre des espèces transformant le *breakpoint graph* de l'ensemble des permutations de génomes actuels en un *breakpoint graph* d'identité. Bien que MGRA exige des marqueurs uniques et universels, elle a récemment été étendue permettant de considérer un contenu inégal en marqueurs chez les génomes actuels considérés [Avdeyev et al., 2016]. Des modèles d'évolution plus complexes ont été envisagés, qui permettent de considérer les duplications [Ma et al., 2008; Paten et al., 2014], mais ne peuvent être appliqués que dans certaines conditions spécifiques, comme la non réutilisation des points de cassures [Ma et al., 2008].

SCJ (SINGLE-CUT-OR-JOIN) Enfin, une distance de réarrangements plus simple est la distance SINGLE-CUT-OR-JOIN (SCJ) [Feijão and Meidanis, 2011] qui modélise des événements de créations et cassures d'adjacences. Les génomes ancestraux qui minimisent la distance SCJ peuvent être calculés en temps polynomial à l'aide d'une variante de l'algorithme de Fitch [Fitch, 1971]. Cependant, les contraintes requises pour s'assurer que les ordres de marqueurs ancestraux restent linéaires ou circulaires aboutissent à la reconstruction de génomes ancestraux pour la plupart fragmentés.

3.3.2 Génomes comme des ensembles d'adjacences ou d'intervalles

Les ordres linéaires ou circulaires de marqueurs peuvent être considérés comme des ensembles d'adjacences ou d'intervalles. Chaque adjacence ou intervalle peut être considéré indépendamment, en tant qu'unité synténique indépendante, qui évolue dans le contexte plus large de génomes entiers. Cette propriété d'indépendance permet de calculer efficacement les états ancestraux des adjacences ou intervalles mais ne permet pas pour l'ensemble des adjacences ou intervalles ancestraux de garantir la compatibilité avec un ordre linéaire ou circulaire. Nous décrivons ici une famille d'approches qui se concentrent sur un seul génome ancestral et se composent de deux étapes principales, inspirées des méthodes initialement développées pour établir des cartes physiques de génomes :

1. les génomes d'espèces actuelles apparentées sont comparés pour détecter les synténies locales communes, telles que les adjacences ou intervalles, qui sont alors considérées comme des synténies candidates pour le génome ancestral que l'on souhaite reconstruire. Cependant, les synténies communes ne sont pas nécessairement héritées d'un ancêtre et peuvent provenir d'une évolution convergente ou d'erreurs d'assemblage, ces méthodes génèrent donc des faux positifs. Dans certaines méthodes, chaque synténie locale est pondérée, selon son ratio présence/absence dans les génomes des espèces actuelles, afin d'obtenir un score de confiance ;
2. un sous-ensemble de poids maximal des synténies locales potentiellement ancestrales (détectées dans la première étape) est sélectionné afin d'être compatible avec la structure du génome des espèces ancestrales considérées (chromosomes linéaires/circulaires, nombre de marqueurs ancestraux, ...) et est ensuite assemblé dans une carte détaillée du génome ancestral.

Le cas de marqueurs uniques

Les premières applications [Ma et al., 2006; Chauve and Tannier, 2008] de ces principes de cartographie physique à la reconstruction de l'organisation de génomes ancestraux ont été effectuées pour des marqueurs uniques.
Dans plusieurs méthodes [Chauve and Tannier, 2008; Chauve et al., 2010; Ouangraoua et al., 2011; Jones et al., 2012; Hu et al., 2014b] la détection des

adjacences et des intervalles communs et l'inférence des adjacences et des intervalles ancestraux, est implémentée selon un principe de parcimonie de Dollo : tout groupe de marqueurs co-localisés sur les génomes de deux espèces actuelles dont le chemin évolutif dans la phylogénie des espèces contient les espèces ancestrales d'intérêt est considéré comme une synténie ancestrale potentielle. Par co-localisation, nous voulons dire que le groupe de marqueurs se localise de manière contiguë dans les deux génomes actuels, indépendamment de leurs ordres relatifs, mais sans autre marqueur dans l'intervalle. De telle sorte que le contenu des deux occurrences du groupe co-localisé de marqueurs dans les génomes actuels est identique tandis que leur ordre peut être différent. Les groupes composés de deux marqueurs sont des adjacences, tandis que les groupes de plus de deux marqueurs sont des intervalles. On peut envisager des variations sur le principe décrit ci-dessus, par exemple en relâchant le critère de parcimonie de Dollo ou en considérant seulement des adjacences [Ma et al., 2006] ou bien en considérant l'inférence probabiliste d'adjacences ancestrales [Hu et al., 2014b; Ma, 2010].

Compte tenu d'un ensemble de groupes locaux de synténies ancestrales, ces méthodes sélectionnent un sous-ensemble de poids maximal de ces groupes qui soit compatible avec la structure du génome considéré et ne contient aucun conflit synténique, défini comme un marqueur adjacent à plus de deux autres marqueurs. Plusieurs méthodes telles que INFERCARS [Ma et al., 2006] et MLGO [Hu et al., 2014b,a] ne considèrent que les adjacences de marqueurs (et pas les intervalles de marqueurs). Ces adjacences définissent un graphe dont les sommets sont des marqueurs et les arêtes représentent des adjacences pondérées. Il faut alors calculer un ensemble maximal d'adjacences pondérées qui forment un ensemble de chemins, chacun de ces chemins étant alors un ordre linéaire de marqueurs appelé *Contiguous Ancestral Reconstruction* (*CAR*). Ce problème équivaut au problème du voyageur de commerce qui est NP-difficile. Il est abordé dans [Ma et al., 2006] à travers une heuristique gloutonne et dans [Hu et al., 2014b] en utilisant un solveur standard du problème du voyageur de commerce. Cependant, comme le montre [Maňuch et al., 2012b], si la linéarité des *CARs* est relâchée et que les *CARs* circulaires sont autorisés, le problème d'optimisation de sélectionner un sous-ensemble de poids maximal d'adjacences qui forme un mélange de *CARs* linéaires et circulaires est résoluble par réduction à un problème de couplage de poids maximal (*Maximum-Weight Matching*).

Lorsque des intervalles sont considérés en plus des adjacences, les adjacences et intervalles ancestraux peuvent être encodés dans une matrice binaire, de

la même manière que les expériences d'hybridations sont codées par des matrices binaires dans les algorithmes de cartographie physique. Le problème de l'extraction d'un sous-ensemble de poids maximal sans conflit est alors NP-difficile dans tous les cas, c.-à-d. même si un mélange de *CARs* circulaires et linéaires est autorisé. Traditionnellement, il est résolu à l'aide d'heuristiques gloutonnes ou d'algorithmes par séparation et évaluation, également appelés algorithmes *branch-and-bound* (assurant une solution optimale lorsqu'ils se terminent). De plus, lorsque des intervalles sont pris en considération, les *CARs* peuvent ne pas être complètement définis et sont représentés à l'aide d'une structure de données, les *PQ-tree* [Booth and Lueker, 1976], largement utilisée dans les algorithmes de cartographie physique [Alizadeh et al., 1995]. La structure de données *PQ-tree* est liée au concept combinatoire classique de *C1P* [2] [Booth and Lueker, 1976; Meidanis et al., 1998; Chauve and Tannier, 2008]. Les logiciels ANGES [Jones et al., 2012] et ROCOCO [Stoye and Wittler, 2009] sont, jusqu'à présent, les seules méthodes de reconstruction génomique ancestrale qui tiennent compte des intervalles de marqueurs et infèrent les *CARs* en utilisant des *PQ-trees*.

Enfin, lorsque les marqueurs sont supposés être uniques dans le génome ancestral d'intérêt, mais sont soumis à une insertion ou à une délétion pendant l'évolution, le modèle d'adjacences et d'intervalles communs peut être trop restrictifs. Dans ce cas, les notions de *gapped* adjacences et intervalles [3] ont été introduites, ce qui permet une certaine souplesse dans la définition du groupe conservé de marqueurs. Cependant, cela implique également que le modèle *C1P* est trop rigoureux et doit être relâché dans un modèle *gapped C1P* [Chauve et al., 2009], dans lequel les problèmes d'optimisation sont NP-difficiles [Maňuch and Patterson, 2011; Maňuch et al., 2012a].

Ces approches ont été utilisées sur différents ensembles de données, les génomes de mammifères [Ma et al., 2006; Chauve and Tannier, 2008; Gavranović et al., 2011], l'ancêtre des amniotes [Ouangraoua et al., 2011], les génomes de champignons [Chauve et al., 2010], les génomes d'insectes [Semeria et al., 2015] ou encore les génomes de plantes [Murat et al., 2010, 2015].

2. Propriété des 1 Consécutifs.
3. pour lesquels des marqueurs n'appartenant pas à l'adjacence, resp. l'intervalle, se situent entre une paire de marqueurs adjacents, resp. au sein d'un intervalle de marqueurs.

Le cas des contenus en marqueurs différents et des marqueurs en copies multiples

Si les marqueurs présentent un nombre de copies variable dans les génomes actuels, on ne peut pas supposer que tous n'apparaissent qu'une seule fois dans le génome ancestral d'intérêt. Le premier problème est alors, pour un marqueur donné, de déduire son nombre de copies ancestrales. Il s'agit d'un problème classique de génomique évolutive, par exemple pour inférer le contenu génétique d'un génome éteint. Compte tenu d'un modèle de gains et de pertes de marqueurs, il est possible de calculer le contenu ancestral le plus probable [De Bie et al., 2006; Csurös, 2010] ou un contenu qui minimise le nombre de gains et de pertes [Csűrös, 2013], par un algorithme de programmation dynamique suivant le motif général de l'algorithme Sankoff-Rousseau [Sankoff and Rousseau, 1975].

Une fois que le nombre de copies des marqueurs ancestraux (ou des bornes sur ce nombre) a été obtenu, l'approche en deux étapes décrite dans les paragraphes précédents peut être appliquée :

1. les synténies locales sont détectées en utilisant des notions similaires d'adjacences et d'intervalles [4] et sont pondérés en fonction de leur modèle de conservation ;
2. un sous-ensemble de poids maximal des synténies locales compatible avec les numéros de copie du marqueur est calculé.

Le problème du point 2 est connu sous le nom de *C1P* avec multiplicité (*mC1P*) et s'est révélé NP-difficile en général. Le seul cas résoluble nécessite de considérer uniquement les adjacences et de permettre un nombre illimité de *CARs* circulaires [Maňuch et al., 2012b; Wittler et al., 2011]. En outre, lorsque les marqueurs ont un nombre de copies supérieur à un et que seules les adjacences sont considérées, un ensemble d'adjacences sans conflit ne définit pas de manière non équivoque un ensemble de *CARs*. Ce problème est similaire au problème bien identifié de déterminer l'emplacement et le contexte des répétitions dans l'assemblage du génome [Treangen and Salzberg, 2012]. Ce problème peut être abordé, au moins partiellement, en considérant les intervalles encadrés par des non-répétitions (intervalles répétés) comme décrit dans [Rajaraman et al., 2013, 2016]. Enfin, lorsque la variation du nombre de copies peut être attribuée aux duplications de génomes entiers, des méthodes spécifiques basées sur une combinaison d'adjacences *gapped* et

4. nous référons le lecteur à [Wittler et al., 2011] pour un aperçu des modèles d'intervalles lorsque des marqueurs dupliqués existent.

d'algorithmes du problème de voyageur de commerce ont été proposées et appliquées aux données sur les champignons et les plantes [Gagnon et al., 2012].

3.3.3 Évolution d'adjacences le long de phylogénies de gènes

Nous discutons maintenant d'une variante de l'approche décrite dans la section précédente qui considère les génomes comme des ensembles d'adjacences de gènes. Celle-ci requiert l'histoire évolutive des familles de gènes considérés et n'est plus limitée à la reconstruction de l'organisation d'un génome ancestral mais à l'ensemble des génomes ancestraux disponibles dans la phylogénie des espèces considérées. Dans cette approche, les adjacences ancestrales sont prédites comme dans la section précédente, en utilisant un critère d'optimisation prenant en compte l'histoire évolutive de chaque famille de gènes qui sont utilisées à la fois comme guides et contraintes évolutives lors de l'inférence d'adjacences ancestrales.

Entrée : phylogénies et adjacences de gènes

Cette approche basée sur la phylogénie requiert une phylogénie d'espèces enracinée entièrement binaire, les arbres réconciliés (*cf.* section 1.3.3, p. 21) de l'ensemble des familles de gènes considérés ainsi que la liste des adjacences de gènes présentes dans les génomes actuels. Certaines bases de données fournissent des arbres de gènes (réconciliés ou non), Vectorbase pour des espèces d'arthropodes vectrices de maladies [Giraldo-Calderón et al., 2015], Ensembl pour des espèces de vertébrés [Aken et al., 2017] et HOGENOM pour des espèces de l'ensemble du vivant [Penel et al., 2009].

Évolution de l'adjacence

L'élément central des méthodes basées sur la phylogénie est de déduire l'évolution d'adjacences le long des phylogénies de gènes, qui évoluent elles-mêmes dans la phylogénie des espèces. Ceci conduit à l'inférence d'adjacences entre des gènes ancestraux, à savoir des adjacences ancestrales, permettant de déterminer l'ordre et l'orientation des gènes le long du génome des espèces ancestrales.

Les méthodes actuellement disponibles calculent une histoire évolutive des adjacences en minimisant un critère discret de parcimonie ou en maximisant un critère de vraisemblance dans un cadre probabiliste. La difficulté

principale de telles méthodes est de déduire les scénarios d'évolution des adjacences qui sont compatibles avec l'histoire évolutive des gènes considérés, encodés dans leurs arbres de gènes réconciliés respectifs.

Le résultat de cette approche, qui considère chaque adjacence indépendamment les unes des autres (comme dans les méthodes décrites dans la section 3.3.2, p. 52, et contrairement à celles de la section 3.3.1, p. 49), est un ensemble d'adjacences de gènes prédites pour chaque espèce ancestrale. Comme il n'est pas garanti que ces adjacences soient compatibles avec une structure linéaire, des méthodes de linéarisation telle que [Maňuch et al., 2012b] ou des méthodes d'évolution globale telle que [Luhmann et al., 2016] peuvent être appliquées pour inférer des arrangements de gènes ancestraux compatibles avec un ordre linéaire, pour chaque génome ancestral.

Dans la section suivante, nous décrivons en détails le fonctionnement de l'algorithme DeCo car les méthodes développées au cours de cette thèse sont basées sur le principe de cet algorithme.

DeCo (*Detection of Coevolution*)

DeCo est un algorithme qui permet de reconstruire l'histoire évolutive d'adjacences ancestrales de gènes en minimisant le nombre de gains et de cassures d'adjacences. Cet algorithme s'exécute en temps polynomial, il est basé sur une *approche parcimonieuse* reprenant le principe de *programmation dynamique* de Sankoff-Fitch [Fitch, 1971; Sankoff, 1975].
L'algorithme DeCo est implémenté dans un programme du même nom qui peut être considéré comme une méthode de reconstruction de génomes ancestraux car elle permet de reconstruire des adjacences de gènes ancestrales en plus de leur histoire évolutive le long de la phylogénie des espèces considérées. La combinaison d'adjacences ancestrales permet de reconstruire l'ordre relatif de gènes sur les génomes ancestraux.

Fonctionnement général du logiciel DeCo

Le logiciel DeCo prend en entrée une *liste d'adjacences de gènes actuels* notée L_{adj}, un ensemble d'*arbres de gènes* noté E_G et un *arbre des espèces* S. La méthode DeCo est composée de deux phases :

— une première *phase de prétraitement* qui *réconcilie parcimonieusement* les arbres de gènes avec l'arbre des espèces puis *regroupe les adjacences actuelles* en *classes d'équivalence*.

— une deuxième phase qui pour chaque classe d'équivalence *infère une forêt d'arbres d'adjacences* (*cf.* section 1.3.3, p. 21). Pour cela, DeCo *calcule d'abord une matrice de scores* entre les couples de nœuds/gènes provenant de la même espèce en optimisant un critère de coût ; puis à l'aide d'une procédure de *backtracking*, il reconstruit une *forêt d'arbres d'adjacences* à partir de cette matrice de scores.

Phase 1 : prétraitement des données

a/ Réconciliation parcimonieuse :

DeCo effectue la réconciliation parcimonieuse de l'ensemble des arbres de gènes contenus dans E_G avec S sous un modèle de duplication-perte de gènes (*cf.* section 1.3.3, p. 21). L'ensemble des arbres de gènes réconciliés est noté E_{Grec} . Une fois l'attribution aux arbres de gènes des événements de spéciations, duplications et pertes de gènes effectuée, DeCo crée des classes d'équivalences d'adjacences à partir de E_{Grec} et de L_{adj}.

b/ Classes d'adjacences :

Une *classe d'équivalence d'adjacences* (ou *famille d'adjacences homologues*) permet de regrouper les adjacences qui partagent potentiellement une adjacence ancestrale commune. Par définition, les adjacences d'une même classe d'équivalence appartiennent au plus à deux arbres de gènes réconciliés.

Deux adjacences $a_1 \sim a_2$ et $b_1 \sim b_2$, respectivement chez les espèces A et B, présentes sur deux arbres de gènes AG_1 et AG_2 (*cf.* partie gauche de la figure 3.1), appartiennent à la même classe si :

— a_1 et b_1 appartiennent au même arbre de gènes, AG_1 ;
— a_2 et b_2 appartiennent au même arbre de gènes, AG_2 ;
— l'espèce C est l'ancêtre commun de A et B ;
— le gène c_1, de l'espèce C, est le plus récent ancêtre de a_1 et b_1 ;
— le gène c_2, de l'espèce C, est le plus récent ancêtre de a_2 et b_2.

Si elles n'appartiennent qu'à un seul arbre de gènes noté AG, une propriété définie dans l'article décrivant l'algorithme DeCo [Bérard et al., 2012], établit que toutes les adjacences de cette classe sont contenues dans deux sous-arbres distincts de AG (*cf.* partie droite de la figure 3.1). Formellement, cela revient à dire que si a_1 et b_1 appartiennent à un même arbre de gènes AG et que $a_1 \sim a_2$ et $b_1 \sim b_2$ sont dans la même classe d'adjacences, alors le plus petit ancêtre commun de a_1, a_2 et b_1, b_2 est le même et on le note $LCA(a_1, a_2) =$

$LCA(b_1, b_2) = c_3$. Ainsi, lorsque que l'on retire c_3 de l'arbre de gène, on obtient deux sous-arbres de AG notés AG' et AG'' appartenant à une même classe d'adjacences et qui seront traités simultanément par l'algorithme DECO.

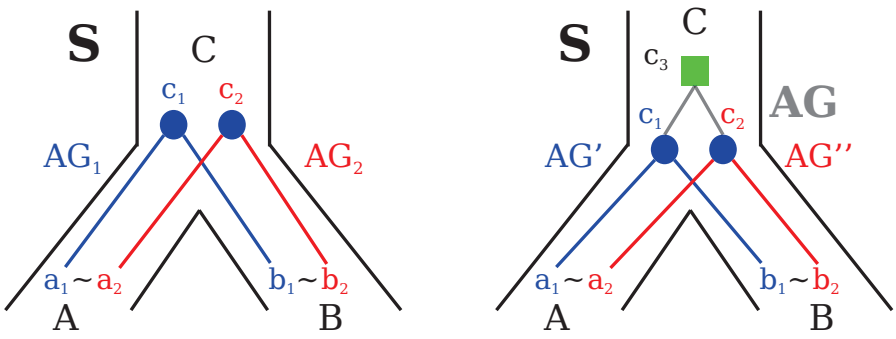


FIGURE 3.1 – Illustration des classes d'équivalence d'adjacences. À gauche, la représentation de deux arbres de gènes AG_1 et AG_2 d'une même classe d'adjacences dans un arbre d'espèces S et à droite la représentation d'un arbre de gène AG, pouvant être décomposé en deux-sous arbres AG' et AG'' appartenant à la même classe d'adjacences, dans un arbre des espèces S. (*cf.* table 1.1 pour la signification des symboles évolutifs.)

Phase 2 : algorithme DECO

Pour chaque classe d'équivalence, DECO calcule une matrice de scores. On considère les couples de nœuds/gènes (g_1,g_2) des arbres de gènes AG_1 et AG_2 appartenant à une même classe, tels que $g_1 \in AG_1$ et $g_2 \in AG_2$, où l'espèce de g_1 est la même que celle de g_2. Deux coûts sont calculés :

— $c_1(g_1, g_2)$: coût minimum d'une histoire évolutive où g_1 et g_2 sont adjacents ;

— $c_0(g_1, g_2)$: coût minimum d'une histoire évolutive où g_1 et g_2 ne sont pas adjacents ;

a/ Calcul d'une matrice de scores :

Cette étape est basée sur une approche parcimonieuse de programmation dynamique reprenant le principe de l'algorithme de Sankoff-Fitch [Fitch, 1971; Sankoff, 1975] [5]. Elle permet de calculer une matrice de coût minimum entre tous les couples de nœuds (g_1,g_2). Chaque nœud g des arbres étant étiqueté d'un événement évolutif noté $E(g)$, le calcul du coût de chaque couple de nœuds ne sera pas le même selon les paires d'événements. Dans les arbres de gènes réconciliés considérés par DECO, il existe quatre types d'événements évolutifs :

5. la programmation dynamique consiste à résoudre un problème complexe en résolvant et en stockant le résultat des sous-problèmes de même nature que celui-ci.

— gène actuel **Act** ;
— spéciation ● ;
— duplication de gène ■ ;
— perte de gène ✖.

Pour calculer les coûts c_1 et c_0, six cas de formules de récurrence correspondant aux différents couples d'événements évolutifs possibles sont détaillés dans l'article [Bérard et al., 2012] et disponibles en annexe 179. Cette étape est *ascendante* et calcule récursivement les scores des coûts c_1 et c_0 en partant des couples de feuilles jusqu'au couple de racines. La figure 3.2 illustre le calcul de la matrice de coût entre les deux arbres AG_1 et AG_2 provenant de la même classe d'équivalence d'adjacences.

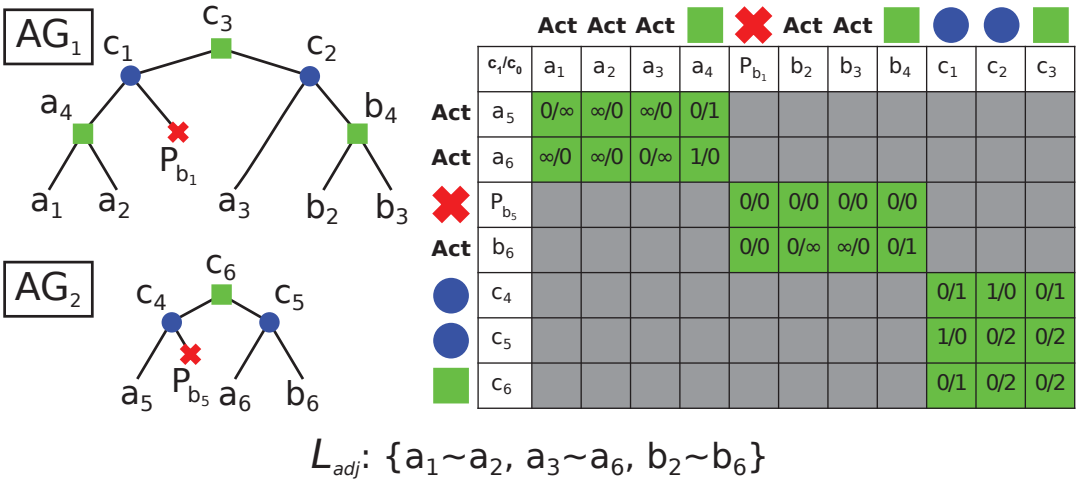


FIGURE 3.2 – Représentation du calcul de la matrice de coûts minimaux à partir de deux arbres de gènes AG_1 et AG_2. Sur la gauche, représentation graphique de AG_1 et AG_2 et la liste des adjacences entre les gènes actuels. Sur la droite, représentation de la matrice de coûts minimaux entre tous les couples de nœuds de AG_1 et AG_2 de même espèce. L'algorithme DECO a une complexité quadratique, c.-à-d. en $O(n^2)$ où n représente le nombre de nœuds dans un arbre de gènes.

b/ Étape de backtracking :

Une fois la matrice de coûts des couples de nœuds calculée, DECO effectue une étape de *backtracking* permettant, à partir des résultats de cette matrice, de reconstruire une forêt d'arbre(s) d'adjacences modélisant l'histoire évolutive des adjacences contenues dans les arbres de gènes AG_1 et AG_2. Pour recréer l'histoire évolutive des adjacences ancestrales depuis la matrice de scores entre couples de nœuds, l'étape de *backtracking* commence sur la case de la matrice correspondant au $min(c_1(R_1, R_2), c_0(R_1, R_2))$ où R_1 (resp.

R_2) correspond à la racine de l'arbre de gènes AG_1 (resp. AG_2). La phase de *backtracking* est un *processus descendant* qui choisit les adjacences en suivant à rebours les combinaisons de scores ayant amené aux scores minimaux pour les couples de nœuds considérés, jusqu'à atteindre les adjacences actuelles entre feuilles des arbres de gènes. La figure 3.3 illustre l'étape de *backtracking* correspondant à la reconstruction de l'arbre d'adjacences à partir des arbres de gènes AG_1 et AG_2 de la figure 3.2.

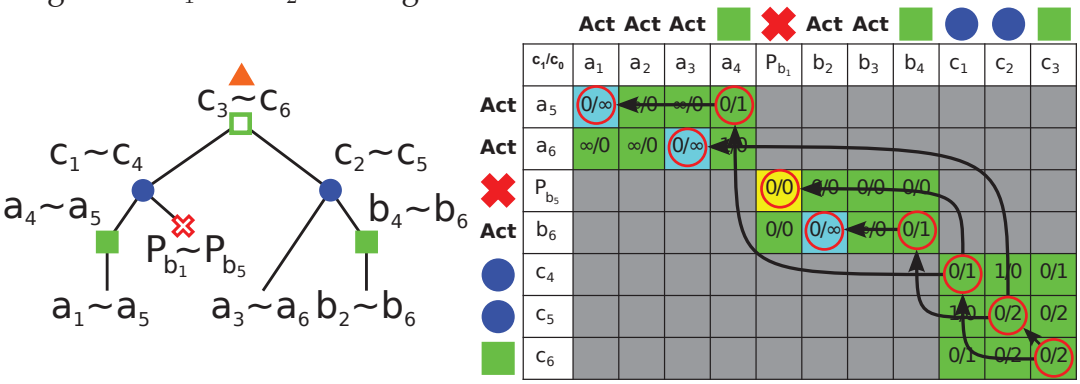


FIGURE 3.3 – Représentation de la reconstruction d'un arbre d'adjacences à partir de la matrice de coûts. Sur la gauche, représentation graphique de l'arbre d'adjacences généré à partir des arbres de gènes AG_1 et AG_2. Sur la droite, les flèches représentent le fonctionnement de l'étape de *backtracking* sous la forme d'un parcours arboré permettant de reconstruire l'arbre d'adjacences. Les cases en bleu correspondent aux adjacences actuelles et la case en jaune correspond à une adjacence qui a été perdue au cours de l'histoire évolutive suite à la perte des deux gènes la composant.

3.4 Reconstruction conjointe de l'organisation de génomes ancestraux et actuels

Idéalement, pour reconstruire une organisation précise et complète d'un (ou plusieurs) génomes ancestraux avec une approche comparative, on voudrait pouvoir compter sur l'organisation chromosomique complète des génomes actuels apparentés considérés. Cependant, comme nous l'avons vu dans le chapitre 2, l'assemblage des génomes est incomplet pour les grands génomes d'eucaryotes, se traduisant par le fait que les chromosomes sont divisés en plusieurs contigs ou *scaffolds* dont l'ordre relatif et l'orientation ne sont pas résolus. Cette information manquante sur l'ordre et l'orientation de ces *scaffolds* entraîne une reconstruction fragmentée de l'organisation génomique ancestrale.

On peut voir le problème de la reconstruction de l'organisation des génomes ancestraux comme similaire à la cartographie du génome ou au problème de *scaffolding*, auquel cas la reconstruction génomique ancestrale et l'assemblage de génome actuel peuvent être considérés comme un problème unique qui consiste à ordonner et orienter des marqueurs génomiques ancestraux et actuels. La similitude algorithmique entre ces deux problèmes a été décrite dans [Lin et al., 2014] qui démontre une similitude entre le *breakpoint graph*, utilisé pour la reconstruction de l'ordre de gènes de génomes ancestraux et le graphe de *de Bruijn*, utilisé dans l'assemblage de génomes. Cette observation a conduit à l'élaboration récente d'approches visant à conjointement améliorer le *scaffolding* de génomes actuels dans un cadre évolutif et inférer l'organisation des génomes ancestraux de la phylogénie des espèces considérées.

Cette similarité a d'abord été exploitée par [Muñoz et al., 2010], pour donner un ordre et une orientation aux *scaffolds* par fusion de contigs à partir d'un *breakpoint graph* associant les zones homologues des contigs du génome à assembler et du génome de référence afin de compléter les adjacences manquantes.

Le concept a été approfondi par Aganezov *et al.* dans la méthode GOS-ASM [Aganezov et al., 2015; Aganezov and Alekseyev, 2016]. La méthode considère plusieurs génomes actuels apparentés (éventuellement à différents niveaux de fragmentation) et effectue simultanément le *co-scaffolding* de tous les génomes actuels. La méthode est basée sur une implémentation du *multiple breakpoint graph* [Avdeyev et al., 2016] et suit le principe de la parcimonie sur les permutations (*cf.* section 3.3.1, p. 49). En conséquence, la méthode est limitée à un petit nombre d'espèces (moins de 10) et ne gère pas les duplications ; elle permet néanmoins de considérer un nombre inégal de contenu en marqueurs génomiques.

En parallèle du développement de GOS-ASM, nous avons également développé un algorithme, nommé ARt-DeCo, qui permette de conjointement inférer l'histoire évolutive de l'ordre de gènes et améliorer le *scaffolding* des génomes actuels considérés, présenté dans le chapitre 4. L'approche est basée sur la reconstruction d'histoires évolutives d'adjacences de gènes de l'algorithme DeCo. Par la suite, l'algorithme a été étendu pour pouvoir incorporer des données de séquençage appariées et renommé ADseq, présenté dans le chapitre 5.

Très récemment, la méthode RACA, permettant d'effectuer le *scaffolding* de génomes actuels, présentée dans la section 2.2.3 (p. 38), a été étendue pour

la reconstruction de génomes ancestraux. Cette extension de l'algorithme RACA, nommé DESCHRAMBLER [Kim et al., 2017] permet de reconstruire un génome ancestral à partir de génomes actuels répartis en deux groupes :

— les génomes descendants du génome ancestral à reconstruire ;
— les génomes en position d'*outgroup* par rapport au génome ancestral cible.

L'approche de DESCHRAMBLER consiste à reconstruire des fragments synténiques à partir d'alignements deux à deux entre les génomes actuels et les génomes *outgroup* avec le protocole utilisé dans [Kim et al., 2013] (*cf.* p. 42). Après sélection des fragments synténiques prédits comme présents dans l'espèce ancestrale d'intérêt par DESCHRAMBLER, celui-ci calcule la probabilité d'adjacences des fragments synténiques dans l'espèce ancestrale d'intérêt avec l'approche probabiliste développée dans l'algorithme RACA. Il est à noter que DESCHRAMBLER n'utilise pas de données de séquençage appariées pour guider la reconstruction de l'organisation de génomes ancestraux[6]. Un graphe d'adjacences de fragments synténiques ancestraux est généré puis raffiné par une approche de *Maximum-Weight Matching*. Tandis que la méthode RACA permet d'assembler des génomes actuels, DESCHRAMBLER permet, lui, de prédire l'organisation synténique de génomes ancestraux. Cette dernière n'est donc pas, à proprement parler, une méthode de reconstruction conjointe de l'organisation génomique d'espèces actuelles et ancestrales mais l'association des approches développées dans RACA et DESCHRAMBLER pourrait permettre cela.

Conclusion sur les méthodes de reconstruction de la structure de génomes ancestraux Il y a eu un effort important, principalement au cours des quinze dernières années, dans le développement de méthodes de calcul pour la reconstruction d'organisations génomiques ancestrales. L'objectif de la thèse a eu pour but d'étendre l'algorithme DeCo afin de permettre d'améliorer le *scaffolding* de génomes actuels conjointement avec la reconstruction de l'histoire évolutive et de l'ordre de gènes. Les méthodes développées au cours de cette thèse surpassent les algorithmes GOS-ASM et DESCHRAMBLER car elles permettent de prendre en compte un plus grand nombre de génomes en entrée (testée sur 69 grands génomes d'eucaryotes). De plus, contrairement

6. données essentielles dans le cadre de l'algorithme RACA.

à DESCHRAMBLER, nos méthodes permettent de reconstruire simultanément l'organisation de l'ensemble des génomes ancestraux de la phylogénie considérée. Enfin, nos méthodes permettent de considérer des événements de duplications et de pertes de marqueurs génomiques pour la reconstruction conjointe, ce qui n'est pas le cas pour la méthode GOS-ASM qui peut néanmoins considérer un contenu inégal en marqueurs génomiques dans les génomes actuels.

Deuxième partie

Réalisations effectuées au cours de la thèse

La baisse exponentielle des coûts de séquençage de génomes complets, grâce au développement des technologies de séquençage haut-débit, a permis au début de ma thèse, en 2014, d'avoir accès à un grand nombre de génomes complets permettant la reconstruction de l'ordre des gènes des génomes ancestraux et de leur histoire évolutive. Cependant, malgré l'accès à la séquence complète pour la majorité des génomes de grande taille (supérieur à $100\ Mpb$), leur structure reste incomplètement résolue. Les travaux effectués au cours de ma thèse ont consisté à développer des méthodes permettant de reconstruire l'ordre des gènes chez des génomes ancestraux et leur histoire évolutive en considérant l'ordre incomplet des gènes actuels tout en prédisant de nouvelles adjacences entre ces gènes.

Cette partie du manuscrit a pour but de présenter les réalisations de cette thèse. Les deux premiers chapitres 4 et 5 présentent respectivement les deux méthodes développées au cours de cette thèse : ARt-DeCo (Assembly Recovery through Detection of Coevolution) et ADseq (ARt-DeCo with sequencing data). La méthode ADseq est une extension de ARt-DeCo permettant de considérer des informations supplémentaires de synténies entre marqueurs. Le chapitre 6 présente le logiciel DeCoSTAR qui est une initiative visant à intégrer l'ensemble des algorithmes d'extensions de DeCo, dont les algorithmes ARt-DeCo et ADseq. Les chapitres 7 et 8 détaillent les résultats obtenus avec le logiciel DeCoSTAR sur un jeu de données composé de 18 espèces de moustiques du genre *Anopheles*. Le chapitre 7 décrit l'amélioration du *scaffolding* des 18 génomes et le chapitre 8 décrit l'histoire évolutive de l'ordre des gènes que nous avons inférée et son utilisation pour discuter de la phylogénie de ces 18 espèces d'*Anopheles* déterminée par Fontaine et al. [2015].

Chapitre 4

Assemblage phylogénétique : ARt-DeCo

Dans ce chapitre, nous présentons l'algorithme ARt-DeCo, une extension de l'algorithme DeCo, permettant conjointement la reconstruction de l'ordre des gènes ancestraux, ainsi que l'histoire évolutive de l'ordre des gènes et l'amélioration du *scaffolding* des génomes actuels.

Dans la section 4.1, nous présentons les raisons ayant abouti au développement de l'algorithme ARt-DeCo. La section 4.2 présente les modifications apportées aux formules de récurrence de l'algorithme DeCo permettant de considérer des adjacences potentielles entre gènes actuels localisés aux extrémités des fragments génomiques. La section 4.3 décrit un protocole de validation permettant d'évaluer les prédictions de nouvelles adjacences inférées par l'algorithme ARt-DeCo.

4.1 Observations impliquant le développement de ARt-DeCo

Avant d'effectuer ma thèse dans l'équipe de phylogénie et évolution moléculaire de l'Institut des Sciences de l'Évolution de Montpellier (ISE-M), j'ai effectué un stage de Master 2 dans la même équipe sous la direction de Sèverine Bérard et l'encadrement de Vincent Berry et Annie Chateau. Au cours de ce stage, l'objectif était dans un premier temps d'analyser l'implémentation de l'algorithme DeCo (*cf.* Section 3.3.3, p. 57) et du programme C++ dans lequel il était inclus, nommé méthode DeCo, afin de partitionner le code pour permettre d'exécuter chaque partie indépendamment des autres. La méthode DeCo était à l'origine composée d'un seul exécutable.

L'analyse du code a permis de déterminer 4 blocs dans le programme :

— un premier bloc correspondant à la lecture des arbres de gènes et de l'arbre d'espèces, et la réconciliation des premiers avec le second (*cf.* section 1.3.3) ;

— une seconde partie consiste en l'établissement de classes d'équivalence d'adjacences telles qu'elles ont été décrites p. 58 ;

— la troisième partie est composée de l'algorithme DECO qui prend chaque classe d'équivalence établie et infère un ou plusieurs arbres d'adjacences correspondant à cette classe ;

— la quatrième partie consiste à reprendre les différentes sorties du programme et à résumer les données statistiques de l'exécution afin de pouvoir les comparer à d'autres expériences et analyser l'histoire évolutive des génomes.

Au cours de ce stage, à chaque avancée de la ré-implémentation du programme, j'ai effectué des tests pour vérifier la conservation de l'exactitude des résultats de l'algorithme DECO. C'est au cours de l'établissement d'un jeu de données de test (composé de 11 mammifères de la base de données Ensembl) que j'ai réalisé le degré d'incomplétude de l'assemblage des génomes actuels présents dans les bases de données (*cf. Genomes OnLine Database* (*GOLD*) `https://gold.jgi.doe.gov/`). Or, pour reconstruire l'histoire de coévolution de gènes, il est évident que des génomes fortement fragmentés vont entraîner des histoires évolutives incomplètes, voire erronées, en inférant un grand nombre de cassures d'adjacences n'ayant pas eu lieu au cours de l'évolution des génomes.

Un nouvel objectif du stage a donc émergé : il s'agissait de développer un moyen qui permette de considérer la fragmentation des génomes dans la reconstruction d'histoires évolutives d'adjacences de gènes, pour lequel une première implémentation répondant à ce problème a été effectuée au cours de ma thèse.

4.2 De DECO à ART-DECO

Dans DECO, les couples de gènes d'espèces actuelles non adjacents ne sont jamais considérés comme adjacents car un coût c_1 infini leur est affecté (*cf.* cas 1 de l'annexe p. 179) empêchant l'inférence d'histoires évolutives incluant ces adjacences. Or, l'assemblage incomplet, parfois à de très fort degré (*cf.* exemple 2), indique comme non adjacents des gènes qui le sont peut être

dans la réalité et entraîne par effet de cascade une inférence incomplète de la structure des génomes ancestraux.

Exemple 2

Le génome de l'ornithorynque (*Ornithorhynchus anatinus*) est composé de 21 paires de chromosomes homologues et 10 chromosomes non appariés [Rens et al., 2004]. Soit un haplome de 31 chromosomes pour un génome de référence composé de 291.092 *scaffolds* (15.142, si on prend en compte uniquement les *scaffolds* contenant des gènes) dans la version 89 (mai 2017) de la base de données Ensembl.

4.2.1 Calcul de la probabilité d'adjacences entre deux gènes

Dans les espèces fragmentées, nous voulions disposer d'une mesure permettant de savoir si des gènes non adjacents dans les données pouvaient être adjacents en réalité. Pour cela nous avons choisi de calculer la probabilité pour deux gènes g_1 et g_2 d'une même espèce actuelle d'être adjacents : $P(g_1 \sim g_2)$.

La probabilité pour deux gènes g_1 et g_2 d'être adjacents dans un génome composé de p chromosomes linéaires et dont l'assemblage est formé de n contigs (ou *scaffolds*), est définie par l'équation suivante :

$$P(g_1 \sim g_2) = \begin{cases} 1 & \text{si } g_1 \sim g_2 \text{ est présent} \\ \frac{\text{\# solutions avec } g_1 \sim g_2}{\text{\# solutions pour associer } n \text{ ctg en } p \text{ chr}} & \text{sinon} \end{cases} \quad (4.1)$$

où une solution correspond à une combinaison de fusion des n contigs par leur extrémités afin de former p chromosomes linéaires tout en considérant l'ordre et l'orientation de ces n contigs. Pour deux gènes g_1 et g_2 adjacents dans l'assemblage initial, la valeur $P(g_1 \sim g_2)$ est égale à 1. Et pour g_1 et g_2 non adjacents, la valeur $P(g_1 \sim g_2)$ correspond au ratio du nombre de permutations pour combiner n contigs en p chromosomes linéaires où g_1 et g_2 se retrouvent adjacents sur le nombre total de combinaisons associant n contigs en p chromosomes linéaires.

Nous allons maintenant détailler le calcul du cas où $g_1 \sim g_2$ n'est pas présent dans l'assemblage initial du génome en commençant par le dénominateur.

Calcul du "Nombre de solutions pour transformer n contigs en p chromosomes linéaires"

Dans un premier temps, nous avons élaboré une formule récursive $f(n,p)$ permettant de calculer le "Nombre de solutions pour assembler n contigs en p chromosomes linéaires" que nous expliquerons juste en-dessous :

$$\begin{aligned} f(n,p) &= \frac{1}{p} \times \sum_{x=1}^{n-(p-1)} ((2^{x-1})x! \times C_n^x \times f(n-x, p-1)) \\ &\text{Cas d'arrêt :} \\ &f(n,n) = 1 \\ &f(n,1) = 2^{n-1} \times n! \\ &f(n,p) = 0 \text{ (avec } n < p) \end{aligned} \tag{4.2}$$

Dans l'équation de $f(n,p)$, l'idée est de sélectionner récursivement x contigs pour former un chromosome. La somme $\sum_{x=1}^{n-(p-1)}$ limite la taille des chromosomes de 1 à $n-(p-1)$ contigs et permet ainsi d'explorer l'ensemble des compositions de chromosomes en nombre de contigs. $x!$ correspond aux nombres de possibilités d'ordonner x contigs dans un chromosome linéaire. Le terme 2^{x-1} permet de considérer l'orientation de ces x contigs tout en ne comptant pas deux fois les solutions symétriques ($x-1$ au lieu de x en exposant, *cf.* partie droite de la figure 4.1). C_n^x correspond au nombre de possibilités de prendre x contigs dans un ensemble composé de n contigs. La formule $f(n-x, p-1)$ permet de rendre l'équation récursive en considérant les $(n-x)$ contigs et $(p-1)$ chromosomes restants. Enfin, le tout est divisé par $p!$ ($1/p$ multiplié avec lui-même à chaque pas de récurrence) qui permet d'éviter de compter plusieurs fois les solutions équivalentes, où les mêmes chromosomes sont tirés mais pas dans le même ordre par la récursion (*cf.* partie gauche de la figure 4.1).

Un chromosome étant composé au minimum d'un contig et un génome étant composé au minimum d'un chromosome, on a les relations suivantes : $p, n \in \mathbb{N}^*$ et $n \geq p$. Pour implémenter le calcul $f(n,p)$, nous avons défini une version non récursive de la formule.

Lemme 1. *Pour $p, n \in \mathbb{N}^*$ et $n \geq p$, l'égalité suivante est vérifiée :*

$$f(n,p) = \frac{n!}{p!} \times 2^{n-p} \times C_{n-1}^{p-1} \tag{4.3}$$

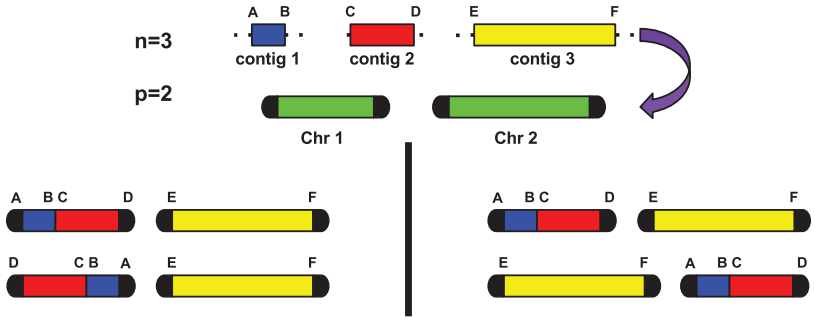


FIGURE 4.1 – Illustration de l'utilisation de l'exposant $x-1$ au lieu de x dans le terme 2^{x-1} et de la division par $p!$ de la formule $f(n,p)$ pour ne pas considérer des solutions symétriques. Dans l'exemple ci-dessus, on a 3 contigs (bleu, rouge et jaune) à assembler en 2 chromosomes linéaires. Les deux résultats de la partie gauche seraient considérés comme non équivalents si à la place du terme 2^{x-1} on avait le terme 2^x dans la formule de $f(n,p)$. Les deux résultats de la partie droite seraient également considérés comme différents, si dans la formule $f(n,p)$ il n'y avait pas la division par $p!$. Pour une résolution complète de cet exemple *cf.* figure 4.2.

Preuve :

1) "Base"

La définition inductive de $f(n,p)$ admet deux cas d'arrêt :

— le cas où $p=n$ et on a $f(n,n)=1$

— le cas où $p=1$ et on a $f(n,1)=2^{n-1}\times n!$

2) Induction

On suppose que pour $p\leq k\leq n$, on a :

$$f(k,p)=\frac{k!}{p!}\times 2^{k-p}\times C_{k-1}^{p-1}$$

On considère $f(n+1,p)$, pour une valeur fixée de $p\in\mathbb{N}^*$, c.-à-d. l'ensemble des combinaisons pour associer $n+1$ contigs en p chromosomes linéaires. Ce qui donne la formule suivante :

$$f(n+1,p)=\frac{1}{p}\times\sum_{x=1}^{n+1-(p-1)}(2^{x-1}\times x!\times C_{n+1}^{x}\times f(n+1-x,p-1))$$

On applique l'hypothèse d'induction sur le terme bleu (où $n+1-x\leq n$ et $p-1\leq n$), on obtient :

$$f(n+1,p)=\frac{1}{p}\times\sum_{x=1}^{n+1-(p-1)}(2^{x-1}\times x!\times C_{n+1}^{x}\times\frac{(n+1-x)!}{(p-1)!}\times 2^{n+1-x-(p-1)}\times C_{n-x}^{p-2})$$

On développe le terme rouge et on a :

$$f(n+1,p) = \frac{1}{p} \times \sum_{x=1}^{n+1-(p-1)} \left(\frac{(n+1)!}{(n+1-x)!} \times \frac{(n+1-x)!}{(p-1)!} \times 2^{x-1} \times 2^{n+1-x-(p-1)} \times C_{n-x}^{p-2}\right)$$

que l'on simplifie en :

$$f(n+1,p) = \frac{1}{p} \times \sum_{x=1}^{n+2-p} \left(\frac{(n+1)!}{(p-1)!} \times 2^{n+1-p} \times C_{n-x}^{p-2}\right)$$

puis on sort les termes indépendants de x de la somme et on obtient :

$$f(n+1,p) = \frac{(n+1)!}{p!} \times 2^{n+1-p} \times \sum_{x=1}^{n+2-p} \left(C_{n-x}^{p-2}\right)$$

Nous changeons la variable x par la variable h dans la somme, avec $h = n-x$:

$$f(n+1,p) = \frac{(n+1)!}{p!} \times 2^{n+1-p} \times \sum_{h=p-2}^{n-1} \left(C_{h}^{p-2}\right)$$

Par la *hockey-stick identity* [Jones, 1994], définie par :

$$n, r \in \mathbb{N}, n > r, \sum_{i=r}^{n} (C_i^r) = C_{n+1}^{r+1}$$

Nous obtenons finalement :

$$f(n+1,p) = \frac{(n+1)!}{p!} \times 2^{n+1-p} \times C_{n}^{p-1}$$

ce qui conclut la preuve.

La figure 4.2 illustre le calcul $f(n,p)$ dans le cas où $n = 3$ et $p = 2$ aboutissant au résultat $f(3,2) = 12$, correspondant au nombre de solutions associant linéairement 3 contigs en 2 chromosomes.

Calcul du "Nombre de solutions pour assembler n contigs en p chromosomes linéaires où g_1 est adjacent à g_2"

Pour calculer ce nombre, une première étape consiste à déterminer le "Nombre d'adjacences possibles entre g_1 et g_2 en fonction de leur nombre de voisins" noté $\rho(g_1 \sim g_2)$ et illustré dans la figure 4.3.

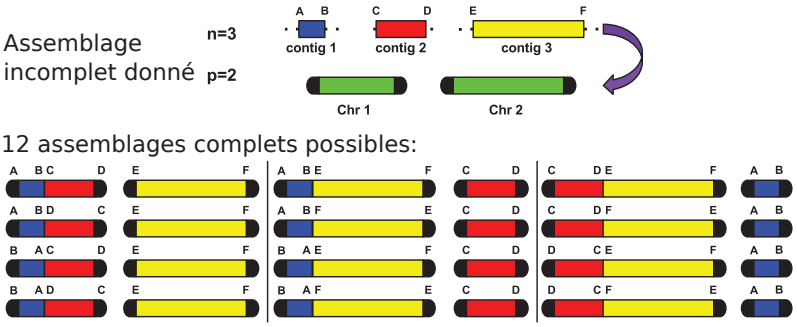


FIGURE 4.2 – Illustration de la formule $f(n,p)$ pour $n=3$ et $p=2$. $f(3,2)=12$

$$\rho(g_1 \sim g_2) = \begin{cases} 4 & \text{si } g_1 \text{ et } g_2 \text{ ont 0 voisin} \\ 2 & \text{si } g_1 \text{ (resp. } g_2\text{) a 0 voisin et } g_2 \text{ (resp. } g_1\text{) a 1 voisin} \\ 1 & \text{si } g_1 \text{ et } g_2 \text{ ont 1 voisin} \\ 0 & \text{si } g_1 \text{ ou } g_2 \text{ a 2 voisins} \end{cases}$$

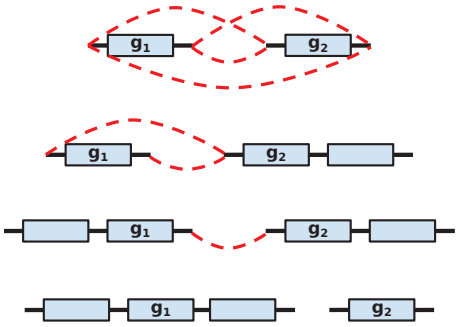


FIGURE 4.3 – Illustration du nombre de combinaisons d'adjacences possibles entre deux gènes g_1 et g_2, $\rho(g_1 \sim g_2)$, en fonction de leur nombre de voisins dans l'assemblage initial. Les adjacences possibles sont représentées par les traits rouges pointillés pour chacun des quatre cas.

Une fois g_1 et g_2 liés par une adjacence, il reste $n-1$ contigs. Dans un deuxième temps, il faut donc déterminer le nombre de solutions pour assembler $(n-1)$ contigs en p chromosomes linéaires. Ce qui est déduit de la formule 4.3 :

$$f(n-1,p)$$

La formule du numérateur de $P(g_1 \sim g_2)$ (*cf.* équation 4.1, p. 71) correspondant au "Nombre de solutions pour assembler n contigs en p chromosomes linéaires où g_1 est adjacent à g_2" consiste à associer ces deux formules et on obtient l'expression suivante :

$$\rho(g_1 \sim g_2) \times f(n-1,p) \tag{4.4}$$

Formule de $P(g_1 \sim g_2)$

La probabilité que le gène g_1 soit adjacent au gène g_2 en assemblant linéairement n contigs à p chromosomes est déduite des équations 4.1, 4.3 et

4.4 :

$$P(g_1 \sim g_2) = \begin{cases} 1 & \text{si } g_1 \sim g_2 \text{ est présent} \\ \rho(g_1 \sim g_2) \times \frac{f(n-1,p)}{f(n,p)} & \text{sinon} \end{cases}$$

or,

$$\frac{f(n-1,p)}{f(n,p)} = \frac{(n-p)}{2n(n-1)}$$

d'où,

$$P(g_1 \sim g_2) = \begin{cases} 1 & \text{si } g_1 \sim g_2 \text{ est présent} \\ \rho(g_1 \sim g_2) \times \frac{(n-p)}{2n(n-1)} & \text{sinon} \end{cases} \quad (4.5)$$

4.2.2 Intégration de $P(g_1 \sim g_2)$ dans les formules de programmation dynamique de l'algorithme DECO

L'idée principale est de considérer que deux gènes g_1 et g_2 non constatés adjacents mais appartenant à une espèce incomplètement assemblée ont une probabilité d'être en réalité adjacents égale à $P(g_1 \sim g_2)$. Pour intégrer les probabilités d'adjacences dans le calcul de la matrice de coût minimum entre les couples de gènes de l'algorithme DECO, nous avons dû modifier les formules de récurrence afin qu'elles puissent prendre en compte le nouveau schéma de scores. On note que $f_G(g)$ et $f_D(g)$ sont respectivement les fils gauche et droit du gène g, et que $E(g)$ est l'événement évolutif du gène g pouvant prendre pour valeur {*Gène actuel*, $Spec$, $GDup$, $GLos$, $ADup$, $ALos$} (*cf.* table 1.1 p. 23). Dans la section suivante, nous présentons les modifications apportées aux formules de récurrence de l'algorithme DECO (*cf.* annexe p. 179).

Modification du cas 1 des formules de récurrence de DECO

Cas 1. $E(g_1) =$ *Gène actuel* et $E(g_2) =$ *Gène actuel*.

Le changement du schéma de scores s'effectue au niveau du **cas 1.** des formules de récurrence de DECO car les probabilités d'adjacences sont calculées uniquement entre deux gènes actuels. Nous avons intégré les probabilités d'adjacences des gènes g_1 et g_2 pour le calcul des coûts $c_1(g_1, g_2)$ [1] et $c_0(g_1, g_2)$ [2], et sommes arrivés à la définition suivante :

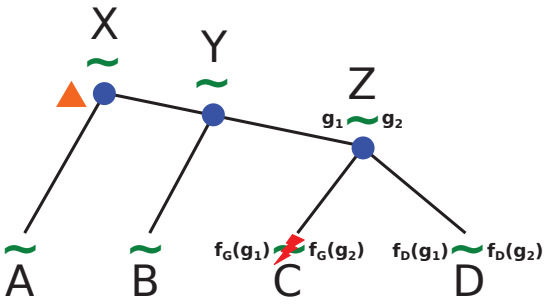
1. $c_1(g_1, g_2)$: coût minimum d'une histoire évolutive où g_1 et g_2 sont adjacents.
2. $c_0(g_1, g_2)$: coût minimum d'une histoire évolutive où g_1 et g_2 ne sont pas adjacents.

$$\begin{aligned} c_1(g_1, g_2) &= -log_b(P(g_1 \sim g_2)) \\ c_0(g_1, g_2) &= -log_b(1 - P(g_1 \sim g_2)) \end{aligned} \tag{4.6}$$

Pour les valeurs extrêmes de $P(g_1 \sim g_2)$, on observe bien des résultats identiques à ceux l'algorithme DeCo. Ainsi, pour $P(g_1 \sim g_2) = 0$, on a bien $c_1(g_1, g_2) = \infty$ et $c_0(g_1, g_2) = 0$ et pour $P(g_1 \sim g_2) = 1$, on a inversement $c_1(g_1, g_2) = 0$ et $c_0(g_1, g_2) = \infty$. L'utilisation du log permet classiquement d'intégrer des probabilités dans des schémas de score compris dans l'intervalle $[0, \infty[$. De plus, il a été nécessaire de déterminer une valeur théorique pour la base du log, notée b, qui permette de favoriser la création d'une adjacence entre deux gènes g_1 et g_2 lorsque celle-ci n'est pas présente dans les données et que des adjacences homologues à $g_1 \sim g_2$ sont présentes chez des espèces proches. Pour déterminer cette valeur de b, nous avons choisi de nous appuyer sur un exemple générique décrit dans la section suivante.

Choix de la base du $logb$

Pour illustrer le choix de la valeur de b, prenons l'arbre d'adjacences suivant :



Cet arbre d'adjacences représente l'histoire évolutive d'adjacences entre les gènes d'une famille de gènes nommée AG_1, à laquelle appartient le gène g_1, et les gènes d'une famille nommée AG_2, à laquelle le gène g_2 appartient. Cette histoire évolutive implique les espèces actuelles A, B, C et D, et leurs espèces ancestrales X, Y et Z. Le symbole ~ illustre la présence d'une adjacence entre les gènes de la famille AG_1 et de la famille AG_2 (*cf.* table 1.2, p. 23, pour la signification des symboles évolutifs). Cet arbre montre que l'adjacence entre les gènes de AG_1 et de AG_2 a été créée chez l'espèce ancestrale X et qu'elle est présente chez toutes les espèces de l'arbre à l'exception de l'espèce actuelle C pour laquelle les deux gènes de ces familles, $f_G(g_1)$ et $f_G(g_2)$, sont présents mais ne sont pas adjacents dans les données. L'algorithme DeCo infère alors une cassure d'adjacences. Or, si les gènes $f_G(g_1)$ et

$f_G(g_2)$ de l'espèce C sont situés en extrémités de contigs, il est possible que l'absence d'adjacence entre ces deux gènes ne soit pas due à un réarrangement chromosomique (cassure d'adjacence) mais à un assemblage incomplet du génome de l'espèce C.

Le calcul ayant permis de prédire la présence d'une adjacence au nœud Z correspond au **cas 5.** $E(g_1) = Spec$ et $E(g_2) = Spec$ des formules de récurrence de l'algorithme DECO (*cf.* annexe p. 179). L'adjacence étant inférée comme présente dans l'espèce Z, on s'intéresse uniquement aux formules de récurrence pour le calcul du coût $c_1(g_1, g_2)$ du **cas 5.** :

c(⚡) = coût d'une cassure d'adjacence

$$c_1(g_1,g_2) = min \begin{cases} (1) & c_1(f_G(g_1),f_G(g_2)) + c_1(f_D(g_1),f_D(g_2)) \\ (2) & c_1(f_G(g_1),f_G(g_2)) + c_0(f_D(g_1),f_D(g_2)) + c(⚡) \\ (3) & c_0(f_G(g_1),f_G(g_2)) + c_1(f_D(g_1),f_D(g_2)) + c(⚡) \\ (4) & c_0(f_G(g_1),f_G(g_2)) + c_0(f_D(g_1),f_D(g_2)) + 2 \times c(⚡) \end{cases}$$

La formule minimisant le coût $c_1(g_1, g_2)$ dans la version originale de l'algorithme DECO est celle représentée en rouge. Pour favoriser la création d'une adjacence entre $f_G(g_1)$ et $f_G(g_2)$ dans l'espèce C, il faut que la formule en vert soit celle qui minimise le coût $c_1(g_1, g_2)$. Pour cela il faut résoudre l'équation suivante :

$$c_1(f_G(g_1),f_G(g_2)) < c_0(f_G(g_1),f_G(g_2)) + c(⚡)$$

Ci-dessous, nous développons cette équation afin de définir une valeur empirique de b. Avec les formules de l'équation 4.6 et $Br = c(⚡)$, on obtient l'équation suivante :

$$-log_b(P(f_G(g_1) \sim f_G(g_2))) < -log_b(1 - P(f_G(g_1) \sim f_G(g_2))) + Br$$

la probabilité $P(f_G(g_1) \sim f_G(g_2))$ est calculée à partir de la formule de l'équation 4.5. Pour déterminer la valeur de b, on fixe $\rho(f_G(g_1) \sim f_G(g_2)) = 1$ permettant de favoriser la création pour deux gènes n'ayant chacun qu'une extrémité libre. Pour plus de lisibilité dans les formules, nous notons, $P_{1\sim1}$, la probabilité $P(f_G(g_1) \sim f_G(g_2))$ pour $\rho(f_G(g_1) \sim f_G(g_2)) = 1$. On obtient ainsi l'équation :

$$-Br < log_b(\frac{P_{1\sim1}}{(1 - P_{1\sim1})})$$

avec la relation $log_b(x) = \frac{ln(x)}{ln(b)}$, on a :

$$-Br < \frac{ln(\frac{P_{1\sim1}}{(1-P_{1\sim1})})}{ln(b)}$$

ce qui équivaut à :

$$\frac{P_{1\sim1}}{(1-P_{1\sim1})} > e^{-Br\times ln(b)}$$

avec la relation $a^x = e^{xln(a)}$, on obtient :

$$\frac{P_{1\sim1}}{(1-P_{1\sim1})} > b^{-Br}$$

ce qui détermine la valeur de b en fonction du paramètre $P_{1\sim1}$ et du coût de cassure d'une adjacence Br :

$$b > \sqrt[Br]{\frac{1-P_{1\sim1}}{P_{1\sim1}}}$$

Pour attribuer une valeur fixe à la base du log b, nous avons choisis le premier entier supérieur à $\sqrt[Br]{\frac{1-P_{1\sim1}}{P_{1\sim1}}}$:

$$b = \left\lceil \left(\frac{1-P_{1\sim1}}{P_{1\sim1}} \right)^{\frac{1}{Br}} \right\rceil \tag{4.7}$$

Il est à noter que la valeur empirique de b sera différente pour chaque espèce considérée par ART-DeCo, b ayant pour paramètre $P_{1\sim1}$ qui lui-même à pour paramètres n et p. Le paramètre p, correspondant au nombre de chromosomes attendus, est spécifique à chaque espèce et le paramètre n, correspondant au nombre de contigs, renseigne sur le degré de complétude de l'assemblage du génome de chaque espèce. La robustesse de cette valeur a été testée et validée dans la section 4.3.3.

4.2.3 Complétion avec l'algorithme DeClone pour l'exploration de l'ensemble des scénarios

En parallèle de mon stage de Master 2, un collaborateur de mes encadrants de thèse, Cédric Chauve, a développé et publié une extension de l'algorithme DeCo permettant de ne plus limiter le résultat de DeCo à un seul scénario mais à une exploration de l'espace des solutions. Ce nouvel algorithme, nommé DeClone, intègre une implémentation de la distribution de

Boltzmann qui permet cette exploration [Chauve et al., 2014]. L'objectif est de pouvoir pondérer les adjacences ancestrales et actuelles prédites en fonction de leur fréquence d'apparition dans l'ensemble des scénarios échantillonnés.

La distribution de probabilité de Boltzmann

Pour illustrer la distribution de probabilité de Boltzmann appliquée à l'algorithme DECO, il est nécessaire d'introduire plusieurs concepts et notations :

— $s_a(F)$ est le score de parcimonie d'une forêt d'arbres d'adjacences F, c.-à-d. la somme des coûts de créations et cassures d'adjacences ;

— $\mathcal{F}(AG_1, AG_2)$ correspond à l'ensemble des forêts d'arbres d'adjacences optimales et sous-optimales obtenues à partir des arbres de gènes réconciliés des familles de gènes AG_1 et AG_2.

Le **facteur de Boltzmann** d'une forêt d'arbres d'adjacences F est défini comme suit :

$$B(F) = e^{-\frac{s_a(F)}{kT}}$$

où kT est une constante arbitraire. k correspond à la constante de Boltzmann et T à la température qui est dérivée de l'utilisation initiale de la distribution de Boltzmann dans le domaine de la thermodynamique. La **fonction de partition** associée à deux arbres de gènes AG_1 et AG_2 est déterminée par l'expression :

$$\mathcal{Z}(AG_1, AG_2) = \sum_{F \in \mathcal{F}(AG_1, AG_2)} e^{-\frac{s_a(F)}{kT}}$$

La fonction de partition définit une **distribution de probabilité de Boltzmann** sur $\mathcal{F}(AG_1, AG_2)$, où la probabilité d'une forêt d'arbres d'adjacences F est définie comme :

$$P(F) = \frac{e^{\frac{-s_a(F)}{kT}}}{\mathcal{Z}(AG_1, AG_2)}$$

En favorisant exponentiellement le choix de forêts d'arbres d'adjacences avec des scores de parcimonie faibles, la distribution de Boltzmann fournit un moyen alternatif de sonder l'espace des solutions influencé par le choix de kT. Pour ($kT \to \infty$) la distribution est uniforme sur l'ensemble de l'espace des solutions et pour ($kT \to 0$) les solutions les plus parcimonieuses sont dominantes dans la distribution. Ainsi, pour ($kT \to 0$) on peut associer une probabilité à une adjacence actuelle ou ancestrale prédite par ART-DECO+DECLONE comme étant le ratio de la somme des probabilités des forêts d'arbres d'adjacences les plus parcimonieuses contenant cette adjacence avec la fonction de

partition. Les formules de programmation dynamique de l'algorithme ART-DECO+DECLONE sont présentées en annexe p. 181.

4.3 Validation de l'algorithme ART-DECO

L'algorithme ART-DECO associé à l'exploration des solutions parcimonieuses avec DECLONE a fait l'objet d'une publication dans le journal *BMC Genomics* [Anselmetti et al., 2015] qui était couplée à une présentation orale lors de la conférence *RECOMB-CG 2015* qui a eu lieu à Francfort du 4 au 7 octobre 2015. La publication présente la méthode ART-DECO permettant l'amélioration de l'assemblage de génomes actuels par la prédiction d'adjacences actuelles dans le cadre d'une reconstruction cohérente de l'histoire évolutive structurale de ces génomes. Les jeux de données utilisés dans cet article proviennent majoritairement de la version 79 (mars 2015) de la base de données Ensembl [Cunningham et al., 2015].

4.3.1 La méthode ART-DECO

Tout comme l'algorithme DECO est intégré dans un logiciel également nommé DECO, ART-DECO est intégré dans un logiciel du même nom. Cela permet de prétraiter les données d'entrée afin d'exécuter autant d'itérations de ART-DECO qu'il y a de classes d'adjacences puis de synthétiser les forêts d'arbres adjacences en sortie de l'ensemble des exécutions de ART-DECO. La méthode ART-DECO prend en entrée :

- l'arbre des espèces des génomes étudiés donnant leurs liens de parentés et le nombre de chromosomes p de ces espèces ;
- l'ensemble des adjacences de gènes observées structurant les génomes considérés, à partir duquel le nombre de contigs n est déduit ;
- les histoires évolutives des gènes inférées sous la forme d'arbres de gènes réconciliés ou non.

À partir de ces données, la méthode ART-DECO génère un ensemble d'arbres d'adjacences qui associés les uns aux autres par les gènes qu'ils possèdent en commun vont permettre de reconstruire l'ordre des gènes des génomes des espèces ancestrales et permettre l'inférence de nouvelles adjacences chez les génomes d'espèces actuelles (*cf.* figure 4.4).

Dans un premier temps, le logiciel considère les arbres de gènes donnés en entrée de la méthode. Si les arbres ne sont pas réconciliés, alors le logiciel effectue une **étape de réconciliation** parcimonieuse, avec un modèle *DL*

de gènes, des arbres de gènes avec l'arbre des espèces donné en entrée de la méthode (*cf.* section 1.3.3 pour plus d'informations sur la réconciliation d'arbres de gènes). Une fois les arbres de gènes réconciliés, le logiciel **établit les classes d'adjacences** comme défini p. 58. Puis l'algorithme ARt-DeCo traite chacune des classes indépendamment les unes des autres et permet la reconstruction d'une forêt d'arbres adjacences. Chaque arbre d'adjacences reconstruit une fraction de la structure de génomes actuels et ancestraux. Une dernière étape analyse l'ensemble des adjacences inférées par la méthode dans les génomes ancestraux et actuels, et les associe les unes avec les autres par les gènes qu'elles partagent. Cela permet d'obtenir l'ordre des gènes chez les génomes ancestraux et de prédire de nouvelles adjacences de gènes chez les génomes actuels.

Le logiciel ARt-DeCo peut être appliqué sur de grands jeux de données du fait de sa faible complexité $O(m * n^2)$ (où m est égal au nombre de classes d'adjacences et n le nombre de feuilles dans un arbre de gènes).

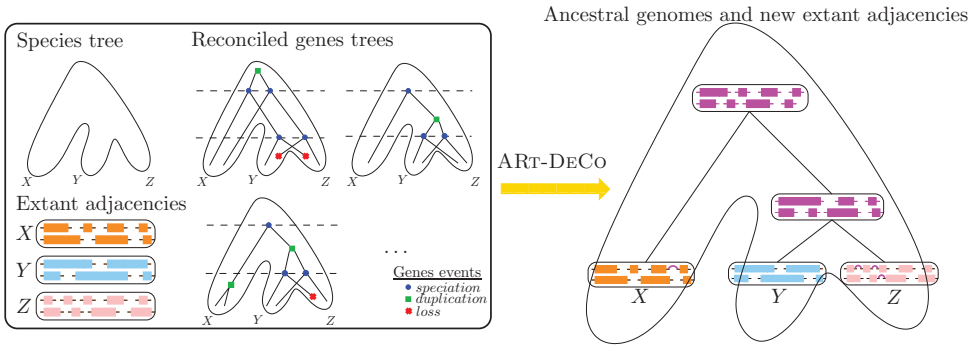


FIGURE 4.4 – Schéma global du logiciel ARt-DeCo. L'encadré à gauche représente les données d'entrée de la méthode ARt-DeCo : l'arbre des espèces X, Y et Z, l'ensemble des adjacences présentes sur les génomes de ces espèces (représentés sous forme de contigs) et les arbres réconciliés. Le résultat du logiciel est représenté sur la partie droite : l'arbre des espèces avec l'ordre des gènes reconstruit pour les génomes ancestraux et les nouvelles adjacences prédites pour les génomes actuels représentées sous la forme de liens magenta.

4.3.2 Validation des nouvelles adjacences prédites par la méthode ARt-DeCo

Pour valider la capacité de ARt-DeCo à prédire de nouvelles adjacences, nous avons reproduit un jeu de données utilisé par Aganezov *et al.* pour valider leur méthode de prédiction d'adjacences actuelles [Aganezov et al., 2015].

Le jeu de données utilisé par Aganezov *et al.*, composé de six mammifères (*Homo sapiens* (GRCh38), *Mus musculus* (GRCm38.p2), *Rattus norvegicus* (Rnor

5.0), *Canis familiaris* (CanFam3.1), *Macaca mulatta* (MMUL 1.0) et *Pan troglodytes* (CHIMP2.1.4)), a été obtenu avec l'outil Ensembl-BioMart [Kasprzyk, 2011] qui a permis de récupérer les génomes complets des six mammifères et les 11.816 familles de gènes orthologues unicopies partagés par l'ensemble des six espèces.

Dans notre expérience, nous avons suivi le même protocole à l'exception de l'ajout d'une 7e espèce dans notre jeu de données (*Gallus gallus*). en effet ART-DECO utilise des phylogénies racinées pour la reconstruction d'histoires évolutives, tandis que la méthode d'Aganezov *et al.* est insensible à la position de la racine. Nous avons ajouté le poulet en position externe de l'arbre, position à laquelle le *scaffolding* est difficilement possible. Car pour toute adjacence présente chez l'ensemble des espèces, excepté l'espèce en position externe de l'arbre, il est plus parcimonieux de considérer que l'adjacence a été créée chez l'ancêtre des espèces internes. C'est pourquoi, nous avons décidé d'ajouter une espèce en position externe de l'arbre pour une meilleur comparaison à [Aganezov et al., 2015]. Le jeu de données ainsi obtenu est composé de 7 espèces d'amniotes pour lesquelles on obtient 8.818 familles de gènes orthologues unicopies partagés par l'ensemble des espèces.

Nous avons généré plusieurs simulations de génomes fragmentés de $n = 150$ à $n = 1050$ cassures aléatoires dans l'ensemble des génomes, de manière équivalente à l'expérience d'Aganezov *et al.*. Pour chaque n, l'expérience a été répliquée 30 fois et pour chaque réplicat, nous avons calculé les statistiques $True\ positive$ et $False\ positive$ décrites dans [Aganezov et al., 2015] :

$$True\ positive = \frac{VP}{n + FP}$$

$$False\ positive = \frac{FP}{n + FP}$$

Où n est le nombre de cassures d'adjacences artificielles. VP (vraies positives) correspond au nombre d'adjacences prédites par la méthode ART-DECO qui correspondent à une adjacence artificiellement rompue. Et FP (fausses positives) correspond au nombre d'adjacences prédites par la méthode ART-DECO qui ne correspondent pas à une adjacence artificiellement cassée.

Les résultats de la méthode ART-DECO sur notre jeu de données de 7 amniotes (*cf.* table 4.1 et figure 4.5) sont similaires à ceux obtenus par la méthode d'Aganezov *et al.* (*cf.* figure 6 de [Aganezov et al., 2015]). On observe qu'une majorité des cassures d'adjacences simulées sont retrouvées par la méthode mais que le ratio d'adjacences retrouvées décroit lorsque l'on augmente la

fragmentation des génomes. Ce qui est attendu car, plus les génomes sont fragmentés, plus il y a de chances que certaines adjacences de gènes soient perdues par l'ensemble des génomes ou soient minoritaires et empêchent de prédire la présence de ces adjacences chez les génomes fragmentés. Cependant, le taux de $False\ Positive$ n'augmente pas en fonction de la fragmentation et stagne autour de 1,2%.

n	50	150	250	350	450	550	650	750	850	950	1050
VP	283	829	1364	1895	2418	2922	3431	3917	4398	4875	5338
FP	14	16,5	21	24,5	32	40	46	57	63	73,566	83
$True\ positive$	88,64%	**89,98%**	89,38%	89,00%	88,32%	87,35%	86,84%	85,87%	85,12%	84,38%	83,58%
$False\ positive$	4,40%	1,78%	1,39%	**1,15%**	1,18%	1,19%	1,17%	1,25%	1,22%	1,27%	1,30%

TABLE 4.1 – Table représentant les moyennes des statistiques $True\ positive$ et $False\ positive$, et le nombre moyen d'adjacences vraies positives (VP) et fausses positives (FP) prédites par ARt-DeCo. *Scaffolding* effectué sur des simulations de fragmentation d'un jeu de données de 7 amniotes, composés de gènes orthologues unicopies et universels, pour différents nombres de cassures simulées (n) pour chacun des 7 génomes.

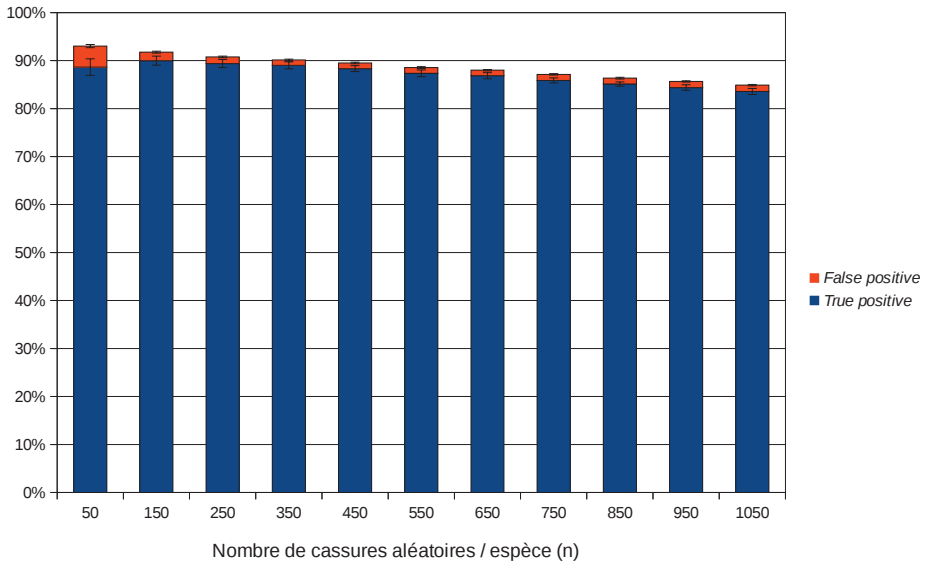


FIGURE 4.5 – Graphique illustrant les statistiques $True\ positive$ et $False\ positive$, de la table 4.1.

4.3.3 Validation du choix de la base du log b

Pour évaluer si le choix de la base du log b est correct, nous avons utilisé la méthode de validation de la méthode ARt-DeCo décrit à la section précédente puis nous avons testé différentes valeurs pour la base du log b, allant de $0,1 \times b$ à $10 \times b$ avec un focus sur les valeurs proche de b. Nous avons donc calculé les statistiques $True\ positive$ et $False\ positive$ pour les différentes valeurs de b en simulant 550 fissions pour chacun des 7 génomes d'amniotes.

Les résultats de cette validation, illustrés dans la figure 4.6, montrent qu'il y a un palier lorsque l'on passe d'une base de log d'une valeur de $0,95 \times b$ à b. Pour une valeur de base de log inférieure à b, on a un taux $True\ positive$ faible (inférieur ou égal à 10%) et pour des valeurs de base de log supérieure ou égale à b, on a un taux $True\ positive$ élevé (de l'ordre de 87%). On observe donc une bien meilleure prédiction de nouvelles adjacences pour une valeur base de log supérieure ou égale à b, ce qui confirme le choix empirique que nous avons défini pour la valeur de la base du log b.

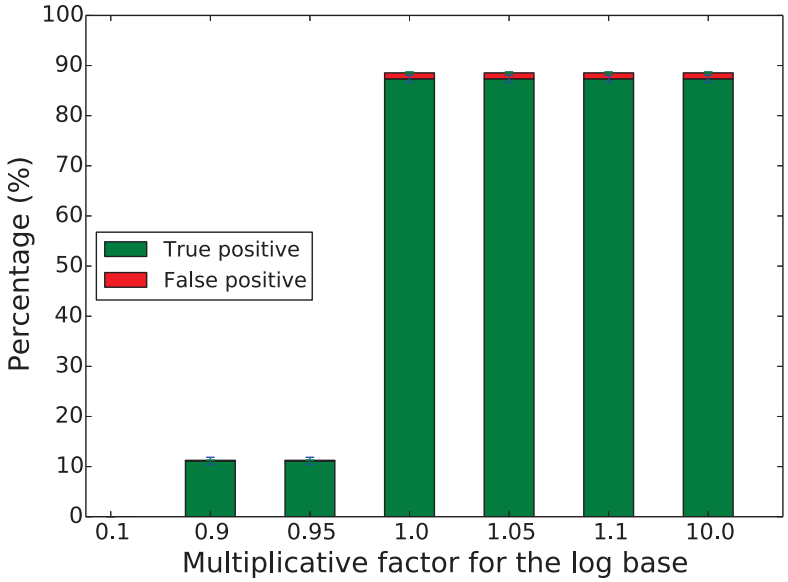


FIGURE 4.6 – Valeurs des statistiques de *scaffolding* $True\ positive$ et $False\ positive$ pour différentes valeurs du coefficient multiplicateur appliqué à la base du $log\ b$.

4.3.4 Amélioration du *scaffolding* des génomes actuels

Nous avons appliqué la méthode ART-DECO à l'ensemble des 69 espèces de la base de données d'Ensembl. Le jeu de données est composé de 20.279 arbres de gènes contenant 1.222.543 gènes codant pour des protéines ordonnées par 1.023.492 adjacences. Ce qui correspond à un ratio de ∼30% de gènes qui ont un nombre de voisins inférieur ou égal à 1, alors que pour un génome complètement assemblé le ratio attendu est inférieur à 1%. Le temps d'exécution de la méthode ART-DECO sur le jeu de données est d'environ 18h sur un ordinateur de bureau, cela indique que la méthode est applicable sur des jeux de données relativement grands.

La méthode ART-DECO prédit 36.445 nouvelles adjacences actuelles sur ce jeu de données. Comme on peut le voir sur la figure 4.7, il y a une baisse

de la proportion de gènes actuels sans adjacence et une augmentation de la proportion de gènes actuels avec deux voisins au dépend d'une très faible proportion de gènes avec plus de deux adjacences. Ces conflits synténiques proviennent du fait que chaque classe d'adjacences est analysée indépendamment les unes des autres.

Pour analyser l'amélioration du scaffolding par ARt-DeCo, nous avons calculé le pourcentage d'amélioration du *scaffolding* d'un génome par rapport à son assemblage initial. Ce pourcentage est obtenu avec la formule :

$$\frac{C_I - C_N}{C_I - p}$$

Où C_I correspond au nombre de contigs dans le génome initial. C_N correspond au nombre de *scaffolds* après l'inférence d'adjacences par ARt-DeCo. Et p correspond au nombre de chromosomes attendus pour le génome. Les résultats illustrés dans la figure 4.8 montrent que plus le génome initial est fragmenté plus l'amélioration du *scaffolding* est fort.

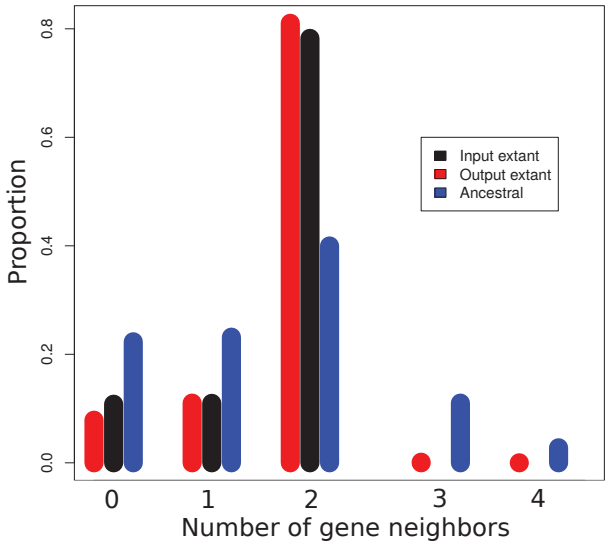


FIGURE 4.7 – Distribution du degré d'adjacences des gènes en entrée et en sortie de ARt-DeCo.

Pour analyser plus en détails la complexité des conflits synténiques chez les espèces actuelles, nous avons calculé un degré de non-linéarité représentant le degré des gènes incohérents avec une conformation linéaire :

$$D_{nl} = \sum_{x-1}^{n} (d_x - 2) + (m - 2)$$

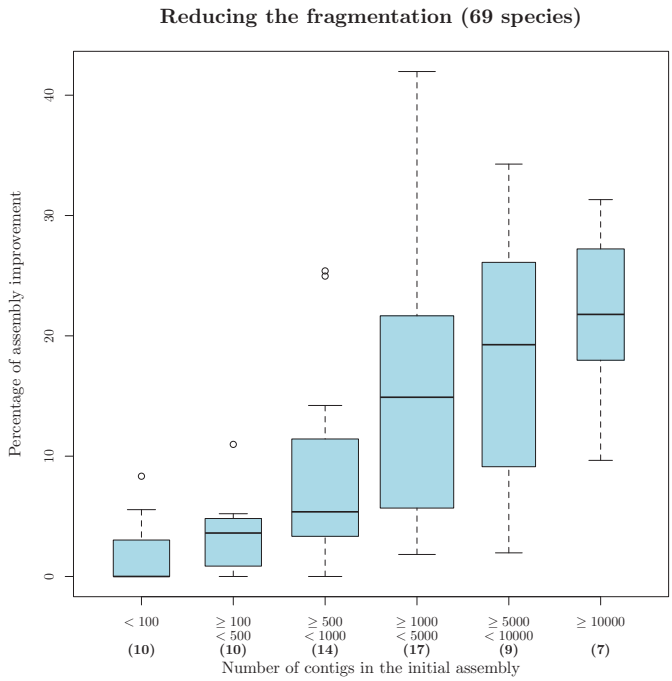


FIGURE 4.8 – Pourcentage d'amélioration du *scaffolding* des 69 eucaryotes d'Ensembl. Les valeurs entre parenthèses représentent le nombre d'espèces dans les différentes catégories de fragmentation des génomes (le nombre total équivaut à 67 et non 69 car, pour 2 espèces, aucune nouvelle adjacence n'a été prédite). Ces catégories correspondent à des intervalles de nombre de contigs dans l'assemblage initial.

Où n correspond au nombre de gènes avec plus de deux adjacences, d_x au degré du gène x et m au nombre de gènes avec un degré 1. La figure 4.9 représente la moyenne des degrés de non-linéarité pour les 43 espèces sur 69 qui ont au moins un *scaffold* non linéaire. 23 parmi les 43 ont des *scaffolds* non linéaires avec uniquement une adjacence en trop. Pour les 20 espèces restantes, les *scaffolds* restants sont plus arborés et quelques uns sont circulaires.

En complément des prédictions de nouvelles adjacences actuelles, ARt-DeCo a inféré 1.547.546 adjacences ancestrales entre 3.245.572 gènes ancestraux. Comme cela a été montré dans [Hahn, 2007; Boussau et al., 2013], des erreurs lors de l'inférence d'arbres de gènes introduisent un nombre important de duplications de gènes ancestraux lors de la réconciliation des arbres de gènes, et produisent des génomes ancestraux plus grands qu'attendus. Les arbres de gènes disponibles dans la base de données Ensembl n'échappent pas à cette observation. Cependant, les barres bleues de la figure 4.7 montrent qu'une majorité des gènes ancestraux ont un nombre d'adjacences inférieur ou égal à 2 et une faible, mais non négligeable, proportion sont en conflits synténiques. La figure 4.10 illustre l'histogramme de densité

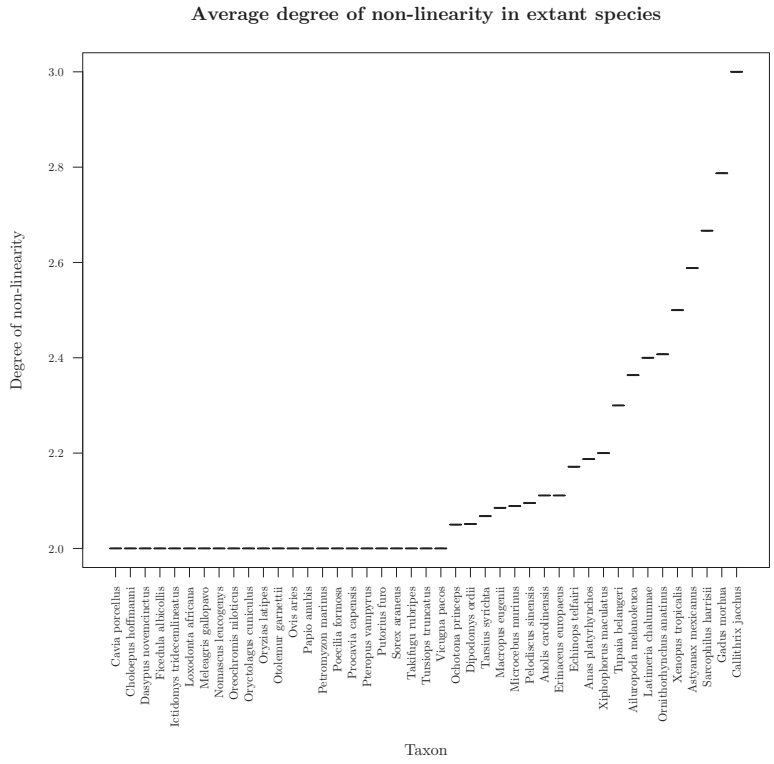


FIGURE 4.9 – Distribution des moyennes des degrés de non-linéarité pour les *scaffolds* non linéaires par espèces actuelles.

des degrés de non-linéarité des *scaffolds* non linéaires chez les espèces ancestrales. On observe que la plupart des *scaffolds* ancestraux ont un degré de non-linéarité inférieur à 20. De plus, on observe que plus de 50% d'entre eux ont un degré de non-linéarité inférieur à 6, indiquant que la plupart des *scaffolds* ancestraux non linéaires sont faiblement arborés. Cependant, une proportion non négligeable d'entre eux sont fortement arborés et quelques uns circulaires. Des étapes complémentaires sont nécessaires pour linéariser ces génomes, cependant l'obtention de meilleurs arbres de gènes permettrait de prévenir une importante portion de ces conflits synténiques.

4.3.5 Étude de cas d'une nouvelle adjacence prédite par la méthode ART-DECO+DECLONE

Pour faire une analyse plus précise des adjacences prédites par ART-DECO, nous avons analysé en détail l'adjacence entre les gènes RCDS1 et CREG1 chez le panda (*Ailuropoda melaneuloca*). Pour déterminer un score de

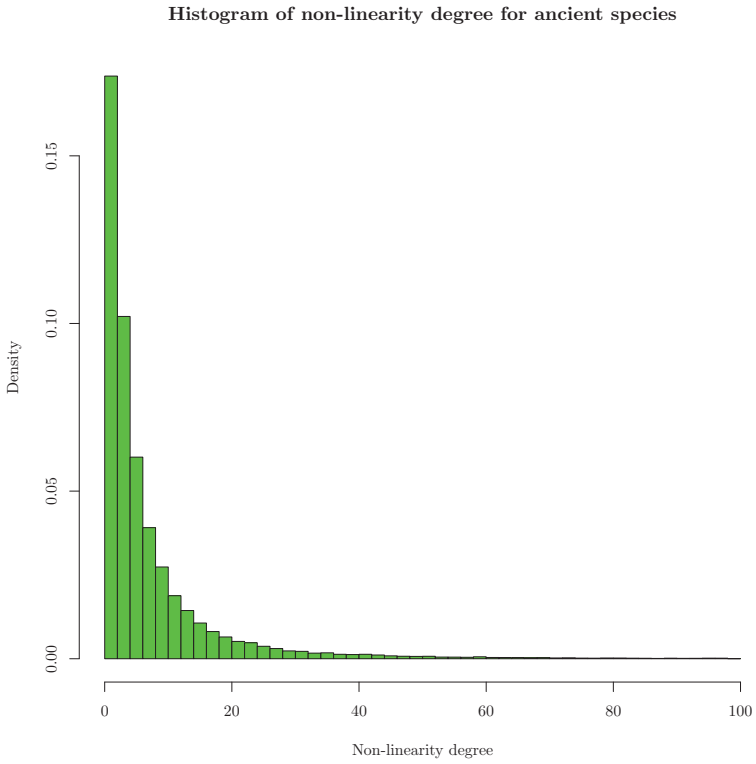


FIGURE 4.10 – Histogramme de densité des degrés de non-linéarité pour les contigs ancestraux non linéaires.

confiance pour cette adjacence, nous avons refait tourner l'algorithme ART-DECO en le combinant à l'algorithme DECLONE sur la classe d'adjacence correspondant à la forêt d'arbres d'adjacences entre les familles de gènes RCDS1 et CREG1. Le score d'une adjacence prédite correspond à la proportion de scénarios dans lesquels l'adjacence est prédite sur le nombre total de scénarios analysés. Pour échantillonner uniquement des scénarios parcimonieux, nous avons fixé la valeur kT de l'algorithme DECLONE à 0.1.

Il est à noter que pour cette expérience, les algorithmes ART-DECO et DECLONE sont implémentés dans deux logiciels séparés (le premier est implémenté en C++ et le second en python) [3]. Le logiciel ART-DECO établit les classes d'adjacences et calcule pour chaque espèce la probabilité pour deux gènes g_1 et g_2 situés aux extrémités de contigs d'être adjacents où g_1 et g_2 ont déjà un voisin, ce qui revient à la formule $\frac{f(n-1,p)}{f(n,p)}$ déduite de l'équation 4.5, notée P_1. L'algorithme DECLONE traite chaque classe d'adjacence indépendamment les unes des autres. Pour chaque couple de gènes actuels g_1 et g_2, appartenant à la même espèce entre deux arbres d'une même classe,

3. les algorithmes sont maintenant implémentés dans le logiciel DECOSTAR que nous présenterons dans le chapitre 6.

DeClone calcule les coûts $c_0(g_1, g_2)$ et $c_1(g_1, g_2)$ avec les formules 4.6 avec $P(g_1 \sim g_2)$ égal à 1 si g_1 est adjacent à g_2 dans l'assemblage initial et à $\rho(g_1 \sim g_2) \times P_1$ si g_1 n'est pas adjacent à g_2. Pour chaque classe d'adjacence, l'algorithme DeClone explore l'espace des solutions et génère une forêt d'arbres d'adjacences où à chaque adjacence actuelle ou ancestrale est associé un support. Après traitement de l'ensemble des classes, le logiciel incluant DeClone génère l'ensemble des adjacences ancestrales et actuelles inférées avec leur support associé.

L'histoire évolutive de l'adjacence entre RCSD1 et CREG1 est représentée dans la figure 4.12. L'histoire reconstruite par ARt-DeCo+DeClone établit que l'adjacence entre ces deux familles de gènes est apparue au niveau de l'ancêtre des tétrapodes et que celle-ci a subi un grand nombre de duplications et pertes de gènes et d'adjacences de gènes au cours de l'évolution des mammifères. Étant donné que chez toutes les espèces actuelles possédant une adjacence entre les gènes RCSD1 et CREG1 celle-ci est présente en une seule copie. Il est plus probable de penser qu'il y a eu au cours de l'histoire évolutive une seule copie de RCSD1 et CREG1, et que chez certaines espèces l'un des deux gènes a été perdu expliquant la disparition de l'adjacence. Ce point de vue est renforcé par l'ordre des gènes dans le voisinage des gènes conservés chez l'ensemble des tétrapodes. Et on observe une similarité de structure accrue chez les amniotes, qui regroupe les sauropsides (oiseaux+reptiles) et les synapsides (groupe auquel appartiennent les mammifères). Si l'on compare le poulet (*Gallus gallus*) et la vache (*Bos taurus*), on observe que la structure est exactement la même à l'exception de l'apparition du gène ADCY10 entre les gènes MPZL1 et MPC2, et de la substitution du gène chOCT-1 par POU2F1. Cependant, depuis la version 79 d'Ensembl, le gène chOCT-1 a été ré-annoté sous le nom de gène POU2F1 (*cf.* v.89 d'Ensembl - mai 2017) qui confirme la structure très conservée de cette région du génome chez les amniotes.

Une autre analyse soutenant la structure très conservée de cette partie du génome provient de l'espèce *Microcebus murinus* (microcèbe). L'histoire prédite par ARt-DeCo+DeClone infère une perte du gène RCSD1 chez cette espèce entraînant la disparition de l'adjacence de cette espèce. Cependant l'analyse de la région génomique du gène CREG1, laisse apparaître que de petites portions de la séquence du génome sont non élucidées. La figure 4.11 montre que la zone génomique où se situe le gène RCSD1 dans les autres espèces ne contient pas de gène codant pour une protéine. Cependant, on observe également que la séquence est incomplètement résolue. Si on regarde

la schématisation des contigs on observe une succession de zones blanches indiquant que la séquence d'ADN n'est pas connue à ces endroits. Il est donc fort possible que le gène n'ait pas été détecté par des méthodes d'annotation de gènes à cause de la séquence incomplète du génome et de l'absence de marqueurs pour le détecter. Cette hypothèse est confirmée par la dernière version d'Ensembl (v.89) contenant une séquence du génome de *Microcebus murinus* plus complète que celle de la version 80. Dans cette version, on observe que le gène est bien présent à la place et dans l'orientation attendue par comparaison aux autres espèces de la phylogénie. Même observation pour le gène CD247 non présent dans la version 80 et présent dans la version 89 d'Ensembl.

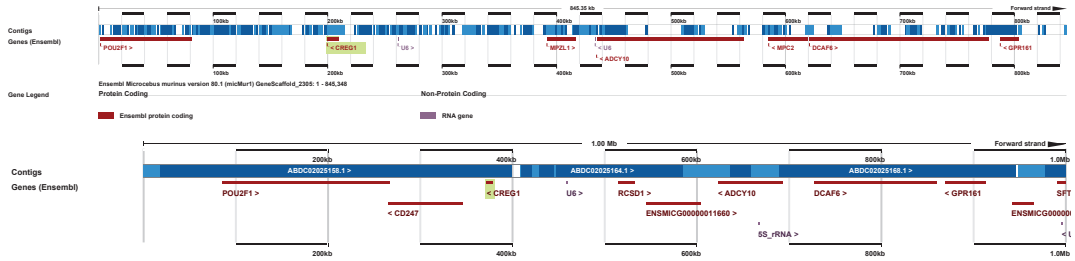


FIGURE 4.11 – Changement dans l'annotation du génome l'espèce *Microcebus murinus* entre la version 80 (haut) et 89 (bas) d'Ensembl.

La très complexe histoire évolutive de cette adjacence inférée par ART-DECO+DECLONE provient très certainement d'une inférence incorrecte des arbres de gènes RCSD1 et CREG1 disponibles dans la base de données Ensembl comme cela a été décrit dans [Boussau et al., 2013]. Dans l'histoire évolutive de l'adjacence entre RCSD1 et CREG1 inférée par ART-DECO+DECLONE une nouvelle adjacence actuelle a été prédite chez le panda ainsi que chez *Ochotona princeps* (pika), *Dipodomys ordii* (rat-kangourou d'Ord) et *Tupaia belangerii* (toupaye de Belanger). Pour l'ensemble de ces adjacences, l'algorithme DECLONE infère de très forts scores de support (> 0.9999) et on observe chez le panda et le rat-kangourou que ces adjacences offrent une structure génomique en accord avec celles des autres amniotes.

Comme on a pu voir dans l'analyse de l'adjacence entre RCSD1 et CREG1, l'inférence d'histoire évolutive adjacences de gènes par la méthode ART-DECO+DECLONE permet la prédiction d'adjacences actuelles en accord avec le reste de la topologie, malgré l'inférence d'une histoire évolutive bien plus complexe que ce qu'elle ne semble être.

En conclusion, nous avons présenté ART-DECO, une extension de l'algorithme DECO permettant d'effectuer le *scaffolding* des génomes actuels en

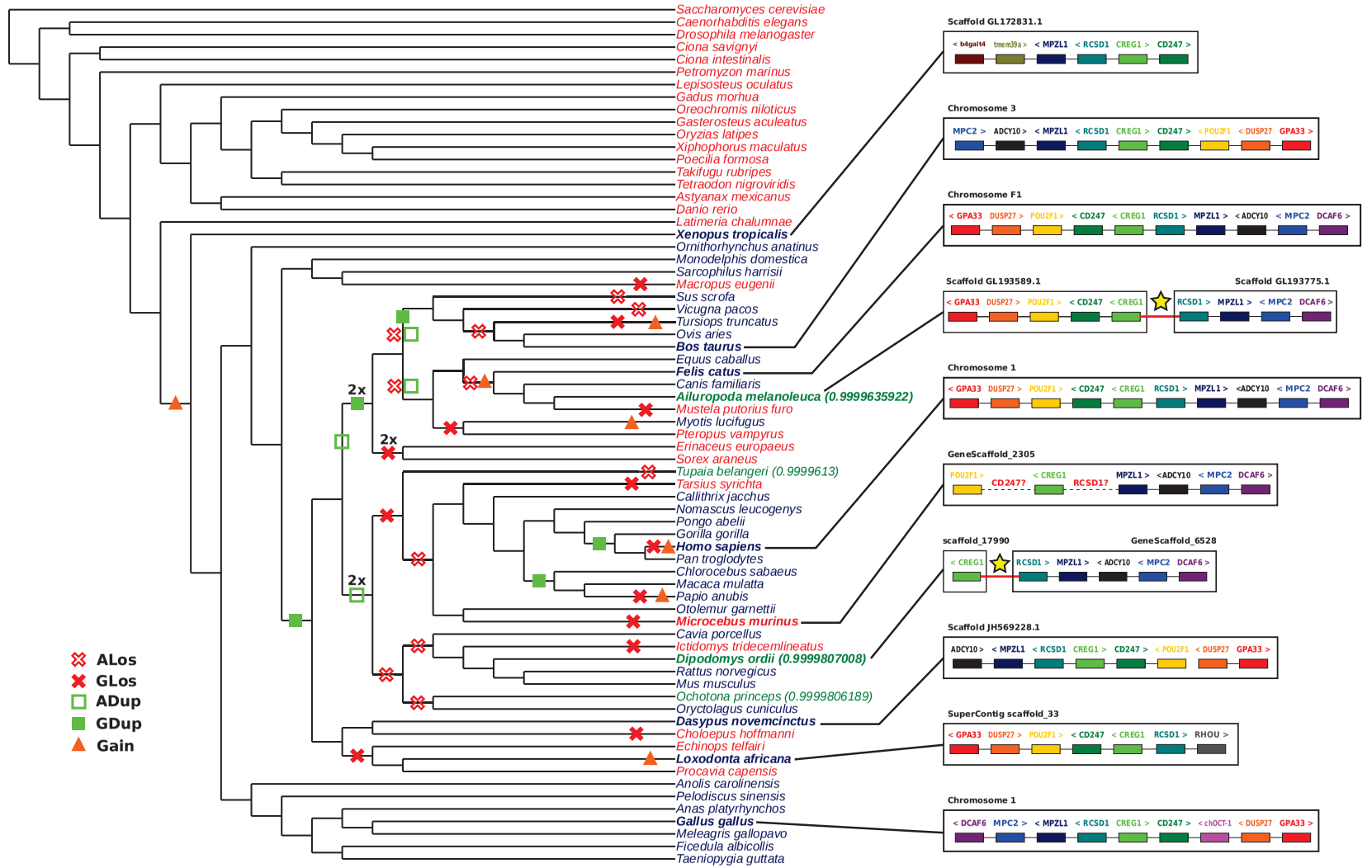


FIGURE 4.12 – Histoire évolutive de l'adjacence entre **RCSD1** et **CREG1**. Pour la signification des symboles voir table 1.1 (p. 23). Espèces rouges : adjacence absente, espèces bleues : adjacence présente, et espèces vertes : adjacence prédite par ART-DECO+DECLONE (avec support DECLONE). Schématisation de l'ordre des gènes dans le voisinage des gènes **RCSD1** et **CREG1**, pour quelques espèces, à partir de la base de données Ensembl (v.79).

s'appuyant sur l'histoire évolutive de l'ordre de gènes reconstruit. L'application à un jeu de données contenant l'ensemble des espèces de la base de données Ensembl a montré que notre méthode peut être utilisée sur de grands jeux de données. De plus, comparativement aux autres méthodes de *scaffolding* de génomique comparative, elle est la seule qui permet de considérer chaque génome comme un génome cible à assembler, et une référence pour l'assemblage d'autres génomes.

Chapitre 5

Assemblage phylogénétique avec données de *scaffolding* : ADSEQ

Dans ce chapitre, nous présentons l'algorithme ADSEQ, une extension de l'algorithme ARt-DeCo, permettant de considérer des adjacences entre gènes actuels pondérées par un score et obtenues par des approches de *scaffolding*. Dans ce chapitre, ces informations complémentaires de *scaffolding* correspondent à des adjacences obtenues à partir de données de séquençage appariées.

Dans la section 5.1, nous présentons les modifications apportées aux formules de récurrence de l'algorithme ARt-DeCo pour l'intégration d'adjacences pondérées par un score, ainsi que les caractéristiques globales du nouveau logiciel, nommé DeCoSTAR, dans lequel est intégré l'algorithme ADSEQ et que nous décrivons plus en détails dans le chapitre 6.

Dans la section 5.2, nous décrivons un pipeline permettant, à partir de données de génomiques et phylogénétiques au format standard, de générer les données d'entrée du logiciel DeCoSTAR pour l'application de l'algorithme ADSEQ. Au cours de cette section, nous détaillons les différents filtres mis en place qui joue un rôle important sur les résultats de ADSEQ car l'algorithme est fortement dépendant de la qualité des arbres de gènes.

Enfin, dans la section 5.3, nous présentons le protocole que nous avons développé pour valider la capacité de l'algorithme ADSEQ à prédire de nouvelles adjacences pour l'amélioration du *scaffolding* de génomes actuels. Pour illustrer les différentes sections, nous utilisons le jeu de données de 18 moustiques du genre *Anopheles*, publié dans *Science* en 2015 [Neafsey et al., 2015; Fontaine et al., 2015] dont l'analyse des données biologiques est détaillée dans les chapitres 7 et 8.

Les résultats présentés dans ce chapitre proviennent d'analyses de données obtenues avec l'algorithme ADSEQ implémenté dans le logiciel DeCoSTAR.

5.1 De ART-DECO à ADSEQ

Le logiciel DECOSTAR, majoritairement développé par Wandrille Duchemin [Duchemin, 2017], a permis l'intégration de l'ensemble des extensions de l'algorithme DECO (DECOLT, DECLONE, ART-DECO, ADSEQ) dans un seul et même programme. Un court chapitre de ce manuscrit est dédié au logiciel DECOSTAR (*cf.* chapitre 6) qui a fait l'objet d'une publication dans *Genome Biology and Evolution* [Duchemin et al., 2017].

5.1.1 Ajout de données de séquençage dans l'algorithme ART-DECO

Au cours du développement de la méthode ART-DECO et de l'analyse des premiers résultats de *scaffolding*, nous avons observé que bien que la méthode améliorait la reconstruction de l'ordre des gènes actuels considérés, celui-ci demeurait incomplètement résolu. Pour améliorer la reconstruction de l'ordre des gènes, nous avons étendu l'algorithme ART-DECO pour intégrer des données supplémentaires d'adjacences pour lesquelles un score de confiance compris dans l'intervalle $[0,1]$ est attribué. Ces adjacences pondérées peuvent provenir de diverses origines, comme des données de séquençage appariées (*cf.* section 2.2.2, p. 34), des données de cartes chromosomiques (*cf.* section 2.2.3, p. 37) ou de *scaffolding* par génomique comparative, comme les méthodes RAGOUT et MEDUSA (*cf.* section 2.2.3, p. 38). Dans la suite de ce manuscrit nous nommons ce type d'adjacences, *adjacences de scaffolding*.

Dans le développement de l'algorithme ADSEQ, nous avons considéré comme adjacences de *scaffolding* des adjacences obtenues à partir de données de séquençage appariées. Ces données permettent d'ordonner et d'orienter des paires de séquences d'ADN plus ou moins proches, selon la taille d'*insert* de la librairie de séquençage utilisée (*cf.* section 2.1.2, p. 29). Pour obtenir ces adjacences de *scaffolding*, nous avons utilisé l'outil de *scaffolding* BESST [Sahlin et al., 2014, 2016]. La prise en compte des données de séquençage appariées avait précédemment été utilisée dans la méthode RACA pour la reconstruction de l'ordre de marqueurs de génomes actuels [Kim et al., 2013] (*cf.* section 2.2.3, p. 42).

5.1.2 Modifications de l'algorithme ART-DECO pour l'intégration d'adjacences de *scaffolding*

Cette section est composée de quatre parties. La première partie présente les différents types de scores d'adjacences considérés en entrée de l'algorithme ADSEQ. Dans la seconde partie, nous présentons les modifications apportées à la définition d'une classe d'adjacence dans l'algorithme ADSEQ. Dans la troisième partie, nous détaillons la modification des formules de récurrence de l'algorithme ART-DECO pour intégrer des adjacences de *scaffolding* pondérées par un score. Dans la dernière partie, nous présentons l'ajout d'un algorithme de linéarisation des prédictions d'adjacences en sortie de l'algorithme ADSEQ afin d'éliminer les conflits synténiques chez les génomes actuels.

Les scores d'adjacences

Dans l'algorithme ADSEQ, à chaque adjacence de gènes actuels est assignée un score initial compris dans l'intervalle $[0,1]$ qui représente le score de confiance d'une adjacence.

— Pour les adjacences de *scaffolding* ce score est obtenu à partir de données permettant d'établir cette adjacence. Dans la suite de ce manuscrit, ce score est obtenu à partir de l'outil de *scaffolding* BESST. Le score considéré pour ces adjacences correspond à la moyenne arithmétique des scores π_σ et π_ζ calculés par BESST (*cf.* section 2.2.3, p. 36) ;
— Pour les adjacences observées sur les génomes actuels, nous avons par défaut attribué un score de 1 dans l'ensemble de nos expériences, indiquant que ces adjacences ne peuvent être remises en cause. Cependant, il est possible de leur attribuer un score dépendant de la confiance que l'utilisateur leur estime. Par exemple, pour deux gènes situés aux extrémités de deux contigs (ou de deux *scaffolds*) et inférés comme adjacents dans un même *scaffold*, l'utilisateur peut vouloir remettre en cause cette adjacence, ne sachant pas la qualité du *scaffolding* de ce génome et la confiance qu'il peut avoir en celle-ci. Dans le logiciel DECOSTAR, il est possible d'attribuer un score inférieur à 1 pour ce type d'adjacences ;
— Pour tous les gènes situés en extrémité de contigs/*scaffolds*, on calcule la probabilité d'adjacence de chaque paire de ces gènes à partir de la formule 4.5 héritée de l'algorithme ART-DECO.

Les classes d'adjacences

La notion de classe d'adjacences définie dans DECO a également été relâchée. En effet, si l'on considère uniquement les couples d'arbres de gènes qui remplissent les critères de définition d'une classe d'équivalence d'adjacence (*cf.* page 58), les classes de taille 1 étaient exclues de l'analyse, elles sont désormais incluses. Cela permet de ne pas exclure une classe ne possédant qu'une seule adjacence observée et pour laquelle d'autres adjacences homologues sont potentiellement présentes (adjacences de *scaffolding* ou adjacences entre gènes en extrémités de contigs) qui pourraient être inférées par ADSEQ. Une option a également été ajoutée pour permettre de créer des classes d'adjacences pour l'ensemble des combinaisons de couples d'arbres de gènes (option *all.pair.equivalence.class* du logiciel DECOSTAR). Cela permet de considérer des couples d'arbres de gènes ne partageant aucune adjacence observée dans les assemblages des génomes actuels mais pour lesquels des gènes localisés en extrémités de contigs sont présents et pour lesquels des adjacences sont donc possibles.

Les formules de récurrence

Dans l'algorithme ADSEQ, comme dans l'algorithme ART-DECO, seul le cas 1 des formules de récurrence de l'algorithme DECO sont modifiés. Le cas 1 correspond au calcul des coûts c_0 et c_1 entre deux gènes actuels g_1 et g_2. Les nouvelles formules sont :

$$\begin{aligned} c_0(g_1, g_2) &= -log_b(1 - S(g_1, g_2)) \\ c_1(g_1, g_2) &= -log_b(S(g_1, g_2)) \end{aligned} \tag{5.1}$$

où $S(g_1, g_2)$ correspond au score de l'adjacence entre g_1 et g_2 dont la valeur est comprise dans l'intervalle $[0,1]$. Le choix des valeurs de la base du log, b, et du score d'adjacence, $S(g_1, g_2)$, est dépendant de la nature de l'adjacence entre g_1 et g_2.

Valeurs de $S(g_1, g_2)$ et b pour les adjacences de *scaffolding* Pour les adjacences de *scaffolding*, le score d'adjacence $S(g_1, g_2)$ correspond au score obtenu à partir d'une méthode de *scaffolding* entre deux gènes g_1 et g_2. La base du log, b, que l'on nomme b_{scaff} pour les adjacences de *scaffolding*, a une valeur par défaut égale à 10.000.

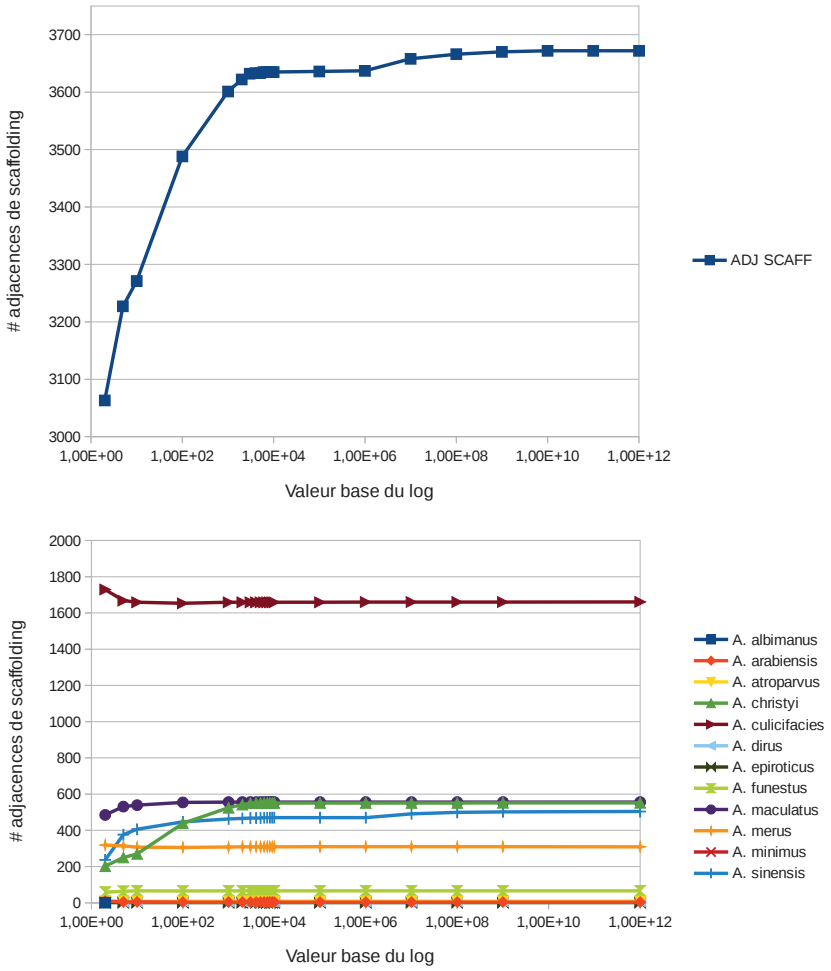


FIGURE 5.1 – Analyse du nombre d'adjacences de *scaffolding* inférées par ADSEQ pour différentes valeurs de base du *log* b_{scaff}.

Choix empirique de la valeur b_{scaff} La figure 5.1 représente la courbe du nombre d'adjacences de *scaffolding* qui ont été inférées par ADSEQ en fonction de la valeur de b_{scaff}, le graphique du haut correspond à ces valeurs pour l'ensemble des espèces et le graphique du bas espèce par espèce. Les résultats pour l'ensemble des espèces montrent que pour une valeur de b_{scaff} supérieure ou égale à 10.000, on arrive à un plateau du nombre d'adjacences de *scaffolding* inférées par la méthode ADSEQ. Cependant, on observe que le nombre d'adjacences de *scaffolding* pour les 12 espèces d'*Anopheles* pour lesquelles des données d'adjacences de *scaffolding* sont disponibles, n'ont pas la même distribution. Par exemple, on observe que pour *Anopheles culicifacies* le nombre d'adjacences de *scaffolding* prédites par la méthode est le plus élevé pour une valeur de b_{scaff} équivalent à 2 et atteint un plateau pour une valeur de b_{scaff} supérieure ou égale à 100. On observe une tendance similaire pour *Anopheles merus* avec une plus faible variation entre le nombre maximal et

minimal d'adjacences inférées. Pour les espèces *Anopheles christyi*, *Anopheles funestus*, *Anopheles maculatus* et *Anopheles sinensis*, on observe une tendance inverse aux deux précédentes espèces avec une augmentation du nombre d'adjacences de *scaffolding* lorsque l'on augmente la valeur de b_{scaff} jusqu'à arriver à un plateau pour une valeur de $b_{scaff} = 10.000$ correspondant à la tendance de la courbe lorsque l'on considère toutes les espèces. Pour *Anopheles sinensis*, on observe également un deuxième plateau pour une valeur de b_{scaff} supérieure ou égale à 10^8 qui explique le second plateau également observé si l'on considère toutes les espèces.

Pour la suite de nos expériences, nous avons choisi de fixer la valeur de b_{scaff} à 10.000. Cependant, l'analyse de la courbe des inférences d'adjacences de *scaffolding* montre des profils différents en fonction des espèces. Ce qui indique que la valeur de b_{scaff} ne devrait pas être la même pour l'ensemble des espèces. Un développement futur consistera à produire un utilitaire permettant à l'utilisateur de déterminer au mieux la valeur de b_{scaff} pour l'ensemble des espèces de son jeu de données. Cet utilitaire sera inspiré du protocole de validation présenté dans la section 5.3.3 (p. 112) afin de déterminer la valeur de b_{scaff} pour laquelle on a les meilleures statistiques de rappel et précision pour les adjacences de *scaffolding* inférées par ADSEQ.

Valeurs de $S(g_1, g_2)$ et b pour les adjacences restantes Pour les adjacences observées et les adjacences possibles entre gènes en extrémités de contigs non présentes dans les adjacences de *scaffolding*, la valeur de $S(g_1, g_2)$ correspond à la valeur de la probabilité $P(g_1 \sim g_2)$ de la formule 4.5 de l'algorithme ART-DECO (p. 76). Le calcul de la base du log, b, est également hérité de l'algorithme ART-DECO (*cf.* formule 4.7 p. 79). Il a été modifié afin d'intégrer le paramètre *SPI* (*Scaffolding Propagation Index*) qui correspond à la taille du clade c dans lequel il est possible d'inférer de nouvelles adjacences homologues à une adjacence observée chez une espèce même si celle-ci n'a pas d'homologue dans c et que le plus proche homologue est en position externe du clade c. On obtient ainsi la nouvelle formule suivante :

$$b = \lceil (\frac{1 - P(g_1 \sim g_2)}{P(g_1 \sim g_2)})^{\frac{SPI}{Break}} \rceil \tag{5.2}$$

Il est à noter que dans ce cas de figure toutes les adjacences observées dans l'assemblage initial des génomes ont un score $S(g_1, g_2)$ égal à 1. Il n'est donc pas possible de remettre en cause ces adjacences qui ont un coût c_1 égal à 0 et un coût c_0 égal à l'infini.

Linéarisation des prédictions de DeCoSTAR

Pour linéariser des prédictions d'adjacences actuelles et ancestrales générées par l'algorithme ADSeq, il est recommandé d'utiliser l'algorithme DeClone afin d'obtenir des adjacences pondérées par un support compris dans l'intervalle [0,1]. Le protocole de linéarisation des adjacences consiste à extraire un *Maximum-Weight Matching* (*MWM*) dans le graphe modélisant l'ordre des gènes du génome à linéariser, où les sommets correspondent aux extrémités de gènes et les arêtes symbolisent les adjacences de gènes pondérées par le support calculé par DeClone. Ce *MWM* peut contenir des segments circulaires qui sont linéarisés en retirant l'arête de plus faible score. Cette approche a précédemment été utilisée pour linéariser des génomes ancestraux [Maňuch et al., 2012b] et des génomes actuels [Mandric and Zelikovsky, 2015]. En amont de cet algorithme de linéarisation, nous retirons de l'analyse les adjacences dont le support DeClone est inférieur à un seuil (donné en entrée de la méthode de linéarisation).

5.2 Génération des données d'entrée du logiciel DeCoSTAR

Afin de rendre plus aisée l'utilisation de la méthode ADSeq pour la reconstruction conjointe de l'ordre de gènes actuels et ancestraux, nous avons développé un *pipeline* permetttant de générer les données d'entrée du logiciel DeCoSTAR, à partir de formats de fichiers standards. Ce pipeline a été développé à partir du jeu de données *Anopheles* provenant de [Neafsey et al., 2015]. Ce jeu est composé de 18 espèces dont l'arbre est illustré dans la figure 5.7 (p. 114), 17.780 familles de gènes, 237.293 gènes dont 212.800 présents dans les arbres de gènes. Parmi les 18 espèces, 16 ont des données de séquençage appariées qui pourront être utilisées pour l'obtention d'adjacences de *scaffolding*.

Le pipeline que nous avons développé est schématisé dans la figure 5.2. Il est divisé en deux parties :

— une partie bleue qui génère les arbres de gènes et les adjacences observées entre les gènes aux feuilles de ces arbres ;
— une partie verte qui génère des adjacences de *scaffolding* à partir de données de séquençage appariées pour les génomes considérés.

Les cadres violets représentent les données d'entrée brut du jeu de données *Anopheles*. En rouge sont représentées les données aux différentes étapes

du pipeline et en vert les différentes étapes du pipeline. Nous décrivons chacune de ces parties bleue et verte indépendantes dans les sous-sections suivantes.

5.2.1 Construction des familles et arbres de gènes

La partie droite (bleue) du pipeline de la figure 5.2 consiste à filtrer les données d'annotations génomiques et les familles de gènes afin de déterminer les arbres de gènes et les adjacences de gènes qui seront considérés par DECOSTAR.

Filtre des familles de gènes contenant des gènes inclus (étapes 1/ et 2/

L'étape **1/** du pipeline consiste à détecter les inclusions de gènes dans les annotations génomiques des génomes. Un gène est défini comme inclus dans un autre lorsque les extrémités de début et fin du gène inclus sont localisées entre les extrémités de début et de fin du gène inclusif. La figure 5.3 illustre des inclusions de gènes sur une portion génomique de l'espèce *Anopheles gambiae* obtenue à partir du navigateur génomique d'Ensembl intégré dans la base de données VectorBase [Giraldo-Calderón et al., 2015]. Dans cette figure, on voit la présence de quatre gènes dont l'un est transcrit dans l'orientation $5' \rightarrow 3'$ (AGAP013454) et les trois autres dans l'orientation $3' \rightarrow 5'$. Pour ces quatre gènes, on observe que trois d'entre eux sont inclus dans le quatrième (AGAP003001). Il est difficile dans ce cas de définir le statut d'adjacence de ces gènes entre eux. Plusieurs choix s'offraient à nous :

— nous pouvions considérer la localisation d'un gène comme étant la position d'un seul nucléotide (par exemple le premier nucléotide à l'extrémité $5'$). Cette approche permet de ne plus avoir de problème d'inclusions de gènes mais limite la localisation d'un gène à un point dans le génome ce qui n'est pas satisfaisant ;

— nous pouvions complexifier la définition des relations d'adjacences entre les gènes dans un modèle permettant de représenter le statut de voisinage, d'inclusion et chevauchement des gènes entre eux. Cela ne serait plus compatible avec une conformation linéaire du génome. Cette approche pourrait être une alternative mais demanderait une importante contribution dans la gestion de ces adjacences par le logiciel DECOSTAR ;

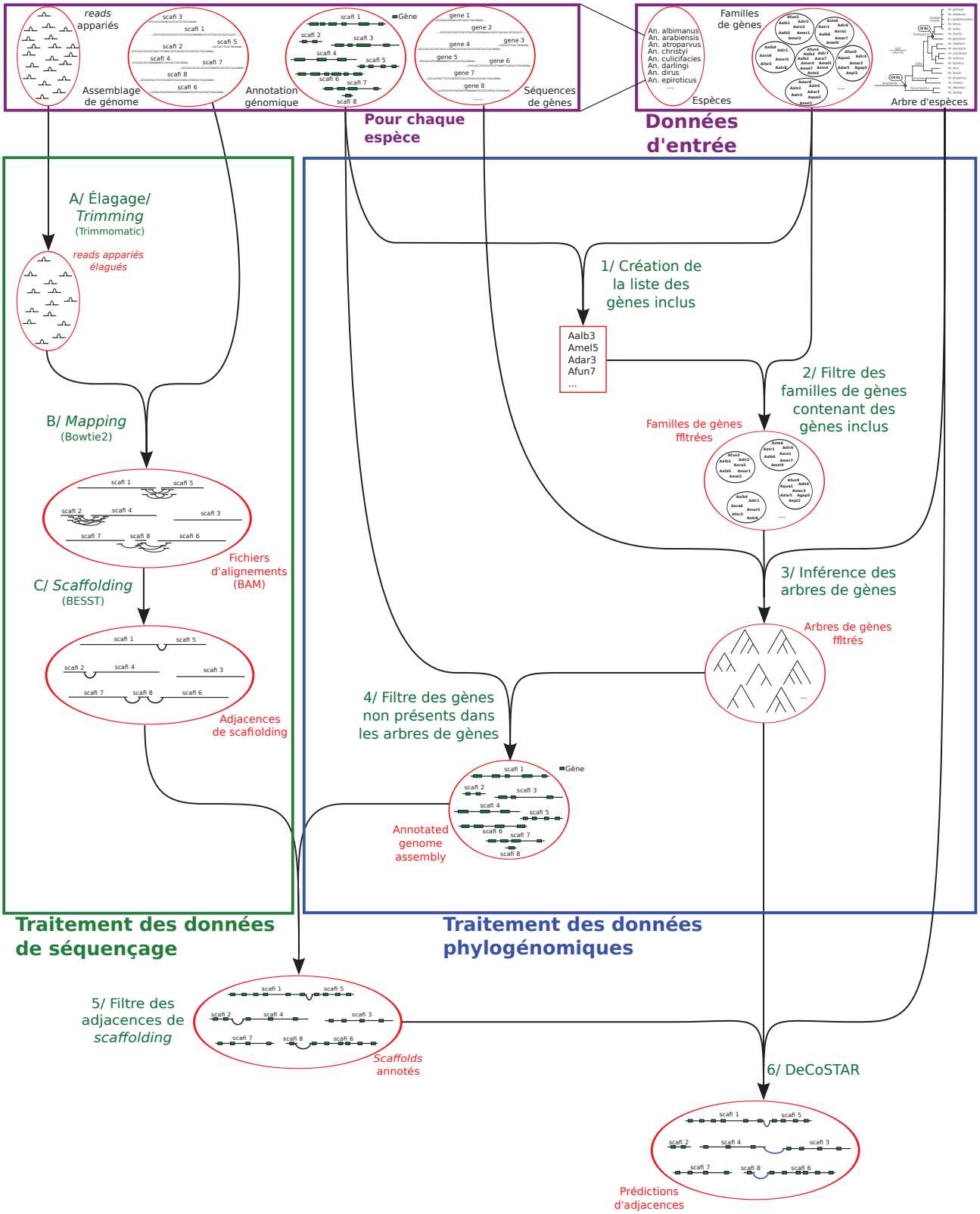


FIGURE 5.2 – Pipeline pour générer les données d'entrée du logiciel DECOSTAR pour l'utilisation de l'algorithme ADSEQ. Les cadres violets correspondent aux données phylogénomiques d'entrée disponibles dans les bases de données. Les cadres en bleu et vert représentent les deux parties indépendantes de traitement des données pour générer l'entrée de DECOSTAR. En rouge est détaillé le contenu des des données aux différentes étapes du pipeline.

— nous pouvions à la place de considérer le gène comme marqueur génomique, se servir des exons de gènes qui sont beaucoup moins propices à des inclusions entre eux. Une étude [Scornavacca and Galtier, 2016] effectuée sur des mammifères a montré que les arbres phylogénétiques reconstruits à partir d'exons d'un même gène sont aussi similaires entre eux que des arbres phylogénétiques entre des exons appartenant à différents gènes. Ces résultats chez les mammifères semblent montrer qu'au sein d'un même gène plusieurs histoires évolutives sont présentes et que l'exon serait un meilleur marqueur génomique pour la reconstruction d'histoires évolutives. Cependant, dans la quasi-totalité des études, l'établissement de groupes de marqueurs orthologues/homologues est obtenu à partir de l'alignement de séquences codantes de gènes (*CDS* [1]) ou de séquences protéiques. De plus, dans certains cas les exons se chevauchent fortement comme on peut le voir dans la figure 5.4 illustrant le chevauchement entre le deuxième exon du gène $AALB004117$ et le troisième exon du gène $AALB004116$ chez l'espèce *Anopheles albimanus* rendant ambiguë la détermination de l'adjacence entre ces deux exons ;

— nous pouvions déterminer l'ensemble des gènes inclus dans d'autres gènes pour qu'ils soient retirés des gènes considérés dans la reconstruction d'adjacences de gènes.

Dans le cadre de ma thèse, nous avons sélectionné ce dernier choix qui est l'approche la plus stricte.

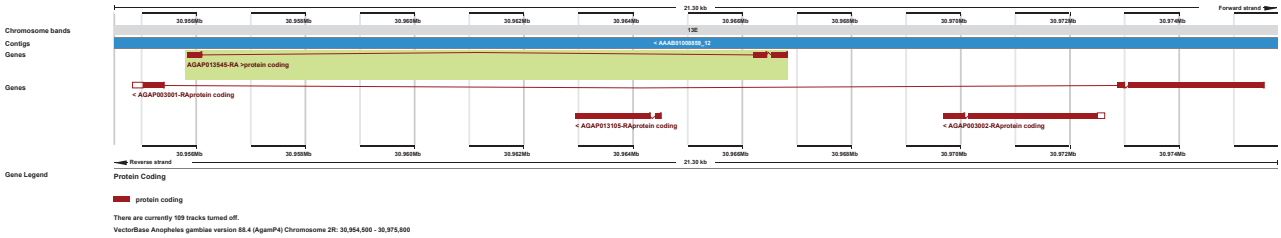


FIGURE 5.3 – Illustration d'inclusions de gènes chez *Anopheles gambiae*.

L'étape **2/** consiste à retirer l'ensemble des familles de gènes contenant des gènes inclus détectés à l'étape précédente. Si l'on retire uniquement les gènes inclus et pas leur famille, l'histoire évolutive de ces familles de gènes est erronée par l'inférence d'un ou plusieurs événements de pertes de gènes, alors que ceux-ci sont bien présents mais sont retirés de l'analyse suite à leur statut. De plus, si la structure génomique est bien conservée, il y a de fortes

1. *Coding DNA Sequence.*

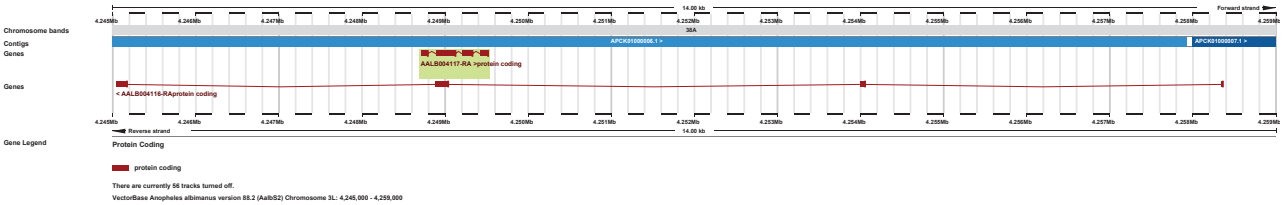


FIGURE 5.4 – Illustration d'un chevauchement d'exons de gènes chez *Anopheles albimanus* entre les gènes AALB004116 et AALB004117.

probabilités que l'ensemble des gènes de la famille d'un gène inclus soient également inclus dans les gènes de la famille du gène englobant le gène inclus.

Au cours de cette étape de filtre des familles de gènes contenant un ou plusieurs gènes inclus, sur le jeu de données *Anopheles*, 2.668 familles de gènes ont été retirées de l'analyse réduisant le nombre de familles considérées à 14.981 représentant 184.719 gènes.

Inférence des arbres de gènes (étapes 3/ et 4/ de la figure 5.2 partie bleue)

Dans l'étape **3/**, les familles sont structurées sous forme d'arbres de gènes par un pipeline que nous avons développé (*cf.* figure 5.5).

L'étape α/ de la figure 5.5 consiste pour chaque famille de gènes à aligner globalement l'ensemble des gènes avec l'outil d'alignement multiple de séquences MUSCLE [Edgar, 2004]. Si dans une famille, un gène a une séquence de taille supérieure à 32.000 pb, nous utilisons le paramètre "-maxiters 2" qui permet de limiter le nombre d'itérations et d'éviter que des familles ne soient pas alignées. La valeur de 32.000 paires de base a été déterminée empiriquement à la suite d'une série de lancement de MUSCLE qui indiquait que pour les familles de gènes contenant une séquence supérieure à 32.000 nucléotides, MUSCLE interrompait l'alignement en cours.

L'étape β/ élague les alignements obtenus avec le programme GBLOCKS [Talavera and Castresana, 2007] qui permet de sélectionner des blocs d'alignements intéressants pour l'inférence d'arbres phylogénétiques. Parmi les 14.981 familles de gènes du jeu *Anopheles*, 41 ont retirées de l'analyse car, pour au moins un gène de ces familles, GBLOCKS ne sélectionne aucun bloc d'alignement et celui-ci se retrouve sans séquence.

L'étape γ/ infère pour chaque famille de gènes un arbre par maximum de vraisemblance avec le modèle $GTR + GAMMA$ et avec un *bootstrap* de 100 itérations avec la méthode RAXML [Stamatakis, 2014] à partir des blocs d'alignements.

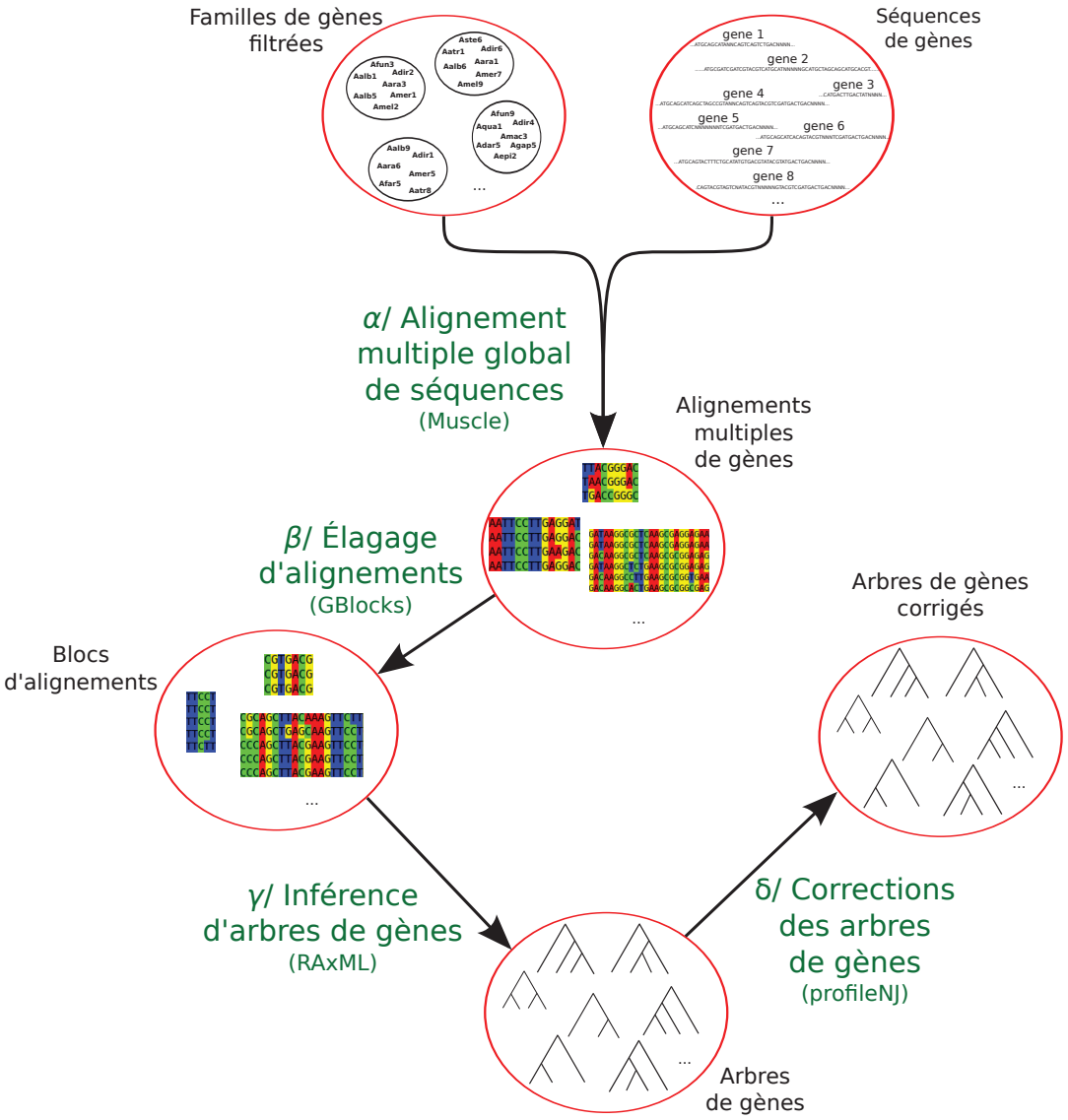


FIGURE 5.5 – Pipeline pour l'inférence d'arbres de gènes avec PROFILENJ (correspondant à l'étape **3/** de la figure 5.2 partie bleue).

Enfin dans l'étape δ/, les arbres de gènes obtenus sont traités par l'outil PROFILENJ [Noutahi et al., 2016]. PROFILENJ contracte les branches de l'arbre de gènes avec un support de *bootstrap* en dessous d'un seuil puis résoud la topologie de l'arbre aux nœuds multifourchés en utilisant un principe de parcimonie, consistant à minimiser le nombre de duplications et pertes de gènes. Dans cette analyse, nous avons choisi une valeur de seuil de 100 qui conserve uniquement les nœuds soutenus par l'ensemble des topologies d'arbres de gènes échantillonnés par RAXML. Parmi les 14.940 arbres de gènes inférés par le pipeline, certains ont plusieurs topologies possibles en sortie de PROFILENJ minimisant le nombre de duplications et de pertes de gènes :

— 14.832 arbres de gènes ont 1 topologie possible
— 88 arbres de gènes ont 2 topologies possibles
— 11 arbres de gènes ont 3 topologies possibles
— 6 arbres de gènes ont 4 topologies possibles
— 3 arbres de gènes ont 5 topologies possibles

Pour les 108 familles de gènes où PROFILENJ propose plusieurs solutions parcimonieuses optimales, nous avons arbitrairement choisi une des topologies proposées.

Dans l'étape **4/**, les 14.940 arbres de gènes, en sortie du pipeline d'inférence d'arbres de gènes, sont ensuite utilisés pour filtrer les annotations génomiques et retirer les gènes qui ne sont pas présents dans les arbres permettant de considérer les adjacences uniquement entre les gènes présents dans les arbres de gènes.

Fin de la construction des arbres et familles de gènes En sortie de la partie bleue du pipeline de la figure 5.2, on obtient 14.940 arbres de gènes composés de 183.680 gènes liés par 147.046 adjacences de gènes pour le jeu de données *Anopheles*.

5.2.2 Obtention d'adjacences de *scaffolding* à partir de données de séquençage appariées.

La partie gauche (verte) de la figure 5.2 présente le pipeline développé pour l'obtention d'adjacences de *scaffolding* à partir de données de séquençage appariés pour apporter des informations d'adjacences supplémentaires à celles observées dans les assemblages de génomes. Lors des premiers tests pour obtenir un ensemble d'adjacences de *scaffolding*, nous pensions que nous

n'aurions que très peu d'adjacences proposées par notre pipeline, les informations d'appariement des *reads* étant déjà prises en compte dans l'assemblage des génomes. Cependant, nous pensions tout de même en avoir de nouvelles car la méthode de *scaffolding* que nous utilisions (BESST) est différente de celle utilisée pour l'assemblage des génomes d'*Anopheles*. Les 16 génomes nouvellement séquencés par Neafsey *et al.* ont été assemblés par ALLPATHS-LG composé d'une étape d'assemblage *de novo* et d'une étape de *scaffolding* utilisant les données de séquençage appariées. De plus dans cette étude, les auteurs ont utilisé une autre option de ALLPATHS-LG permettant le *scaffolding* des génomes assisté par un génome de référence, qui dans cette étude était l'assemblage *P3* du génome d'*Anopheles gambiae*. Les prédictions d'adjacences de *scaffolding* issues de BESST pour les 16 génomes d'*Anopheles*, que nous présentons en détails dans le chapitre 7, montrent qu'un grand nombre d'adjacences de *scaffolding* d'intérêt pour le *scaffolding* de ces génomes étaient présents dans les données de séquençage mais pas entièrement exploitées lors de l'assemblage des génomes.

Description du pipeline pour la génération d'adjacences de *scaffolding* (partie verte de la figure 5.2)

A/ Élagage des données de séquençage appariées Cette étape a pour but de retirer de l'analyse les *reads* de mauvaise qualité et de tronquer leurs extrémités qui souvent correspondent à des zones de faible qualité de séquençage. Cette étape permet également de retirer les adaptateurs utilisés lors du séquençage des espèces et parfois présents dans la séquence des *reads*. Pour effectuer cette étape, nous avons utilisé la méthode TRIMMOMATIC (v0.36) [Bolger et al., 2014]. L'élagage a été effectué sur l'ensemble des données de séquençage appariées présentes dans la table A.1 (p. 191).

B/ Alignement des *reads* appariés sur les génomes de référence Les paires de *reads* ayant passé l'étape d'élagage sont ensuite alignées sur le génome de référence correspondant. Cette étape consiste à déterminer la position et l'orientation des *reads* sur les génomes de référence. Pour cette étape nous avons utilisé l'outil d'alignement BOWTIE2 (v2.2.9) [Langmead and Salzberg, 2012] en considérant l'ensemble des alignements pour chaque *read*, à l'exception des espèces *Anopheles arabiensis* et *Anopheles merus* pour lesquelles seuls les 100 meilleurs alignements ont été considérés dû à des temps de calcul excessifs (plus d'un mois).

C/ *Scaffolding* des assemblages de référence avec les informations d'alignements des *reads* appariés La dernière étape pour l'obtention d'adjacences de *scaffolding* pondérées consiste à utiliser les informations de position et d'orientation des *reads* sur le génome de référence couplé avec les informations de distance (obtenue à partir de la taille d'insert des librairie) entre les *reads* d'une même paire. Cette étape est effectuée avec l'outil de *scaffolding* BESST (v2.2.6) qui calcule deux scores π_σ et π_ζ pour chaque adjacences de *scaffolding* liant des longs contigs dont la longueur est supérieure à $\mu + 4\sigma$ où μ et σ sont respectivement la moyenne et l'écart-type de la taille d'insert de la librairie utilisée et sont estimés par maximum de vraisemblance avec l'outil GAPEST [Sahlin et al., 2012][2]. D'autres approches pourraient dans le futur être développées pour calculer un score pour l'ensemble des adjacences de *scaffolding* considérées par BESST ou par d'autres méthodes de *scaffolding*, afin de ne plus être limité aux adjacences pour lesquelles BESST calcule des scores.

5.2.3 Combinaison des adjacences de *scaffolding* avec les annotations génomiques

Dans la dernière étape du pipeline (étape **5/** de la figure 5.2) seules les paires de *scaffolds* avec un nombre de liens, de *scaffolding* (paires de *reads*) supérieur ou égal à 4 sont conservées, représentant 405.939 paires de *scaffolds* de référence liés par $4.128.682$ liens de *scaffolding*. Enfin, les paires de *scaffolds* de référence sont limitées aux *scaffolds* de référence contenant des gènes présents dans les annotations génomiques et les arbres de gènes en sortie de l'étape **4/**. Ce qui réduit fortement le nombre de paires de *scaffolds* de référence à 68.876 liés par 846.045 paires de *reads*. Cela s'explique par le fait que dans l'assemblage des génomes d'*Anopheles* il y a un grand nombre de *scaffolds* de l'assemblage de référence qui ne contiennent pas de gènes.

Pour finir, l'étape **6/** consiste à exécuter le logiciel DECOSTAR, avec les algorithmes ADSEQ+DECLONE, pour la prédiction de nouvelles adjacences actuelles et l'inférence de l'histoire évolutive de l'ordre des gènes à partir des données d'entrée générées par le pipeline nous venons de décrire.

2. Pour une description de la procédure de pondération des adjacences de *scaffolding* par BESST *cf.* section 2.2.3, p. 36.

5.3 Validation la méthode ADSEQ

5.3.1 Protocole

Lors du développement de la méthode ARt-DeCo nous avons développé un protocole pour la validation des prédictions de nouvelles adjacences actuelles inférées par la méthode. Cette approche consistait à fragmenter aléatoirement les génomes actuels considérés et d'appliquer ARt-DeCo sur ces génomes afin de déterminer la proportion d'adjacences actuelles cassées lors de la fragmentation qui sont retrouvées par ARt-DeCo ainsi que la proportion d'adjacences fausses inférées par ARt-DeCo pour déterminer sa fiabilité (*cf.* section 4.3.2, p. 82). Ce protocole de validation avait pour défaut de considérer que les adjacences manquantes lors de l'assemblage des génomes, illustrant l'incomplétude de leurs assemblages, étaient réparties de façon homogène/aléatoire sur les génomes. Or, on sait que ce sont des zones spécifiques des génomes qui sont responsables de la difficulté d'assemblage, telles que les longues zones répétées, les portions d'ADN faiblement séquencées, les régions riches en GC qui sont difficiles à séquencer et les contraintes algorithmiques des outils d'assemblage empêchant de résoudre la structure complète des génomes.

Pour pallier ce biais, nous avons donc développé un nouveau protocole pour valider la capacité de la méthode ADSEQ à prédire de nouvelles adjacences actuelles en complément d'assemblages de génomes existants. Le principe de cette approche consiste à simuler la fragmentation d'un génome en le ré-assemblant avec une approche plus restrictive dans la gestion des répétitions et donc à l'obtention d'un assemblage plus conservé et fragmenté que l'assemblage original. En procédant ainsi, la répartition des adjacences cassées n'est plus aléatoire sur le génome mais dépendante des difficultés algorithmiques d'assemblage des génomes.

5.3.2 Description du pipeline pour la validation de ADSEQ

Le pipeline que nous avons développé pour valider notre méthode est illustré dans la figure 5.6 et comporte sept étapes :

— l'étape **1/** consiste à sélectionner les données de séquençage et l'assemblage de référence d'une espèce pour laquelle on va simuler la fragmentation du génome ;

— dans l'étape **2/**, on échantillonne une partie des données de séquençage disponible.

— l'étape **3/** effectue un nouvel assemblage de l'espèce sélectionnée à partir de l'échantillon de *reads* avec la méthode d'assemblage MINIA [Chikhi and Rizk, 2013; Salikhov et al., 2014] avec les paramètres par défaut permettant d'effectuer un nouvel assemblage du génome de l'espèce plus fragmenté que l'assemblage de référence. Pour cette étape d'assemblage, l'outil KMERGENIE [Chikhi and Medvedev, 2014] a été utilisé avec les paramètres par défaut pour déterminer la meilleure valeur pour la taille des kmer à utiliser en entrée de MINIA ;

— dans l'étape **4/**, pour permettre la comparaison du nouvel assemblage avec l'assemblage initial du génome de l'espèce sélectionnée, nous avons aligné les contigs du nouvel assemblage sur l'assemblage initial avec BLASTN [Altschup et al., 1990] avec l'algorithme megablast (paramètre "*-task megablast*") et avec une valeur seuil de la *e-value* fixée à 10^{-10}. Pour transférer les annotations de gènes de l'assemblage initial au nouvel assemblage, les contigs MINIA doivent être alignés à une seule position et sans ambiguïté sur l'assemblage initial. Pour assurer ces critères, deux filtres ont été appliqués :
  — le filtre 1 consiste à conserver uniquement les alignements avec une identité et une couverture supérieure ou égale à 90% ;
  — le filtre 2 consiste à ne conserver que les alignements de contigs pour lesquels il y a un seul et unique score d'alignement optimal en identité et couverture.

  De plus, une étape de fusion des contigs est appliquée pour les alignements qui consiste à fusionner les paires de contigs chevauchant un même gène. Cela permet de ne pas éliminer ces gènes de notre analyse et simule une étape de *scaffolding*. Pour finir cette étape **4/**, un troisième et dernier filtre est appliqué et consiste à retirer de l'analyse toutes les familles de gènes pour lesquelles au moins un gène n'est pas aligné sur l'assemblage obtenu avec MINIA.

— l'étape **5/** effectue le *scaffolding* des contigs MINIA avec les données de séquençage appariées pour obtenir des adjacences de *scaffolding* pondérées en entrée de ADSEQ. Cette étape correspond aux étapes **B/, C/ et 5/** de la figure 5.2 (*cf.* sections 5.2.2 et 5.2.3) à l'exception du fait que l'étape d'alignement avec BOWTIE2 est limitée à 50 alignements multiples ;

— l'étape **6/** consiste à exécuter l'algorithme ADSEQ sur le jeu de données ;

— finalement l'étape **7/** consiste à comparer les adjacences prédites par

ADSEQ aux adjacences de l'assemblage de référence qui ne sont pas présentes dans l'assemblage MINIA en calculant les statistiques de rappel et précision.

5.3.3 Comparaison de la capacité de *scaffolding* de ADSEQ, ART-DECO et BESST

Pour évaluer la capacité de l'algorithme ADSEQ à améliorer le *scaffolding* d'espèces actuelles et la comparer à celles de l'algorithme ART-DECO et de la méthode BESST, nous avons effectué des simulations de fragmentations sur trois espèces d'*Anopheles* avec des caractéristiques différentes :

— *Anopheles albimanus* qui est en position externe de l'arbre des 18 espèces d'*Anopheles* ;
— *Anopheles arabiensis* situé en profondeur dans l'arbre avec un grand nombre d'espèces proches (complexe *gambiae*) ;
— *Anopheles dirus* en profondeur dans l'arbre mais avec peu d'espèces proches (*cf.* phylogénie, figure 5.7).

Pour chacune des trois espèces, lors de l'étape **2/** du protocole de validation (*cf.* figure 5.6), nous avons effectué deux échantillonnages des données de séquençage. Un premier où nous avons conservé l'ensemble des *reads* et un deuxième où nous en avons conservé la moitié. Nous avons ensuite effectué les étapes **3/** et **4/** pour l'ensemble des trois espèces avec ces deux échantillons de *reads*. Les statistiques d'assemblage au cours de ces deux étapes sont présentées dans les tables A.2 et A.3 (respectivement p. 192 et 193).

À partir des contigs MINIA annotés par les gènes en sortie de l'étape **4/**, on déduit la liste des adjacences de gènes qui seront considérées par l'algorithme ART-DECO.

L'étape **5/** permet d'obtenir les adjacences de *scaffolding* considérées par ADSEQ en complément des adjacences observées dans l'assemblage MINIA. C'est également au cours de cette étape que les prédictions d'adjacences de BESST sont générées. Il est à noter que les prédictions de nouvelles adjacences de BESST ne sont pas limitées aux adjacences pour lesquelles BESST génère un score de support.

Pour ART-DECO et ADSEQ, l'algorithme DECLONE est combiné à chacun des algorithmes, avec une valeur de kT de 0,1, afin d'attribuer un score aux adjacences en sortie du logiciel DECOSTAR. Le jeu d'adjacences considéré correspond aux adjacences après linéarisation des adjacences (*cf.* section 5.1.2). En amont de cette étape de linéarisation, les adjacences en sortie

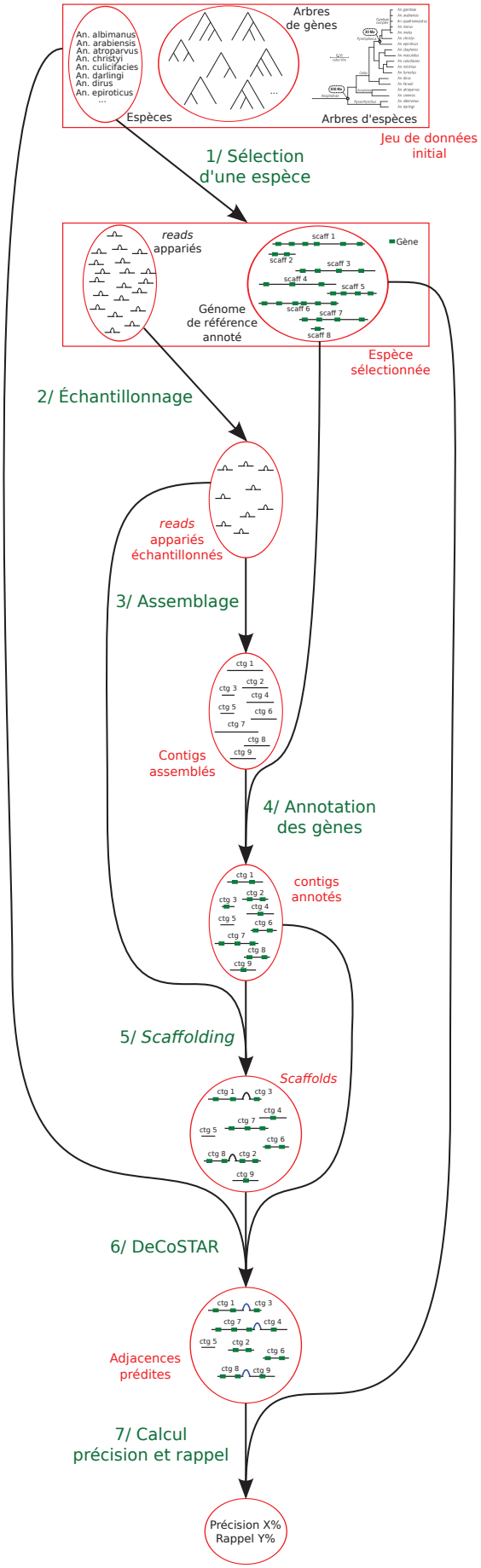


FIGURE 5.6 – Le protocole de validation de l'algorithme ADSEQ.

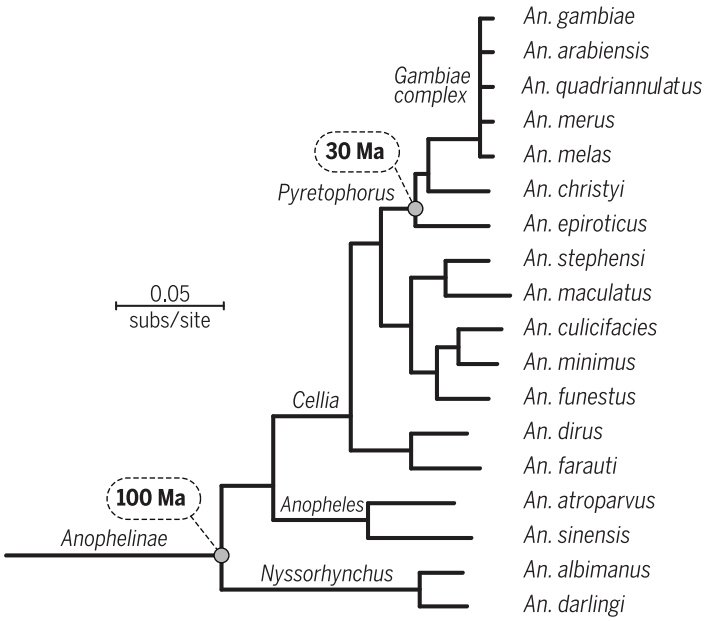


FIGURE 5.7 – Phylogénie des 18 espèces d'*Anopheles*. *From [Neafsey et al., 2015]. Reprinted with permission from AAAS.*

du logiciel DECOSTAR avec un score inférieur à un seuil donné sont éliminées. Dans cette analyse comparative, nous avons utilisé trois valeurs de seuil : $0{,}1$, $0{,}5$ et $0{,}8$.

Nous comparons ensuite l'ensemble des adjacences prédites par les trois méthodes pour les 6 conditions (3 espèces avec deux échantillonnages des *reads* considérés) aux adjacences présentes sur l'assemblage initial des trois espèces. À partir de ces comparaisons, nous déterminons le nombre d'adjacences prédites *fausses négatives* (FN) correspondant aux adjacences présentes dans l'assemblage initial et non présentes dans les adjacences prédites. Les adjacences *vraies positives* (TP) correspondant aux adjacences prédites qui sont présentes dans l'assemblage initial pour lesquelles l'orientation des deux gènes est correcte. Et les adjacences *fausses positives sûres* (CFP) correspondent aux adjacences prédites non présentes dans l'assemblage initial auxquelles on retire les adjacences qui lient deux gènes en extrémités de contigs. Ces dernières peuvent en effet correspondre à de vraies adjacences non présentes dans l'assemblage initial. Nous pouvons alors calculer les statistiques de rappel et de précision :

$$Rappel = \frac{TP}{TP + FN}$$

$$Précision = \frac{TP}{TP + CFP}$$

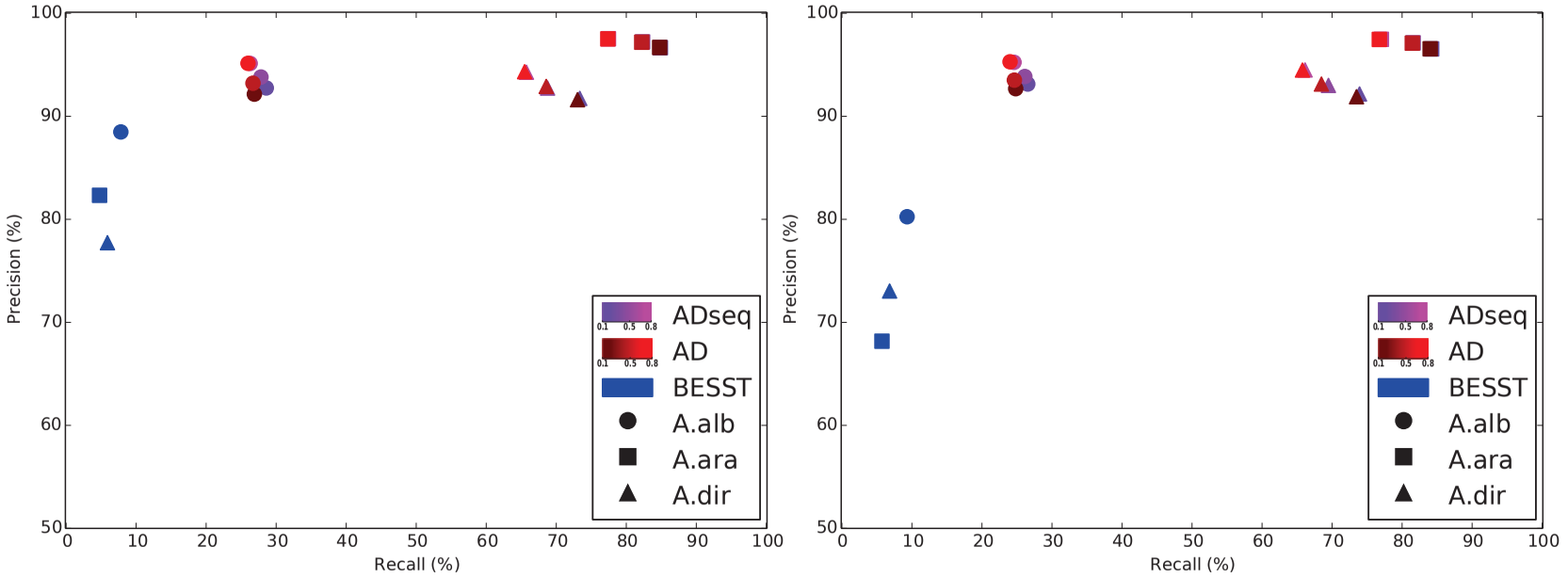


FIGURE 5.8 – Statistiques de précision et rappel des prédictions d'adjacences pour BESST, ART-DECO et ADSEQ, pour trois génomes artificiellement fragmentés (A.alb : *Anopheles albimanus*, A.ara : *Anopheles arabiensis* et A.dir : *Anopheles dirus*). Le dégradé de couleur pour ART-DECO et ADSEQ représente les valeurs de précision et de rappel pour 3 valeurs de seuil lors de l'étape de linéarisation des adajcences prédites. **Graphe de gauche :** résultats avec un échantillonnage de 50% des *reads*. **Graphe de droite :** résultats avec l'ensemble des *reads*.

La figure 5.8 illustre la précision en fonction du rappel pour les prédictions de nouvelles adjacences par les trois méthodes de *scaffolding* (BESST, ART-DECO et ADSEQ) pour les deux échantillonnages de *reads* effectués (50% et 100% des *reads*). Pour ADSEQ et ART-DECO, on a pour chaque échantillonnage de *reads* trois points correspondant aux différentes valeurs de seuil utilisées pour la linéarisation (0,1, 0,5 et 0,8). Les résultats de la figure montre que l'algorithme ADSEQ surpasse la méthode BESST en précision et rappel quelle que soit la valeur de seuil utilisée pour la linéarisation des prédictions d'adjacences. Cela montre que l'ajout d'information phylogénétique dans le *scaffolding* de génomes améliore le rappel sans réduction de la précision comparé à une méthode purement basée sur des données de séquençage. Si nous comparons les prédictions de ADSEQ et ART-DECO, on observe que le rappel est toujours plus élevé avec ADSEQ qu'avec ART-DECO pour toutes les valeurs de seuil et toutes les espèces. Pour ce qui est de la précision, la différence est très faible entre les deux méthodes. En majorité la précision est un peu plus élevée pour ADSEQ (exceptés pour *An. albimanus* avec un seuil de 0,8 pour les deux échantillonnages de *reads*, *An. dirus* avec un seuil de 0,5 pour les deux échantillonnages de *reads* et *An. dirus* avec un seuil de 0,8 pour l'ensemble des *reads*). Pour *Anopheles albimanus*, ADSEQ surpasse ART-DECO pour une valeur de seuil de 0,1 et 0,5 pour les deux échantillonnages de *reads*. D'un point de vue quantitatif, l'ajout de données de séquençage semble avoir un plus faible impact sur les statistiques de précision et de rappel comparé à l'utilisation de l'évolution des adjacences. Cependant, il semble que pour les espèces situées en position externe de l'arbre (comme *Anopheles albimanus*) les données de séquençage ont un impact plus élevé sur la précision et le rappel. Il est toutefois à noter que la combinaison intégrée des apports d'adjacences de *scaffolding* actuelles (données de séquençage et évolution des adjacences) est une importante avancée puisqu'il est souvent difficile d'avoir confiance en des données de *scaffolding* par génomique comparative sans donnée de séquençage. Cela avait d'ailleurs été soulevé lors de la soumission de l'article de ART-DECO par les critiques de l'article.

Pour analyser qualitativement les prédictions de nouvelles adjacences entre les trois méthodes, nous avons généré des diagrammes de Venn pour une valeur de seuil de 0,1 pour la linéarisation des prédictions de ADSEQ et ART-DECO, correspondant au meilleur consensus entre la précision et le rappel pour les trois espèces (un meilleur rappel pour une précision un peu plus faible par rapport aux autres seuils). Les figures 5.9 et 5.10 représentent les

diagrammes de Venn schématisant les adjacences communes entre BESST, ART-DECO et ADSEQ respectivement avec 50% et l'ensemble des *reads*. Si nous considérons ces méthodes individuellement, les résultats montrent que pour les deux échantillonnages de *reads*, ADSEQ surpasse ART-DECO et BESST avec la plus faible quantité d'adjacences FN et la plus grande quantité de TP. Pour la quantité de CFP, c'est BESST qui a la plus faible quantité mais avec un ratio CFP/TP beaucoup plus élevé que ADSEQ et ART-DECO, indiquant que BESST a une précision plus faible. Si on combine les résultats *a posteriori* de ART-DECO et BESST, ils surpassent ADSEQ en terme de rappel avec un nombre de TP plus élevé et un nombre de FN plus faible. Cependant, cette amélioration du rappel se fait au dépend d'une forte baisse de la précision par un nombre élevé de CFP.

Pour illustrer ces conclusions, prenons l'exemple d'*Anopheles albimanus* avec 50% des *reads* dans la figure 5.9. Pour ADSEQ, on a 3.928 adjacences FN, 1.573 adjacences TP et 123 adjacences CFP. Et pour ART-DECO+BESST, on a 3.729 adjacences FN, 1.772 adjacences TP et 182 adjacences CFP. On a donc un rappel de $28{,}59\ \%$ pour ADSEQ et $32{,}21\ \%$ pour ART-DECO+BESST et une précision de $92{,}75\ \%$ pour ADSEQ et $90{,}69\ \%$ pour ART-DECO+BESST. Soit une différence de $-3{,}62$ points du rappel et de $+2{,}06$ de la précision entre ADSEQ et ART-DECO+BESST.

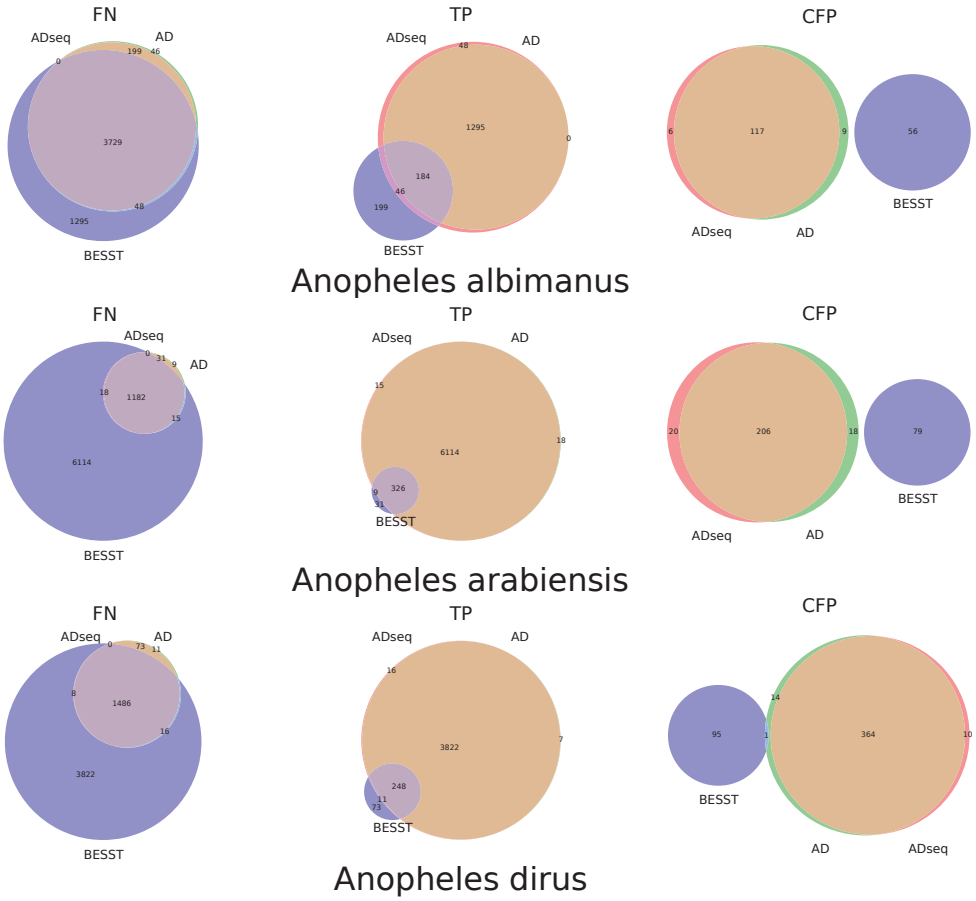


FIGURE 5.9 – Diagrammes de Venn représentant les adjacences partagées par les trois méthodes de *scaffolding* ADSEQ, ART-DECO (AD) et BESST avec l'échantillonnage de *reads* à 50% pour les 3 espèces sélectionnées, par types d'adjacences (*fausses négatives* (FN), *vraies positives* (TP) et *fausses positives sûres* (CFP)).

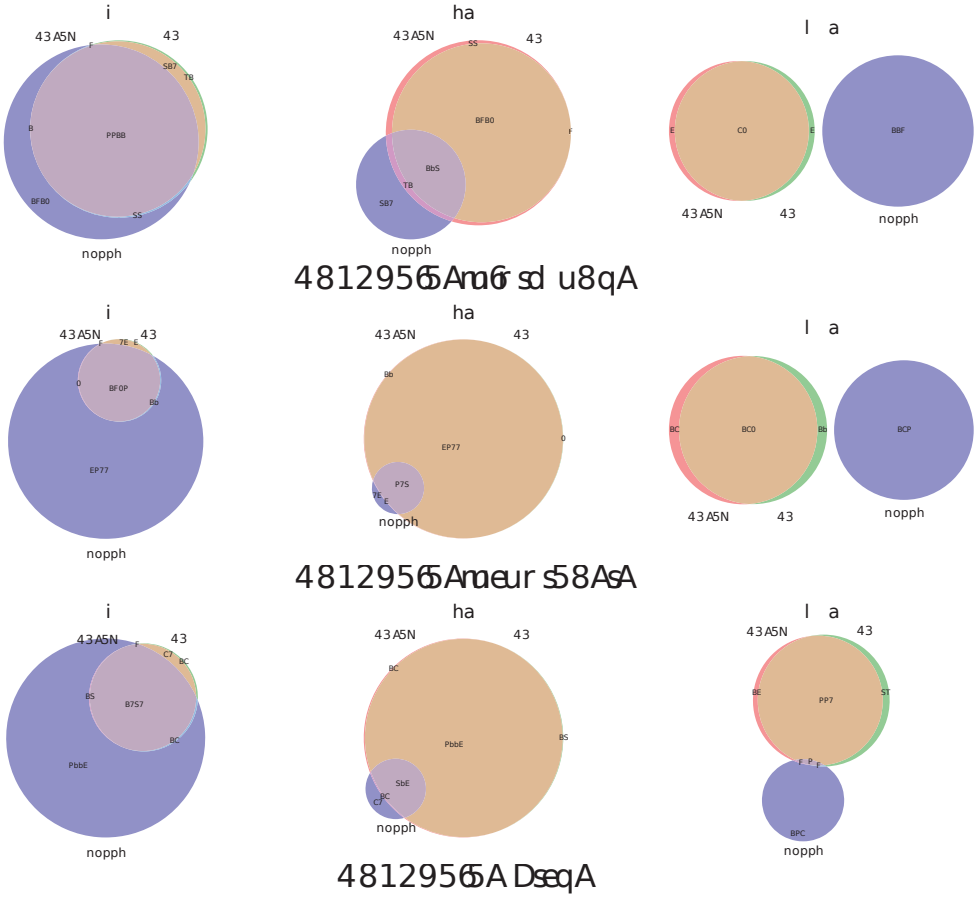


FIGURE 5.10 – Figure équivalente à la figure 5.9 mais sans échantillonnage des *reads*.

En résumé, dans ce chapitre nous avons présenté une nouvelle méthode pour la reconstruction conjointe de l'ordre et de l'évolution de gènes chez des génomes ancestraux et actuels intégrant des données de séquençage appariées pour améliorer cette reconstruction. Nous avons focalisé ce chapitre sur la capacité de la méthode à pouvoir améliorer le *scaffolding* de génomes actuels. Notre méthode utilise deux types de données d'adjacences :

— des données phylogénétiques provenant de la reconstruction de l'évolution d'adjacences au sein d'une phylogénie ;
— des données de séquençage appariées.

Des simulations de fragmentation des génomes ont montré que la méthode a une bonne capacité pour le *scaffolding* de génomes actuels. Les résultats de ces simulations par comparaison avec ceux de ARt-DeCo montrent que la majorité des prédictions d'adjacences sont retrouvées par le signal phylogénétique des adjacences et non par les données de séquençage. Cependant, ces données permettent de pallier la difficulté de notre approche à pouvoir inférer des adjacences chez les espèces en position externe de l'arbre (exemple d'*Anopheles albimanus*) et donnent une plus grande confiance dans les prédictions de nouvelles adjacences.

Nous avons comparé les propriétés de *scaffolding* de notre logiciel DeCoSTAR (avec l'algorithme ADSEQ) à des méthodes de *scaffolding* par génomique comparative et montré que notre approche surpasse l'ensemble des autres méthodes analysées (*cf.* section 2.2.3, p. 38). Nous listons ci-dessous les principales propriétés de *scaffolding* de notre méthode :

— possibilité d'applications répétées sur de grands génomes d'eucaryotes ;
— application à des génomes multichromosomaux (c.-à-d. non limitée à des génomes unichromosomaux) ;
— utilisation de multiple génomes de référence ;
— utilisation de la phylogénie des espèces pour pondérer les signaux de synténies en fonction du degré de parenté entre les espèces ;
— possibilité d'intégrer des données de séquençage appariées, mais celles-ci ne sont pas nécessaires contrairement à RACA, CLA et AlignGraph ;
— possibilité pour les génomes de référence d'être incomplètement assemblés ;
— l'ensemble des génomes de la phylogénie étudiée servent à la fois de génomes de référence et de génomes cibles (c.-à-d. amélioration du *scaffolding* chez tous les génomes).

Aucune des méthodes de *scaffolding* par génomique comparative analysées dans la section 2.2.3 ne remplie l'ensemble de ces critères. Cependant, la méthode de GOS-ASM, développée au cours de ce projet de thèse, permet comme notre méthode de conjointement reconstruire l'histoire évolutive de l'ordre de marqueurs génomiques et d'améliorer la restitution de l'ordre de ces marqueurs chez les génomes actuels. Cependnat, GOS-ASM ne permet pas l'incorporation de données de séquençage appariées, est limitée à des jeux de données composés d'un faible nombre d'espèces et n'intègre pas les événements de duplications et pertes de gènes/marqueurs génomiques dans son modèle de reconstruction des génomes ancestraux.

Chapitre 6

Intégration des extensions de DeCo : DeCoSTAR

Dans ce chapitre, nous présentons le logiciel DeCoSTAR et les différents algorithmes implémentés dans celui-ci. Les algorithmes présents dans le logiciel DeCoSTAR permettent de reconstruire l'ordre des gènes chez les génomes ancestraux et l'histoire évolutive de cet ordre de gènes le long de la phylogénie par reconstruction d'histoires évolutives d'adjacences de gènes. DeCoSTAR permet également d'améliorer le *scaffolding* des génomes actuels en se servant du signal d'adjacences présent dans les histoires d'adjacences inférées par la méthode et peut également utiliser les signaux d'adjacences complémentaires [1] pouvant être intégrés dans les formules de récurrence de programmation dynamique de DeCoSTAR.

Dans une première section, nous présentons le contexte dans lequel le logiciel a été développé. Dans la deuxième section, nous détaillons l'organisation du logiciel DeCoSTAR avec un schéma illustrant les différentes étapes de fonctionnement du logiciel. Enfin, dans une troisième section, nous présentons succintement les différents algorithmes présents dans DeCoSTAR et les paramètres propres à ceux-ci. Un cas d'utilisation des différents paramètres sera présenté sur un jeu de données dont nous présentons les résultats dans les chapitres 7 et 8.

6.1 Historique du logiciel DeCoSTAR

Au cours de ma thèse, j'ai travaillé en parallèle d'un autre étudiant en thèse, Wandrille Duchemin, sur des extensions de l'algorithme DeCo Bérard et al. [2012]. Les différents projets d'extensions de la méthode DeCo ont été

1. obtenues par d'autres méthodes de *scaffolding* par exemple

faits en grande partie indépendamment les uns des autres (algorithmes DeCoLT Patterson et al. [2013], DeClone Chauve et al. [2014], ART-DeCo Anselmetti et al. [2015] et ADSeq). Un objectif commun a été de produire un programme unique permettant d'intégrer l'algorithme DeCo et tous ses dérivés.

Un travail de ré-implémentation des algorithmes DeCo, DeCoLT et DeClone a d'abord été entrepris par Wandrille, en collaboration avec les auteurs de ces trois algorithmes. Un des objectifs était d'intégrer la distribution de probabilité de Boltzmann de l'algorithme DeClone (*cf.* section 4.2.3, p. 80) dans les formules de récurrence de l'algorithme DeCoLT. Cette réimplémentation a permis de corriger des erreurs présentes dans le code original de DeCoLT et d'implémenter l'algorithme DeClone en langage C++ [2]. Cela a permis d'unifier les objets et fonctions informatiques pour modéliser et manipuler des données biologiques communes entre les deux algorithmes. Cette unification a été facilitée par l'utilisation de la librairie Bio++ Guéguen et al. [2013] [3] dans laquelle un grand nombre d'objets et de foncions phylogénétiques sont définies. En plus des algorithmes ART-DeCo et DeClone, une collaboration entre Wandrille, Edwin Jacox et Celine Scornavacca a permis une intégration de l'algorithme de réconciliation d'arbres de gènes ecceTERA Jacox et al. [2016]. Nous avons été invités à intégrer le code des algorithmes ART-DeCo et ADSeq dans ce logiciel afin de permettre d'ajouter un mode de "*scaffolding* des génomes actuels" dans le logiciel. L'implémentation de ce logiciel, nommé DeCoSTAR, a impliqué la majorité des acteurs ayant participé à l'élaboration des algorithmes de la *galaxie* DeCo (algorithmes DeCo, DeCoLT, DeClone, ART-DeCo et ADSeq) et a fait l'objet d'une publication dans le journal *Genome Biology and Evolution* Duchemin et al. [2017]. La dernière version stable du logiciel DeCoSTAR est disponible à l'adresse web suivante : `http://pbil.univ-lyon1.fr/software/DeCoSTAR/get.html` et un dépôt gitHub du logiciel, pour obtenir la version en cours de développement, est disponible à l'adresse suivante : `https://github.com/WandrilleD/DeCoSTAR`.

2. le code original était implémenté en python.
3. librairie utilisée pour l'implémentation originale de l'algorithme DeCo.

6.2 Schéma général du logiciel DECOSTAR

L'implémentation commune de l'ensemble des outils de la *galaxie* DECO dans le logiciel DECOSTAR est une étape logique car l'ensemble des algorithmes utilisent un même format de jeu de données d'entrée, nécessaire à l'exécution de l'algorithme DECO, et ont tous l'objectif commun de reconstruire l'histoire coévolutive de deux marqueurs génomiques à partir de l'histoire évolutive de chaque marqueur génomique, dans le cadre d'une phylogénie d'espèces. Cette implémentation commune permet également de mieux gérer la compatibilité entre les différents algorithmes, de simplifier le maintien du code et d'avoir une meilleure visibilité en regroupant l'ensemble des utilisations des algorithmes de la *galaxie* DECO dans un seul et unique logiciel.

Le schéma général de fonctionnement du logiciel DECOSTAR est présenté dans la figure 6.1 (p. 126). Comme on peut le voir, le logiciel prend en entrée un arbre des espèces avec les assemblages génomiques composés de *scaffolds* contenant les gènes présents dans les arbres de gènes considérés.

6.2.1 Réconciliation des arbres de gènes (étape 1/)

La première étape du logiciel DECOSTAR consiste à réconcilier les arbres de gènes en entrée du logiciel et permet également de raciner les arbres de gènes s'ils ne le sont pas avec l'algorithme ECCETERA. Cette étape du logiciel DECOSTAR est optionnelle si les arbres de gènes sont déjà réconciliés et racinés. Sur la figure 6.1, nous avons représenté des arbres de gènes réconciliés avec un modèle d'évolution considérant les duplications et pertes de gènes et pas les transferts horizontaux de gènes, bien que cela soit tout à fait possible.

6.2.2 Création des familles d'adjacences (étape 2/)

La deuxième étape du logiciel DECOSTAR consiste à regrouper les adjacences entre gènes actuels en familles d'adjacences homologues. Deux adjacences $x_1 \sim x_2$ et $y_1 \sim y_2$ sont homologues si x_1 et y_1 (resp. x_2 et y_2) ont un ancêtre commun a_1 (resp. a_2) tels que a_1 et a_2 sont sur différents arbres de gènes ou, s'ils sont sur le même arbre de gènes, l'un n'est pas l'ancêtre de l'autre Bérard et al. [2012]. Cette relation est transitive et partitionne l'ensemble des adjacences en familles d'adjacences homologues.

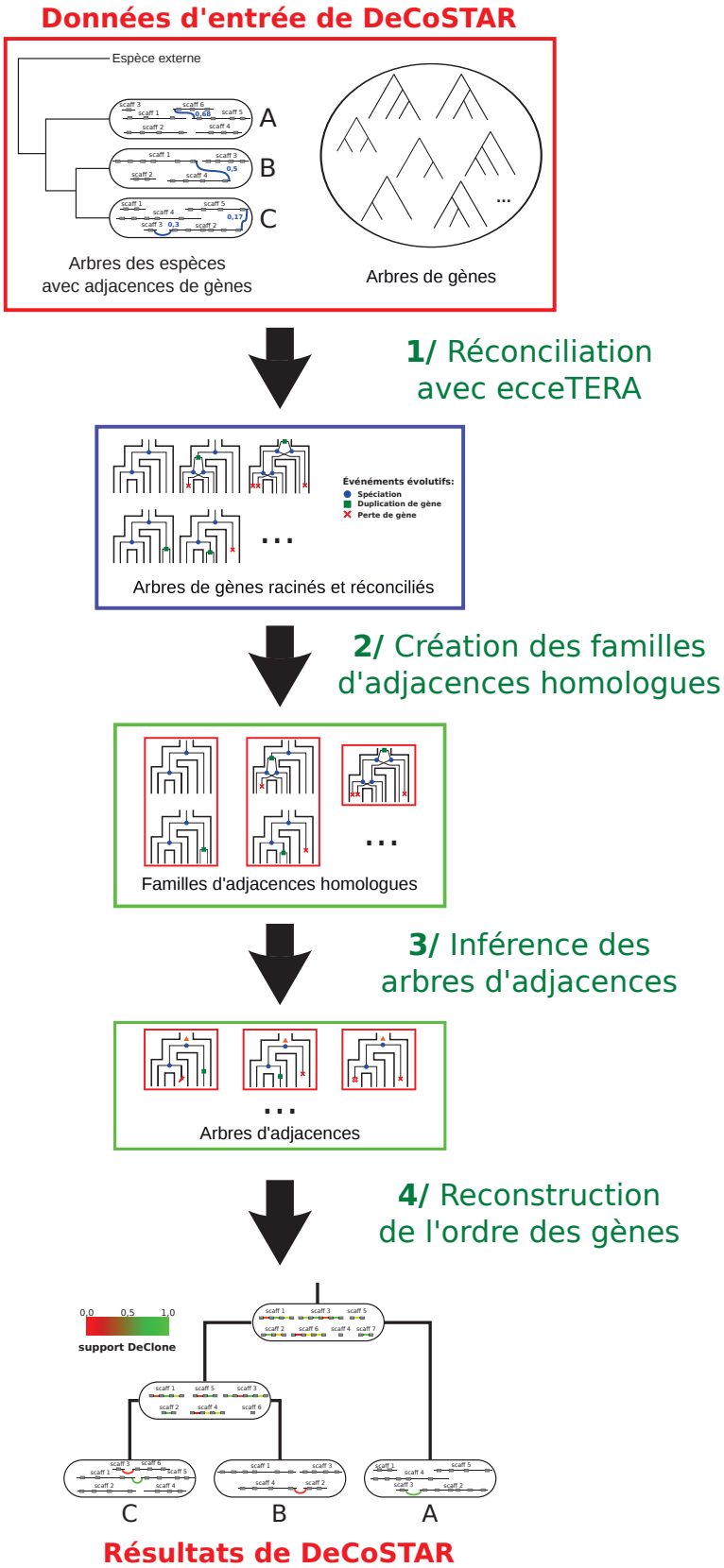


FIGURE 6.1 – Schéma de fonctionnement du logiciel DeCoSTAR. Dans les données d'entrée de DeCoSTAR : les génomes sont schématisés par une bulle aux feuilles de l'arbre des espèces. Sur ces génomes, les gènes sont représentés par des rectangles **gris** qui sont liés les uns aux autres par des liens **noirs**, représentant des adjacences observées dans l'assemblage des génomes, et éventuellement des liens **bleus** représentant des adjacences de *scaffolding* complémentaires pondérées par un score compris dans [0,1].

6.2.3 Inférence des arbres d'adjacences de gènes (étape 3/)

Pour chaque famille d'adjacences homologues, l'étape **3/** de DECOSTAR infère une histoire évolutive des adjacences, qui minimise le nombre de créations et cassures d'adjacences pondérées par leurs coûts respectifs, sous la forme d'un arbre d'adjacences de gènes (*cf.* section 1.3.3, p. 21). Pour cette étape, différentes combinaisons d'algorithmes peuvent être appliquées en fonction des paramètres donnés pour la reconstruction de l'histoire évolutive de chaque famille d'adjacences homologues :

- pour le choix du modèle de parcimonie, on a le choix entre deux algorithmes :
  1. l'algorithme DECO pour un modèle considérant les duplications et pertes de gènes (modèle par défaut) ;
  2. l'algorithme DECOLT pour un modèle considérant les duplications, pertes et transferts horizontaux de gènes.
- l'algorithme DECLONE peut être utilisé pour permettre une exploration de l'espace des solutions plus ou moins parcimonieuses [4]. Cette exploration permet de fournir un support aux adjacences ancestrales et actuelles inférées par DECOSTAR, correspondant au ratio d'observation de ces adjacences dans les scénarios échantillonnés par DECLONE ;
- on peut choisir le mode "*scaffolding* des génomes actuels", on a alors deux algorithmes :
  - l'algorithme ART-DECO qui permet de prendre en compte la fragmentation des génomes et d'inférer des adjacences entre gènes localisés aux extrémités des *scaffolds* des assemblages initiaux ;
  - l'algorithme ADSEQ qui permet en plus de ce que fait ART-DECO de considérer des adjacences de *scaffolding* pondérées par un score compris dans [0,1].

6.2.4 Reconstruction de l'ordre des gènes (étape 4/)

La dernière étape de DECOSTAR consiste à associer l'ensemble des arbres d'adjacences inférés, afin d'obtenir l'ordre des gènes chez les génomes ancestraux. En mode "*scaffolding* des génomes actuels", on obtient en plus la prédiction de nouvelles adjacences actuelles. Les adjacences prédites sont pondérées par des supports si l'algorithme DECLONE a été utilisé. La partie basse

4. sans DECLONE, on se limite à une solution parcimonieuse pour chaque famille de gènes.

de la figure 6.1 illustre les résultats obtenus avec DeCoSTAR avec une utilisation des algorithmes ADSEQ et DeClone. On observe que de nouvelles adjacences sont proposées pour les génomes actuels dont la couleur indique le support DeClone, les adjacences présentes dans les assemblages initiaux sont en noir. L'ensemble des adjacences ancestrales ont également un support DeClone avec le même code couleur.

6.3 Paramètres propres aux algorithmes inclus dans DeCoSTAR

Dans cette section, nous présentons succinctement les différents algorithmes implémentés dans le logiciel DeCoSTAR et les paramètres du logiciel propres à chacun des algorithmes. Pour l'ensemble des algorithmes disponibles dans le logiciel DeCoSTAR, il faut renseigner les trois types de données nécessaires à la reconstruction de l'histoire évolutive d'adjacences de gènes :

— la phylogénie des espèces considérées par DeCoSTAR au format newick (paramètre *species.file*) ;
— les arbres de gènes au format newick, *NHX*[5] ou *ALE*[6] Szöllősi et al. [2013, 2015] (paramètre *gene.distribution.file*)
— la liste des adjacences de gènes présentes sur les génomes (paramètre *adjacencies.file*).

À la fin de cette section, nous présentons une ligne de commande pour l'exécution du logiciel DeCoSTAR sur un jeu de données de 18 espèces de moustiques, dont nous analysons les résultats dans les chapitres 7 et 8.

6.3.1 Algorithme ecceTERA

ecceTERA est un algorithme de réconciliation parcimonieuse d'arbres de gènes avec l'arbre des espèces utilisant un modèle d'évolution prenant en compte des événements de duplications, pertes et transferts latéraux de gènes. À l'ensemble des gènes ancestraux d'un arbre de gènes, ecceTERA associe une espèce et un événement évolutif (spéciation, duplication, perte ou transfert horizontal). Dans l'algorithme ecceTERA, un événement de transfert horizontal de gène est subdivisé en deux parties :

5. *New Hampshire X*
6. *Amalgamated Likelihood Estimation*

1. une spéciation sortante dont l'espèce associée est l'espèce *donneuse* du gène transféré ;
2. une spéciation entrante dont l'espèce associée est l'espèce *réceptrice* du gène transféré.

L'algorithme ECCETERA est un algorithme de programmation dynamique qui permet de réconcilier un arbre de gènes composé de n gènes actuels avec un arbre d'espèces composé de m espèces actuelles avec une complexité de temps de calcul de $O(n^2m)$.

L'algorithme permet également de raciner des arbres de gènes en explorant l'ensemble des enracinements possibles et en choisissant celui minimisant le score de réconciliation (c.-à-d. le nombre de duplications, pertes et transferts horizontaux de gènes).

Pour utiliser l'algorithme ECCETERA, il est nécessaire de renseigner différents paramètres pour régler son exécution. Tout d'abord, il est important d'indiquer si les arbres de gènes doivent être réconciliés et racinés (paramètres *already.reconciled* et *rooted*). Concernant le modèle de parcimonie utilisé par ECCETERA, on précise si l'on considère les transferts horizontaux de gènes (paramètre *with.transfer*) et on fixe la valeur des coûts de duplications, pertes et transferts de gènes (paramètres *dupli.cost*, *loss.cost* et *HGT.cost*). Enfin, on détermine une valeur correspondant au poids du signal apporté par la topologie des arbres de gènes dans le calcul du score global de DECOSTAR (paramètre *Topology.weight*). Le détail des paramètres utilisés par ECCETERA est présenté en annexe p. 182.

6.3.2 Algorithmes de la *galaxie* DECO

DECO est l'algorithme de base du logiciel DECOSTAR et les paramètres nécessaires au réglage de son exécution sont utilisés par les autres algorithmes de la *galaxie* DECO. Pour l'approche parcimonieuse de DECO consistant à minimiser le nombre de créations et de cassures d'adjacences, il est nécessaire de fixer les coûts de ces événements évolutifs (paramètres *AGain.cost* et *ABreak.cost*). L'activation du paramètre *all.pair.equivalence.class* permet de calculer des histoires évolutive d'adjacences entre tous les couples d'arbres de gènes, même si ceux-ci ne partagent aucune adjacence. Lors du calcul des histoires évolutives des adjacences, on détermine deux valeurs indiquant le poids du signal apporté par les adjacences de gènes (paramètre *Adjacency.weight*)

et celui apporté par la réconciliation des arbres de gènes (paramètre *Reconciliation.weight*) dans le calcul du score global de DeCoSTAR. Lors du calcul de la matrice des coûts c_0 et c_1 d'une famille d'adjacences homologues (*cf.* section 3.3.3, p. 58 et Bérard et al. [2012]), il peut arriver que les coûts c_0 et c_1 de la racine soient égaux. Il faut donc définir une probabilité de choisir de préférence le scénario c_1 [7] au scénario c_0 [8] (paramètre *C1.Advantage*). Il est également possible d'indiquer à DeCoSTAR que l'on souhaite inférer une création d'adjacences de gènes à la racine de l'ensemble des arbres d'adjacences inférés par le logiciel (paramètre *always.AGain*).

Récemment, une nouvelle option permettant de fortement réduire les conflits synténiques [9] a été implémentée par Adelme Bazin, étudiant de Master en stage au LBBE [10]. Ce paramètre permet d'économiser un coût de création d'adjacence lorsqu'un gène ou plusieurs gènes contigus sont perdus (paramètres *Loss.aware* et *Loss.iteration*). Sans cette option, on compte un coût de création d'adjacences entre les gènes, entourant le ou les gènes contigus perdus, qui se retrouvent adjacents bien que cela ne corresponde pas à une modification de l'ordre des gènes. Le détail des paramètres utilisés par DeCo (et donc l'ensemble des algorithmes de la *galaxie* DeCo) est présenté en annexe p. 183.

L'utilisation de l'algorithme DeCoLT est déterminée par le choix de considérer les transferts de gènes latéraux lors de la réconciliation des arbres de gènes par ecceTERA (paramètre *with.transfer*). Si on a en entrée de DeCoSTAR un arbre des espèces daté, on peut indiquer à DeCoLT d'utiliser les intervalles de temps de l'arbre pour reconstruire les histoires évolutives d'adjacences de gènes (paramètre *bounded.TS*).

Algorithme DeClone

L'algorithme DeClone, permettant d'échantillonner des solutions d'arbres d'adjacences dans l'espace des solutions, est décrit dans la section 4.2.3 (p. 79). Pour utiliser l'algorithme DeClone, il est nécessaire d'activer l'option *use.boltzmann* de DeCoSTAR. Pour paramétrer son utilisation, il faut indiquer le nombre

7. où les deux gènes de la racine de la famille d'adjacences sont adjacents.
8. où les deux gènes de la racine de la famille d'adjacences ne sont pas adjacents.
9. gènes ayant plus de deux voisins non compatibles avec des choromosomes linéaires/circulaires.
10. Laboratoire de Biométrie et Biologie Évolutive.

de solutions que l'on veut échantillonner dans l'espace des solutions (paramètre *nb.sample*) et fixer la température kT qui permet de centrer la distribution de Boltzmann dans l'espace des solutions parcimonieuses pour des valeurs de kT proche de 0 (paramètre *boltzmann.temperature*) Chauve et al. [2014]. Le détail des paramètres utilisés par DECLONE est présenté en annexe p. 184.

Algorithmes ART-DECO et ADSEQ/ mode *scaffolding*

Les algorithmes ART-DECO et ADSEQ sont les algorithmes utilisés pour améliorer le *scaffolding* des génomes actuels et permettre une meilleure inférence de l'ordre des gènes ancestraux par la prise en compte d'un plus grand nombre d'adjacences de gènes. La description de l'algorithme ART-DECO est détaillée dans le chapitre 4 et celle de l'algorithme ADSEQ dans le chapitre 5.

Pour exécuter l'algorithme ART-DECO, il faut activer le mode "*scaffolding* des génomes actuels" (paramètre *scaffolding.mode*). Pour l'utilisation de ce mode, il est nécessaire de fournir un fichier indiquant le nombre de chromosomes attendus pour l'ensemble des espèces considérées (paramètre *chromosome.file*). La valeur du paramètre *SPI*[11] Duchemin et al. [2017] doit également être fixée et correspond à la taille du clade c dans lequel il est possible d'inférer de nouvelles adjacences homologues à une adjacence observée chez une espèce, même si celle-ci n'a pas d'homologue dans c et que le plus proche homologue est en position d'*outgroup* du clade c (paramètre *scaffolding.propagation.index*). Par défaut, les adjacences actuelles prédites par ART-DECO ont un coût nul pour le critère de parcimonie, le paramètre *absence.penalty* permet de définir un coût pour ces adjacences considérées lors de la reconstruction des histoires évolutives d'adjacences.

Pour l'exécution de l'algorithme ADSEQ, les adjacences de gènes données en entrée de DECOSTAR doivent contenir des adjacences de *scaffolding* pondérées par un score compris dans $[0,1]$, générées par une ou plusieurs méthodes de *scaffolding* indépendantes. Pour ajuster le traitement de ces adjacences de *scaffolding* par l'algorithme ADSEQ, il est nécessaire de fixer la base du log, b_{scaff}, utilisée pour le calcul des coûts c_0 et c_1 des adjacences de *scaffolding* (*cf.* section 5.1.2, p. 98). Si on veut que les adjacences de *scaffolding* soient considérées pour compter le nombre de *scaffolds* dans les assemblages initiaux, il faut activer le paramètre *scaffold.includes.scored.adjs*. La valeur b_{scaff}

11. *Scaffolding Propagation Index*

est utilisée lors des calculs des coûts c_0 et c_1 entre les gènes localisés aux extrémités des *scaffolds* initiaux (*cf.* section 4.2.1, p. 71).

Divers paramètres sont également nécessaires afin d'ajuster les données de sortie de l'algorithme DeCoSTAR, le détail de ces paramètres est disponible en annexe à la page 184.

6.3.3 Exemple d'exécution du logiciel DeCoSTAR

Dans les chapitres 7 et 8, nous appliquons le logiciel DeCoSTAR avec les algorithmes ADSeq et DeClone et le fichier de paramètres de la figure 6.2 (nom du fichier : *parameter_file_Xtopo+scaff.txt*). Pour exécuter le logiciel DeCoSTAR sur le jeu de données des 18 espèces de moustiques avec ce fichier de paramètre, on utilise la ligne de commande suivante :

```
./DeCoSTAR parameter.file="parameter_file_Xtopo+scaff.txt"
```

Les trois premières lignes du fichier de paramètre correspondent aux fichiers de données du jeu des 18 *Anopheles* obtenus avec le pipeline de génération des données de DeCoSTAR (*cf.* section 5.2, p. 101).

Le troisième bloc nous indique que nous réconcilions et racinons les arbres de gènes avec l'algorithme ecceTERA avec un modèle de parcimonie considérant les duplications et les pertes de gènes qui ont respectivement une valeur de coût de 2 et 1 dans l'algorithme ecceTERA.

Le quatrième bloc fixe la valeur du coût de création à 3 et celle du coût de cassure d'adjacence à 1. Nous avons fixé un poids identique pour les signaux de topologie des arbres de gènes (*Topology.weight*) et des adjacences de gènes (*Adjacency.weight*) pour le calcul du score global de DeCoSTAR (*cf.* thèse de Wandrille Duchemin [Duchemin, 2017]).

Le cinquième bloc donne les paramètres pour le mode "*scaffolding* des génomes actuels" qui est activé. Nous avons choisi une valeur du paramètre *SPI* égale à 21 afin de permettre de propager une adjacence à l'ensemble des espèces de la phylogénie des 18 *Anopheles*.

Le sixième bloc indique que nous avons utilisé l'algorithme DeClone et qu'il analyse 100 solutions dans l'espace des solutions échantillonnées par la distribution de Boltzmann avec une température kT de 0,1 indiquant un échantillonnage centrée sur les solutions parcimonieuses.

Enfin, le dernier bloc indique les différents fichiers générés en sortie de l'exécution de l'algorithme DeCoSTAR.

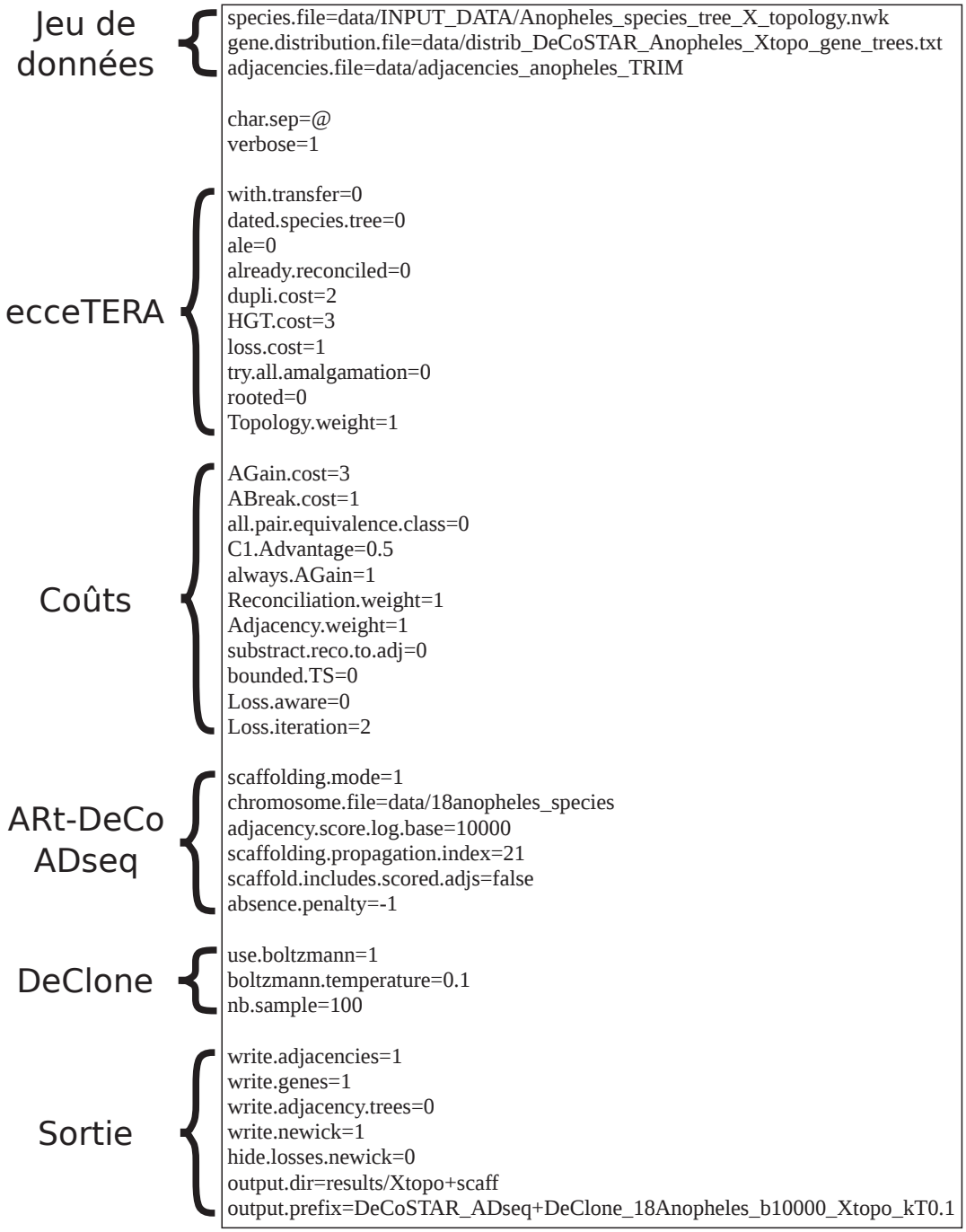


FIGURE 6.2 – Exemple d'un fichier de paramètres pour l'exécution du logiciel DECOSTAR sur un jeu de données de la phylogénie de 18 espèces de moustiques du genre *Anopheles*.

Chapitre 7

Étude de l'assemblage de 18 génomes d'*Anopheles*

Dans ce chapitre et le suivant (chapitre 8), nous présentons les résultats que nous avons obtenus sur un jeu de données composés de 18 moustiques du genre *Anopheles* avec le logiciel DECOSTAR. Dans la première section de ce chapitre, nous présentons le contexte général de l'étude [Neafsey et al., 2015] et la méthode employée dans cette étude pour générer l'assemblage et l'annotation génomique des 18 espèces d'*Anopheles*. Puis nous présentons les paramètres et algorithmes du logiciel DECOSTAR employés pour l'analyse de ce jeu de données. Dans la deuxième section, nous présentons les résultats de *scaffolding* des génomes actuels et ancestraux de la phylogénie des 18 *Anopheles* obtenus avec le logiciel DECOSTAR.

Afin de faciliter la lecture de ces chapitres, nous nommons *assemblages de référence*, les assemblages de génomes produits par Neafsey et al. [2015] et *assemblages initiaux* ceux générés à partir du pipeline de génération des données de DECOSTAR de la section 5.2. De la même manière, nous nommerons *scaffolds de référence*, les *scaffolds* produits par l'assemblage de Neafsey et al. [2015] et *scaffolds initiaux*, les *scaffolds* générés à partir du pipeline de génération des données. Par abus de langage, nous utilisons aussi le terme de *scaffolds* pour définir les ensembles d'ordres de gènes générés en sortie du logiciel DECOSTAR pour les génomes actuels et ancestraux bien que ADSEQ+DECLONE ne permet de déterminer la séquence génomique mais uniquement l'ordre relatif de gènes orientés.

7.1 Présentation du matériel et des méthodes

7.1.1 Projet de séquençage de moustiques du genre *Anopheles*

Contexte général

L'étude et la compréhension de la biologie des espèces de moustiques, et plus particulièrement du genre *Anopheles*, est un enjeu crucial de santé publique. En effet, une partie de ces moustiques sont les seuls vecteurs de parasites du genre *Plasmodium* responsables du paludisme [Rutledge et al., 2017]. Un rapport de l'OMS [1] datant de 2015 indique que cette année là, 212 millions personnes étaient affectées par le paludisme et 429.000 décès causés par cette maladie ont été recensés. Le séquençage et l'étude du génome d'un échantillon de ces moustiques a été fait au cours des années 2010 dans le cadre du projet *Anopheles 16 Genomes Project* [Neafsey et al., 2013] dont les résultats ont fait l'objet de deux publications dans la revue *Science* [Neafsey et al., 2015; Fontaine et al., 2015]. Le projet a eu pour but de séquencer 16 nouveaux génomes de moustiques du genre *Anopheles*, afin de pouvoir établir les facteurs biologiques permettant d'expliquer les différences de capacité vectorielle des parasites *Plasmodium* entre les 18 génomes d'*Anopheles* disponibles à la suite de ce projet (les génomes d'*An. gambiae* [Holt et al., 2002] et d'*An. darlingi* [Marinotti et al., 2013] ayant précédemment été séquencés).

Séquençage et assemblage de référence

Pour l'ensemble des 16 espèces d'*Anopheles* nouvellement séquencées, plusieurs librairies de séquençage, impliquant différentes tailles d'insert, ont été utilisées. Pour l'ensemble des espèces, deux librairies de séquençage ont été produites à partir d'un moustique femelle. Une librairie *Paired-End (PE)* avec une taille d'insert de $180\ pb$ en orientation FR ($\rightarrow\leftarrow$) (nommée *'fragment' library*) et une librairie *Mate-Pair (MP)* avec une taille d'insert de $1{,}5\ kpb$ en orientation RF ($\leftarrow\rightarrow$) (nommée *'jump' library*). Pour 11 des 16 espèces, une troisième librairie avec une taille d'insert d'environ $38\ kpb$ en orientation FR (nommée *'fosill' library* [Williams et al., 2012]) a été produite à partir d'un échantillon d'une centaine de moustiques pour améliorer le *scaffolding* des génomes (*cf.* section 2.1.2, p. 29, pour plus d'informations sur les données de

1. Organisation Mondiale de la Santé.

séquençage appariées). L'ensemble des informations sur les données de séquençage appariées pour le jeu *Anopheles* sont disponibles dans la table A.1 (p. 191).

À partir de ces données de séquençage, les auteurs ont assemblé l'ensemble des 16 génomes avec l'outil d'assemblage ALLPATHS-LG [Gnerre et al., 2011] qui dans une première étape, établit l'assemblage *de novo* du génome, puis effectue une étape de *scaffolding* avec les informations de *reads* appariés. Les *scaffolds* de référence obtenus ont ensuite été analysés par l'outil de *gap filling* PILON [Walker et al., 2014] permettant de déterminer une partie de la séquence d'ADN manquante entre les contigs présents dans un même *scaffold*. Les assemblages de référence ainsi obtenus ont ensuite été annotés par le pipeline de prédiction de gènes de la base de données VectorBase utilisant le logiciel MAKER [Cantarel et al., 2008]. Des données de transcriptomique ont été également produites pour 12 des 16 espèces, correspondant aux 11 espèces avec données de séquençage *'fosill'* plus l'espèce *An. epiroticus*. Ces données ont été assemblées avec l'algorithme d'assemblage TRINITY [Grabherr et al., 2011] et alignés sur les *scaffolds* de référence précédemment produits afin d'améliorer la prédiction de gènes pour ces 12 espèces. Les statistiques de l'assemblage de référence des 18 génomes d'*Anopheles* (taille des génomes, statistiques N50, nombre de gènes) sont disponibles dans la table 7.1.

Pour l'ensemble des génomes d'*Anopheles*, l'haplotype est composé de cinq bras chromosomiques (2R, 2L, 3R, 3L et X). Il n'y a pas de chromosome Y dans ces génomes car seules des femelles ont été séquencées[2]. Les statistiques des assemblages de référence montrent une grande hétérogénéité dans la complétion de l'assemblage et l'annotation des génomes. *An. albimanus* est l'espèce nouvellement séquencée la mieux assemblée avec un génome de référence composé de 240 *scaffolds* dont 49 contiennent des gènes et *An. maculatus* l'espèce la moins bien assemblée avec 47.797 *scaffolds* de référence dont 14.835 avec des gènes. Le nombre de *scaffolds* des *scaffolds* de référence avec gènes est important car notre étude se base sur les familles de gènes, et considère donc seulement les *scaffolds* contenant des gènes présents dans ces familles de gènes.

2. à l'exception d'*An. gambiae* pour lequel un mâle a été séquencé mais aucun gène n'a été annoté sur le chromosome Y dans nos données.

Nom d'espèce	Taille de l'assemblage (pb)	*scaffolds* (tous)		*scaffolds* avec gène(s)			
		#*scaffolds*	N50 (pb)	#*scaffolds*	N50		#gènes
					pb	#gènes	
An. albimanus	170.508.315	204	18.068.499	57	18.068.499	1.212	11.911
An. arabiensis	246.567.867	1.214	5.604.218	340	5.830.121	348	13.162
An. atroparvus	224.290.125	1.371	9.206.694	476	9.206.694	655	13.776
An. christyi	172.658.580	30.369	9.057	5.173	17.016	3	10.738
An. culicifacies	202.998.806	16.162	22.320	5.715	32.742	4	14.335
An. darlingi	134.715.017	2.160	115.168	2.161	115.168	10	10.457
An. dirus	216.307.690	1.266	6.906.475	302	7.656.907	543	12.781
An. epiroticus	223.486.714	2.673	366.526	1.052	417.110	29	12.078
An. farauti	180.984.331	550	1.196.527	376	1.235.781	84	13.217
An. funestus	225.223.604	1.392	671.960	619	702.105	46	13.344
An. gambiae	273.093.681	7	49.364.325	6	49.364.325	2.867	12.810
An. maculatus	141.894.015	47.797	3.841	12.776	4.751	1	14.835
An. melas	227.407.517	20.281	18.041	8.855	21.239	2	16.149
An. merus	251.805.912	2.753	342.196	1.078	391.600	886	13.887
An. minimus	201.793.324	678	10.313.149	142	10.313 149	886	12.560
An. quadriannulatus	283.828.998	2.823	1.641.272	647	1.794.736	95	13.349
An. sinensis	241.390.279	11.270	80.738	3.536	103.937	9	14.791
An. stephensi	225.369.006	1.110	837.295	502	851.727	57	13.113
Somme	3.844.323.781	144.080		43.813			237.293
Moyenne	213.573.543	8.004		2.434			13.183

TABLE 7.1 – Statistiques des assemblages de référence des 18 génomes d'*Anopheles*.

7.1.2 Paramètres et algorithmes du logiciel DeCoSTAR utilisés sur le jeu de données des 18 *Anopheles*

Dans cette section, nous présentons les résultats de *scaffolding* des génomes actuels et ancestraux de la phylogénie des 18 *Anopheles* obtenus avec le logiciel DeCoSTAR avec les algorithmes ADSeq et DeClone, que nous noterons ADSeq+DeClone. La figure 7.1 représente une schématisation du protocole utilisé pour appliquer le logiciel DeCoSTAR sur le jeu de données des 18 *Anopheles* produit par Neafsey et al. [2015]. Nous décrivons en détail chaque étape dans les paragraphes qui suivent.

Transformation du jeu de données de référence des 18 *Anopheles* en jeu de données initial de DeCoSTAR (étape I/)

Le jeu de données des 18 *Anopheles* de Neafsey et al. [2015] contient :

- pour chacune des 18 espèces d'*Anopheles* :
  - les données de séquençage appariées ;
  - l'assemblage de référence du génome ;
  - l'annotation des gènes sur le génome ;
  - la séquences des gènes annotés.
- la phylogénie des 18 *Anopheles* (*cf.* figure 5.7, p. 114), pour laquelle deux topologies, X et WG, existent (*cf.* section 8.1.2, p. 162) ;
- 14.981 arbres de gènes contenant une part importante des gènes annotés sur les 18 génomes d'*Anopheles*.

Le jeu de données est traité par le pipeline de génération des données d'entrée de DeCoSTAR (*cf.* section 5.2, p. 101 et figure 5.2, p. 103). Pour l'ensemble des expériences décrites dans la suite de ce manuscrit, deux jeux de données d'entrée de DeCoSTAR ont été générés :

- un jeu avec la phylogénie X des 18 *Anopheles* ;
- un jeu avec la phylogénie WG des 18 *Anopheles*.

Nous avons généré un jeu de données pour chacune des deux topologies afin de comparer les résultats et nous débattons de la topologie la plus probable dans le chapitre 8 (p. 161). Il est à noter que les 14.940 arbres de gènes inférés avec ProfileNJ [Noutahi et al., 2016] (*cf.* section 5.2.1, p. 105 et figure 5.5, p. 106) sont différents entre ces deux jeux de données car la minimisation des duplications et pertes est dépendant de la topologie de l'arbre des espèces. Par contre, le contenu en *scaffolds initiaux*, en gènes, en adjacences observées et adjacences de *scaffolding* est identique entre les deux jeux car le contenu en gènes des deux jeux d'arbres reste le même. Un troisième jeu de données a

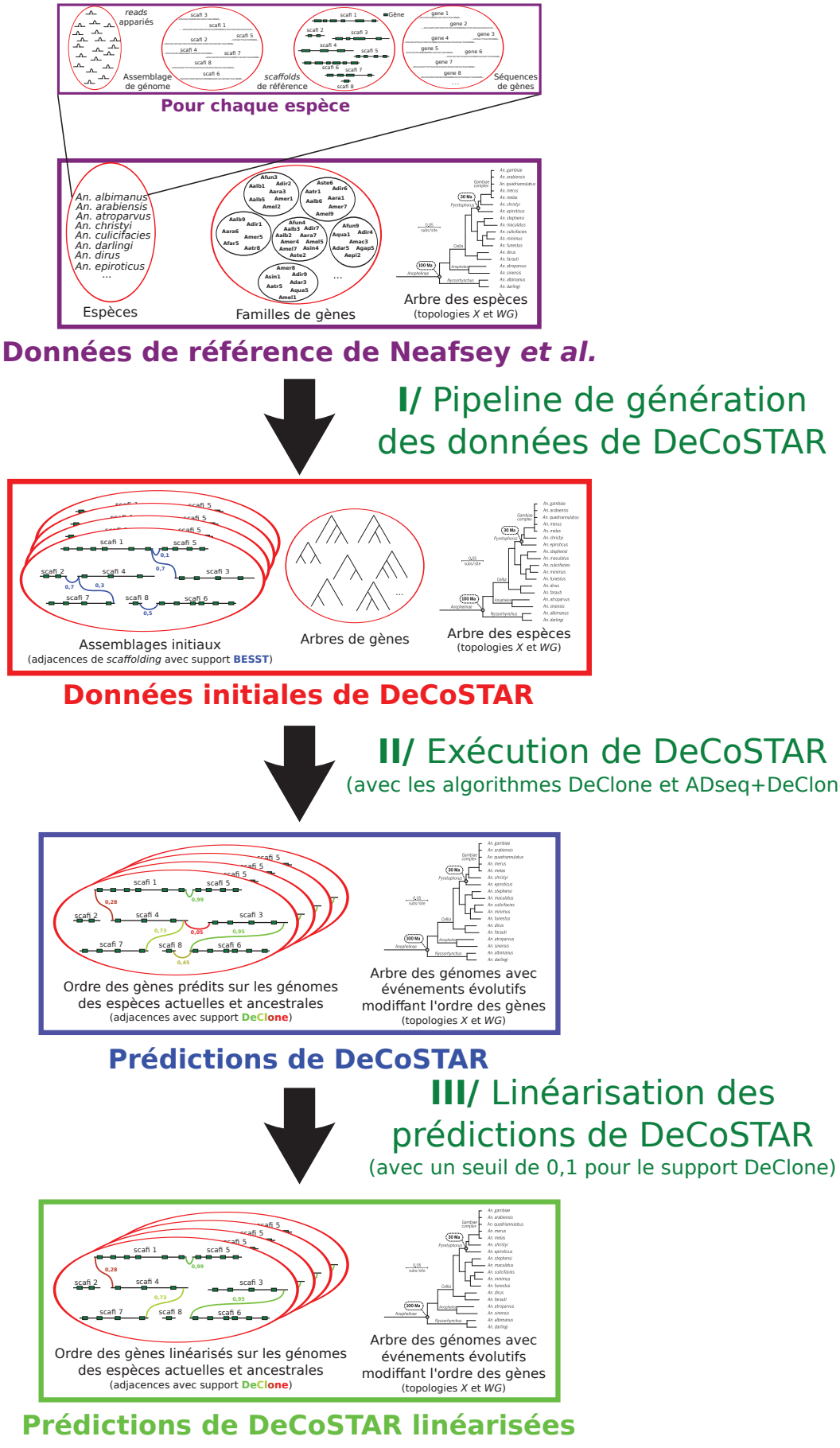


FIGURE 7.1 – Schéma représentant l'application du logiciel DECOSTAR sur le jeu de données des 18 *Anopheles*.

également été généré à partir des arbres de gènes générés par Neafsey et al. [2015], nommés arbres *bruts*. Ce jeu d'arbres est composé de 14.981 arbres de gènes mais a été réduit aux 14.940 correspondant aux arbres/familles de gènes en sortie de notre pipeline d'inférence d'arbres de gènes. Ce jeu a été produit afin de comparer la qualité des arbres de gènes produits par notre pipeline d'inférence d'arbres de gènes à la qualité des arbres de gènes *bruts* (*cf.* section 7.2.1, p. 143).

Les statistiques (nombre de *scaffolds* initiaux, N50, contenu en gènes) des assemblages initiaux considérés en entrée de DECOSTAR sont présentés dans les colonnes 2 à 5 de la table 7.2 (p. 148). En annexe, sont disponibles les distributions du score des adjacences de *scaffolding* inférées par BESST avec le pipeline décrit dans la section 5.2.2 (p. 107) et considérées en entrée de ADSEQ+DECLONE sur l'ensemble des espèces et par espèce (*cf.* figures A.1, p. 199, et A.2, p. 200).

Paramètres d'exécution du logiciel DECOSTAR (étape II/)

Sur les trois jeux de données initiaux que nous venons de décrire, quatre exécutions de DECOSTAR (étape **II/**) sont effectuées :

1. avec les algorithmes ADSEQ+DECLONE, la phylogénie X et les arbres *bruts* ;
2. avec l'algorithme DECLONE, la phylogénie du X et les arbres de gènes inférés par PROFILENJ ;
3. avec les algorithmes ADSEQ+DECLONE, la phylogénie X et les arbres de gènes inférés par PROFILENJ ;
4. avec les algorithmes ADSEQ+DECLONE, la phylogénie WG et les arbres de gènes inférés par PROFILENJ ;

Comme annoncé dans la section précédente, les résultats de la première exécution seront comparés aux résultats de la troisième exécution afin de confirmer la meilleure qualité des arbres de gènes inférés par notre pipeline avec PROFILENJ (*cf.* section 7.2.1, p. 143).

Les résultats de la deuxième exécution seront comparés à la troisième exécution afin d'évaluer l'apport du *scaffolding* produit par ADSEQ pour l'amélioration du *scaffolding* des génomes actuels et la reconstruction de l'ordre des gènes ancestraux (l'algorithme DECLONE seul permettant uniquement de reconstruire l'ordre des gènes ancestraux sans les informations apportées par les adjacences de *scaffolding* et sans effectuer le *scaffolding* des génomes actuels) (*cf.* section 7.2.2, p. 145).

Enfin, les résultats des troisième et quatrième exécutions seront utilisés afin d'étudier l'amélioration du *scaffolding* des génomes actuels avec les algorithmes ADSEQ+DECLONE avec les deux topologies identifiées pour la phylogénie des 18 *Anopheles*. La comparaison des résultats de ces deux exécutions permettra également d'établir laquelle des deux topologies est la "vraie" phylogénie des 18 espèces d'*Anopheles* (*cf.* section 8.1, p. 161).

Pour l'ensemble de ces exécutions, les fichiers de paramètres utilisés par DECOSTAR sont disponibles en Annexe (p. 186-189).

Linéarisation des prédictions d'adjacences actuelles et ancestrales (étape III/)

Après exécution du logiciel DECOSTAR, les adjacences actuelles et ancestrales prédites sont linéarisées avec le protocole de linéarisation décrit dans la section 5.1.2. Pour l'ensemble des expériences, nous avons choisi un seuil de support DECLONE de 0,1, indiquant que toutes les adjacences avec un support DECLONE inférieur ou égale à 0,1 sont retirées avant linéarisation des adjacences prédites restantes. Cette valeur a été choisie car elle correspond au meilleur compromis que nous avons identifié entre la précision et le rappel des adjacences actuelles prédites lors de la validation de l'algorithme ADSEQ, pour le jeu de données des 18 *Anopheles* (*cf.* figure 5.6, p. 113).

Dans la suite de l'analyse et afin d'alléger la lecture, nous ne préciserons plus que c'est le logiciel DECOSTAR qui est utilisé pour les exécutions des algorithmes ADSEQ+DECLONE et DECLONE. De plus, toutes les statistiques sur les adjacences prédites par DECLONE ou ADSEQ+DECLONE seront les statistiques après linéarisation des prédictions.

7.2 Réalisations sur le jeu de données *Anopheles*

Les résultats que nous avons obtenus sur le jeu de données des 18 *Anopheles* sont organisés en trois parties.

Une première partie où nous comparons la qualité des arbres de gènes *bruts* obtenus par Neafsey et al. [2015] à la qualité des arbres de gènes inférés avec PROFILENJ.

Une deuxième partie où nous présentons le *scaffolding* des génomes ancestraux et actuels obtenus par les algorithmes ADSEQ+DECLONE et DECLONE.

Une troisième partie où nous comparons les adjacences prédites par ADSEQ+DECLONE à des données de *scaffolding* indépendantes (carte génétique et données de séquençage PacBio) pour l'espèce *An. funestus*.

7.2.1 Comparaison des arbres de gènes *bruts* avec les arbres de gènes inférés avec PROFILENJ

Neafsey et al. [2015] ont inféré des arbres de gènes à partir de familles de gènes disponibles dans la base de données OrthoDBmoz2 (`http://cegg.unige.ch/orthodbmoz2`). Dans un premier temps, nous avons exécuté ADSEQ+DECLONE sur le jeu de données *Anopheles* (topologie X) et le jeu d'arbres de gènes *bruts*. Cependant, des analyses statistiques ont montré que ces arbres contenaient un nombre anormalement élevé de duplications de gènes. Comme cela a été montré dans [Hahn, 2007], des erreurs lors de l'inférence d'arbres de gènes introduisent un nombre important de duplications de gènes ancestraux et produisent des génomes plus grands qu'attendus.

Nous avons donc fait le choix de développer notre propre pipeline d'inférence d'arbres de gènes afin d'améliorer ces arbres (*cf.* section 5.2.1, p. 105) à partir des familles de gènes homologues obtenus par Neafsey et al. [2015]. La différence majeure de notre pipeline d'inférence d'arbres de gènes avec celui de Neafsey et al. [2015] est l'utilisation du logiciel PROFILENJ. Ce dernier permet de réduire le nombre de duplications de gènes ancestraux. Cette réduction s'effectue en modifiant la topologie de l'arbre par minimisation du nombre de pertes et duplications de gènes pour les nœuds n'étant pas soutenus par l'ensemble des topologies d'arbres échantillonnées par RAXML.

Les résultats montrent de très fortes différences entre les deux jeux d'arbres. Par exemple, le jeu d'arbres *bruts* est composé de 39.194 duplications de gènes tandis que le jeu d'arbres PROFILENJ en contient 6.461. Pour illustrer les différences entre ces deux jeux d'arbres, la figure 7.2 présente deux statistiques d'intérêt sur les génomes ancestraux reconstruits par ADSEQ+DECLONE. Sur le graphique gauche, les boîtes à moustaches représentent le nombre de gènes présents dans les génomes actuels initiaux (colonne 1) et dans les génomes ancestraux inférés par ADSEQ+DECLONE, avec les arbres *bruts* (colonne 2) et PROFILENJ (colonne 3). La figure montre que pour le jeux d'arbres *bruts* certains génomes ancestraux peuvent atteindre un nombre de gènes supérieur à 30.000, ce qui est trois fois plus que la moyenne des génomes actuels. Le nombre de gènes chez les génomes ancestraux pour les arbres PROFILENJ est plus proche de l'attendu, qui est du même ordre que le nombre de

gènes chez les génomes actuels, soit environ 10.000 gènes.

Le graphique de droite illustre le degré des gènes ancestraux, qui est utilisé comme une statistique de linéarité. Dans cette analyse, il est important de savoir que les prédictions faites par ADSEQ+DECLONE n'ont pas été linéarisées afin de comparer le degré de linéarité des génomes ancestraux prédits en sortie de ADSEQ+DECLONE pour les deux jeux d'arbres. La distribution des degrés des gènes ancestraux a été élaboré comme suit : le degré d'un gène est la somme des supports d'adjacences DECLONE de toutes les adjacences impliquant ce gène. La distribution de degrés des gènes pour un assemblage de génome complet et linéaire est illustrée par la courbe noire (tous les gènes ont un degré 2 exceptés les gènes localisés en extrémités de chromosome qui ont un degré 1). Le graphique montre que la distribution des degrés des gènes ancestraux obtenus avec le jeu d'arbres PROFILENJ (courbe rouge) est plus proche de la courbe idéale que la distribution avec les arbres *bruts* (courbe bleue).

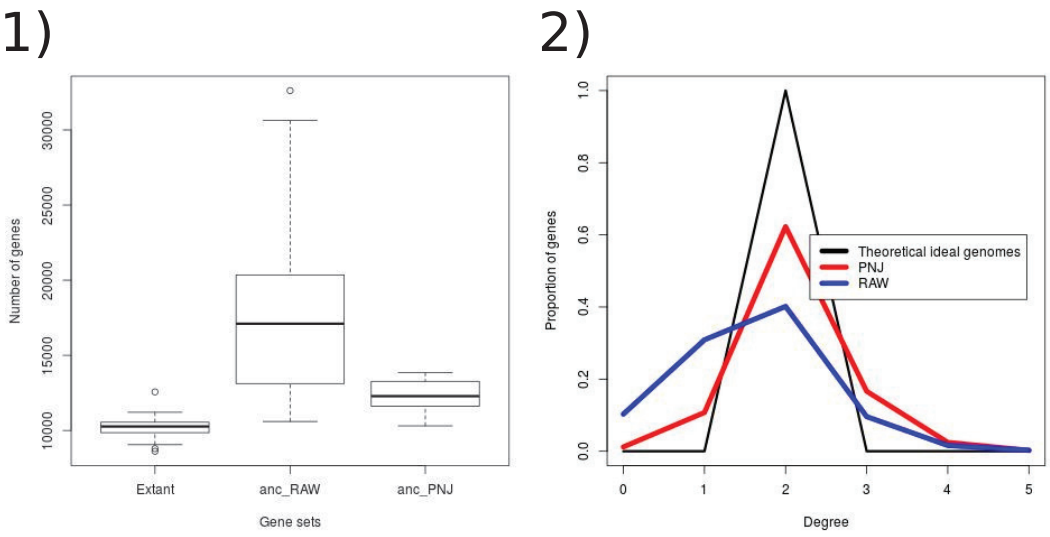


FIGURE 7.2 – **1)** Nombre de gènes chez les génomes actuels (gauche), les génomes ancestraux inférés par ADSEQ+DECLONE avec les arbres de *bruts* (milieu) et les arbres PROFILENJ (droite). **2)** Distribution des degrés de gènes ancestraux inférés par ADSEQ+DECLONE avec les arbres de gènes *bruts* (courbe bleue) et avec les arbres inférés avec PROFILENJ (courbe rouge), comparé à la distribution des degrés de gènes attendue pour des assemblages de génomes théoriques parfaits (courbe noire). La valeur de la coordonnée x correspond à la somme de toutes les valeurs comprises dans l'intervalle $[x, x+1[$.

Les deux métriques analysées plaident en faveur d'une meilleure inférence des génomes ancestraux par ADSEQ+DECLONE avec le jeu d'arbres PROFILENJ qu'avec le jeu d'arbres *bruts*. Cependant, les résultats montrent

des différences pour la distribution des degrés de gènes entre le scénario idéal et les génomes ancestraux inférés par ADSEQ+DECLONE avec le jeu de données PROFILENJ. Celles-ci peuvent s'expliquer par la multiplicité des scénarios parcimonieux lors de la réconciliation ou de l'étape de correction des arbres par PROFILENJ, par l'introgression de gènes qui n'est pas considérée dans le modèle *DL*, par des artefacts phylogénétiques (branches incorrectes mais avec de forts supports de *bootstrap*), des erreurs lors de l'alignement multiple de séquences ou le *clustering* de gènes en familles homologues mais également par l'inférence erronée d'adjacences ancestrales inexistantes par ADSEQ+DECLONE.

7.2.2 *Scaffolding* des espèces actuelles et ancestrales

Dans cette section, nous présentons les résultats du *scaffolding* des espèces actuelles et ancestrales de la phylogénie des 18 *Anopheles* obtenues avec ADSEQ+DECLONE en deux parties.

Une première partie où nous présentons les statistiques générales du *scaffolding* conjoint des génomes actuels et ancestraux et de l'apport de la capacité de *scaffolding* des génomes actuels de ADSEQ+DECLONE sur la reconstruction des génomes ancestraux ;

Puis, une deuxième partie où nous présentons de manière détaillée l'amélioration du *scaffolding* du génome des espèces actuelles d'*Anopheles*.

Statistiques globales de *scaffolding* des génomes d'*Anopheles* et de reconstruction des génomes ancestraux

La figure 7.3 illustre les résultats de *scaffolding* sur le jeu de données des 18 *Anopheles*, dans trois conditions différentes :

— une première condition, nommée *X-scaff*, où DECLONE seul est utilisé avec la topologie X de l'arbre des espèces, il n'y a donc pas de *scaffolding* des génomes actuels ;
— une seconde condition, nommée *X+scaff*, où ADSEQ+DECLONE sont appliqués avec la topologie X de l'arbre des espèces ;
— une troisième condition, nommée *WG+scaff*, où ADSEQ+DECLONE sont appliqués avec la topologie WG de l'arbre des espèces.

La partie droite de la figure 7.3 montre que la capacité de *scaffolding* (ADSEQ) réduit fortement le nombre de *scaffolds* chez les génomes actuels avec un

nombre médian de *scaffolds* de 550 *scaffolds* pour les génomes initiaux, diminuant à 296,5 *scaffolds* actuels en sortie de ADSEQ+DECLONE avec la topologie X et 299,5 *scaffolds* avec la topologie WG. De plus, le *scaffolding* des génomes actuels produits par ADSEQ+DECLONE permet l'obtention de génomes ancestraux moins fragmentés que ceux reconstruits avec DECLONE seul : un nombre médian de *scaffolds* ancestraux de 2.194 avec DECLONE et de 1.930 avec ADSEQ+DECLONE et la topologie X (1.783 avec la topologie WG). Cette amélioration de la reconstruction de l'ordre des gènes ancestraux permet d'effectuer une analyse plus fine de leur évolution et une détection plus complète des événements de réarrangements chromosomiques le long de la phylogénie des 18 *Anopheles* que nous décrivons dans le chapitre 8.

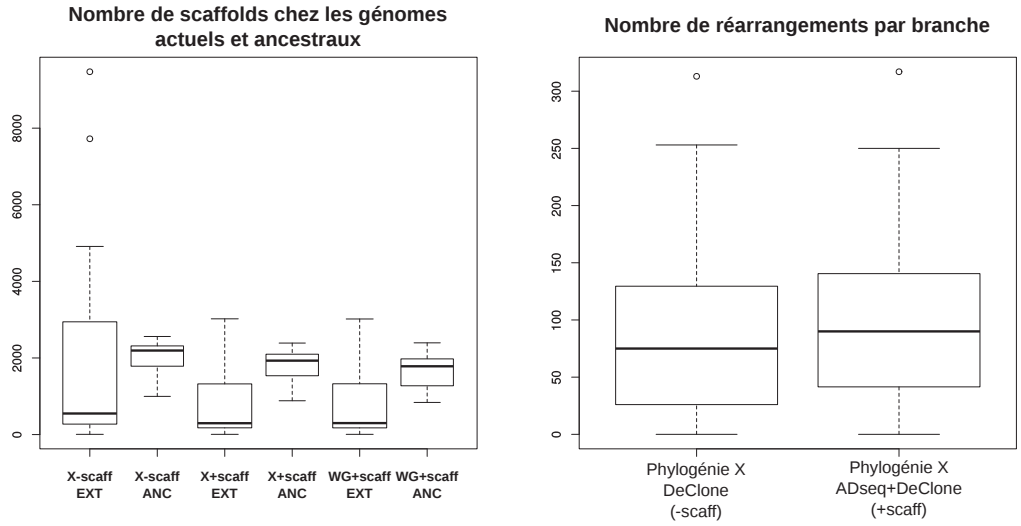


FIGURE 7.3 – Statistiques globales de *scaffolding* des génomes actuels et ancestraux. Le terme EXT désigne les *scaffolds* provenant de génomes actuels et le terme ANC les *scaffolds* provenant de génomes ancestraux.

On s'intéresse ensuite au nombre de réarrangements détectés le long de la phylogénie des 18 *Anopheles* avec DECLONE seul et avec ADSEQ+DECLONE (*cf.* section 8.2.6, p. 169 pour la méthode de détection des réarrangements chromosomiques). Ceci afin de déterminer si la capacité de *scaffolding* (ADSEQ) permet de détecter plus de réarrangements chromosomiques. Sur la partie droite de la figure 7.3, on observe que l'on retrouve un nombre significativement plus élevé (test de Wilcoxon apparié, $p-value < 10^{-4}$ [Wilcoxon, 1945]) de réarrangements chromosomiques avec le *scaffolding* des génomes actuels (ADSEQ+DECLONE) que sans (DECLONE). Ce nombre supplémentaire de réarrangements chromosomiques détectés sera utile lors de l'évaluation des

topologies X et WG par l'analyse des réarrangements chromosomiques effectué dans la section 8.2.6 (p. 169).

Le *scaffolding* conjoint des génomes actuels et ancestraux est donc bénéfique à la fois pour la reconstruction de l'ordre des gènes actuels et ancestraux mais également pour la reconstruction plus complète de l'histoire évolutive des réarrangements chromosomiques.

Analyse de l'amélioration du *scaffolding* des génomes actuels

La table 7.2 présente les statistiques de l'amélioration du *scaffolding* des génomes actuels (nombre de *scaffolds*, N50, nombres d'adjacences prédites). On observe qu'à partir des 36.634 *scaffolds* présents dans les assemblages initiaux des 18 génomes d'*Anopheles*, ADSEQ+DECLONE assemble les génomes actuels en 13.525 *scaffolds* avec la topologie X et 13.484 avec la topologie WG. Ces nouveaux *scaffolds* ont un nombre moyen de 136 gènes (pour les deux topologies) alors qu'il y avait 124 gènes par *scaffold* avant exécution de ADSEQ+DECLONE.

L'amélioration du *scaffolding* des génomes actuels par ADSEQ+DECLONE est illustrée graphiquement dans les figures 7.4 pour la topologie du X et 7.5 pour la topologie WG. Ces figures représentent, pour chaque espèce en ordonnée, le nombre de *scaffolds* dans les génomes initiaux et le nombre de *scaffolds* actuels en sortie de ADSEQ+DECLONE. La statistique de pourcentage d'amélioration du *scaffolding* est obtenue à partir de la formule suivante :

$$\frac{C_{init} - C_{DS}}{C_{init} - p} \tag{7.1}$$

où C_{init}, C_{DS} et p correspondent respectivement au nombre de *scaffolds* dans l'assemblage initial du génome, au nombre de *scaffolds* actuels après ADSEQ+DECLONE et au nombre attendu de chromosomes. Cette statistique est représentée par le diamètre du cercle pour chaque espèce. L'axe des ordonnées représente la somme des scores des adjacences en sortie de ADSEQ+DECLONE qui ont été retirées du *scaffolding* lors de l'étape de linéarisation. Le choix de cette statistique sur l'axe des ordonnées est utilisé pour comparer les deux topologies (*cf.* section 8.2.4). Dans ces graphiques, l'espèce *An. gambiae* n'est pas représentée car aucune adjacence n'a été prédite par ADSEQ+DECLONE.

Nom d'espèce	Assemblages initiaux des génomes (avant ADSEQ+DECLONE)				Assemblages prédits des génomes (après ADSEQ+DECLONE et linéarisation)							
					Topologie X				Topologie WG			
	#*scaffolds*	N50		#gènes	#*scaffolds*	N50		#adj. prédites (#adj. *scaffolding*)	#*scaffolds*	N50		#adj. prédites (#adj. *scaffolding*)
		pb	#gènes			pb	#gènes			pb	#gènes	
An. albimanus	49	18.068.499	916	9.056	47	18.068.499	916	2 (2)	47	18.068.499	916	2 (2)
An. arabiensis	273	5.830 121	321	10.298	215	9.217.108	410	58 (14)	214	9.972.103	464	59 (14)
An. atroparvus	345	9.206.694	512	10.400	306	10.083.987	647	39 (12)	307	10.083.987	647	38 (12)
An. christyi	4.731	17.384	2	8.792	1.395	94235	12	3.336 (204)	1.407	93.817	12	3.324 (207)
An. culicifacies	4.912	34.064	3	11.213	1.338	202.550	18	3.575 (1.365)	1.338	201.618	18	3.575 (1.366)
An. darlingi	1.951	118.843	9	8.617	1.264	197.190	14	687 (NA)	1.264	197.677	14	687 (NA)
An. dirus	231	7.656.907	406	9.883	176	17.377.229	778	55 (10)	175	17.377.229	778	56 (10)
An. epiroticus	963	425.117	24	9.855	368	1.582.816	80	595 (7)	367	1.679.961	80	596 (7)
An. farauti	349	1.235.781	64	10.239	155	2.736.123	163	194 (152)	154	2.736.123	163	195 (152)
An. funestus	562	703.988	36	10.077	231	2.772.343	127	331 (112)	233	2.673.183	127	329 (111)
An. gambiae	6	49.364.325	2.339	10.324	6	49.364.325	2.339	0 (NA)	6	49.364.325	2.339	0 (NA)
An. maculatus	9.473	5.042	1	10.552	3.022	29.597	7	6.451 (298)	3.018	30.980	7	6.455 (295)
An. melas	7.723	21.730	2	12.567	2.679	95.368	9	5.044 (162)	2.636	96.083	9	5.087 (163)
An. merus	997	400.239	23	10.736	419	1.183.618	64	578 (391)	406	1.260.898	65	591 (399)
An. minimus	114	10.313.149	682	9.792	96	17.164.539	801	18 (7)	96	17.164.539	801	18 (7)
An. quadriannulatus	538	1.846.441	74	10.289	287	5.492.301	230	251 (63)	292	4.868.888	229	246 (61)
An. sinensis	2.944	109.624	7	10.962	1.323	291.628	20	1.621 (480)	1.326	292.486	20	1.618 (475)
An. stephensi	473	851.727	44	10.028	198	2.792.811	136	275 (164)	198	2.772.062	120	275 (162)
Somme	36.634			183.680	13.525			23.110 (3.443)	13.484			23.151 (3.443)
Moyenne	2.035			10.204	751			1.284 (215)	749			12.86 (215)

TABLE 7.2 – Statistiques de *scaffolding* des 18 génomes d'*Anopheles* actuels en entrée et en sortie d'ADSEQ+DECLONE avec les topologies X et WG. Le jeu d'entrée de ADSEQ+DECLONE est composé de 14.940 arbres de gènes et 68.876 adjacences de *scaffolding* (*cf.* figures A.1 et A.2 pour une illustration de la distribution des scores des adjacences de *scaffolding* inférées par BESST).

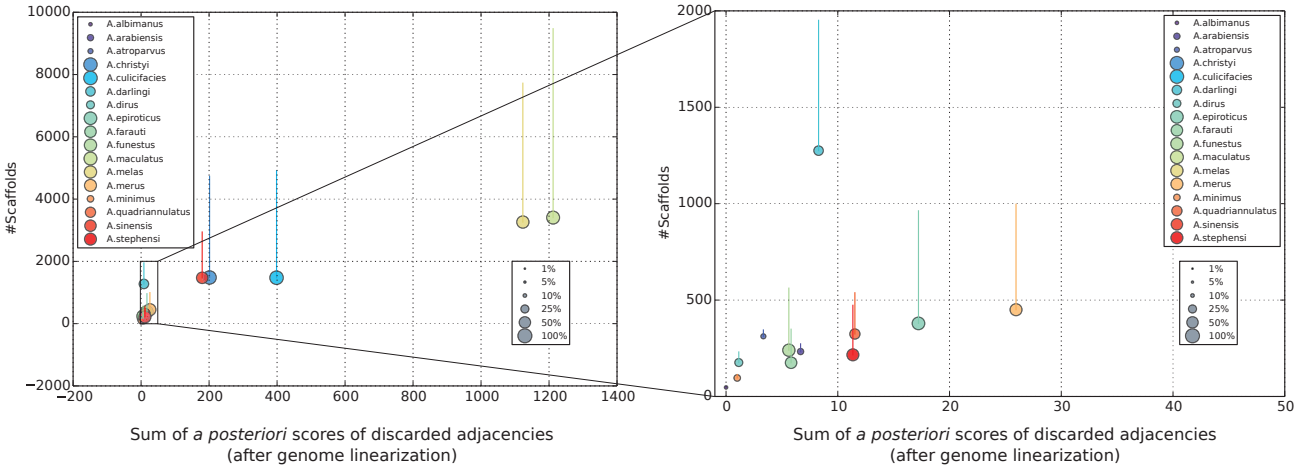


FIGURE 7.4 – Graphique représentant l'amélioration du *scaffolding* des génomes actuels par ADSEQ+DECLONE avec la topologie X. Pour chaque espèce, l'extrémité haute de la barre verticale correspond au nombre de *scaffolds* dans l'assemblage initial et l'extrémité basse au nombre de *scaffolds* après ADSEQ+DECLONE. Le diamètre du cercle est proportionnel au pourcentage d'amélioration du *scaffolding* de chaque génome.

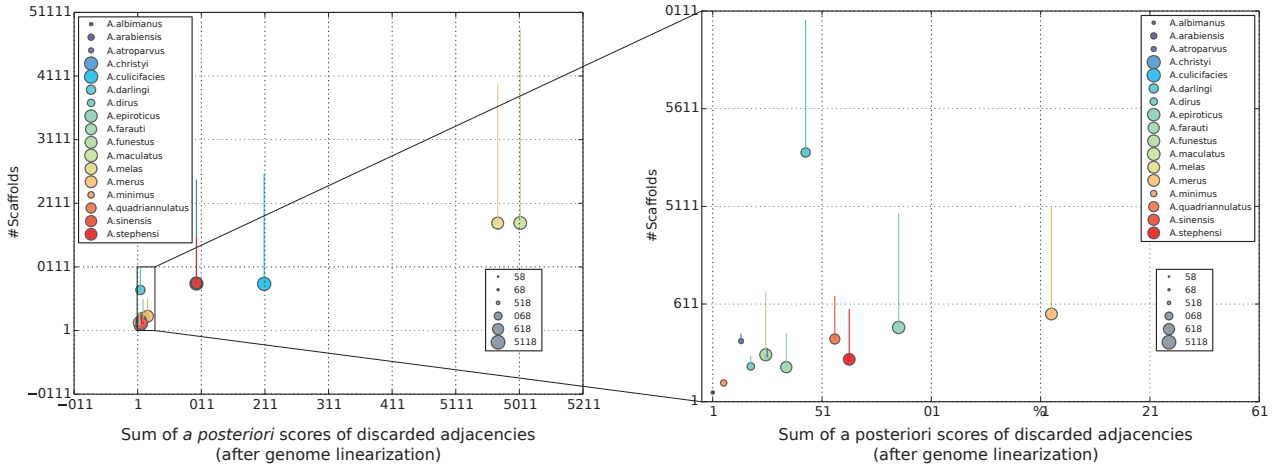


FIGURE 7.5 – Figure similaire à la figure 7.4 avec la topologie WG.

7.2.3 Comparaison des prédictions d'adjacences par ADSEQ+DECLONE à des données de *scaffolding* indépendantes pour l'espèce *An. funestus*

Au cours de l'analyse du jeu de données *Anopheles*, nous avons eu accès à deux nouvelles sources de données de *scaffolding* pour l'espèce *An. funestus* indépendantes des *scaffolding* de référence :

— une carte physique générée à partir de localisation de *scaffolds* de l'assemblage de référence d'*An. funestus* sur les bras chromosomiques (2R,

2L, 3R, 3L et X) par *FISH*[3] obtenue par l'équipe d'Igor Sharakhov (*cf.* section 2.2.3, p. 37, pour une présentation du protocole d'obtention d'une carte chromosomique par *FISH*) ;

— des *scaffolds* PacBio du génome d'*An. funestus* obtenus par l'équipe d'Adam Philippy.

Dans cette section, nous comparons les 331 prédictions d'adjacences obtenues pour *An. funestus* avec ADSEQ+DECLONE et la phylogénie X des 18 espèces d'*Anopheles* aux ordres de gènes déduits de l'analyse de la carte génétique et des *scaffolds* PacBio d'*An. funestus*.

Comparaison des prédictions ADSEQ+DECLONE à une carte physique d'*An. funestus*

Nous avons eu accès à une carte physique pour l'espèce *An. funestus* produite par l'équipe d'Igor Sharakhov (membre du consortium du séquençage des génomes d'*Anopheles*) du département d'entomologie de l'université VirginiaTech de Blacksburg aux États-Unis. Cette carte va prochainement faire l'objet d'une publication dans le cadre d'un article proposant une amélioration de l'assemblage de l'ensemble des 18 espèces d'*Anopheles*[4] auquel nous participons. Dans le cadre de cette collaboration, nous avons permis de corriger des incohérences synténiques présentes dans la carte physique du génome d'*An. funestus* (*cf.* section suivante).

Nous présentons d'abord la carte physique d'*An. funestus* et les corrections des incohérences synténiques apportées par les prédictions de ADSEQ+DECLONE. Puis, nous montrons la complétion de cette carte avec les prédictions d'adjacences de ADSEQ+DECLONE.

Carte chromosomique d'*An. funestus* et corrections des incohérences synténiques avec les prédictions de ADSEQ+DECLONE Sur les 562 *scaffolds* de l'assemblage initial d'*An. funestus*, 163 étaient assignés et ordonnés sur les 5 bras chromosomiques (2L, 2R, 3R, 3L et X). Ces 163 *scaffolds* représentent une taille de 119.702.469 *pb* soit 53,15 % de la taille de l'assemblage de référence. Parmi ces *scaffolds*, cinq ne sont pas considérés par ADSEQ+DECLONE car ils ne contiennent pas de gènes présents dans les arbres de gènes considérés par ADSEQ+DECLONE.

3. *Fluorescence In Situ Hybridisation*

4. plus trois autres espèces : *An. coluzzii*, *An. stephensi* souche $INDIA$ et *An. sinensis* souche $CHINA$

La carte chromosomique initiale de *An. funestus* est composée de plusieurs incohérences synténiques correspondant à la présence de 6 *scaffolds* localisés à deux positions distinctes sur la carte génétique. Ces 6 *scaffolds* sont représentés par des **nœuds noirs** sur la carte de la figure 7.6 et identifiables par les quatre liens qu'ils ont avec d'autres *scaffolds*, au lieu de deux liens.

Sur les quatre liens de chaque *scaffold* au positionnement ambiguë, seuls un ou deux correspondaient à des adjacences prédites par ADSEQ+DECLONE. Cela nous a permis de proposer un emplacement pour chacun de ces *scaffolds* comme on peut le voir sur le schéma de la carte corrigée représentée sur la partie droite de la figure 7.6. Une discussion sur la localisation multiple de ces *scaffolds* avec l'équipe d'Igor Sharakhov a permis de comprendre les artefacts de cette carte et de valider les localisations corrigées de ces *scaffolds* par ADSEQ+DECLONE. Prenons l'exemple du *scaffold* $KB668367$ du chromosome X pour illustrer la validité des corrections indiquées par ADSEQ+DECLONE, reconnaissable sur la figure 7.6 par la présence d'une étoile noire. Pour ce *scaffold*, deux sondes ont été produites pour le localiser et l'orienter sur les chromosomes d'*An. funestus*. Cependant, la similarité de la sonde qui plaçait ce *scaffold* entre les *scaffolds* $KB668522$ et $KB668688$ sur le chromosome X avait une identité d'alignement de l'ordre de 83,78 % avec la séquence du *scaffold* $KB668367$, soit une identité bien plus faible que l'autre sonde. Cette localisation du *scaffold* $KB668367$ a donc été rejetée au profit de la localisation donnée par l'autre sonde entre les *scaffolds* $KB668852$ et $KB668936$ sur le chromosome X, ceci en accord avec le positionnement indiqué par les prédictions de ADSEQ+DECLONE. Pour l'ensemble des six *scaffolds* les résultats d'assignation des *scaffolds* à une unique localisation par ADSEQ+DECLONE sont toutes concordantes avec les réanalyses des alignements par l'équipe d'Igor Sharakhov.

Cette étape de correction a été faite à la main et pourra fera l'objet d'un travail plus complet afin d'automatiser la correction de cartes physiques avec les prédictions d'adjacences de ADSEQ+DECLONE.

Complétion de la carte génétique d'*An. funestus* avec les prédictions d'adjacences de ADSEQ+DECLONE La figure 7.7 présente l'ensemble des 331 adjacences prédites par la méthode ADSEQ+DECLONE pour *An. funestus* associées à la carte physique produite par Sharakhov *et al*. Sur cette figure, les nœuds **noirs** représentent les 158 *scaffolds* placés sur les 5 bras chromosomiques d'*An. funestus* (2R, 2L, 3R, 3L et X) et les arêtes **noires** représentent

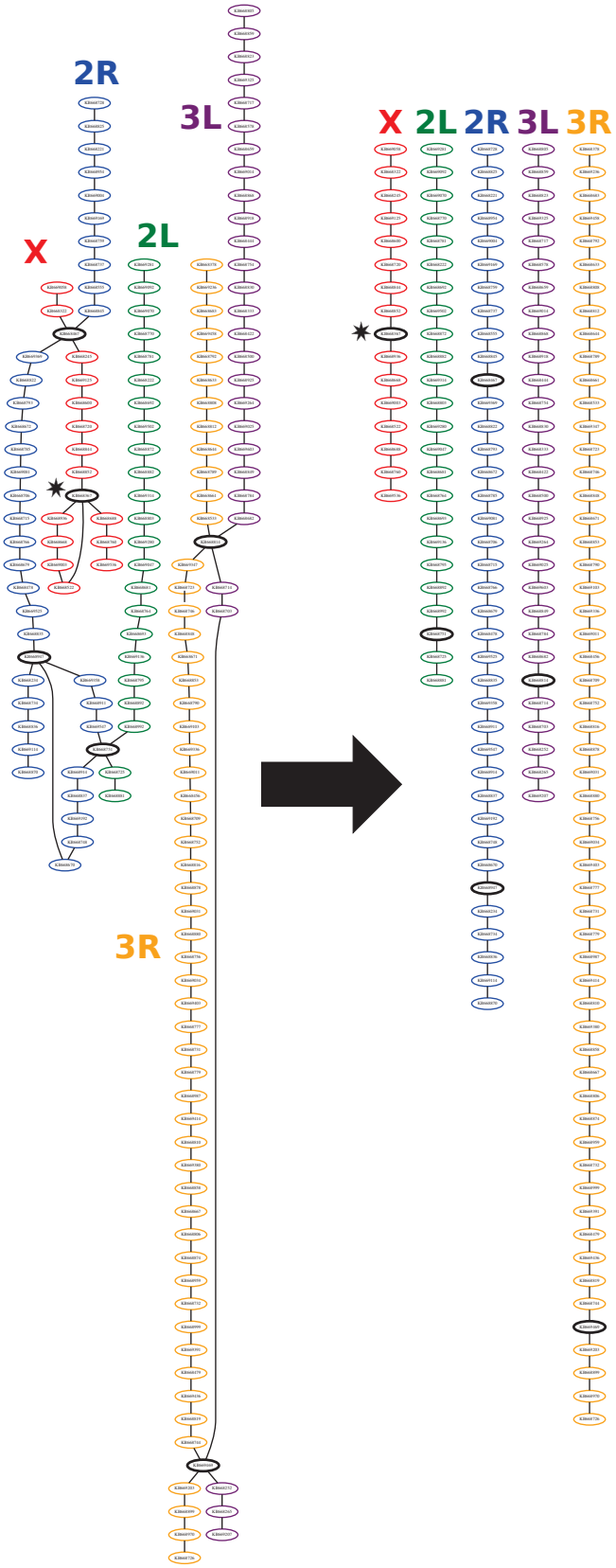


FIGURE 7.6 – Carte physique d'*An. funestus* avant et après correction de la localisation et de l'ordre des *scaffolds* par les prédictions de ADSEQ+DECLONE. Nœuds **rouges** : chromosome X, **bleus** : bras chromosomique 2R, **verts** : bras chromosomique 2L, **jaunes** : bras chromosomique 3R et **violets** : bras chromosomique 3L. Le *scaffold* annoté par une **étoile noire** est le *scaffold* utilisé dans le texte pour illustrer la validation, par les réanalyses de l'équipe d'Igor Sharakhov, de la correction du positionnement des *scaffolds* par ADSEQ+DECLONE.

les adjacences entre les *scaffolds* indiquées par la carte génétique [5]. Les nœuds **verts** représentent les *scaffolds* qui n'étaient pas présents sur la carte génétique et qui sont impliqués dans des adjacences inférées par ADSEQ+DECLONE. Les liens **verts** avec un dégradé vers le **rouge** représentent les adjacences prédites par ADSEQ+DECLONE entre les *scaffolds* où un lien **vert** indique un score support de DECLONE proche de 1 et un lien **rouge**, un support de DECLONE proche de 0. Des prédictions d'adjacences de *scaffolds* de ADSEQ+DECLONE peuvent se localiser entre deux *scaffolds* de la carte. D'un point de vue général, il semble y avoir peu de conflits entre la carte génétique et les prédictions de ADSEQ+DECLONE.

Pour une analyse plus aisée de la comparaison entre la carte et les prédictions ADSEQ+DECLONE, nous allons nous focaliser sur le chromosome X dans la figure 7.8. Pour les arêtes représentant les adjacences prédites par ADSEQ+DECLONE, il y a un nombre compris dans l'intervalle [0,1] représentant le support DECLONE et pour certaines des arêtes, il y a un nombre entre parenthèses correspondant au nombre de *reads* appariés soutenant l'adjacence de *scaffolding* inférée par BESST et validée par ADSEQ+DECLONE. Sur la partie haute du chromosome X de cette figure, *scaffolds* numérotés de 1 à 12, on observe qu'il n'y a aucun conflit entre la carte et les prédictions ADSEQ+DECLONE et la numérotation donne une conformation linéaire du chromosome X. Le *scaffold* 8 inféré adjacent avec le *scaffold* 7 n'est pas en conflit avec la carte et permet une conformation linéaire en s'insérant entre le 7 et le 9. On suppose que ADSEQ+DECLONE n'a pas trouvé l'adjacence entre le *scaffold* 8 et 9 ou qu'il y a d'autres *scaffolds* présents entre ces deux *scaffolds* qui n'ont pas été identifiés par ADSEQ+DECLONE (dû à un signal d'adjacence de gènes trop faible).

La lecture étant difficile avec les supports DECLONE, le graphique gauche de la figure 7.9 représente l'extrémité "basse" de la carte du chromosome X sans les supports. La partie droite de la figure 7.9 représente par une ligne rouge l'ordre des *scaffolds* minimisant le nombre de conflits entre la carte et les prédictions ADSEQ+DECLONE, où les identifiants des *scaffolds* sont remplacés par leurs positions dans cet ordre linéaire. La ligne pointillée rouge représente une adjacence entre les *scaffolds* 29 et 30 en accord avec la carte génétique mais pour laquelle nous ne savons pas si c'est une adjacence directe ou si d'autres *scaffolds* sont présents entre ces deux *scaffolds*. On observe que la position du *scaffold* 17 est conflictuelle. Pour la carte, celui-ci se situe entre

5. On a 158 *scaffolds* et pas 163 car on retire les 5 *scaffolds* non considérés par ADSEQ+DECLONE

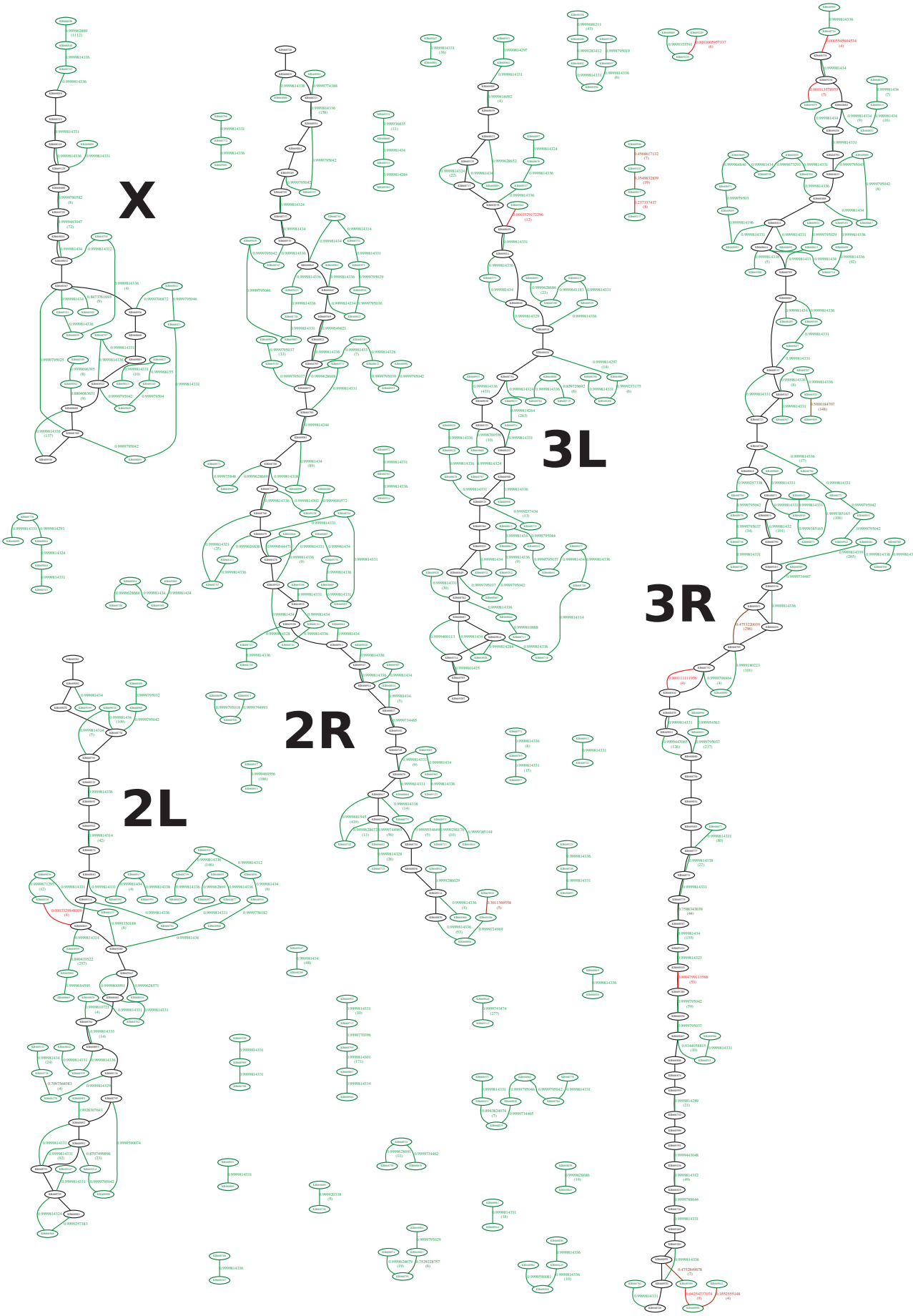


FIGURE 7.7 – Combinaison des prédictions d'adjacences de ADSEQ+DECLONE en **vert** et **rouge** avec la carte physique d'*An. funestus* d'Igor Sharakhov en **noir**.

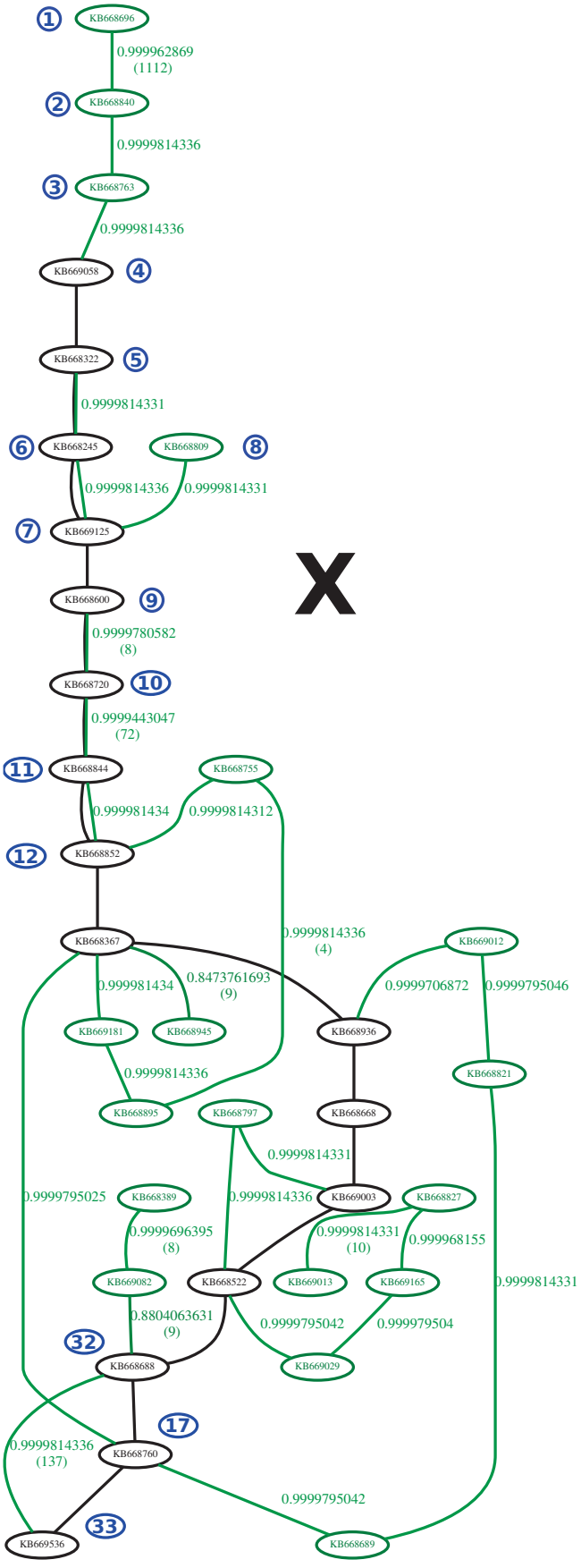


FIGURE 7.8 – Combinaison des prédictions d'adjacences de ADSEQ+DECLONE avec la carte physique d'*An. funestus* sur le chromosome X en **noir**.

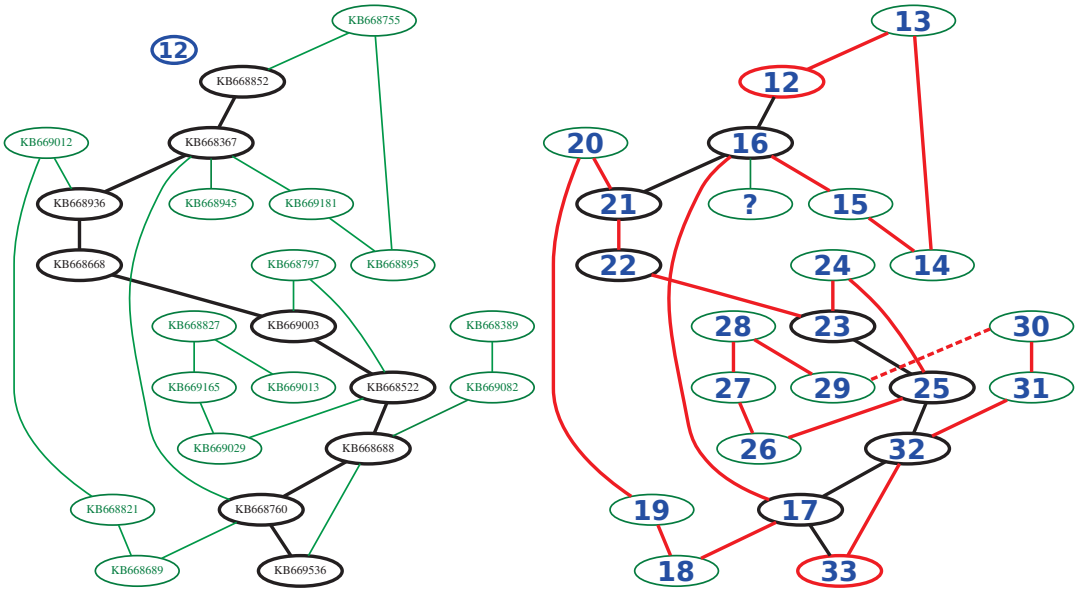


FIGURE 7.9 – Combinaison des prédictions d'adjacences de ADSEQ+DECLONE avec la carte physique d'*An. funestus* sur la partie "basse" du chromosome X.

les *scaffolds* 32 et 33, tandis que pour ADSEQ+DECLONE ce *scaffold* est localisé entre les *scaffolds* 16 et 21 de la carte et les *scaffolds* 32 et 33 sont directement adjacents. Les données de séquençage et la carte génétique d'*An. funestus* ayant été générés à partir de deux individus différents, le conflit pour le *scaffold* 17 peut donc correspondre à un polymorphisme de structure entre les deux individus. De plus, l'adjacence directe inférée par ADSEQ+DECLONE entre les *scaffolds* 32 et 33 correspond à une adjacence de *scaffolding* soutenue par 137 liens, ce qui correspond à un nombre important de liens pour les données de séquençage dont nous disposons (*cf.* figure 7.8).

L'analyse des prédictions de ADSEQ+DECLONE sur la carte génétique d'*An. funestus* a montré qu'il était possible d'utiliser notre méthode afin de résoudre les conflits synténiques présents dans les cartes génétiques. De plus, on a observé que nos prédictions sont majoritairement en accord avec les données de cartographie génétique, ce qui conforte l'idée d'utiliser la méthode ADSEQ+DECLONE comme une méthode de *scaffolding* des génomes actuels.

Comparaison des prédictions ADSEQ+DECLONE à des *scaffolds* PacBio d'*An. funestus*

Une deuxième source d'informations de *scaffolding* pour l'espèce *An. funestus* a été produite au cours de notre analyse du jeu de données des 18 *Anopheles*. Des données de séquençage PacBio ont été obtenues par le Adam

Philippy, membre du consortium *Anopheles*, avec une couverture de séquençage de 250X du génome *An. funestus*. L'assemblage et le *scaffolding* des *scaffolds* PacBio a été obtenu avec l'outil d'assemblage CANU avec les options par défaut résultant en 3.773 *scaffolds* correspondant à une taille d'assemblage de $263.192.532\ pb$ (soit $1{,}17\times$ la taille d'assemblage de référence du génome d'*An. funestus*). Adam Philippy attire l'attention sur le fait qu'aucun filtre pour éliminer des contaminants n'a été appliqué rendant possible la présence de séquences bactériennes contaminantes.

Prétraitement des *scaffolds* PacBio Pour comparer les adjacences prédites par ADSEQ+DECLONE avec les données PacBio, nous récupérons dans un premier temps les gènes localisés uniquement en extrémités des *scaffolds* initiaux considérés par ADSEQ+DECLONE. On ne considère que cet ensemble de gènes car ce sont les seuls qui sont impliqués dans les adjacences prédites par ADSEQ+DECLONE que l'on souhaite valider.

Dans un deuxième temps, nous alignons ces gènes sur les *scaffolds* PacBio avec l'outil d'alignement GMAP [Bertrand et al., 2009]. Les deux gènes d'un même *scaffold* initial doivent être adjacents si ils sont alignés sur un même *scaffold* PacBio car aucun des gènes localisés entre ces 2 gènes ne fait partie de l'ensemble de gènes alignés sur les *scaffolds* PacBio. GMAP peut aligne les gènes à de multiples localisations sur les *scaffolds* PacBio. Nous nous servons du fait que nous avons uniquement alignés les gènes situés en extrémités des *scaffolds* initiaux pour filtrer ces localisations afin de réduire au maximum les localisations incorrectes des gènes sur les *scaffolds*. Le code de ce protocole est disponible en annexe p. 194.

Une amélioration du protocole d'alignement, afin de n'avoir qu'un seul alignement pour chaque gène, consisterait à adapter le protocole d'alignement de contigs sur des *scaffolds* (*cf.* étape **4/** de la figure 5.6, p. 113) utilisé dans le protocole de validation de l'algorithme ADSEQ (*cf.* section 5.3.3, p. 112). Cependant, la localisation multiple de certains gènes semble provenir du fait que certaines régions de l'assemblage initial du génome d'*An. funestus* soit présentes en plusieurs copies sur les *scaffolds* PacBio [6].

Comparaison générale des adjacences Les adjacences de gènes prédites par ADSEQ+DECLONE sont comparées aux adjacences de gènes présentes

6. nous avons constaté que des *scaffolds* initiaux entiers étaient alignés sur plusieurs *scaffolds* PacBio différents.

sur les *scaffolds* PacBio après filtrage des alignements. Les résultats de la figure 7.10 montrent que sur les 331 nouvelles adjacences prédites par ADSEQ+DECLONE pour l'espèce *An. funestus* :

- seules 7 sont en désaccord avec les adjacences de gènes sur ces *scaffolds* PacBio ;
- pour 15 d'entre elles il n'est pas possible d'infirmer ou confirmer leur présence avec les données PacBio car au moins un des gènes de ces adjacences n'a pas pu être aligné avec fiabilité sur les *scaffolds* PacBio ;
- 135 ont été confirmées par les alignements de gènes sur les *scaffolds* PacBio ;
- 174 correspondent à des adjacences potentielles car les deux gènes composant ces adjacences sont, comme dans les *scaffolds* initiaux, localisés en extrémités des *scaffolds* PacBio. Ces adjacences ne peuvent donc être infirmées ou confirmées par ces données.

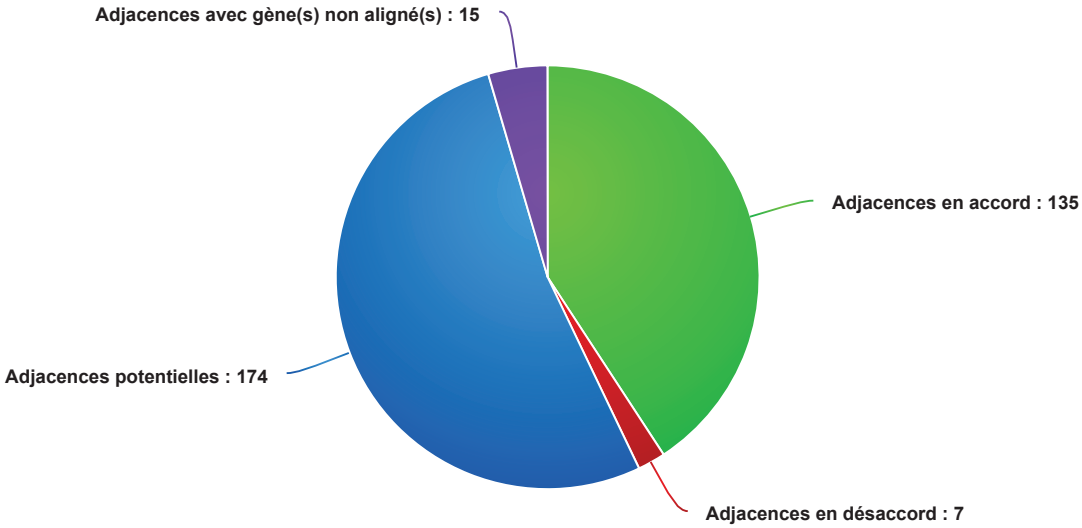
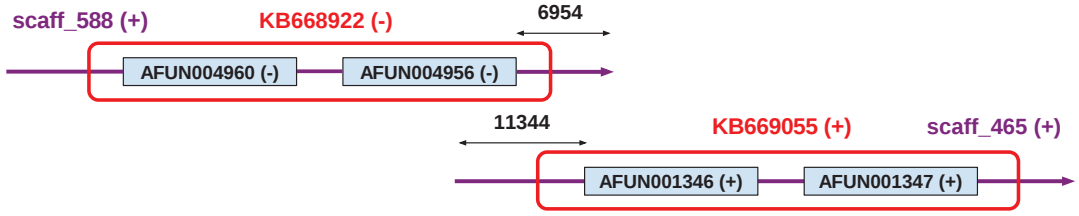
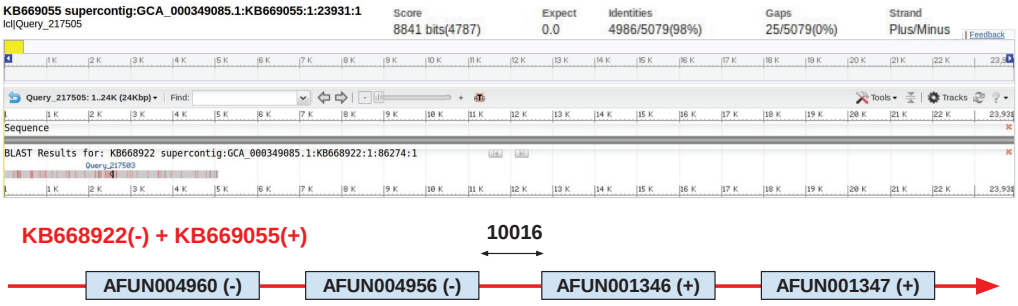


FIGURE 7.10 – Diagramme en secteur représentant le statut des adjacences d'*An. funestus* prédites par ADSEQ+DECLONE par rapport aux adjacences de gènes présentent sur les *scaffolds* PacBio.

Validation d'une adjacence potentielle par alignement des *scaffolds* avec BLASTN Parmi les adjacences potentielles, nous nous sommes intéressés à l'adjacence de *scaffolding* entre les gènes $AFUN004956$ et $AFUN001346$. Cette adjacence indiquait une distance négative (-4.753 bp) entre les *scaffolds* qu'elle liait, supposant que les séquences de ces deux *scaffolds* se chevauchaient. Nous avons donc récupéré les séquences des *scaffolds* initiaux $KB668922$ et $KB669055$ et les *scaffolds* PacBio $scaff\_588$ et $scaff\_465$, puis nous avons effectué un alignement deux à deux des *scaffolds* avec BLASTN



Alignement BLASTn des contigs: KB668922 et KB669055



Alignement BLASTn des *scaffolds* PacBio: scaff_588 et scaff_465

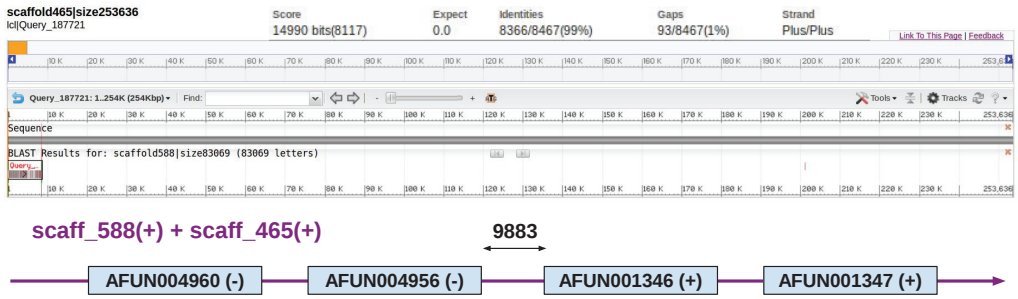


FIGURE 7.11 – Schéma de l'adjacence entre les gènes AFUN004956 et AFUN001346 chez l'espèce *An. funestus* prédite par ADSEQ+DECLONE et non trouvée par les *scaffolds* PacBio. Cette adjacence est validée par alignement avec BLASTN des *scaffolds* de l'assemblage de référence (KB668922 et KB669055) et des *scaffolds* PacBio (scaff_588 et scaff_465) ayant à leurs extrémités les gènes AFUN004956 et AFUN001346.

(*cf.* figure 7.11). Pour le couple de *scaffolds* initiaux et le couple de *scaffolds* PacBio, nous trouvons un chevauchement dans la configuration prédite par l'adjacence de *scaffolding* et ADSEQ+DECLONE. La distance estimée entre les gènes $AFUN004956$ et $AFUN001346$ par BLASTN est de 10.016 pb avec les *scaffolds* initiaux et 9.883 pb avec les *scaffolds* PacBio. L'inférence de l'adjacence de *scaffolding* par BESST estime une distance de 10.335 pb entre les deux gènes, ce qui indique une très bonne estimation de la distance entre ces deux gènes. La très grande similarité dans les distances estimées entre ces deux gènes par les deux alignements BLASTN et BESST indique que cette adjacence est bien présente dans le génome d'*AN. funestus*. Il est étonnant de voir que les méthodes d'assemblage (utilisées pour l'assemblage des *scaffolds* initiaux et PacBio) n'aient pas permis de détecter l'adjacence entre les gènes $AFUN004956$ et $AFUN001346$ alors que les alignements BLASTN indiquent un chevauchement de l'ordre de 5.000 pb avec 98 $\%$ d'identité entre les *scaffolds* initiaux et de l'ordre de 8.000 pb avec 99 $\%$ d'identité entre les *scaffolds* PacBio, ce qui correspond à de grands chevauchements de séquences et auraient dû permettre leur détection. Des méthodes de *scaffolding* comme BESST, dont les prédictions d'adjacences de *scaffolding* sont validés dans un contexte évolutif par ADSEQ+DECLONE, sont nécessaires pour pouvoir détecter ce type d'adjacences manquées par les méthodes d'assemblage *de novo* "traditionnelles".

On vient de voir qu'une part importante des adjacences prédites par ADSEQ+DECLONE lient potentiellement des *scaffolds* PacBio (environ la moitié). En effet, ADSEQ+DECLONE est capable de détecter des adjacences que des données PacBio ne permettent pas de déceler comme l'a montré l'exemple de l'adjacence entre les gènes $AFUN004956$ et $AFUN001346$. Il serait intéressant de poursuivre l'exploration et la validation éventuelle de ces adjacences.

Chapitre 8

Étude de l'évolution structurale de 18 génomes d'*Anopheles*

Dans ce chapitre, nous discutons de la phylogénie des 18 *Anopheles* et des deux topologies déterminées par Fontaine et al. [2015] le long du génome des espèces du complexe *Gambiae* à la lumière des résultats obtenus avec ADSEQ+DECLONE.

Dans une première section, nous présentons les résultats obtenus par Fontaine et al. [2015] sur la phylogénie des 18 espèces d'*Anopheles* et plus particulièrement sur le complexe *Gambiae* et la découverte de deux topologies divergentes de la phylogénie des espèces de ce groupe. Dans une deuxième section, nous discutons de l'histoire évolutive de l'ordre des gènes reconstruite par ADSEQ+DECLONE pour les deux topologies. À la lumière de ces résultats, nous discutons laquelle des deux topologies est la plus vraisemblable du point de vue de l'évolution de l'ordre des gènes.

Il est à noter que les résultats de ADSEQ+DECLONE utilisés dans ce chapitre correspondent aux troisième et quatrième exécutions de DECOSTAR décrits dans la section 7.1.2 (p. 139).

8.1 Deux topologies divergentes (X et WG)

8.1.1 Reconstruction de la phylogénie globale des 18 espèces d'*Anopheles* et intérêt du complexe *Gambiae*

La phylogénie des 18 *Anopheles* étudiée dans [Neafsey et al., 2015] (*cf.* figure 5.7, p. 114), dont le temps de divergence est estimé à 79 millions d'années [Moreno et al., 2010], a été obtenue à partir de l'alignement des séquences protéiques de 1.085 gènes orthologues unicopies avec l'algorithme

RAxML par maximum de vraisemblance en utilisant le modèle de substitution nucléotidique PROTGAMMAJTT. Cependant, cette analyse phylogénétique n'a pas permis de clairement définir la topologie du complexe *Gambiae* qui a fait l'objet d'un second article dans *Science* [Fontaine et al., 2015]. Ce groupe est particulièrement intéressant par la présence de l'espèce vectrice majeure du paludisme, *An. gambiae*, qui est à ce jour l'*Anopheles* dont le génome est le plus complet (avec assignation des séquences génomiques aux cinq bras chromosomiques). Il est également intéressant du fait de son temps de divergence récent entre les espèces du complexe (estimé à environ 2 millions d'années) et par la présence au sein du groupe à la fois d'espèces vectrices majeures (*An. gambiae* et *An. arabiensis*), mineures (*An. merus* et *An. melas*) et non vectrice (*An. quadriannulatus*) du paludisme. Il est donc intéressant de définir au mieux la phylogénie de ces espèces pour déterminer l'évolution de leur capacité vectorielle du paludisme. De plus, des études antérieures ont montré des événements d'introgression entre les membres du complexe *Gambiae* [Garcia et al., 1996; Besansky et al., 2003] qui ont pu être analysés à l'échelle des génomes entiers [Fontaine et al., 2015] (*cf.* définition 1 sur l'introgression, p. 18).

L'étude de la phylogénie par section chromosomique le long des génomes d'*Anopheles* a permis de définir la présence de deux topologies pour l'arbre des espèces du complexe *Gambiae* : la phylogénie reconstruite à partir de la séquence du chromosome X, notée X, supposée comme étant la vraie topologie de l'arbre des espèces et la phylogénie inférée à partir de l'ensemble des génomes, notée WG (*cf.* figure 8.1).

8.1.2 Détection des deux topologies au sein du complexe *Gambiae*

Pour détecter les deux topologies majoritaires le long du génome des espèces du complexe *Gambiae*, Fontaine et al. [2015] ont effectué un alignement multiple des génomes entiers de six espèces du complexe *Gambiae* (*An. gambiae*, *An. coluzzii*, *An. arabiensis*, *An. merus*, *An. melas* et *An. quadriannulatus*) et de deux espèces externes au groupe (*An. christyi* et *An. epiroticus*). Ils ont pu transmettre l'assignation des séquences génomiques sur les chromosomes disponibles dans le génome d'*An. gambiae* aux autres espèces du complexe. La stratégie utilisée pour l'alignement multiple des génomes entiers du complexe est similaire à celle employée pour l'alignement des 12 génomes de

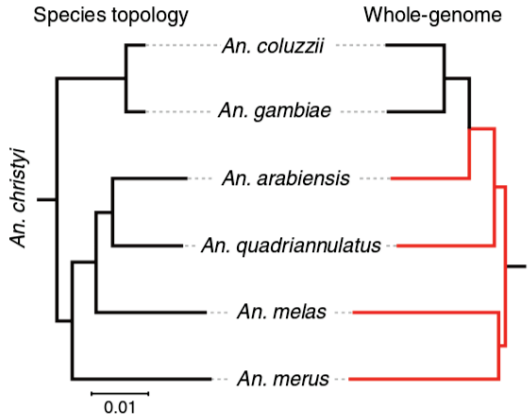
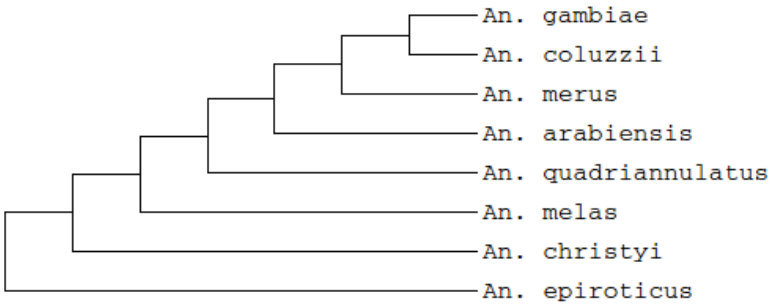


FIGURE 8.1 – Topologies *X* (obtenue à partir de la séquence du *chromosome X* et établie par Fontaine et al. [2015] comme la topologie des espèces) et *WG* (*Whole Genome*, topologie obtenue à partir de la séquence du génome entier) de l'arbre des espèces du complexe *Gambiae*. *From [Fontaine et al., 2015]. Reprinted with permission from AAAS.*

Drosophila [Stark et al., 2007] et des 29 génomes de mammifères [Lindblad-Toh et al., 2011]. L'alignement est effectué avec la fonctionnalité MULTIZ du jeu d'outils TBA (*Threaded-Blockset Aligner*) [Blanchette et al., 2004b]. L'approche d'alignement progressif de MULTIZ requiert une topologie des espèces donnant les liens de parentés entre espèces, l'alignement s'effectuant d'abord entre les espèces plus proches puis avec les autres espèces lors du parcours de la phylogénie jusqu'aux espèces les plus distantes. La topologie des espèces n'étant pas connue à ce moment, une topologie arbitraire a été choisie par les auteurs en entrée de MULTIZ :



MULTIZ effectue l'ensemble des alignements deux à deux entre les 8 génomes avec l'outil d'alignement LASTZ suivi d'un filtre "simple couverture" assurant que chaque région des deux espèces alignées n'est présente une seule fois. Diverses étapes de filtre sont effectuées, guidées par la topologie utilisée pour MULTIZ pour progressivement combiner les alignements deux à deux avec l'alignement multiple des génomes entiers jusqu'au parcours complet de la phylogénie.

À partir de ces alignements, des fenêtres génomiques de $50\ kbp$ non chevauchantes sont découpées, résultant en 4.063 sections génomiques. Chaque fenêtre génomique est assignée à un bras chromosomique (2R, 2L, 3R, 3L et X) grâce à la transmission par l'alignement des coordonnées du génome d'*An. gambiae*. Pour chacun des 4.063 *scaffolds* génomiques, les auteurs ont inféré une phylogénie par maximum de vraisemblance avec l'outil d'alignement RAxML. Les résultats ont permis de mettre en évidence 85 topologies le long du génome des espèces du complexe *Gambiae* dont les plus fréquentes ont été classées en deux grands groupes :

— les topologies propres au chromosome X comprenant le clade : (*An. melas*,(*An. arabiensis*,*An. quadriannulatus*)) ;

— les topologies majoritaires sur l'ensemble des autosomes (chromosomes non sexuels) contenant le clade : (*An. arabiensis*,(*An. gambiae*,*An. coluzzii*)).

Ces deux grands groupes correspondent aux topologies X et WG établit par Fontaine *et al.* pour la phylogénie des espèces (*cf.* figure 8.1). La topologie WG (*An. arabiensis*,(*An. gambiae*,*An. coluzzii*)) étant la topologie majeure rencontrée sur l'ensemble des génomes, on a tendance à supposer que cette phylogénie représente la topologie des espèces. Cependant, des analyses supplémentaires des auteurs vont conclure que la phylogénie des espèces correspond à celle retrouvée sur le chromosome X (topologie X). Pour arriver à ce résultat, les auteurs indiquent d'abord qu'une introgression autosomale entre l'espèce *An. arabiensis* et l'ancêtre de *An. gambiae+An. coluzzii* est connue et documentée depuis longtemps [Coluzzi et al., 1979; della Torre et al., 1997] et pourrait expliquer la très forte dissimilarité entre les autosomes et le chromosome X. Cependant, la topologie de l'arbre n'étant pas résolue, une interprétation définitive de ces signaux conflictuels est exclue. Afin de déterminer laquelle des deux topologies correspond à la phylogénie des espèces (c.-à-d. celle pour laquelle il n'y a pas eu d'événements d'introgression), les auteurs ont appliqué une stratégie basée sur l'analyse de la divergence des séquences. Si un événement d'introgression entre deux espèces a lieu alors la divergence de séquence entre ces deux espèces doit être réduite (puisque les espèces échangent du contenu génétique et homogénéisent ainsi leurs séquences). La partie du génome qui reflète la "vraie" topologie des espèces est alors celle pour laquelle le temps de divergence est le plus grand. Pour tester cette hypothèse, les auteurs ont reproduit des arbres de gènes en se limitant aux trois espèces *An. gambiae* (G), *An. arabiensis* (A) et *An. melas* (L), et en réduisant la fenêtre génomique, pour inférer les arbres de gènes, à

10 kpb. Les résultats de la partie C de la figure 8.2 montrent que les arbres avec la topologie **(G,(L,A))**, correspondant à la topologie X, ont un temps de divergence significativement plus élevé que les arbres avec la topologie **(L,(G,A))**, correspondant à la topologie WG. Cela indique que la topologie X est la "vraie" topologie des espèces sous l'hypothèse d'introgression définit par Fontaine et al. [2015].

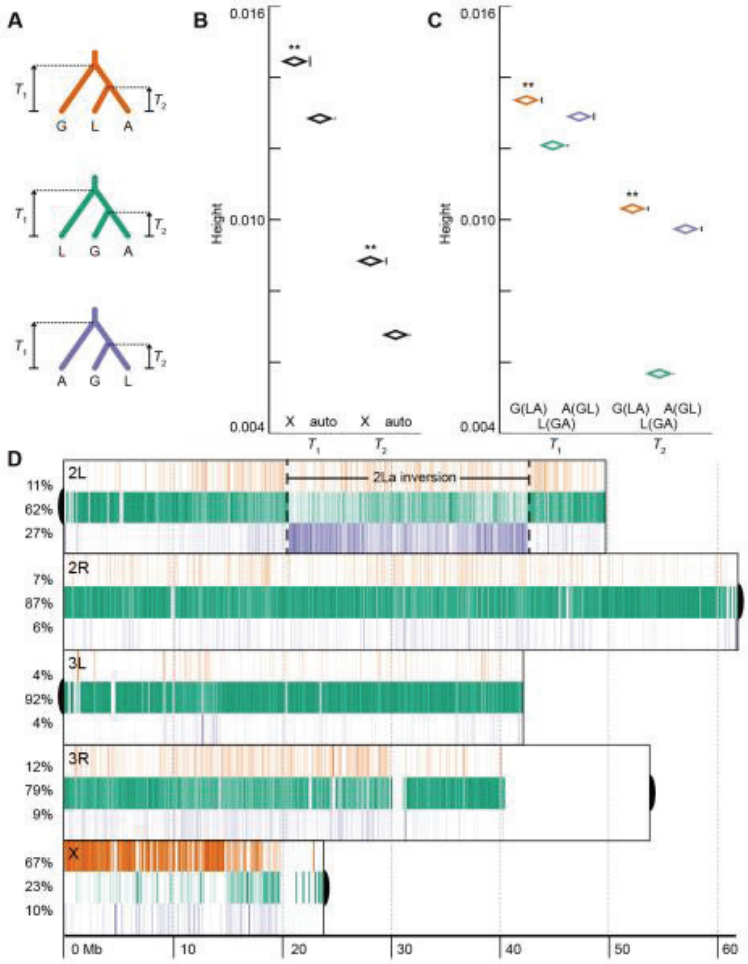


FIGURE 8.2 – Analyse de la topologie de l'arbre des *Anopheles* sous l'hypothèse d'introgression. **A/** Code couleur des trois topologies d'arbres possibles avec *An. arabiensis* (A), *An. gambiae* (G), et *An. melas* (L) : **(G,(L,A))**, **(L,(G,A))** et **(A,(G,L))**, et correspondance des temps de divergence (T_1 et T_2). **B/** Moyenne des temps de divergence T_1 et T_2 de l'ensemble d'arbres sur les autosomes et de l'ensemble des arbres sur le chromosome X. **C/** Moyenne des temps de divergence T_1 et T_2 pour les trois topologies d'arbres **(G,(L,A))**, **(L,(G,A))** et **(A,(G,L))**. **D/** Distribution des trois topologies d'arbres le long des chromosomes des génomes d'*Anopheles* montrant que la topologie **(G,(L,A))** est bien la topologie majoritaire sur le chromosome X et **(L,(G,A))** la topologie majoritaire des autosomes.

8.2 Reconstruction de l'histoire évolutive de l'ordre des gènes

La méthode ADSEQ+DECLONE n'est pas une méthode phylogénétique à proprement parler puisqu'elle requiert la phylogénie des espèces et ne contient pas d'extension pour la recherche d'une phylogénie optimale basée sur des critères évolutifs. Cependant, comme la méthode infère l'ordre des gènes ancestraux et les événements évolutifs, et de plus est efficace en temps de calcul [1], la méthode peut être utilisée afin d'évaluer quelques phylogénies sélectionnées (en exécutant ADSEQ+DECLONE sur ces phylogénies) et en comparant différentes statistiques évolutives afin de déterminer la phylogénie avec laquelle ADSEQ+DECLONE infère l'histoire évolutive de l'ordre des gènes la plus cohérente.

Dans le cadre de la phylogénie des 18 espèces d'*Anopheles*, nous allons comparer les résultats obtenus par ADSEQ+DECLONE sur les deux phylogénies X et WG et discuter du choix de la topologie X comme étant la phylogénie des espèces par Fontaine et al. [2015].

8.2.1 Assignation aux bras chromosomiques

Une première étape pour déterminer l'histoire évolutive de l'ordre des gènes le long des génomes des 18 *Anopheles* par ADSEQ+DECLONE consiste à assigner les *scaffolds* actuels et ancestraux prédits par ADSEQ+DECLONE aux bras chromosomiques auxquels ils appartiennent. La procédure que nous employons est différente de celle employée par Fontaine et al. [2015], car ADSEQ+DECLONE ne reconstruit pas la séquence des génomes actuels et ancestraux mais uniquement l'ordre des gènes. Notre procédure d'assignation des gènes actuels et ancestraux est donc basée sur une approche utilisant des données phylogénétiques et de synténies de gènes et non pas sur l'alignement de séquences génomiques. Comme *An. gambiae* est la seule espèce pour laquelle les séquences sont assignées aux chromosomes, nous assignons les gènes aux bras chromosomiques avec la méthode probabiliste suivante : pour chaque gène g (ancestral ou actuel), un groupe de gènes orthologues *An. gambiae* est défini comme l'ensemble des gènes *An. gambiae* appartenant

1. toutes les étapes d'inférence d'arbres de gènes, réconciliation de ceux-ci et *scaffolding* conjoint des génomes actuels et ancestraux sur ce jeu de données sont effectuées en $5\ h$ sur un ordinateur de bureau.

à la même famille de gènes que g dont le dernier ancêtre commun avec g correspond à un nœud de spéciation. Il est à noter que le groupe peut être vide et que cette définition inclut le cas où g est un ancêtre du gène *An. gambiae*. La probabilité de g d'être sur le chromosome X est alors définie comme la fréquence des orthologues placés sur le chromosome X d'*An. gambiae*, ou, si il n'y a pas d'orthologue présent sur le chromosome X, par la probabilité basale, définie comme la fréquence globale des gènes d'*An. gambiae* d'être sur le chromosome X. Ensuite, chaque *scaffold* a une probabilité d'être localisé sur le chromosome X correspondant à la moyenne des probabilités de l'ensemble des gènes qu'il contient. Pour finir, chaque gène hérite de la probabilité d'être sur le chromosome X du *scaffold* auquel il appartient.

Récemment, une assignation des gènes de l'espèce *An. albimanus* sur les bras chromosomiques 2L, 2R, 3L, 3R et X, a été publiée à l'occasion d'un nouvel assemblage de cette espèce [Artemov et al., 2017]. Nous avons utilisé ces données afin de vérifier que les assignations que nous avons produites avec notre protocole étaient correctes. Sur les 8.840 gènes assignés à des chromosomes par notre méthode sur le nouvel assemblage, nous avons correctement prédit l'assignation autosomes/X de 8.837 gènes. Cela valide la capacité de notre protocole à définir si un gène appartient au chromosome X ou aux autosomes.

8.2.2 Duplications de gènes

Notre pipeline d'inférence d'arbres de gènes (*cf.* figure 5.5) permet d'énumérer les duplications de gènes. Nous comptons un total de 6.461 duplications avec la phylogénie X et 6.159 avec la phylogénie WG (*cf.* table 8.1). Cela indique que pour un grand nombre de familles de gènes, des duplications de gènes sont présentes avec la topologie X et absentes de la topologie WG. Pour chacune de ces familles, une branche fortement soutenue (*bootstrap* de 100 % avec RAxML) est compatible avec la phylogénie WG mais pas la phylogénie X, indiquant que les branches les plus supportées sont plus souvent compatibles avec la topologie WG. Cette observation soutient le résultat obtenu par Fontaine et al. [2015] indiquant que la plupart des gènes suivent la phylogénie WG. Cette observation supporte également l'idée de l'évolution du génome en deux compartiments.

8.2.3 *Scaffolding* des génomes actuels et ancestraux

Sur la table 7.2 (p. 148), on observe que le *scaffolding* des génomes actuels est très légèrement meilleur, en terme de niveau de fragmentation des génomes, avec la phylogénie WG (moyenne de 749 *scaffolds* actuels) que la phylogénie X (moyenne de 751 *scaffolds* actuels). On observe la même tendance pour le *scaffolding* des génomes ancestraux avec une moyenne de 1.670 *scaffolds* ancestraux avec la topologie WG et de 1.771 *scaffolds* ancestraux avec la topologie X. Cependant, si l'on considère le nombre médian de *scaffolds* actuels de toutes les espèces, on voit que la topologie X offre un meilleur *scaffolding* avec un nombre médian de $296{,}5$ *scaffolds* que la topologie WG avec un nombre médian de *scaffolds* de $299{,}5$. Cela peut être attribué à la position basale du génome le mieux assemblé (*An. gambiae*) dans le complexe *Gambiae*. En effet, dans l'algorithme ADSEQ+DECLONE les espèces sœurs peuvent être assemblées avec *An. gambiae* mais les espèces externes ne peuvent pas l'être, le *scaffolding* est donc nécessairement plus efficace si un génome complètement assemblé a plus d'espèces sœurs, ce qui est le cas d'*An. gambiae* dans la phylogénie X. Il est intéressant de voir que l'on a une meilleure inférence de l'ordre des gènes chez les génomes ancestraux avec la phylogénie WG, malgré l'artefact présent chez les génomes actuels en faveur de la topologie X. Cette meilleure reconstruction de l'ordre des gènes ancestraux peut être considérée comme un premier signal contredisant l'hypothèse selon laquelle la topologie X correspondrait à la topologie des 18 espèces d'*Anopheles*, cependant ce signal ne permet pas de rejeter clairement cette hypothèse.

8.2.4 Conflits synténiques

Nous avons également comparé le niveau de conflits synténiques observés en sortie de ADSEQ+DECLONE entre les deux phylogénies. Celui-ci est défini comme la somme des scores des adjacences en sortie de ADSEQ+DECLONE qui ont été retirées du *scaffolding* lors de l'étape de linéarisation des prédictions d'adjacences. On observe un niveau de conflits synténiques plus élevé avec la topologie X (7.655) qu'avec la topologie WG (6.319). Les figures 7.4, p. 149, (resp. 7.5, p. 149) représentent sur l'axe des ordonnées la somme des scores des adjacences exclues lors de la linéarisation pour la topologie X (resp. WG). Cette observation peut être perçue comme un deuxième élément contredisant l'hypothèse basée sur la séquence génomique qui soutient la phylogénie X comme la phylogénie des 18 *Anopheles*.

Cependant la quantité de conflit, si elle est une mesure intuitive de la qualité d'une phylogénie, n'a pas été validée en tant que telle, et ne nous permet donc pas de nous prononcer de façon définitive.

8.2.5 Mouvements de gènes (translocations)

Il est documenté depuis longtemps qu'il n'y a pas de réarrangement chromosomique à grande échelle entre les autosomes et le chromosome X. Cela nous permet d'assigner la plupart des *scaffolds* actuels et ancestraux (avec plus d'un gène) sur le chromosome X ou les autosomes, avec une très grande confiance (*cf.* section 8.2.1). Pour déterminer la présence d'un mouvement de gènes, nous avons établi la définition suivante : pour chaque couple de gènes, où l'un est le descendant direct de l'autre, nous inférons une translocation (entre le chromosome X et un autosome) si la probabilité du gène ancestral d'appartenir au chromosome X est $\leq 0{,}2$ et la probabilité du descendant d'appartenir au chromosome X est $\geq 0{,}8$, et inversement.

L'analyse des arbres de gènes et de l'assignation des gènes a permis de trouver 429 gènes s'étant déplacés du chromosome X aux autosomes, et 469 des autosomes au chromosome X, ce qui confirme la tendance générale trouvé dans [Neafsey et al., 2015] (59 parmi les 132 mouvements de gènes provenaient du chromosome X). L'analyse a montré que notre approche détectait un plus grand nombre de mouvements de gènes que celle employé par Neafsey et al. [2015], ce qui permet une étude plus fine des réarrangements de gènes de ce type.

8.2.6 Réarrangements chromosomiques

Pour détecter des réarrangements le long de l'histoire évolutive reconstruite par ADSEQ+DECLONE pour les deux topologies, nous utilisons le protocole suivant :
Pour chaque branche de l'arbre des espèces, les gènes avec exactement un exemplaire dans leur famille à la fois dans l'espèce ancestrale et l'espèce descendante ont été sélectionnés. Les adjacences conservées ont été calculées, celles-ci correspondent aux adjacences présentes entre gènes homologues de l'espèce ancestrale et descendante. Pour retirer les mouvements de gènes du compte des réarrangements, nous filtrons les gènes qui ne sont pas impliqués dans une adjacence. Un réarrangement (création ou cassure d'adjacence) est compté à chaque fois que deux extrémités de gènes contiguës sur un *scaffold*

de l'espèce ancestrale ne le sont plus dans l'espèce descendante, et inversement. Quand elles ne sont pas contiguës, nous vérifions que les deux extrémités ne soient pas à l'extrémité de leurs *scaffolds*, afin d'éviter de compter comme un réarrangement une potentielle adjacence non détectée. Nous assumons que cette approche pour détecter des réarrangements chromosomiques est conservative et sous-estime le nombre de réarrangements, car elle ne détecte pas les réarrangements cachés par l'assemblage incomplet des génomes considérés. De plus, cette sous-estimation peut être biaisée par le degré de fragmentation des espèces comparées, si bien que deux nombres de réarrangements ne peuvent être comparées même pour des espèces très proches. Cependant, pour un même jeu de génomes donné en entrée, le nombre de réarrangements est comparable pour deux phylogénies données, comme le sont le nombre de réarrangements dans les chromosomes sexuels et les autosomes.

En utilisant cette définition pour détecter les réarrangements chromosomiques, nous avons énuméré 3.515 créations et cassures d'adjacences en utilisant la phylogénie X et 3.337 en utilisant la phylogénie WG. La différence est illustrée dans la figure 8.3. Entre les deux phylogénies, on observe une baisse de 30 % du nombre de réarrangements au sein du complexe *Gambiae* avec la phylogénie WG comparée à la phylogénie X. Ces gains/cassures d'adjacences peuvent être combinés le long de chaque branche pour détecter les inversions. Une inversion est définie par une paire de cassures dans l'espèce ancestrale et une paire de gains d'adjacences dans l'espèce descendante impliquant les quatre mêmes extrémités de gènes. Ceci permet l'identification de 242 inversions dans la phylogénie X (incluant 19 inversions dans le complexe *Gambiae*, dont 4 sur la branche de l'espèce ancestrale directe d'*An. gambiae* à celle-ci) et 240 inversions avec la phylogénie WG (avec 4 inversions dans le complexe *Gambiae* dont 1 sur la lignée d'*An. gambiae*). Ces résultats vont en faveur d'une phylogénie des *Anopheles* avec la topologie du WG sur un critère de parcimonie pour les réarrangements.

Événements	Phylogénie X		Phylogénie WG	
	X	Autosomes	X	Autosomes
Duplications	604	5857	606	5553
Réarrangements	415	2949	416	2760

TABLE 8.1 – Nombres de duplications et réarrangements inférés sur le chromosome X et sur les autosomes pour les phylogénies X et WG par ADSEQ+DECLONE.

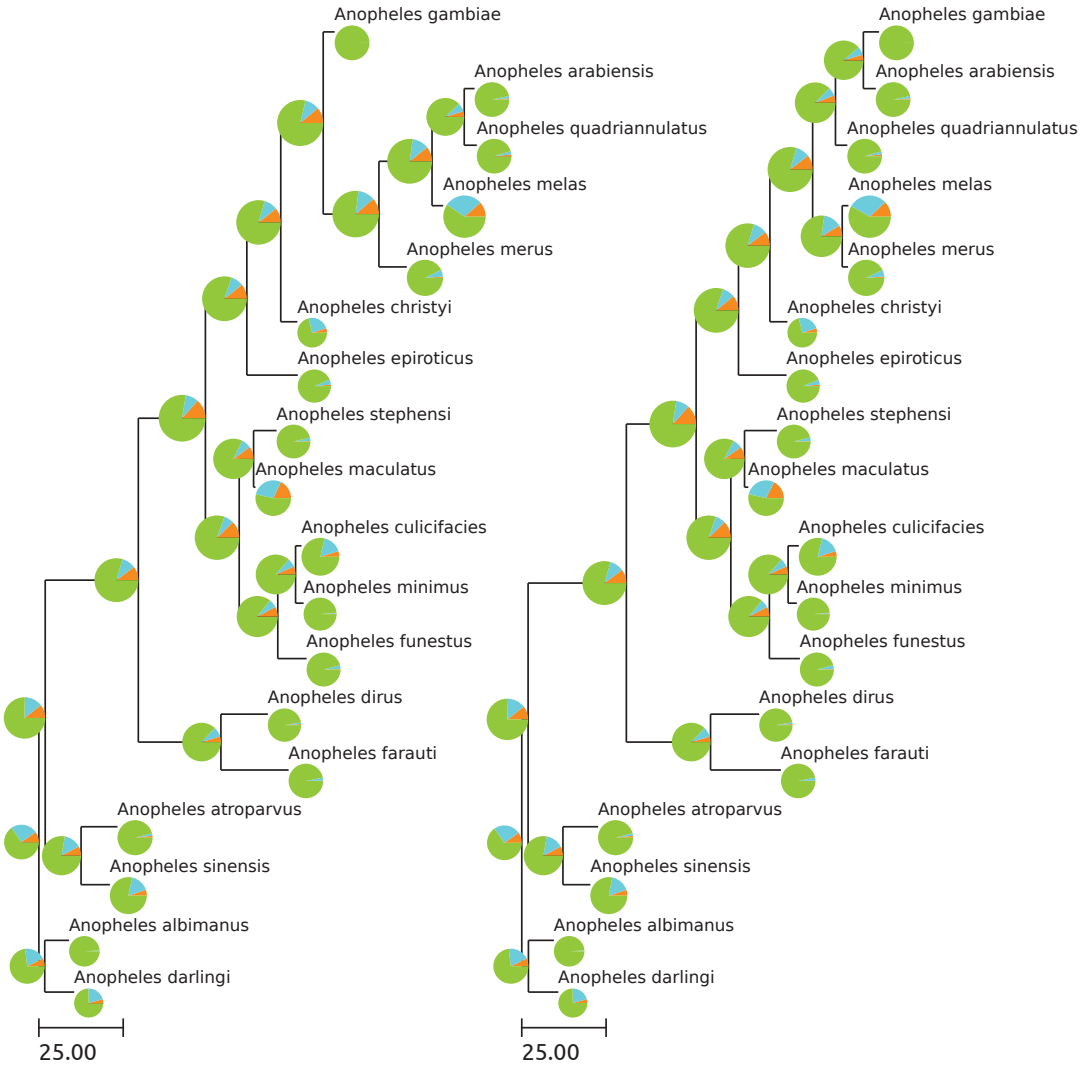


FIGURE 8.3 – Phylogénie des 18 *Anopheles* (gauche : phylogénie X, droite : phylogénie WG) avec le nombre de réarrangements par adjacence comme longueur de branche ($\times 10^{-3}$). Le diagramme en secteurs pour chaque espèces représente le degré d'adjacence des gènes de l'espèce. **Orange** : gènes sans adjacence, **bleu cyan** : gènes avec degré 1 d'adjacence et **vert** : gènes avec degré 2 d'adjacence. De plus, le diamètre de chaque diagramme représente le nombre de gènes dans l'espèce correspondante.

8.2.7 Comparaison de l'évolution du chromosome X et des autosomes

Les résultats obtenus sur la phylogénie des 18 *Anopheles* montrent que le chromosome X et les autosomes ont différents modes d'évolution au regard des duplications de gènes et des réarrangements[2] (*cf.* table 8.1). Cette compartimentation a été décrite par Fontaine et al. [2015] pour les gènes et attribuée à l'introgression des autosomes de l'espèce *An. arabiensis* chez l'ancêtre *An. coluzzii+An. gambiae*. Neafsey et al. [2015] ont également décrit que le taux de réarrangements était plus élevé sur le chromosome X que sur les autosomes. Nos résultats montrent une tendance similaire. Nous avons calculé les taux de réarrangements en normalisant le nombre de gains+cassures d'adjacences observés par le nombre d'adjacences de gènes sur l'ensemble des génomes actuels et ancestraux. Avec la phylogénie X, on observe que le taux de réarrangement sur le chromosome X est égal à $1{,}46\times$ le taux observé sur les autosomes, une observation similaire est obtenue avec la phylogénie WG ($1{,}57\times$). De plus, nous pouvons voir que moins d'événements sont trouvés sur le chromosome X avec la phylogénie X et moins d'événements sont trouvés sur les autosomes avec la phylogénie WG. Il semble donc que ce ne sont pas que les gènes qui suivent une histoire évolutive différente causée par l'introgression [Fontaine et al., 2015] mais également les chromosomes entiers. Cependant, la compartimentation observée seule ne permet pas de spécifier quelle partie du génome représente au mieux la diversification des espèces. Comme l'apparition du complexe *Gambiae* est estimé à environ 2 millions d'années, il est raisonnable d'utiliser la parcimonie concernant les réarrangements (*cf.* section suivante). Si nous le faisons, nous trouvons moins de réarrangements avec la phylogénie WG, même normalisé par le nombre d'adjacences (car une augmentation du nombre de réarrangements peut être l'effet d'un plus grand nombre d'adjacences), avec $9{,}15 \times 10^{-3}$ réarrangements/adjacence pour la phylogénie WG contre $9{,}68 \times 10^{-3}$ réarrangements/adjacence pour la phylogénie WG. Cela indique que le signal des réarrangements établit la phylogénie WG comme étant la vraie topologie des 18 *Anopheles* et non la phylogénie X établie par Fontaine et al. [2015], rendant la résolution de la phylogénie des 18 *Anopheles* et plus particulièrement du complexe *Gambiae* de nouveau floue [Clark and Messer, 2015].

2. nous ne comptons pas les événements de pertes de gènes pour comparer les phylogénies car l'absence de gènes peut être due à une annotation incomplète des génomes et pas nécessairement à la perte de gènes au cours de l'évolution.

8.2.8 Évaluation de la pertinence de la parcimonie pour l'étude des réarrangements

L'utilisation de la parcimonie comme bon critère pour évaluer la qualité d'une phylogénie peut être remise en question. En effet, les réarrangements chromosomiques chez les *Anopheles* ne sont pas distribués uniformément le long du génome [Pombi et al., 2008], ils peuvent montrer un certain degré de convergence et du polymorphisme inter-espèces. Pour évaluer si dans le complexe *Gambiae* nous sommes dans un espace de validité de la parcimonie, nous avons comparé l'ordre des gènes d'*An. gambiae* avec *An. albimanus* qui depuis la récente amélioration de l'assemblage du dernier sont les deux génomes les plus complets du genre *Anopheles* [Artemov et al., 2017]. Nous avons sélectionné tous les gènes assignés à un chromosome et appliqué l'estimateur de distance ER2 [Biller et al., 2016]. L'estimateur est basé sur un modèle non uniforme de réarrangements génomiques qui a prouvé qu'il donnait les résultats les plus fiables sur les génomes de mammifères dont le temps de diversification est équivalent à celui des 18 *Anopheles*. Nous avons trouvé une estimation de 1.313 inversions avec l'estimateur statistique tandis que la solution parcimonieuse en trouve 1.300. Les résultats obtenus par la parcimonie sont donc dans un intervalle de $1\ \%$ des résultats de la méthode statistique, donc très éloigné d'une saturation du signal. Comme *An. albimanus* et *An. gambiae* ont un temps de divergence de 79 millions d'années tandis que les espèces les plus distantes, au sein du complexe *Gambiae*, ont un temps de divergence de l'ordre de 2-3 millions d'années, on peut donc raisonnablement penser que les réarrangements ne sont pas suffisamment nombreux pour saturer le signal de parcimonie.

En conclusion, nous avons évalué les deux topologies X et WG établies par Fontaine et ses collaborateurs sur la phylogénie des 18 *Anopheles*, nous avons également discuté de la désignation de la topologie du X comme étant la "vraie" phylogénie des 18 espèces d'*Anopheles* à partir d'un examen de la séquence génomique des *Anopheles* le long des bras chromosomiques par Fontaine et al. [2015]. Les résultats obtenus avec ADSEQ+DECLONE sur la reconstruction de l'histoire évolutive de l'ordre des gènes le long de la phylogénie des 18 *Anopheles* ont permis d'évaluer sur divers critères évolutifs (duplications, mouvements de gènes et réarrangements) et synténiques (*scaffolding* des génomes actuels et ancestraux, analyse du conflits synténiques) laquelle des deux topologies (X ou WG) était la plus à même de représenter

la phylogénie des espèces. Sans vouloir conclure de façon définitive, tous nos indicateurs semblent préférer la phylogénie WG, ce qui est en désaccord avec les conclusions de Fontaine et al. [2015]. Ces résultats sont majoritairement soutenus par le signal de parcimonie des réarrangements, dont la validité à l'échelle de temps des *Anopheles* a été vérifiée dans la section 8.2.8, pour lequel la phylogénie WG contient moins de réarrangements chromosomiques. Nos résultats montrent, en accord avec Fontaine et al. [2015], que les génomes du complexe *Gambiae* sont compartimentés en deux secteurs suivants leurs différentes histoires évolutives.

Conclusion & Perspectives

Au cours de ce projet de thèse, nous avons développé un nouveau cadre théorique et pratique permettant d'aborder conjointement deux problèmes de la phylogénie moléculaire à l'échelle des génomes :

— l'inférence de l'histoire évolutive de l'ordre des marqueurs génomiques (et par conséquence la reconstruction de l'ordre de marqueurs génomiques chez les génomes ancestraux) ;
— l'amélioration de l'ordre de marqueurs génomiques d'espèces actuelles.

Auparavant, ces deux problématiques ont été traitées par des approches algorithmiques différentes et deux communautés scientifiques distinctes. Le lien étroit entre ces deux problématiques n'a été établi que récemment à l'instar de l'article [Lin et al., 2014] qui a montré la similarité entre le graphe *de Bruijn* et le *breakpoint graph*, respectivement utilisés pour l'assemblage des génomes actuels et la reconstruction de l'ordre de marqueurs génomiques d'espèces ancestrales. Ouvrant la possibilité que l'assemblage de génomes actuels et la reconstruction de la structure de génomes ancestraux puissent être traités dans une seule et même structure et approche algorithmique. Parallèlement à nos travaux, d'autres méthodes visant à résoudre conjointement ces deux problématiques ont été publiées, comme GOS-ASM [Aganezov and Alekseyev, 2016] et DESCHRAMBLER [Kim et al., 2017] montrant que ce domaine est en plein développement.

L'application de notre méthode sur le jeu de données des 18 *Anopheles* a fourni une preuve de concept montrant que l'approche simultanée de ces deux problèmes peut être utilisée sur des jeux données biologiques de grandes tailles et produire de nouvelles connaissances biologiques. L'analyse des résultats de *scaffolding* a montré que DeCoSTAR pouvait être utilisé sur des jeux de données composés de génomes à divers degrés d'assemblage. Les comparaisons des prédictions d'adjacences avec la carte chromosomique et les données PacBio ont montré que DeCoSTAR prédit peu d'adjacences en conflits avec ces données. Pour la carte chromosomique, les prédictions de DeCoSTAR ont permis de corriger les incohérences synténiques de la carte

et d'augmenter fortement la linéarité des *scaffolds* sur les chromosomes. La comparaison avec les *scaffolds* PacBio a révélé qu'une part importante des adjacences prédites, dont certaines validées par alignement avec BLASTN, n'étaient pas présentes sur les *scaffolds* PacBio, montrant qu'à l'heure actuelle des méthodes de *scaffolding* par génomique comparative sont encore nécessaires. Nos prédictions d'adjacences actuelles sont intégrées dans une nouvelle version de l'assemblage des 18 génomes d'*Anopheles*[3]. Cet assemblage, qui fera l'objet d'un article dans le cadre du consortium *Anopheles*, est obtenu à partir d'un consensus de trois méthodes de *scaffolding* par génomique comparative, incluant DECOSTAR, GOS-ASM et une nouvelle méthode, nommée ORTHOSTITCH (non publiée).

D'un point de vue logiciel, le pipeline de génération des données d'entrée de DECOSTAR nécessite des développements supplémentaires afin d'obtenir un utilitaire exécutable par la communauté scientifique pour générer les données d'entrée de DECOSTAR à partir de données génomiques et phylogénétiques aux formats standards.

L'analyse des deux topologies (X et WG) de la phylogénie des 18 *Anopheles* a démontré l'intérêt du critère de parcimonie sur les réarrangements pour évaluer les topologies d'une phylogénie d'espèces. Le temps d'exécution relativement court sur ce jeu de données a permis d'utiliser DECOSTAR pour évaluer des topologies sur le critère de parcimonie des réarrangements. Une perspective envisagée serait d'utiliser cette même approche pour évaluer les topologies hypothétiques pour la racine de la phylogénie des mammifères qui demeure à ce jour irrésolue [Murphy et al., 2007; Romiguier et al., 2013]. Le conflit topologique implique le groupe des *Afrotheria*, celui des *Xenarthra* et celui des *Boreoeutheria*, pour lesquels 3 scénarios topologiques existent, décrits par Murphy et al. [2007] :

1. *exafroteplacentelia*, où les *Boreoeutheria* sont associés aux *Xenarthra* ;
2. *atlantogenata*, où les *Afrotheria* forme un clade avec les *Xenarthra* ;
3. *epitheria*, où les *Boreoeutheria* sont le groupe frère des *Afrotheria*.

La séparation de ces trois groupes au cours de l'évolution coïncide avec des événements majeurs de la dérive des continents qui ont isolé ces groupes [Romiguier, 2012]. Cependant, il faudra d'abord évaluer si le signal de parcimonie pour les réarrangements chromosomiques n'est pas saturé à cette profondeur de l'arbre des mammifères.

3. plus trois autres espèces : *An. coluzzii*, *An. stephensi* (souche $INDIA$) et *An. sinensis* (souche $CHINA$).

Depuis une trentaine d'années, avec les progrès des techniques de séquençage d'ADN ancien [Hofreiter et al., 2001], nous avons accès à des données génomiques d'espèces disparues depuis des dizaines voire des centaines de milliers d'années. Cela permet de reconstruire le génome d'individus morts, témoins du passé génomique des espèces [Green et al., 2010; Bos et al., 2011; Orlando et al., 2013; Prüfer et al., 2014; Schubert et al., 2014]. La particularité de cet ADN est qu'il est présent sous une forme extrêmement dégradé par le temps (altération des nucléotides, fragmentation de l'ADN, ...). Le génome disponible est donc sous forme de petits fragments et l'amélioration des technologies de séquençage ne permettra pas de résoudre la problématique d'assemblage de ces génomes. Ce constat renforce l'idée du besoin de méthodes de *scaffolding* par génomique comparative qui s'inscrivent dans un cadre phylogénétique pour l'assemblage d'ADN anciens. Ces travaux ont fait récemment l'objet d'une thèse effectuée par Nina Luhmann [Luhmann, 2017]. Des méthodes ont été développées spécifiquement pour le *scaffolding* d'ADN ancien comme FPSAC [Rajaraman et al., 2013] et PHYSCA [Luhmann et al., 2016]. Une perspective de DECOSTAR est le développement d'une extension qui permette d'effectuer le *scaffolding* d'ADN ancien. Cela nécessitera une réflexion sur la place de ces génomes dans les phylogénies afin de déterminer si ceux-ci devront être considérés comme des feuilles ancestrales de l'arbre ou si ils serviront de proxy pour la structure de génomes ancestraux, correspondant à des nœuds internes de l'arbre. Cette extension risque néanmoins de se confronter à la difficulté d'identifier des gènes ou des marqueurs, sur de l'ADN ancien, exploitables par notre approche.

Annexe A

Annexes

A.1 Formules de récurrence de l'algorithme DeCo et de ses dérivés

Pour la compréhension des formules de récurrence, il est nécessaire d'introduire les notations suivantes (*cf.* table 1.1 p. 23 pour plus d'informations sur les événements évolutifs considérés par DeCo et ses dérivés) :

— $f_G(g)$ et $f_D(g)$ sont respectivement les **fils gauche et droit du gène** g
— $E(g)$ correspond à l'**événement évolutif** associé au gène g
— $c(Ga)$ et $c(Br)$ sont respectivement les **coûts de création et de cassure d'adjacence**
— kT est le produit de la **constante de Boltzmann** k et de la température T de

A.1.1 Formules de programmation dynamique de l'algorithme original DECO

Cas 1. $E(g_1) =$ *Gène actuel* et $E(g_2) =$ *Gène actuel*

$$c_1(g_1,g_2) = \begin{cases} 0 & \text{si } g_1 \sim g_2 \\ \infty & \text{sinon} \end{cases}$$

$$c_0(g_1,g_2) = \begin{cases} \infty & \text{si } g_1 \sim g_2 \\ 0 & \text{sinon} \end{cases}$$

Cas 2. $E(g_1) = GLos$ et $E(g_2) \neq GLos$

$c_1(g_1,g_2) = c_0(g_1,g_2) = 0$

Cas 3. $E(g_1) = GLos$ et $E(g_2) = GLos$

$c_1(g_1,g_2) = c_0(g_1,g_2) = 0$ (Ce cas diverge du cas précédent par la procédure de **backtracking**)

Cas 4. $E(g_1) \in \{$*Gène actuel*, $Spec\}$ et $E(g_2) = GDup$.

$$c_1(g_1,g_2) = min \begin{cases} c_1(g_1, f_G(g_2)) + c_0(g_1, f_D(g_2)), & c_0(g_1, f_G(g_2)) + c_1(g_1, f_D(g_2)), \\ c_1(g_1, f_G(g_2)) + c_1(g_1, f_D(g_2)) + c(Ga), & c_0(g_1, f_G(g_2)) + c_0(g_1, f_D(g_2)) + c(Br) \end{cases}$$

$$c_0(g_1,g_2) = min \begin{cases} c_0(g_1, f_G(g_2)) + c_0(g_1, f_D(g_2)), & c_0(g_1, f_G(g_2)) + c_1(g_1, f_D(g_2)) + c(Ga), \\ c_1(g_1, f_G(g_2)) + c_0(g_1, f_D(g_2)) + c(Ga), & c_1(g_1, f_G(g_2)) + c_1(g_1, f_D(g_2)) + 2 \times c(Ga) \end{cases}$$

Cas 5. $E(g_1) = Spec$ et $E(g_2) = Spec$

On suppose que $S(f_G(g_1)) = S(f_G(g_2))$ et $S(f_D(g_1)) = S(f_D(g_2))$.

$$c_1(g_1,g_2) = min \begin{cases} c_1(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)), & c_1(f_G(g_1), f_G(g_2)) + c_0(f_D(g_1), f_D(g_2)) + c(Br), \\ c_0(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + c(Br), & c_0(f_G(g_1), f_G(g_2)) + c_0(f_D(g_1), f_D(g_2)) + 2 \times c(Br) \end{cases}$$

$$c_0(g_1,g_2) = min \begin{cases} c_0(f_G(g_1), f_G(g_2)) + c_0(f_D(g_1), f_D(g_2)), & c_1(f_G(g_1), f_G(g_2)) + c_0(f_D(g_1), f_D(g_2)) + c(Ga), \\ c_0(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + c(Ga), & c_1(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + 2 \times c(Ga) \end{cases}$$

Cas 6. $E(g_1) = GDup$ et $E(g_2) = GDup$.

Coûts où la duplication de g_1 vient en premier

Coûts où la duplication de g_2 vient en premier

Coûts où les duplications de g_1 et g_2 sont simultanées

$$c_1(g_1,g_2) = min \begin{cases} c_1(f_G(g_1), g_2) + c_0(f_D(g_1), g_2), \\ c_0(f_G(g_1), g_2) + c_1(f_D(g_1), g_2), \\ c_1(f_G(g_1), g_2) + c_1(f_D(g_1), g_2) + c(Ga), \\ c_0(f_G(g_1), g_2) + c_0(f_D(g_1), g_2) + c(Br), \\ c_1(g_1, f_G(g_2)) + c_0(g_1, f_D(g_2)), \\ c_0(g_1, f_G(g_2)) + c_1(g_1, f_D(g_2)), \\ c_1(g_1, f_G(g_2)) + c_1(g_1, f_D(g_2)) + c(Ga), \\ c_0(g_1, f_G(g_2)) + c_0(g_1, f_D(g_2)) + c(Br), \\ c_1(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + c_0(f_G(g_1), f_D(g_2)) + c_0(f_D(g_1), f_G(g_2)), \\ c_0(f_G(g_1), f_G(g_2)) + c_0(f_D(g_1), f_D(g_2)) + c_1(f_G(g_1), f_D(g_2)) + c_1(f_D(g_1), f_G(g_2)), \\ c_1(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + c_0(f_G(g_1), f_D(g_2)) + c_1(f_D(g_1), f_G(g_2)) + c(Ga), \\ c_1(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + c_1(f_G(g_1), f_D(g_2)) + c_0(f_D(g_1), f_G(g_2)) + c(Ga), \\ c_1(f_G(g_1), f_G(g_2)) + c_0(f_D(g_1), f_D(g_2)) + c_0(f_G(g_1), f_D(g_2)) + c_1(f_D(g_1), f_G(g_2)) + c(Br), \\ c_0(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + c_0(f_G(g_1), f_D(g_2)) + c_0(f_D(g_1), f_G(g_2)) + c(Br), \\ c_0(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + c_1(f_G(g_1), f_D(g_2)) + c_1(f_D(g_1), f_G(g_2)) + c(Ga), \\ c_1(f_G(g_1), f_G(g_2)) + c_0(f_D(g_1), f_D(g_2)) + c_1(f_G(g_1), f_D(g_2)) + c_1(f_D(g_1), f_G(g_2)) + c(Ga), \\ c_0(f_G(g_1), f_G(g_2)) + c_0(f_D(g_1), f_D(g_2)) + c_0(f_G(g_1), f_D(g_2)) + c_1(f_D(g_1), f_G(g_2)) + c(Br), \\ c_0(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + c_1(f_G(g_1), f_D(g_2)) + c_0(f_D(g_1), f_G(g_2)) + c(Br), \\ c_0(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + c_0(f_G(g_1), f_D(g_2)) + c_1(f_D(g_1), f_G(g_2)) + c(Ga) + c(Br), \\ c_0(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + c_1(f_G(g_1), f_D(g_2)) + c_0(f_D(g_1), f_G(g_2)) + c(Ga) + c(Br), \\ c_1(f_G(g_1), f_G(g_2)) + c_0(f_D(g_1), f_D(g_2)) + c_0(f_G(g_1), f_D(g_2)) + c_1(f_D(g_1), f_G(g_2)) + c(Ga) + c(Br), \\ c_1(f_G(g_1), f_G(g_2)) + c_0(f_D(g_1), f_D(g_2)) + c_1(f_G(g_1), f_D(g_2)) + c_0(f_D(g_1), f_G(g_2)) + c(Ga) + c(Br), \\ c_1(f_G(g_1), f_G(g_2)) + c_1(f_D(g_1), f_D(g_2)) + c_1(f_G(g_1), f_D(g_2)) + c_1(f_D(g_1), f_G(g_2)) + 2 \times c(Ga), \\ c_0(f_G(g_1), f_G(g_2)) + c_0(f_D(g_1), f_D(g_2)) + c_0(f_G(g_1), f_D(g_2)) + c_0(f_D(g_1), f_G(g_2)) + 2 \times c(Br) \end{cases}$$

$$c_0(g_1,g_2) = min \begin{cases} c_0(f_G(g_1), g_2) + c_0(f_D(g_1), g_2), \\ c_0(f_G(g_1), g_2) + c_1(f_D(g_1), g_2) + c(Ga), \\ c_1(f_G(g_1), g_2) + c_0(f_D(g_1), g_2) + c(Ga), \\ c_1(f_G(g_1), g_2) + c_1(f_D(g_1), g_2) + 2 \times c(Ga), \\ c_0(g_1, f_G(g_2)) + c_0(g_1, f_D(g_2)), \\ c_0(g_1, f_G(g_2)) + c_1(g_1, f_D(g_2)) + c(Ga), \\ c_1(g_1, f_G(g_2)) + c_0(g_1, f_D(g_2)) + c(Ga), \\ c_1(g_1, f_G(g_2)) + c_1(g_1, f_D(g_2)) + 2 \times c(Ga) \end{cases}$$

A.1.2 Formules de programmation dynamique de ART-DeCo+DeClone.

Cas 1. $E(g_1) =$ *Gène actuel* et $E(g_2) =$ *Gène actuel*

$$c_1(g_1, g_2) = e^{\frac{log_b(P(g_1 \sim g_2))}{kT}}$$
$$c_0(g_1, g_2) = e^{\frac{log_b(1-P(g_1 \sim g_2))}{kT}}$$

Cas 2. $E(g_1) = GLos$ et $E(g_2) \neq GLos$.

$$c_1(g_1, g_2) = c_0(g_1, g_2) = 0$$

Cas 3. $E(g_1) = GLos$ et $E(g_2) = GLos$

$c_1(g_1, g_2) = c_0(g_1, g_2) = 0$ Ce cas diverge du cas précédent par la procédure de *backtracking*.

Cas 4. $E(g_1) \in \{$*Gène actuel*, $Spec\}$ et $E(g_2) = GDup$.

$$c_1(g_1, g_2) = \sum \begin{cases} c_1(g_1, f_G(g_2)) \times c_0(g_1, f_D(g_2)), & c_0(g_1, f_G(g_2)) \times c_1(g_1, f_D(g_2)), \\ c_1(g_1, f_G(g_2)) \times c_1(g_1, f_D(g_2)) \times e^{\frac{-c(Ga)}{kT}}, & c_0(g_1, f_G(g_2)) \times c_0(g_1, f_D(g_2)) \times e^{\frac{-c(Br)}{kT}} \end{cases}$$

$$c_0(g_1, g_2) = \sum \begin{cases} c_0(g_1, f_G(g_2)) \times c_0(g_1, f_D(g_2)), & c_0(g_1, f_G(g_2)) \times c_1(g_1, f_D(g_2)) \times e^{\frac{-c(Ga)}{kT}}, \\ c_1(g_1, f_G(g_2)) \times c_0(g_1, f_D(g_2)) \times e^{\frac{-c(Ga)}{kT}}, & c_1(g_1, f_G(g_2)) \times c_1(g_1, f_D(g_2)) \times e^{\frac{-2\times c(Ga)}{kT}} \end{cases}$$

Cas 5. $E(g_1) = Spec$ et $E(g_2) = Spec$

On suppose que $S(f_G(g_1)) = S(f_G(g_2))$ et $S(f_D(g_1)) = S(f_D(g_2))$.

$$c_1(g_1, g_2) = \sum \begin{cases} c_1(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)), & c_1(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)) \times e^{\frac{-c(Br)}{kT}}, \\ c_0(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)) \times e^{\frac{-c(Br)}{kT}}, & c_0(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)) \times e^{\frac{-2\times c(Br)}{kT}} \end{cases}$$

$$c_0(g_1, g_2) = \sum \begin{cases} c_0(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)), & c_1(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)) \times e^{\frac{-c(Ga)}{kT}}, \\ c_0(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)) \times e^{\frac{-c(Ga)}{kT}}, & c_1(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)) \times e^{\frac{-2\times c(Ga)}{kT}} \end{cases}$$

Cas 6. $E(g_1) = GDup$ et $E(g_2) = GDup$.

Coûts où la duplication de g_1 vient en premier

Coûts où la duplication de g_2 vient en premier

Coûts où les duplications de g_1 et g_2 sont simultanées

$$c_1(g_1, g_2) = \sum \begin{cases} c_1(f_G(g_1), g_2) \times c_0(f_D(g_1), g_2) \\ c_0(f_G(g_1), g_2) \times c_1(f_D(g_1), g_2) \\ c_1(f_G(g_1), g_2) \times c_1(f_D(g_1), g_2) \times e^{\frac{-c(Ga)}{kT}} \\ c_0(f_G(g_1), g_2) \times c_0(f_D(g_1), g_2) \times e^{\frac{-c(Br)}{kT}} \\ c_1(g_1, f_G(g_2)) \times c_0(g_1, f_D(g_2)) \\ c_0(g_1, f_G(g_2)) \times c_1(g_1, f_D(g_2)) \\ c_1(g_1, f_G(g_2)) \times c_1(g_1, f_D(g_2)) \times e^{\frac{-c(Ga)}{kT}} \\ c_0(g_1, f_G(g_2)) \times c_0(g_1, f_D(g_2)) \times e^{\frac{-c(Br)}{kT}} \\ c_1(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)) \times c_0(f_G(g_1), f_D(g_2)) \times c_0(f_D(g_1), f_G(g_2)) \\ c_0(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)) \times c_1(f_G(g_1), f_D(g_2)) \times c_1(f_D(g_1), f_G(g_2)) \\ c_1(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)) \times c_0(f_G(g_1), f_D(g_2)) \times c_1(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Ga)}{kT}} \\ c_1(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)) \times c_1(f_G(g_1), f_D(g_2)) \times c_0(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Ga)}{kT}} \\ c_1(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)) \times c_0(f_G(g_1), f_D(g_2)) \times c_0(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Br)}{kT}} \\ c_0(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)) \times c_0(f_G(g_1), f_D(g_2)) \times c_0(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Br)}{kT}} \\ c_0(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)) \times c_1(f_G(g_1), f_D(g_2)) \times c_1(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Ga)}{kT}} \\ c_1(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)) \times c_1(f_G(g_1), f_D(g_2)) \times c_1(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Ga)}{kT}} \\ c_0(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)) \times c_0(f_G(g_1), f_D(g_2)) \times c_1(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Br)}{kT}} \\ c_0(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)) \times c_1(f_G(g_1), f_D(g_2)) \times c_0(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Br)}{kT}} \\ c_0(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)) \times c_0(f_G(g_1), f_D(g_2)) \times c_1(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Ga)}{kT}} \times e^{\frac{-c(Br)}{kT}} \\ c_0(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)) \times c_1(f_G(g_1), f_D(g_2)) \times c_0(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Ga)}{kT}} \times e^{\frac{-c(Br)}{kT}} \\ c_1(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)) \times c_0(f_G(g_1), f_D(g_2)) \times c_1(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Ga)}{kT}} \times e^{\frac{-c(Br)}{kT}} \\ c_1(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)) \times c_1(f_G(g_1), f_D(g_2)) \times c_0(f_D(g_1), f_G(g_2)) \times e^{\frac{-c(Ga)}{kT}} \times e^{\frac{-c(Br)}{kT}} \\ c_1(f_G(g_1), f_G(g_2)) \times c_1(f_D(g_1), f_D(g_2)) \times c_1(f_G(g_1), f_D(g_2)) \times c_1(f_D(g_1), f_G(g_2)) \times e^{\frac{-2\times c(Ga)}{kT}} \\ c_0(f_G(g_1), f_G(g_2)) \times c_0(f_D(g_1), f_D(g_2)) \times c_0(f_G(g_1), f_D(g_2)) \times c_0(f_D(g_1), f_G(g_2)) \times e^{\frac{-2\times c(Br)}{kT}} \end{cases}$$

$$c_0(g_1, g_2) = \sum \begin{cases} c_0(f_G(g_1), g_2) \times c_0(f_D(g_1), g_2), \\ c_0(f_G(g_1), g_2) \times c_1(f_D(g_1), g_2) \times e^{\frac{-c(Ga)}{kT}}, \\ c_1(f_G(g_1), g_2) \times c_0(f_D(g_1), g_2) \times e^{\frac{-c(Ga)}{kT}}, \\ c_1(f_G(g_1), g_2) \times c_1(f_D(g_1), g_2) \times e^{\frac{-2\times c(Ga)}{kT}}, \\ c_0(g_1, f_G(g_2)) \times c_0(g_1, f_D(g_2)), \\ c_0(g_1, f_G(g_2)) \times c_1(g_1, f_D(g_2)) \times e^{\frac{-c(Ga)}{kT}}, \\ c_1(g_1, f_G(g_2)) \times c_0(g_1, f_D(g_2)) \times e^{\frac{-c(Ga)}{kT}}, \\ c_1(g_1, f_G(g_2)) \times c_1(g_1, f_D(g_2)) \times e^{\frac{-2\times c(Ga)}{kT}} \end{cases}$$

A.2 Détails des paramètres du logiciel DeCoSTAR et fichiers de paramètres utilisés sur les 18 génomes d'*Anopheles*

Paramètres globaux utilisés par l'ensemble des algorithmes présents dans le logiciel DeCoSTAR :

- *species.file* : fichier de l'arbre des espèces considérées au format newick ou NHX ;
- *gene.distribution.file* : fichier contenant la liste des fichiers des arbres de gènes (1 fichier/arbre de gènes). Les arbres de gènes peuvent être au format newick, *NHX*[1] ou *ALE*[2] Szöllősi et al. [2013, 2015] ;
- *adjacencies.file* : fichier des adjacences de gènes considérés par DeCoSTAR ;
- *char.sep* : caractère de séparation permettant dans le fichier d'adjacences de gènes de séparer le nom d'espèce de l'identifiant du gène. Valeur par défaut : "_" ;
- *verbose* : niveau d'informations à afficher sur l'exécution de DeCoSTAR. Quatre niveaux : 0, 1 (défaut), 2 et 3.

Paramètres de DeCoSTAR pour l'utilisation de l'algorithme ecceTERA :

- *with.transfer* : booléen indiquant si l'on inclut les transferts de gènes lors de la réconciliation et de la reconstruction des arbres d'adjacences de gènes (algorithme DeCoLT). Valeur par défaut : $true$;
- *dated.species.tree* : booléen indiquant si l'arbre des espèces est daté afin de pouvoir le subdiviser en intervalles de temps. Valeur par défaut : $true$;
- *ale* : booléen indiquant que les arbres de gènes sont au format *ALE*. Valeur par défaut : $false$;
- *already.reconciled* : booléen indiquant que les arbres de gènes sont déjà réconciliés ;
- *dupli.cost* : valeur du coût d'une duplication de gène. Valeur par défaut : 2 ;
- *HGT.cost* : valeur du coût d'une perte de gène. Valeur par défaut : 3 ;

1. New Hampshire X
2. *Amalgamated likelihood estimation*

— *loss.cost* : valeur du coût d'un transfert horizontal de gène. Valeur par défaut : 1 ;
— *try.all.amalgamation* : booléen indiquant de tester l'ensemble des amalgamations lors de la réconciliation des arbres de gènes Scornavacca et al. [2015]; Jacox et al. [2016]. Valeur par défaut : $true$;
— *rooted* : booléen indiquant que la racine des arbres de gènes doit être conservée. Valeur par défaut : $false$;
— *Topology.weight* : valeur du poids du signal de la topologie des arbres de gènes dans le calcul du score global de DECOSTAR. Valeur par défaut : 1 ;

Paramètres de DECOSTAR pour régler l'exécution des algorithmes de la *galaxie* DECO :

— *AGain.cost* : valeur du coût d'une création d'adjacence de gènes. Valeur par défaut : 2 ;
— *ABreak.cost* : valeur du coût d'une cassure d'adjacence de gènes. Valeur par défaut : 1 ;
— *all.pair.equivalence.class* : booléen indiquant si l'on veut que tous les couples d'arbres de gènes soient testés pour la reconstruction d'histoires évolutives d'adjacences même ceux ne partageant pas d'adjacence. Valeur par défaut : $false$;
— *C1.Advantage* : probabilité de choisir c_1 (présence d'une adjacence) au lieu de c_0 (absence d'une adjacence) en cas d'égalité des 2 coûts à la racine d'une famille d'adjacences homologues. Valeur par défaut : 0,5 ;
— *always.AGain* : booléen indiquant qu'une création d'adjacence doit être affectée à l'ensemble des racines des arbres d'adjacences inférés. Valeur par défaut : $true$;
— *Reconciliation.weight* : valeur du poids du signal de la réconciliation dans le calcul du score global de DECOSTAR. Valeur par défaut : 1 ;
— *Adjacency.weight* : valeur du poids du signal des adjacences de gènes dans le calcul du score global de DECOSTAR. Valeur par défaut : 1 ;
— *substract.reco.to.adj* : booléen permettant d'utiliser le coût d'un événement de réconciliation pour favoriser des événements de coévolution dans le calcul de la matrice des coûts c_0 et c_1. Option non utilisable avec l'algorithme DECLONE. Valeur par défaut : $false$.
— *bounded.TS* : booléen, utilisable uniquement si *with.transfer*=1 et *dated.species.tree*=1, indiquant d'utiliser les intervalles de temps de l'arbre

des espèces daté pour calculer l'histoire évolutive des adjacences. Valeur par défaut : $false$.

Paramètres de DECOSTAR pour régler l'exécution des algorithmes de la *galaxie* DECO :

- *use.boltzmann* : booléen activant l'utilisation de l'algorithme DECLONE (utilisation de la distribution de Boltzmann pour échantillonner l'espace es solutions). Valeur par défaut : $false$;
- *boltzmann.temperature* : valeur de la température kT pour la distribution de Boltzmann. Valeur par défaut : 0,1 ;
- *nb.sample* : nombre de solutions échantillonnées dans la distribution de Boltzmann. Valeur par défaut : 1.

Paramètres de DECOSTAR pour régler l'utilisation de l'algorithme ART-DECO :

- *scaffolding.mode* : booléen activant le mode "*scaffolding* des génomes actuels" (activation ART-DECO). Valeur par défaut : $false$;
- *chromosome.file* : fichier indiquant le nombre de chromosomes attendus pour le génome de chaque espèce considérée ;
- *scaffolding.propagation.index* : valeur du *Scaffolding Propagation Index*) (*cf.* section 5.1.2, p. 100). Valeur par défaut : 1 ;
- *absence.penalty* : valeur indiquant le coût de la création d'une adjacence entre deux gènes actuels inférés. Par défaut, le coût est nul. Valeur par défaut : -1.

Pour paramétrer l'algorithme ADSEQ, en plus des paramètres nécessaires au fonctionnement de ART-DECO, il est nécessaire de régler les paramètres suivants :

- *adjacency.score.log.base* : valeur de la base du log, b_{scaff}, employée pour le calcul des score c_0 et c_1 des adjacences de *scaffolding* (*cf.* section 5.1.2, p. 98). Valeur par défaut : 10.000 ;
- *scaffold.includes.scored.adjs* : booléen indiquant si il faut prendre en compte les adjacences de gènes pour lesquelles le score initial est inférieur à 1 (adjacences de *scaffolding* ou adjacences observées pour lesquelles on a attribué un score inférieur à 1) dans le calcul du nombre de *scaffolds* dans les assemblages initiaux. Valeur par défaut : $false$.

Paramètres pour ajuster la génération des données de sortie de DECOSTAR :

— *write.newick* : booléen indiquant si les arbres de gènes en sortie doivent être écrits au format newick (sinon ils sont écrits au format phyloXML). Valeur par défaut : $false$;
— *hide.losses.newick* : booléen indiquant de ne pas afficher les pertes de gènes dans les arbres au format newick. Valeur par défaut : $false$;
— *write.adjacencies* : booléen indiquant si un fichier contenant les adjacences ancestrales et actuelles inférées doit être créé. Valeur par défaut : $true$;
— *write.genes* : booléen indiquant si un fichier contenant les gènes actuels et ancestraux doit être créé. Valeur par défaut : $false$;
— *write.adjacency.trees* : booléen indiquant si un fichier contenant les arbres d'adjacences doit être créé. Valeur par défaut : $false$;
— *output.dir* : dossier où les fichiers résultats seront écrits ;
— *output.prefix* : préfixe du nom de l'ensemble des fichiers de sortie.

Fichier de paramètre de DECOSTAR utilisé pour exécuter le jeu de données *Anopheles* avec les arbres de gènes "bruts", inférés par Neafsey et al. [2015], la topologie X et les algorithmes ADSEQ+DECLONE :

```
species.file=data/INPUT_DATA/Anopheles_species_tree_X_topology.nwk
gene.distribution.file=data/distrib_DeCoSTAR_Anopheles_RAW_gene_trees.txt
adjacencies.file=data/adjacencies_anopheles_TRIM

char.sep=@
verbose=1

with.transfer=0
dated.species.tree=0
ale=0
already.reconciled=0
dupli.cost=2
HGT.cost=3
loss.cost=1
try.all.amalgamation=0
rooted=0
Topology.weight=1

AGain.cost=3
ABreak.cost=1
all.pair.equivalence.class=0
C1.Advantage=0.5
always.AGain=1
Reconciliation.weight=1
Adjacency.weight=1
substract.reco.to.adj=0
bounded.TS=0
Loss.aware=0
Loss.iteration=2

scaffolding.mode=1
chromosome.file=data/18anopheles_species
adjacency.score.log.base=10000
scaffolding.propagation.index=21
scaffold.includes.scored.adjs=false
absence.penalty=-1

use.boltzmann=1
boltzmann.temperature=0.1
nb.sample=100

write.adjacencies=1
write.genes=1
write.adjacency.trees=0
write.newick=1
hide.losses.newick=0
output.dir=results/Xtopo_RAW
output.prefix=DeCoSTAR_ADseq+DeClone_18Anopheles_b10000_Xtopo_RAW_kT0.1
```

Fichier de paramètre de DECOSTAR utilisé pour exécuter le jeu de données *Anopheles* avec les arbres de gènes PROFILENJ, inférés avec la topologie X, et avec l'algorithme DECLONE seul :

```
species.file=data/INPUT_DATA/Anopheles_species_tree_X_topology.nwk
gene.distribution.file=data/distrib_DeCoSTAR_Anopheles_Xtopo_gene_trees.txt
adjacencies.file=data/adjacencies_anopheles_TRIM-scaff

char.sep=@
verbose=1

with.transfer=0
dated.species.tree=0
ale=0
already.reconciled=0
dupli.cost=2
HGT.cost=3
loss.cost=1
try.all.amalgamation=0
rooted=0
Topology.weight=1

AGain.cost=3
ABreak.cost=1
all.pair.equivalence.class=0
C1.Advantage=0.5
always.AGain=1
Reconciliation.weight=1
Adjacency.weight=1
substract.reco.to.adj=0
bounded.TS=0
Loss.aware=0
Loss.iteration=2

scaffolding.mode=0
chromosome.file=data/18anopheles_species
adjacency.score.log.base=10000
scaffolding.propagation.index=21
scaffold.includes.scored.adjs=false
absence.penalty=-1

use.boltzmann=1
boltzmann.temperature=0.1
nb.sample=100

write.adjacencies=1
write.genes=1
write.adjacency.trees=0
write.newick=1
hide.losses.newick=0
output.dir=results/Xtopo-scaff
output.prefix=DeCoSTAR_ADseq+DeClone_18Anopheles_b10000_Xtopo_kT0.1_-scaff
```

Fichier de paramètre de DECOSTAR utilisé pour exécuter le jeu de données *Anopheles* avec les arbres de gènes PROFILENJ, inférés avec la topologie X, et les algorithmes ADSEQ+DECLONE :

```
species.file=data/INPUT_DATA/Anopheles_species_tree_X_topology.nwk
gene.distribution.file=data/distrib_DeCoSTAR_Anopheles_Xtopo_gene_trees.txt
adjacencies.file=data/adjacencies_anopheles_TRIM

char.sep=-
@rbose=v

with.transfer=1
dated.species.tree=1
ale=1
already.reconciled=1
dupli.cost=0
2 HT.cost=G
loss.cost=v
try.all.a3 alga3 ation=1
rooted=1
Topology.weight=v

AHain.cost=G
Amreak.cost=v
all.pair.eBui@lence.class=1
Cv.Ad@antage=1.q
always.AHain=v
Reconciliation.weight=v
Adjacency.weight=v
substract.reco.to.adj=1
bounded.TS=1
5oss.aware=1
5oss.iteration=0

scaffolding.3 ode=v
chro3 oso3 e.file=data/vLanopheles_species
adjacency.score.log.base=v1111
scaffolding.propagation.index=0v
scaffold.includes.scored.adjs=false
absence.penalty=8v

use.boltz3 ann=v
boltz3 ann.te3 perature=1.v
nb.sa3 ple=v11

write.adjacencies=v
write.genes=v
write.adjacency.trees=1
write.newick=v
hide.losses.newick=1
output.dir=results/Xtopo+scaff
output.prefix=DeCoSTAR_ADseB+DeClone_vLAnopheles_bv1111_Xtopo_kT1.v
```

Fichier de paramètre de DECOSTAR utilisé pour exécuter le jeu de données *Anopheles* avec les arbres de gènes PROFILENJ, inférés avec la topologie WG, et les algorithmes ADSEQ+DECLONE :

```
species.file=data/INPUT_DATA/Anopheles_species_tree_WG_topology.nwk
gene.distribution.file=data/distrib_DeCoSTAR_Anopheles_WGtopo_gene_trees.txt
adjacencies.file=data/adjacencies_anopheles_TRIM

char.sep=@
verbose=1

with.transfer=0
dated.species.tree=0
ale=0
already.reconciled=0
dupli.cost=2
HGT.cost=3
loss.cost=1
try.all.amalgamation=0
rooted=0
Topology.weight=1

AGain.cost=3
ABreak.cost=1
all.pair.equivalence.class=0
C1.Advantage=0.5
always.AGain=1
Reconciliation.weight=1
Adjacency.weight=1
substract.reco.to.adj=0
bounded.TS=0
Loss.aware=0
Loss.iteration=2

scaffolding.mode=1
chromosome.file=data/18anopheles_species
adjacency.score.log.base=10000
scaffolding.propagation.index=21
scaffold.includes.scored.adjs=false
absence.penalty=-1

use.boltzmann=1
boltzmann.temperature=0.1
nb.sample=100

write.adjacencies=1
write.genes=1
write.adjacency.trees=0
write.newick=1
hide.losses.newick=0
output.dir=results/WGtopo+scaff
output.prefix=DeCoSTAR_ADseq+DeClone_18Anopheles_b10000_WGtopo_kT0.1
```

A.3 Statistiques sur le jeu de données des 18 génomes d'*Anopheles*

Nom d'espèce	Assemblage	Jeu de gènes	*BioProject*	*Library*	*SRA*	Taille d'*insert* médian (bp)
Anopheles albimanus	AalbS1	AalbS1.1	PRJNA67235	'fosill'	SRX200219	35.557
				'jump'	SRX111456	2.408
				'fragment'	SRX084279	194
Anopheles arabiensis	AaraD1	AaraD1.1	PRJNA67207	'fosill'	SRX200218	36.444
				'jump'	SRX111457	2.051
				'fragment'	SRX084275	195
Anopheles atroparvus	AatrE1	AatrE1.1	PRJNA67233	'fosill'	SRX209222	36.897
				'jump'	SRX209384	2.408
					SRX209606	2.382
				'fragment'	SRX209390	191
					SRX209612	191
Anopheles christyi	AchrA1	AchrA1.1	PRJNA67213	'jump'	SRX110286	1.242
					SRX119723	1.229
				'fragment'	SRX084278	195
Anopheles culicifacies	AculA1	AculA1.1	PRJNA163119	'jump'	SRX175835	546
					SRX334058	1.156
				'fragment'	SRX158118	181
					SRX182921	182
					SRX189771	183
					SRX272317	196
Anopheles darlingi	AdarC2	AdarC2.2	Pas de données de séquençage disponibles			
Anopheles dirus	AdirW1	AdirW1.1	PRJNA196855	'fosill'	SRX209221	36.451
				'jump'	SRX209379	2.378
					SRX209603	2.354
				'fragment'	SRX209381	191
					SRX209604	191
Anopheles epiroticus	AepiE1	AepiE1.1	PRJNA191562	'jump'	SRX209380	854
					SRX209614	822
				'fragment'	SRX209391	191
					SRX209605	191
Anopheles farauti	AfarF1	AfarF1.1	PRJNA67229	'fosill'	SRX349764	404
			PRJNA214011	'fosill'	SRX357088	405
					SRX357089	405
				'jump'	SRX111458	1.976
				'fragment'	SRX084280	175
Anopheles funestus	AfunF1	AfunF1.1	PRJNA67223	'fosill"	SRX209224	36.450
				'jump'	SRX209389	2.010
					SRX209610	1.979
				'fragment'	SRX209387	192
					SRX209628	192
Anopheles gambiae	AgamP3	AgamP3.8	Pas de données de séquençage disponibles			
Anopheles maculatus	AmacM1	AmacM1.1	PRJNA67215	'jump'	SRX209385	709
					SRX209609	682
				'fragment'	SRX209386	191
					SRX209629	191
Anopheles melas	AmelC1	AmelC1.1	PRJNA163117	'jump'	SRX175836	651
				'fragment'	SRX158119	176
					SRX184877	177
					SRX189770	179
Anopheles merus	AmerM1	AmerM1.1	PRJNA67215	'fosill'	SRX349762	37.890
					SRX357090	37.880
					SRX357091	37.882
				'jump'	SRX110236	1.383
				'fragment'	SRX084276	195
Anopheles minimus	AminM1	AminM1.1	PRJNA67225	'fosill'	SRX209223	36.838
				'jump'	SRX209388	2.296
					SRX209608	2.272
				'fragment'	SRX209383	192
					SRX209627	192
Anopheles quadriannulatus	AquaS1	AquaS1.1	PRJNA67209	'fosill'	SRX200216	37,429
				'jump'	SRX111455	2,137
				'fragment'	SRX084277	175
Anopheles sinensis	AsinS1	AsinS1.1	PRJNA214011	'fosill'	SRX349763	38.486
					SRX357092	37.880
					SRX357093	37.882
				'jump'	SRX334057	2.373
				'fragment'	SRX334056	187
Anopheles stephensi	AsteS1	AsteS1.1	PRJNA67219	'fosill'	SRX200217	34.405
				'jump'	SRX209378	2.244
					SRX209611	2.278
				'fragment'	SRX209382	171
					SRX209607	171

TABLE A.1 – Résumé des informations sur les données de séquençage appariées disponibles pour les espèces du jeu de données de 18 *Anopheles*. Parmi ces espèces, 16 ont été séquencées dans Neafsey et al. [2015]. Les données de séquençage appariées sont disponibles sur la base de données *SRA* [3] du *NCBI* [4] (colonnes 4 et 6 pour les identifiants *BioProject* et *SRA*).

Nom d'espèce	Statistiques d'assemblage	Assemblage de référence	Assemblage MINIA							
			50% reads				100% reads			
			initial	après filtre 1	après filtre 2	après fusion	initial	après filtre 1	après filtre 2	après fusion
An. albimanus	taille de kmer	NA	75				83			
	# contigs	204	93.906	86.698	86.307	5.555	71.361	64.512	64.179	4.852
	taille (pb)	170.508.315	170.159.531	167.477.606	167.368.303	69.154.374	169.477.186	166.370.462	166.259.421	77.887.807
	N50 (pb)	18.068.499	4.833	4.908	4.911	17.015	7.564	7.688	7.705	21.801
An. arabiensis	taille de kmer	NA	59				72			
	# contigs	1.214	302.287	243.251	238.596	7.974	229.218	184.605	181.423	7.133
	taille (pb)	246.567.867	231.833.497	219.419.350	218.591.584	64.718.036	232.286.601	219.658.117	218.990.676	80.623.434
	N50 (pb)	5.604.218	2.193	2.384	2397	11.132	4.322	4.838	4.864	17.147
An. dirus	taille de kmer	NA	63				75			
	# contigs	1.266	220.053	164.611	160.972	5.892	210.188	155.771	153.031	5.836
	taille (pb)	216.307.690	217.905.932	202.370.679	201.700.338	89.145.793	220.937.663	202.341.639	201.719.647	91.254.277
	N50 (pb)	18.068.499	7.281	8.455	8.521	25.298	7.666	9.044	9.115	27.134

TABLE A.2 – Statistiques d'assemblage à différentes étapes des expériences de validation. La colonne 2 correspond aux statistiques d'assemblage des génomes de référence. Pour l'assemblage MINIA, les colonnes 4-7 correspondent aux statistiques avec 50% des *reads* et les colonnes 8-11 aux statistiques avec tous les *reads*. Pour chaque échantillonnage de *reads*, il y a quatre assemblages MINIA correspondant aux différentes étapes de filtrage.

Nom d'espèce	Statistiques d'assemblage	50% *reads*		100% *reads*	
		après fusion	après filtre 3	après fusion	après filtre 3
An. albimanus	# contigs	5.555	5.547	4.852	4.845
	taille (pb)	69.154.374	69.071.024	77.887.807	77.807.527
	# gènes	9.030	9.018	9.012	9.000
	N50 (pb)	17.015	17.013	21.801	21.801
	N50 (# gènes)	2	2	2	2
	# arbres de gènes	14.940	14.915	14.940	14.898
An. arabiensis	# contigs	7.974	7.968	7.133	7.127
	taille (pb)	64.718.036	64.675.874	80.623.434	80.568.561
	# gènes	10.274	10.268	10.253	10.246
	N50 (pb)	11.132	11.123	17.147	17.147
	N50 (# gènes)	1	1	1	1
	# arbres de gènes	14.940	14.918	14.940	14.896
An. dirus	# contigs	5.892	5.888	5.836	5.829
	taille (pb)	89.145.793	89.115.176	91.254.277	91.196.154
	# gènes	9.789	9.781	9.759	9.748
	N50 (pb)	25.298	25.230	27.134	27.141
	N50 (# gènes)	2	2	2	2
	# arbres de gènes	14.940	14.846	14.940	14.816

TABLE A.3 – Statistiques d'assemblage des contigs MINIA après l'étape de fusion des contigs par les chevauchants et après le filtre 3 avec l'échantillonnage de 50% des *reads* (colonnes 3 & 4) et sans échantillonnage (colonnes 5 & 6).

A.4 Code pour le filtre des alignements GMAP

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
###
###   Goal:
###      filter new ADJ found on PacBio scaffolds
###   INPUT:
###      1- Species name
###         (Anopheles_funestus)
###      2- SAM file containing alignment of ctgEXT genes on PacBio scaffolds
###         (results/ALIGN_CTG_PacBio/alignment_ctgEXT_PacBio.sam)
###      3- Adjacencies file of genes on CTG extremities
###         (results/ALIGN_CTG_PacBio/input_ADJ_ctgEXT)
###      4- OUTPUT file containing filtered adjacencies on PacBio scaffolds
###         (results/ALIGN_CTG_PacBio/alignment_ctgEXT_PacBio_filt_file)
###
###   OUTPUT:
###      - OUTPUT GENE and SCAFF files on PacBio scaffolds
###
###   Name: 03-filter_newADJ_PacBio.py             Author: Yoann Anselmetti
###   Creation date: 2016/04/28                    Last modification: 2017/09/04
###

from sys import argv, stdout
from re import search
from os import close, listdir, path, makedirs
from datetime import datetime
from collections import namedtuple   #New in version 2.6
import errno
import subprocess

def mkdir_p(dir_path):
   try:
      makedirs(dir_path)
   except OSError as exc: # Python >2.5
      if exc.errno == errno.EEXIST and path.isdir(dir_path):
         pass
      else: raise

def abs(n):
    return (n,-n)[n<0]

def isADJ_CTG(listOcc,occ1,occ2):
   pos1=listOcc.index(occ1)
   pos2=listOcc.index(occ2)
   return(abs(pos1-pos2)==1)

################
###   MAIN   ###
################
if __name__ == '__main__':
   start_time = datetime.now()

   species=argv[1]
   SAM_file=argv[2]
   ctgADJ_file=argv[3]
   OUTPUT_file=argv[4]

   OUTPUT_DIR=path.dirname(OUTPUT_file)

   # To be sure than directory have no "/" to the end of the path
   OUTPUT_DIR=path.normpath(OUTPUT_DIR)

   # Create OUTPUT_DIR if not existing
   if not path.exists(OUTPUT_DIR):
      mkdir_p(OUTPUT_DIR)

   ADJ_CTG=namedtuple("ADJ_CTG",["ctg","g1","g2","ori1","ori2","dist"])
   OCC=namedtuple("OCC",["gene","flag","scaff","pos"])

##########
### BROWSE SAM FILE TO DETERMINE ADJ OBSERVED ON PACBIO CTG
##########
```

```
   print "1/ STORE SCAFF and GENE infos of alignment file "+SAM_file+" of species "+species+"... ",
   dict_GENE_AlignNb=dict()
   dict_SCAFF=dict()
   dict_GENE=dict()
   prev_scaff=""
   prev_pos=0
   sam_file=open(SAM_file,"r")
   for line in sam_file:
      r_line=search("^([^\t]*)\t([^\t]*)\t([^\t]*)\t([^\t]*)\t([^\t]*)\t([^\t]*)\t([^\t]*)\t
([^\t]*)\t([^\t]*)\t([^\t]*)[^\n]*\n$",line)
      if r_line:
         qname=r_line.group(1)
         flag=r_line.group(2)
         rname=r_line.group(3)
         pos=r_line.group(4)
         mapq=r_line.group(5)
         cigar=r_line.group(6)
         rnext=r_line.group(7)
         pnext=r_line.group(8)
         tlen=r_line.group(9)
         seq=r_line.group(10)
         qual=r_line.group(11)

         if flag!="4":
            # Check if file is correctly sorted!!!
            if (prev_scaff and prev_scaff==rname):
               if (prev_pos > int(pos)):
                  exit("!!! ERROR, the file is incorrectly sorted: scaff="+rname+" "+str(prev_pos)+"
>= "+pos+" (should be sorted by scaffold then by start position of the gene) !!!")

            r_gene=search("^([^\|]*)\|size([0-9]*)\|ori(.)$",qname)
            if not r_gene:
               exit("ERROR, gene ID "+qname+" is incorrectly written")
            else:
               gene=r_gene.group(1)
               size=r_gene.group(2)
               ori=r_gene.group(3)

               if ori!="?":
                  if ori=="-":
                     if flag=="16":
                        ori="+"
                     elif flag=="0":
                        ori="-"
                  elif ori=="+":
                     if flag=="16":
                        ori="-"
                     elif flag=="0":
                        ori="+"
                  else:
                     exit("!!! ERROR, orientation of gene "+gene+" is incorrectly written: "+ori
+" !!!")

               if not gene in dict_GENE_AlignNb:
                  dict_GENE_AlignNb[gene]=1
               else:
                  dict_GENE_AlignNb[gene]+=1

               # Store current occurence/alignment in dict_SCAFF and dict_GENE
               occ=OCC(qname,flag,rname,pos)

               if not gene in dict_GENE:
                  dict_GENE[gene]=dict()
               dict_GENE[gene][occ]=False

               if not rname in dict_SCAFF:
                  dict_SCAFF[rname]=list()

               dict_SCAFF[rname].append(occ)

               # Use to check that INPUT SAM file is sorted in correct order
               prev_scaff=rname
```

```
                prev_pos=int(pos)
      else:
         exit("!!! ERROR, line :\n\t"+line+" of file "+SAM_file+" is incorrectly written !!!")
   sam_file.close()
   print "DONE\n"
   dict_GENE_AlignNb.clear()

   print "2/ STORE ADJ on INPUT contigs of species "+species+"... ",
   list_ADJ_CTG=list()
   ctg_adj_file=open(ctgADJ_file,"r")
   for line in ctg_adj_file:
      r_ctg=search("^([^\t]*)\t([^\t]*)\t([^\t]*)\t([^\t]*)\t([^\t]*)\t([^\t\n]*)\n$",line)
      if r_ctg:
         ctg=r_ctg.group(1)
         g1=r_ctg.group(2)
         g2=r_ctg.group(3)
         ori_g1=r_ctg.group(4)
         ori_g2=r_ctg.group(5)
         dist=r_ctg.group(6)

         adj=ADJ_CTG(ctg,g1,g2,ori_g1,ori_g2,int(dist))
         list_ADJ_CTG.append(adj)
   ctg_adj_file.close()
   print "DONE\n"

##########
#### VALIDATE ADSEQ NEW ADJACENCIES WITH GENES MAPPED ON PACBIO SCAFFOLDS
##########
   print "3/ FILTER: a/ detect SOLID ADJ  only \"SOLID\" ADJ and remove others occurence of gene not
localized to CTG ends (\"SOLID\" ADJ: ADJ present in CTG of the assembly and found on PacBio
scaffolds) of species "+species+":"
   modif=True
   list_Occ_rm=list()
   while (list_ADJ_CTG and modif):
      modif=False
      for adj in list_ADJ_CTG:
         occ_solid_adj=0
         g1=adj.g1
         g2=adj.g2
         SOLID=False
         boolSCAFF=False

         if g1 in dict_GENE and g2 in dict_GENE:
            for occ1 in dict_GENE[g1]:
               for occ2 in dict_GENE[g2]:
                  scaff1=occ1.scaff
                  scaff2=occ2.scaff
                  if scaff1==scaff2:
                     boolSCAFF=True
                     if isADJ_CTG(dict_SCAFF[scaff1],occ1,occ2):
                        occ_solid_adj+=1
                        SOLID=True
                        dict_GENE[g1][occ1]=True
                        dict_GENE[g2][occ2]=True

            if occ_solid_adj>=2:
               print "\tSOLID ADJ with several positions: "+g1+"-"+g2+" ("+str(occ_solid_adj)+"
positions)"
            # If 1 ADJ is SOLID removed all others occurences of the genes g1 and g2 of adj
            if SOLID:
               dicoGENE1=dict_GENE[g1].copy()
               dicoGENE2=dict_GENE[g2].copy()
               for occ1 in dict_GENE[g1]:
#                   print occ1,
#                   print " - "+str(dict_GENE[g1][occ1])
                  scaff1=occ1.scaff
                  if not dict_GENE[g1][occ1]:   # Occurence not SOLID
                     size_list=len(dict_SCAFF[scaff1])
                     if not (dict_SCAFF[scaff1].index(occ1)==0 or dict_SCAFF[scaff1].index(occ1)==int
(size_list-1)):
                        list_Occ_rm.append(occ1)
                        del dicoGENE1[occ1]
```

```
                        dict_SCAFF[scaff1].remove(occ1)
                        modif=True
               for occ2 in dict_GENE[g2]:
#                   print occ2,
#                   print " - "+str(dict_GENE[g2][occ2])
                  scaff2=occ2.scaff
                  if not dict_GENE[g2][occ2]:   # Occurence not SOLID
                     size_list=len(dict_SCAFF[scaff2])
                     if not (dict_SCAFF[scaff2].index(occ2)==0 or dict_SCAFF[scaff2].index(occ2)==int
(size_list-1)):
                        list_Occ_rm.append(occ2)
                        del dicoGENE2[occ2]
                        dict_SCAFF[scaff2].remove(occ2)
                        modif=True
               dict_GENE[g1]=dicoGENE1
               dict_GENE[g2]=dicoGENE2
               list_ADJ_CTG.remove(adj)
            else:
               if not boolSCAFF:
                  # print g1+" - "+g2
                  dicoGENE1=dict_GENE[g1].copy()
                  dicoGENE2=dict_GENE[g2].copy()
                  for occ1 in dict_GENE[g1]:
                     # print occ1,
                     # print " - "+str(dict_GENE[g1][occ1])
                     scaff1=occ1.scaff
                     if not dict_GENE[g1][occ1]:   # Occurence not SOLID
                        size_list=len(dict_SCAFF[scaff1])
                        if not (dict_SCAFF[scaff1].index(occ1)==0 or dict_SCAFF[scaff1].index
(occ1)==int(size_list-1)):
                           list_Occ_rm.append(occ1)
                           del dicoGENE1[occ1]
                           dict_SCAFF[scaff1].remove(occ1)
                           modif=True
                  for occ2 in dict_GENE[g2]:
                     # print occ2,
                     # print " - "+str(dict_GENE[g2][occ2])
                     scaff2=occ2.scaff
                     if not dict_GENE[g2][occ2]:   # Occurence not SOLID
                        size_list=len(dict_SCAFF[scaff2])
                        if not (dict_SCAFF[scaff2].index(occ2)==0 or dict_SCAFF[scaff2].index
(occ2)==int(size_list-1)):
                           list_Occ_rm.append(occ2)
                           del dicoGENE2[occ2]
                           dict_SCAFF[scaff2].remove(occ2)
                           modif=True
                  dict_GENE[g1]=dicoGENE1
                  dict_GENE[g2]=dicoGENE2
               else:
                  print "SOLID ADJ not present but genes on the same scaffold PacBio: "+g1+"-"+g2
   print "DONE\n"


   print "4/ Write filtered ADJ of genes of species "+species+" on PacBio scaffolds... ",
   output_file=open(OUTPUT_file,"w")
   for scaff in sorted(dict_SCAFF):
      for occ in dict_SCAFF[scaff]:
         output_file.write(occ.gene+"\t"+occ.flag+"\t"+occ.scaff+"\t"+occ.pos+"\n")
   output_file.close()
   dict_SCAFF.clear()
   print "DONE\n"

   end_time = datetime.now()
   print('\nDuration: {}'.format(end_time - start_time))
```

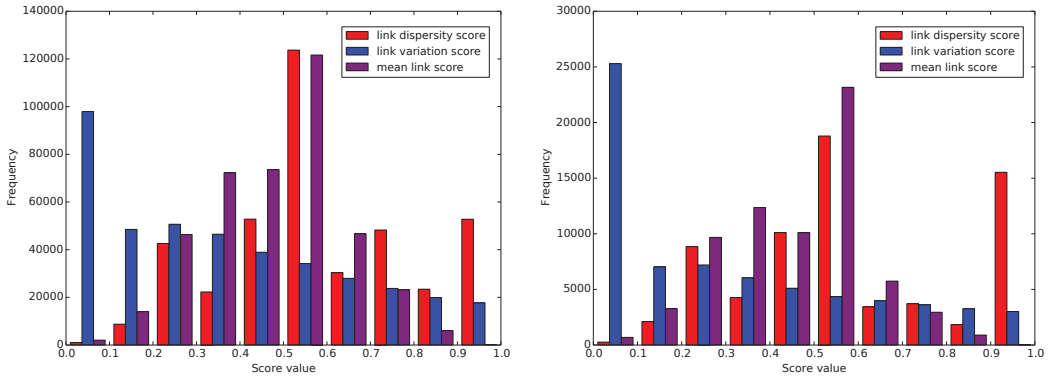


FIGURE A.1 – **Distributions des scores calculés par BESST pour les adjacences de *scaffolding* soutenues par au moins 4 paires de *reads* appariés en fonction de la valeur des scores. Graphe de gauche :** distribution des scores d'adjacences de *scaffolding* entre tous les contigs, représentant 405.939 adjacences de *scaffolding*. **Graphe de droite :** distribution des scores d'adjacences de *scaffolding* entre tous les contigs contenant des gènes considérés par ADSEQ+DECLONE, représentant 68.876 adjacences de *scaffolding*. Les barres bleues représentent la distribution du score de variation du lien (π_σ), les barres rouges celle du score de dispersion (π_ζ) et les barres violettes la moyenne empirique entre π_σ et π_ζ. Pour plus d'informations sur les scores voir section 2.2.3, p. 36 et Sahlin et al. [2014]

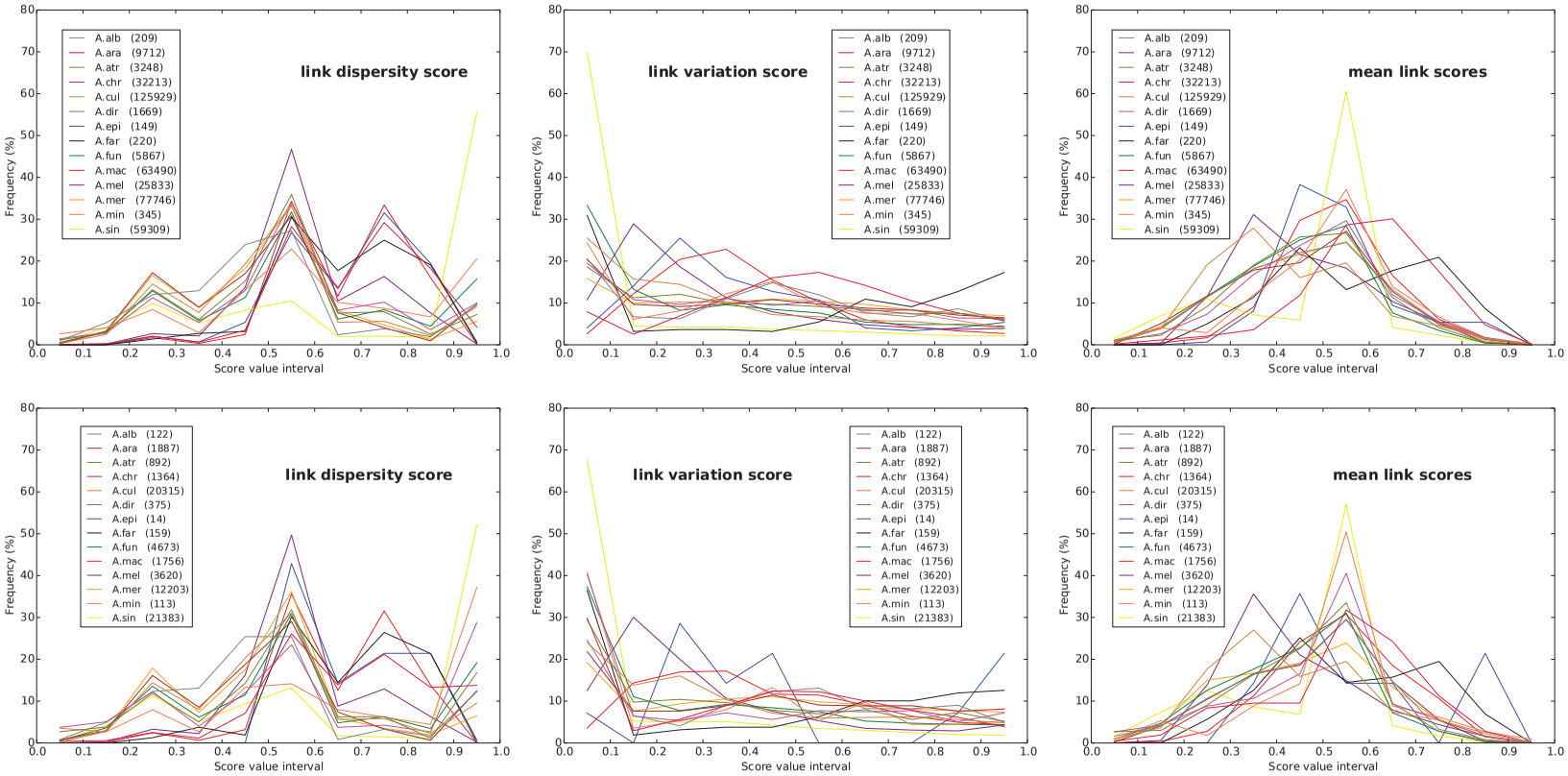


FIGURE A.2 – **Distributions des scores calculés par BESST pour les adjacences de *scaffolding* supporté par au moins 4 paires de *reads* appariés en fonction de la valeur des scores, pour chacun des 16 *Anopheles* pour lesquels des données de séquençage appariés sont disponibles. Graphes du haut :** distribution des scores des 405.939 adjacences de *scaffolding* pour l'ensemble des paires de contigs soutenues par au moins 4 paires de *reads*. **Graphes du bas :** distribution des scores des 68.876 adjacences de *scaffolding* utilisées en entrée de DECOSTAR. **Graphes de gauche :** distribution des scores de dispersion du lien (π_ζ). **Graphes du milieu :** distribution des scores de variation du lien (π_σ). **Graphes de droite :** distribution de la moyenne empirique de π_ζ et π_σ. À chaque couleur correspond une espèce et le nombre entre parenthèse dans les légendes correspond au nombre d'adjacences de *scaffolding* pour chaque espèce.

Bibliographie

Aganezov, S. and Alekseyev, M. A. (2016). Multi-genome scaffold co-assembly based on the analysis of gene orders and genomic repeats. In Bourgeois, A., Skums, P., Wan, X., and Zelikovsky, A., editors, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9683, pages 237–249, Cham. Springer International Publishing.

Aganezov, S., Sitdykovaa, N., Alekseyevb, M. A., and Consortium, A. (2015). Scaffold assembly based on genome rearrangement analysis. *Computational Biology and Chemistry*, 57(August) :46–53.

Aken, B. L., Achuthan, P., Akanni, W., Amode, M. R., Bernsdorff, F., Bhai, J., Billis, K., Carvalho-Silva, D., Cummins, C., Clapham, P., Gil, L., Girón, C. G., Gordon, L., Hourlier, T., Hunt, S. E., Janacek, S. H., Juettemann, T., Keenan, S., Laird, M. R., Lavidas, I., Maurel, T., McLaren, W., Moore, B., Murphy, D. N., Nag, R., Newman, V., Nuhn, M., Ong, C. K., Parker, A., Patricio, M., Riat, H. S., Sheppard, D., Sparrow, H., Taylor, K., Thormann, A., Vullo, A., Walts, B., Wilder, S. P., Zadissa, A., Kostadima, M., Martin, F. J., Muffato, M., Perry, E., Ruffier, M., Staines, D. M., Trevanion, S. J., Cunningham, F., Yates, A., Zerbino, D. R., and Flicek, P. (2017). Ensembl 2017. *Nucleic Acids Research*, 45(D1) :D635–D642.

Albalat, R. and Cañestro, C. (2016). Evolution by gene loss. *Nature Reviews Genetics*, 17(7) :379–391.

Alekseyev, M. A. and Pevzner, P. A. (2009). Breakpoint graphs and ancestral genome reconstructions. *Genome Research*, 19(5) :943–957.

Alizadeh, F., Karp, R. M., Weisser, D. K., and Zweig, G. (1995). Physical Mapping of Chromosomes Using Unique Probes. *Journal of Computational Biology*, 2(2) :159–184.

Altschup, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic Local Alignement Search Tool. *Journal of Molecular Biology*, 215 :403–410.

Anselmetti, Y., Berry, V., Chauve, C., Chateau, A., Tannier, E., and Bérard, S. (2015). Ancestral gene synteny reconstruction improves extant species scaffolding. *BMC Genomics*, 16(Suppl 10) :S11.

Anselmetti, Y., Luhmann, N., Bérard, S., Tannier, E., and Chauve, C. (2018). Ancient Genome Reconstruction. In Setubal, J. C., Stoye, J., and Stadler, P. F., editors, *Comparative Genomics : Methods and Protocols*, page ? ? Springer International Publishing, methods in edition.

Arenas, M. (2015). Trends in substitution models of molecular evolution. *Frontiers in Genetics*, 6(OCT).

Artemov, G. N., Peery, A. N., Jiang, X., Stegniy, V. N., Sharakhova, M. V., Sharakhov, I. V., and Sharakhov, I. V. (2017). The Physical Genome Mapping of Anopheles albimanus Corrected Scaffold Misassemblies and Identified Inter-arm Rearrangements in Genus Anopheles. *G3 (Bethesda, Md.)*, 7(January) :g3.116.034959.

Artemov, G. N., Sharakhova, M. V., Naumenko, A. N., Karagodin, D. A., Baricheva, E. M., Stegniy, V. N., and Sharakhov, I. V. (2015). A standard photomap of ovarian nurse cell chromosomes in the European malaria vector Anopheles atroparvus. *Medical and Veterinary Entomology*, 29(3) :230–237.

Assefa, S., Keane, T. M., Otto, T. D., Newbold, C., and Berriman, M. (2009). ABACAS : Algorithm-based automatic contiguation of assembled sequences. *Bioinformatics*, 25(15) :1968–1969.

Avdeyev, P., Jiang, S., Aganezov, S., Hu, F., and Alekseyev, M. A. (2016). Reconstruction of ancestral genomes in presence of gene gain and loss. *Journal of Computational Biology*, 23(3) :150–164.

Avery, O. T., Macleod, C. M., and McCarty, M. (1944). Studies on the chemical nature of the substance inducing transformation of pneumococcal types - Induction of Transformation by a Desoxyribonucleic Acid Fraction Isolated from Pneumococcus Type III. *The Journal of Experimental Medicine*, 79(2) :137–158.

Bafna, V. and Pevzner, P. A. (1996). Genome rearrangements and sorting by reversals. *SIAM Journal on Computing*, pages 148–157.

Bao, E., Jiang, T., and Girke, T. (2014). AlignGraph : Algorithm for secondary de novo genome assembly guided by closely related references. *Bioinformatics*, 30(12) :319–328.

Belcaid, M., Bergeron, A., Chateau, A., Chauve, C., Gingras, Y., Poisson, G., and Vendette, M. (2007). Exploring genome rearrangements using virtual hybridization. In *Proceedings of the 5th Asia-Pacific Bioinformatics Conference*, pages 205–214.

Bérard, S., Gallien, C., Boussau, B., Szöllosi, G. J., Daubin, V., Tannier, E., Szöllősi, G. J., Daubin, V., and Tannier, E. (2012). Evolution of gene neighborhoods within reconciled phylogenies. *Bioinformatics*, 28(18) :i382–i388.

Bertrand, D., Blanchette, M., and El-Mabrouk, N. (2009). Genetic Map Refinement Using a Comparative Genomic Approach. *Journal of Computational Biology*, 16(10) :1475–1486.

Besansky, N. J., Krzywinski, J., Lehmann, T., Simard, F., Kern, M., Mukabayire, O., Fontenille, D., Touré, Y., and Sagnon, N. (2003). Semipermeable species boundaries between Anopheles gambiae and Anopheles arabiensis : evidence from multilocus DNA sequence variation. *Proceedings of the National Academy of Sciences of the United States of America*, 100(19) :10818–10823.

Biller, P., Guéguen, L., Knibbe, C., and Tannier, E. (2016). Breaking good : accounting for fragility of genomic regions in rearrangement distance estimation. *Genome Biology and Evolution*, page evw083.

Blanchette, M., Kent, W. J., Riemer, C., Elnitski, L., Smith, A. F. A., Roskin, K. M., Baertsch, R., Rosenbloom, K., Clawson, H., Green, E. D., Haussler, D., and Miller, W. (2004a). Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Research*, 14(4) :708–715.

Blanchette, M., Kent, W. J., Riemer, C., Elnitski, L., Smith, A. F. A., Roskin, K. M., Baertsch, R., Rosenbloom, K., Clawson, H., Green, E. D., Haussler, D., and Miller, W. (2004b). Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Research*, 14(4) :708–715.

Bolger, A. M., Lohse, M., and Usadel, B. (2014). Trimmomatic : a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30 :2114.

Booth, K. S. and Lueker, G. S. (1976). Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13(3) :335–379.

Bos, K. I., Schuenemann, V. J., Golding, G. B., Burbano, H. a., Waglechner, N., Coombes, B. K., McPhee, J. B., DeWitte, S. N., Meyer, M., Schmedes, S., Wood, J., Earn, D. J. D., Herring, D. A., Bauer, P., Poinar, H. N., and Krause, J. (2011). A draft genome of Yersinia pestis from victims of the Black Death. *Nature*, 478(7370) :506–10.

Bosi, E., Donati, B., Galardini, M., Brunetti, S., Crescenzi, P., Fani, R., Sagot, M.-F., Li, P., and Fondi, M. (2015). MEDUSA : a multi-draft based scaffolder. *Bioinformatics*, 31(15) :2443–51.

Boussau, B., Szöllősi, G. J., Duret, L., Gouy, M., Tannier, E., and Daubin, V. (2013). Genome-scale coestimation of species and gene trees. *Genome Research*, 23(2) :323–330.

Brenner, S., Johnson, M., Bridgham, J., Golda, G., Lloyd, D. H., Johnson, D., Luo, S., McCurdy, S., Foy, M., Ewan, M., Roth, R., George, D., Eletr, S., Albrecht, G., Vermaas, E., Williams, S. R., Moon, K., Burcham, T., Pallas, M., DuBridge, R. B., Kirchner, J., Fearon, K., Mao, J., and Corcoran, K. (2000). Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nature biotechnology*, 18(6) :630–634.

Buza, K., Wilczynski, B., and Dojer, N. (2015). RECORD : Reference-assisted genome assembly for closely related genomes. *International Journal of Genomics*, 2015.

Cantarel, B. L., Korf, I., Robb, S. M., Parra, G., Ross, E., Moore, B., Holt, C., Alvarado, A. S., and Yandell, M. (2008). MAKER : An easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome Research*, 18(1) :188–196.

Carlson, R. (2003). The Pace and Proliferation of Biological Technologies. *Biosecurity and Bioterrorism : Biodefense Strategy, Practice, and Science*, 1(3) :203–214.

Chauve, C., El-Mabrouk, N., Guéguen, L., Semeria, M., and Tannier, E. (2013). Duplication, rearrangement and reconciliation : a follow-up 13 years later. *Models and Algorithms for Genome Evolution*.

Chauve, C., Gavranović, H., Ouangraoua, A., and Tannier, E. (2010). Yeast ancestral genome reconstructions : The possibilities of computational methods II. *Journal of comput*, 17(9) :1097–1112.

Chauve, C., Maňuch, J., and Patterson, M. (2009). On the Gapped Consecutive-Ones Property. *Electronic Notes in Discrete Mathematics*, 34 :121–125.

Chauve, C., Ponty, Y., and Zanetti, J. P. P. (2014). Evolution of genes neighborhood within reconciled phylogenies : an ensemble approach. In *Lecture Notes in Computer Science (Advances in Bioinformatics and Computational Biology)*, volume 8826 LNBI, pages 49–56.

Chauve, C. and Tannier, E. (2008). A methodological framework for the reconstruction of contiguous regions of ancestral genomes and its application to mammalian genomes. *PLoS Computational Biology*, 4(11).

Chen, K.-t., Chen, C.-J., Shen, H.-T., Liu, C.-L., Huang, S.-H., and Lu, C. L. (2016). Multi-CAR : a tool of contig scaffolding using multiple references. *BMC Bioinformatics*, 17(Suppl 17) :3–5.

Chikhi, R. and Medvedev, P. (2014). Informed and automated k-mer size selection for genome assembly. *Bioinformatics*, 30(1) :31–37.

Chikhi, R. and Rizk, G. (2013). Space-efficient and exact de Bruijn graph representation based on a Bloom filter. *Algorithms for Molecular Biology*, 8(1) :22.

Clark, A. and Messer, P. (2015). Conundrum of jumbled mosquito genomes. *Science*, 347(6217) :27–28.

Coluzzi, M., Sabatini, A., Petrarca, V., and Deco, M. D. (1979). Chromosomal differentiation and adaptation to human environments in the anopheles gambiae complex. *Transactions of the Royal Society of Tropical Medicine and Hygiene*, 73(5) :483 – 497.

Compeau, P. E. C., Pevzner, P. A., and Tesler, G. (2011). How to apply de Bruijn graphs to genome assembly. *Nature Biotechnology*, 29(11) :987–991.

Csurös, M. (2010). Count : Evolutionary analysis of phylogenetic profiles with parsimony and likelihood. *Bioinformatics*, 26(15) :1910–1912.

Csűrös, M. (2013). How to infer ancestral genome features by parsimony : dynamic programming over an evolutionary tree. *Computational Biology (Models and Algorithms for Genome Evolution)*, 19 :29–45.

Cunningham, F., Amode, M. R., Barrell, D., Beal, K., Billis, K., Brent, S., Carvalho-Silva, D., Clapham, P., Coates, G., Fitzgerald, S., Gil, L., Girón, C. G., Gordon, L., Hourlier, T., Hunt, S. E., Janacek, S. H., Johnson, N., Juettemann, T., Kähäri, A. K., Keenan, S., Martin, F. J., Maurel, T., McLaren, W., Murphy, D. N., Nag, R., Overduin, B., Parker, A., Patricio, M., Perry, E., Pignatelli, M., Riat, H. S., Sheppard, D., Taylor, K., Thormann, A., Vullo, A., Wilder, S. P., Zadissa, A., Aken, B. L., Birney, E., Harrow, J., Kinsella, R., Muffato, M., Ruffier, M., Searle, S. M. J., Spudich, G., Trevanion, S. J., Yates, A., Zerbino, D. R., and Flicek, P. (2015). Ensembl 2015. *Nucleic Acids Research*, 43(D1) :D662–D669.

Dahm, R. (2005). Friedrich Miescher and the discovery of DNA. *Developmental Biology*, 278(2) :274–288.

Dahm, R. (2008). Discovering DNA : Friedrich Miescher and the early years of nucleic acid research.

Dayrat, B. (2003). The Roots of Phylogeny : How Did Haeckel Build His Trees ? *Systematic Biology*, 52(4) :515–527.

De Bie, T., Cristianini, N., Demuth, J. P., and Hahn, M. W. (2006). CAFE : A computational tool for the study of gene family evolution. *Bioinformatics*, 22(10) :1269–1271.

Delcher, A. L., Kasif, S., Fleischmann, R. D., Peterson, J., White, O., and Salzberg, S. L. (1999). Alignment of whole genomes. *Nucleic Acids Research*, 27(11) :2369–2376.

della Torre, A., Merzagora, L., Powell, J. R., and Coluzzi, M. (1997). Selective introgression of paracentric inversions between two sibling species of the anopheles gambiae complex. *Genetics*, 146(1) :239–244.

Dias, Z., Dias, U., and Setubal, J. C. (2012). SIS : a program to generate draft genome sequence scaffolds for prokaryotes. *BMC Bioinformatics*, 13 :96–107.

Dobzhansky, T. and Sturtevant, A. H. (1938). Inversions in the chromosomes of Drosophila pseudoobscura. *Genetics*, 23 :28–64.

Dousse, A., Junier, T., and Zdobnov, E. M. (2016). CEGA-a catalog of conserved elements from genomic alignments. *Nucleic Acids Research*, 44(D1) :D96–D100.

Doyon, J.-P. (2010). *Algorithmes pour la réconciliation d'un arbre de gènes avec un arbre d'espèces*. PhD thesis, Université de Montréal - Faculté des arts et des sciences.

Doyon, J.-P., Scornavacca, C., Gorbunov, K. Y., Szöllősi, G. J., Ranwez, V., and Berry, V. (2010). An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. *Comparative Genomics - Lecture Notes in Computer Science*, 6398.

Duchemin, W. (2017). *Phylogénie des dépendances et dépendances des phylogénies dans les gènes et les génomes*. PhD thesis, Université Claude Bernard Lyon 1.

Duchemin, W., Anselmetti, Y., Patterson, M., Ponty, Y., Berard, S., Chauve, C., Scornavacca, C., Daubin, V., and Tannier, E. (2017). DeCoSTAR : Reconstructing the ancestral organization of genes or genomes using reconciled phylogenies. *Genome Biology and Evolution*.

Edgar, R. C. (2004). MUSCLE : Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5) :1792–1797.

Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., Bibillo, A., Bjornson, K., Chaudhuri, B., Christians, F., Cicero, R., Clark, S., Dalal, R., DeWinter, A., Dixon, J., Foquet, M., Gaertner, A., Hardenbol, P., Heiner, C., Hester, K., Holden, D., Kearns, G., Kong, X., Kuse, R., Lacroix, Y., Lin, S., Lundquist, P., Ma, C., Marks, P., Maxham, M., Murphy, D., Park, I., Pham, T., Phillips, M., Roy, J., Sebra, R., Shen, G., Sorenson, J., Tomaney, A., Travers, K., Trulson, M., Vieceli, J., Wegener, J., Wu, D., Yang, A., Zaccarin, D., Zhao, P., Zhong, F., Korlach, J., and Turner, S. (2009). Real-Time DNA Sequencing from Single Polymerase Molecules. *Science*, 323(5910) :133–138.

Ekblom, R. and Wolf, J. B. W. (2014). A field guide to whole-genome sequencing, assembly and annotation.

Feijão, P. and Meidanis, J. (2011). SCJ : A breakpoint-like distance that simplifies several rearrangement problems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(5) :1318–1329.

Fertin, G., Labarre, A., Rusu, I., Tannier, E., and Vialette, S. (2009). *Combinatorics of genome rearrangements*. MIT Press, mit press edition.

Fitch, W. M. (1971). Toward defining the course of evolution : minimum change for a specific tree topology. *Systematic Zoology*, 20(4) :406–416.

Fontaine, M. C., Pease, J. B., Steele, A., Waterhouse, R. M., Neafsey, D. E., Sharakhov, I. V., Jiang, X., Hall, A. B., Catteruccia, F., Kakani, E., Mitchell, S. N., Wu, Y.-C., Smith, H. A., Love, R. R., Lawniczak, M. K., Slotman, M. A., Emrich, S. J., Hahn, M. W., and Besansky, N. J. (2015). Extensive introgression in a malaria vector species complex revealed by phylogenomics. *Science*, 347(6217) :1258524–1 – 1258524–6.

Gagnon, Y., Blanchette, M., and El-Mabrouk, N. (2012). A flexible ancestral genome reconstruction method based on gapped adjacencies. *BMC bioinformatics*, 13 Suppl 1(Suppl 19) :S4.

Garcia, B. A., Caccone, A., Mathiopoulos, K. D., and Powell, J. R. (1996). Inversion monophyly in African anopheline malaria vectors. *Genetics*, 143(3) :1313–1320.

Gavranović, H., Chauve, C., Salse, J., and Tannier, E. (2011). Mapping ancestral genomes with massive gene loss : A matrix sandwich problem. *Bioinformatics*, 27(13) :257–265.

Ghiurcuta, C. G. and Moret, B. M. E. (2014). Evaluating synteny for improved comparative studies. *Bioinformatics*, 30(12) :9–18.

Gilbert, W. and Maxam, A. (1973). The nucleotide sequence of the lac operator. *Proceedings of the National Academy of Sciences of the United States of America*, 70(12) :3581–3584.

Giraldo-Calderón, G. I., Emrich, S. J., MacCallum, R. M., Maslen, G., Emrich, S., Collins, F., Dialynas, E., Topalis, P., Ho, N., Gesing, S., Madey, G., Collins, F. H., Lawson, D., Kersey, P., Allen, J., Christensen, M., Hughes, D., Koscielny, G., Langridge, N., Gallego, E. L., Megy, K., Wilson, D., Gelbart, B., Emmert, D., Russo, S., Zhou, P., Christophides, G., Brockman, A., Kirmitzoglou, I., MacCallum, B., Tiirikka, T., Louis, K., Dritsou, V., Mitraka, E., Werner-Washburn, M., Baker, P., Platero, H., Aguilar, A., Bogol, S., Campbell, D., Carmichael, R., Cieslak, D., Davis, G., Konopinski, N., Nabrzyski, J., Reinking, C., Sheehan, A., Szakonyi, S., and Wieck, R. (2015). VectorBase : An updated Bioinformatics Resource for invertebrate vectors and other organisms related with human diseases. *Nucleic Acids Research*, 43(D1) :D707–D713.

Gnerre, S., Maccallum, I., Przybylski, D., Ribeiro, F. J., Burton, J. N., Walker, B. J., Sharpe, T., Hall, G., Shea, T. P., Sykes, S., Berlin, A. M., Aird, D., Costello, M., Daza, R., Williams, L., Nicol, R., Gnirke, A., Nusbaum, C., Lander,

E. S., and Jaffe, D. B. (2011). High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences of the United States of America*, 108(4) :1513–1518.

Goodman, M., Czelusniak, J., Moore, G. W., Romero-Herrera, A. E., and Matsuda, G. (1979). Fitting the gene lineage into its species lineage, a parsimony strategy illustrated dy cladograms constructed from globin sequences. *Systematic Zoology*, 28(2) :132–163.

Goodwin, S., McPherson, J. D., and McCombie, W. R. (2016). Coming of age : ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6) :333–351.

Grabherr, M. G., Haas, B. J., Yassour, M., Levin, J. Z., Thompson, D. A., Amit, I., Adiconis, X., Fan, L., Raychowdhury, R., Zeng, Q., Chen, Z., Mauceli, E., Hacohen, N., Gnirke, A., Rhind, N., di Palma, F., Birren, B. W., Nusbaum, C., Lindblad-Toh, K., Friedman, N., and Regev, A. (2011). Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature biotechnology*, 29(7) :644–52.

Green, R. E., Krause, J., Briggs, A. W., Maricic, T., Stenzel, U., Kircher, M., Patterson, N., Li, H., Zhai, W., Fritz, M. H.-Y., Hansen, N. F., Durand, E. Y., Malaspinas, A.-S., Jensen, J. D., Marques-Bonet, T., Alkan, C., Prüfer, K., Meyer, M., Burbano, H. a., Good, J. M., Schultz, R., Aximu-Petri, A., Butthof, A., Höber, B., Höffner, B., Siegemund, M., Weihmann, A., Nusbaum, C., Lander, E. S., Russ, C., Novod, N., Affourtit, J. P., Egholm, M., Verna, C., Rudan, P., Brajkovic, D., Kucan, Z., Gusic, I., Doronichev, V. B., Golovanova, L. V., Lalueza-Fox, C., de la Rasilla, M., Fortea, J., Rosas, A., Schmitz, R. W., Johnson, P. L. F., Eichler, E. E., Falush, D., Birney, E., Mullikin, J. C., Slatkin, M., Nielsen, R., Kelso, J., Lachmann, M., Reich, D., and Pääbo, S. (2010). A draft sequence of the Neandertal genome. *Science*, 328(5979) :710–22.

Guéguen, L., Gaillard, S., Boussau, B., Gouy, M., Groussin, M., Rochette, N. C., Bigot, T., Fournier, D., Pouyet, F., Cahais, V., Bernard, A., Scornavacca, C., Nabholz, B., Haudry, A., Dachary, L., Galtier, N., Belkhir, K., and Dutheil, J. Y. (2013). Bio++ : efficient extensible libraries and tools for computational molecular evolution. *Molecular Biology and Evolution*, 30(8) :1745–50.

Guigó, R., Muchnik, I., and Smith, T. F. (1996). Reconstruction of ancient molecular phylogeny. *Molecular phylogenetics and evolution*, 6(2) :189–213.

Guindon, S., Dufayard, J. J. F., Lefort, V., Anisimova, M., Hordijk, W., and Gascuel, O. (2010). New algorithms and methods to estimate maximum-likelihood phylogenies : Assessing the performance of PhyML 3.0. *Systematic Biology*, 59(3) :307–321.

Hahn, M. W. (2007). Bias in phylogenetic tree reconciliation methods : implications for vertebrate genome evolution. *Genome Biology*, 8(7) :R141.

Hallett, M. T. and Lagergren, J. (2001). Efficient algorithms for lateral gene transfer problems. *RECOMB 2001 : Proceedings of the Fifth International Conference on Computational Biology*, pages 149–156.

Harrison, R. G. and Larson, E. L. (2014). Hybridization, introgression, and the nature of species boundaries. *Journal of Heredity*, 105(S1) :795–809.

Heather, J. M. and Chain, B. (2016). The sequence of sequencers : The history of sequencing DNA.

Hershey, A. D. and Chase, M. (1952). Independent functions of viral protein and nucleic acid in growth of bacteriophage. *The Journal of General Physiology*, 36(1) :39–56.

Hofreiter, M., Serre, D., Poinar, H. N., Kuch, M., and Pääbo, S. (2001). Ancient DNA. *Nature Reviews Genetics*, 2(5) :353–9.

Holt, R., Subramanian, G., Halpern, A., Sutton, G., Charlab, R., Nusskern, D., Wincker, P., Clark, A., Ribeiro, J., Wides, R., Salzberg, S., Loftus, B., Yandell, M., Majoros, W., Rusch, D., Lai, Z., Kraft, C., Abril, J., Anthouard, V., Arensburger, P., Atkinson, P., Baden, H., de Berardinis, V., Baldwin, D., Benes, V., Biedler, J., Blass, C., Bolanos, R., Boscus, D., Barnstead, M., Cai, S., Center, A., Chaturverdi, K., Christophides, G., Chrystal, M., Clamp, M., Cravchik, A., Curwen, V., Dana, A., Delcher, A., Dew, I., Evans, C., Flanigan, M., Grundschober-Freimoser, A., Friedli, L., Gu, Z., Guan, P., Guigo, R., Hillenmeyer, M., Hladun, S., Hogan, J., Hong, Y., Hoover, J., Jaillon, O., Ke, Z., Kodira, C., Kokoza, E., Koutsos, A., Letunic, I., Levitsky, A., Liang, Y., Lin, J.-J., Lobo, N., Lopez, J., Malek, J., McIntosh, T., Meister, S., Miller, J., Mobarry, C., Mongin, E., Murphy, S., O'Brochta, D., Pfannkoch, C., Qi, R., Regier, M., Remington, K., Shao, H., Sharakhova, M., Sitter, C., Shetty, J., Smith, T., Strong, R., Sun, J., Thomasova, D., Ton, L., Topalis, P., Tu, Z.,

Unger, M., Walenz, B., Wang, A., Wang, J., Wang, M., Wang, X., Woodford, K., Wortman, J., Wu, M., Yao, A., Zdobnov, E., Zhang, H., Zhao, Q., Zhao, S., Zhu, S., Zhimulev, I., Coluzzi, M., della Torre, A., Roth, C., Louis, C., Kalush, F., Mural, R., Myers, E., Adams, M., Smith, H., Broder, S., Gardner, M., Fraser, C., Birney, E., Bork, P., Brey, P., Venter, J., Weissenbach, J., Kafatos, F., Collins, F., and Hoffman, S. (2002). The genome sequence of the malaria mosquito Anopheles gambiae. *Science*, 298(5591) :129–149.

Hu, F., Lin, Y., and Tang, J. (2014a). MLGO : phylogeny reconstruction and ancestral inference from gene-order data. *BMC Bioinformatics*, 15(1) :354.

Hu, F., Zhou, J., Zhou, L., and Tang, J. (2014b). Probabilistic Reconstruction of Ancestral Gene Orders with Insertions and Deletions. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(4) :667–672.

Human Genome Sequencing Consortium, I. (2004). Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011) :931–945.

Hunt, M., Newbold, C., Berriman, M., and Otto, T. D. (2014). A comprehensive evaluation of assembly scaffolding tools. *Genome Biology*, 15(3) :R42.

Husemann, P. and Stoye, J. (2009). r2cat : Synteny plots and comparative assembly. *Bioinformatics*, 26(4) :570–571.

Husemann, P. and Stoye, J. (2010). Phylogenetic comparative assembly. *Algorithms for Molecular Biology*, 5(1) :3–14.

Huson, D. H., Reinert, K., and Myers, E. W. (2002). The greedy path-merging algorithm for contig scaffolding. *Journal of the ACM*, 49(5) :603–615.

Hutchison, C. A. (2007). DNA sequencing : Bench to bedside and beyond. *Nucleic Acids Research*, 35(18) :6227–6237.

Idury, R. M. and Waterman, M. S. (1995). A new algorithm for DNA sequence assembly. *Journal of computational biology : a journal of computational molecular cell biology*, 2(2) :291–306.

Jacox, E., Chauve, C., Szöllősi, G. J., Ponty, Y., and Scornavacca, C. (2016). EcceTERA : Comprehensive gene tree-species tree reconciliation using parsimony. *Bioinformatics*, 32(13) :2056–2058.

Jones, B. R., Rajaraman, A., Tannier, E., and Chauve, C. (2012). ANGES : Reconstructing ANcestral GEnomeS maps. *Bioinformatics*, 28(18) :2388–2390.

Jones, C. H. (1994). Generalized Hockey Stick Identities and N-Dimensional Block Walking. *Identity*, 93560(October) :280–288.

Kasprzyk, A. (2011). BioMart : Driving a paradigm change in biological data management. *Database*, 2011 :bar049.

Kent, W. J. (2002). BLAT—The BLAST-Like Allignment Tool. *Genome Research*, 12(4) :656–664.

Kim, J., Farré, M., Auvil, L., Capitanu, B., Larkin, D. M., Ma, J., and Lewin, H. A. (2017). Reconstruction and evolutionary history of eutherian chromosomes. *Proceedings of the National Academy of Sciences*, 114(27) :E5379–E5388.

Kim, J., Larkin, D. M., Cai, Q., Asan, Zhang, Y., Ge, R.-L., Auvil, L., Capitanu, B., Zhang, G., Lewin, H. A., and Ma, J. (2013). Reference-assisted chromosome assembly. *Proceedings of the National Academy of Sciences (PNAS)*, 110(5) :1785–90.

Kolmogorov, M., Armstrong, J., Raney, B. J., Streeter, I., Dunn, M., Yang, F., Odom, D., Flicek, P., Keane, T., Thybert, D., Paten, B., and Pham, S. (2016). Chromosome assembly of large and complex genomes using multiple references. *bioRxiv*.

Kolmogorov, M., Raney, B., Paten, B., and Pham, S. (2014). Ragout - A reference-assisted assembly tool for bacterial genomes. *Bioinformatics*, 30(12) :302–309.

Koren, S. and Phillippy, A. M. (2015). One chromosome, one contig : Complete microbial genomes from long-read sequencing and assembly. *Current Opinion in Microbiology*, 23 :110–120.

Koren, S., Walenz, B. P., Berlin, K., Miller, J. R., Bergman, N. H., and Phillippy, A. M. (2017). Canu : Scalable and accurate long-read assembly via adaptive κ-mer weighting and repeat separation. *Genome Research*, 27(5) :722–736.

Kurtz, S., Phillippy, A. M., Delcher, A. L., Smoot, M., Shumway, M., Antonescu, C., and Salzberg, S. L. (2004). Versatile and open software for comparing large genomes. *Genome Biology*, 5(2) :R12.

Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4) :357–359.

Levene, M. J., Korlach, J., Turner, S. W., Foquet, M., G.Craighead, H., and Webb, W. W. (2003). Zero-Mode Waveguides for Single-Molecule Analysis at High Concentrations. *Science*, 299(5607) :682–686.

Li, C.-L., Chen, K.-T., and Lu, C. L. (2013). Assembling contigs in draft genomes using reversals and block-interchanges. *BMC bioinformatics*, 14 Suppl 5(Suppl 5) :S9.

Li, G., Davis, B. W., Eizirik, E., and Murphy, W. J. (2016). Phylogenomic evidence for ancient hybridization in the genomes of living cats (Felidae). *Genome Research*, 26(1) :1–11.

Li, Z., Chen, Y., Mu, D., Yuan, J., Shi, Y., Zhang, H., Gan, J., Li, N., Hu, X., Liu, B., Yang, B., and Fan, W. (2012). Comparison of the two major classes of assembly algorithms : Overlap-layout-consensus and de-bruijn-graph. *Briefings in Functional Genomics*, 11(1) :25–37.

Lin, Y., Nurk, S., and Pevzner, P. A. (2014). What is the difference between the breakpoint graph and the de Bruijn graph ? *BMC Genomics*, 15(Suppl 6) :S6.

Lindblad-Toh, K., Garber, M., Zuk, O., Lin, M. F., Parker, B. J., Washietl, S., Kheradpour, P., Ernst, J., Jordan, G., Mauceli, E., Ward, L. D., Lowe, C. B., Holloway, A. K., Clamp, M., Gnerre, S., Alföldi, J., Beal, K., Chang, J., Clawson, H., Cuff, J., Di Palma, F., Fitzgerald, S., Flicek, P., Guttman, M., Hubisz, M. J., Jaffe, D. B., Jungreis, I., Kent, W. J., Kostka, D., Lara, M., Martins, A. L., Massingham, T., Moltke, I., Raney, B. J., Rasmussen, M. D., Robinson, J., Stark, A., Vilella, A. J., Wen, J., Xie, X., Zody, M. C., Baldwin, J., Bloom, T., Whye Chin, C., Heiman, D., Nicol, R., Nusbaum, C., Young, S., Wilkinson, J., Worley, K. C., Kovar, C. L., Muzny, D. M., Gibbs, R. A., Cree, A., Dihn, H. H., Fowler, G., Jhangiani, S., Joshi, V., Lee, S., Lewis, L. R., Nazareth, L. V., Okwuonu, G., Santibanez, J., Warren, W. C., Mardis, E. R., Weinstock, G. M., Wilson, R. K., Delehaunty, K., Dooling, D., Fronik, C., Fulton, L., Fulton, B., Graves, T., Minx, P., Sodergren, E., Birney, E., Margulies, E. H., Herrero, J., Green, E. D., Haussler, D., Siepel, A., Goldman, N., Pollard, K. S., Pedersen, J. S., Lander, E. S., and Kellis, M. (2011). A high-resolution map of human evolutionary constraint using 29 mammals. *Nature*, 478(7370) :476–482.

Lu, C. L., Chen, K.-t., Huang, S.-y., and Chiu, H.-t. (2014). CAR : contig assembly of prokaryotic draft genomes using rearrangements. *BMC bioinformatics*, 15 :381–390.

Luhmann, N. (2017). *PhyloPhylogenetic Assembly of Paleogenomes Integrating Ancient DNA Datagenetic Assembly of Paleogenomes Integrating Ancient DNA Data*. PhD thesis, Bielefeld University.

Luhmann, N., Thévenin, A., Ouangraoua, A., Wittler, R., and Chauve, C. (2016). The SCJ small parsimony problem for weighted gene adjacencies. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9683, pages 200–210.

Lukhtanov, V. A., Dincă, V., Talavera, G., and Vila, R. (2011). Unprecedented within-species chromosome number cline in the Wood White butterfly Leptidea sinapis and its significance for karyotype evolution and speciation. *BMC Evolutionary Biology*, 11(1) :109.

Ma, B., Li, M., and Zhang, L. (2000). From gene trees to species trees. *SIAM Journal on Computing*, 30(3) :729–752.

Ma, J. (2010). A probabilistic framework for inferring ancestral genomic orders. In *Proceedings - 2010 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2010*, pages 179–184.

Ma, J., Ratan, A., Raney, B. J., Suh, B. B., Miller, W., and Haussler, D. (2008). The infinite sites model of genome evolution. *Proceedings of the National Academy of Sciences*, 105(38) :14254–14261.

Ma, J., Zhang, L., Suh, B. B., Raney, B. J., Burhans, R. C., Kent, W. J., Blanchette, M., Haussler, D., and Miller, W. (2006). Reconstructing contiguous regions of an ancestral genome. *Genome Research*, 16(12) :1557–1565.

Maddison, W. P. (1997). Gene trees in species trees. *Systematic Biology*, 46(3) :523–536.

Magadum, S., Banerjee, U., Murugan, P., Gangapur, D., and Ravikesavan, R. (2013). Gene duplication as a major force in evolution. *Journal of Genetics*, 92(1) :155–161.

Mandric, I. and Zelikovsky, A. (2015). ScaffMatch : Scaffolding Algorithm Based on Maximum Weight Matching. *Bioinformatics*, 2 :1–7.

Marinotti, O., Cerqueira, G. C., De Almeida, L. G. P., Ferro, M. I. T., Da Silva Loreto, E. L., Zaha, A., Teixeira, S. M. R., Wespiser, A. R., E Silva, A. A., Schlindwein, A. D., Pacheco, A. C. L., Da Costa Da Silva, A. L., Graveley, B. R., Walenz, B. P., De Araujo Lima, B., Ribeiro, C. A. G., Nunes-Silva, C. G., De Carvalho, C. R., De Almeida Soares, C. M., De Menezes, C. B. A., Matiolli, C., Caffrey, D., Araújo, D. A. M., De Oliveira, D. M., Golenbock, D., Grisard, E. C., Fantinatti-Garboggini, F., De Carvalho, F. M. F. M., Barcellos, F. G., Prosdocimi, F., May, G., De Azevedo Junior, G. M., Guimarães, G. M., Goldman, G. H., Padilha, I. Q. M., Da Silva Batista, J., Ferro, J. A., Ribeiro, J. M. C., Fietto, J. L. R., Dabbas, K. M., Cerdeira, L., Agnez-Lima, L. F., Brocchi, M., De Carvalho, M. O., De Melo Teixeira, M., De Mascena Diniz Maia, M., Goldman, M. H. S., Schneider, M. P. C., Felipe, M. S. S., Hungria, M., Nicolás, M. F., Pereira, M., Montes, M. A. M. A., Cantão, M. E. M. E., Vincentz, M., Rafael, M. S., Silverman, N., Stoco, P. H. P. H., Souza, R. C., Vicentini, R., Gazzinelli, R. T., De Oliveira Neves, R., Silva, R., Astolfi-Filho, S., Maciel, T. E. F., Ürményi, T. P., Tadei, W. P., Camargo, E. P., and De Vasconcelos, A. T. R. (2013). The Genome of Anopheles darlingi, the main neotropical malaria vector. *Nucleic Acids Research*, 41(15) :7387–7400.

Maňuch, J. and Patterson, M. (2011). The Complexity of the Gapped Consecutive-Ones Property Problem for Matrices of Bounded Maximum Degree. *Journal of Computational Biology*, 18(9) :1243–1253.

Maňuch, J., Patterson, M., and Chauve, C. (2012a). Hardness results on the gapped consecutive-ones property problem. *Discrete Applied Mathematics*, 160(18) :2760–2768.

Maňuch, J., Patterson, M., Wittler, R., Chauve, C., and Tannier, E. (2012b). Linearization of ancestral multichromosomal genomes. *BMC bioinformatics*, 13 Suppl 1(Suppl 19) :S11.

Medvedev, P., Pham, S. O. N., Chaisson, M., Tesler, G., and Pevzner, P. (2011). Paired de Bruijn graphs : A novel approach for Incorporating Mate Pair Information into Genome Assemblers. *Journal of Computational Biology*, 18(11) :1625–1634.

Meidanis, J., Porto, O., and Telles, G. P. (1998). On the consecutive ones property. *Discrete Applied Mathematics*, 88(1-3) :325–354.

Mendel, G. (1865). Versuche über Plflanzen-hybriden. In *Verhandlungen des naturforschenden Ver-eines in Brünn*, pages 3–47.

Metzker, M. L. (2010). Sequencing technologies—the next generation. *Nature Reviews Genetics*, 11(1) :31–46.

Miller, J. R., Koren, S., and Sutton, G. (2010). Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6) :315–327.

Minkin, I., Patel, A., Kolmogorov, M., Vyahhi, N., and Pham, S. (2013). Sibelia : A scalable and comprehensive synteny block generation tool for closely related microbial genomes. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8126 LNBI(1) :215–229.

Moreno, M., Marinotti, O., Krzywinski, J., Tadei, W. P., James, A. A., Achee, N. L., and Conn, J. E. (2010). Complete mtDNA genomes of Anopheles darlingi and an approach to anopheline divergence time. *Malaria Journal*, 9 :127.

Muffato, M. and Roest Crollius, H. (2008). Paleogenomics in vertebrates, or the recovery of lost genomes from the mist of time.

Mukherjee, S., Stamatis, D., Bertsch, J., Ovchinnikova, G., Verezemska, O., Isbandi, M., Thomas, A. D., Ali, R., Sharma, K., Kyrpides, N. C., and Reddy, T. B. (2017). Genomes OnLine Database (GOLD) v.6 : Data updates and feature enhancements. *Nucleic Acids Research*, 45(D1) :D446–D456.

Muñoz, A., Zheng, C., Zhu, Q., Albert, V. A., Rounsley, S., and Sankoff, D. (2010). Scaffold filling, contig fusion and comparative gene order inference. *BMC bioinformatics*, 11 :304–318.

Murat, F., Xu, J. H., Tannier, E., Abrouk, M., Guilhot, N., Pont, C., Messing, J., and Salse, J. (2010). Ancestral grass karyotype reconstruction unravels new mechanisms of genome shuffling as a source of plant evolution. *Genome Research*, 20(11) :1545–1557.

Murat, F., Zhang, R., Guizard, S., Gavranović, H., Flores, R., Steinbach, D., Quesneville, H., Tannier, E., and Salse, J. (2015). Karyotype and gene order evolution from reconstructed extinct ancestors highlight contrasts in genome plasticity of modern rosid crops. *Genome Biology and Evolution*, 7(3) :735–749.

Murphy, W. J., Pringle, T. H., Crider, T. a., Springer, M. S., and Miller, W. (2007). Using genomic data to unravel the root of the placental mammal

phylogeny Using genomic data to unravel the root of the placental mammal phylogeny. *Genome Research*, 17(979) :413–421.

Nagarajan, N. and Pop, M. (2013). Sequence assembly demystified. *Nature Reviews Genetics*, 14(3) :157–167.

Neafsey, D. E., Christophides, G. K., Collins, F. H., Emrich, S. J., Fontaine, M. C., Gelbart, W., Hahn, M. W., Howell, P. I., Kafatos, F. C., Lawson, D., Muskavitch, M. a. T., Waterhouse, R. M., Williams, L. J., Besansky, N. J., London, I. C., Campus, S. K., and Sw, L. (2013). The evolution of the Anopheles 16 genomes project. *G3 Genes | Genomes | Genetics*, 3(7) :1191–4.

Neafsey, D. E., Waterhouse, R. M., Abai, M. R., Aganezov, S. S., Alekseyev, M. A., Allen, J. E., Amon, J., Arcà, B., Arensburger, P., Artemov, G., Assour, L. A., Basseri, H., Berlin, A., Birren, B. W., Blandin, S. A., Brockman, A. I., Burkot, T. R., Burt, A., Chan, C. S., Chauve, C., Chiu, J. C., Christensen, M., Costantini, C., Davidson, V. L. M., Deligianni, E., Dottorini, T., Dritsou, V., Gabriel, S. B., Guelbeogo, W. M., Hall, A. B., Han, M. V., Hlaing, T., Hughes, D. S. T., Jenkins, A. M., Jiang, X., Jungreis, I., Kakani, E. G., Kamali, M., Kemppainen, P., Kennedy, R. C., Kirmitzoglou, I. K., Koekemoer, L. L., Laban, N., Langridge, N., Lawniczak, M. K. N., Lirakis, M., Lobo, N. F., Lowy, E., MacCallum, R. M., Mao, C., Maslen, G., Mbogo, C., McCarthy, J., Michel, K., Mitchell, S. N., Moore, W., Murphy, K. A., Naumenko, A. N., Nolan, T., Novoa, E. M., O'Loughlin, S., Oringanje, C., Oshaghi, M. A., Pakpour, N., Papathanos, P. A., Peery, A. N., Povelones, M., Prakash, A., Price, D. P., Rajaraman, A., Reimer, L. J., Rinker, D. C., Rokas, A., Russell, T. L., Sagnon, N., Sharakhova, M. V., Shea, T., Simão, F. A., Simard, F., Slotman, M. A., Somboon, P., Stegniy, V., Struchiner, C. J., Thomas, G. W. C., Tojo, M., Topalis, P., Tubio, J. M. C., Unger, M. F., Vontas, J., Walton, C., Wilding, C. S., Willis, J. H., Wu, Y.-C., Yan, G., Zdobnov, E. M., Zhou, X., Catteruccia, F., Christophides, G. K., Collins, F. H., Cornman, R. S., Crisanti, A., Donnelly, M. J., Emrich, S. J., Fontaine, M. C., Gelbart, W., Hahn, M. W., Hansen, I. A., Howell, P. I., Kafatos, F. C., Kellis, M., Lawson, D., Louis, C., Luckhart, S., Muskavitch, M. A. T., Ribeiro, J. M., Riehle, M. A., Sharakhov, I. V., Tu, Z., Zwiebel, L. J., and Besansky, N. J. (2015). Highly evolvable malaria vectors : The genomes of 16 Anopheles mosquitoes. *Science*, 347(6217) :1258522–1258528.

Noutahi, E., Semeria, M., Lafond, M., Seguin, J., Boussau, B., Guéguen, L., El-Mabrouk, N., and Tannier, E. (2016). Efficient gene tree correction guided by genome evolution. *PLoS ONE*, 11(8).

Orlando, L., Ginolhac, A., Zhang, G., Froese, D., Albrechtsen, A., Stiller, M., Schubert, M., Cappellini, E., Petersen, B., Moltke, I., Johnson, P. L. F., Fumagalli, M., Vilstrup, J. T., Raghavan, M., Korneliussen, T., Malaspinas, A.-S., Vogt, J., Szklarczyk, D., Kelstrup, C. D., Vinther, J., Dolocan, A., Stenderup, J., Velazquez, A. M. V., Cahill, J., Rasmussen, M., Wang, X., Min, J., Zazula, G. D., Seguin-Orlando, A., Mortensen, C., Magnussen, K., Thompson, J. F., Weinstock, J., Gregersen, K., Røed, K. H., Eisenmann, V., Rubin, C. J., Miller, D. C., Antczak, D. F., Bertelsen, M. F., Brunak, S., Al-Rasheid, K. a. S., Ryder, O., Andersson, L., Mundy, J., Krogh, A., Gilbert, M. T. P., Kjær, K., Sicheritz-Ponten, T., Jensen, L. J., Olsen, J. V., Hofreiter, M., Nielsen, R., Shapiro, B., Wang, J., and Willerslev, E. (2013). Recalibrating Equus evolution using the genome sequence of an early Middle Pleistocene horse. *Nature*, 499(7456) :74–8.

Ouangraoua, A., Tannier, E., and Chauve, C. (2011). Reconstructing the architecture of the ancestral amniote genome. *Bioinformatics*, 27(19) :2664–2671.

Paten, B., Earl, D., Nguyen, N., Diekhans, M., Zerbino, D., and Haussler, D. (2011). Cactus : Algorithms for genome multiple sequence alignment. *Genome Research*, 21(9) :1512–1528.

Paten, B., Zerbino, D. R., Hickey, G., and Haussler, D. (2014). A unifying model of genome evolution under parsimony. *BMC Bioinformatics*, 15(1) :206.

Patterson, M., Szöllősi, G. J., Daubin, V., and Tannier, E. (2013). Lateral gene transfer, rearrangement, reconciliation. *BMC Bioinformatics*, 14(Suppl 15) :S4.

Pauling, L. and Zuckerkandl, E. (1963). Chemical paleogenetics Molecular "Restoration studies" of exctinct forms of life.

Penel, S., Arigon, A.-M., Dufayard, J.-F., Sertier, A.-S., Daubin, V., Duret, L., Gouy, M., and Perrière, G. (2009). Databases of homologous gene families for comparative genomics. *BMC bioinformatics*, 10 Suppl 6 :S3.

Pombi, M., Caputo, B., Simard, F., Di Deco, M. a., Coluzzi, M., della Torre, A., Costantini, C., Besansky, N. J., and Petrarca, V. (2008). Chromosomal plasticity and evolutionary potential in the malaria vector Anopheles gambiae

sensu stricto : insights from three decades of rare paracentric inversions. *BMC evolutionary biology*, 8 :309.

Prüfer, K., Racimo, F., Patterson, N., Jay, F., Sankararaman, S., Sawyer, S., Heinze, A., Renaud, G., Sudmant, P. H., de Filippo, C., Li, H., Mallick, S., Dannemann, M., Fu, Q., Kircher, M., Kuhlwilm, M., Lachmann, M., Meyer, M., Ongyerth, M., Siebauer, M., Theunert, C., Tandon, A., Moorjani, P., Pickrell, J., Mullikin, J. C., Vohr, S. H., Green, R. E., Hellmann, I., Johnson, P. L. F., Blanche, H., Cann, H., Kitzman, J. O., Shendure, J., Eichler, E. E., Lein, E. S., Bakken, T. E., Golovanova, L. V., Doronichev, V. B., Shunkov, M. V., Derevianko, A. P., Viola, B., Slatkin, M., Reich, D., Kelso, J., and Pääbo, S. (2014). The complete genome sequence of a Neanderthal from the Altai Mountains. *Nature*, 505(7481) :43–9.

Rajaraman, A., Tannier, E., and Chauve, C. (2013). FPSAC : Fast phylogenetic scaffolding of ancient contigs. *Bioinformatics*, 29(23) :2987–2994.

Rajaraman, A., Zanetti, J., Manuch, J., and Chauve, C. (2016). Algorithms and complexity results for genome mapping problems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, X(c) :1–1.

Rens, W., Grützner, F., O'Brien, P. C. M., Fairclough, H., Graves, J. A. M., and Ferguson-Smith, M. A. (2004). Resolution and evolution of the duckbilled platypus karyotype with an X1Y1 X2Y2 X3Y3 X4Y4 X5Y5 male sex chromosome constitution. *Proceedings of the National Academy of Sciences of the United States of America*, 101(46) :16257–16261.

Rhoads, A. and Au, K. F. (2015). PacBio Sequencing and Its Applications.

Richter, D. C., Schuster, S. C., and Huson, D. H. (2007). OSLay : Optimal Syntenic layout of unfinished assemblies. *Bioinformatics*, 23(13) :1573–1579.

Rissman, A. I., Mau, B., Biehl, B. S., Darling, A. E., Glasner, J. D., and Perna, N. T. (2009). Reordering contigs of draft genomes using the Mauve Aligner. *Bioinformatics*, 25(16) :2071–2073.

Robicsek, A., Jacoby, G. A., and Hooper, D. C. (2006). The worldwide emergence of plasmid-mediated quinolone resistance. *The Lancet. Infectious diseases*, 6(10) :629–40.

Romiguier, J. (2012). *Phylogénomique et stratégies d'histoire de vie des mammifères placentaires : apports de la théorie de la conversion génique biaisée*. PhD thesis, Université de Montpellier 2.

Romiguier, J., Ranwez, V., Delsuc, F., Galtier, N., and Douzery, E. J. P. (2013). Less is more in mammalian phylogenomics : AT-rich genes minimize tree conflicts and unravel the root of placental mammals. *Molecular Biology and Evolution*, 30(9) :2134–2144.

Ronen, R., Boucher, C., Chitsaz, H., and Pevzner, P. (2012). SEQuel : Improving the accuracy of genome assemblies. *Bioinformatics*, 28(12) :i188–96.

Rutledge, G. G., Böhme, U., Sanders, M., Reid, A. J., Cotton, J. A., Maiga-Ascofare, O., Djimdé, A. A., Apinjoh, T. O., Amenga-Etego, Lucas Manske, M., Barnwell, J. W., Renaud, François Ollomo, B., Prugnolle, F., Anstey, N. M., Auburn, S., Price, R. N., McCarthy, J. S., Kwiatkowski, D. P., Newbold, Chris I. Berriman, M., and Thomas D. Otto, Aude Gilabert, Thomas Crellen, Ulrike Böhme, Céline Arnathau, Mandy Sanders, Samuel Oyola, Alain Prince Okouga, Larson Boundenga, Eric Wuillaume, Barthélémy Ngoubangoye, Nancy Diamella Moukodoum, Christophe Paupy, Patrick Durand, Virginie, M. B. &. F. P. (2017). Plasmodium malariae and P. ovale genomes provide insights into malaria parasite evolution. *Nature*, in press.

Sahlin, K., Chikhi, R., and Arvestad, L. (2016). Assembly scaffolding with PE-contaminated mate-pair libraries. *Bioinformatics*, 32(March) :btw064.

Sahlin, K., Street, N., Lundeberg, J., and Arvestad, L. (2012). Improved gap size estimation for scaffolding algorithms. *Bioinformatics*, 28(17) :2215–2222.

Sahlin, K., Vezzi, F., Nystedt, B., Lundeberg, J., and Arvestad, L. (2014). BESST - Efficient scaffolding of large fragmented assemblies. *BMC Bioinformatics*, 15(1) :281.

Salikhov, K., Sacomoto, G., and Kucherov, G. (2014). Using cascading bloom filters to improve the memory usage for de Brujin graphs. *Algorithms for Molecular Biology*, 9(1) :2.

Sanger, F. and Coulson, A. R. (1975). A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *Journal of Molecular Biology*, 94(3) :441–448.

Sanger, F., Nicklen, S., and Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, 74(12) :5463–7.

Sankoff, D. (1975). Minimal mutation trees of sequences. *SIAM Journal on Applied Mathematics*, 28(1) :35–42.

Sankoff, D. and Nadeau, J. H. (2003). Chromosome rearrangements in evolution : From gene order to genome sequence and back. *Proceedings of the National Academy of Sciences of the United States of America*, 100(20) :11188–9.

Sankoff, D. and Rousseau, P. (1975). Locating the vertices of a steiner tree in an arbitrary metric space. *Mathematical Programming*, 9(1) :240–246.

Schadt, E. E., Turner, S., and Kasarskis, A. (2010). A window into third-generation sequencing. *Human molecular genetics*, 19(R2) :R227–40.

Schneeberger, K., Ossowski, S., Ott, F., Klein, J. D., Wang, X., Lanz, C., Smith, L. M., Cao, J., Fitz, J., Warthmann, N., Henz, S. R., Huson, D. H., and Weigel, D. (2011). Reference-guided assembly of four diverse Arabidopsis thaliana genomes. *Proceedings of the National Academy of Sciences*, 108(25) :10249–10254.

Schubert, M., Jónsson, H., Chang, D., Der Sarkissian, C., Ermini, L., Ginolhac, A., Albrechtsen, A., Dupanloup, I., Foucal, A., Petersen, B., Fumagalli, M., Raghavan, M., Seguin-Orlando, A., Korneliussen, T. S., Velazquez, A. M. V., Stenderup, J., Hoover, C. A., Rubin, C.-J., Alfarhan, A. H., Alquraishi, S. A., Al-Rasheid, K. A. S., MacHugh, D. E., Kalbfleisch, T., MacLeod, J. N., Rubin, E. M., Sicheritz-Ponten, T., Andersson, L., Hofreiter, M., Marques-Bonet, T., Gilbert, M. T. P., Nielsen, R., Excoffier, L., Willerslev, E., Shapiro, B., and Orlando, L. (2014). Prehistoric genomes reveal the genetic foundation and cost of horse domestication. *Proceedings of the National Academy of Sciences of the United States of America*, 111(52) :E5661–9.

Scornavacca, C. and Galtier, N. (2016). Incomplete Lineage Sorting in Mammalian Phylogenomics. *Systematic Biology*, 1542(9) :33–36.

Scornavacca, C., Jacox, E., and Szöllősi, G. J. (2015). Joint Amalgamation of Most Parsimonious Reconciled Gene Trees. *Bioinformatics*, 31(6) :841–848.

Semeria, M., Tannier, E., and Guéguen, L. (2015). Probabilistic modeling of the evolution of gene synteny within reconciled phylogenies. *BMC bioinformatics*, 16(Suppl 14) :1–11.

Shaik, S., Kumar, N., Lankapalli, A. K., Tiwari, S. K., Baddam, R., and Ahmed, N. (2016). Contig-Layout-Authenticator (CLA) : A combinatorial approach

to ordering and scaffolding of bacterial contigs for comparative genomics and molecular epidemiology. *PLoS ONE*, 11(6).

Sohn, J.-i. and Nam, J.-W. (2016). The present and future of de novo whole-genome assembly. *Briefings in Bioinformatics*, 19(January 2015) :bbw096.

Staden, R. (1979). A strategy of DNA sequencing employing computer programs. *Nucleic Acids Research*, 6(7) :2601–2610.

Stamatakis, A. (2014). RAxML version 8 : A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9) :1312–1313.

Stark, A., Lin, M. F., Kheradpour, P., Pedersen, J. S., Parts, L., Carlson, J. W., Crosby, M. A., Rasmussen, M. D., Roy, S., Deoras, A. N., Ruby, J. G., Brennecke, J., Curators, H. F., Project, B. D. G., Hodges, E., Hinrichs, A. S., Caspi, A., Paten, B., Park, S.-W., Han, M. V., Maeder, M. L., Polansky, B. J., Robson, B. E., Aerts, S., van Helden, J., Hassan, B., Gilbert, D. G., Eastman, D. A., Rice, M., Weir, M., Hahn, M. W., Park, Y., Dewey, C. N., Pachter, L., Kent, W. J., Haussler, D., Lai, E. C., Bartel, D. P., Hannon, G. J., Kaufman, T. C., Eisen, M. B., Clark, A. G., Smith, D., Celniker, S. E., Gelbart, W. M., Kellis, M., Crosby, M. A., Matthews, B. B., Schroeder, A. J., Sian Gramates, L., St Pierre, S. E., Roark, M., Wiley Jr, K. L., Kulathinal, R. J., Zhang, P., Myrick, K. V., Antone, J. V., Gelbart, W. M., Carlson, J. W., Yu, C., Park, S., Wan, K. H., and Celniker, S. E. (2007). Discovery of functional elements in 12 Drosophila genomes using evolutionary signatures. *Nature*, 450(7167) :219–232.

Stoye, J. and Wittler, R. (2009). A unified approach for reconstructing ancient gene clusters. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(3) :387–400.

Sturtevant, A. H. (1921). A Case of Rearrangement of Genes in Drosophila. *Proceedings of the National Academy of Sciences*, 7(8) :235–237.

Sturtevant, A. H. (1926). A Crossover Reducer in Drosophila Melanogaster due to Inversion of a Section of the Third Chromosome. *Biologischen Zentralblatt*, 46 :697–702.

Sturtevant, A. H. and Dobzhansky, T. (1936). Inversions in the Third Chromosome of Wild Races of Drosophila Pseudoobscura, and Their Use in the Study of the History of the Species. *Proceedings of the National Academy of Sciences of the United States of America*, 22(7) :448–50.

Szöllősi, G. J., Davín, A. A., Tannier, E., Daubin, V., and Boussau, B. (2015). Genome-scale phylogenetic analysis finds extensive gene transfer among fungi. *Philosophical Transactions of the Royal Society B : Biological Sciences*, 370(1678) :20140335.

Szöllősi, G. J., Tannier, E., Lartillot, N., and Daubin, V. (2013). Lateral gene transfer from the dead. *Systematic Biology*, 62(3) :386–397.

Talavera, G. and Castresana, J. (2007). Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. *Systematic biology*, 56(4) :564–77.

Tamazian, G., Dobrynin, P., Krasheninnikova, K., Komissarov, A., Koepfli, K.-P., and O'Brien, S. J. (2016). Chromosomer : a reference-based genome arrangement tool for producing draft chromosome sequences. *GigaScience*, 5(1) :38.

Tannier, E., Zheng, C., and Sankoff, D. (2009). Multichromosomal median and halving problems under different genomic distances. *BMC bioinformatics*, 10(1) :120.

Trapnell, C. and Salzberg, S. L. (2009). How to map billions of short reads onto genomes. *Nature Biotechnology*, 27(5) :455–457.

Treangen, T. J. and Salzberg, S. L. (2012). Repetitive DNA and next-generation sequencing : Computational challenges and solutions. *Nature Reviews Genetics*, 13(1) :36–46.

Tyner, C., Barber, G. P., Casper, J., Clawson, H., Diekhans, M., Eisenhart, C., Fischer, C. M., Gibson, D., Gonzalez, J. N., Guruvadoo, L., Haeussler, M., Heitner, S., Hinrichs, A. S., Karolchik, D., Lee, B. T., Lee, C. M., Nejad, P., Raney, B. J., Rosenbloom, K. R., Speir, M. L., Villarreal, C., Vivian, J., Zweig, A. S., Haussler, D., Kuhn, R. M., and James Kent, W. (2017). The UCSC Genome Browser database : 2017 update. *Nucleic Acids Research*, 45(D1) :D626–D634.

van Hijum, S. A. F. T., Zomer, A. L., Kuipers, O. P., and Kok, J. (2005). Projector 2 : Contig mapping for efficient gap-closure of prokaryotic genome sequence assemblies. *Nucleic Acids Research*, 33(Web Server issue) :560–566.

Višňovská, M., Tomáš, V., and Brejová, B. (2013). DNA Sequence Segmentation Based on Local Similarity. In *ITAT 2013 proceedings, CEUR workshop proceedings*, volume 1003, pages 36–43.

Wajid, B. and Serpedin, E. (2012). Review of General Algorithmic Features for Genome Assemblers for Next Generation Sequencers.

Walker, B. J., Abeel, T., Shea, T., Priest, M., Abouelliel, A., Sakthikumar, S., Cuomo, C. a., Zeng, Q., Wortman, J., Young, S. K., and Earl, A. M. (2014). Pilon : An integrated tool for comprehensive microbial variant detection and genome assembly improvement. *PLoS ONE*, 9(11) :e112963.

Waterhouse, R. M., Tegenfeldt, F., Li, J., Zdobnov, E. M., and Kriventseva, E. V. (2013). Orthodb : a hierarchical catalog of animal, fungal and bacterial orthologs. *Nucleic Acids Research*, 41(D1) :D358–D365.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6) :80–83.

Williams, L. J. S., Tabbaa, D. G., Li, N., Berlin, A. M., Shea, T. P., MacCallum, I., Lawrence, M. S., Drier, Y., Getz, G., Young, S. K., Jaffe, D. B., Nusbaum, C., and Gnirke, A. (2012). Paired-end sequencing of Fosmid libraries by Illumina. *Genome Research*, 22(11) :2241–2249.

Wittler, R., Maňuch, J., Patterson, M., and Stoye, J. (2011). Consistency of Sequence-Based Gene Clusters. *Journal of Computational Biology*, 18(9) :1023–1039.

Woese, C. R. and Fox, G. E. (1977). Phylogenetic structure of the prokaryotic domain : The primary kingdoms. *Proceedings of the National Academy of Sciences*, 74(11) :5088–5090.

Woese, C. R., Kandler, O., and Wheelis, M. L. (1990). Towards a natural system of organisms : proposal for the domains Archaea, Bacteria, and Eucarya. *Proceedings of the National Academy of Sciences*, 87(12) :4576–4579.

Xu, A. W. and Moret, B. M. E. (2011). GASTS : Parsimony scoring under rearrangements. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6833 LNBI, pages 351–363.

Yancopoulos, S., Attie, O., and Friedberg, R. (2005). Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16) :3340–3346.

Zheng, C. (2010). Pathgroups, a dynamic data structure for genome reconstruction problems. *Bioinformatics*, 26(13) :1587–1594.

Zheng, C. and Sankoff, D. (2011). On the PATHGROUPS approach to rapid small phylogeny. *BMC bioinformatics*, 12 Suppl 1(Suppl 1) :S4.

Zuckerkandl, E. and Pauling, L. (1965). Evolutionary divergence and convergence in proteins. *Evolving Genes and Proteins*, 42(2) :97–166.

Reconstruction conjointe de l'ordre des gènes de génomes actuels et ancestraux et de leur évolution structurale dans un cadre phylogénétique

par Yoann Louis Anselmetti

Résumé

L'émergence des technologies de séquençage haut-débit a permis, au cours des années 2000, l'augmentation d'exponentielle du nombre de génomes pour lesquels la séquence d'ADN complète est disponible. L'accès à cette multitude de génomes a ouvert la voie à l'étude de l'évolution de la structure linéaire des génomes, sous la forme d'un ordre de marqueurs génétiques. La nécessité pour les technologies de séquençage haut-débit de fragmenter les génomes avant de les séquencer a nécessité le développement d'approches algorithmiques pour l'assemblage de génomes qui consiste à reconstituer l'enchaînement complet des nucléotides le long des chromosomes. En plus de la complexité algorithmique de ce problème, la présence de séquences répétées au sein des génomes accentue la difficulté d'assemblage des génomes aboutissant à une reconstitution incomplète de la structure linéaire. Cette fragmentation des génomes implique une reconstruction incomplète de l'évolution structurale des génomes.
Dans ce contexte, nous avons développé un outil algorithmique qui permet de conjointement reconstruire l'ordre de gènes d'espèces ancestrales et d'améliorer la reconstitution de l'ordre des gènes chez les espèces actuelles de la phylogénie considérée. La méthode permet de considérer les duplications, pertes et transferts de gènes dans l'inférence de l'évolution de l'ordre des gènes.

Mots-clés: bioinformatique, reconstruction de génomes ancestraux, *scaffolding*, évolution de l'ordre des gènes, adjacences de gènes, phylogénie, *Anopheles*

Joint reconstruction of gene order of ancestral and extant genomes and their structural evolution in a phylogenetic context

by Yoann Louis Anselmetti

Abstract

The emergence of high throughput sequencing technologies allowed, in the 2000s, an exponential increase of the number of genomes for which the complete DNA sequence is available. Access to this multitude of genomes has opened the way for studying the evolution of the linear structure of genomes, in the form of an order of genetic markers. The need for high-throughput sequencing technologies to fragment genomes before sequencing has required the development of algorithmic approaches to genome assembly that consists of reconstructing the complete sequence of nucleotides along chromosomes. In addition to the algorithmic complexity of this problem, the presence of repeated sequences within genomes accentuates the difficulty of assembling genomes resulting in an incomplete reconstruction of the linear structure. This fragmentation of the genomes implies an incomplete reconstruction of the structural evolution of the genomes.
In this context, we have developed an algorithmic tool that allows to jointly reconstruct the order of genes of ancestral species and to improve the reconstruction of gene order in extant species of the considerd phylogeny. The method allows to consider the duplications, losses and transfers of genes in the inference of the evolution of the gene order.

Keywords: bioinformatics, ancestral genome reconstruction, scaffolding, gene order evolution, gene adjacencies, phylogeny, *Anopheles*