Université de Montréal

Étude de l'évolution des génomes par duplications, pertes et réarrangements

par
Olivier Tremblay Savard

Département de biochimie et médecine moléculaire
Faculté de médecine

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en Bio-informatique

Octobre, 2013

Université de Montréal
Faculté des études supérieures

Cette thèse intitulée:

Étude de l'évolution des génomes par duplications, pertes et réarrangements

présentée par:

Olivier Tremblay Savard

a été évaluée par un jury composé des personnes suivantes:

Sylvie Hamel,	président-rapporteur
Nadia El-Mabrouk,	directeur de recherche
B. Franz Lang,	membre du jury
Anthony Labarre,	examinateur externe
Maja Krajinovic,	représentant du doyen de la FES

Thèse acceptée le: . . . . . .

RÉSUMÉ

La duplication est un des évènements évolutifs les plus importants, car elle peut mener à la création de nouvelles fonctions géniques. Durant leur évolution, les génomes sont aussi affectés par des inversions, des translocations (incluant des fusions et fissions de chromosomes), des transpositions et des délétions. L'étude de l'évolution des génomes est importante, notamment pour mieux comprendre les mécanismes biologiques impliqués, les types d'évènements qui sont les plus fréquents et quels étaient les contenus en gènes des espèces ancestrales. Afin d'analyser ces différents aspects de l'évolution des génomes, des algorithmes efficaces doivent être créés pour inférer des génomes ancestraux, des histoires évolutives, des relations d'homologies et pour calculer les distances entre les génomes.

Dans cette thèse, quatre projets reliés à l'étude et à l'analyse de l'évolution des génomes sont présentés :

1. Nous proposons deux algorithmes pour résoudre des problèmes reliés à la duplication de génome entier : un qui généralise le problème du *genome halving* aux pertes de gènes et un qui permet de calculer la double distance avec pertes.
2. Nous présentons une nouvelle méthode pour l'inférence d'histoires évolutives de groupes de gènes orthologues répétés en tandem.
3. Nous proposons une nouvelle approche basée sur la théorie des graphes pour inférer des gènes in-paralogues qui considère simultanément l'information provenant de différentes espèces afin de faire de meilleures prédictions.
4. Nous présentons une étude de l'histoire évolutive des gènes d'ARN de transfert chez 50 souches de *Bacillus*.

Mots clés: Algorithme, inférence, histoire évolutive, génome ancestral, homologie, in-paralogie, duplication, réarrangement génomique, perte.

ABSTRACT

Gene duplication is one of the most important types of events affecting genomes during their evolution because it can create novel gene function. During the evolution process, genomes are also affected by inversions, translocations (including chromosome fusions and fissions), transpositions and deletions. Studying the evolution of genomes is important to get a better understanding of the biological mechanisms involved, which types of events are more frequent than others and what was the gene content in the ancestral species just to name a few. In order to analyze these different aspects of genome evolution, efficient algorithms need to be developed to infer ancestral genomes, evolutionary histories, homology relationships between genes and to compute distances between genomes.

In this thesis, four different projects related to the study and analysis of genome evolution are presented:

1. We developed two algorithms to solve problems related to whole genome duplication: one that generalizes the genome halving problem to gene losses, and one that allows to compute the double distance with losses.

2. We developed a new method to infer evolutionary histories of orthologous tandemly arrayed gene clusters.

3. We proposed a new graph-theoretic approach to infer inparalogs that simultaneously considers the information given by multiple species in order to make better inferences of inparalogous gene pairs.

4. We studied the evolutionary history of the tRNA genes of 50 *Bacillus* strains.

Keywords: Algorithm, inference, evolutionary history, ancestral genome, homology, inparalogy, duplication, genomic rearrangement, loss.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES SIGLES

ADH	Alcool DésHydrogénase
ADN	Acide DésoxyriboNucléique
ADNc	Acide DésoxyriboNucléique complémentaire
ARN	Acide RiboNucléique
ARNt	Acide RiboNucléique de transfert
ARNm	Acide RiboNucléique messager
CMH	Complexe Majeur d'Histocompatibilité
DCJ	Double Cut and Join
DGE	Duplication de Génome Entier
GDH	Glutamate DésHydrogénase
GRT	Gènes Répétés en Tandem
GTP	Guanosine TriPhosphate
MCL	Markov CLuster algorithm
MRB	Meilleur Résultat Bidirectionnel
pb	paires de bases
TAG	Tandemly Arrayed Gene
WGD	Whole Genome Duplication

À mon filleul Félix.

Peut-être qu'un jour, tu liras cette thèse...

REMERCIEMENTS

J'aimerais remercier tous les gens qui ont été impliqués de près ou de loin dans les travaux de cette thèse et qui ont rendu mon travail plus facile par leur aide et leur support.

Je voudrais tout d'abord remercier ma directrice de recherche, Nadia El-Mabrouk, pour sa grande disponibilité, ses conseils constructifs et son aide précieuse dans la rédaction des articles, la préparation des présentations orales et la relecture de ma thèse. Je tiens à la remercier de m'avoir accueilli dans son laboratoire pour la première fois lors de mon stage de deuxième année de baccalauréat et ensuite pour mon doctorat.

Je voudrais ensuite remercier les collègues avec qui j'ai eu le plaisir de travailler pendant ces dernières années dans le laboratoire de Nadia : Mathieu Lajoie, Denis Bertrand, Yves Gagnon, Krister Swenson, Manuel Lafond et Billel Benzaid. Je tiens à remercier spécialement Yves qui, en plus d'avoir été un excellent collègue de travail, est un grand ami sur qui je peux toujours compter et ce, depuis le début du baccalauréat. Je voudrais également remercier particulièrement Mathieu, Denis et Krister, qui ont été des mentors pour moi. Je remercie Mathieu pour m'avoir encadré pendant mon stage dans le laboratoire et pour ses travaux qui ont inspiré un des projets de cette thèse. Je remercie Denis pour m'avoir guidé dans la réalisation des deux premiers projets de mon doctorat, pour sa disponibilité (même une fois rendu à l'autre bout du monde !) et pour toutes les discussions intéressantes. Je remercie Krister pour sa rigueur scientifique, pour ses connaissances théoriques et pour les confrontations légendaires de NHL94.

Je tiens à exprimer ma reconnaissance envers ma famille qui m'a toujours supporté dans mon cheminement professionnel et personnel.

Je remercie tous les professeurs qui se sont impliqués dans les programmes de bioinformatique de l'Université de Montréal. Je veux également remercier le personnel de soutien que j'ai côtoyé pendant ces dernières années, notamment : Élaine Meunier, Marie

Pageau, Audrey Noël, Philipe Lampron et Marie Robichaud.

Je remercie tous les membres du jury pour avoir accepté de réviser cette thèse.

En terminant, je remercie les différents organismes qui m'ont supporté financièrement durant mes études doctorales : le Fonds de recherche du Québec - Nature et technologies (FRQNT), le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), la Faculté des études supérieures et postdoctorales de l'Université de Montréal (FESP) et les bourses d'excellence biT.

CHAPITRE 1

INTRODUCTION

La duplication de gènes est un processus fondamental dans l'évolution des espèces car elle joue un rôle important dans la création de nouvelles fonctions. Plusieurs mécanismes biologiques sont à l'origine des duplications de gènes, comme par exemple la duplication de génome entier (DGE), la duplication en tandem, la transposition duplicative (à l'origine des duplications segmentales) et la rétroposition. Cependant, les gènes ne sont pas uniquement sujets à des amplifications. Des pertes de gènes ou de segments d'ADN comptant plusieurs gènes peuvent également se produire durant l'évolution des génomes. Enfin, plusieurs types de réarrangements génomiques peuvent affecter l'ordre et l'orientation des gènes. Des traces d'évènements d'inversion, de transposition, de translocation, de fusion et de fission de chromosomes sont souvent observées dans les génomes.

Cette thèse porte sur l'étude de différents problèmes reliés à l'analyse des histoires évolutives de génomes ayant été affectés par tous ces types d'évènements évolutifs : inférence de génomes ancestraux, calcul de distances entre génomes, inférence d'histoires évolutives et identification de relations d'homologie entre les gènes. Plus précisément, les différents modes de duplications sont au centre des différents projets de cette thèse. Les articles présentés dans les chapitres 4 et 5 explorent respectivement la duplication de génome entier et la duplication en tandem. Le travail présenté dans le chapitre 6 s'intéresse à l'inférence de relations d'in-paralogie entre les gènes dans le but d'analyser la fréquence des différents types de duplication, tandis que le chapitre 7 présente une étude de l'évolution des gènes d'ARN de transfert (ARNt) chez les bactéries par différents modes de duplication, par pertes et par réarrangements.

Le chapitre 2 de cette thèse présente en détail les concepts biologiques qui sont à la

base de l'évolution des génomes. Les différents mécanismes biologiques responsables de l'apparition de nouvelles copies de gènes, de la délétion de séquences d'ADN ainsi que des réarrangements génomiques sont abordés. Nous présentons également dans ce chapitre les principes de sélection naturelle qui déterminent la fixation de nouveaux caractères dans une population ainsi que les différents types de relations d'homologie possibles entre les gènes.

Dans le chapitre 3, les principes méthodologiques de base reliés à l'étude de l'évolution des génomes sont décrits. Un historique des derniers développements algorithmiques concernant les différents problèmes d'inférence abordés dans les chapitres 4 à 7 est présenté en détail. Le calcul de distances entre génomes diploïdes et tétraploïdes (ayant subi une DGE), l'inférence de génomes ancestraux pré-dupliqués, l'inférence d'histoires évolutives de groupes de gènes dupliqués en tandem, la détection de relations d'in-paralogies et l'inférence d'histoires évolutives de gènes d'ARN de transfert sont abordés dans ce chapitre.

Le chapitre 4 correspond à un article publié dans *Journal of Computational Biology* en 2011 [146]. Il s'agit d'une version plus complète d'un article présenté lors de la conférence RECOMB-CG 2010 [66]. Dans le cadre de ce travail, nous avons présenté deux algorithmes permettant de résoudre deux problèmes reliés au calcul de distance entre des génomes qui ont été doublés par des DGEs. Nous avons d'abord généralisé l'algorithme d'inférence de génome pré-dupliqué (problème du *genome halving*) d'El-Mabrouk et Sankoff [51] afin de permettre de faire l'inférence à partir d'un génome actuel ayant perdu des copies de gènes. Nous avons prouvé que ce nouvel algorithme conserve la même complexité (linéaire) que la version précédente. Nous avons également développé une heuristique pour calculer la distance entre un génome parfaitement dupliqué et un génome dupliqué réarrangé ayant perdu ou non des copies de gènes. Les simulations ont démontré que cette heuristique est très rapide et précise. Ces algorithmes sont nécessaires pour in-

férer des génomes ancestraux dans une phylogénie qui contient des évènements de DGE. L'intégration des pertes de gènes dans les deux méthodes présentées constitue un développement algorithmique majeur, car il est beaucoup plus réaliste d'un point de vue biologique de considérer qu'après une duplication de génome entier, ce ne sont pas toutes les copies de gènes qui sont conservées.

Une nouvelle méthode pour l'inférence d'histoires évolutives de groupes de gènes répétés en tandem (GRT) est présentée au chapitre 5. Ce chapitre correspond à un article publié dans le journal *BMC Bioinformatics* en 2011 [145]. Les groupes de GRT représentent des ensembles de gènes ayant des rôles divers : transport moléculaire, système immunitaire, développement embryonnaire n'en sont que quelques exemples. Connaître l'histoire évolutive d'un groupe de GRT est essentiel pour mieux comprendre les mécanismes de duplication en tandem et prédire les fonctions spécifiques de copies récemment dupliquées. La méthode que nous avons proposée est une extension d'un algorithme développé dans notre laboratoire, dédié à l'étude évolutive d'un groupe de GRT dans une espèce unique. Notre nouvelle méthode permet, quant à elle, d'étudier simultanément l'évolution de groupes de GRT orthologues chez plusieurs espèces, ce qui permet d'inférer plus précisément les inversions et les pertes de gènes. Les résultats obtenus sur des jeux de données simulées ont montré que notre algorithme est très efficace pour inférer le nombre total ainsi que la distribution des tailles des duplications. Nos expériences menées sur les gènes des protocadhérines ont également permis de confirmer des hypothèses sur leur évolution.

Peu d'algorithmes ont été développés spécifiquement pour l'inférence de gènes in-paralogues (créés après une spéciation donnée). Il s'agit toutefois d'un domaine intéressant, puisque lorsque les spéciations en question sont récentes, les gènes in-paralogues représentent nécessairement des duplications récentes. Il est donc possible d'utiliser les relations d'in-paralogie entre les gènes pour étudier les modes de duplications, car il est très probable que les gènes qui ont été dupliqués récemment n'ont pas été réarrangés de-

puis leur création (leur position dans le génome est alors la même qu'après la duplication). Nous avons présenté, dans le cadre d'un article publié dans le journal *BMC Bioinformatics* en 2012 [163] (chapitre 6), une nouvelle approche se basant sur la théorie des graphes pour détecter des gènes in-paralogues. Deux algorithmes ont été développés : un algorithme d'approximation 2/3 (assurant de trouver une solution qui correspond à 2/3 de la solution optimale) et une heuristique qui est plus rapide et plus efficace sur les graphes denses. Notre approche considère simultanément l'information provenant de plusieurs espèces, ce qui nous permet de détecter des paires d'in-paralogues moins similaires (qui peuvent être ignorées par les autres méthodes) et nous protège des faux positifs qui peuvent provenir des pertes de gènes. Notre analyse des génomes de six vertébrés et deux drosophiles ont suggéré que plusieurs duplications en tandem récentes se sont produites chez la souris et que plusieurs transpositions duplicatives récentes ont eu lieu chez l'humain.

Un grand nombre d'études ont porté sur la structure et l'identification des séquences d'ARN de transfert. Cependant, relativement peu d'analyses ont été menées sur l'évolution de ces gènes du point de vue du nombre et de l'organisation dans le génome. Avoir une meilleure connaissance de l'évolution des gènes d'ARNt peut nous amener à mieux comprendre les mécanismes évolutifs qui sont les plus fréquents et l'origine de nouveaux gènes d'ARNt. Le chapitre 7 correspond à un manuscrit en cours de rédaction portant sur les résultats de l'analyse de l'évolution des gènes d'ARNt de 50 souches de *Bacillus*. Dans le cadre de cette étude, nous avons observé que ces gènes ont surtout été affectés par les duplications et les délétions durant leur évolution. Nous avons également observé plusieurs longues inversions autour des axes de réplication des génomes (origine ou terminus). Nos découvertes les plus intéressantes ont été deux évènements de substitution d'ARNt : une qui a semblé avoir créé un nouvel ARNt initiateur chez *Bacillus amyloliquefaciens* DSM7 et une qui a transformé un ARNt-sérine en ARNt-thréonine chez *Bacillus coagulans* 2-6. La première substitution a malheureusement été invalidée plus tard par un re-séquençage

du génome en question.

Finalement, une discussion sur les forces et les faiblesses des méthodes présentées dans cette thèse ainsi que plusieurs pistes de travaux futurs est présentée dans le chapitre 8.

CHAPITRE 2

CONTEXTE BIOLOGIQUE

2.1 Introduction

Dans ce chapitre, les différents types de duplications, les mécanismes biologiques qui les rendent possibles ainsi que le devenir des gènes dupliqués sont tout d'abord présentés dans la section 2.2. Les sections suivantes (2.3, 2.4 et 2.5) portent sur les autres types d'évènements qui remodèlent l'ordre et le contenu en gènes des génomes considérés dans cette thèse : les pertes, les réarrangements et la conversion génique. Enfin, la section 2.6 résume les différentes relations hiérarchiques qui existent entre les gènes, c'est-à-dire les différents types d'homologie.

2.2 Duplication de gènes

Les quatre modes de duplication les plus connus sont la duplication de génome entier, la duplication en tandem, la duplication segmentale et la rétroposition.

2.2.1 Duplication de génome entier

La duplication de génome entier (DGE) est l'évènement le plus spectaculaire menant à la création de nouvelles copies de gènes. Comme son nom l'indique, la duplication de génome entier consiste en une duplication de tous les chromosomes. Elle est le résultat d'une erreur lors de la méiose. En effet, elle se produit lorsque les paires de chromosomes homologues ne sont pas séparés lors de l'anaphase 1 ou 2.

La duplication de génome entier est un évènement qui a été très présent dans l'évolution des plantes. À la fin de 2012, on détectait au moins une DGE dans l'histoire évolutive

de chaque plante à fleurs séquencée [108]. En effet, une DGE se serait produite très tôt dans l'histoire évolutive des angiospermes, probablement chez l'ancêtre de toutes les plantes à fleurs [37, 151]. De plus, on a découvert des preuves de DGE supplémentaires chez les graminées [132], les astéracées [13] et les brassicacées [113].

Des études ont montré qu'un évènement de duplication de génome entier s'est produit au moins une fois [81, 115] chez l'ancêtre des vertébrés et probablement même deux fois [82, 106, 155]. Il y aurait eu ensuite une autre DGE dans la lignée des poissons téléostéens (poissons osseux) [90, 120, 177] et une DGE supplémentaire dans la lignée des poissons salmoniformes [5].

Une duplication de génome entier a également été identifiée dans l'histoire évolutive des levures [74, 92].

À la suite d'un évènement de duplication de génome entier, on assiste souvent à la perte d'une des deux copies de chaque gène. Ce processus, qui se nomme fractionnement, permet à l'expression des gènes du génome dupliqué de revenir à un niveau normal. Toutefois, il peut parfois être bénéfique pour l'organisme d'avoir un niveau d'expression plus élevé pour certains gènes, ce qui explique que pour de tels gènes, toutes les copies sont conservées. Un bon exemple est celui des gènes Hox. Ces gènes codent pour des facteurs de transcription qui contrôlent le développement embryonique le long de l'axe antérieur-postérieur de presque tous les métazoaires. Chez la plupart des espèces, les gènes Hox sont organisés en groupes de gènes (clusters). On croit présentement que la duplication de génome entier est directement responsable de l'amplification du nombre de groupes de gènes Hox chez les vertébrés. En effet, les mammifères possèdent 4 clusters de gènes Hox (en accord avec la théorie des deux DGE chez l'ancêtre des vertébrés), alors que les poissons téléostéens en possèdent 7 (une DGE supplémentaire et la perte d'un groupe) et le saumon 13 (une autre DGE supplémentaire suivie d'une perte d'un cluster).

2.2.2 Duplication en tandem

La duplication en tandem correspond à une duplication d'un segment d'ADN contenant un seul (duplication en tandem simple) ou plusieurs (duplication en tandem multiple) gènes, qui place la copie directement à la suite du segment original.

Le mécanisme biologique reconnu comme responsable de cet évènement évolutif est la recombinaison inégale lors de la méiose. Lors d'un évènement de recombinaison normal, qu'on appelle recombinaison homologue allélique, les deux chromosomes sont parfaitement alignés comme dans l'exemple de la Figure 2.1 a). L'évènement de recombinaison échange alors des segments de gènes entre les deux chromosomes. Dans l'exemple de la Figure 2.1 a), les gènes oranges ont été échangés entre les deux chromosomes homologues. Lors d'une recombinaison inégale, qu'on nomme aussi recombinaison homologue non-allélique, il y a un appariement inégal des chromosomes homologues. Dans l'exemple de la Figure 2.1 b), on peut voir que cet alignement inégal des chromosomes peut mener soit à la duplication du gène vert, soit à la perte du gène vert.

Plus une région contient de séquences répétées, plus les chances qu'il se produise un appariement inégal à cet endroit sont grandes. Une duplication en tandem peut donc être l'élément déclencheur d'une série de duplications en tandem qui mèneront à la création de groupes de gènes répétés en tandem (GRT). Plusieurs familles de gènes, jouant différents rôles biologiques importants, sont organisées en groupes de GRT. Les gènes des récepteurs olfactifs (détection des odeurs), les protocadhérines (développement neuronal) et les gènes de l'hémoglobine (transport de l'oxygène) en sont des exemples.

2.2.3 Duplication segmentale

La duplication segmentale correspond à une duplication d'un large segment de chromosome. D'après des études menées sur le génome humain, la taille des duplications seg-

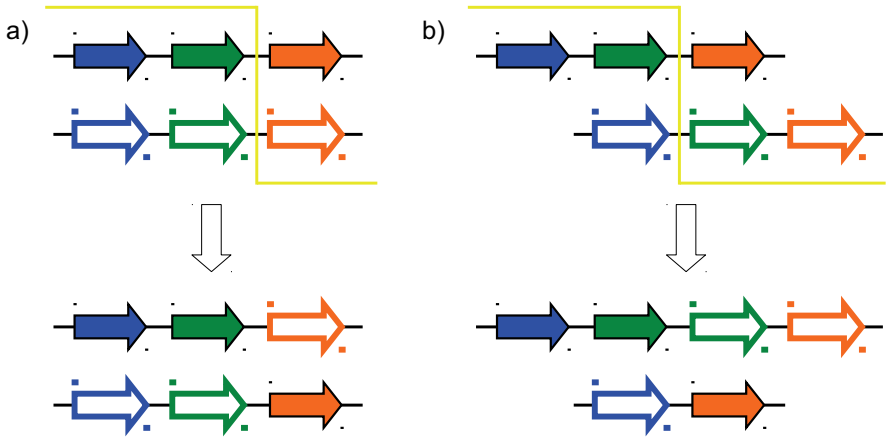

Figure 2.1 – Exemples de recombinaison. Les gènes sont représentés par des flèches. Chaque ligne horizontale représente un chromosome homologue et la ligne jaune représente l'endroit ou se produit l'échange lors de la recombinaison. Les gènes provenant d'un chromosome homologue ont une surface coloriée alors que ceux de l'autre ont seulement le contour en couleur. a) Un exemple de recombinaison homologue allélique. Les chromosomes homologues sont parfaitement alignés lors de l'évènement de recombinaison. b) Un exemple de recombinaison homologue non-allélique. Les chromosomes homologues ne sont pas alignés correctement lors de la recombinaison.

mentales peut varier entre 1 et 200 kpb [12, 34]. On nomme le phénomène biologique responsable des duplications segmentales *transposition duplicative*. Il a également été nommé *duplication dérivée* (*drift duplication*) [56] puisque les segments dupliqués peuvent se retrouver à une grande distance du segment original et même sur différents chromosomes. Toutefois, on en connaît encore peu sur le fonctionnement du mécanisme de transposition duplicative. Une des hypothèses serait que, chez les primates, un évènement de recombinaison impliquant des répétitions Alu — des séquences répétées d'une longueur d'environ 300 pb qu'on retrouve chez les primates [86] — serait responsable d'une partie des duplications segmentales. En effet, on a observé dans le génome humain qu'un tiers des duplications segmentales étaient flanquées de séquences Alu [11]. Les séquences d'ADN satellite — longs segments d'ADN constitués de courtes répétitions en tandem d'ADN non codant — pourraient également être impliquées, puisqu'on a également observé un

enrichissement de ces séquences aux frontières des duplications segmentales [11].

2.2.4 Rétroposition

La rétroposition se produit lorsqu'un ARN messager (ARNm) est rétrotranscrit en ADN complémentaire (ADNc) et ensuite réinséré n'importe où dans le génome. Étant donné que le processus de rétroposition implique un intermédiaire d'ARNm, les gènes dupliqués par ce mécanisme possèdent plusieurs caractéristiques particulières : ils ne possèdent pas d'introns, ils se terminent souvent par un segment poly(A), un seul gène peut être dupliqué à la fois et les promoteurs ne sont pas dupliqués avec le gène. Les nouvelles copies de gènes créées par rétroposition peuvent se retrouver sur des chromosomes différents et il est plutôt rare qu'elles puissent être transcrites à cause de la perte du promoteur. Toutefois, il peut arriver que la copie se place par hasard en aval d'un promoteur, ce qui permettra sa transcription.

Il ne faut pas confondre la rétroposition avec la rétrotransposition. En effet, les rétrotransposons sont des éléments génétiques capables de s'amplifier par eux-même dans le génome, car ils encodent également la transcriptase inverse qui permet de transcrire l'ARNm en ADNc.

2.2.5 Sélection naturelle

Les mutations, les duplications, les pertes de gènes ou les réarrangements génomiques sont des évènements qui se produisent par hasard. Toutefois, il faut souvent (mais pas nécessairement) plus que du hasard afin que des modifications au niveau du génome se fixent dans une population. C'est la sélection naturelle qui entre en jeu.

La sélection naturelle est un concept qui a été formulé par Charles Darwin dans son célèbre livre *On the Origin of Species* [39]. C'est le processus par lequel certain phéno-

types, apparaissant par hasard chez certains individus dans une population, seront favorisés tandis que d'autres seront défavorisés. Après une longue période de temps, la sélection naturelle a un effet sur la prévalence des traits biologiques dans une population, ce qui peut également mener à la création de nouvelles espèces. Le terme *sélection naturelle* a été choisi pour faire contraste avec celui de *sélection artificielle*, qui consiste en l'intervention de l'être humain dans le processus de reproduction des plantes ou des animaux d'élevage.

Le processus de sélection naturelle, qui a été introduit par Darwin suite à une observation des phénotypes, affecte évidemment aussi le génotype. Les gènes sont soumis à différents types de sélection naturelle, qui sont présentés en détail dans les sections suivantes.

2.2.5.1 Sélection positive

La sélection positive est le principe selon lequel les individus qui possèdent des traits biologiques avantageux auront plus de chances de survivre et de se reproduire, ce qui aura comme effet, après un certain temps, d'augmenter le nombre d'individus qui possèdent ces traits dans la population par hérédité [40]. Au niveau des gènes, la sélection positive est le processus qui encourage la rétention des mutations qui sont bénéfiques pour un individu. Ce type de sélection peut agir plus fortement dans les cas où un organisme se retrouve dans un nouvel environnement ou lorsque son environnement est grandement modifié.

La sélection positive peut prendre deux formes : elle peut être directionnelle ou non directionnelle [36]. La sélection positive directionnelle se produit lorsque des changements d'acides aminés rendent une protéine plus efficace dans son rôle et de plus en plus d'individus se retrouvent avec cette nouvelle protéine dans la population. La sélection positive non directionnelle quant à elle se produit par exemple lorsqu'un environnement est constamment en état de changement et différentes sortes de mutations sont tolérées dans le but de

maintenir une certaine capacité à survivre dans la population.

Le gène de la lysozyme, une enzyme qui détruit les parois des bactéries, a subi un épisode de sélection positive directionnelle dans la lignée des colobinés [116, 182]. Les singes colobinés sont différents des autres primates, car ils possèdent un système complexe de pré-estomacs où les bactéries digèrent la cellulose des plantes par fermentation. Ce système est suivi d'un vrai estomac, dans lequel les bactéries sont détruites par l'action de la lysozyme qui est fortement exprimée. Chez les autres primates, qui possèdent un estomac simple, la lysozyme est exprimée seulement dans la partie inférieure de l'estomac (la région du pylore). Chez les colobinés, le gène de la lysozyme a évolué de façon à permettre à l'enzyme de fonctionner dans les fluides de l'estomac (résistance à un pH très bas et au clivage par la pepsine).

La famille de gènes du complexe majeur d'histocompatibilité (CMH) est un exemple de sélection positive non directionnelle. Chez l'humain, cette famille se situe sur le chromosome 6 et contient 421 gènes (dont 60% sont exprimés) condensés dans une région de 7.6 Mpb [85]. Chez tous les vertébrés, les gènes du CMH de classe I et II encodent pour des glycoprotéines exprimées à la surface de toutes les cellules nucléées qui lient des protéines se retrouvant dans les cellules (étrangères ou non). Les molécules du CMH de classe I et II présentent donc des antigènes aux lymphocytes T cytotoxiques (un type de cellules du système immunitaire), qui peuvent ensuite provoquer la lyse de la cellule. Les gènes du CMH de classe I et II sont extrêmement polymorphiques [54], c'est-à-dire qu'on retrouve plusieurs allèles pour chaque locus de gène dans les populations. Étant donné qu'une certaine molécule peut reconnaître plus facilement un certain pathogène, les individus qui expriment différents types de gènes du CMH peuvent être avantagés dans une population exposée à plusieurs pathogènes, ce qui explique pourquoi la sélection positive non directionnelle influence ces gènes [88].

2.2.5.2 Sélection négative

La sélection négative, également nommée sélection purificatrice, se produit lorsqu'une protéine est parfaitement adaptée à l'accomplissement de son rôle et qu'une modification de cette protéine serait néfaste ou même létale pour l'organisme. La sélection négative assure une certaine stabilité dans les populations par le fait que les individus ayant des mutations qui les rendent moins adaptés disparaîtront de la population. La plupart des protéines sont sous l'effet de la sélection négative, car elles sont le résultat d'une longue période d'évolution et d'adaptation et des modifications risquent de les rendre non fonctionnelles. On n'a qu'à penser aux maladies humaines qui sont le résultat de mutations génétiques pour obtenir des exemples de protéines qui subissent une pression de sélection négative.

2.2.5.3 Sélection neutre

La sélection neutre représente en fait une absence de sélection positive ou négative. Les séquences qui ne sont affectées par aucune pression de sélection peuvent être modifiées sans que ce soit défavorable pour l'organisme. La sélection neutre peut se produire tout de suite après une duplication de gène par exemple, alors qu'un des deux gènes peut être modifié librement en autant que l'autre copie du gène reste fonctionnelle.

2.2.5.4 Identification du type de sélection

L'approche la plus utilisée pour identifier quelle sélection est à l'oeuvre est celle de la comparaison du nombre de substitutions non synonymes (représenté par d_n) avec le nombre de substitutions synonymes (représenté par d_s). Contrairement à une substitution non synonyme, une substitution synonyme est une substitution silencieuse, c'est-à-dire qu'elle ne changera pas l'acide aminé qui sera intégré dans la protéine. Plusieurs

méthodes existent pour estimer les nombres de substitutions synonymes et non synonymes [89, 102, 104, 119, 122, 183]. Elles nécessitent toutes trois étapes : le calcul du nombre de sites synonymes et non synonymes, le calcul du nombre de différences synonymes et non synonymes entre les deux séquences homologues étudiées (ce qui nécessite un alignement des séquences) et une correction statistique pour tenir compte des substitutions invisibles (celles qui ont eu lieu au même site).

Après avoir obtenu les valeurs de d_n et d_s, on utilise le ratio d_n/d_s pour identifier le type de sélection. Une valeur de d_n/d_s inférieure à 1 est un indice de sélection négative : les substitutions synonymes peuvent se produire, mais les substitutions non synonymes sont très rares. Un ratio d_n/d_s supérieur à 1 est une bonne indication que la sélection positive est à l'oeuvre. Dans ce cas, les changements d'acides aminés dans la protéine sont bénéfiques pour l'organisme et le nombre de substitutions non synonymes dépasse celui des substitutions synonymes. Enfin, une valeur de d_n/d_s égale à 1 indique que la séquence ne subit aucune pression de sélection, c'est-à-dire qu'elle est sous sélection neutre.

2.2.6 Devenir des gènes dupliqués

Tout de suite après la duplication d'un gène, une des deux copies est redondante et totalement superflue. Pendant une certaine période de temps, la pression de sélection est relaxée, ce qui permet une évolution neutre d'une des deux copies. Quatre scénarios sont ensuite possibles pour ces gènes : la conservation de fonction, la néo-fonctionnalisation, la sous-fonctionnalisation ou la pseudogénisation. Lynch et Conery ont estimé que le taux de création et rétention des gènes dupliqués est d'environ 0.01 par gène par million d'années chez les eucaryotes [107].

2.2.6.1 Conservation de fonction

Dans certains cas, la création d'une deuxième copie de gène peut être bénéfique pour l'organisme en augmentant la quantité de la protéine exprimée dans la cellule. Dans ce cas, une forte sélection négative permet de conserver la fonction de toutes les copies du gène. La préservation d'un ensemble de copies presque identiques d'un même gène peut également être le résultat du mécanisme de conversion génique. Afin de vérifier si c'est la conversion génique ou la sélection purificatrice qui est à l'oeuvre, il suffit de vérifier la quantité de mutations synonymes entre les copies. En effet, la sélection négative ne devrait pas empêcher les mutations synonymes de se produire, tandis que la conversion génique homogénéise complètement les séquences.

La conservation de fonction peut également se produire après une duplication de génome entier. En effet, à la suite d'une DGE, certains gènes ne sont pas affectés par le fractionnement. Cela se produit lorsqu'un certain dosage de protéines doit être maintenu pour former un complexe protéique ou pour conserver l'efficacité d'un réseau de transcription ou de signalement par exemple. Une perte de cet équilibre peut alors être très néfaste pour l'organisme et même létale [22, 23, 131, 165–167]. C'est pour cette raison que toutes les copies de gènes sensibles au dosage doivent être conservées après une DGE. On appelle cette théorie l'hypothèse de l'équilibre des gènes (*Gene Balance Hypothesis*) ou l'hypothèse de l'équilibre du dosage (*Dosage Balance Hypothesis*).

2.2.6.2 Néo-fonctionnalisation

La néo-fonctionnalisation est le phénomène le plus rare et le plus spectaculaire pouvant se produire après la duplication d'un gène. Une des deux copies du gène accumule des mutations qui lui permettront d'acquérir une nouvelle fonction. La nouvelle fonction reste habituellement similaire à celle d'origine (par exemple, la nouvelle fonction peut conserver

sensiblement le même mécanisme enzymatique).

Un exemple de néo-fonctionnalisation a été documenté chez l'humain. Il s'agit du gène *GLUD2* (situé sur le chromosome X) qui a été dupliqué par rétroposition du gène *GLUD1* (qui se retrouve sur le chromosome 10). Ces deux gènes codent pour la glutamate déshydrogénase (GDH), une enzyme mitochondriale qui désamine le glutamate. Alors que *GLUD1* est exprimé dans plusieurs tissus, *GLUD2* l'est principalement dans les testicules et le cerveau. Il a été démontré que les nouvelles fonctions du gène *GLUD2* facilitent son activité dans les tissus nerveux entre autres en permettant à la GDH qu'il exprime de fonctionner à un pH plus bas et de résister à l'inhibition de la GTP (guanosine triphosphate) qui se retrouve en plus grande concentration dans le cerveau [135].

Un deuxième exemple de ce processus a été identifié chez la drosophile. Dans ce cas, c'est le gène chimérique jingwei (JGW) qui a acquéri une nouvelle fonction. Ce gène a d'abord été formé lorsque le gène de l'alcool déshydrogénase (ADH) s'est inséré par rétroposition dans un intron du gène yande (YND). Après quelques substitutions d'acides aminés subséquentes en dehors du site actif, le gène JGW est devenu une nouvelle déshydrogénase avec des affinités de substrats différentes de celles de l'ADH. Contrairement à l'ADH qui a une fonction de désintoxication et d'assimilation de l'éthanol, des études suggèrent que JGW pourrait jouer un rôle dans la synthèse ou la dégradation d'hormones et de phéromones [184].

2.2.6.3 Sous-fonctionnalisation

Un autre phénomène possible pour les deux copies du gène est la sous-fonctionnalisation. Ceci peut se produire lorsque le gène d'origine possède au moins deux fonctions. Dans ce cas, chacune des deux copies peut perdre une fonction (pas la même) par des mutations dégénératives et n'en conserver qu'une seule. Les différentes copies du gène possèdent

alors des fonctions complémentaires et sont assujetties à une forte sélection purificatrice qui permet à l'organisme de maintenir la totalité de la fonction ancestrale. Il se peut également que chaque copie se spécialise et devienne plus efficace dans la fonction qu'elle a retenue. À ce moment, les deux copies du gène sont affectées par des mutations adaptatives (c'est-à-dire non neutres).

La paire de gènes dupliqués SIR3 et OCR1 chez *Saccharomyces cerevisiae* est un cas évident de sous-fonctionnalisation. Ces deux gènes, provenant d'une duplication de génome entier, possèdent chacun une fonction différente : SIR3 joue un rôle d'extinction de gènes (processus épigénétique empêchant l'expression de gènes) tandis que ORC1 participe au complexe de reconnaissance de l'origine de réplication. Il a été démontré que *Saccharomyces kluyveri*, une levure qui n'a pas subi de DGE, possède une protéine homologue qui remplace les deux fonctions de SIR3 et OCR1 [164].

La sous-fonctionnalisation peut s'effectuer d'une autre manière : dans le cas où le gène d'origine a besoin de deux promoteurs pour être transcrit dans deux tissus différents, chacune des deux copies du gène (et leurs promoteurs, qui ont été dupliqués aussi) peuvent conserver un seul des deux promoteurs (encore une fois, chaque copie conservera un promoteur différent). Il en résultera que chaque copie du gène ne pourra dorénavant être exprimé que dans un seul tissu. C'est ce qui s'est produit dans le cas des gènes eng1 et eng1b chez le poisson zèbre. Ces gènes, créés par un évènement de DGE, sont des facteurs de transcription qui jouent un rôle dans le développement embryonique. Le gène eng1 est exprimé dans le *pectoral appendage bud* alors que eng1b est exprimé dans le cerveau postérieur et la colonne vertébrale. Il se trouve que le seul gène eng1 chez la souris (qui n'a pas subi la DGE spécifique aux poissons téléostéens) est exprimé dans les deux tissus [62].

2.2.6.4 Pseudogénisation

La pseudogénisation est la conséquence la plus fréquente d'une duplication de gène. Cela peut se produire de différentes façons :

- une des deux copies du gène se met à accumuler des mutations et perd complètement sa fonction ou ne peut plus être transcrite (par apparition d'un codon stop à l'intérieur du gène par exemple) ;
- la nouvelle copie de gène a été dupliquée sans son promoteur de transcription ;
- un autre évènement de duplication ou de transposition a inséré une séquence d'ADN à l'intérieur du gène ;
- un réarrangement génomique a coupé le gène en deux ou l'a séparé de son promoteur de transcription.

Un bon exemple de pseudogénisation se retrouve dans la famille de gènes des récepteurs olfactifs chez l'humain. La taille de cette famille de gènes est pratiquement identique chez l'humain et la souris (environ 1000 gènes). Cependant, plus de 70% de ces gènes sont en fait des pseudogènes chez l'humain, alors que ce pourcentage est presque nul chez la souris [140, 141]. Cette pseudogénisation importante s'est mise en marche lors de la divergence des hominidés [140] et s'explique probablement par le fait que l'olfaction est devenue moins importante chez ces derniers avec l'amélioration des capacités visuelles.

2.3 Pertes de gènes

Les pertes de gènes peuvent se produire de deux manières. Comme mentionné dans la section 2.2.2, un évènement de recombinaison inégale peut amener la perte d'un segment de gènes. Lorsqu'une région d'ADN contenant plusieurs gènes est perdue en un seul évènement, on va plutôt dire qu'il s'est produit un évènement de délétion de gènes.

La deuxième façon d'obtenir une perte de gène est par pseudogénisation, comme ex-

pliqué dans la section 2.2.6.4.

La perte de gène n'est pas létale s'il existe une autre copie dans le génome. Elle est même parfois favorable pour des raisons d'équilibre de dosage par exemple (voir section 2.2.6.1). En effet, dans le cas où une duplication de petite échelle (duplication en tandem ou duplication segmentale) affecte seulement une partie d'un réseau de protéines sensibles au dosage, la perte subséquente de ces gènes permettra de retrouver l'équilibre.

2.4 Réarrangements génomiques

Les réarrangements génomiques sont des évènements évolutifs qui, contrairement aux duplications et aux délétions, ne changent pas le contenu en gènes du génome, mais seulement l'organisation des gènes. Les réarrangements génomiques ont été abondamment étudiés chez l'humain, car ils provoquent souvent des fausses couches, des maladies ou des problèmes de développement. Les cinq types de réarrangements sont les inversions, les translocations, les fusions, les fissions et les transpositions.

Une inversion est un évènement qui se produit à l'intérieur d'un chromosome et qui inverse un segment d'ADN. La recombinaison inégale peut causer des inversions, par exemple lorsque des répétitions inversées sur les chromosomes homologues sont appariées [156]. Des inversions sont fréquemment observées chez les bactéries autour de l'origine ou du terminus de réplication [49]. Il serait donc possible qu'un mécanisme relié à la réplication soit responsable de ces évènements dans les génomes circulaires [73, 162].

Un échange de deux extrémités de chromosomes différents se nomme une translocation (réciproque). Par exemple, une translocation entre les chromosomes 11 et 22 a été documentée chez l'humain. Les enfants des individus porteurs de cette translocation sont susceptibles d'avoir le syndrome der(22), causant entre autres des déficiences mentales et des problèmes cardiaques. Il a été démontré que des séquences riches en AT et palin-

dromiques se retrouvent à l'endroit de l'échange dans les deux chromosomes [46]. Selon l'hypothèse des auteurs, ces régions ont le potentiel de former des structures en épingle à cheveux qui sont sujettes aux cassures double-brin. À la suite des cassures, une recombinaison illégitime peut se produire et causer la translocation.

La fusion (deux chromosomes qui se joignent ensemble) et la fission (un chromosome qui se sépare en deux) sont souvent considérées comme des cas particuliers de translocation (non réciproque). Le chromosome 2 de l'humain est un exemple classique de fusion de chromosomes. Il a été prouvé, par comparaison du caryotype humain avec celui des autres hominidés, que ce chromosome, le deuxième plus long du génome humain, est le résultat d'une fusion de 2 chromosomes ancestraux [45, 174, 175]. Cependant, plusieurs généticiens suggèrent que la fusion de chromosomes ne peut être le résultat d'un seul évènement de translocation non réciproque, mais se produit plutôt par une translocation réciproque entre deux chromosomes acrocentriques (ayant un bras très court et un bras long) [110, 148]. Lorsque le bras court d'un chromosome est échangé avec le bras long d'un autre, un des deux chromosomes est tellement petit qu'il risque ensuite d'être perdu lors de la méiose, ce qui donne l'apparence d'une fusion (car il ne reste plus que le long chromosome). Une fission de chromosome peut quant à elle se produire lorsque la cassure sépare le centromère en deux fragments fonctionnels et que des télomères sont ensuite ajoutés aux extrémités [148].

La transposition provoque le déplacement d'un segment d'ADN à un autre endroit dans le chromosome ou sur un chromosome différent sans qu'aucune similarité de séquence ne soit nécessaire. Des enzymes, appelées *transposases*, catalysent la transposition en coupant un segment d'ADN et en le recollant ailleurs dans le génome. Les transposases ont été réparties en cinq grandes familles, selon leur mécanisme [38].

2.5 Conversion génique

La conversion génique implique le remplacement total ou partiel de la séquence d'un gène *accepteur* par une copie de la séquence d'un autre gène *donneur*. Les gènes accepteurs et donneurs doivent avoir une grande similarité de séquence pour que cet évènement puisse se produire. Le premier exemple de conversion génique chez les mammifères a été identifié chez l'humain, dans le cas des gènes HBG1 et HBG2 qui encodent les hémoglobines γ présentes pendant le développement embryonnaire.

La conversion génique est une forme de recombinaison qui n'implique pas d'enjambement (*crossover*) des chromosomes homologues. Plusieurs modèles ont été proposés pour expliquer le mécanisme biologique de la conversion génique. Une revue détaillée de ceux-ci est présentée dans [32].

La longueur des séquences copiées par un évènement de conversion génique est assez courte. On a estimé qu'elle se situe en moyenne entre 200 et 1000 pb chez les mammifères [32]. Une conversion génique peut se produire entre des gènes situés sur le même chromosome, sur des chromatides soeurs, sur des chromosomes homologues ou même sur des chromosomes différents. La conversion génique peut également être allélique ou non-allélique. Dans une étude menée chez l'humain, il a été démontré que la distance entre deux gènes sur un même chromosome est inversement proportionnelle à la fréquence des évènements de conversion entre les deux [147]. De plus, chez les souris et les rats, une étude a montré que les conversions géniques entre des paires de gènes situées sur les mêmes chromosomes sont 15 fois plus fréquentes que celles entre des paires provenant de chromosomes différents [57].

2.6 Homologie

Au sens large, l'homologie désigne une relation de parenté entre des structures ou des gènes, du fait que ceux-ci proviennent d'un ancêtre commun. Il ne faut pas confondre l'homologie avec l'analogie. Des structures anatomiques analogues peuvent avoir été créées par un phénomène d'évolution convergente, c'est-à-dire une évolution totalement indépendante qui s'est produite dans différentes lignées et qui a donné comme résultat des caractéristiques similaires chez différentes espèces non apparentées. La capacité de voler pour les chauves-souris et les oiseaux par exemple est apparue de manière indépendante dans chaque lignée. Toutefois, les membres antérieurs (qui ont divergé indépendamment pour devenir des ailes) des chauves-souris et des oiseaux sont homologues.

Il y a trois différents sous-types d'homologies : l'orthologie, la paralogie et la xénologie. La paralogie peut encore être sous-divisée en trois catégories : l'in-paralogie, l'out-paralogie et l'ohnologie. Regardons en détail ces différents types d'homologie de gènes dans les sections suivantes.

2.6.1 Orthologie

Deux gènes sont orthologues lorsque leur divergence date d'un évènement de spéciation. En d'autres mots, deux gènes sont orthologues si le dernier ancêtre commun des gènes correspond à un noeud de spéciation. Par exemple, dans la figure 2.2, le gène a_1 est orthologue aux gènes b_1, b_2, et c_1, tandis que le gène a_2 est orthologue aux gènes b_3, c_2, b_4 et c_3.

2.6.2 Paralogie

On qualifie deux gènes de paralogues lorsqu'ils ont commencé à diverger après un évènement de duplication, c'est-à-dire lorsque leur dernier ancêtre commun est un noeud de

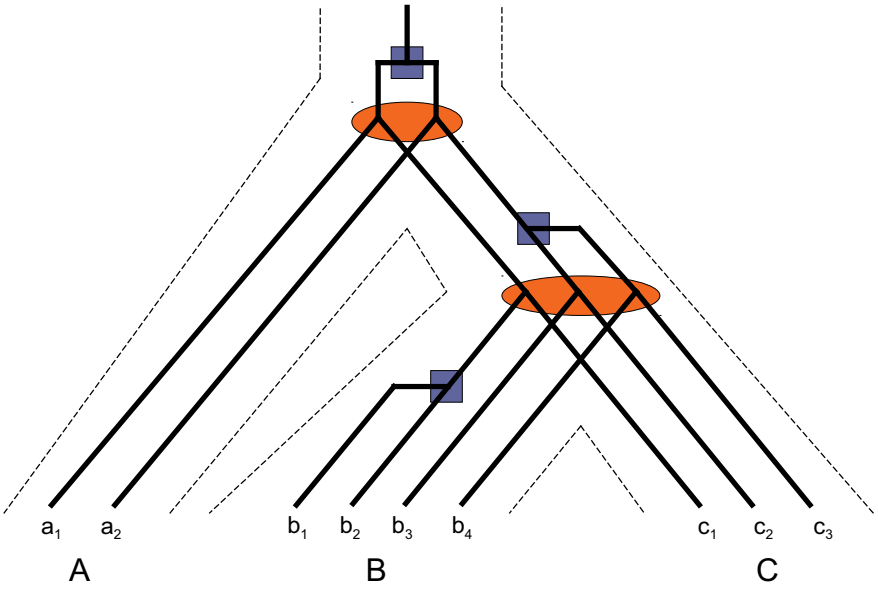

Figure 2.2 – Exemple d'histoire évolutive d'un gène ancestral jusqu'à trois espèces actuelles. Dans cette histoire, les espèces actuelles A, B et C possèdent respectivement 2 (a_1 et a_2), 4 (b_1, b_2, b_3 et b_4) et 3 (c_1, c_2 et c_3) copies du gène qui était présent dans l'ancêtre. Les noeuds de spéciation sont représentés par un cercle orange, tandis que les noeuds de duplication sont représentés par des carrés bleus.

duplication. Contrairement à une relation d'orthologie qui, par définition, ne peut s'appliquer qu'à des gènes qui sont dans des espèces différentes, une relation de paralogie peut exister entre deux gènes d'une même espèce ou de différentes espèces. Dans l'exemple de la figure 2.2, le gène a_1 est paralogue au gène a_2, mais il est également paralogue aux gènes b_3, c_2, b_4 et c_3.

La paralogie se sous-divise en trois catégories :

1. **In-paralogie**. Par rapport à une spéciation en particulier, on dit de deux gènes qu'ils sont in-paralogues s'ils ont été créés par un évènement de duplication survenu après cette spéciation. Dans la figure 2.2 par exemple, si on considère la spéciation du bas (le dernier ancêtre commun des espèces B et C), les gènes b_1 et b_2 sont in-

paralogues.

2. **Out-paralogie**. Toujours par rapport à une spéciation spécifique, deux gènes seront des out-paralogues dans le cas où ils ont été créés par une duplication qui s'est produite avant cette spéciation. Les gènes c_2 et c_3 de la figure 2.2 sont un exemple d'out-paralogie par rapport à la spéciation du bas.
3. **Onhologie**. Les gènes ohnologues sont des gènes paralogues provenant d'un évènement de duplication de génome entier. Cette nomenclature a été choisie en l'honneur du célèbre généticien Susumu Ohno.

2.6.3 Xénologie

Les xénologues sont des gènes homologues qui sont le résultat d'un évènement de transfert horizontal. Le transfert horizontal (ou transfert latéral) est un transfert d'un gène entre deux espèces ne s'étant pas produit par reproduction.

CHAPITRE 3

MÉTHODES D'INFÉRENCE

3.1 Introduction

Ce chapitre sert à introduire plus en détail les problèmes algorithmiques qui ont été étudiés dans le cadre de cette thèse et pour lesquels les chapitres 4 à 7 sont consacrés. La représentation des génomes, les arbres de gènes et d'espèces ainsi que la réconciliation sont à la base des méthodes présentées dans ce chapitre et sont définis à la section 3.2. Calculer la distance évolutive entre des paires de génomes est essentiel dans un grand nombre d'études de génomique comparative. Bien que les résultats de cette thèse ne soient pas directement reliés au calcul de distances, nous nous en servons dans nos inférences de génomes ancestraux. En particulier, au chapitre 5, notre méthode d'inférence de groupes ancestraux de GRT se base sur la distance d'inversions et pertes entre deux génomes. La section 3.3 présente les méthodes de distances les plus étudiées et les plus utilisées pour la comparaison de génomes représentés par des permutations sur un alphabet de gènes. Afin d'introduire les problèmes algorithmiques traités au chapitre 4, nous abordons à la section 3.4 le problème de l'inférence de génomes ancestraux dans le cas d'une évolution par duplication de génomes entiers (DGE). Plus précisément, le problème d'inférer le génome ancestral tel qu'il était tout de suite avant la duplication du génome et celui du calcul de distances entre deux génomes qui ont été dupliqués sont présentés. En introduction au chapitre 5 portant sur l'inférence évolutive de groupes de gènes dupliqués en tandem, nous présentons à la section 3.5 un état de l'art sur les méthodes algorithmiques dédiées à ce problème. La section 3.6 porte sur les approches existantes pour l'inférence de relations d'in-paralogie entre les gènes, qui est le problème abordé au chapitre 6. Finalement, la

section 3.7 introduit le chapitre 7 dédié à l'étude évolutive du répertoire de gènes d'ARN de transfert des génomes.

3.2 Définitions

3.2.1 Représentation d'un génome

Les génomes peuvent posséder un seul ou plusieurs chromosomes : on les qualifie respectivement d'*unichromosomiques* et *multichromosomiques*. Chaque chromosome, qui contient un ensemble de gènes, peut être représenté par une chaîne de caractères sur un alphabet Σ, où chaque caractère $c \in \Sigma$ représente un gène. On représente l'orientation des gènes sur le chromosome (correspondant à l'orientation de sa transcription) par des signes ($+$ ou $-$; on omet habituellement le signe $+$). L'inversion d'un segment de chromosome représenté par la chaîne de caractères $X = x_1 x_2 \dots x_r$ donne le segment $-X = -x_r - x_{r-1} \cdots - x_1$. Un chromosome entier est équivalent à son inverse. Un chromosome *circulaire* est une chaîne de caractères $x_1 x_2 \dots x_r$ où l'on considère que le gène x_1 est à la suite du gène x_r. Un chromosome qui n'est pas circulaire est *linéaire*. Les génomes unichromosomiques possèdent habituellement un chromosome circulaire, tandis que les génomes multichromosomiques possèdent habituellement des chromosomes linéaires.

3.2.2 Arbre de gènes et arbre d'espèces

Un arbre de gènes est un arbre binaire enraciné qui représente les relations évolutives inférées entre les gènes. Chaque feuille de l'arbre de gènes est étiquetée et chaque étiquette représente un gène. Les noeuds internes de l'arbre représentent des gènes ancestraux et la racine représente l'ancêtre commun à l'origine de tous les gènes. Il existe plusieurs approches pour l'inférence d'un arbre de gènes à partir d'un alignement multiple des séquences en nucléotides ou en acides aminés : des méthodes de parcimonie, de dis-

tance, de maximum de vraisemblance et des méthodes bayésiennes. Une revue exhaustive de ces méthodes est faite dans [59].

Un arbre d'espèces, quant à lui, représente l'histoire évolutive des espèces considérées. Il s'agit d'un arbre binaire et enraciné dans lequel chaque feuille représente une espèce actuelle et chaque noeud interne représente une espèce ancestrale. On utilise habituellement un ensemble de gènes orthologues chez toutes les espèces étudiées pour inférer les arbres des espèces.

Notons que les arbres de gènes et d'espèces peuvent être non binaires. Toutefois, dans le cadre de cette thèse, ces cas particuliers ne sont pas considérés.

3.2.3 Réconciliation d'arbres

La réconciliation d'un arbre de gènes avec un arbre d'espèces a été introduite par Goodman en 1979 [71] et largement étudiée par la suite [1, 9, 26, 30, 33, 43, 44, 55, 75, 76, 111, 128–130, 186, 195]. Il s'agit d'une méthode qui revient à "emboîter" l'arbre de gènes dans l'arbre des espèces et à en déduire une histoire de duplications et pertes pour la famille de gènes. En particulier, la réconciliation permet d'inférer le nombre de copies de gènes dans les génomes ancestraux (représentant les noeuds internes de l'arbre d'espèces). Un exemple de réconciliation est présenté à la figure 3.1.

3.3 Calcul de distances entre génomes

Avec l'apparition des cartes génétiques (la première carte génétique a été produite par Sturtevant pour le chromosome X de la drosophile en 1913 [157]), associant des marqueurs génétiques à leur position sur les chromosomes, et plus tard des séquences complètes de génomes pour un nombre sans cesse grandissant d'espèces, il était essentiel de développer des méthodes pour mesurer la distance entre les génomes. Obtenir une mesure

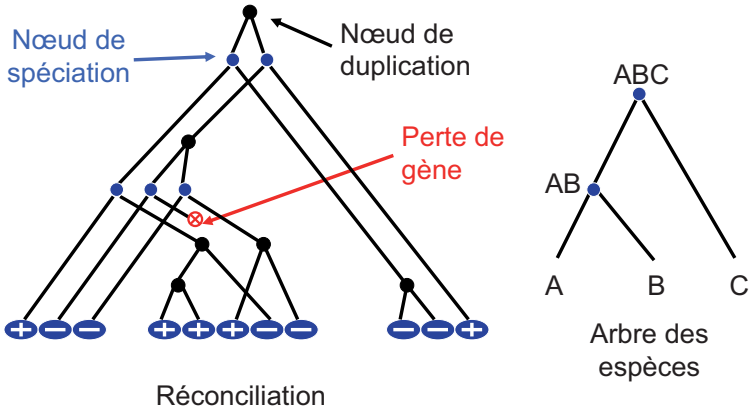

Figure 3.1 – Exemple d'une réconciliation. Les gènes de chaque espèce actuelle sont représentés par des signes qui indiquent leur orientation. Une réconciliation d'un arbre de gènes avec un arbre des espèces permet d'identifier dans l'arbre de gènes les noeuds de duplication (en noir), les noeuds de spéciation (en bleu) ainsi que des pertes de gènes (en rouge). Gauche : Un arbre réconcilié. Droite : L'arbre des espèces correspondant à l'arbre réconcilié. Les feuilles de l'arbre des espèces (A, B et C) représentent les génomes actuels, alors que les noeuds internes (AB et ABC) représentent des génomes ancestraux.

de la distance entre des génomes est nécessaire pour la résolution de plusieurs problèmes d'inférence, comme par exemple celui de la reconstruction de génomes ancestraux. En particulier, le problème de la petite phylogénie (*small phylogeny problem*) consiste à reconstruire tous les génomes ancestraux d'une phylogénie donnée en minimisant le nombre total d'évènements évolutifs. Les approches développées pour résoudre le problème de la petite phylogénie nécessitent de pouvoir mesurer la distance évolutive sur chaque branche de l'arbre.

Dans les sections suivantes, les distances de réarrangements les plus utilisées sont introduites. Les résultats algorithmiques classiques pour le calcul de ces distances entre deux génomes G_1 et G_2, représentés par des permutations signées sur un alphabet Σ, sont présentés.

3.3.1 Distance de points de cassure

La notion de point de cassure (*breakpoint*) dans un génome a été introduite pour la première fois en 1938 par Dobzhansky et Sturtevant, dans un article portant sur les inversions dans les chromosomes d'une espèce de drosophile (*D. pseudoobscura*) [42]. Lorsque l'on compare les génomes (plus précisément, les ordres des gènes) chez deux espèces, un point de cassure représente un endroit où une adjacence est brisée, c'est-à-dire qu'elle est présente dans une des espèces mais pas dans l'autre.

La distance de points de cassure a été tout d'abord étudiée dans le cas de génomes circulaires unichromosomiques [143, 144, 172]. Dans la figure 3.2, on présente la comparaison de deux génomes pour lesquels on considère ou pas l'orientation des gènes. Étant donné que les génomes sont circulaires, on doit prendre en compte l'adjacence entre le dernier gène et le premier sur la représentation linéaire. La distance de points de cassure correspond au nombre de points de cassure dans un des deux génomes comparés (ce nombre est toujours identique pour les deux génomes). Dans l'exemple de la figure 3.2 a), cette distance est égale à 2. En effet, lorsque l'on ne considère pas l'orientation des gènes, les adjacences sont commutatives dans le contexte d'évènements d'inversion : l'adjacence *fa* dans le génome G_1 est équivalente à l'adjacence *af* dans le génome G_2 de la figure 3.2 a). Pour l'exemple de la figure 3.2 b), dans lequel on a ajouté les signes des gènes, la distance de points de cassure est égale à 3. Ceci est dû au fait que l'adjacence *fa* du génome G_1 n'est pas équivalente à l'adjacence *a-f* du génome G_2.

Dans le cas de génomes linéaires multichromosomiques, il faut considérer les adjacences télomériques (adjacences avec un bout de chromosome) en plus des adjacences entre les gènes pour calculer la distance de points de cassure [133, 159]. La formule suivante, proposée dans [159], donne alors cette distance (d_{BP}) :

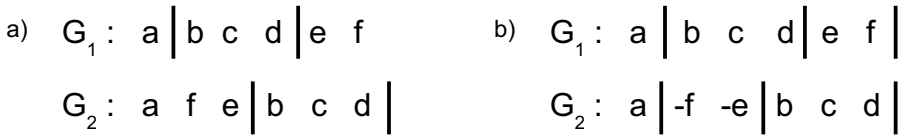

Figure 3.2 – Exemples de points de cassure entre deux génomes circulaires (linéarisés) G_1 et G_2. a) Dans cet exemple, on ne considère pas l'orientation des gènes. On compte 2 points de cassure entre les génomes. b) En ajoutant l'orientation des gènes dans les génomes de l'exemple a), on compte maintenant 3 points de cassure.

$$d_{BP}(G_1, G_2) = n - a(G_1, G_2) - e(G_1, G_2)/2$$

où n est égal au nombre de gènes dans chacun des deux génomes (en d'autres mots, $n = |\Sigma|$), $a(G_1, G_2)$ représente le nombre d'adjacences communes entre les gènes et $e(G_1, G_2)$ est égal au nombre de télomères communs dans les deux génomes. La figure 3.3 présente un exemple de deux génomes de 6 gènes répartis sur différents chromosomes (lignes). Il n'y a qu'une seule adjacence commune entre G_1 et G_2 (ab) et quatre télomères communs (le télomère avant a, celui après c, celui avant d et celui après f). La distance de points de cassure entre ces deux génomes est donc égale à 3 $(6 - 1 - 4/2)$.

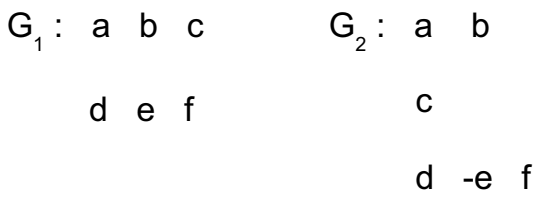

Figure 3.3 – Exemples de deux génomes multichromosomiques linéaires G_1 et G_2 orientés. Chaque ligne représente un chromosome différent.

En résumé, la distance de points de cassure se calcule facilement en temps linéaire. De ce fait, elle est beaucoup utilisée en génomique comparative. L'inconvénient de cette

distance est qu'elle ne permet pas de retracer l'histoire des évènements ayant provoqué les cassures. Dans la suite, d'autres distances en lien direct avec les évènements de réarrangement sont présentées. Le graphe de points de cassure, utilisé pour calculer les distances de réarrangements, est tout d'abord introduit.

3.3.2 Graphe de points de cassure

Les deux algorithmes décrits dans les sections suivantes (sections 3.3.3 et 3.3.4) nécessitent la construction d'un *graphe de points de cassure* représentant les génomes G_1 et G_2. L'analyse des propriétés du graphe (nombre de cycles, nombre de chemins, présence de certains types de cycles, etc.) permet ensuite de faire les calculs de distances.

Dans un graphe de points de cassure, on représente chaque gène une seule fois. Afin de permettre la représentation de l'orientation d'un gène, chaque gène est représenté par deux sommets : un sommet pour la queue (étiqueté avec *t* pour *tail*) et un sommet pour la tête (étiqueté avec *h* pour *head*). Ainsi, un gène a dans l'orientation positive sera représenté par deux sommets a^t et a^h (la queue du gène suivie par la tête de gauche à droite). Si le gène a est dans l'orientation inverse, on le représente par les sommets a^h et a^t.

On représente ensuite les adjacences des deux génomes dans le même graphe en utilisant deux types d'arêtes (on utilise habituellement deux couleurs différentes). Dans cette thèse, nous utiliserons des arêtes noires (traits gras) pour les adjacences d'un génome et grises (traits minces) pour celles de l'autre.

Dans le cas de génomes circulaires, chaque sommet (représentant un bout de gène) est nécessairement adjacent à deux autres sommets (un dans chaque génome). Dans le cas de génomes linéaires multichromosomiques, comme dans le cas de la distance de points de cassure, il faut tenir compte des adjacences aux télomères. En fait, on peut tout simplement ne pas avoir de sommets dans le graphe pour les télomères. Dans ce cas, un sommet qui

est au bout d'un chromosome dans les deux génomes n'aura aucune arête, un sommet qui est au bout d'un chromosome dans un seul des deux génomes aura une seule arête et tous les autres sommets auront deux arêtes.

Tous les chemins et les cycles du graphe de points de cassure sont alternés, c'est-à-dire que deux arêtes consécutives sont de couleurs différentes. Un sommet qui n'a aucune arête est considéré comme un chemin de taille 0.

La figure 3.4 présente le graphe de points de cassure correspondant aux génomes linéaires multichromosomiques de la figure 3.3.

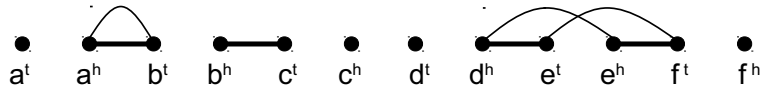

Figure 3.4 – Graphe de points de cassure pour l'exemple de la figure 3.3. Les arêtes noires représentent les adjacences du génome G_1, alors que les arêtes grises (arêtes courbées) représentent les adjacences du génome G_2.

3.3.3 Distance de réarrangements

Le premier algorithme polynomial pour le calcul de la distance de réarrangements (plus précisément, la distance d'inversions et translocations) a été développé par Hannenhalli et Pevzner [80]. Par la suite, des algorithmes linéaires ont été développés pour calculer la distance en inversions seulement [10, 16] et translocations seulement [100]. L'algorithme de Hannenhalli et Pevzner a ensuite été reformulé en 2002 par Tesler [160] pour corriger certains détails et améliorer la vitesse d'exécution en y intégrant l'algorithme de Bader *et al.* [10] (qui permet de calculer la distance d'inversions en temps linéaire).

Comme mentionné plus haut, l'approche utilisée pour calculer la distance de réarrangements entre deux génomes nécessite d'abord la construction d'un graphe de points de cassure. La formule suivante donne alors la distance de réarrangements (d_R) entre les gé-

nomes G_1 et G_2 :

$$d_R(G_1,G_2) = n + N - (c(G_1,G_2) + p(G_1,G_2) - p_{G_1G_1}) + h(G_1,G_2)$$

où n est le nombre de gènes, N est le nombre de chromosomes du génome G_1 (0 dans le cas circulaire), $c(G_1,G_2)$ est le nombre de cycles, $p(G_1,G_2)$ est le nombre de chemins (0 dans le cas circulaire) et $p_{G_1G_1}$ est le nombre de chemins qui relient deux extrémités du génome G_1. Le paramètre $h(G_1,G_2)$ représente le nombre de composantes particulières du graphe : les obstacles (*hurdles*) et les forteresses (*fortresses*). Ces composantes ne seront pas définies ici, car elles sont relativement rares. Une description détaillée se trouve dans l'article de Tesler [160].

La distance de réarrangements pour l'exemple de la figure 3.4 est donc égale à 2 ($6+2-2-5+1$). En effet, il faut compter chaque sommet qui n'a pas d'arête comme un chemin (de taille 0) dans le graphe.

3.3.4 Distance DCJ

La distance DCJ (*double cut and join*) a été introduite en 2005 par Yancopoulos *et al.* [181] et reformulée ensuite par Bergeron *et al.* [17, 18]. Cette distance permet de considérer, en plus des inversions et des translocations, les transpositions et les échanges de blocs. Le calcul de cette distance se fait en temps linéaire avec des algorithmes beaucoup plus simples (qui nécessitent moins de prétraitements et moins de paramètres dans la formule) que ceux de la distance de réarrangements.

Une opération de DCJ correspond à faire d'abord deux coupures dans un génome et ensuite à recoller les 4 bouts entre eux, de n'importe quelle façon. Afin de reproduire un évènement de transposition ou d'échange de blocs avec le modèle DCJ, il faut faire deux opérations DCJ consécutives en passant par un génome intermédiaire circulaire (ce

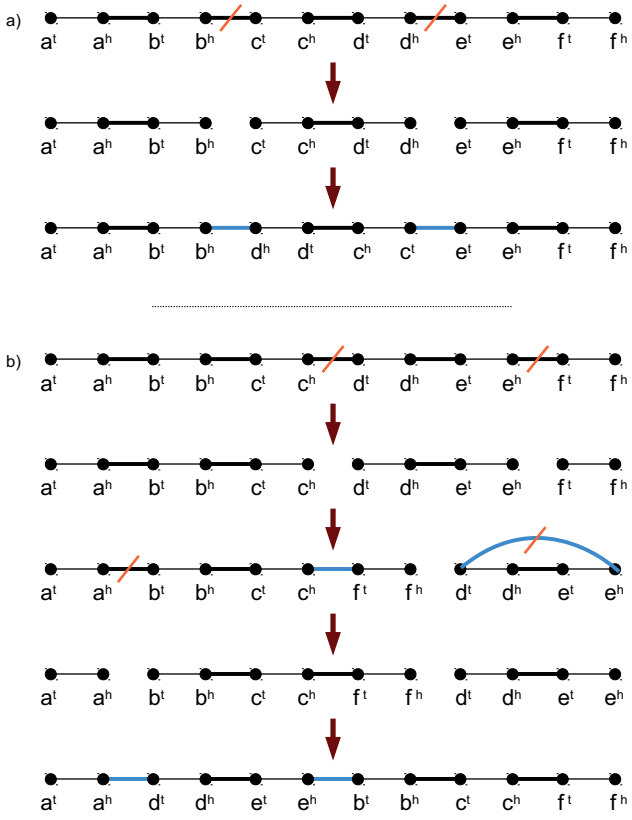

Figure 3.5 – Exemples de réarrangements génomiques modélisés par des opérations DCJ. Les coupures sont représentées par des barres obliques oranges, tandis que les nouveaux liens sont dessinés en bleu. a) Une opération DCJ correspondant à une inversion du segment cd. Le génome de départ, $abcdef$, devient $ab - d - cef$. b) Deux opérations DCJ consécutives qui représentent une transposition du segment de. Le génome de départ $abcdef$ devient $adebcf$.

qui n'est toutefois pas biologiquement réaliste). Deux exemples de réarrangements génomiques modélisés par des opérations DCJ sont présentés à la figure 3.5.

La distance DCJ peut également être calculée à l'aide d'un graphe de points de cassure (voir section 3.3.2). La formule suivante permet de calculer cette distance (d_{DCJ}) pour les génomes G_1 et G_2 :

$$d_{DCJ}(G_1, G_2) = n - c(G_1, G_2) - p_{even}/2$$

où n est le nombre de gènes, $c(G_1, G_2)$ est le nombre de cycles alternés et p_{even} est le nombre de chemins alternés avec un nombre pair d'arêtes.

Dans l'exemple de la figure 3.4, la distance DCJ est égale à 2 $(6-2-4/2)$.

3.4 Calcul de distances entre des génomes ayant été doublés

Les méthodes présentées dans la section 3.3 permettaient de calculer des distances entre des génomes possédant le même contenu en gènes ainsi qu'une seule copie de chaque gène. Toutefois, il a été démontré que plusieurs génomes actuels ont subi, durant leur évolution, des évènements de DGE (voir section 2.2.1). Immédiatement après une DGE, un génome possède exactement deux copies de chaque chromosome et donc deux copies de chaque gène : un tel génome sera qualifié de *parfaitement dupliqué*. Un génome ancestral parfaitement dupliqué qui est affecté par des réarrangements génomiques durant son évolution donnera un génome qu'on qualifie de *réarrangé dupliqué*.

Dans le contexte du problème de la petite phylogénie (mentionné dans la section 3.3), de nouvelles méthodes sont donc nécessaires pour (1) reconstruire des génomes ancestraux pré-dupliqués à partir de génomes actuels réarrangés dupliqués et (2) calculer des distances entre des génomes parfaitement dupliqués et des génomes réarrangés dupliqués. Le premier problème, appelé le problème du *genome halving*, est présenté dans la sec-

tion 3.4.1, alors que le deuxième, qu'on nomme le problème de la double distance, est décrit dans la section 3.4.2.

3.4.1 Inférence d'un génome pré-dupliqué

Le problème de l'inférence d'un génome pré-dupliqué, appelé *genome halving*, correspond à inférer un génome pré-dupliqué (immédiatement avant la DGE) qui minimise la distance évolutive avec un génome connu réarrangé dupliqué (habituellement un génome actuel). Notons que ce problème est clairement équivalent à celui d'inférer un génome parfaitement dupliqué. Un algorithme exact a été développé par El-Mabrouk et Sankoff [51] pour résoudre ce problème en temps linéaire en utilisant la distance de réarrangements. Des algorithmes qui utilisent la distance DCJ [118, 171] et la distance de points de cassure [159] ont également été développés.

Afin de réduire l'ensemble de solutions optimales, il est possible d'utiliser un troisième génome (n'ayant pas été doublé) comme groupe externe. Soit le génome réarrangé dupliqué G et le génome externe O. Ce problème, nommé *guided genome halving*, consiste à trouver le génome pré-dupliqué M qui minimise la somme des distances $d(G,M)+d(M,O)$. Le problème du *guided genome halving* est NP-difficile dans le cas de la distance de points de cassure [193] et NP-complet dans le cas de la distance DCJ [159] (complexité inconnue pour la distance de réarrangements). Quelques heuristiques ont été développées dans les dernières années pour résoudre ce problème [68, 190, 191, 193].

Toutefois, il est rare qu'un génome, après avoir été doublé par une DGE, conserve les deux copies de chaque gène (voir section 2.2.1). La plupart des génomes actuels ont été réarrangés et ont perdu des copies de gènes depuis l'évènement de DGE : on nomme ces génomes des *génomes réarrangés dupliqués avec pertes*. Dans le cadre du premier travail de cette thèse, présenté dans le chapitre 4, nous avons généralisé l'algorithme d'inférence

de génome pré-dupliqué d'El-Mabrouk et Sankoff [51] afin qu'il puisse s'appliquer à un génome réarrangé dupliqué avec pertes.

3.4.2 Double distance

La distance entre un génome parfaitement dupliqué et un génome dupliqué réarrangé se nomme *double distance*. Le problème du calcul de la double distance a été prouvé polynomial dans le cas de la distance de points de cassure, mais NP-difficile pour la distance DCJ [159]. La complexité du calcul de la double distance est encore inconnue pour la distance de réarrangements.

La difficulté de ce problème réside dans le fait que la relation d'orthologie entre les copies de gènes des deux génomes est inconnue (voir la figure 3.6 pour un exemple). En apposant préalablement des étiquettes (indiquant que le gène est la copie 1 ou 2) sur le génome réarrangé dupliqué, le problème devient équivalent à celui de trouver un étiquetage des gènes du génome parfaitement dupliqué qui minimise la distance entre les deux génomes. Lorsque l'on s'intéresse à minimiser la distance de réarrangements ou la distance DCJ, on peut encore reformuler le problème d'une autre manière : en construisant d'abord le graphe de points de cassure pour les adjacences du génome réarrangé dupliqué (le génome G dans la figure 3.6), l'objectif est de trouver les adjacences du génome parfaitement dupliqué (le génome D dans la figure 3.6) qui maximiseront le nombre de cycles dans le graphe de points de cassure. En effet, plus on forme de cycles dans le graphe de points de cassure, plus on risque d'obtenir une petite distance. Nous avons nommé cette reformulation du problème : le problème de la double distance faible (*weak double distance problem*). Nous avons présenté une heuristique pour résoudre ce problème dans le cas de la distance de réarrangements et celui de la distance DCJ également dans le cadre du premier travail de cette thèse (voir le chapitre 4).

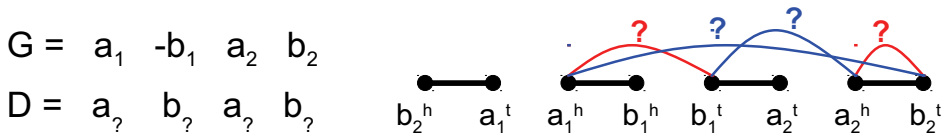

Figure 3.6 – Illustration du problème de la correspondance entre les copies de gènes dans le calcul de la double distance. Dans cet exemple, les deux génomes sont circulaires unichromosomiques. Le génome G est un génome réarrangé dupliqué, tandis que le génome D est un génome parfaitement dupliqué. En ne sachant pas quelle copie du génome G correspond à quelle copie du génome D, il y a deux paires d'arêtes possibles (en rouge et en bleu) pour représenter chaque paire d'adjacences dans le graphe de points de cassure.

Comme mentionné plus haut, il est rare qu'un génome qui a été doublé subisse seulement des réarrangements et ne perde aucune de ses deux copies de gènes. Il est donc nécessaire de pouvoir calculer la double distance dans le cas où le génome actuel n'est pas seulement dupliqué et réarrangé, mais possède au plus deux copies de chaque gène. Pour résoudre cette version du problème, il faut trouver le meilleur endroit pour réinsérer les gènes perdus dans le génome réarrangé dupliqué avec pertes en plus de trouver un étiquetage des gènes qui minimisera la distance. Nommons un génome réarrangé dupliqué avec pertes dans lequel on a réinséré les copies de gènes perdues une *extension*. Une *extension optimale* est alors une extension qui minimise la distance à un génome parfaitement dupliqué. Nous avons prouvé qu'il existe toujours une extension optimale dans laquelle les gènes réinsérés préservent une des adjacences (gauche ou droite) du génome parfaitement dupliqué. Comme démontré dans la figure 3.7, cela laisse au maximum 4 endroits possibles pour réinsérer les copies perdues. Toujours dans le cadre du premier projet de cette thèse, une heuristique pour le calcul de la double distance avec pertes est présentée dans le chapitre 4 pour la distance de réarrangements et la distance DCJ.

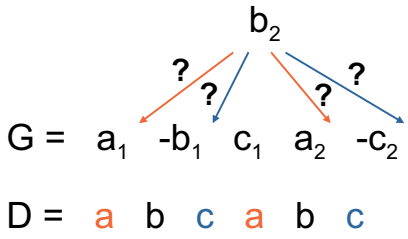

Figure 3.7 – Possibilités de réinsertion d'un gène perdu pour obtenir une extension optimale. Le génome G est un génome réarrangé dupliqué avec pertes et le génome D est un génome parfaitement dupliqué. Le gène b_2 a été perdu et doit être réinséré. Afin de préserver une adjacence gauche (en orange) ou une adjacence droite (en bleu) du génome D, 4 endroits sont possibles pour la réinsertion du gène b_2.

3.5 Inférence d'histoires évolutives de groupes de gènes dupliqués en tandem

L'étude de l'évolution des groupes de GRT (voir la section 2.2.2) a reçu beaucoup d'attention dans la dernière décennie. En effet, plusieurs méthodes d'inférence d'histoires évolutives spécifiques aux groupes de GRT ont été développées avec comme objectifs de permettre d'identifier la fonction spécifique de chaque copie de gène, de mieux comprendre les mécanismes d'amplification des GRT et d'analyser le nombre et la taille des évènements évolutifs qui agissent sur ces familles de gènes.

Dans les sections suivantes, un résumé des différentes approches d'inférence développées pour analyser l'évolution des groupes de GRT est présenté.

3.5.1 Le modèle de duplications en tandem

En 1977, Fitch [61] a introduit le modèle évolutif de duplications en tandem, qui propose qu'un unique gène ancestral peut donner naissance à un groupe de GRT par une série de duplications plaçant les nouvelles copies d'un gène à côté de la copie d'origine. Ces duplications en tandem peuvent être *simples* (duplication d'un seul gène) ou *multiples* (duplications simultanées de gènes adjacents). La figure 3.8 montre un exemple d'un groupe de GRT ayant évolué selon ce modèle.

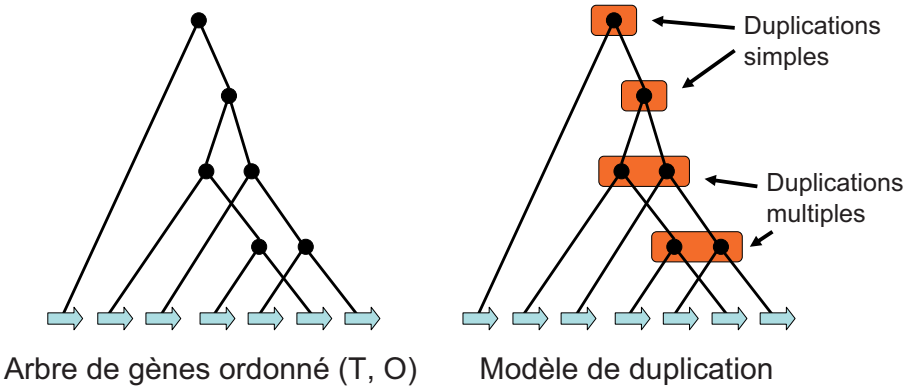

Figure 3.8 – Exemple d'un groupe de GRT ayant évolué selon le modèle de duplication en tandem de Fitch. Les gènes (feuilles de l'arbre), tous descendants d'un même gène ancestral, sont ici représentés par des flèches qui indiquent leur orientation. Gauche : Un arbre de gènes ordonné (T, O), où T est l'arbre de gènes et O est l'ordre des feuilles de l'arbre, représentant l'ordre des gènes sur le chromosome. Droite : Un modèle de duplication est un arbre de gènes ordonné dans lequel on a identifié les noeuds de duplications simples et multiples.

Plusieurs des méthodes présentées dans les sections suivantes sont basées sur l'analyse d'un *arbre de gènes ordonné* (T, O), où T est l'arbre de gènes et O est l'ordre des feuilles de l'arbre, représentant l'ordre et l'orientation des gènes sur le chromosome. En d'autres mots, dans un arbre de gènes ordonné, les feuilles de l'arbre sont placées de gauche à droite dans le même ordre que les gènes sur le chromosome. Un exemple d'arbre de gènes ordonné est présenté à la figure 3.8-gauche. Un arbre de gènes ordonné qui peut induire une histoire de duplication est appelé un *arbre de duplication*. Enfin, un *modèle de duplication* (voir la figure 3.8-droite) est tout simplement un arbre de gènes ordonné dans lequel on a identifié les noeuds de duplications simples et multiples, ce qui permet d'illustrer une histoire de duplication.

3.5.2 Inférence d'histoires de duplication en tandem

À partir du modèle de duplications en tandem de Fitch, plusieurs études ont considéré le problème de la reconstruction d'une histoire de duplication en tandem d'un groupe de

GRT [20, 53, 158, 185].

Tang *et al.* [158] ont décrit un algorithme (WINDOW) pour la reconstruction de ce qu'ils nomment un modèle de duplication (figure 3.8-droite). Cet algorithme identifie les évènements de duplication du présent vers le passé en cherchant à chaque étape une paire de fenêtres adjacentes de coût minimal, où chacune des deux fenêtres (de même taille) représente une moitié des gènes obtenus après une duplication en tandem. Le coût d'une paire de fenêtres est la moyenne des distances des paires de gènes correspondants dans chaque fenêtre. Cette méthode est assurée de toujours retrouver une histoire de duplication en tandem, mais ne permet pas de vérifier si le groupe de gènes a effectivement évolué selon ce modèle évolutif.

Elemento *et al.* [53] ont d'abord développé un algorithme permettant de vérifier si un arbre de gènes ordonné correspond à une histoire de duplication en tandem. Ils ont ensuite proposé deux façons de reconstruire des histoires de duplication parcimonieuses : (1) utiliser un programme de reconstruction phylogénétique et vérifier par la suite si les arbres obtenus sont des arbres de duplication et (2) explorer toutes les histoires de duplication possibles pour un nombre donné de gènes et conserver uniquement les arbres de duplication les plus parcimonieux. La première approche ne garantit pas de trouver une histoire de duplication puisqu'il est possible qu'aucune des phylogénies inférées ne corresponde à un arbre de duplication. La seconde méthode permet toujours de trouver une histoire de duplication optimale, mais elle nécessite une exploration exhaustive de toutes les histoires possibles.

Zhang *et al.* [185] ont proposé un algorithme linéaire pour la reconstruction d'un modèle de duplication à partir d'une phylogénie, s'il existe. Les auteurs utilisent les propriétés des phylogénies associées à des modèles de duplication (qui sont un peu similaires aux propriétés des arbres binaires de recherche) afin d'identifier les ensembles de noeuds internes d'une phylogénie qui représentent des duplications multiples (les noeuds restants

représentent des duplications simples).

Bertrand et Gascuel [20] ont décrit un ensemble de réarrangements topologiques qui permettent, à partir d'un arbre de duplication, de parcourir tout l'espace des arbres de duplication. Ils ont ensuite défini une méthode de recherche locale qui permet, en explorant tout l'espace des arbres de duplication produits initialement par d'autres méthodes, d'améliorer les résultats de ces autres méthodes, c'est-à-dire d'identifier des arbres de duplication plus parcimonieux.

Toutefois, à cause des réarrangements et des pertes qui affectent aussi les groupes de GRT, il est souvent impossible de reconstruire une histoire évolutive pour une famille de GRT en ne considérant que les duplications [67].

3.5.3 Intégration des inversions

Une généralisation du modèle de duplications en tandem de Fitch, permettant les inversions, a donc été considérée dans notre laboratoire [98]. Afin de simplifier le problème, les auteurs se sont limités à l'inférence de duplications en tandem simples. Le problème peut alors être reformulé de la sorte : à partir d'un arbre de gènes ordonné (T, O), trouver un ordre O' en appliquant un minimum d'inversions tel que l'arbre de gènes ordonné (T, O') soit un arbre de duplication. Un algorithme exact ainsi qu'une heuristique basée sur la distance de points de cassure ont été développés. Il a été démontré par des expériences sur des données simulées que l'heuristique, qui est beaucoup plus rapide que l'algorithme exact, donne une bonne approximation de la distance d'inversions lorsque le nombre d'inversions n'est pas trop élevé.

Ce modèle a ensuite été étendu à l'étude de groupes de GRT orthologues chez différentes espèces [21]. Cette extension prend en entrée un arbre de gènes représentant tous les gènes étudiés chez toutes les espèces, ainsi que la phylogénie de ces espèces. La première

étape de l'approche consiste à faire une réconciliation de l'arbre de gènes avec l'arbre des espèces (un exemple d'arbre de gènes réconcilié est présenté à la figure 3.1). Il suffit ensuite de trouver les ordres de gènes ancestraux qui minimisent la somme des inversions nécessaires sur chaque branche de l'arbre des espèces. Encore une fois, en utilisant les propriétés des arbres de duplication simple, les auteurs ont pu développer un algorithme exact pour solutionner ce problème. Les avantages de cette extension, par rapport à la méthode précédente, sont que des pertes de gènes peuvent être inférées à l'aide de la réconciliation et que les inversions peuvent être inférées avec plus de précision en considérant l'information provenant des différentes espèces.

3.5.4 Analyses basées sur les dot-plots

Un autre type d'approche, basé sur l'analyse d'un dot-plot de l'alignement d'un groupe de GRT avec lui-même (dans le cas d'une étude chez une seule espèce), ou d'un dot-plot de l'alignement de groupes de GRT orthologues (plusieurs espèces), a été considéré pour reconstruire des histoires évolutives pour des groupes de GRT. Une série d'articles a récemment été publiée par le même groupe de chercheurs sur ces méthodes [153, 187–189]. Dans tous ces travaux, le même prétraitement est utilisé. Tout d'abord, BLASTZ [149] est employé pour obtenir les alignements des groupes de GRT. Un dot-plot des alignements est produit et il est ensuite traité afin de satisfaire la propriété de la fermeture transitive (si un alignement existe entre les régions A et B ainsi qu'entre les régions B et C, alors il doit y avoir un alignement entre les régions A et C également). Les alignements sont par la suite étendus au maximum et chaînés (c'est-à-dire que deux alignements proches peuvent être reliés). Enfin, les extrémités des alignements sont utilisées pour définir les *segments atomiques*. Les histoires évolutives sont inférées du présent vers le passé et, par exemple, un évènement de duplication est inféré en identifiant deux segments atomiques alignés et

en retirant un des deux segments du dot-plot.

Le premier article de cette série [188] s'intéresse à l'inférence d'une histoire évolutive d'un groupe de GRT chez une seule espèce. Un premier algorithme pour la reconstruction d'histoires évolutives parcimonieuses de duplications (en tandem ou non et inversées ou non) est d'abord décrit. Les auteurs proposent ensuite un algorithme stochastique, basé sur une méthode de Monte Carlo, pour la reconstruction d'histoires évolutives avec délétions (en plus des duplications) qui permet d'obtenir des solutions sous-optimales tout en favorisant les histoires avec moins d'évènements évolutifs. Seules les délétions qui permettent d'inférer ultérieurement des duplications sont considérées.

L'article suivant [187] décrit une extension des algorithmes précédents à l'étude de groupes orthologues de GRT chez plusieurs espèces. Dans le cas d'une étude chez deux espèces par exemple, les séquences des régions analysées chez les deux espèces sont tout simplement placées une à la suite de l'autre avant le prétraitement. Des séquences orthologues sont identifiées à partir des alignements et les évènements évolutifs sont ensuite inférés jusqu'à ce qu'il ne reste que des séquences orthologues chez les deux espèces. C'est alors qu'un évènement de spéciation est inféré et la séquence complète d'une espèce est retirée. L'algorithme poursuit ensuite l'inférence des évènements s'étant produits avant la spéciation.

Dans [153], les auteurs proposent un algorithme combinatoire pour la reconstruction d'histoires évolutives avec duplications (en tandem ou non et inversées ou non) et délétions. Cette nouvelle méthode de parcimonie utilise un *graphe de contraintes* afin d'identifier des priorités sur l'ordre d'inférence des évènements (par exemple, une duplication qui s'insère à l'intérieur d'une autre région dupliquée doit être inférée en premier).

L'avantage de l'approche des dot-plots est qu'elle permet d'utiliser de l'information provenant des régions non fonctionnelles, contrairement aux approches basées sur les arbres de gènes ordonnés. Toutefois, les traces des évènements évolutifs dans les régions

non fonctionnelles sont effacées rapidement, car elles sont continuellement affectées par des mutations. De plus, l'efficacité de ces méthodes dépend beaucoup de la qualité du prétraitement et les auteurs ne donnent aucune information sur le fonctionnement et la performance des algorithmes qui appliquent la fermeture transitive et le chaînage des alignements dans les dot-plots, qui sont loin d'être des problèmes triviaux.

3.5.5 DILTAG

Dans le cas des deux méthodes citées dans la section 3.5.3, seules les duplications simples étaient considérées. Cette hypothèse, qui permettait d'avoir des solutions algorithmiques exactes, limitait l'application de ces algorithmes à quelques groupes de GRT qui avaient selon toute évidence évolué de cette façon. C'est pour cette raison qu'une heuristique plus générale, l'algorithme DILTAG [97], a été développée afin d'inférer un ensemble d'histoires évolutives optimales pour un groupe de GRT chez une seule espèce, en fonction d'un modèle de coûts permettant des duplications de taille variable, en tandem ou inversées, des délétions et des inversions. Les expériences sur des jeux de données simulées ont montré que les évènements évolutifs récents peuvent être inférés assez précisément par cet algorithme lorsque les arbres de gènes sont exacts. Malgré l'incertitude associée aux évènements plus anciens, la distribution des tailles des duplications était inférée avec une certaine précision.

Une limitation évidente de DILTAG réside dans le fait qu'il ne peut être appliqué qu'à un unique groupe de GRT. La suite logique était donc de développer une extension à plusieurs espèces, puisque la génomique comparative est une approche plus appropriée pour inférer des pertes de gènes et des inversions. Le développement de cet algorithme, nommé Multi-DILTAG, constitue le deuxième travail de cette thèse et fait l'objet du chapitre 5.

3.6 Inférence de gènes in-paralogues

Malgré le fait que plusieurs algorithmes ont été développés pour l'inférence de gènes orthologues (des revues de ces méthodes sont présentées dans [31, 58]), l'inférence de gènes in-paralogues a reçu relativement peu d'attention. L'identification de relations d'in-paralogie entre les gènes est pourtant très importante, car elle permet notamment d'étudier les évènements de duplication. En effet, lorsque l'on considère des spéciations récentes (l'in-paralogie est définie par rapport à un certaine spéciation ; voir la section 2.6.2), les gènes in-paralogues sont conséquemment le résultat de duplications récentes. On peut alors supposer qu'aucun évènement de réarrangement génomique n'est venu perturber l'emplacement de ces copies dupliquées dans le génome et ainsi analyser le type de duplication qui a eu lieu [56, 124].

Même si les méthodes phylogénétiques sont reconnues comme étant les plus précises pour identifier des relations d'orthologie et de paralogie [6, 28, 31, 64, 87, 95], elles sont impraticables dans le cadre d'une étude sur des protéomes entiers. C'est pourquoi les approches développées se basent sur l'analyse de la similarité des séquences.

Dans les sections suivantes, les quelques méthodes existantes pour la détection de gènes in-paralogues sont présentées. À la base, ces approches emploient la même méthodologie : une grande similarité entre des gènes d'espèces différentes est un signe d'orthologie et un gène qui est plus similaire avec un autre du même génome est un indice d'in-paralogie. Dans le chapitre 6, une nouvelle méthode d'inférence de gènes in-paralogues se basant sur l'analyse des propriétés d'un graphe de similarité coloré est présentée.

3.6.1 InParanoid

InParanoid [136] compare l'ensemble des séquences protéiques de deux espèces pour inférer des relations d'orthologie et d'in-paralogie. La première étape consiste donc à faire

des comparaisons de toutes les séquences des deux espèces entre elles à l'aide du programme BLAST [7] afin d'identifier des meilleurs résultats bidirectionnels (MRBs) (*bidirectional best hits*). Soit deux gènes a_1 (appartenant au génome 1) et a_2 (appartenant au génome 2), a_1 et a_2 sont des MRBs si et seulement si il n'existe aucun gène b_1 (du génome 1) qui est plus similaire à a_2 que a_1 et, réciproquement, aucun gène b_2 (du génome 2) qui est plus similaire à a_1 que a_2. Toutes ces paires de séquences, les MRBs, vont former les paires de gènes orthologues principales.

La deuxième étape consiste à identifier les gènes in-paralogues pour chacune des paires d'orthologues principales, s'ils existent. Pour chaque paire principale (qui est formée d'un gène a_1 dans l'espèce 1 et a_2 dans l'espèce 2), tous les gènes du génome 1 qui sont plus similaires au gène a_1 qu'à n'importe quel gène de l'autre espèce sont inférés in-paralogues au gène a_1 (idem pour le gène a_2). Un exemple graphique est présenté dans la figure 3.9. Un score de confiance est ensuite calculé pour chaque paire d'in-paralogues, indiquant tout simplement une mesure de sa distance avec l'orthologue de la paire principale. Le score se situe entre 100%, si le gène est identique à l'orthologue principal, et 0% si le gène est à la limite de ne pas être considéré comme in-paralogue (sur la bordure du cercle dans la figure 3.9).

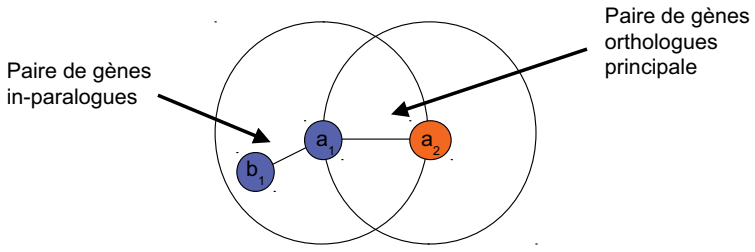

Figure 3.9 – Exemple graphique d'inférence d'une paire de gènes in-paralogues avec InParanoid. La paire de gènes orthologues principale est formée du gène a_1 (du génome 1, en bleu) et du gène a_2 (appartenant au génome 2, en orange). La distance physique entre les gènes dans la figure correspond à la distance entre les gènes (l'inverse de la similarité). Étant donné que le gène b_1 est plus similaire au gène a_1 qu'à n'importe quel autre gène du génome 2, InParanoid infère la paire $b_1 - a_1$ comme in-paralogue.

La dernière étape consiste à regrouper ou séparer les groupes d'orthologues qui se chevauchent après l'étape de l'identification des gènes in-paralogues.

3.6.1.1 MultiParanoid

Les auteurs d'InParanoid ont plus tard développé MultiParanoid [4], une extension du programme précédent permettant d'obtenir des résultats pour plus que deux espèces. En fait, MultiParanoid permet seulement d'assembler les résultats de plusieurs recherches faites avec InParanoid afin de former des groupes d'orthologues pour plusieurs espèces. Par exemple, dans le cas où on voudrait étudier les génomes 1, 2 et 3, il faut d'abord utiliser InParanoid pour les trois comparaisons possibles entre les génomes (1-2, 1-3 et 2-3). Ensuite, si on a identifié un même groupe d'orthologues dans différentes comparaisons de paires de génomes, MultiParanoid les regroupe ensemble (principe de transitivité).

3.6.2 OrthoMCL

OrthoMCL [101] est une méthode relativement similaire à InParanoid, dans le sens qu'elle utilise sensiblement la même approche pour l'identification de gènes in-paralogues (que les auteurs nomment "paralogues récents"). Comme InParanoid, OrthoMCL est une méthode de regroupement (*clustering*) de gènes utilisant les MRBs de BLAST. Cependant, OrthoMCL permet d'étudier plus que deux espèces à la fois et utilise une méthode probabiliste, le programme MCL (*Markov Cluster algorithm*), pour le partitionnement des gènes en groupes d'orthologues et de paralogues récents.

Il est important de noter qu'étant donné que les gènes de plusieurs espèces (plus de deux) peuvent être regroupés par cette méthode, les gènes in-paralogues identifiés dans ces groupes ne sont pas nécessairement in-paralogues par rapport aux spéciations les plus récentes dans la phylogénie de ces espèces ; des gènes out-paralogues peuvent être inclus

dans les résultats.

3.6.3 OrthoInspector

OrthoInspector [105] est une quatrième méthode qui permet d'inférer des relations d'in-paralogie en plus des relations d'orthologie. OrthoInspector se base aussi sur les résultats de comparaisons BLAST entre tous les gènes et permet d'analyser plusieurs espèces à la fois, mais assemble les groupes d'in-paralogues en premier, contrairement aux autres méthodes présentées plus haut. Contrairement à OrthoMCL, les relations d'in-paralogie sont clairement définies en indiquant l'évènement de spéciation correspondant aux in-paralogues inférés. Plus précisément, toutes les paires de gènes dans une espèce sont qualifiées d'in-paralogues par rapport à une deuxième espèce si la distance entre la paire est inférieure à celle calculée avec n'importe quel gène dans la deuxième espèce.

Après cette première étape, on se retrouve avec, dans chaque espèce étudiée, des gènes seuls et des groupes de gènes in-paralogues entre eux. Pour simplifier, disons qu'un groupe de gènes in-paralogues peut ne contenir qu'un seul gène. OrthoInspector identifie ensuite des relations d'orthologies dans le cas où un des gènes d'un groupe d'in-paralogues d'une espèce possède une grande similarité avec un des gènes d'un autre groupe chez une autre espèce et vice versa (pas nécessairement le même gène dans les deux directions ; la méthode ne considère pas les MRBs).

Enfin, OrthoInspector détecte les contradictions dans les relations d'orthologie inférées, sans toutefois les corriger. Par exemple, si deux groupes d'in-paralogues dans deux espèces ont été identifiés comme orthologues, mais qu'un des gènes possède une grande similarité avec un gène ne faisant pas partie d'aucun des deux groupes, une contradiction est signalée.

3.7 Inférence de l'évolution des gènes d'ARNt

Un grand nombre d'études ont été réalisées dans les dernières décennies sur la fonction et la structure (secondaire et tertiaire) des ARNs de transfert (ARNt). Cependant, l'analyse de l'évolution des gènes d'ARNt du point de vue de leur nombre, de leur organisation dans le génome et des mécanismes évolutifs impliqués a reçu peu d'attention avant tout récemment.

Dans les sections suivantes, les stratégies qui ont été employées récemment pour étudier l'évolution des gènes d'ARNt sont décrites.

3.7.1 Approche basée sur l'identification de régions orthologues

Dans une étude menée sur l'évolution des ARNt chez 12 espèces de drosophiles [137], Rogers *et al.* ont utilisé les régions flanquantes des gènes d'ARNt afin d'identifier des ensembles de gènes orthologues. En effet, il est difficile d'utiliser seulement les séquences des ARNt pour identifier des relations d'orthologie ou de paralogie, car ces séquences sont très courtes et très similaires entre elles.

Pour chacune des espèces étudiées, les auteurs ont recherché les régions flanquantes de chaque gène d'ARNt (5 kpb de chaque côté) dans le génome de *Drosophila melanogaster*, qui a été utilisé comme génome de référence. Une relation d'orthologie était inférée à chaque fois que les séquences flanquantes des deux côtés étaient retrouvées dans le génome de référence et qu'un gène d'ARNt était présent au milieu. On construisait ensuite des ensembles d'orthologues, dans le cas où plusieurs gènes d'ARNt de différentes espèces étaient reliés au même locus chez *D. melanogaster*. Ces ensembles ont par la suite été filtrés en comparant cette fois les séquences des ARNt afin de retirer de chaque ensemble les gènes qui étaient sous un certain seuil de similarité.

Ces ensembles d'orthologues ont ensuite été utilisés pour inférer des duplications et

des pertes sur la phylogénie des espèces. Premièrement, pour chaque ensemble d'orthologues, on plaçait une seule duplication le plus bas possible dans l'arbre de manière à ce que toutes les espèces représentées dans l'ensemble possèdent la copie. Deuxièmement, on inférait des pertes sur les branches qui menaient à des espèces ne possédant pas la copie en question.

3.7.2 Méthodes basées sur l'alignement

Dans le cadre d'une analyse des gènes d'ARNt d'espèces relativement proches, comme dans le cas de souches de bactéries qui ont divergé récemment (souches de *Bacillus* par exemple), il a été démontré qu'on peut reformuler le problème de l'inférence d'une histoire évolutive en un problème d'alignement [83]. Cette reformulation du problème permet d'inférer des évènements évolutifs qui sont visibles dans l'alignement de l'ordre des gènes de deux génomes. Autrement dit, en appliquant cette approche sur des espèces ayant divergé récemment, on peut supposer que peu d'évènements évolutifs chevauchants se sont produits et que la majorité des évènements sont donc visibles dans l'alignement des génomes. Deux algorithmes se basant sur ce principe ont été developpés dans notre laboratoire et sont présentés dans les sections suivantes.

3.7.2.1 Algorithme exact

Un algorithme exact de programmation linéaire a tout d'abord été développé [83]. Afin de simplifier le problème d'inférence, seulement les évènements de duplication et de perte ont été considérés. Ceci a comme avantage que chaque alignement optimal (minimisant un coût pour les duplications et les pertes) correspond directement à un génome ancestral. En effet, contrairement aux réarrangements génomiques qui peuvent être appliqués sur l'un ou l'autre des génomes comparés, les duplications et les pertes sont des évènements

asymétriques qui, une fois inférés, ne peuvent donner lieu qu'à un seul ancêtre (voir la figure 3.10).

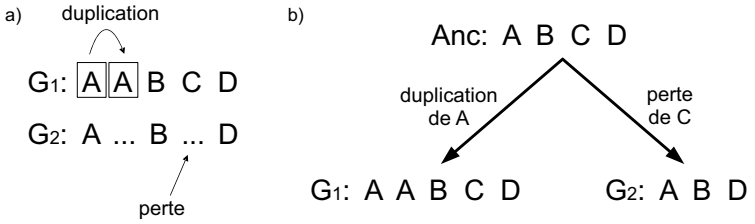

Figure 3.10 – Exemple d'un alignement de génomes et de l'ancêtre correspondant. a) Alignement des génomes G_1 et G_2 résolu (dans lesquels on a étiqueté les trous par des duplications ou des pertes de gènes). b) L'histoire évolutive correspondante à l'alignement montré en a). Il n'y a aucune ambiguïté quant à la séquence ancestrale.

Le désavantage de cette méthode, en plus du fait qu'elle ne considère pas les réarrangements génomiques (transpositions et inversions) et les substitutions, est que dans le pire cas, la programmation linéaire peut prendre un nombre d'étapes exponentiel pour résoudre l'alignement. Toutefois, le temps d'exécution de cette approche exacte était raisonnable pour les expériences avec les données biologiques (moins d'une minute).

3.7.2.2 Heuristique de programmation dynamique

Afin d'améliorer le temps d'exécution, un deuxième algorithme basé sur l'alignement des génomes a été développé dans notre laboratoire [15]. Cette approche utilise la programmation dynamique pour faire l'alignement et permet de considérer un plus grand nombre d'évènements évolutifs : en plus des duplications (inversées ou pas) et des pertes, les inversions, les transpositions (inversées ou pas) et les substitutions peuvent être inférées. En d'autres termes, les trous dans l'alignement peuvent correspondre aussi à des transpositions (en plus des duplications et des pertes), alors que les autres positions de l'alignement, en plus de correspondre à des *matches*, peuvent être associées à des inversions et des substitutions.

Une étape supplémentaire est cependant nécessaire pour inférer correctement les séquences ancestrales, puisque les évènements *symétriques* (c'est-à-dire les évènements pour lesquels il est impossible de savoir dans quel génome ils se sont produits en faisant une comparaison de deux génomes, comme les transpositions, les inversions et les substitutions) peuvent être appliqués aux deux génomes comparés sans affecter le coût de l'alignement. Une comparaison avec un ou plusieurs génomes proches dans la phylogénie des espèces étudiées est donc nécessaire afin d'identifier correctement le génome qui a subi l'évènement en question.

Le dernier projet de cette thèse a porté sur l'utilisation de cette méthode dans le but d'étudier l'évolution des gènes d'ARNt chez 50 souches de *Bacillus*. Le chapitre 7 présente un manuscrit (non publié) sur les résultats obtenus.

CHAPITRE 4

GENOME HALVING AND DOUBLE DISTANCE WITH LOSSES

Olivier Tremblay Savard[1], **Yves Gagnon**[1], **Denis Bertrand**[1] **et Nadia El-Mabrouk**[1]

Article publié dans le *Journal of Computational Biology* en 2011 [146].

4.1 Contributions

Olivier Tremblay Savard et Yves Gagnon sont considérés comme les co-premiers auteurs de l'article. Tous les auteurs ont participé à la conception des algorithmes. **Olivier Tremblay Savard**, Yves Gagnon et Denis Bertrand ont implémenté l'heuristique pour le calcul de la double distance (avec et sans pertes de gènes). **Olivier Tremblay Savard**, Yves Gagnon et Denis Bertrand ont conçu et réalisé les expériences sur les données simulées. Tous les auteurs ont participé à la rédaction de l'article.

[1]DIRO, Université de Montréal, Canada

4.2 Abstract

Given a phylogenetic tree involving Whole Genome Duplication events, we contribute to solving the problem of computing the rearrangement and DCJ distances on a branch of the tree linking a duplication node d to a speciation node or a leaf s. In the case of a genome G at s containing exactly two copies of each gene, the *genome halving problem* is to find a perfectly duplicated genome D at d minimizing the rearrangement distance with G. We generalize the existing exact linear-time algorithm for genome halving to the case of a genome G with missing gene copies. In the case of a known ancestral duplicated genome D, we develop a greedy approach for computing the distance between G and D, called the *double distance*. Two algorithms are developed in both cases of a genome G containing exactly two copies of each gene, or at most two copies of each gene (with missing gene copies). These algorithms are shown time-efficient and very accurate for both the rearrangement and DCJ distances.

4.3 Introduction

Whole genome duplication (WGD), which has the effect of simultaneously doubling all the chromosomes of a genome, is probably the most spectacular evolutionary event leading to the creation of multiple gene copies. Right after the WGD event, a genome D_{predup} is transformed into a *perfectly duplicated genome* $D = (D_{predup} \oplus D_{predup})$ containing a complete set of duplicated chromosomes. However, subsequent evolutionary events such as rearrangements, losses and local duplications blur this initial perfect duplicate status. Usually, a hypothesis that a given species has been subject to a WGD event during its evolution is based on the fact that syntenies found in pairs (exactly two paralogous regions) cover a high proportion of the genome. Such evidence has shown up across the whole eukaryote spectrum, from the protist *Giardia* to the yeast species [74], including most

plant lineages, several insects, fishes, amphibians, and even mammalian species [127]. In plant lineages, the angiosperm genomes that have been completely sequenced to date all show evidence of WGD events : three ancient polyploidy events have been revealed by the *Arabidopsis thaliana* genome [24, 27], one by the rice genome that might characterize all monocots (in the grass family, maize reveals an additional WGD) [142], and others by the poplar, grape and papaya genomes [151].

Being able to reconstruct the ancestral pre-duplicated genome is essential from both genome and organismal evolutionary standpoints. In particular, it allows to trace back the last evolutionary events that have occurred *en route* from this ancestor to the present-day genomes, understand the specificity of a given lineage by looking at the differences that separate it from its closest evolutionary neighbor, study the variation in rearrangement and loss rates among the different branches of a phylogenetic tree, and the consequences of such variations on the species. In most cases, analyzing the duplication status of syntenies in extant species allows to position the WGD events on the corresponding phylogenetic tree with confidence, leading to a tree with additional *WGD nodes* that, in contrast to the speciation nodes, each has a single child. In addition, simple assumptions can be used to infer the content of ancestral genomes from that of extant species. However, inferring ancestral gene orders is far from being a simple task.

In the case of genomes with single gene copies, many algorithms have been developed to solve the *rearrangement phylogeny problem*, which consists in inferring gene orders at the internal nodes of a tree so that the sum of distances among all branches is minimized. The most natural distance between two gene orders is the minimum number of rearrangement events required to transform one gene order into the other. The rearrangements that have been most studied by the genome rearrangement community are inversions and reciprocal translocations (including fusion and fission). More recently, another distance that has been extensively studied is the Double Cut-and-Join (DCJ) distance which re-

presents a greater repertoire of rearrangement events while giving rise to simpler formal results [17, 18, 181].

A prerequisite for applying any of the algorithms developed for solving the rearrangement phylogeny problem to a phylogeny with WGD nodes is to be able to compute the distance on a branch of the phylogeny. However, this is far from being straightforward, as the orthology relationship between duplicated genes is not set. In particular, computing the distance between a rearranged duplicated genome G (a genome with exactly two copies of each gene but in any order) and a perfectly duplicated genome D, called the *double distance* in [68, 159], has been shown to be NP-hard for the DCJ distance [159]. When the ancestral genome D is unknown, the *genome halving problem* seeks for a perfectly duplicated genome D minimizing the rearrangement distance between G and D. In 2003, El-Mabrouk and Sankoff have presented the first formal result related to genome duplication, which is an exact linear-time algorithm for solving the genome halving problem [51]. Our results have been reformulated by Alekseyev and Pevzner [2] using an alternative representation of the breakpoint graph. Subsequently, Sankoff and colleagues [191, 193], and more recently Gavranović and Tannier [68], used variations of the genome halving strategy (*Guided Genome Halving* or GGH) to find the preduplicated ancestor of a doubled genome in the presence of a non-duplicated outgroup [191, 193].

In this paper, we contribute to solving a number of problems related to the computation of the rearrangement and DCJ distances on a branch of a phylogenetic tree connecting a first WGD node to a speciation node or a leaf, in both cases of a known and unknown pre-duplicated genome (label of the WGD node). In the case of an unknown ancestral genome, our result is a generalization of the genome halving algorithm to a genome G with missing gene copies. In the case of a known ancestral genome D, we develop two greedy algorithms for both cases of a genome G containing exactly two copies of each gene, or at most two copies of each gene (with missing gene copies). These algorithms are

shown time-efficient and very accurate for both the rearrangement and DCJ distances.

4.4 Preliminaries

Let Σ be a set of n genes. A *string* is a sequence of genes from Σ, where each gene is signed ($+$ or $-$) depending on its transcriptional orientation. The *reverse* of a string $X = x_1 x_2 \dots x_r$ is the string $-X = -x_r\ -x_{r-1} \dots - x_1$. A *chromosome* is a string, and a *genome* is a collection of chromosomes. The reverse of an entire chromosome is considered to be equivalent to the initial chromosome. A *unichromosomal* genome has a single chromosome, and a *multichromosomal* genome has at least two nonempty chromosomes $C_1, C_2, \dots, C_N$. A *circular chromosome* is a string $x_1 \dots x_r$, where x_1 is considered to follow x_r. A chromosome that is not circular is *linear*. To represent its endpoints, we add an "artificial gene", denoted O, at each extremity. In other words, a linear chromosome is a string of the form $Ox_1 \dots x_r O$.

In this paper, we consider both uni- and multichromosomal genomes. As most unichromosomal genomes are formed by a circular chromosome, and most multichromosomal genomes are formed by linear chromosomes, only circular unichromosomal genomes, and linear multichromosomal genomes are considered here.

4.4.1 Evolutionary events and genomic distances

All the following evolutionary events apply to both uni- and multichromosomal genomes, except translocations that are only relevant for multichromosomal genomes.

- A *reversal* (or *inversion*) is an operation that replaces some proper substring (*i.e.* any substring that is not the full string) of a chromosome by its reverse.
- A *translocation* between two chromosomes $X = X_1 X_2$ and $Y = Y_1 Y_2$ is an event transforming them into the two chromosomes $X_1 Y_2$ and $Y_1 X_2$, or into $X_1(-Y_1)$ and

$(-Y_2)X_2$. Two special cases of reciprocal translocations are *fusions* (if one of the two chromosomes generated by the translocation is an empty string) and *fissions* (if one of the two input chromosomes is the empty string).

– A *Whole Genome Duplication* (WGD) is an event resulting in the duplication of the genomic content. More precisely, in the case of a multichromosomal genome $G = \{C_1, C_2, \ldots, C_N\}$, a WGD transforms G into a multichromosomal genome $D = \{C_1, C'_1, C_2, C'_2, \ldots, C_N, C'_N\}$ containing $2N$ chromosomes where, for each $1 \leq i \leq N$, $C_i = C'_i$. In the case of a circular genome G represented by the string $x_1 x_2 \ldots x_r$, a WGD transforms G into the circular genome D represented by the string $x_1 x_2 \ldots x_r\ x'_1 x'_2 \ldots x'_r$, where, for each $1 \leq j \leq r$, $x_j = x'_j$. Therefore, the doubled circular genome stays unichromosomal.
– Finally, a *loss* is an operation removing a proper substring from a chromosome.

A *rearrangement event* will refer to an inversion or a translocation event. The *rearrangement distance* between two genomes G and H (with the same gene content or not), denoted $d_R(G,H)$, is the minimum number of rearrangement events in a scenario transforming G into H. In the case of genomes with single gene copies, computing the inversion and/or translocation distance has been shown to be a polynomial-time problem, and the best developed method runs in linear time [10, 16].

Another distance that has been extensively studied in the last years is the DCJ distance [17, 18, 181]. A *Double-Cut-and-Join (DCJ)* is an operation that "cuts" two adjacencies pq and rs in a genome, and replaces them by either pr and qs, or ps and qr. The repertoire of DCJ operations include inversions, reciprocal translocations, fusions and fissions, but also other "artificial" rearrangement operations such as the creation of "intermediate" circular chromosomes. Using such a circular intermediate, a transposition can actually be mimicked by two DCJ operations. Computing the DCJ distance between two signed permutations is a linear-time problem [181].

4.4.2 Genome definitions

In what follows, we consider G to be a genome defined on a set Σ of genes, i.e. g is in G if $g \in \Sigma$.

- G is an *extension* of a genome H, if the gene content of H is a subset of the gene content of G, and there is a sequence of gene insertions transforming H into G.
- G is a *singleton genome* if each gene is present exactly once in G.
- G is a *Rearranged Duplicated genome (RD genome)* if each gene is present exactly twice in G.
- G is a *perfectly duplicated genome* (or *duplicated genome* for short) if :
  - *The multichromosomal case :* G is an RD genome containing an even number $2N$ of chromosomes, with two identical copies of each chromosome. If D is the set of the N different chromosomes, then we write $G = (D \oplus D)$.
  - *The circular case :* G is an RD genome and there is a string D such that G is exactly D followed by D. We also write $G = (D \oplus D)$.
- G is a *Rearranged Duplicated genome with Losses (RDL genome)* if each gene is present at least once and at most twice in G.
- G is a *Duplicated genome with Losses (DL genome)* if each gene is present in one or two copies in G, and if a duplicated genome D can be obtained from G by an appropriate insertion of an additional copy of each *singleton* (gene present in one copy in G). In other words, there is an extension of G that is a duplicated genome.

Let G be an RD genome and H be an RDL genome. We define the evolutionary cost $\mathscr{E}(G,H)$ as the minimum number of inversions, translocations and losses required to transform G into H.

4.4.3 The breakpoint graph

In a series of papers [78–80], Hannenhalli and Pevzner (hereafter HP) developed polynomial-time algorithms for computing the rearrangement distance (inversion only, translocation only, or inversion+translocation) between two singleton genomes G and H on Σ. The algorithms all depend on a bicolored graph $\mathscr{B}(G,H)$, called the *breakpoint graph*, constructed from G and H as follows (Tesler's formalism [160]).

4.4.3.1 Graph $\mathscr{B}(G,H)$

If a gene x of Σ has a positive sign, replace it by the pair $x^t x^h$, and if it is negative, replace it by $x^h x^t$. Then the set V of vertices of $\mathscr{B}(G,H)$ is the set of x^t and x^h for all x in Σ. Any two vertices of V that are adjacent in some chromosome of G, other than x^t and x^h deriving from the same x, are connected by a black edge, and any two that are adjacent in H are connected by a gray edge. Notice that adjacencies to O are not represented.

In the case of circular chromosomes, each vertex in V is incident to exactly one black and one gray edge, and thus the graph uniquely decomposes into $c(G,H)$ disjoint cycles of alternating edge colors (*alternating cycles* for short).

In the case of G and H being multichromosomal genomes, let an *endpoint vertex of* G (resp. of H) be a vertex of V adjacent to O in G (resp. in H). Consider the *degree of a vertex* as being the number of edges incident to this vertex. Then any vertex has degree zero if it is an endpoint in both G and H, one if it is an endpoint in exactly one of the two genomes, and two otherwise. Thus, the graph decomposes into $c(G,H)$ cycles and $p(G,H)$ paths of alternating edge color. Notice that a path may contain only one vertex and no edges. We denote by p_{GG} (resp. p_{HH}) the number of paths linking two endpoints of G (resp. of H). If G and H have the same number of chromosomes, then $p_{GG} = p_{HH}$. Otherwise, suppose w.l.o.g. that G has more chromosomes than H, then $p_{HH} \leq p_{GG}$.

4.4.3.2 The rearrangement distance

Although somehow different algorithms are required for sorting by translocation only, inversion only or inversion+translocation, all results in [78–80], revisited by Tesler [160] for multichromosomal genomes, can be summarized by a unique formula given below :

$$\mathsf{HP}: \quad d_R(G,H) = n + N - C(G,H) + h(G,H)$$

where n is the number of genes, N is the number of chromosomes of G, and $C(G,H) = c(G,H) + p(G,H) - p_{GG}$. In the case of circular genomes, $N = p(G,H) = p_{GG} = 0$. As for $h(G,H)$, it is a correction parameter that has a different value depending on the considered model. In all cases, $h(G,H)$ is related to the decomposition of $\mathscr{B}(G,H)$ into components, where a *component* is a maximal set of crossing cycles and paths. A component is termed *good* if it can be transformed into a set of cycles of size 1 by increasing the number of cycles at each step, and *bad* otherwise. The parameter $h(G,H)$ reflects the number of bad components of the graph. As the probability for a component to be bad is low, the value of $h(G,H)$ is usually low compared to the dominating parameter $C(G,H)$.

4.4.3.3 The DCJ distance

Based on the breakpoint graph, the DCJ distance between G and H can be expressed as follows [17, 159] :

$$\mathsf{DCJ}: \quad d_{DCJ}(G,H) = n - \left(c(G,H) + \frac{p_{even}}{2} \right)$$

where p_{even} is the number of paths with an even number (≥ 0) of edges.

4.5 Genome Halving with Losses

Given an RD genome G, the *genome halving problem* is to find a duplicated genome D minimizing the rearrangement distance with G. In other words, let $d_R(G)$ be the minimum rearrangement distance between G and any duplicated genome D. Then the problem is to find a duplicated genome D such that $d_R(G) = d_R(G,D)$.

In [51] El-Mabrouk and Sankoff have developed a linear-time algorithm, called Algorithm Dedouble, for the reversals-only version of the problem (in the case of unichromosomal genomes), the translocations-only version, and the version with both reversals and translocations. The approach was to start from a *partial breakpoint graph* $\mathscr{B}(G)$, i.e. the breakpoint graph with the set of edges restricted to the black edges representing G (see Figure 4.2.(b) for an example), and to complete this graph with a set of "valid" gray edges, i.e. gray edges representing a duplicated genome D (thin edges in Figure 4.2.(c)), in a way maximizing the number of cycles and paths (parameters $c(G,D)$ and $p(G,D)$ in the HP formula). The second step was then to perform modifications on the obtained graph in order to remove bad components that can be avoided, and obtain a duplicated genome D minimizing the rearrangement distance with G (i.e. minimizing the HP formula).

Here, we seek to generalize Algorithm Dedouble to a present-day genome G containing both duplicated genes and singletons, i.e. to an RDL genome. Let G be a present-day RDL genome. We assume that G has evolved from an ancestral singleton genome through a WGD, and a sequence of inversions, translocations and loss events. We are then interested in finding such a pre-duplicated singleton genome D_{predup} minimizing the number of rearrangements needed to obtain G (see model M_1 in Figure 4.1). Note that we do not attempt to minimize the number of losses.

The following theorem allows to reduce the evolutionary model to a simpler one (model M_2 in Figure 4.1), where all losses occur first, followed by all rearrangement events.

M_1: G $\xleftarrow{Inv,\ Trans,\ Loss}$	$D = (D_{predup} \oplus D_{predup})$	$\xleftarrow{WGD}$ D_{predup}	
RDL genome	Duplicated genome	Singleton genome	
M_2: G $\xleftarrow{Inv,\ Trans}$ A $\xleftarrow{Loss}$	$D = (D_{predup} \oplus D_{predup})$	$\xleftarrow{WGD}$ D_{predup}	
RDL genome DL genome	Duplicated genome	Singleton genome	
M_3: G $\xleftarrow{Loss}$ G' $\xleftarrow{Inv,\ Trans}$	$D = (D_{predup} \oplus D_{predup})$	$\xleftarrow{WGD}$ D_{predup}	
RDL genome RD genome	Duplicated genome	Singleton genome	

Figure 4.1 – Evolutionary models. Evolutionary models M_1, M_2 and M_3, considered for a present-day rearranged duplicated genome with losses G. Direction of evolution is represented by arrows orientation.

Theorem 1. *Let G be an RDL genome and D be a duplicated genome. Then there exists a DL genome A with the same gene content as G, such that $d_R(G,A) = d_R(G,D)$.*

Proof. By induction on $n = \mathscr{E}(G,D)$

1. The property is trivially verified for $n = 0$ and $n = 1$.

2. Suppose the induction hypothesis is verified for a given $n \geq 1$. Now suppose that $\mathscr{E}(G,D) = n+1$, and let $\mathscr{E} = E_1, E_2, \ldots, E_n, E_{n+1}$ be a sequence of $n+1$ events transforming a duplicated genome D into an RDL genome G. Let G' be the genome obtained after performing the sequence of n events $\mathscr{E}' = E_1, E_2, \ldots, E_n$ on D. Then $n = \mathscr{E}(G',D)$ as otherwise (if n is not the minimum number of events transforming D into G') $n+1$ would not be the minimum number of events transforming D into G. Moreover, by the induction hypothesis, there exists a DL genome A' for which $d_R(G',A') = d_R(G',D)$.

If E_{n+1} is a rearrangement event, the DL genome A with the same gene content as G is equal to A'. Then, we have $d_R(G,A) = d_R(G,A') = d_R(G',A') + 1$ and $d_R(G,D) = d_R(G',D) + 1$. Therefore, $d_R(G,A) = d_R(G,D)$.

Otherwise, E_{n+1} is a loss event. Let A be the DL genome obtained from A' by remo-

ving the genes that are removed by the loss operation E_{n+1}. Then, it is easy to see that a minimum sequence of k rearrangement events transforming A' into G' can be converted into a sequence of k rearrangement events transforming A into G (just by removing the lost genes from the inverted or translocated segments). Therefore $d_R(G,A) \leq d_R(G',A')$. Similarly, a minimum sequence of k rearrangement events transforming A into G can be converted into a sequence of k rearrangement events transforming A' into G'. Therefore, $d_R(G,A) = d_R(G',A') = d_R(G,D)$. □

Corollary 1. *Let G be an RDL genome, and A be a DL genome with the same gene content as G, minimizing the cost $d_R(G,A)$. If D is the duplicated genome that is an extension of A, then $d_R(G) = d_R(G,D)$.*

Proof. Let A be a DL genome with the same gene content as G minimizing the cost $d_R(G,A)$, and D be the duplicated genome that is an extension of A. Then we have $d_R(G,D) = d_R(G,A)$. Suppose $d_R(G) \neq d_R(G,A)$, i.e. $d_R(G,A) > d_R(G)$. Let D' be a duplicated genome such that $d_R(G,D') = d_R(G)$. Then, from Theorem 1, there is a DL genome A' such that $d_R(G,A') = d_R(G,D') = d_R(G)$. And thus $d_R(G,A') < d_R(G,A)$, which is a contradiction with the fact that A minimizes the rearrangement cost. □

Therefore, finding a duplicated genome D such that $d_R(G) = d_R(G,D)$ can be reduced to the problem of finding a DL genome A with the gene content of G such that $d_R(G,A)$ is minimal over all DL genomes with the gene content of G. In other words, loss events can be ignored.

To find such a DL genome A, we use a generalization of Algorithm Dedouble, called Algorithm Dedouble-RDL(G), that proceeds as follows :

1. Consider the RD genome G' obtained from G by "gluing" singletons to an adjacent gene. More precisely, consider a given orientation for chromosomes. Then, for each

maximum sequence S of singletons in G : (1) if S is a chromosome, then just remove this chromosome ; (2) otherwise, if S is connected to a left extremity of a chromosome, then replace its successor x (the gene representing the right adjacency of S in G) by the artificial gene $x' = Sx$; (3) otherwise, if S is not connected to a left extremity of a chromosome, then replace its predecessor x (possibly already updated in step (2)) by a new artificial gene x' representing the sequence xS.

2. Use Algorithm Dedouble to infer a duplicated genome A' from G'.
3. Recover a DL genome A from A' by replacing each of its artificial gene by its corresponding sequence of singletons, and by adding all removed chromosomes of G (formed exclusively of singletons).

The following theorem immediately follows from the fact that Algorithm Dedouble outputs a doubled genome A' minimizing the distance to G', and that singletons are preserved in the same order in G and A.

Theorem 2. *Let G be an RDL genome and A be the DL genome resulting from* Algorithm Dedouble-RDL(G). *Then* $d_R(G,A) = d_R(G)$.

4.6 An algorithm for the Double Distance

Let G be an RD genome and $D = (D_{predup} \oplus D_{predup})$ be a duplicated genome. The problem of computing the DCJ distance between G and D has already been shown to be an NP-hard problem [159], contrary to the polynomial-time complexity of computing the distance between two singleton genomes. This difference in complexity is the result of the missing one-to-one orthology relationship between the gene copies. In other words, given a labelling of the genes in G, the problem is to find a labelling of the genes in D leading to a minimum distance between G and D.

Consider a given beginning gene, in the case of a circular genome, or a given order and left-to-right orientation of chromosomes in the case of a multichromosomal genome G. Then, for each gene x (present in two copies in G and also in D), label the first occurrence of x in G as x_1 and the second as x_2. Let $\mathscr{B}(G)$ be the partial breakpoint graph for G. To complete this partial graph, each double adjacency (x^r, y^s) in D (where $r,s \in \{t,h\}$) should be represented in a completed graph $\mathscr{B}(G,D^L)$, where D^L is a labelling of genome D, by either one of the following pairs of gray edges : $\{(x_1^r, y_1^s),(x_2^r, y_2^s)\}$, or $\{(x_1^r, y_2^s),(x_2^r, y_1^s)\}$. Each of these two cases leads to a different labelling of the gene copies in D. The problem is then to choose the pairs of gray edges allowing to minimize the HP formula in the case of the rearrangement distance, or the DCJ formula in the case of minimizing the DCJ distance.

Here, we focus on maximizing the dominating value $C(G,D)$ in the HP formula. In the case of genome halving, this simplification has been called the Weak Genome Halving Problem [2]. We similarly define our simplified problem as follows :

Weak Double Distance Problem. *For a given labelled RD genome G and an unlabelled duplicated genome D, find a labelling D^L of D such that the number of alternating cycles $C(G,D)$ of the breakpoint graph $\mathscr{B}(G,D^L)$ is maximized over all possible labellings of D.*

Notice that, in the case of a circular genome, a labelling of D maximizing the parameter $C(G,D)$ also maximizes the DCJ formula, as $C(G,D) = c(G,D)$ in this case. In the multichromosomal case, a labelling of D maximizing $C(G,D)$ is likely to also maximize the DCJ formula, though there is no guarantee for that.

Clearly, the "best" exhaustive approach trying all possible labellings for D has a worst running-time complexity in $O(n.2^n)$ for $n = |\Sigma|$. Indeed, D has 2^n possible labellings, and for each labelling, the most efficient approach for computing the rearrangement distance between G and D is linear.

4.6.1 Circular genomes

Let G be a circular RD genome and D be a circular duplicated genome. We consider the contracted breakpoint graph representation $\mathscr{CB}(D,G)$ defined as follows : the set of vertices of $\mathscr{CB}(D,G)$ is $V = \{x^r, \text{for all } x \in \Sigma \text{ and } r \in \{t,h\}\}$. Any two vertices which are adjacent in D (except the extremities of a same gene) are connected by two parallel gray edges, and any two adjacent in G (except the extremities of a same gene) are connected by a black edge (see Figure 4.2.(a)). Such representation has previously been used in the context of genome halving for circular [3] and multichromosomal genomes [68], with the difference that each gray edge was represented exactly once. It follows that each vertex of $\mathscr{CB}(D,G)$ is adjacent to exactly two gray edges and two black edges.

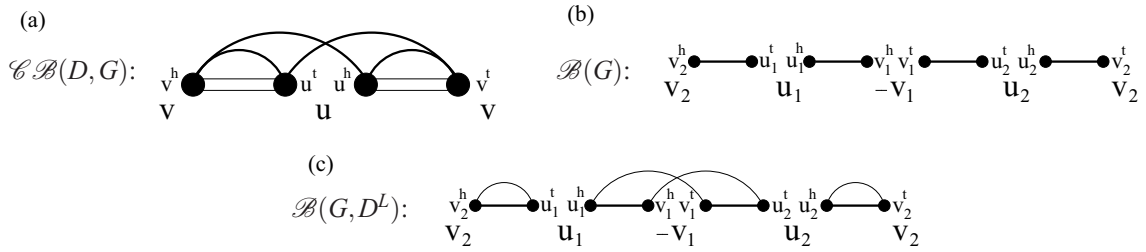

Figure 4.2 – Contracted, partial and completed breakpoint graphs. (a) The contracted breakpoint graph $\mathscr{CB}(D,G)$ for the circular RD genome $G = (u\ -v\,u\,v)$ and the circular duplicated genome $D = (u\,v) \oplus (u\,v)$. Gray edges (thin lines) represent genome D and black edges (thick lines) represent genome G. (b) The partial breakpoint graph $\mathscr{B}(G)$. (c) The completed breakpoint graph $\mathscr{B}(G,D^L)$ corresponding to the labelling $G = (u_1\ -v_1\,u_2\,v_2)$ and $D^L = (u_1\,v_1) \oplus (u_2\,v_2)$. This labelling D^L, leading to 3 cycles, is optimal. The resulting rearrangement distance is 1.

Now consider the partial breakpoint graph $\mathscr{B}(G)$ of the labelled RD genome G (see Figure 4.2.(b)). Let C be an alternating edge-colour cycle in $\mathscr{CB}(D,G)$ with the set of vertices V, the set of black edges $b = \{b_1,\ldots,b_m\}$, and the set of gray edges $g = \{g_1,\ldots,g_m\}$. Then we can construct a corresponding cycle in $\mathscr{B}(G)$. More precisely, let $b^{\mathscr{B}} = \{b_1^{\mathscr{B}},\ldots,b_m^{\mathscr{B}}\}$ be a set of black edges in $\mathscr{B}(G)$ corresponding to $\{b_1,\ldots,b_m\}$ (i.e., for each i, the two vertices adjacent to $b_i^{\mathscr{B}}$ are two labelled copies of the vertices adjacent

to b_i), and let $V_b^{\mathscr{B}}$ be the set of all the vertices adjacent to $b_i^{\mathscr{B}}$, for all i. Then there is a set of gray edges $g^{\mathscr{B}} = \{g_1^{\mathscr{B}}, \ldots, g_m^{\mathscr{B}}\}$ corresponding to $\{g_1, \ldots, g_m\}$ and linking vertices of $V_b^{\mathscr{B}}$, allowing to form a single alternating cycle in $\mathscr{B}(G)$.

This observation leads to a greedy approach for labelling the genome D, or equivalently completing the partial graph $\mathscr{B}(G)$. Formally, a *completed graph* $\mathscr{B}(G, D^L)$ is a graph obtained from $\mathscr{B}(G)$ by adding gray edges such that each vertex of $\mathscr{B}(G, D^L)$ is adjacent to exactly 2 edges (one black and one gray) (Figure 4.2.(c)).

The general idea of Algorithm Double-Distance(G,D) given in Figure 4.3 is : at each step, pick an alternating cycle of minimum size from $\mathscr{CB}(D, G)$, construct the corresponding cycle in $\mathscr{B}(G)$, and then remove from $\mathscr{CB}(D, G)$ all used edges. The algorithm stops when the partial graph is completed.

Lemma 1. *Algorithm Double-Distance(G,D) terminates and results in a completed graph* $\mathscr{B}(G, D^L)$.

Proof. We will show that, at each step, if the graph $\mathscr{CB}(D, G)$ is non-empty (it still contains edges), then it contains an alternating cycle. Notice first that if the graph $\mathscr{CB}(D, G)$ is a non-empty balanced graph (i.e. every vertex has the same number of incident gray and black edges), then it contains at least one alternating cyle. This is a consequence of the fact that a balanced graph contains an alternating eulerian cycle in every connected component [96, 134].

Suppose that at a given step of the algorithm, $\mathscr{CB}(D, G)$ is a non-empty unbalanced graph. Clearly such a graph can not be the input of the algorithm, as at the beginning, each vertex has two adjacent edges of each color. On the other hand, it can not be the graph obtained after an iteration of the algorithm, as when edges adjacent to a vertex are removed, they are removed by bicolored pairs (i.e. one gray and one black edge).

Algorithm Double-Distance(G,D)
Input : $\mathscr{C}\mathscr{B}(D,G)$ *and the partial graph* $\mathscr{B}(G)$;
Output : The graph $\mathscr{B}(G)$ *completed with gray edges (i.e.* $\mathscr{B}(G,D^L)$) ;

1. **For** $CSize = 1$ to n **Do** ;
2. **For** $CVertex = b_1^l$ to b_n^l **Do**
3. **If** $\mathscr{C}\mathscr{B}(D,G)$ is empty (i.e. no edges left)
4. **Return** ;
5. **If** there is an alternating cycle C of size $CSize$ beginning at $CVertex$ **Then**
   Construct a corresponding cycle in $\mathscr{B}(G)$ *:*
6. Let b be the set of black edges and g the set of gray edges of C ;
7. Consider a set of black edges $b^{\mathscr{B}}$ corresponding to b in $\mathscr{B}(G)$ and not considered in a previous step ;
8. Let $V_b^{\mathscr{B}}$ be the set of vertices of $b^{\mathscr{B}}$;
9. Construct a set of gray edges $g^{\mathscr{B}}$ corresponding to g and linking vertices of $V_b^{\mathscr{B}}$ in $\mathscr{B}(G)$;
10. Remove from $\mathscr{C}\mathscr{B}(D,G)$ all edges of C ;
11. **End If**
12. **End For**
13. **End For**

Figure 4.3 – Algorithm Double-Distance(G,D). A greedy approach for completing the partial graph $\mathscr{B}(G)$ with gray edges representing the genome D. Here, $n = |\Sigma|$ is the number of different genes, and $b_1, b_2, \ldots, b_n$ is a left-to-right ordering of the black edges of $\mathscr{C}\mathscr{B}(D,G)$. For each i, b_i^l is the vertex representing the left adjacency of b_i. The size of a cycle is the number of black (or equivalently gray) edges of the cycle.

Therefore, if the graph $\mathscr{CB}(D,G)$ is non-empty, it is balanced and thus it contains at least one alternating cycle. Moreover, as only a finite number of alternating cycles can emerge from this graph, the algorithm is guaranteed to terminate (because at each step, at least one alternating cycle is removed).

Finally, since all gray edges of $\mathscr{CB}(D,G)$ are inserted in $\mathscr{B}(G)$, and each vertex of $\mathscr{B}(G)$ is considered only once as an adjacency of a new inserted gray edge (line 7), clearly when no edges remain in $\mathscr{CB}(D,G)$, the obtained graph $\mathscr{B}(G)$ is a completed graph. □

Let $D^L = u_{1,\alpha_1} \dots u_{r,\alpha_r} u_{1,\overline{\alpha_1}} \dots u_{r,\overline{\alpha_r}}$ be a labelling for $D = u_1 \dots u_r u_1 \dots u_r$, where, for each $1 \leq i \leq r$, $\alpha_i \in \{1,2\}$, and $\overline{\alpha_i}$ denotes the complementary element of α_i in $\{1,2\}$. A bi-circular representation of D^L is the pair of circular chromosomes $\{u_{1,\alpha_1} \dots u_{r,\alpha_r}$, $u_{1,\overline{\alpha_1}} \dots u_{r,\overline{\alpha_r}}\}$.

Lemma 2. *The gray edges of the completed graph resulting from the execution of Algorithm Double-Distance(G,D) represent either a labelling of D, or a bi-circular representation of a labelling of D.*

Proof. At each execution of the internal For Loop (line 2), the algorithm selects an alternating cycle C in $\mathscr{CB}(D,G)$, and constructs a corresponding cycle in $\mathscr{B}(G)$. In other words, for each gray edge g in C linking two vertices x^r, y^s, for r and $s \in \{t,h\}$, we construct, in $\mathscr{B}(G)$, a corresponding edge $g^{\mathscr{B}}$ linking two labelled copies x^r_α and y^s_β of x and y respectively, for α and $\beta \in \{1,2\}$. Assume w.l.o.g. that $x^r_\alpha = x^h_1$ and that $y^s_\beta = y^t_1$. Creating this gray edge in $\mathscr{B}(G)$ is equivalent to labelling the genome D so that the two following adjacencies are created : (x^h_1, y^t_1) and (x^h_2, y^t_2) (the other possibility being (x^h_1, y^t_2) and (x^h_2, y^t_1)). In order to end up with a labelling of D, at each step the created adjacencies should not be in conflict with the ones already created. Suppose this is not the case. In other words, the newly created adjacencies (x^h_1, y^t_1) and (x^h_2, y^t_2) lead to a conflict. This can happen in one of the two following situations :

– We have already created, in a previous step of the algorithm, an adjacency (z^u, y^t), with $z^u \neq x^h$ and $u \in \{t,h\}$. This is impossible as (z^u, y^t) would have been an adjacency of D. But since D is a duplicated genome, it can not involve two different adjacencies for a given gene extremity y^t.
– We have already formed, following the previous steps of the algorithm, a segment of adjacencies resulting in y_1 being on the left-side of x_1, i.e. a segment of form $+y_1 \ldots +x_1$. This means that we also have created the homologous segment $+y_2 \ldots + x_2$. Then having to add the gray edge (x_1^h, y_1^t) means that D is a duplicated circular genome of form $+y \ldots +x +y \ldots +x$. Therefore, adding the last two remaining gray edges (x_1^h, y_1^t) and (x_2^h, y_2^t) results in a bi-circular representation of the labelling $+y_1 \ldots +x_1 +y_2 \ldots +x_2$ of D.

□

Following the proof of Lemma 2, it is immediate to see that, if the set of strings represented by the gray edges of the output completed graph of Algorithm Double-Distance(G,D) is not a labelling of D, then changing the last chosen set of gray edges results in a correct labelling of D. More precisely, instead of adding the last two gray edges (x_1^h, y_1^t) and (x_2^h, y_2^t) the right choice would have been to add the two gray edges (x_1^h, y_2^t) and (x_2^h, y_1^t).

4.6.1.1 Complexity

As each vertex is adjacent to two black edges, finding an alternating cycle of size k beginning at a given vertex of $\mathscr{CB}(D,G)$ (line 5) can be done in $O(2^k)$ time. Therefore, the algorithm has a worst running-time complexity bounded by $\Sigma_{k=1}^{n} n.2^k$, which is not better than the exhaustive approach in $O(n.2^n)$. However, as demonstrated in the experimental part of this paper, it is actually a much faster approach in practice. This is due to the edge removal step (line 7), which allows to reduce the graph quickly, and to stop the process

after a small number of iterations.

4.6.2 Multichromosomal genomes

In the case of G and D being multichromosomal genomes, we define the contracted breakpoint graph $\mathscr{CB}(D,G)$ as before, except that it contains an additional vertex O such that any endpoint vertex in D is connected to O by two gray edges, and any endpoint vertex in G is connected to O by a black edge. It follows that, except for O that is adjacent to $2N_G$ black edges and $2N_D$ gray edges, N_G being the number of chromosomes of G, and N_D the number of chromosomes of D, each other vertex is adjacent to exactly two gray edges and two black edges.

Algorithm Double-Distance(G,D) can be used in the case of multichromosomal genomes if we replace line 3 with " **If** $\mathscr{CB}(D,G)$ is acyclic". At the end of the algorithm, $\mathscr{CB}(D,G)$ is acyclic and the only remaining paths connect two vertices that are both endpoints of G, or both endpoints of D (as a path connecting two endpoints of two different genomes would have been closed by Algorithm Double-Distance(G,D) to form a cycle). Then, to complete the graph $\mathscr{B}(G)$, it suffices to add the remaining paths of $\mathscr{CB}(D,G)$.

Due to the $2(N_G+N_D)$ edges incident to O, the worst-time complexity is the one for circular genomes multiplied by $N_G.N_D$, i.e. $O(n.N_G.N_D.2^n)$. Hopefully in practice, n is not a tight upper bound as exploration eventually stops for much smaller cycle sizes.

4.7 Double Distance with Losses

Similarly to genome halving, we aim to generalize the double distance to genomes with possibly missing gene copies. More precisely, given an RDL genome G and a duplicated genome $D = (D_{predup} \oplus D_{predup})$ on the same set of genes, how can we compute the distance between G and D ? Notice first that a simple generalization of the strategy used

for Algorithm Dedouble-RDL which would consist in (1) "gluing" the singletons of G to an adjacent gene and removing the corresponding copies from D, and (2) applying Algorithm Double-Distance to the obtained genomes G' and D', does not solve the problem. Indeed, the distance obtained after applying Algorithm Double-Distance(G',D') would only be a lower bound of the distance between genomes G and D. For example, suppose that an optimal sequence of rearrangements are performed on G' to transform it into D', and that the singletons are then unglued to form the DL genome A. The problem is that the perfectly duplicated genome D is not necessarily an extension of A.

As reducing the evolutionary model M_1 to M_2 does not help in this case, we instead reduce it to the symmetrical model M_3 (Figure 4.1.(c)), where all rearrangements occur first, followed by all losses. The next theorem justifies this reduction.

Theorem 3. *Let G be an RDL genome and D be a duplicated genome. Then there is an RD genome G' that is an extension of G, such that $d_R(G',D) = d_R(G,D)$.*

Proof. Proof by induction, very similar to the one of Theorem 1. □

The problem thus reduces to the one of finding an RD genome G' that is an extension of G, minimizing the weak double distance to D. Such a genome G' is called an *optimal extension of G*.

In the following section we focus on circular genomes. We subsequently explain, in section 4.7.2, the modifications that should be introduced in the case of multichromosomal genomes.

4.7.1 Circular genomes

The following lemma allows to limit the possible insertion positions of a missing gene in G.

Lemma 3. *Let x be a singleton in G, labelled x_1. Then, there is an optimal extension G' of G such that the inserted copy x_2 of x has a preserved (left or right) adjacency with D.*

Proof. Let G' be an optimal extension of G. More precisely, G' is an RD genome that is an extension of G and $d_R(G',D)$ is minimal over all RD genomes that are extensions of G. Let $\mathscr{R} = \{r_1,\ldots,r_p\}$ be an optimal sequence of inversions transforming G' into D. If G' is a labelled genome, then applying $\mathscr{R}$ to G' results in a labelled genome D. Let L_2 and R_2 be respectively the left and right adjacencies of x_2 in D. If x_2 is left-adjacent to one copy of L or right-adjacent to one copy of R in G', then the lemma is verified. Otherwise, consider the genome G''_{glue} obtained from G' by removing the copy x_2, and gluing it to either L_2 or R_2. W.l.o.g., let's say that x_2 is glued to L_2, i.e. L_2 is replaced by $L''_2 = L_2\,x_2$. Then, for each $r_i \in \mathscr{R}$ acting on the segment X_i, consider the inversion r'_i acting on the segment X'_i obtained from X_i by replacing L_2 by L''_2, and removing x_2 from X_i, if applicable. Then, applying the sequence of inversions $\mathscr{R}' = \{r'_1,\ldots,r'_p\}$ to G''_{glue} and then ungluing L''_2 gives rise to the genome D. Therefore, the genome G'' obtained by ungluing L''_2 in G''_{glue} is an optimal extension genome of G verifying the property that x_2 has a preserved adjacency with D. □

In other words, x_2 can be inserted in G adjacent to one of the two copies of its left or right neighbor in D. The problem is to find the appropriate neighbor and copy number, as this would influence the number of operations required to place x_1 adjacent to the other copies of its neighbors. The idea of the algorithm will therefore be to create the adjacencies for x_1 before those for x_2.

We consider a tricolored graph $\mathscr{CB}(D,G)$ obtained by adding to the contracted breakpoint graph representation, introduced in the previous section, a new set of "dotted edges" defined as follows : for each vertex y^s representing an extremity of a singleton y of G (for $s \in \{t,h\}$) , construct a dotted edge linking y^s to its adjacent vertex in D (see Figure 4.5.(a)

left). Algorithm Double-Distance-with-Loss(G,D), presented in Figure 4.4, takes as input the tricolored graph $\mathscr{CB}(D,G)$ and the partial breakpoint graph $\mathscr{B}(G)$, and completes $\mathscr{B}(G)$ with appropriate gray edges, but also additional black edges, corresponding to the missing singleton copies that have to be inserted in G to produce an extension G'.

In the following developments, as well as in Figure 4.4, alternating cycles only refer to cycles of black and gray alternating edge colors (i.e. dotted edges are not considered in the cycles). The algorithm proceeds as follows (see Figure 4.5 for an example) :

1. The For Loop 2 - 9 : Proceed as in Algorithm Double-Distance(G,D)), i.e. pick an alternating cycle of minimum size from $\mathscr{CB}(D,G)$, construct the corresponding cycle in $\mathscr{B}(G)$, and then remove from $\mathscr{CB}(D,G)$ all used edges. This step is performed as long as $\mathscr{CB}(D,G)$ contains an alternating cycle.

   It is easy to see that when we enter the For Loop with a graph $\mathscr{CB}(D,G)$ containing at least one cycle, and we leave this Loop with $\mathscr{CB}(D,G)$ being non-empty, then there is at least one single gray edge with a parallel dotted edge, which allows to enter the following For Loop.

2. The For Loop 13 - 31 : For each singleton extremity y^s that has been considered in the previous step (i.e. the adjacency of y^s_1 has been created in $\mathscr{B}(G)$), add the second copy of this singleton extremity (i.e. y^s_2) at the right place in $\mathscr{B}(G)$, and form the corresponding alternating cycle of size 1. Lines 14 - 17 allow to concatenate adjacent singletons into a single block, and lines 18 - 30 give all the details about the appropriate modifications (insertions and removals) of edges in $\mathscr{B}(G)$ and $\mathscr{CB}(D,G)$ resulting from the insertion of a singleton at the appropriate position in G.

Unfortunately, it happens (very rarely) that the graph $\mathscr{CB}(D,G)$ obtained as an output of the For Loop 13 - 31 is acyclic, which prevents any modification by the For Loop 2 -

Algorithm Double-Distance-with-Loss(G,D)
Input : $\mathscr{CB}(D,G)$ and the partial graph $\mathscr{B}(G)$;
Output : The partial graph $\mathscr{B}(G)$ completed (i.e. $\mathscr{B}(G,D^L)$), and an RD genome G' that is an extension of G ;

1. **While** $\mathscr{CB}(D,G)$ is not empty (i.e. has edges left) **Do**
2. **For** $CSize = 1$ to n **Do** ;
3. **For** $CVertex = b_1^l$ to b_n^l **Do**
4. **If** there is a cycle C of size $CSize$ beginning at $CVertex$ **Then**
5. Construct a corresponding cycle in $\mathscr{B}(G)$ (instructions 6 to 9 of Algo. Double-Distance) ;
6. Remove from $\mathscr{CB}(D,G)$ all edges of C ;
7. **End If**
8. **End For**
9. **End For**
10. **If** $\mathscr{CB}(D,G)$ does not contain any single gray edge **Then**
11. Choose an arbitrary gray edge (x^r, y^s) with a parallel dotted edge
12. **End If**
13. **For** each single gray edge (x^r, y^s) that has a parallel dotted edge *or* the chosen edge of Line 11 **Do**
14. **If** x and y are both singletons **Then**
15. Create the block B representing the adjacency (x^r, y^s)
16. Remove from $\mathscr{CB}(D,G)$ the two vertices x^r, y^s, and their adjacent edges ;
17. Replace in both $\mathscr{CB}(D,G)$ and $\mathscr{B}(G)$ the vertex $x^{\overline{r}}$ by $B^{\overline{r}}$ and the vertex $y^{\overline{s}}$ by $B^{\overline{s}}$;
18. **Otherwise** {*Among x, y, only one is a singleton*}
19. Let y be the singleton vertex ;
20. Let (x^r, z^u) be the remaining black edge of $\mathscr{CB}(D,G)$ adjacent to x^r ;
21. In $\mathscr{B}(G)$:
22. Remove the black edge (x^r, z^u) ;
23. Add the black edges (x^r, y^s) and $(y^{\overline{s}}, z^u)$;
24. Add the gray edge (x^r, y^s) ;
25. In $\mathscr{CB}(D,G)$:
26. Remove the black edge (x^r, z^u) ;
27. Remove the gray and dotted edges (x^r, y^s) ;
28. Remove the dotted edge adjacent to $y^{\overline{s}}$;
29. Add the black edge $(y^{\overline{s}}, z^u)$;
30. **End If**
31. **End For**
32. **End While**
33. **Return** (The genome G' deduced from the black edges of $\mathscr{B}(G)$) ;

Figure 4.4 – Algorithm Double-Distance-with-Loss(G,D). The notation $\overline{s}$ for $s \in \{t,h\}$ refers to the complement of s in this set. More precisely, if $s = t$ then $\overline{s} = h$ and if $s = h$ then $\overline{s} = t$. A "single gray edge" is a gray edge that has no parallel gray edge.

9, and does not allow then to re-enter the For Loop 13 - 31, as no single gray edge exists. This is the reason of Instructions 10 - 12, allowing to remove any gray edge with a parallel dotted edge, and then proceeding with the For Loop 13 - 31.

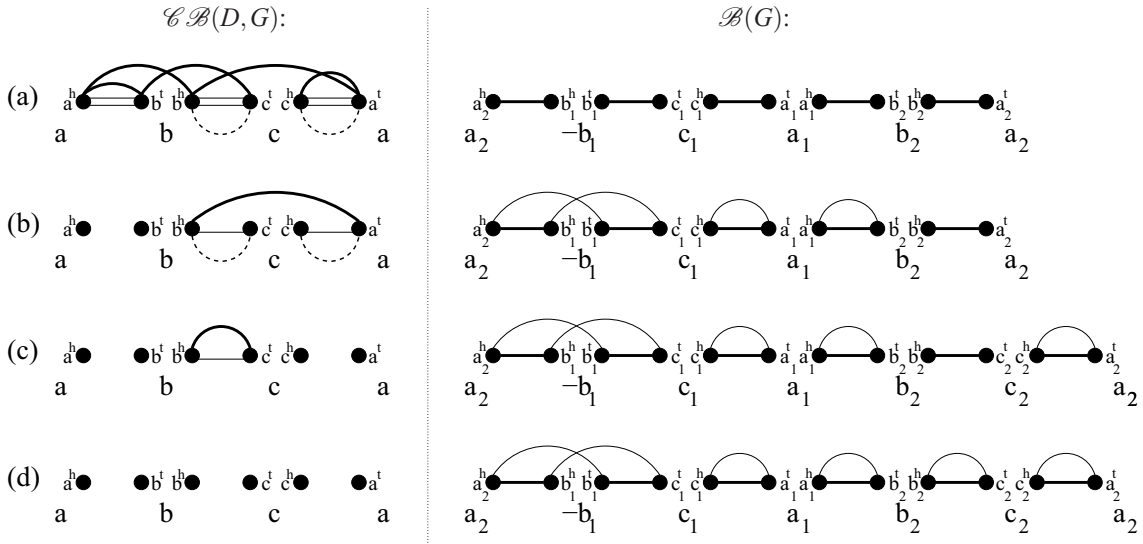

Figure 4.5 – Using Algorithm Double-Distance-with-Loss(G,D) on a simple example. An execution of Algorithm Double-Distance-with-Loss(G,D) with $D = (abc) \oplus (abc)$ and $G = (a\ -bcab)$. The evolution of the contracted breakpoint graph $\mathscr{CB}(D,G)$ and the partial breakpoint graph $\mathscr{B}(G)$ are shown respectively on the left and right sides of the figure. (a) The initial graphs. Gene c is a singleton and, in $\mathscr{CB}(D,G)$, each of its extremities is connected by a dotted edge to its adjacent vertex in D. (b) The current graphs after executing the For Loop 2 - 9. No more alternating cycles are present in $\mathscr{CB}(D,G)$. (c) The current graphs after executing the For Loop 13 - 31. The second copy of c is inserted in $\mathscr{B}(G)$ and edges are updated in both graphs. (d) The current graphs after a second execution of the For Loop 2 - 9. As $\mathscr{CB}(D,G)$ is empty (no edges left), the algorithm stops.

Lemma 4. *The completed graph $\mathscr{B}(G,D^L)$ output by* Algorithm Double-Distance-with-Loss *satisfies :*

1. *Its set of black edges represent an extension of G.*
2. *Its set of gray edges represent, either a labelling of D or a bi-circular representation of a labelling of D.*

Proof.

1. Follows from the fact that the algorithm ends up with an empty graph $\mathscr{CB}(D,G)$ (no edges remain in $\mathscr{CB}(D,G)$). Therefore, at the end, for each singleton in G, two vertices and one black edge have been added in $\mathscr{B}(G)$, leading to a genome G' with all missing copies of G inserted.

2. Same proof as for Lemma 2. □

4.7.2 Multichromosomal genomes

In the case of G and D being multichromosomal genomes, the right and/or left neighbor in D of a singleton gene can be a chromosome end (represented by O in the contracted breakpoint graph ; see section 4.6.2). Since O is a special node that is adjacent to the endpoint genes of all the chromosomes, some modifications have to be made to Algorithm Double-Distance-with-Loss(G,D) for multichromosomal genomes.

The special node O is not considered as a singleton node in the contracted breakpoint graph, so the only part of Algorithm Double-Distance-with-Loss(G,D) that has to be changed is between lines 18 - 30. When working on the dotted edge (x^r, y^s), if $x^r = O$, then there are two possibilities : the singleton gene y is, in D, (1) adjacent to a gene and a chromosome end, or (2) adjacent to two chromosome ends (i.e. y is the only gene on a chromosome). In the first case, we can simply work on the dotted edge adjacent to $y^{\overline{s}}$, which is the dotted edge representing the other adjacency of y in D. In the second case, both y^s and $y^{\overline{s}}$ are connected to O by a dotted edge. Algorithm Double-Distance-with-Loss(G,D) can then be used on any of these two dotted edges if we skip lines 20, 22 and 26 and set $z^u = O$.

4.8 Results

Since the generalization of the genome halving problem to a present-day RDL genome has been proved to be an exact algorithm executing in linear time, we only test the performance of the proposed method to compute the double distance. We generated datasets through simulated evolutions between a duplicated genome D and an RD or RDL genome G for both circular and multichromosomal genomes, as follows.

Simulated datasets : We first determine n, the number of genes, and N, the number of chromosomes in D_{predup}. We also define l, the percentage of genes in D_{predup} that is lost after the WGD. Then, we generate D by applying a WGD, and a series of rearrangement and/or loss events are performed on D to obtain G. The rearrangements are simply the ones allowed by our model, namely inversions only in the case of circular genomes or inversions and translocations (including fusions and fissions) in the case of multichromosomal genomes. The number of rearrangement events, μ, is a parameter chosen prior to the data generation, and the size of each rearrangement is chosen randomly. As for the rates of rearrangement operations, we chose (Inv : Trans : Fus+Fiss) = (5 : 4 : 1) to follow the rates reported for a lineage where a WGD occurred [74].

In order to validate the distances obtained with our greedy approach, we use an exact algorithm described below.

Exact algorithm : If G is an RDL genome, we generate all possible RD genomes by reinserting the missing gene copies at all possible positions following Lemma 3. The following algorithm can be used with RD genomes. Let L (*resp.* L^*) be a complete (*resp.* partial) labelling of the gene copies of D, and $\mathscr{B}(G, D^{L^*})$ the breakpoint graph where the only defined gray edges are those adjacent to the genes of L^*. The idea is to compute a lower bound for $d_R(G,D)$ as we progressively construct L^*. More precisely, if at one step we

have c cycles and p paths in $\mathscr{B}(G,D^{L^*})$, we know that the number of cycles in $\mathscr{B}(G,D^L)$ will be at most equal to $c+p$. Thus it is possible to use the following lower bound in a branch and bound strategy : $d_R(G,D) \geq n-c-p$.

Due to the high running-time complexity of the exact method, validation with the exact distance can only be done for "simple" datasets obtained with a low number of genes, a low number of rearrangements, and a maximum of two singletons. For datasets that were too complex for the exact algorithm, we estimated the accuracy of our greedy algorithm for the double distance by comparing the inferred distance with the number of rearrangements performed between D and G in the simulated evolution.

4.8.1 Time efficiency

Since the running-time complexity is a function of n for the exact approach, we generated genomes containing different numbers of genes to evaluate the time efficiency of our greedy heuristic. For the exact method, n varies from 10 to 100, with an increment of 10. The parameters μ, N and l are arbitrarily fixed to 15, 4 and 0 respectively. For Algorithm Double-Distance-with-Loss(G,D), n varies from 100 to 1000 with an increment of 100 and we plotted the results for l equal to 0, 10 and 50%. With μ fixed to 15, the running-time of Algorithm Double-Distance-with-Loss(G,D) does not vary (below 0.001 seconds for all values of n). Thus, the number of rearrangements has been changed to $\mu = n$ in order to see a variation in the running-time. For each of those n values, multiple datasets were generated and the running time was averaged.

We can clearly observe the exponential running-time of the exact approach when the number of genes increases (see Figure 4.6 left). In contrast, Algorithm Double-Distance-with-Loss(G,D) is less limited by the genome size and more by the number of rearrange-

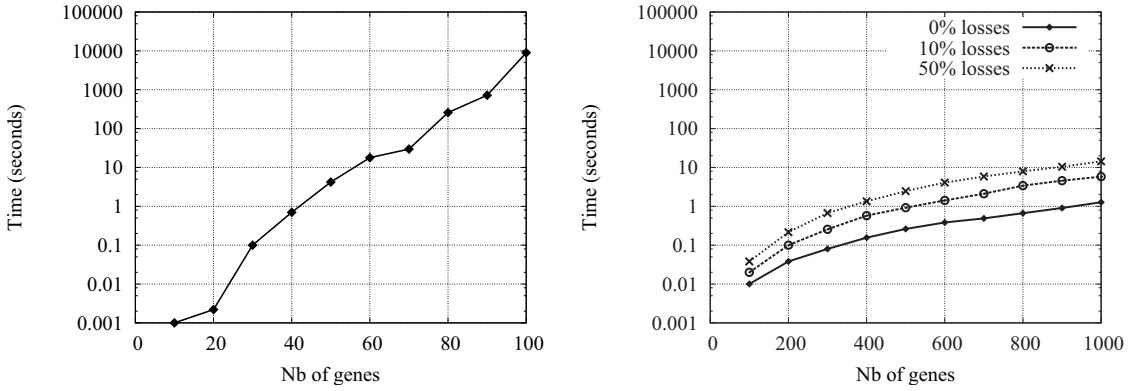

Figure 4.6 – Running-time experiments. Left : Running-time of the exact algorithm computing the double distance without losses between D and G with various number of genes and a fixed number of rearrangements (μ=15). Right : Running-time of Algorithm Double-Distance-with-Loss(G,D) to compute the double distance with various number of genes and rearrangements ($\mu = n$) and different gene loss percentages.

ments. In Figure 4.6 right, we can see that even for datasets with a high number of rearrangements ($\mu = n$), the running-time, for 0% losses, remains under or close to 1 second. Obviously, the more losses there are in G, the slower is the algorithm, but the running-time remains less than the anticipated worst-time complexity.

4.8.2 Heuristic accuracy

4.8.2.1 Comparison with the exact approach

We now test whether Algorithm Double-Distance-with-Loss(G,D) infers an accurate rearrangement distance by comparing its results against those of the exact approach. Recall that because of the high running-time complexity of the exact approach, we can only perform this algorithm on simple datasets exhibiting low numbers of genes, rearrangements and losses. The genomes were generated with n fixed to 25, N to 4 for multichromosomal genomes, μ varying from 0 to 50 by increments of 5 and zero, one or two singletons. For

each value of μ, 500 datasets were simulated. The error rate is the proportion of datasets for which the exact method found a more accurate distance than Algorithm Double-Distance-with-Loss(G,D). Results are averaged over all datasets showing a comparable number of rearrangement events.

As observed in Figure 4.7, the error rate of Algorithm Double-Distance-with-Loss(G,D) is close to 0 when the number of rearrangements is less than 15. Moreover, the distance inferred by Algorithm Double-Distance-with-Loss(G,D) is on average really close to the optimal distance for both types of genomes (circular and multichromosomal). In fact, when the distance is not the same, it differs on average by 1 rearrangement and at most by 2 (which occurred only once in our simulations). Naturally, the error rate of Algorithm Double-Distance-with-Loss(G,D) is more apparent when the number of rearrangements and losses increases. This behavior is due to the fact that when a high number of rearrangements is performed, different cycles of equal size can be selected and a choice must be made affecting the remaining set of cycles. The presence of missing gene copies will also produce more errors because the reinsertion procedure introduces more choices. As stated before, in this experiment we seek to optimize the rearrangement distance, but we obtain similar results if we seek to optimize the DCJ distance (results not shown).

4.8.2.2 Complex datasets

As a final experiment, simulations were performed with $n = 1000$, $N = 8$ for multichromosomal genomes, μ varying from 0 to 1000, and l equal to 0, 25 and 50%. The distances obtained with Algorithm Double-Distance-with-Loss(G,D) are compared with μ. Results shown in Figure 4.8 demonstrate that our method infers distances close to the number of rearrangement events performed on the original genome (for circular and multichromosomal genomes). However, when the number of rearrangement events increases,

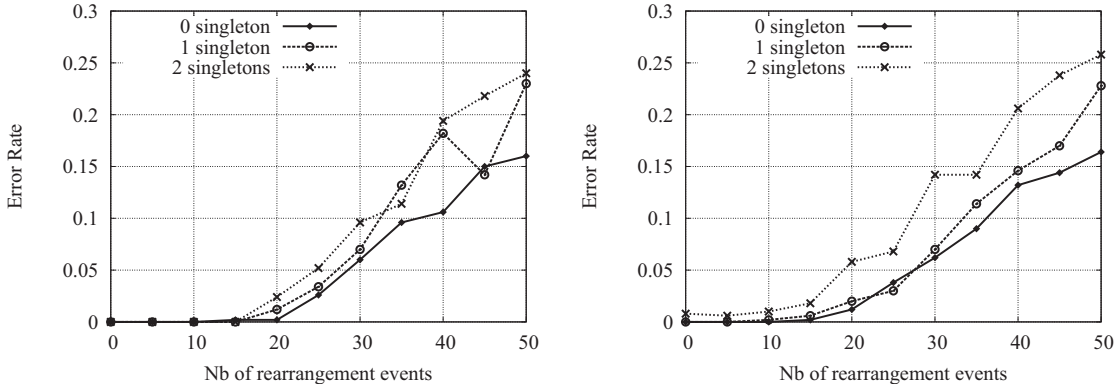

Figure 4.7 – Error rates. Comparison of Algorithm Double-Distance-with-Loss(G,D) with the exact approach for genomes of size 50 right after the WGD, showing the error rate of the inferred rearrangement distance for circular (left) and multichromosomal genomes (right). Error rates were computed for genomes with zero, one and two singletons.

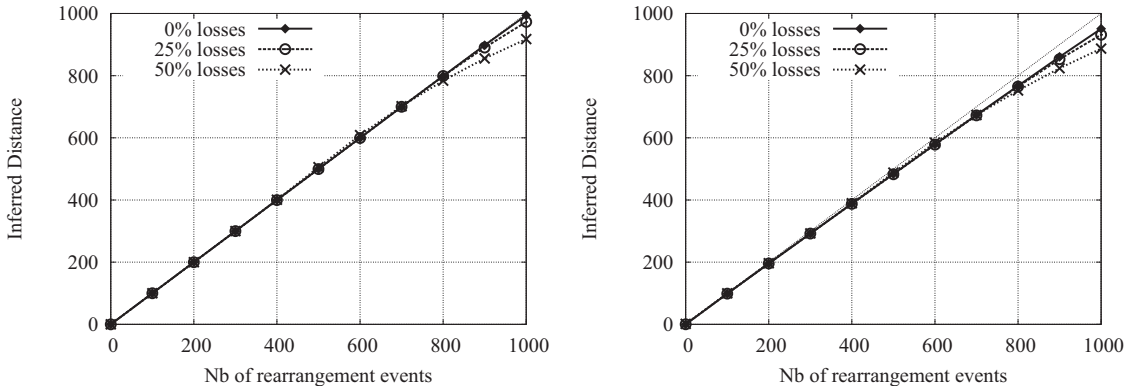

Figure 4.8 – Comparison of the inferred rearrangement distances with the real number of rearrangements. Inferred rearrangement distances with complex datasets (n = 1000) and different gene loss percentages, for circular genomes (left) and multichromosomal genomes (right).

our approach underestimates that value. Notice that the more losses there are, the lesser the distance is because we reinsert the missing gene copies next to one of their adjacent genes in D. As in the comparison with the exact approach, the results are similar with the DCJ distance (not shown).

4.9 Conclusion

We presented a linear time algorithm to solve the genome halving problem for genomes with missing gene copies. We also presented a greedy heuristic (Algorithm Double-Distance(G,D)) to compute the distance between an RD genome G and a duplicated genome D for the rearrangement and DCJ distances. Finally, we generalized this algorithm so that genome G can be an RD or RDL genome (Algorithm Double-Distance-with-Loss(G,D)). Our experiments on simulated datasets showed that Algorithm Double-Distance-with-Loss(G,D) is time-efficient and accurate.

The proposed heuristic for the double distance could be adapted to genomes that have undergone more than one WGD, thus increasing the running-time complexity as the number of possible labellings for a gene would increase. Our algorithm could then be used for the rearrangement phylogeny problem with genomes that have evolved through one or more whole genome duplications. Indeed, this method would allow to compute distances efficiently on all branches of such a phylogeny and consequently, an algorithm for the median problem could be used on the tree.

Another interesting future work will concern the generalization of Algorithm Double-Distance-with-Loss(G,D) for genomes G and D both being RD or RDL genomes. The current approach, using the contracted breakpoint graph, can not be used directly when the genome D is not a perfectly duplicated genome.

CHAPITRE 5

EVOLUTION OF ORTHOLOGOUS TANDEMLY ARRAYED GENE CLUSTERS

Olivier Tremblay Savard[1], **Denis Bertrand**[2] **et Nadia El-Mabrouk**[1]

Article publié dans le journal *BMC Bioinformatics* en 2011 [145].

5.1 Contributions

Olivier Tremblay Savard, Denis Bertrand et Nadia El-Mabrouk ont conçu l'algorithme. **Olivier Tremblay Savard** a implémenté l'algorithme. **Olivier Tremblay Savard** a conçu et réalisé les expériences sur les données simulées. **Olivier Tremblay Savard** et Denis Bertrand ont conçu et réalisé les expériences sur les données biologiques. **Olivier Tremblay Savard** et Nadia El-Mabrouk ont rédigé l'article. Denis Bertrand a participé à la révision du manuscrit.

[1]DIRO, Université de Montréal, Canada
[2]Computational and Mathematical Biology, Genome Institute of Singapore, Singapour

5.2 Abstract

Background : Tandemly Arrayed Gene (TAG) clusters are groups of paralogous genes that are found adjacent on a chromosome. TAGs represent an important repertoire of genes in eukaryotes. In addition to tandem duplication events, TAG clusters are affected during their evolution by other mechanisms, such as inversion and deletion events, that affect the order and orientation of genes. The DILTAG algorithm developed in [97] makes it possible to infer a set of optimal evolutionary histories explaining the evolution of a single TAG cluster, from an ancestral single gene, through tandem duplications (simple or multiple, direct or inverted), deletions and inversion events.

Results : We present a general methodology, which is an extension of DILTAG, for the study of the evolutionary history of a set of orthologous TAG clusters in multiple species. In addition to the speciation events reflected by the phylogenetic tree of the considered species, the evolutionary events that are taken into account are simple or multiple tandem duplications, direct or inverted, simple or multiple deletions, and inversions. We analysed the performance of our algorithm on simulated data sets and we applied it to the protocadherin gene clusters of human, chimpanzee, mouse and rat.

Conclusions : Our results obtained on simulated data sets showed a good performance in inferring the total number and size distribution of duplication events. A limitation of the algorithm is however in dealing with multiple gene deletions, as the algorithm is highly exponential in this case, and the problem becomes quickly intractable.

5.3 Background

Gene duplication is a fundamental process in the evolution of species [125], especially in eukaryotes [25, 35, 48, 77, 107, 170], where it is believed to play a leading role for the creation of novel gene functions. Several mechanisms are at the origin of gene duplications, among them tandem repeat through unequal crossing-over during recombination. As this phenomenon is facilitated by the presence of repetitive sequences, a single duplication can induce a chain reaction leading to further duplications, eventually creating large *Tandemly Arrayed Gene (TAG) clusters* : groups of paralogous genes that are adjacent on a chromosome. TAGs account for about one-third of the duplicated genes in eukaryotes [194]. In human, they represent about 15% of all genes [150] . In *Arabidopsis*, 17% of the total predicted genes are members of TAG clusters [161], and in maize, about 35% of the genes were predicted to belong to TAG clusters [117].

Deciphering the evolutionary history of a TAG cluster is important to provide new insights into the mechanisms of gene amplification, and to answer several questions regarding the nature and size of duplication and other evolutionary events that have shaped TAG clusters. In most biology-oriented studies, a gene tree is obtained by applying a classical phylogenetic method to an alignment of the amino acid sequences corresponding to the collected gene sequences, and a duplication scenario is proposed for the gene family, based on a careful analysis of this gene tree (see for example [194] for the study of the 22-kDA prolamin gene amplification in grass genomes). Although such manual analysis may be useful to propose amplification scenarios for families of limited size and simple organization, it is usually impractical to infer more general evolutionary scenarios for large TAG clusters affected, in addition to duplications, by other events such as segmental deletion, that may lead to gene loss, and rearrangements (such as inversions or inverted duplications), that may affect gene order and transcriptional orientations.

The *tandem-duplication model of evolution*, first introduced by Fitch in 1977 [61], assumes that, from a single ancestral gene at a given position in the chromosome, the locus grows through a series of consecutive duplications placing the newly created copy next to the original one. Such tandem duplications may be *simple* (duplication of a single gene) or *multiple* (simultaneous duplication of neighboring genes). Based on this idea, a number of theoretical studies have considered the problem of reconstructing the tandem-duplication history of a TAG cluster [20, 53, 158, 185]. However, due to rearrangements and losses, it is often impossible to reconstruct a duplication history for a TAG cluster [67], even from well-supported gene trees.

In [98], Lajoie *et al.* considered a generalization of the tandem-duplication model allowing for inversions. The model was then extended in [21] to the study of orthologous TAG clusters in different species. A similar work, considering more operations (translocations, fusions, fissions, duplications in tandem or not), but requiring more preliminary information (gene and species trees with branch length) has also been done [112]. Various other heuristic and probabilistic methods have been developed for reconstructing a hypothetical ancestral sequence and a most parsimonious set of duplications (in tandem or not) and other evolutionary events leading to the observed gene cluster [152, 168, 187, 188]. They are based on a preprocessing of a self-alignment dot-plot of a cluster, or the dot-plot of a pairwise-alignment of two clusters. Although these methods are useful to infer evolutionary events in well-conserved regions, they are less appropriate when there is a lot of noise in the dot-plots due to the alignments of nunfunctional regions which are continuously affected by mutations.

In both methods developed by our group [21, 98], only simple duplications were considered. This assumption, while allowing for exact algorithmic solutions, is an important limitation to its applicability (see for example [99]). For this reason, Lajoie *et al.* have developed a more general heuristic, the DILTAG algorithm [97], allowing us to infer a set of

optimal evolutionary histories for a gene cluster in a single species, according to a general cost model involving variable length duplications, in tandem or inverted, deletions and inversions. Experiments on simulated data showed that the most recent evolutionary events can be inferred accurately when the exact gene trees are used. Despite the uncertainty associated with the deeper parts of the reconstructed histories, they can be used to infer the duplication size distribution with some precision. DILTAG has been used recently in [65] to infer an evolutionary scenario for the Maltase gene clusters in *Drosophila*.

A clear limitation of DILTAG is the fact that it is applicable only to a single cluster. The benefit of an extension to multiple species is obvious, as comparative genomics is clearly a more appropriate approach to infer loss and inversion events. In particular, considering an outgroup may help in choosing among many possible optimal evolutionary scenarios for a gene cluster.

In this paper we present an extension of DILTAG to the study of a set of orthologous TAG clusters in multiple species. In other words, in addition to multiple duplication (in tandem or inverted), deletion and inversion events, the speciation events reflected by a given phylogenetic tree for the set of species are also taken into account. We develop Multi-DILTAG, a heuristic algorithm that is shown on simulated data sets to be very accurate in inferring the total number and size distribution of duplication events.

5.4 Methods

5.4.1 Data

Preliminary to all the developments in this paper is the identification of m orthologous TAG clusters in m genomes of interest. In other words, given a gene family F of interest, a tandemly arrayed sequence (called TAG cluster) of paralogous genes from F has already been identified in each genome, and such m TAG clusters have already been pointed out

as orthologs. For example, gene orders and clusters orthology for the protocadherin gene family has been identified for human and several other mammalian and fish species [178, 180].

We denote by $\mathscr{O} = \{O_1, O_2, \ldots, O_m\}$ the set of m TAG clusters, *i.e.* for $1 \leq i \leq m$, O_i is the signed order of the family members in genome i. The sign $(+/-)$ of a gene represents its transcriptional orientation.

In addition to the observed gene orders, we also assume that a gene tree is available for the *TAG family*, *i.e.* the set of genes contained in the m TAG clusters. A *gene tree* T for a TAG family is a rooted binary tree with labelled leaves, where each label represents an unsigned gene copy. A leaf labelled by a gene copy in genome i is said to *belong to genome* i. For conciseness, we make no distinction between a leaf and its label. The pair $(T, \mathscr{O})$ is called the *ordered gene tree* for the gene family. Finally, we assume that the species tree, reflecting the speciation history of the m considered genomes, is also available. See Figure 5.1 for an example.

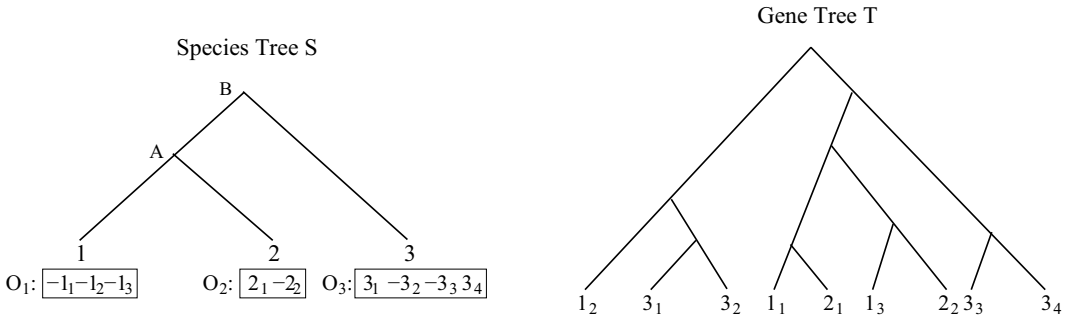

Figure 5.1 – Species and gene trees for the three genomes 1, 2, 3. Left : The species tree for three genomes $1,2,3$. Three orthologous TAG clusters $\{O_1, O_2, O_3\}$ are identified in the three genomes. The notation j_i denotes the ith gene in genome j. Right : A gene tree for the gene family.

5.4.2 The Evolutionary Model

Our evolutionary model is an extension of the one introduced by Fitch [61] for TAGs, which considers only tandem duplications resulting from unequal crossing-over during meiosis. However, TAGs are shaped during their evolution by other events affecting the gene order, orientation and content of the clusters. For example, Shoja and Zhang [150] have observed that more than 25% of all neighboring pairs of TAGs in human, mouse and rat have non-parallel orientations. The Fitch model of evolution does not apply to such data. Our model extends the Fitch model of evolution by considering deletion events affecting gene content, as well as inversion and inverted duplication events affecting gene orientation.

Below is a formal definition of the evolutionary model considered in this paper. In this definition, a *cherry* of T is a pair of leaves (l,r) separated by a single vertex, called its *root*.

Definition 1 : An *evolutionary history* for $(T,\mathscr{O})$ is a sequence of ordered gene trees $((T^1,\mathscr{O}^1),(T^2,\mathscr{O}^2),\ldots,(T^h,\mathscr{O}^h)=(T,\mathscr{O}))$, such that for each $1\leq k\leq h$, $\mathscr{O}^k=\{O_1^k,\ldots,O_i^k,\ldots,O_{n_k}^k\}$ is a set of n_k gene orders corresponding to orthologous TAG clusters on n_k genomes, where :

1. T^1 is a tree consisting of a single leaf u, and $\mathscr{O}^1=\{O_1^1\}=\{(\pm u)\}$.
2. For $1\leq k<h$, there is a unique genome i such that $(T^{k+1},\mathscr{O}^{k+1})$ can be obtained from $(T^k,\mathscr{O}^k)$ by applying one of the following evolutionary events on (T^k,O_i^k) :
   (a) **Duplication :** A sub-sequence $(u_p,u_{p+1},\ldots,u_q)$ of O_i^k is replaced by a sequence of new elements $(l_p,l_{p+1},\ldots,l_q,r_p,r_{p+1},\ldots,r_q)$, where, for each $p\leq x\leq q$, l_x and r_x have the same sign as u_x. Moreover, each leaf u_x in T^k is replaced by the cherry (l_x,r_x).

(b) **Inverted-duplication :** A sub-sequence $(u_p, u_{p+1}, \ldots, u_q)$ of O_i^k is replaced by $(-(l_q), -(l_{q-1}), \ldots, -(l_p), r_p, r_{p+1}, \ldots, r_q)$ or $(l_p, l_{p+1}, \ldots, l_q, -(r_q), -(r_{q-1}), \ldots, -(r_p))$, where, for each $p \leq x \leq q$, l_x and r_x have the same sign as u_x. Moreover, each leaf u_x of T_k is replaced by the cherry (l_x, r_x).

(c) **Inversion :** A sub-sequence $(u_p, u_{p+1}, \ldots, u_q)$ of O_i^k is replaced by $(-(u_q), -(u_{q-1}), \ldots, -(u_p))$ and T^k remains unchanged.

(d) **Deletion :** A sub-sequence $(u_p, u_{p+1}, \ldots, u_q)$ of O_i^k is deleted, and the corresponding leaves (genes) are removed from T^k (each removed gene corresponds to a gene loss).

(e) **Speciation :** The complete order $O_i^k = (u_1, \ldots, u_t)$ is replaced by $\{(l_1, \ldots, l_t), (r_1, \ldots, r_t)\}$, where, for each $1 \leq x \leq t$, l_x and r_x have the same sign as u_x. Moreover, each leaf u_x belonging to genome i is replaced by the cherry (l_x, r_x).

Any evolutionary history $\mathscr{H}$ for $(T, \mathscr{O})$ induces a unique species tree S obtained from the speciation events of $\mathscr{H}$. We say that $\mathscr{H}$ is *consistent with S*.

Finally, a *simple-event* will refer to an event acting on a single gene. For example, a simple-deletion will refer to the deletion of a single gene. A simple-deletion event is also referred to as a *loss event*. Moreover, a *general-duplication* will refer to a duplication that does not necessarily place the duplicated genes next to the original copies (not necessarily in tandem). An example of an evolutionary history is given in Figure 5.2.

We are now ready to formulate our optimization problem :

MINIMUM-EVOLUTION PROBLEM :

Input : An ordered gene tree $(T, \mathscr{O})$ and a species tree S.

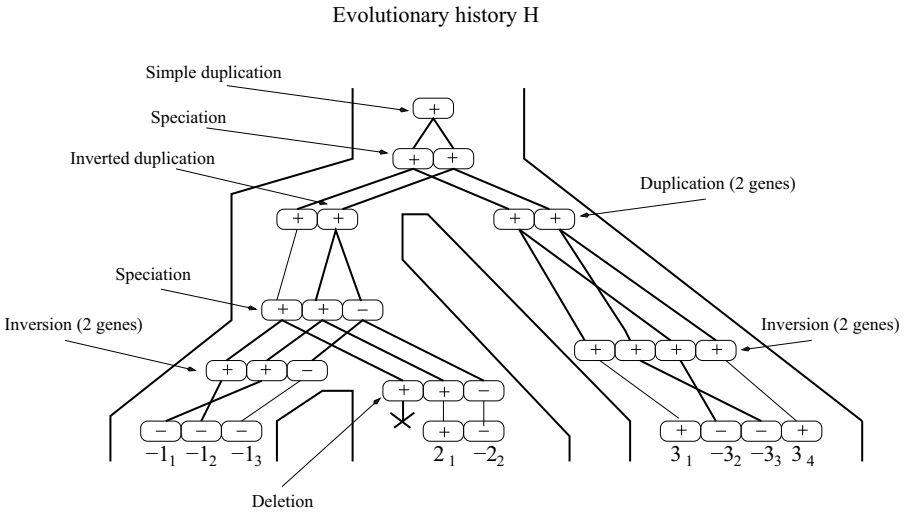

Figure 5.2 – An evolutionary history leading to the gene tree of Figure 5.1. This history is consistent with the species tree S of Figure 5.1.

Output : A most parsimonious evolutionary history $\mathscr{H}$ for $(T, \mathscr{O})$ consistent with S.

The "most parsimonious" constraint given above can be most naturally expressed in terms of number of events. Alternatively, a cost can be associated to each event depending on the type and size of the event (*i.e.* number of genes affected by this event), and the "most parsimonious" history would be the history of minimum cost, where the cost of a history is simply the sum of costs associated with its events. This latter approach is the one considered in [97].

5.4.3 The DILTAG method

The DILTAG algorithm [97] allows the inference of a set of most parsimonious histories of duplications, inverted-duplications, inversions and deletions (*i.e.* all events introduced in Definition 1 except speciation), originally acting on a single ancestral gene to produce a given extant TAG cluster represented by a given ordered gene tree (T, O).

DILTAG proceeds by exploring a "history graph" (search space), where vertices correspond to ordered gene trees and edges correspond to evolutionary events. More precisely, an edge from (T^i, O^i) to (T^j, O^j) is defined if and only if (T^i, O^i) can be transformed into (T^j, O^j) through one event, and each edge is weighted by the cost of its corresponding event. This graph is actually simplified into a finite graph, without loss of information, by considering deletions only in combination with duplication events. The history graph is constructed backwards, *i.e.* starting at vertex (T, O), and constructing edges in their opposite direction (backward-edges) by exploring the neighborhood of each vertex.

It is shown in [97] that, given a vertex representing an ordered gene tree (T, O), its *duplication* and *inverted-duplication* neighborhoods are both linear (in the size of T) in space, whereas its *inversion*, *duplication-with-deletion* and *inverted-duplication-with-deletion* neighborhoods are all quadratic in space. However the size of the whole search-space is clearly exponential, which makes an exhaustive search through the whole graph impossible for gene trees of reasonable size. A greedy heuristic is therefore developed that only keeps, in a queue, the most promising partial evolutionary histories obtained after exploring a given depth of the history-graph.

The input of DILTAG is an ordered gene tree (T, O) with n leaves, and the output is a set of shortest backward-paths in the history graph from (T, O) to a tree containing a single vertex. For the purpose of our new Multi-DILTAG algorithm, it is easy to modify DILTAG in order to reach an ancestral genome with g genes, for any $1 \leq g \leq n$: simply stop the procedure as soon as we attain the right number of genes. Notice that the attained ancestor is ordered, *i.e.* defined by an ordered sequence of g genes. It can be seen as an ordered tree (T', O') with T' being reduced to a set of g vertices and no edges. We will make no distinction between an ordered tree with no edges and a gene order.

In Section 5.4.5, the input and output of DILTAG will be as follows :

Input : An ordered gene tree (T, O) and a number g of ancestral genes ;
Output : The cost of a shortest backward-path from (T, O) to an ancestral genome with g genes, together with the *solution graph* composed by the actual set of shortest paths, and the *solution set* of ancestral gene orders attained.

Finally, we need the following definition for the subsequent developments : given two vertices x and y of the oriented history graph, if there is an edge oriented from x to y (there is an evolutionary event transforming x into y), then we say that y is a *predecessor* of x.

5.4.4 A two step method for multiple species

Back to our evolutionary model on multiple species, we aim to find a most parsimonious evolutionary history for $(T, \mathscr{O})$ that is consistent with S. This problem has been considered in [21], but in the more restricted case of *simple-duplications*, and no *inverted-duplications*. A two step methodology has been considered :

1. Reconciliation Step : Ignoring gene orders, infer a history of *simple-general-duplication, simple-deletion and speciation* for T consistent with S, by using a reconciliation approach [71]. Conceptually, a *reconciliation* R between a gene tree T and a species tree S is a tree accounting for the evolutionary history of the species and all genes of the gene family, including lost and missing gene copies, by simple-general-duplication, speciation and loss. R can be "embedded" into S, reflecting the duplication and deletion events leading to the observed tree T. Such embedding allows to infer the number of genes at the speciation nodes of S, as well as the evolutionary relationships between ancestral gene copies. A reconciliation between the gene tree T and the species tree S of Figure 5.1 is given in Figure 5.3. Notice that

this reconciliation does not lead to the observed gene order.

2. Minimization Step : Reinserting the gene order and sign information $\mathscr{O}$ on the leaves of S, infer the order and sign of genes at internal nodes of S allowing to minimize the total number of events involved in a history of $(T, \mathscr{O})$.

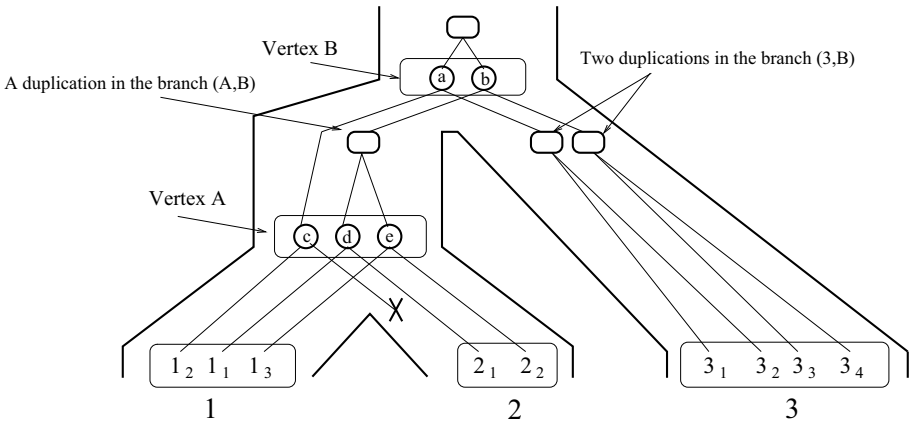

Figure 5.3 – A reconciliation R between the gene tree and the species tree of Figure 5.1. Ovals are speciation vertices and squares are duplication vertices. The crossed line indicates a gene loss. The genome sets of internal nodes are $G(A) = \{c,d,e\}$ and $G(B) = \{a,b\}$. As for the pre-speciation sets, we have $PG(A) = G(B) = \{a,b\}$, $PG(1) = G(A) = \{c,d,e\}$, $PG(3) = G(B) = \{a,b\}$, but $PG(2) = \{d,e\}$, which is different from $G(A)$ due to the loss of gene c.

We use the same two-step methodology here. As for the first step, any existing reconciliation method can be used. In particular, the so called Lowest Common Ancestor (LCA) mapping between a gene tree and a species tree, formulated in [76, 130] and widely used [26, 30, 44, 55, 75, 111, 128–130, 186], defines a reconciliation tree R that minimizes both the simple-general-duplication and simple-deletion events.

In the following developments, we will consider the "embedded" representation of a reconciliation tree R into the species tree S. More precisely :

– A *leaf* of R is an extant gene and maps to a leaf of S, *i.e.* the extant genome to which it belongs.

– A *duplication vertex* of R is an internal vertex which corresponds to a duplication event. It maps to a branch of S, *i.e.* the lineage in which the duplication occurred (see Figure 5.3).
– A *speciation vertex* of R is an internal vertex which corresponds to an ancestral gene at the time of a speciation event. It maps to an internal vertex of S, *i.e.* the ancestral genome to which it belongs. It has either one child (in the case of a gene loss), or two children each belonging to a different lineage. The set of speciation vertices mapping to a vertex A of S is the *genome set* $G(A)$ *of* A. If A is not the root, let B be the father of A. Then the *pre-speciation genome set* $PG(A)$ *of* A is the subset of $G(B)$ containing the vertices of $G(B)$ with a child in the branch (A,B), in other words, the genes in $G(B)$ that have not been lost after speciation on the branch going to A. We have $|PG(A)| \leq |G(B)|$ (see Figure 5.3).

Considering now the Minimization Step, if only *simple-duplications* are allowed, the problem has been shown in [21] to be equivalent to the one of finding gene orders at internal nodes of S minimizing a global inversion distance. In this context, the evolutionary model can be reduced to the one where all duplications occur first, followed by all inversions. The problem is then to find the minimum number of inversions, yielding a forest of simple-duplication trees. Using properties of simple-duplication trees, it is possible to define an exact and efficient algorithm for this problem. All these simplifications and shortcuts do not hold anymore for simultaneous duplications and deletions of multiple genes. In the following section, we focus on the Minimization Step.

5.4.5 Multi-DILTAG : Extension of DILTAG to multiple species

Our algorithm is a generalization of DILTAG that proceeds with the whole species tree S and produces a solution set for each internal vertex, and a solution graph with additional

speciation edges. Figure 5.4 illustrates the algorithm execution at each internal vertex A of S.

Initially, the solution set of each leaf is reduced to the gene order observed at that leaf, and the solution graph is reduced to the set of vertices defined by the ordered gene trees at the leaves. We then extend the solution graph by exploring S bottom-up, and for each internal vertex A, we compute a solution set $\mathscr{S}_A$ by performing DILTAG respectively on the left branch (A_l, A) and right branch (A_r, A) of S (with A_l and A_r being respectively the left and right child of A), and taking, as potential orders at A, the union of genome sets $PG(A_l)$ and $PG(A_r)$ obtained respectively in the left and right branch. However, due to gene losses, gene orders in $PG(A_l)$ do not necessarily have the same number of genes as gene orders in $PG(A_r)$. We therefore consider all possible extensions of gene orders, by reinserting lost copies in any possible way, and take the union of all sets obtained as the solution set $\mathscr{S}_A$. We then define a single "speciation edge" in the solution graph from each vertex representing a gene order in $\mathscr{S}_A$ to each vertex representing a gene order in $PG(A_l) \cup PG(A_r)$. As the only evolutionary events likely to have occurred on these edges of the history graph are inversions and deletions, we label each speciation edge (x, y) by the minimum Inversions+Deletions (ID) distance allowing to transform x into y. In the literature, the problem of computing the ID-distance between two permutations has already been considered, and a polynomial-time algorithm exists [52, 114].

More precisely, the Multi-DILTAG algorithm traverses the tree bottom-up, and for each internal node A proceeds as follows :

1. For each of $s \in \{l, r\}$, execute DILTAG on each element of $\mathscr{S}_{A_s}$, and stop as soon as the attained gene order contains $|PG(A_s)|$ genes. The set of all ancestral gene orders obtained (output of DILTAG) form an initial *pre-speciation* set $\mathscr{PS}_{A_s}$, further truncated as follows : if MIN is the minimum cost obtained over all elements of $\mathscr{S}_{A_s}$, we

remove from $\mathscr{PS}_{A_s}$ all elements O that are not attained with the cost MIN. Moreover, we remove from the partial current solution graph all the predecessors of O that are not linked to another element of $\mathscr{PS}_{A_s}$ by a minimum-cost path.

2. For each of $s \in \{l, r\}$, construct the set $\mathscr{PS}'_{A_s}$ by replacing each gene order O of $\mathscr{PS}_{A_s}$ by the set of all possible orders obtained from O by inserting the genes lost on the branch (A, A_s).
3. Compute $\mathscr{S}_A = \mathscr{PS}'_{A_l} \cup \mathscr{PS}'_{A_r}$. The solution graph is extended by adding one vertex per each element of $\mathscr{S}_A$.
4. Let $O \in \mathscr{S}_A$, and suppose, w.l.o.g. that $O \in \mathscr{PS}'_{A_l}$. Then complete the solution graph by constructing an oriented "speciation edge" from O to the vertex corresponding to its originating order in A_l, and an oriented edge from O to the vertex corresponding to each element of $\mathscr{PS}_{A_r}$ giving rise to the minimum ID-distance with O.

5.5 Results and Discussion

We implemented our algorithm and applied it to simulated data sets to evaluate its execution time and precision in terms of the number and size distribution of the inferred duplications. Then, we applied it to the protocadherin gene clusters of four mammalian species to infer the duplication size distribution and the number of events that occurred in the evolutionary history of these species.

5.5.1 Experiments on simulated data sets

Ordered gene trees were generated by simulating evolutionary histories consistent with balanced species trees of 2, 4 or 8 leaves. Note that we also tested our algorithm on unbalanced species trees to ensure that it does not affect its accuracy (data not shown). Unless

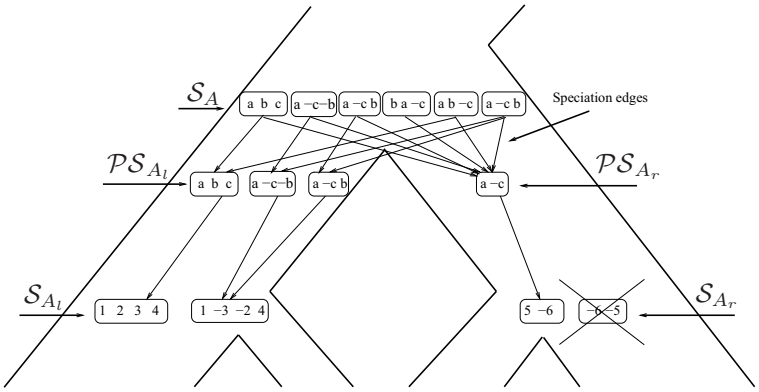

Figure 5.4 – Computation of the solution set $\mathscr{S}_A$ at the internal node A of a species tree S by Multi-DILTAG. DILTAG is executed on each of the two branches (A_l, A) and (A_r, A), leading to the two pre-speciation sets $\mathscr{PS}_{A_l}$ and $\mathscr{PS}_{A_r}$. In each branch, only minimum-cost paths are kept, which explains the removal of one gene order (indicated by a cross) from $\mathscr{S}_{A_r}$. The gene missing from the gene order of $\mathscr{PS}_{A_r}$ is reinserted in all possible positions (and all possible signs, which is not shown), and the resulting set is added to $\mathscr{PS}_{A_l}$ to form $\mathscr{S}_A$. Appropriate speciation edges are then added from the elements of $\mathscr{S}_A$ to the elements of $\mathscr{PS}_{A_l} \cup \mathscr{PS}_{A_r}$.

stated otherwise, the size of each event was sampled according to a geometric distribution of parameter $p = 0.5$, truncated by the number of genes in the ancestral cluster immediately preceding this event. The geometric distribution was chosen to represent biological data, in which smaller events are observed more frequently. We also tested $p = 0.3$ and $p = 0.8$, which give respectively more and less large events, and the results were similar (data not shown). All the results shown below are averaged over 50 replicates.

Similarly to the DILTAG algorithm, we define the penalty cost of an event e of size m (acting on a segment of m genes) as $\alpha_e + m\beta_e$, where α_e is the opening cost and β_e the extension cost of e. Our results were obtained with the same values used in [97] to test the DILTAG algorithm, namely :

- $\alpha_{t-dup} = 100$; $\beta_{t-dup} = 1$,
- $\alpha_{i-dup} = 100$; $\beta_{i-dup} = 1$,
- $\alpha_{del} = 500$; $\beta_{del} = 1$,
- $\alpha_{inv} = 500$; $\beta_{inv} = 1$.

5.5.1.1 Execution time

Our algorithm was implemented in C++ and runs on a typical Linux workstation. Figure 5.5 shows the execution time of Multi-DILTAG. The left diagram shows results for balanced species trees of 2, 4 and 8 leaves. The depth d of the extant genomes for trees with 2, 4 and 8 leaves are respectively 2, 3 and 4. We generated histories with n single, n double tandem duplications (simultaneous duplication of 2 genes) and 2 inversions on each branch of the species tree. At each step in the curves, n is incremented by 1 and thus the number of genes in each extant genome is equal to $3dn+1$. Note that this is the only experiment in which we used fixed tandem duplication sizes (1 or 2), and we did this only to get the same number of genes in every genome.

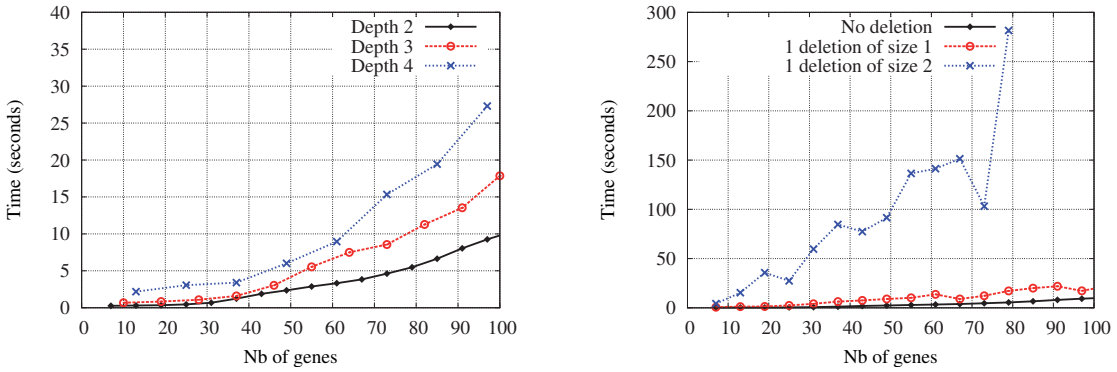

Figure 5.5 – Execution time. Left : Execution time of Multi-DILTAG on genomes containing a fixed number of genes on all the leaf genomes. Balanced species trees of maximum depth 2 (2 leaves), 3 (4 leaves) and 4 (8 leaves) were generated. Right : Execution time of Multi-DILTAG on species trees with two leaves, for simulated histories with no deletion (the same curve as the one indicated by a plain black line on the left diagram), 1 deletion of size 1 and 1 deletion of size 2.

Figure 5.5 right then shows the effect of introducing deletions. Only histories with 2 extant genomes were generated, and we plotted the running times for simulated histories containing no deletion, 1 deletion of size 1 and 1 deletion of size 2.

Clearly the execution time of Multi-DILTAG is exponential in the number of genes in

extant genomes. Nevertheless, it is possible to get results in under 30 seconds for a family of approximately 100 genes in 8 species. On the other hand, deletions of size greater than 1 slows down Multi-DILTAG dramatically. The idea of considering all possible extensions of gene orders, by reinserting lost copies in any possible way, results in an exponential number of orders in the number of copies to reinsert and the size of the orders in which we make the insertions.

5.5.1.2 Number of duplications

We now evaluate the ability of Multi-DILTAG to infer the correct total number of duplications (direct + inverted). We simulated evolutionary histories containing as many duplications as inverted duplications with 2 (Figure 5.6 left), 4 (Figure 5.6 center) and 8 (Figure 5.6 right) extant genomes, and we plotted the total number of duplications inferred for histories generated with 0 %, 33 % and 50 % of inversions.

More precisely, for each x, we generate a history with a total of x duplications together with 0, $x/2$ or x inversions, respectively leading to the curves for 0 %, 33 % and 50 % of inversions. The total number of events performed for each value of x is distributed evenly on the branches of the species tree.

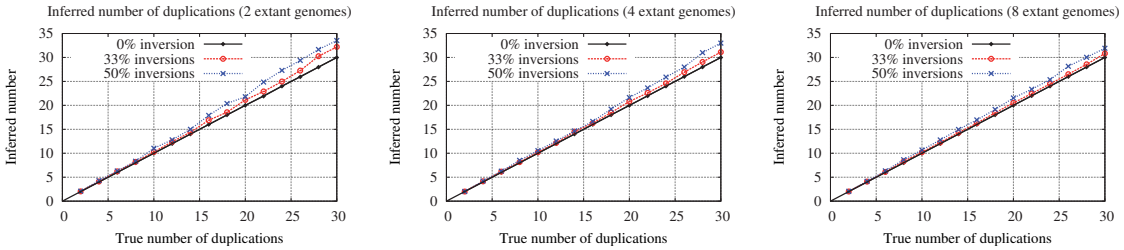

Figure 5.6 – Number of duplications. Inferred number of duplications (direct + inverted) for histories containing duplications and respectively 0 %, 33 % and 50 % of inversions. Left : Two extant genomes. Center : Four extant genomes. Right : Eight extant genomes.

As we see, Multi-DILTAG is almost perfect in inferring the total number of duplications when there are no inversions. The presence of inversions induces a small overestimation in the inferred number of duplications. As noticed in [97], this can be explained by the size limit of the DILTAG priority queue used to explore the search space and the chosen cost configuration, which may lead to choosing a history with more duplications in order to infer fewer inversions.

Notice that the overestimation is a little bit more pronounced in Figure 5.6 left. This can be easily explained by the fact that there are fewer branches in the balanced species tree containing 2 extant genomes than in the ones of 4 and 8 extant genomes. Therefore, for the same total number of duplications, more inversions are present on each branch of the smallest species tree.

5.5.1.3 Duplication size distribution

Finally, we measure the accuracy of Multi-DILTAG for inferring the duplication size distribution. Histories containing 2 (Figure 5.7 left), 4 (Figure 5.7 center) and 8 (Figure 5.7 right) extant genomes were generated. In all cases, 4 tandem duplications, 1 inverted duplication, 1 inversion and 1 deletion of size 1 or 2 were simulated on each branch of the corresponding balanced species tree.

Clearly, Multi-DILTAG is able to infer the duplication size distribution very accurately for the three data sets. We can only observe a slight overestimation of duplications of size 1 and underestimation of duplications of size 2.

We do not report the correctness of the inferred duplication events because a lot of equivalent optimal evolutionary histories are obtained by Multi-DILTAG, so it is possible that most of the inferred duplications do not correspond to the simulated duplications.

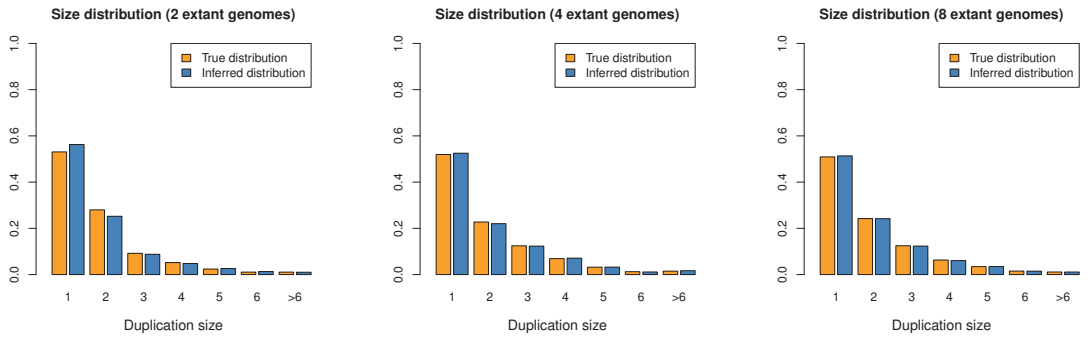

Figure 5.7 – Comparison between the true and the inferred duplication size distributions. Histories were generated with 4 tandem duplications, 1 inverted duplication, 1 inversion and 1 deletion of size 1 or 2 on each branch of the species tree. Left : Two extant genomes. Center : Four extant genomes. Right : Eight extant genomes.

5.5.2 Experiments on the protocadherin gene clusters

We applied Multi-DILTAG to the three protocadherin (Pcdh) gene clusters (α, β and γ) in human, chimpanzee, mouse and rat (for the α cluster only). It is believed that protocadherins play a role in synaptic development and neuronal survival [94, 169, 173]. Each gene in the protocadherin clusters consists of a single *variable* exon. In the α and γ clusters only, there are three additional *constant* exons at their 3' end that are alternatively cis-spliced to each variable exon. This kind of genomic organization suggests a mode of evolution through tandem duplications and deletions of the variable exons in each cluster (inversions and inverted duplications are not allowed here as they would be deleterious).

We downloaded most of the protein sequences for the three protocadherin gene clusters from the UCSC Genome Browser (http ://genome.ucsc.edu/) for human (February 2009, hg19), chimpanzee (October 2010, panTro3), mouse (July 2007, mm9) and rat (November 2004, rn4). Missing genes in the downloaded sequences for chimpanzee were downloaded manually from UniProt (http ://www.uniprot.org/). The rat β and γ clusters were discarded from our experiments because some gene sequences could not be found.

We restricted our analysis to the regions of the variable exons encoding ectodomains 2 and 3, since it has been shown that these regions are the most divergent and retain most of the phylogenetic signal [123, 178]. The human and mouse CDH12 genes were used as an outgroup. The protein sequences were aligned with ProbCons version 1.12 [41] and rooted gene trees were obtained with MrBayes version 3.1.2 [138], using the Jones-Taylor-Thornton substitution matrix [91] and $500,000$ MCMC iterations.

We then applied Multi-DILTAG to the first hundred most probable trees obtained for each Pcdh cluster, averaging our results proportionally to the posterior probability of each tree. However, recall that our algorithm computes the minimal ID-distance on each speciation edge of the solution graph. As mentioned earlier, inversions are not allowed in the case of the protocadherin gene clusters, so the inferred evolutionary histories that contain inversions are discarded from our results. The presence of these inversions might be the result of an incorrect input gene tree, or might simply show that Multi-DILTAG is unable to find the correct evolutionary history for this input tree. Note that only 14 gene trees (on a total of 300) caused inversions to appear in the inferred histories. The posterior cumulative probability (according to MrBayes) of the considered gene trees for the α, β and γ clusters are respectively 0.504, 0.690 and 0.409.

To ensure that the results do not significantly depend on the choice of the cost parameters, we used three different configurations : ($\alpha_{del} = 500$; $\beta_{del} = 1$), ($\alpha_{del} = 250$; $\beta_{del} = 250$) and ($\alpha_{del} = 1$; $\beta_{del} = 500$).

The number of events inferred by Multi-DILTAG on each branch of the species tree and the duplication size distributions for the three protocadherin gene clusters are presented in Figure 5.8.

As we could expect from the well-conserved number of genes between the studied species, almost all the events occurred on the branch above the last common ancestor of these species (Figure 5.8 left). We can also see that there is an important fraction of multiple

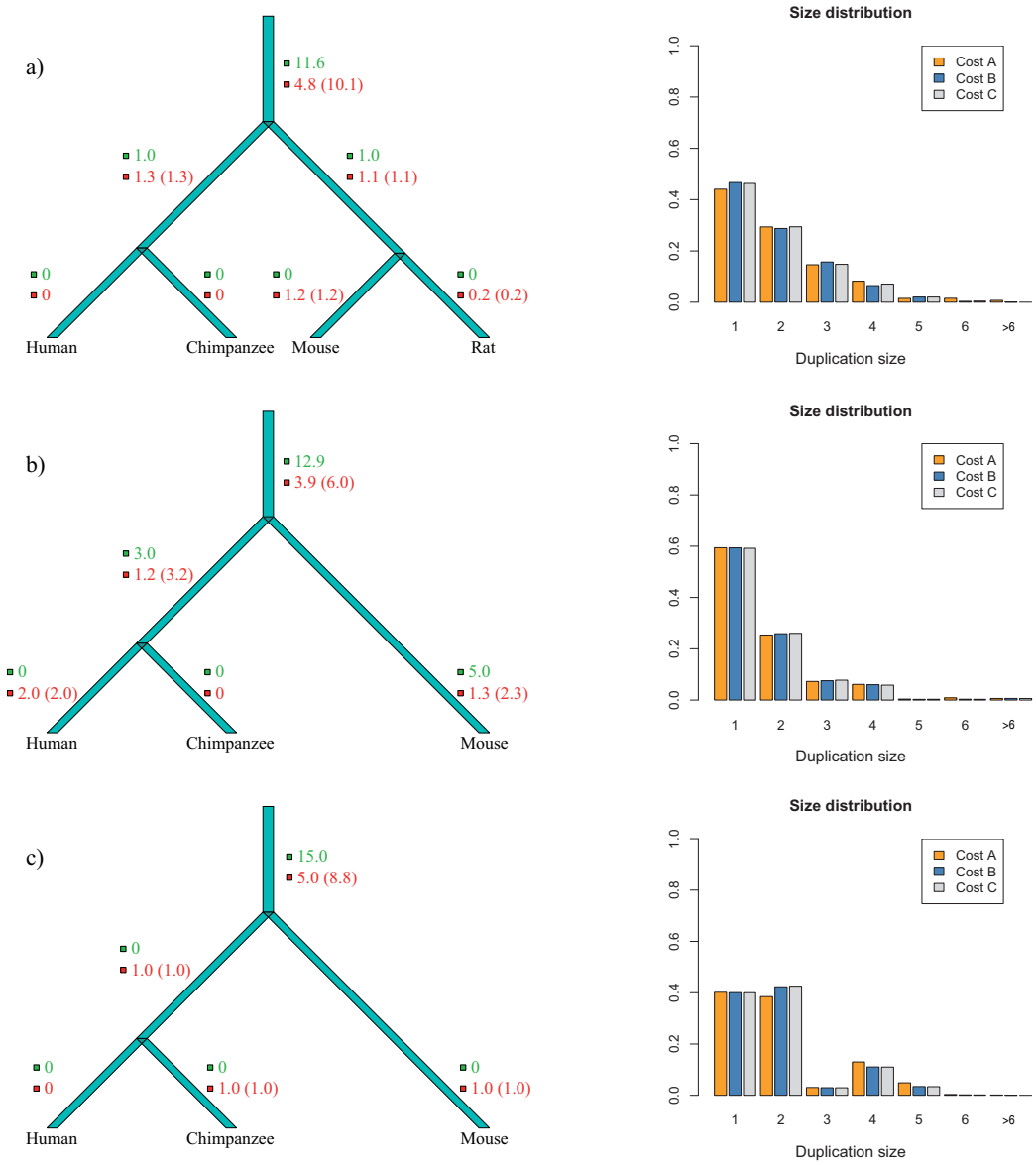

Figure 5.8 – Results for the Pcdh-α (a), Pcdh-β (b) and Pcdh-γ (c) clusters. Left : Estimated number of events on each branch of the species tree for cost configuration A ($\alpha_{del} = 500$; $\beta_{del} = 1$). The number of tandem duplications is written in green, the number of deletions is written in red and the number of gene losses is shown in parentheses. Right : Inferred duplication size distributions for the three different cost configurations considered : A ($\alpha_{del} = 500$; $\beta_{del} = 1$), B ($\alpha_{del} = 250$; $\beta_{del} = 250$) and C ($\alpha_{del} = 1$; $\beta_{del} = 500$).

gene duplications in the size distributions (Figure 5.8 right). Another interesting fact is that approximately the same number of double tandem duplications and single tandem duplications were inferred in the Pcdh-γ cluster (Figure 5.8 (c) right). This tends to confirm the hypothesis suggested in [179] that the Pcdh-γ cluster evolved by duplications involving pairs of genes.

5.6 Conclusions

We presented Multi-DILTAG, a generalization of DILTAG for the study of the evolutionary history of a set of orthologous TAG clusters in multiple species, with an evolutionary model allowing for simple or multiple tandem duplications, direct or inverted, simple or multiple deletions, and inversion events. Our results showed that our algorithm is very robust in inferring the number and size distribution of duplications. We then applied Multi-DILTAG to the protocadherin gene clusters of human, chimpanzee, mouse and rat to estimate the number of events among the different branches of the species tree and the duplication sizes. A short-term future work will concern the application of our algorithm to other sets of orthologous gene clusters.

However, a clear limitation of Multi-DILTAG is the time complexity of the approach taken to deal with deleted genes. An important future work will be to develop a fast heuristic to find an optimal set of extensions of gene orders without reinserting the lost copies in any possible way.

5.7 Competing interests

The authors declare that they have no competing interests.

CHAPITRE 6

A GRAPH-THEORETIC APPROACH FOR INPARALOG DETECTION

Olivier Tremblay Savard[1] **et Krister M. Swenson**[1]

Article publié dans le journal *BMC Bioinformatics* en 2012 [163].

6.1 Contributions

Olivier Tremblay Savard et Krister M. Swenson ont conçu les algorithmes. **Olivier Tremblay Savard** a implémenté les algorithmes. **Olivier Tremblay Savard** a conçu et réalisé les expériences sur les données simulées et biologiques. **Olivier Tremblay Savard** et Krister M. Swenson ont rédigé l'article.

[1]DIRO, Université de Montréal, Canada

6.2 Abstract

Understanding the history of a gene family that evolves through duplication, speciation, and loss is a fundamental problem in comparative genomics. Features such as function, position, and structural similarity between genes are intimately connected to this history ; relationships between genes such as orthology (genes related through a speciation event) or paralogy (genes related through a duplication event) are usually correlated with these features. For example, recent work has shown that in human and mouse there is a strong connection between function and inparalogs, the paralogs that were created since the speciation event separating the human and mouse lineages. Methods exist for detecting inparalogs that either use information from only two species, or consider a set of species but rely on clustering methods. In this paper we present a graph-theoretic approach for finding lower bounds on the number of inparalogs for a given set of species ; we pose an edge covering problem on the similarity graph and give an efficient 2/3-approximation as well as a faster heuristic. Since the physical position of inparalogs corresponding to recent speciations is not likely to have changed since the duplication, we also use our predictions to estimate the types of duplications that have occurred in some vertebrates and drosophila.

6.3 Introduction

Gene duplication and subsequent modification or loss is a fundamental biological process that is well known to create novel gene function [125]. The first step in most multi-gene studies is to infer the historical relationship of the genes in question ; *orthologous* genes are related through a speciation event in the history while *paralogous* genes are related through duplication events. Due to the accelerated rate of divergence of genes after duplication events [72, 93, 103, 107], it is generally understood that a pair of paralogous genes are likely to have diverged more than a pair of orthologous genes. Certain recent paralogs, however, may not have had time to diverge significantly. Therefore, paralogs can be further categorized into those that have been created since a particular speciation (*inparalogs*), and those that were created before the speciation (*outparalogs*)[154].

If the speciation in question is a relatively recent speciation then inparalogs represent recent duplications. Thus, they have been used to study properties of duplications under the assumption that the inparalogs have not had time to significantly diverge from the state directly following the duplication [56, 124]. Another recent study has shown that for mouse and human, sequence identity for inparalogs is the strongest predictor of gene function (*e.g.* much stronger than orthology)[121].

This motivates the study of large-scale detection of inparalogs. Tree-based inference such as reconciliation is considered to be the most accurate and comprehensive way to infer gene relationships[6, 31, 87]. However, large scale application of such methods has historically been limited due to the large amount of computation that must be done to obtain an accurate gene and species tree, from which the reconciliation can be calculated.

Thus, many studies rely on tools based on pairwise similarity measures between genes. Although a daunting number of tools have been developed for orthology detection (due to its relationship to function)[31, 58], relatively few have been conceived with inpara-

log detection specifically in mind. To our knowledge, the only methods that explicitly consider inparalog detection are InParanoid[136], MultiParanoid[4], OrthoMCL[101], and OrthoInspector[105], all of which employ the same basic methodology for inparalog detection : the best similarity between genes in different genomes is evidence for orthology, inparalogs are then inferred to be all pairs of genes within one of the genomes that are more similar than the putative ortholog pair. MultiParanoid has a clustering method built on top of the InParanoid framework to deal with multiple genomes.

In this paper we simultaneously consider the global information given by multiple genes in multiple genomes ; this extra information affords us the power to detect less similar pairs of inparalogs, and provides robustness against gene loss. In particular, our approach gives a lower bound on the number of inparalog pairs, based on finding an "orthogonal edge cover" of the colored similarity graph proposed in Zheng et al. [192]. In this graph, each vertex represents a gene and its color represents the genome it belongs to. The edges represent the similarity (*e.g.* sequence, domain, structure, regulatory, isoform, etc.) between the genes. The idea behind our method is to cover the maximum number of genes by orthology relationships. The genes that are left uncovered are considered to have arisen through duplication, and any such gene that is similar enough to a covered gene is considered to be inparalogous to that gene.

The covering step of the method corresponds to finding a so-called maximum orthogonal edge cover of the graph, a problem first posed for finding functional ortholog sets [192]. We propose two algorithms for this optimization problem : one approximation algorithm which covers at least 2/3rds of the number of vertices of a maximum orthogonal edge cover, and a heuristic that is shown to be faster and more efficient on dense graphs.

We apply our method to the genomes of human, chimpanzee, mouse, rat, zebrafish, pufferfish, *Drosophila melanogaster*, and *Drosophila simulans*. We show compelling examples of inparalogs that would not be detected by the other methods (*e.g.* InParanoid). Finally,

we show that the distribution of the physical distance between inparalog pairs that we compute is consistent with that of Ezawa et al. [56].

6.4 Inparalogs and multiple species

Given species A and B, inparalogs are pairs of genes such that one was duplicated from the other since the speciation separating A and B. The pairwise nature of this definition has led to tools like InParanoid which consider only two genomes at a time. We later motivate the use of many species when inferring inparalogy relationships. In particular, we show that by considering more than two species at once we can 1) be robust to gene loss, and 2) find low similarity inparalogs.

We generalize the definition of inparalogy to consider multiple species with a known phylogeny. For a set of species S, any duplication occurring on a branch connected to a leaf gives rise to an inparalog pair for S. A *lowest speciation* for S is a speciation on the species tree of S that has no more recent speciation with regards to the species from S.

Definition 1. *An inparalog pair for a species set S is a pair of genes* (a,b)*, such that a was duplicated from b after a lowest speciation for S.*

In the genealogy of Figure 6.1(c), there are no inparalogs since the duplication d occurred before the speciation of mouse and rat. On the other hand, the duplication d in Figure 6.1(d) does correspond to an inparalog pair since it occurred after the speciation between mouse and human/chimp. Note that Definition 1 is a generalization of the traditional definition.

6.5 Inparalogs and edge covers

InParanoid builds sets of inparalog pairs which it then must merge based on an extensive set of rules. We forgo this complicated merging process by considering the pairwise similarities in a global fashion. Further, our method is robust to gene loss due to the fact that we consider the genes from multiple genomes at once.

Consider the graph $G(V,E)$, where V has one vertex per gene and E has an edge $e = \{v,u\}$ with weight $w(e)$ corresponding to the similarity of gene v and u. The vertices are colored by the genome that they come from. We refer to G as the *similarity graph*. Figure 6.1(a) shows a component of the similarity graph for human, chimp, mouse, and rat, along with the simplest gene history consistent with this information.

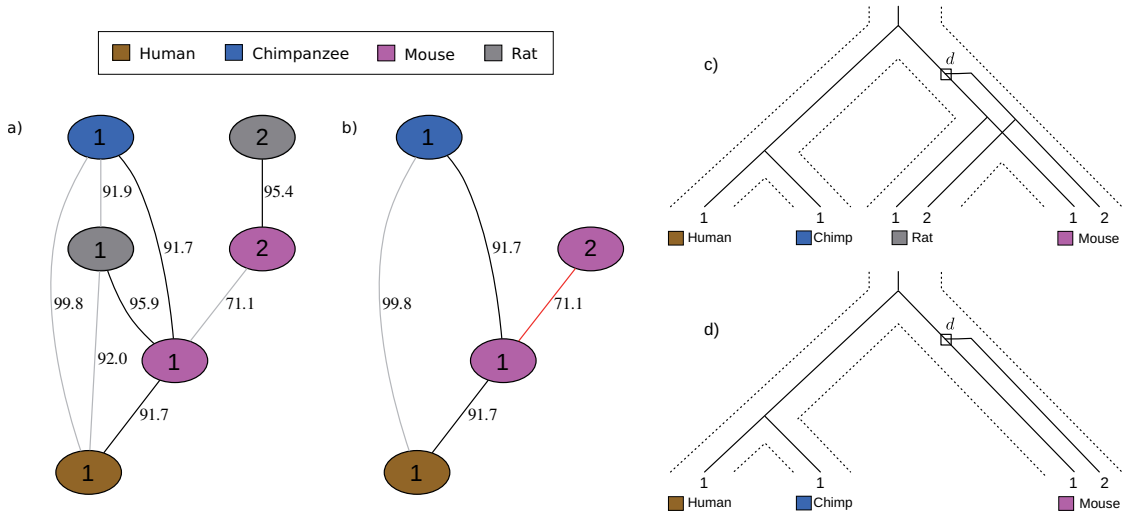

Figure 6.1 – Inparalogs and multiple species. (a) is a connected component of the similarity graph (see Section 6.7.2.1 for a description of the data) consistent with the gene history depicted in (c). With the complete information from all four species there are no inparalogs inferred, as the duplication d is above the speciation between mouse and rat. (b) is the subgraph of (a) with rat removed from consideration. One inparalog is inferred in the mouse with respect to human and chimp (as depicted in (d)).

The similarity graph holds the global information corresponding to the history of all

gene families. An important property about the genes in the similarity graph is the following. Call any maximal set of genes that originated from the root or from a duplication event an *ortholog set*. For example, Figure 6.1(c) shows two ortholog sets : the leaves labeled 2 originate from the duplication d, while the leaves labeled 1 originate at the root of the tree.

Property 1 (orthogonality[192])**.** *Any ortholog set corresponds to a subgraph of G where there exists at most one vertex with a given color.*

An *orthogonal subgraph* of G is a graph $G' = (V, E')$ where $E' \subseteq E$ and every connected component is orthogonal. We can consider the edges E' to be the evidence for orthology sets. In the similarity graphs of Figures 6.1 and 6.2 each component induced by only the black edges is orthogonal whereas the graph with all edges is not.

Our method is based on the observation that inparalogs belong to orthology sets of size one, whereas in the absence of losses all other paralogs will be orthologous to at least one other gene. Figure 6.1(d) depicts a history where there is only a single copy of gene 2 in the mouse which is inparalogous to gene 1 of the mouse, whereas Figure 6.1(c) has no single copy gene and no inparalogs.

Thus, for a given subgraph of G we consider genes that have at least one edge incident to them to be *covered*, since covered vertices represent those genes that are orthologous to at least one other gene. This motivates the following approach :

1. find an orthogonal subgraph of G such that the maximum number of vertices are covered, and then
2. mark as inparalogs all uncovered vertices with high similarity to some other gene in the same genome.

Step 1 corresponds to solving the maximum orthogonal edge cover problem. In Section 6.6 we present an $O(|V|^{(1.5)}|E|)$ 2/3-approximation algorithm for this problem, along

with a faster and simpler heuristic. Step 2 is implemented in two different ways. The first way calls an uncovered gene inparalogous to the highest weight neighbor that is covered, provided that weight is above some threshold. The second considers the possibility that a chain of duplications originated from a single gene, and is described in the next section.

6.5.1 Chains of duplications

A chain of multiple duplications, each originating from the previous duplicate copy, will result in multiple uncovered vertices of a single color, as depicted for zebrafish in Figure 6.2. For this reason, we have an indirect version of step 2 that builds a maximum spanning tree of uncovered vertices under the premise that they are all inparalogs. In the example of Figure 6.2 the maximum spanning tree happens to be a path of orange vertices.

6.5.2 Further motivation

The simplest notion of inparalogy requires only a single genome and a measure of similarity between genes : the most closely related genes would then just be the proposed inparalogs. For example, Ezawa et al. [56] used the synonymous distance and a threshold to identify recently created paralogs. InParanoid uses a little more information by including two genomes and taking genes as inparalogs if their similarity is greater than the best orthology assignment for either one. We motivate the need for a global approach with two examples from the data in Section 6.7.2.1, where we applied our algorithm to the whole genomes of two great hominoids, two rodents, two fish, and two flies.

Figure 6.2 shows an example where a pair of inparalogs for a low scoring pair is detected by our algorithm. In particular, genes NP_956275.1, NP_001037819.2, and NP_997852.1 are marked as inparalogous with respect to the speciation between zebrafish and pufferfish. The genes from human, chimp, mouse, rat, along with NP_001037819.2 from zebra-

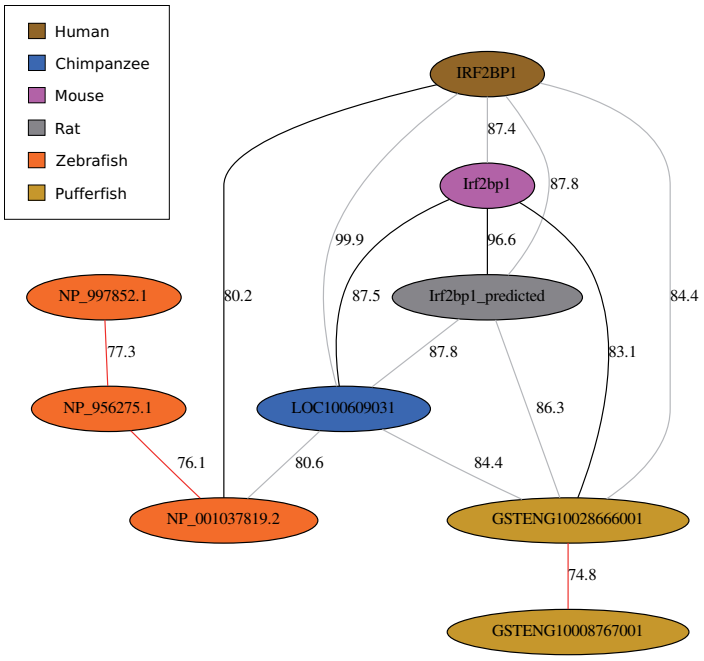

Figure 6.2 – A connected component of the similarity graph. See Section 6.7.2.1 for a description of the data. Black edges are part of our cover while red edges highlight inferred inparalog relationships. Despite the low similarity between genes NP_956275.1 and NP_001037819.2, a clear signal for their inparalogy is present. The relationship is confirmed by gene annotations. InParanoid does not detect such inparalogy for the speciation between zebrafish and pufferfish.

fish, are all annotated as “interferon regulatory factor 2-binding protein 1”. Remarkably, NP_956275.1 is annotated as “interferon regulatory factor 2-binding protein 2-A”, and the beginning of the gene NP_997852.1 is annotated as “Interferon regulatory factor 2-binding protein zinc finger”. The genes for the inparalogy marked in pufferfish have yet to be annotated in the NCBI database. Note that, at 77.0, the similarity between NP_001037819.2 from zebrafish and GSTENG10028666001 from pufferfish is below our threshold of 80 and is not depicted. Since the similarity between NP_001037819.2 and NP_956275.1 is less than this, InParanoid does not mark the two as inparalogs with respect to the zebrafish/pufferfish speciation ; the InParanoid7 database labels the two as inparalogs with

respect to *Drosophila melanogaster*.

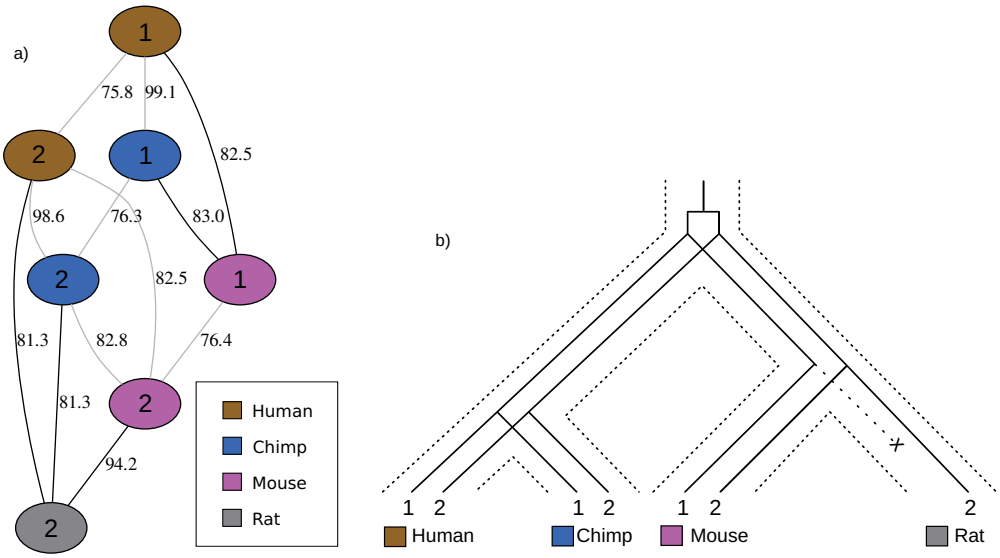

Figure 6.3 – Losses in the context of multiple genomes. (a) is a connected component of the similarity graph (see Section 6.7.2.1 for a description of the data) consistent with the gene history depicted in (b) representing protocadherin gamma b1 (1) and b2 (2) genes. The protocadherin gamma b1 gene is known to have been lost in rat [178]. It is essential that human or chimp be considered in the analysis with mouse and rat, otherwise a false inparalog relationship would be inferred between the two mouse genes.

A slightly more general, but simpler, approach than that of InParanoid would consider the similarity graph for two genomes ; in this case the graph is bipartite. Thus, a maximum matching on the weighted graph covers the maximum number of genes with the maximum amount of global similarity. The uncovered vertices are then candidates to be inparalogs ; those that are similar enough to other genes are considered to be inparalogous to those genes. While this method may not suffer from the problem of lower similarity between inparalogs (illustrated in Figure 6.2), it may suffer from that of Figure 6.3. In this case we see a simple example where consideration of only two genomes (mouse and rat) will hide the fact that there was a loss of gene 1 in rat, resulting in a false inparalog assignment of genes 1 and 2 in mouse.

6.5.3 Thresholds

Step 1 of our algorithm calls for a subset of the edges that results in a minimum number of uncovered vertices. Note that this measure does not have anything to do with the number of edges or the weight of the edges that are chosen ; the maximum orthogonal partition problem is inherently unweighted. For this reason, our method requires a threshold for interspecies similarity scores ; all edges labeled above the threshold will be considered significant. Similarly, to reduce false positives, the intraspecies similarity scores may have a different threshold.

An interspecies threshold that is too high will yield an unweighted graph with components that are very small, and we will lose the power of the multiple genome inference. An interspecies threshold that is too low may yield large components that have too many optimal solutions. While there may be some question as to what threshold is the best, we have yet to do a detailed study on this. Instead, we have chosen conservative thresholds for both measures ; all the results reported in this paper have interspecies threshold of 80 and intraspecies threshold of 70.

6.6 Maximum orthogonal edge cover

In this section we describe the algorithms for maximum orthogonal edge cover. Our 2/3-approximation algorithm runs in $O(|V|^{(1.5)}|E|)$ time. While the heuristic has the same worst-case complexity, we show in Section 6.7.1 that the running times are faster in practice.

Take a set S with a color function $c : S \mapsto \mathbb{N}$.

Definition 2. *An* orthogonal partition *of set S is a partition $\cup_i S_i = S$ such that for any distinct $s, r \in S_i$, $c(s) \neq c(r)$.*

Take a graph $G = (V, E)$ with a color function $c : V \mapsto \mathbb{N}$. Traditionally, an *edge cover* is a subset $E' \subseteq E$ such that all vertices are present in at least one edge of E'; for this we reserve the term *perfect edge cover*. For our purposes, we relax the definition of "edge cover" to be *any* subset of E.

Definition 3. *An* edge cover *of G is a subset of E.*

Consider the partition of V induced by the connected components of an edge cover.

Definition 4. *An* orthogonal edge cover *of a graph G is an edge cover where the induced partition on V (by the "is connected to" relation) is orthogonal.*

A maximum orthogonal edge cover of G is an orthogonal edge cover which covers the maximum number of vertices, over all possible orthogonal edge covers.

Let $v(E)$ denote the vertex set for some edge set E. The maximum orthogonal edge cover (MAX-OREC) problem can be stated as follows :

Input : Undirected graph $G = (V, E)$ and color function $c : V \mapsto \mathbb{N}$.

Solution : An orthogonal edge cover $E' \subseteq E$ (*i.e.* for each connected component C of $G' = (V, E')$ we have $c(s) \neq c(r)$ for all distinct $s, r \in C$).

Measure : The number of vertices covered (*i.e.* $|v(E')|$)

We present a 2/3-approximation algorithm for MAX-OREC. Our approach is to first compute edges that cover the maximum number of vertices for each color, while ignoring the orthogonality constraint. We then show that the connected components of this edge cover have a particular structure, allowing us to ensure orthogonality without removing too many edges.

6.6.1 Bipartite matchings

Consider the bipartite graph $B(x) = (U, W, F)$ where U is the subset of V with color x, $W = V \setminus U$, and F consists of the edges that span U and W (*i.e.* $U = \{v : v \in V, c(v) = x\}$ and $F = \{(u,w) : (u,w) \in E, u \in U, w \in W\}$). The following property on orthogonal edge covers holds.

Property 2. *A maximum matching $M(x)$ on $B(x)$ covers the maximum (over any edge cover) number of vertices of color x, without breaking the orthogonality constraint.*

Now take a maximum orthogonal edge cover Q^*, and an edge cover consisting of all the edges from all the bipartite matchings $R = \cup_i M(i)$. The following is a direct consequence of Property 2.

Lemma 5. $|v(Q^*)| \leq |v(R)|$.

If every connected component of $H = (V, R)$ has an orthogonal edge cover, then $|v(R)| = |v(Q^*)|$.

We show in the next section that, while we can not always find an orthogonal edge cover for every component, we can always find an edge cover that includes at least 2/3rds of the vertices in the component.

6.6.2 Covering bounded degree graphs

Consider the neighborhood of a particular vertex v with color x from the graph $H = (V, R)$ described in the previous section. Any vertex in the neighborhood of v with color y is a result of the matching $M(x)$ or $M(y)$. Thus, there are at most two vertices of color y connected to v. We call a graph with this property *2-neighborhood-limited* (2NL). We use the fact that H is 2NL to show that we can find an orthogonal edge cover that includes at least 2/3rds of the maximum number of possible vertices.

Call a path in a component *alternating* if vertices in the path alternate between two colors. The *length* of a path is the number of vertices in the path. Then we have the following two lemmas.

Lemma 6. *A connected 2NL graph G that contains no odd-length alternating path has a perfect orthogonal edge cover.*

Proof. Any even-length alternating path can be covered by taking every other edge on the path (a perfect matching). Consider the graph $G' = (V, E')$ where E' is E without the edges that are removed from perfect matchings on even-length alternating paths. Note that this graph has no alternating paths and that the degree of all vertices is at least one.

Take a minimal edge cover $C' \subseteq E'$ that covers all the vertices of V. C' is composed only of stars (every component has no simple path of length greater than two), otherwise it would not be minimal. Since no edge of C' links two vertices corresponding to the same color and there exists no alternating paths in G', the edge cover C' must be orthogonal. □

Lemma 7. *Each odd-length alternating path can contribute to at most one uncovered vertex in a maximum orthogonal edge cover.*

Proof. Every other edge on an odd-length alternating path can be matched, leaving a single vertex uncovered. □

Lemmata 6 and 7 imply the following algorithm for finding an approximate orthogonal edge cover on a 2NL graph where $maximumMatching(H')$ returns a maximum matching on H' and $minPerfectEdgeCover(H'')$ returns a minimal perfect edge cover for each component of H'' that has more than one vertex (*i.e.* a perfect edge cover where removing any edge will result in an uncovered vertex). The correctness of the algorithm follows the same reasoning as the proof of Lemma 6.

Algorithm getMAX-2NL-OREC($H = (V,R)$)
1. $P = \{$the set of edges in alternating paths of $H\}$
2. $H' = (v(P), P)$
3. $E'' = maximumMatching(H') \cup (R \setminus P)$
4. $H'' = (V, E'')$
5. **Return** $minPerfectEdgeCover(H'')$

Figure 6.4 – Algorithm getMAX-2NL-OREC($H = (V,R)$)

6.6.3 Bringing things together

Say Algorithm getMAX-2NL-OREC($H = (V,R)$) returns an edge cover Q for a 2NL graph while an optimal edge cover is Q^*. Then the following is a direct consequence of Lemmata 6 and 7.

Lemma 8. $|v(Q)| > |v(Q^*)| - p$ *where p is the number of odd-length paths in a 2NL graph.*

So counting the number of odd-length paths gives us an idea of how far we could be from the optimal. Since the shortest possible odd-length path has three vertices, and two of them can be covered, we get the desired approximation guarantee.

Lemma 9. $|v(Q)| > \frac{2}{3}|v(Q^*)|$

Now, using Section 6.6.1 along with Algorithm getMAX-2NL-OREC($H = (V,R)$), we can use Algorithm getMAX-OREC(G) to approximate the MAX-OREC problem where $M(x)$ is the maximum bipartite matching between the vertices with color x and all the other vertices. Say Algorithm getMAX-OREC(G) returns an edge cover O while an optimal edge cover is O^*. Then the following are a direct consequence of Lemmata 5,8 and 9.

Theorem 4. $|v(O)| > |v(O^*)| - p$ *where p is the number of odd-length paths in a graph.*

Theorem 5. $|v(O)| > \frac{2}{3}|v(O^*)|$

Algorithm getMAX-OREC(G)
1. $R \leftarrow \emptyset$
2. **For** each color x in G **Do**
3. $\quad R \leftarrow R \cup M(x)$
4. **End For**
5. $O \leftarrow \emptyset$
6. **For** each component C of $H = (V,R)$ **Do**
7. $\quad O \leftarrow O \cup$ getMAX-2NL-OREC(C)
8. **End For**
9. **Return** O

Figure 6.5 – Algorithm getMAX-OREC(G)

6.6.4 Running time

The running time of Algorithm getMAX-2NL-OREC($H = (V,R)$) is $O(\sqrt{|V|}|E|)$ since a minimal perfect edge cover and a maximum matching [84] can be computed in $O(\sqrt{|V|}|E|)$ time, while listing alternating paths takes linear time. The running time of Algorithm getMAX-OREC(G) is therefore $O(|C|\sqrt{|V|}|E|)$, where C is the number of colors. In the worst case this bound is $O(|V|^{(1.5)}|E|)$ since there are at most $O(|V|)$ colors.

6.6.5 A fast heuristic

We also developed a practical algorithm for MAX-OREC. It is simpler to implement and runs faster in practice and performs better on dense graphs (see Section 6.7.1). The algorithm does the following :

1. compute H, the union over all maximum bipartite matchings for every pair of colors, and then
2. compute $minPerfectEdgeCover(H)$.

Note that the main difference with the approximation algorithm is that we do not compute the same maximum bipartite matchings.

6.7 Results and discussion

6.7.1 Experiments on simulated datasets

We implemented the 2/3-approximation algorithm and the heuristic in C++ and we applied them to simulated datasets in order to compare their performance. We generated random graphs using the $G(n,p)$ model introduced by Gilbert [69]. A random graph $G(n,p)$ has n nodes and for each $n(n-1)/2$ possible pairs of nodes, an edge is created with probability p. The expected number of edges in a $G(n,p)$ graph is $\binom{n}{2}p$. The probability p corresponds to the expected percentage of completeness of the random graph.

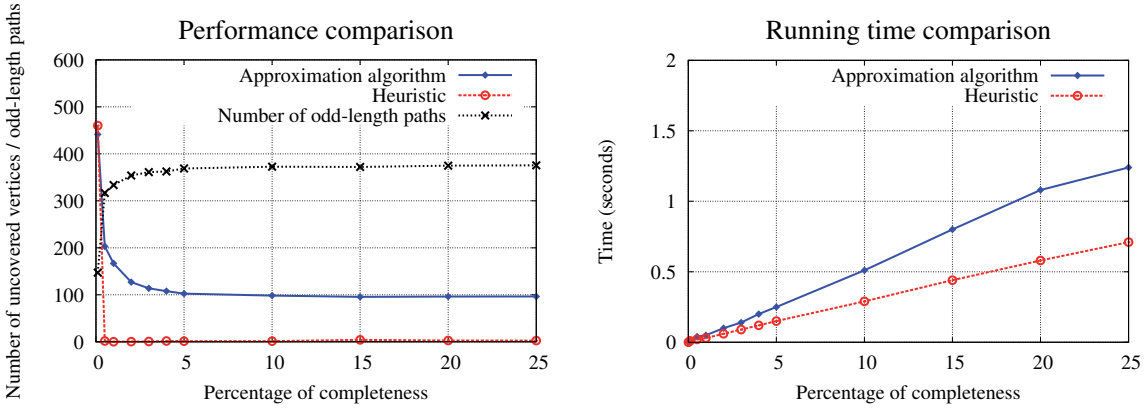

Figure 6.6 – Comparison of the performance of the 2/3-approximation algorithm and the heuristic. The results are averaged over 100 random graphs of 2500 vertices (genes) and 5 colors (genomes). Left : Comparison of the number of uncovered vertices. The number of odd-length alternating paths is also shown. Right : Running time comparison.

Figure 6.6-left shows the number of uncovered vertices (averaged over 100 replicates) given by both algorithms for random graphs of 2500 nodes and a varying expected percentage of completeness. Clearly, the heuristic performs a lot better than the approximation

algorithm on dense graphs. It is also faster than the approximation algorithm (Figure 6.6-right). However, the approximation algorithm covers more vertices than the heuristic on really sparse graphs (under 0.1% of completeness). We also show the average number of odd-length alternating paths that is found by the approximation algorithm. Since the number of uncovered vertices given by the approximation algorithm is always significantly lower than the number of odd-length alternating paths (except for very sparse graphs), it is clear that the approximation algorithm covers more vertices than the worst expected case (*i.e.* 2/3 of the optimal maximum number of covered vertices).

6.7.2 Experiments on real datasets

In this section, we present an analysis of the inparalog pairs inferred by our approach on the genomes of human, chimpanzee, mouse, rat, zebrafish, pufferfish, *Drosophila melanogaster*, and *Drosophila simulans*. We first describe how we obtained the data and then we show an example of how we can use recent inparalogs to study modes of duplication.

6.7.2.1 Creating the input graph

We used CoGe :SynMap [109] interface to Last (Blast variant) to make all-versus-all pairwise comparisons between the studied species and the self comparisons. SynMap is usually used to find syntenic regions containing a minimum number of genes (block size), but in the context of this experiment, we simply used it with a block size of one to identify all the homologous genes. We discarded similarity edges when one gene in the pair was more than 1.25 times longer than the other.

6.7.2.2 Modes of duplication and recent inparalogs

The most studied duplication mechanisms are whole genome duplication, tandem duplication and retrotransposition. Whole genome duplication has the effect of simultaneously doubling all the chromosomes of a genome. It has been shown that whole genome duplication has occurred at least once [81, 115] and maybe twice [82, 106, 155] in the vertebrate ancestor. Then, a fish-specific round of genome duplication was reported by studies conducted on teleost fish [90, 120, 177] and an additional round was shown to have occurred in the salmonid fish lineage [5]. As the name implies, tandem duplication creates adjacent duplicate gene copies. It is believed that unequal crossing-over during meiosis is the principal mechanism responsible for the creation of tandem duplicates [61]. The third well-studied duplication mechanisms is retrotransposition, which usually produces intronless gene copies that can end up anywhere in the genome. In mammals, LINE-1 retrotranposons are mainly responsible for creating those duplicates [126].

Another mode of duplication that has been receiving more attention in the recent years is the one responsible for the creation of segmental duplications. It has been named duplicative transposition in [47] and drift duplication in [56]. This kind of duplication can create in one step duplicate gene copies (with introns, as opposed to retrotransposition) that are transposed anywhere in the genome, even on different chromosomes. The biological mechanisms behind duplicative transposition are not yet fully understood, but it is believed that Alu repeats could be involved in primates [11].

In order to better understand duplication mechanisms and study the relative rates of the different types of duplications, it is interesting to study recently created gene duplicates. For example, a study on recently emerged paralogs in human, mouse, zebrafish, *Drosophila melanogaster*, *Drosophila pseudoobscura* and *Caenorhabditis elegans* suggested that drift duplication occurs nearly as often as tandem duplication in vertebrates [56].

6.7.2.3 Analysis of the inparalog pairs

We identified inparalog pairs in the studied genomes and retrieved information on their physical distance and percent similarity. Figure 6.7 presents, for each species, the distribution of the inparalog pairs for three classes of physical distance and four classes of percent similarity. The three classes of physical distance represent inparalog pairs separated by less than 50 kbp, inparalog pairs separated by more than 50 kbp and unlinked inparalog pairs, *i.e.* inparalog pairs that are located on different chromosomes. The 50 kbp boundary was chosen as an attempt to separate possible tandem duplication events (physical distance of less than 50 kbp) from the other more far-reaching duplication mechanisms (like retrotransposition and duplicative transposition). We used three boundaries to divide the inparalog pairs into four percent similarity classes, in order to verify if there is a correlation between the age of the duplication events and the physical distance.

Only mouse and *D. melanogaster* have inparalog pairs with a physical distance of less than 50 kbp, which suggests that tandem duplication could occur more frequently in those species. Inparalog pairs having a physical distance of more than 50 kbp are rare in chimpanzee, rat, zebrafish and pufferfish, but make for an important fraction of the inparalog pairs of human, mouse and the two drosophilas. The most interesting case is in human, where more than 20% of the $>$50 kbp inparalog pairs are recent ($>$95% similarity). This could suggest that a significant number of linked duplicative transpositions or retrotranspositions occurred relatively recently in human, which is consistent with the findings of Ezawa et al. [56].

For all the species, a large fraction of the inparalog pairs are unlinked. This is especially true for zebrafish and pufferfish, where more than 80% of the inparalog pairs are located on different chromosomes. Interestingly, the majority of the unlinked pairs in the fish species have a low percent similarity. We hypothesize that this could be the result of

ongoing fractionation after the fish-specific whole genome duplication. Human, chimpanzee and rat all have at least 10% of recent unlinked inparalog pairs (>95% similarity). This could be evidence of recent duplicative transpositions or retrotransposition. Older unlinked inparalog pairs (<95% similarity) do not necessarily correspond to older duplicative transposition events. For example, a scenario involving tandem duplication followed by genomic rearrangement events could have produced the same results.

6.8 Conclusion

We presented a new graph-theoretic approach for the detection of inparalogs. Our method uses a maximum orthogonal edge cover on the similarity graph and then identifies inparalogs in the set of uncovered vertices. We developed a 2/3-approximation algorithm for this problem and a heuristic that was shown to be faster and more efficient on dense graphs. Note that our method is not suitable for finding orthologous gene relationships since our edge covers aggressively leave the minimum number of genes unmatched. Zheng et al. [192] discuss other objective functions on the similarity graph that are more suitable for orthology detection.

We have shown compelling examples of why using the information for multiple species gives more accurate inparalog predictions and how our method allows us to infer inparalogs that would not have been found by other methods like InParanoid. We then presented an example of how we can use recent inparalogs to study modes of duplication. Our analysis of the genomes of human, chimpanzee, mouse, rat, zebrafish, pufferfish, *D. melanogaster* and *D. simulans* suggested that many recent tandem duplications occurred in mouse and that a significant number of linked duplicative transpositions or retrotranspositions occurred relatively recently in human.

We did not show speed comparisons with other existing methods like InParanoid be-

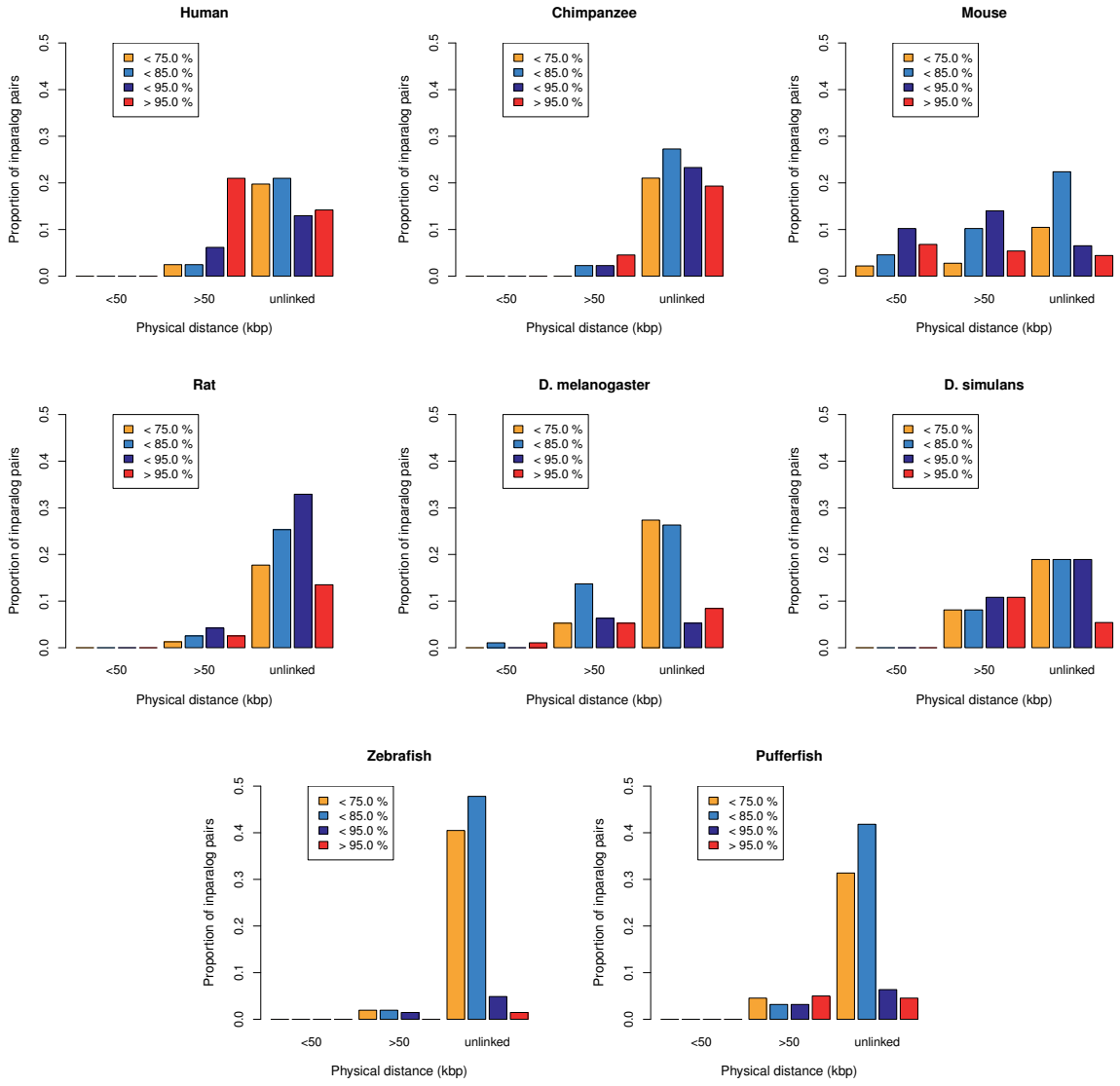

Figure 6.7 – Proportions of inparalog pairs inferred in the 8 species studied.

cause our method was very fast on real data. The results on the real datasets were obtained in 10 seconds on a typical Linux workstation.

On the methodological side, algorithmic improvements that consider edge weights while finding an edge cover are possible, as well as improved preprocessing of the data. The question remains as to which other measures of similarity our method is most powerful with.

On the evaluation side, we attempted to make large-scale comparisons against inparalogy given by reconciliation (Ensembl gene trees), but we were not able to convert in an automated manner a statistically significant number of gene names from SynMap to Ensembl IDs in order to do so. While computing statistics — like the number of inparalog pairs shared with a method like InParanoid — are possible, direct comparison as to which method finds the correct inparalog relationships remains difficult since few independent methods or bench experiments exist for finding such relationships.

6.9 Competing interests

The authors declare that they have no competing interests.

CHAPITRE 7

EVOLUTION OF TRNA GENES IN BACILLUS

Olivier Tremblay Savard[1] **et Nadia El-Mabrouk**[1]

Ceci est un manuscrit non publié qui présente les résultats de l'étude de l'évolution des gènes d'ARNt de 50 espèces de *Bacillus*. La méthode algorithmique employée pour obtenir ces résultats est présentée dans la section 3.7.2.2.

7.1 Contributions

Olivier Tremblay Savard a recueilli les données sur le contenu et l'ordre des gènes d'ARNt. **Olivier Tremblay Savard** a inféré les histoires évolutives et a fait les corrections manuelles nécessaires. **Olivier Tremblay Savard** a fait les analyses sur les histoires évolutives. **Olivier Tremblay Savard** et Nadia El-Mabrouk ont rédigé le manuscrit.

[1]DIRO, Université de Montréal, Canada

7.2 Introduction

Transfer RNAs are among the most important and ancient genes. They bridge the gap between genetic information and protein expression, and are the major constituents of the cellular translation machinery. Accordingly, a tremendous number of studies have addressed tRNA sequence identification and structure prediction. Yet, little is known about the evolution of tRNA genes in terms of number, organization and functional specificity, on a genome-wide scale. A first comprehensive analysis of the genomic organization of tRNAs in Eukarya has been conducted in 2010 [19], revealing an extensive variability of organization among lineages, which is in striking contrast to the extreme levels of sequence-conservation of tRNA genes. Other genome-wide studies conducted on specific lineages [137, 176] also revealed a rapid evolution of tRNA gene families through duplication and loss. Having a clear picture of tRNA repertoire evolution is expected to shed light on many important questions related to protein expression, such as the link between tRNA copy number and protein synthesis, the evolution of the genetic code, and tRNA functional shift. A number of studies have indeed suggested that tRNA families are not static, as various evolutionary events may have the effect of substituting a tRNA with another alloacceptor (recognizing a different amino-acid).

In this paper, the bacterium *Bacillus* is used as a model organism to study tRNA evolution. The problem we address can be formulated as follows. We are given a set of genomes annotated for tRNAs, and a species tree for the corresponding taxa. We want to infer tRNA gene content and order information of ancestral genomes identified with each of the internal nodes of the tree, together with an evolutionary scenario leading to the observed genome organization. This problem is known in the comparative genomics literature as the *small phylogenetic problem*, which has been widely studied for various restrictions on genome structure and models of evolution, most of them being difficult and developed

heuristics being time-consuming. Focusing on a cherry of the species tree, the problem reduces to the one of comparing two genomes, namely the *two species small phylogenetic problem*, which has also been extensively studied[50, 60], and has been proven difficult (NP-hard) for most problem variants. But of much more concern to biologists than details about optimality and efficiency, is the non-uniqueness of solutions. When compared genomes have sufficiently diverged so that corresponding gene orders are almost unrelated, an exponential number of evolutionary scenarios verifying the given optimality criteria can be inferred, leading to a useless and non-informative huge number of equally likely ancestral predictions. Focusing on a large set of Bacillus strains that have diverged on a short time-scale is an attempt to eliminate as far as possible sources of non-uniqueness in the construction, and at the same time makes the problem tractable. As only few events separate two neighboring genomes, these events can be assumed to be non-overlapping (each gene is involved in at most one event) and thus still "visible" in extant species.

Recently, considering an evolutionary model restricted to duplications and losses, we reformulated the comparison of two genomes as an alignment problem : find an alignment minimizing a given cost. Interestingly, such an alignment can directly be translated into an evolutionary scenario of "visible" events, and leads to a unique ancestral genome.

We applied this framework to the 50 completely sequenced *Bacillus* strains with the phylogeny taken from the Pathosystems Ressource Integration Center (PATRIC) [70]. *Bacillus* species are Gram-positive, rod-shaped bacteria. They can be aerobic or facultative anaerobic and they produce endospores that are normally resistant to heat, radiation and desinfectants. *Bacillus anthracis* (anthrax) and *Bacillus cereus* (food poisoning) are pathogens for humans, while *Bacillus thuringiensis* is a pathogen for insects and can be used as biological pest control. Some *Bacillus* species are used industrially to produce enzymes and antibiotics. *Bacillus amyloliquefaciens*, for example, is used to produce the well-known BamH1 restriction enzyme.

The history obtained after performing a pipeline of validations and manual curation reveals an evolution of the tRNA repertoire largely affected by duplications and losses, with losses being predominant. Among observed rearrangements, inversions are predominant over transpositions. Moreover, almost all large inversions have occurred around the terminus of replication, which is in agreement with previous studies on rearrangements in bacteria [162]. Our study also revealed two tRNA substitutions that we analysed in details.

7.3 Biological results

7.3.1 Data

We analysed the evolutionary history of 50 fully sequenced bacteria in the *Bacillus* genus (including 8 *Geobacillus* strains). The tRNA gene content and order are taken from GenBank [14]. The phylogeny of the studied strains shown in Figure 7.1 is taken from the Pathosystems Ressource Integration Center (PATRIC) [70].

In order to align the genomes correctly, we needed to know the locations of the origin and the terminus of replication for each strain. We used T-A and G-C skews from Comparative Genometrics [139] and Oriloc [63] to find those locations.

7.3.2 Execution of the algorithm

For the inference of the evolutionary history on the phylogeny, we proceed step by step, applying our algorithm on one cherry (two siblings) at a time from the bottom to the top of the tree. We then validate the events of an optimal scenario, and possibly correct it, by considering the neighboring strains in the phylogeny (siblings of the considered cherry). Here is how we deal with each type of event.

1. Substitutions, inversions and transpositions are symmetrical operations that can be

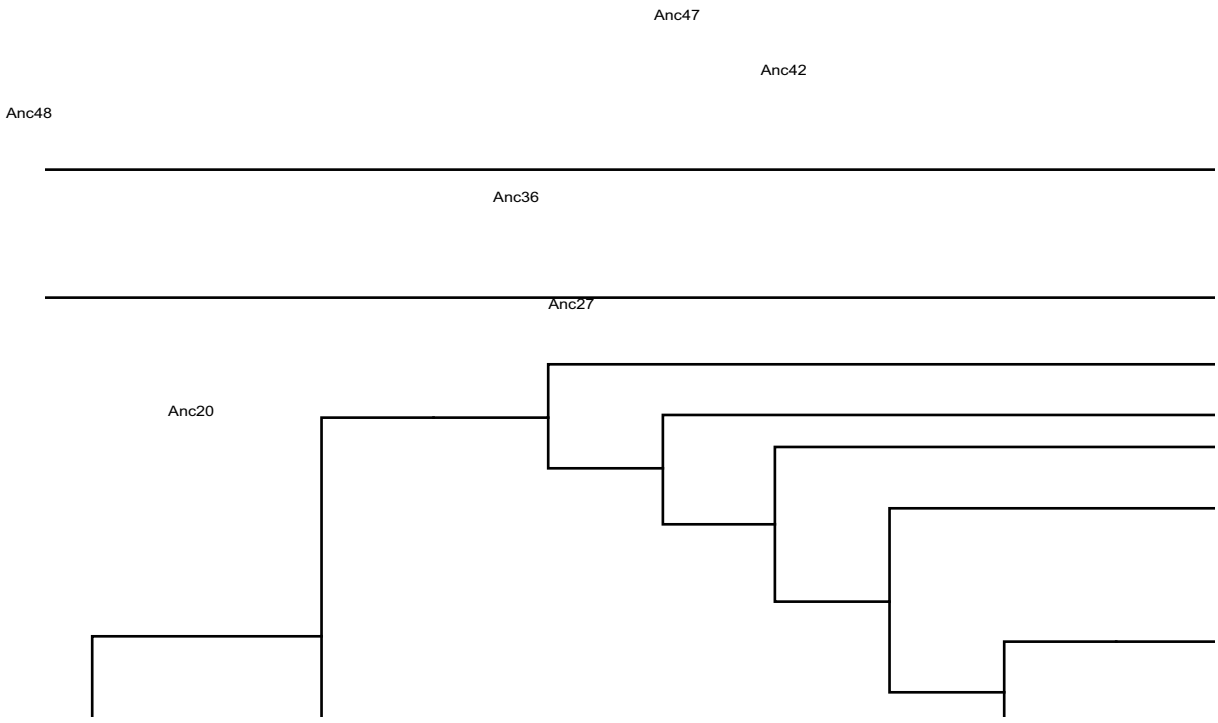

Figure 7.1 – Phylogenetic tree of the 50 studied *Bacillus* strains.

indiscriminately applied to one or the other of the two sequences in a pairwise alignment, leading to two equally parsimonious scenarios from two possible ancestral genomes. A good way to discriminate between the two scenarios is to check which gene (in case of substitution) or gene order (in case of inversion or transposition) is present at the same location in the neighboring strains of the phylogeny.

2. Duplications and deletions are interchangeable in an evolutionary scenario. As shown in Figure 7.2, it can be more parsimonious to infer a duplication instead of a deletion, or the opposite. We validate the choice made by the algorithm by checking the neighboring strains : if the considered segment is found duplicated in those strains,

then we infer a deletion, otherwise we infer a duplication.

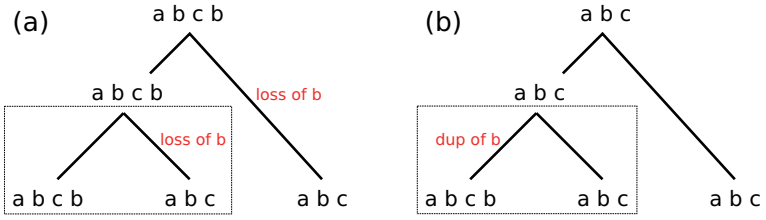

Figure 7.2 – Two possible evolutionary histories for a phylogeny containing three genomes (leaves). For the first comparison between the two genomes at the bottom of the tree (inside the dashed box), there are two possible scenarios with the same cost : in scenario (a), a loss of gene b is inferred while in scenario (b), a duplication of gene b is inferred. However, when the whole phylogeny is considered, scenario (b) is clearly the most parsimonious one (only one event is necessary to explain the phylogeny, instead of two in scenario (a)).

The algorithm is run with a default cost of one for all events. In other words, the edit distance, which is the minimum number of operations required to transform one genome into the other, is considered. Operations can be of any size, except losses and substitutions that are of size one. As substitutions are generally due to sequence divergence, considering substitutions independent among sites is biologically sound. As for losses, restriction to single gene losses is a methodological requirement to avoid biases towards very long deletions (if a loss of any size costs one, then any alignment costs at most 2 : simply delete the entire first genome and then delete the entire second genome). To cope with losses of size greater than one, we perform a post-processing by simply grouping all consecutive gene losses into a single event.

7.3.3 Genome representation

For the sake of presentation, we subdivide the genome into blocks and use a color code for blocks. We stand on the tRNA operon subdivision available for the *B. cereus* ATCC 14579 [29] to define most of these blocks : each tRNA operon was simply considered

to be a block. Blocks that are identical or very similar in terms of tRNA gene content and order are assigned the same color. Details are given in Figure 7.3 (a). Each circular genome is linearized so that both endpoints represent the origin of replication. A vertical line shows the location of the terminus of replication. A consensus representation of blocks is given in 7.3 (c). On the opposite, variable regions are drawn in white. The block length is proportional to the number of tRNA genes inside it. Figure 7.3 (b) illustrates where the tRNA gene blocks are located on the chromosome of *Bacillus cereus* ATCC 14579. In all the studied genomes, the majority of tRNA genes are found near the origin of replication and variable regions tend to be located far from it.

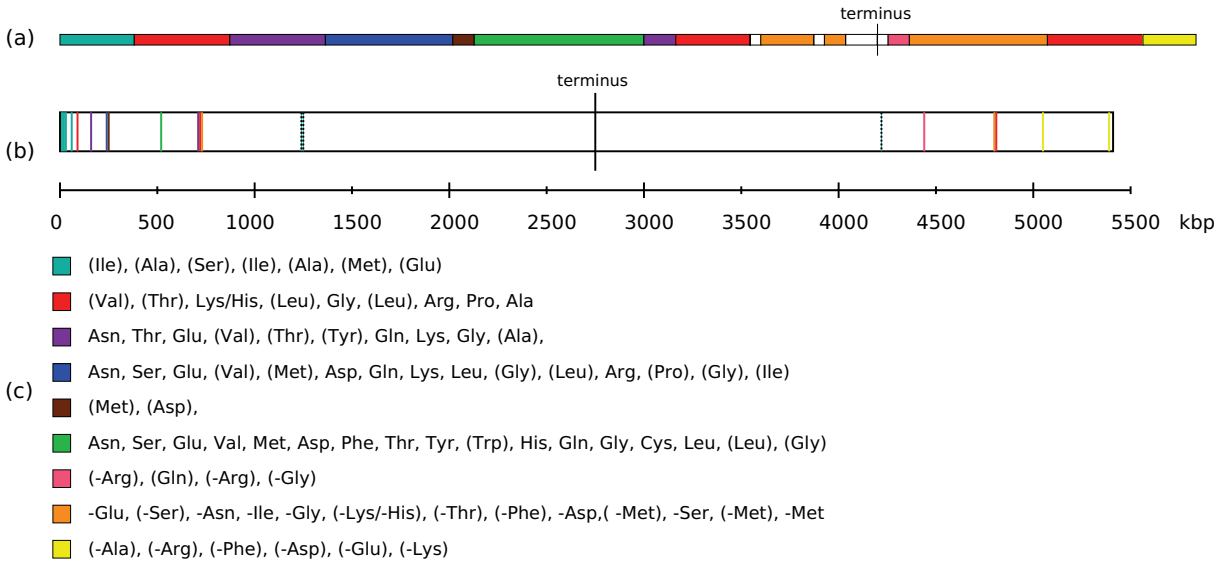

Figure 7.3 – Genome representation. (a) Representation of tRNA syntenic blocks on the *Bacillus cereus* ATCC 14579 genome. (b) Locations of the tRNA blocks on the whole genome of *Bacillus cereus* ATCC 14579. Dashed lines represent the white variable blocks. (c) Ordered sequences of tRNA isoacceptor families, representing the signature of coloured blocks. A slash (/) between two tRNA genes indicates that one or the other can be found at that position. The tRNA genes inside parentheses are absent from the block in some strains.

7.3.4 The evolutionary history at a glance

In total, 95 duplications, 141 deletions, 23 inversions, 12 transpositions and 2 substitutions were inferred, showing that duplications and losses are prevalent over rearrangement events. Moreover, deletions are more frequent than duplications. This can be explained by the fact that after a duplication, there is an excess of the tRNA genes that were copied and the loss of some of those genes is not deleterious.

As we can see in Figure 7.4, for both duplications and deletions, shorter events are much more frequent than longer ones. However, in the deletions bar graph, there seems to be an abnormal peak at size 9. This is due to the numerous deletions of the red block (which contains 9 tRNA genes) that were observed in many genomes (see the next paragraph).

Almost one third of the inversions were very large events affecting 30 tRNA genes or more. The longest inversion had a size of 65. All the inversion events except those of size 1 had (seemingly) the terminus of replication as a pivot. Note that, because of the circular nature of the bacterial chromosomes, the same inversions (*i.e.* using same breakpoints) made around the origin of replication would give the same chromosomes if we read them in the opposite direction. For the sake of simplicity, we will assume that those inversions occurred around the terminus of replication.

As for the transposition events, not enough were inferred to be able to see a clear pattern in the size distribution. Only one was relatively large (15 tRNA genes). Note that predicted transpositions could in fact be the result of a series of inversion events.

7.3.5 Major events

In this section we summarize the inferred evolutionary events on the 50 studied *Bacillus* strains. Figure 7.5 is a condensed representation of the phylogeny, where we omitted the lineages in which very few events were inferred. The largest inferred evolutionary

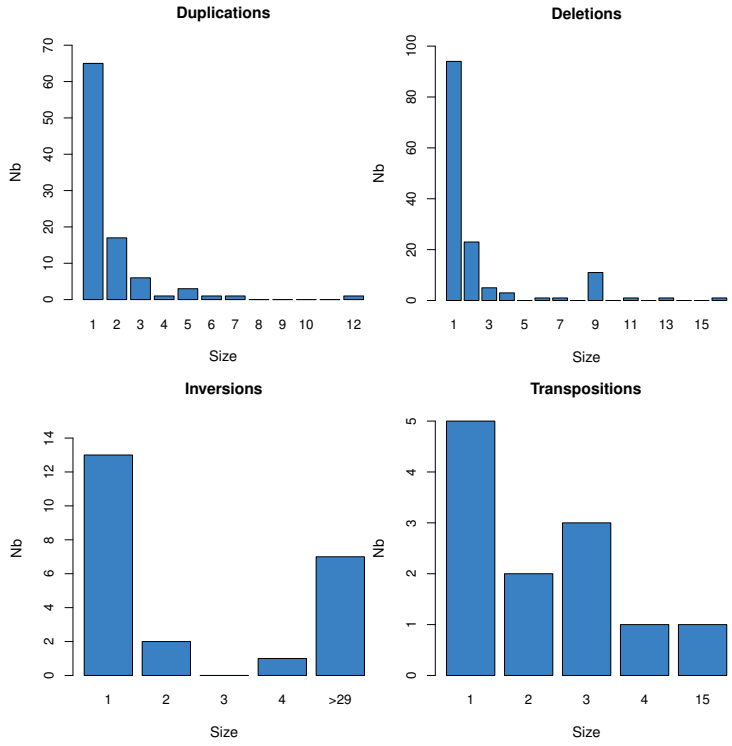

Figure 7.4 – Size distributions of all the events inferred on the whole phylogeny.

events are framed by a gray box. All framed duplications and transpositions were further validated by using whole genome alignment dotplots, and verifying that non-coding DNA between the tRNA genes were also highly similar between the source and the target of the event.

Ancestor 20 : We inferred a large inversion around the terminus of replication on the branch leading to *B. anthracis* str. CDC 684. We also observed a duplication of the green block in *B. thuringiensis* serovar chinensis CT-43. In the whole subtree of the *B. anthracis*, *B. cereus* and *B. thuringiensis* species (not shown entirely in Figure 7.5), we counted 6 deletions of one of the red blocks.

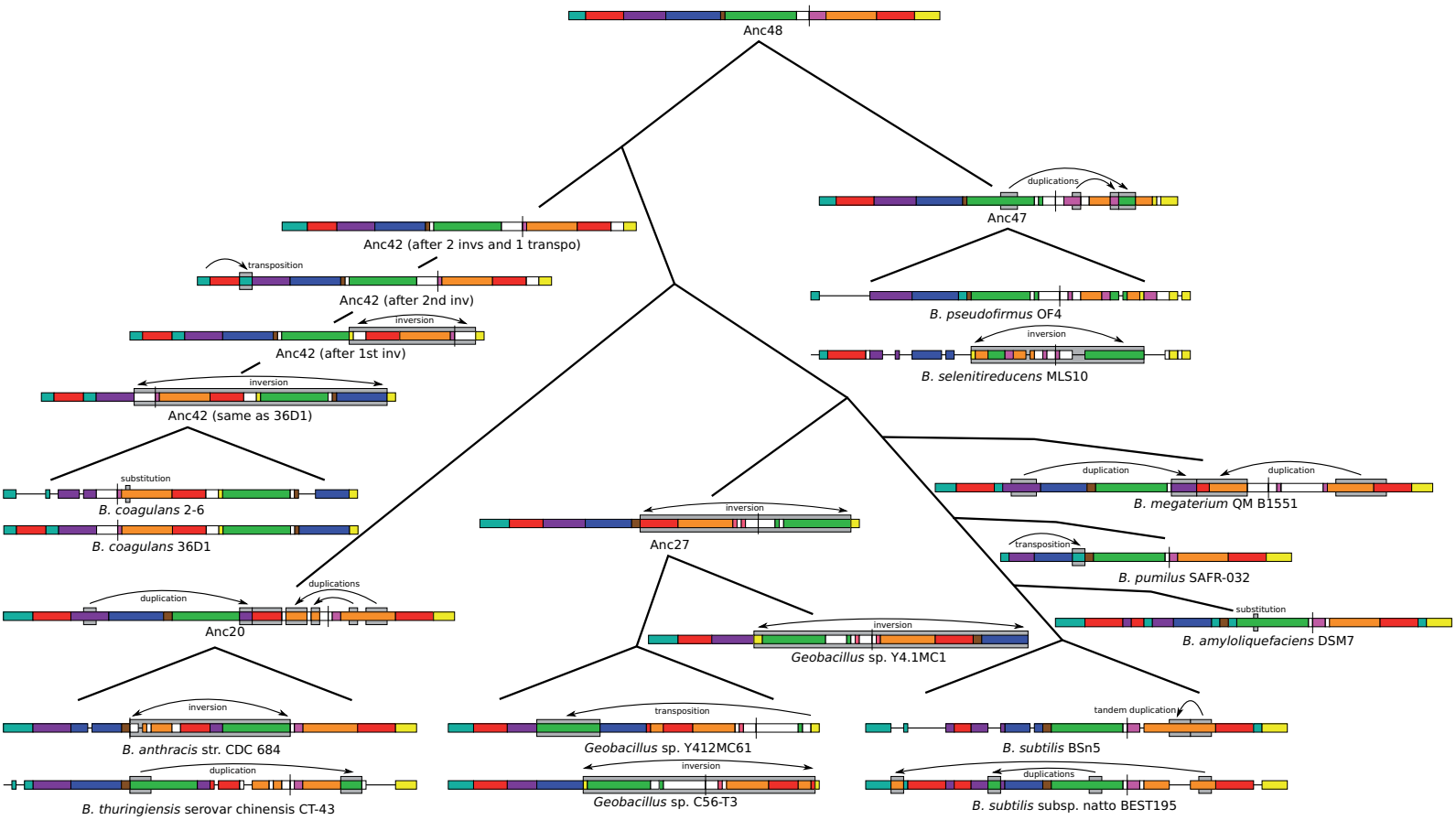

Figure 7.5 – Evolutionary history of the *Bacillus* genus. Only major events have been reported. An alignment is given for the two genomes representing each cherry of the tree.

Ancestor 27 : Three big events occurred in the *Geobacillus* subtree. First, we can see a transposition of the green block in *Geobacillus* sp. Y412MC61. Second, there has been a large inversion around the terminus on the branch leading to *Geobacillus* sp. C56-T3. Third, another large inversion around the terminus occurred on the branch leading to *Geobacillus* sp. Y4.1MC1.

Ancestor 36 : In *B. subtilis* BSn5, the orange block has been elongated by a tandem duplication of five genes. Two duplications of size 3 occurred on the branch leading to *B. subtilis* subsp. natto BEST195, copying part of the orange block inside the turquoise block and part of the green block inside the purple block. Moreover, we observed many duplicated [Ile,Ala] blocks (turquoise) in the *B. amyloliquefaciens* subtree. They have all been inserted inside existing rRNA blocks ([16S,23S,5S]), between the 16S and the 23S genes.

Ancestor 42 : For the *B. coagulans* strains, one transposition and two successive large inversions around the terminus of replication were necessary to explain the observed syntenies.

Ancestor 47 : A large inversion around the terminus of replication and many tRNA gene deletions were inferred on the branch leading to *B. selenitireducens* MLS10. Moreover, this is the only subtree in which we find tRNA-Sec genes. More precisely, *B. selenitireducens* MLS10, *B. cellulosilyticus* DSM 2522 (not shown in Figure 7.5) and *B. pseudofirmus* OF4 are the only strains of all the ones studied here containing at least one tRNA-Sec gene (note that the tRNA-Sec gene in *B. pseudofirmus* OF4 was not annotated in Genbank but found by a BLAST search). We predicted that the tRNA-Sec gene was present in the last common ancestor of the 50 studied strains.

Ancestor 48 : We can see a big duplicated region in Anc20 between the green block and the terminus of replication. This region was created by 4 separate duplication events : a duplication of part of the purple block (3 tRNA genes), a duplication of the end of the

red block (7 tRNA genes) and two duplications of parts of the orange block (5 and 2 tRNA genes). A large inversion around the terminus of replication occurred on the branch leading to Anc27. The last interesting events are two duplications that copied part of the pink block and part of the green block inside the orange block of Anc47. Finally, the last common ancestor of the 50 *Bacillus* species studied (Anc48) is shown at the top of the tree.

7.3.6 tRNA substitutions

We found one gene substitution in the *B. amyloliquefaciens* DSM7 strain when comparing it with *B. amyloliquefaciens* LL3. This is in fact a tRNA-Met gene that was affected by a few point mutations (only 4, as shown in Figure 7.6), one of them effectively changing the anticodon from CAT to CAC, which is why it was annotated as a tRNA-Val in the DSM7 strain.

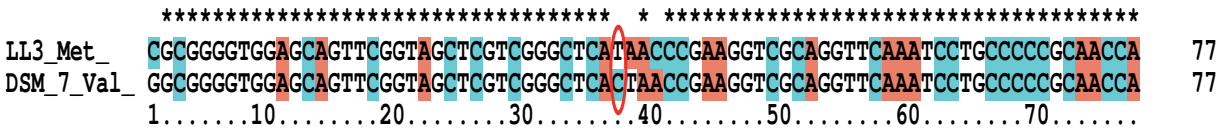

Figure 7.6 – Sequence alignment of tRNA-Met in *B. amyloliquefaciens* LL3 and tRNA-Val in *B. amyloliquefaciens* DSM7. The third position of the anticodon is circled in red.

However, it often takes more than a few mutations in a tRNA for it to be recognized and charged by a new tRNA synthetase. In order to predict if this new tRNA-Val in DSM7 is really recognized by valyl-tRNA synthetase and charged with valine, we used TFAM [8] on its nucleotide sequence and also on the one of the tRNA-Met in LL3 to validate their annotation. Interestingly, both tRNA genes are classified as initiator methionine tRNAs.

In light of this prediction by TFAM, we first hypothesized that the tRNA-Val gene in

the DSM7 strain could in fact be a tRNA-iMet with a CAC anticodon recognizing GTG start codons. Unfortunately, the resequencing of the region containing this new tRNA-Val gene in DSM7 showed that the 4 mutations presented in the alignment of Figure 7.6 were in fact the result of sequencing errors in the GenBank sequence of DSM7.

A second tRNA substitution was detected in *B. coagulans* 2-6 when comparing it with *B. coagulans* 36D1. A tRNA-Ser gene seems to have mutated to a tRNA-Thr gene according to the Genbank annotations. In fact, only 3 point mutations have occurred and one of them changed the anticodon from GCT to GGT, which explains the new annotation in the 2-6 strain (see Figure 7.7).

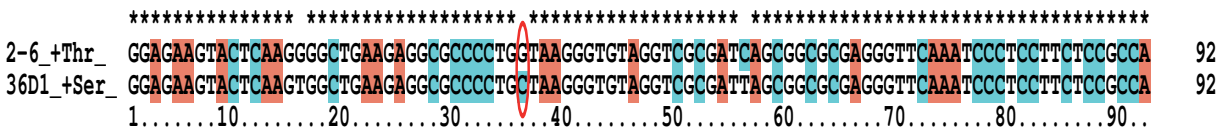

Figure 7.7 – Sequence alignment of tRNA-Thr in *B. coagulans* 2-6 and tRNA-Ser in *B. coagulans* 36D1. The second position of the anticodon is circled in red.

Once again, we used TFAM to infer the identity class of both tRNA genes. Even with the anticodon change, TFAM predicts that the tRNA gene of interest in strain 2-6 is still a tRNA-Ser (instead of a tRNA-Thr). Is this tRNA still charged with serine or is it charged with threonine ? Further validation with wet lab experiments will be necessary to answer this question. Serine and threonine have similar side chains and properties, so the insertion of a serine instead of a threonine in a protein might not be a problem. It is also possible that we are dealing with another sequencing error.

7.3.7 Operons

It has been suggested in [29] that some tRNA genes downstream of rRNA operons (containing 16S, 23S and 5S genes) could be transcribed independently because of a promotor located between the 23S and the 5S genes. Based on the results of this study, we have identified those tRNA operons downstream of the rRNA operons in the strains we are studying (see Figure 7.8 for the locations if the tRNA operons in the genome of *B. cereus* ATCC 14579).

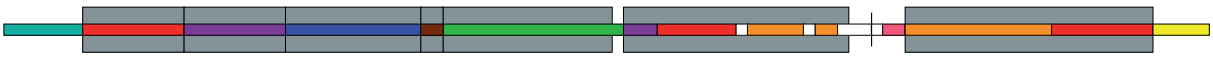

Figure 7.8 – Location of the tRNA operons in the *B. cereus* ATCC 14579 genome. The operons are framed in gray boxes.

We analysed the biggest events that we predicted in our evolutionary history to check if they were disturbing the tRNA operons. All the inversions and transpositions that were inferred did not break the operons, which is not very surprising because the operons are relatively small compared to the genome lengths. It is also probably detrimental to have an inversion inside an operon putting part of the genes on the opposite strand. Another constraint for rearrangements is that the majority of genes in bacteria are found on the leading strand (the strand that is pointing away from the origin of replication). Thus, the selective pressure is likely to be strongly against the transfer of genes from the leading strand to the lagging strand. Inversion events occurring around one of the replication axes are more common since they keep the genes on the leading strand.

Deletions of tRNA genes cannot break operons. However, it is interesting to know what happens with duplicated tRNA segments. Most duplications that we inferred are inserting the copied tRNA genes inside an existing tRNA operon (the tandem duplication

of five genes inside the orange block in *B. subtilis* BSn5 for example). We also observed duplications of tRNA-Ile and tRNA-Ala genes inside rRNA operons (between the 16S and the 23S genes). Both *B. megaterium* strains have a recent large duplicated block of tRNA genes right after the green block. There is a rRNA operon upstream of this block which suggests that it could be a new tRNA operon that was created. The same kind of mechanism occurred on the branch leading to Anc20 and gave rise to a new operon.

CHAPITRE 8

CONCLUSION

Dans cette thèse, plusieurs nouvelles méthodes algorithmiques permettant d'étudier différents aspects de l'évolution des génomes ont été présentées. En particulier, nous avons développé des algorithmes pour l'inférence de génomes ancestraux pré-dupliqués, le calcul de distances entre génomes doublés, l'inférence d'histoires évolutives de groupes de GRT et la détection de gènes in-paralogues.

Nos algorithmes pour l'inférence de génomes pré-dupliqués (*genome halving*) et le calcul de la double distance, tous deux applicables à des génomes doublés ayant perdu des copies de gènes, sont beaucoup plus appropriés pour analyser des données biologiques réelles. Notre algorithme pour le *genome halving* avec pertes, une généralisation de l'algorithme d'El-Mabrouk et Sankoff [51], conserve la complexité linéaire de son prédécesseur. L'algorithme pour le calcul de la double distance, quant à lui, a été prouvé très rapide et précis sur des données simulées. L'heuristique proposée pour l'inférence d'histoires évolutives de groupes de GRT, Multi-DILTAG, permet d'étudier l'évolution de plusieurs groupes de GRT orthologues chez plusieurs espèces, en plus de considérer tous les évènements évolutifs de DILTAG (créé précédemment dans notre laboratoire) : duplications en tandem simples ou multiples (inversées ou non), délétions et inversions. Les analyses menées sur des données simulées ont démontré que Multi-DILTAG infère le nombre et la distribution de la taille des duplications de manière très précise. Nous avons aussi développé une nouvelle méthode pour l'inférence de relations d'in-paralogies entre les gènes. Celle-ci est basée sur l'analyse d'un graphe de similarité coloré, contrairement aux autres méthodes existantes qui utilisent plutôt des approches de regroupement. Nous avons démontré que notre approche, qui permet d'analyser simultanément l'information globale

provenant de différentes espèces, permet de détecter des paires de gènes in-paralogues qui peuvent être ignorées par les autres méthodes tout en étant robuste contre les erreurs pouvant être causées par les pertes de gènes.

Ces méthodes ont également été utilisées afin d'analyser des données biologiques. Nous avons étudié l'évolution des trois groupes de GRT (α, β et γ) des protocadhérines chez l'humain, le chimpanzé, la souris et le rat avec Multi-DILTAG. Nos résultats ont confirmé l'hypothèse selon laquelle les gènes du groupe des protocadhérines γ ont surtout évolué par duplications de paires de gènes. Notre nouvelle méthode pour la détection de gènes in-paralogues, employée sur les génomes complets de l'humain, du chimpanzé, de la souris, du rat, du poisson-zèbre, du tétraodon (poisson-globe), de *Drosophila melanogaster* et de *Drosophila simulans*, nous a permis d'analyser les fréquences des différents types de duplications. Plus précisément, nos résultats suggèrent que plusieurs duplications en tandem chez la souris et un nombre important de transpositions duplicatives ou rétropositions chez l'humain ont eu lieu relativement récemment. Enfin, notre analyse de l'histoire évolutive des gènes d'ARNt de 50 souches de bactéries du genre *Bacillus* a confirmé que ces gènes ont principalement évolué par duplications et pertes de gènes. Nous avons observé plusieurs grandes inversions autour des axes de réplication du génome et quelques transpositions. Deux substitutions de gènes d'ARNt, s'étant produites par quelques mutations ponctuelles, ont été identifiées. La première semblait avoir transformé un ARNt-méthionine initiateur en un autre reconnaissant spécifiquement les codons GTG (au lieu de ATG), mais a été plus tard invalidée après un re-séquençage du gène. La deuxième substitution a transformé un ARNt-sérine en ARNt-thréonine.

Plusieurs pistes de travaux futurs sont envisageables pour poursuivre le travail qui a été accompli dans cette thèse. Une question intéressante dans le cas de l'analyse des génomes ayant subi des DGEs est de trouver des preuves que ces évènements ont eu lieu. Les méthodes que nous avons proposées dans le chapitre 4 supposent qu'on a déjà prouvé que

les DGEs ont eu lieu dans l'histoire des espèces étudiées. Par exemple, notre algorithme pour le *genome halving* avec pertes tente de reconstruire le génome tel qu'il était avant une duplication complète en minimisant la distance de ce génome avec le génome actuel. Il serait intéressant de voir si on pourrait utiliser cet algorithme pour valider une hypothèse de DGE. Il pourrait exister un certain seuil de distance au-dessus duquel l'hypothèse de la DGE serait peu probable pour expliquer l'état du génome actuel. L'algorithme pourrait également tenter de reconstruire un ancêtre pour lequel seulement un ou quelques-uns des chromosomes ont été doublés et évaluer si cette hypothèse semble meilleure que celle de la DGE en comparant les distances équivalentes aux deux hypothèses. Pour ce qui est de la double distance, notre heuristique permet de comparer un génome réarrangé dupliqué avec pertes avec un génome parfaitement dupliqué. Comme mentionné dans la conclusion du chapitre 4 (section 4.9), le plus gros problème encore ouvert est de trouver une façon de calculer efficacement la distance entre deux génomes réarrangés dupliqués avec pertes.

Dans le cas de Multi-DILTAG, on peut penser à plusieurs extensions qui pourraient améliorer cette méthode. Il serait possible d'employer une approche de maximisation d'espérance pour avoir des coûts plus réalistes pour les évènements inférés. L'idée serait de relancer l'inférence plusieurs fois de suite en ajustant les coûts à chaque itération de façon à refléter la fréquence des évènements qui ont été inférés jusqu'à temps que les résultats convergent. On pourrait également diviser les gènes en deux ou plusieurs parties et considérer au départ une phylogénie de ces différents domaines au lieu de prendre toujours les gènes en entiers. Ceci pourrait nous permettre d'étudier l'évolution de groupes de GRT qui ont plusieurs domaines ou exons qui peuvent évoluer de manière indépendante. Cette façon de faire pourrait également nous permettre d'identifier des évènements de conversion génique. Ensuite, étant donné que les arbres de gènes, qui sont nécessaires à la méthode, peuvent contenir des erreurs et qu'un arbre erroné peut engendrer une histoire évolutive moins parcimonieuse, il serait intéressant que la méthode puisse proposer

des modifications à l'arbre de gènes de manière à diminuer le nombre d'évènement évolutifs. Enfin, en ajoutant au modèle évolutif des évènements de duplication qui agissent à plus grande échelle, comme par exemple la transposition duplicative, on pourrait étudier simultanément l'histoire évolutive de tous les groupes de GRT d'une même famille qui se retrouvent à différents endroits sur un chromosome ou sur différents chromosomes.

Pour ce qui est de notre méthode pour l'inférence de gènes in-paralogues, notre approche passe par la résolution du problème de la couverture d'arêtes orthogonale maximum (*maximum orthogonal edge cover*) sur le graphe de similarité coloré. Nous ne savons pas encore si ce problème est polynomial ou NP-difficile. Il serait intéressant de voir si on peut prouver que ce problème est NP-difficile. Sinon, nous pourrions améliorer les résultats de notre méthode en développant un algorithme exact polynomial pour résoudre ce problème.

Finalement, il pourrait être intéressant de pousser un peu plus loin les résultats obtenus dans notre analyse de l'évolution des gènes d'ARNt chez *Bacillus*. Dans un article portant sur l'évolution de ces gènes chez différentes souches de *Escherichia coli*, on a estimé que le taux de duplications et délétions était d'environ 1 par million d'années [176], tandis que dans une autre étude portant sur les gènes d'ARNt de 12 espèces de drosophiles, on a évalué que ce taux était de 2.18 par million d'années [137]. Il serait important d'évaluer ce taux dans le cas des 50 souches de *Bacillus* que nous avons étudiées afin de le comparer avec ceux des autres études. Il serait aussi intéressant de proposer une phylogénie corrigée des 50 souches étudiées, c'est-à-dire une phylogénie qui minimiserait le nombre total d'évènements évolutifs.

BIBLIOGRAPHIE

[1] O. Akerborg, B. Sennblad, L. Arvestad et J. Lagergren. Simultaneous bayesian gene tree reconstruction and reconciliation analysis. *Proceedings of the National Academy of Sciences USA*, 106:5714–5719, 2009.

[2] M.A. Alekseyev et P.A. Pevzner. Colored de bruijn graphs and the genome halving problem. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 4(1):98 – 107, 2007.

[3] M.A. Alekseyev et P.A. Pevzner. Whole genome duplications, multi-break rearrangements, and genome halving problem. Dans *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 665 – 679, 2007.

[4] Andrey Alexeyenko, Ivica Tamas, Gang Liu et Erik L.L. Sonnhammer. Automatic clustering of orthologs and inparalogs shared by multiple proteomes. *Bioinformatics*, 22(14):e9–e15, 2006.

[5] F.-W. Allendorf et G.-H. Thorgaard. Tetraploidy and the evolution of salmonid fishes. Dans B.-J. Turner, éditeur, *Evolutionary genetics of fishes*, pages 1–46. Plenum Press, New York, 1984.

[6] Adrian M. Altenhoff et Christophe Dessimoz. Phylogenetic and functional assessment of orthologs inference projects and methods. *PLoS Comput Biol*, 5(1): e1000262, 01 2009.

[7] S. F. Altschul, W. Gish, W. Miller, E. W. Myers et D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403 – 410, 1990.

[8] D.H. Ardell et S.G.E. Andersson. TFAM detects co-evolution of trna identity rules

with lateral transfer of histidyl-trna synthetase. *Nucleic Acids Research*, 34:893–904, 2006.

[9] L. Arvestad, A.-C. Berglund, J. Lagergren et B. Sennblad. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. Dans *RECOMB '04 Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*, pages 326–335, 2004.

[10] D.A. Bader, B.M.E Moret et M. Yan. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *Journal of Computational Biology*, 8:483 – 491, 2001.

[11] J.-A. Bailey, G. Liu et E.-E. Eichler. An alu transposition model for the origin and expansion of human segmental duplications. *The American Journal of Human Genetics*, 73:823–834, 2003.

[12] J.-A. Bailey, A.-M. Yavor, H.-F. Massa, B.-J. Trask et E.-E. Eichler. Segmental duplications : Organization and impact within the current human genome project assembly. *Genome Research*, 11:1005–1017, 2001.

[13] M.-S. Barker, N.-C. Kane et M. Matvienko et al. Multiple paleopolyploidizations during the evolution of the compositae reveal parallel patterns of duplicate gene retention after millions of years. *Molecular Biology and Evolution*, 25:2445–2455, 2008.

[14] Dennis A. Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J. Lipman, James Ostell et Eric W. Sayers. Genbank. *Nucleic Acids Research*, 41 (D1):D36–D42, 2013.

[15] B. Benzaid, R. Dondi et N. El-Mabrouk. Duplication-loss genome alignment : Complexity and algorithm. Dans *Language and Automata Theory and Applications*, volume Lecture Notes in Computer Science 7810, pages 116–127. Springer Berlin Heidelberg, 2013.

[16] A. Bergeron, J. Mixtacki et J. Stoye. Reversal distance without hurdles and fortresses. Dans *Combinatorial Pattern Matching, LNCS*, volume 3109, pages 388 – 399, 2004.

[17] A. Bergeron, J. Mixtacki et J. Stoye. A unifying view of genome rearrangements. Dans *Algorithms in Bioinformatics, LNCS*, volume 4175 de *WABI*, pages 163 – 173, 2006.

[18] A. Bergeron, J. Mixtacki et J. Stoye. A new linear time algorithm to compute the genomic distance via the double cut and join distance. *Theoretical Computer Science*, 410(51):5300 – 5316, 2009.

[19] C. Bermudez-Santana, C. S. Attolini, T. Kirsten, J. Engelhardt, S.J. Prohaska, S. Steigele et P. Stadler. Genomic organization of eukaryotic tRNAs. *BMC Genomics*, 11(270), 2010.

[20] D. Bertrand et O. Gascuel. Topological rearrangements and local search method for tandem duplication trees. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2:15–28, 2005.

[21] D. Bertrand, M. Lajoie et N. El-Mabrouk. Inferring ancestral gene orders for a family of tandemly arrayed genes. *Journal of Computational Biology*, 15(8):1063-1077, 2008.

[22] J.-A. Birchler, U. Bhadra, M.-P. Bahdra et D.-L. Auger. Dosage-dependent gene regulation in multicellular eukaryotes : implications for dosage compensation, aneuploid syndromes, and quantitative traits. *Developmental Biology*, 234:275 – 288, 2001.

[23] J.-A. Birchler et K.-J. Newton. Modulation of protein levels in chromosomal dosage series in maize : the biochemical basis of aneuploid syndromes. *Genetics*, 99:247 – 266, 1981.

[24] G. Blanc, K. Hokamp et K.H. Wolfe. A recent polyploidy superimposed on older large-scale duplications in the Arabidopsis genome. *Genome Research*, 13:137 – 144, 2003.

[25] T. Blomme, K. Vandepoele, S. De Bodt, C. Silmillion, S. Maere et Y. van de Peer. The gain and loss of genes during 600 millions years of vertebrate evolution. *Genome Biology*, 7:R43, 2006.

[26] P. Bonizzoni, G. Della Vedova et R. Dondi. Reconciling a gene tree to a species tree under the duplication cost model. *Theoretical Computer Science*, 347:36–53, 2005.

[27] J.E. Bowers, B.A. Chapman, J. Romg et A.H. Paterson. Unravelling angiospermn genome evolution by phylogenetic analysis of chromosomal duplication events. *Nature*, 422:433 – 438, 2003.

[28] D. Brown et K. Sjolander. Functional classification using phylogenomic inference. *PLoS Computational Biology*, 2:e77, 2006.

[29] Benjamin Candelon, Kévin Guilloux, S. Dusko Ehrlich et Alexei Sorokin. Two distinct types of rrna operons in the bacillus cereus group. *Microbiology*, 150(3): 601–611, 2004.

[30] C. Chauve et N. El-Mabrouk. New perspectives on gene family evolution : losses in reconciliation and a link with supertrees. Dans S. Batzoglou, éditeur, *Research in Molecular Biology (RECOMB 2009)*, volume 5541 de *Lecture Notes in Computer Science*, pages 46–58. Springer, 2009.

[31] F. Chen, A. J. Mackey, J. K. Vermunt et D. S. Roos. Assessing performance of orthology detection strategies applied to eukaryotic genomes. *PLoS ONE*, 2(4): e383, 04 2007.

[32] J.-M. Chen, D. N. Cooper, N. Chuzhanova, C. Férec et G. P. Patrinos. Gene conversion : mechanisms, evolution and human disease. *Nature Reviews Genetics*, 8:762–775, 2007.

[33] K. Chen, D. Durand et M. Farach-Coton. Notung : A program for dating gene duplications and optimizing gene family trees. *Journal of Computational Biology*, 7:429–447, 2000.

[34] International Human Genome Mapping Consortium. A physical map of the human genome. *Nature*, 409:934–941, 2001.

[35] J.-A. Cotton et R.-D.-M. Page. Rates and patterns of gene duplication and loss in the human genome. *Proceedings of the Royal Society of London. Series B*, 272: 277–283, 2005.

[36] C. J. Creevey et J. O. McInerney. An algorithm for detecting directional and non-directional positive selection, neutrality and negative selection in protein coding dna sequences. *Gene*, 300:43–51, 2002.

[37] L. Cui, P.-K. Wall et J.-H. Leebens-Mack. Widespread genome duplications throughout the history of flowering plants. *Genome Research*, 16:738–749, 2006.

[38] M. J. Curcio et K. M. Derbyshire. The outs and ins of transposition : From mu to kangaroo. *Nature Reviews Molecular Cell Biology*, 4:865–877, 2003.

[39] C. Darwin. *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. John Murray, London, 1859.

[40] C. Darwin et A. Wallace. On the tendency of species to form varieties ; and on the perpetuation of varieties and species by natural means of selection. *Proceedings of the Linnean Society of London*, 3(9):45–62, 1858.

[41] C.-B. Do, M.-S.-P. Mahabhashyam, M. Brudno et S. Batzoglou. Probcons : Probabilistic consistency-based multiple sequence alignment. *Genome Research*, 15: 330–340, 2005.

[42] T. Dobzhansky et A. Sturtevant. Inversions in the chromosomes of drosophila pseudoobscura. *Genetics*, 23:28–64, 1938.

[43] J.-P. Doyon, V. Ranwez, V. Daubin et V. Berry. Models, algorithms and programs for phylogeny reconciliation. *Briefings in Bioinformatics*, 12(5):392–400, 2011.

[44] D. Durand, B.-V. Haldórsson et B. Vernot. A hybrid micro-macroevolutionary approach to gene tree reconstruction. *Journal of Computational Biology*, 13:320–335, 2006.

[45] B. Dutrillaux. Chromosomal evolution in primate. tentative phylogeny from microcebus murinus (prosimian to man). *Human Genetics*, 48:251–314, 1979.

[46] L. Edelmann, E. Spiteri, K. Koren, V. Pulijaal, M. G. Bialer, A. Shanske, R. Goldberg et B. E. Morrow. At-rich palindromes mediate the constitutional t(11 ;22) translocation. *The American Journal of Human Genetics*, 68:1–13, 2001.

[47] E.-E. Eichler. Recent duplication, domain accretion and the dynamic mutation of the human genome. *TRENDS in Genetics*, 17:661–669, 2001.

[48] E.-E. Eichler et D. Sankoff. Structural dynamics of eukaryotic chromosome evolution. *Science*, 301:793–797, 2003.

[49] J. A. Eisen, J. F. Heidelberg, O. White et S. L. Salzberg. Evidence for symmetric chromosomal inversions around the replication origin in bacteria. *Genome Biology*, 1(6):research 0011.1–0011.9, 2000.

[50] N. El-Mabrouk. *Mathematics of Evolution and Phylogeny*, chaptitre Genome rearrangement with gene families, pages 291- 320. Oxford University Press, 2005.

[51] N. El-Mabrouk et D. Sankoff. The reconstruction of doubled genomes. *SIAM Journal on Computing*, 32(1):754 – 792, 2003.

[52] Nadia El-Mabrouk. Genome rearrangement by reversals and insertions/deletions of contiguous segments. Dans Raffaele Giancarlo et David Sankoff, éditeurs, *Proceedings of the Eleventh Annual Symposium on Combinatorial Pattern Matching (CPM 2000)*, volume 1848 de *Lecture Notes in Computer Science*, pages 222–234, 2000.

[53] O. Elemento, O. Gascuel et M-P. Lefranc. Reconstructing the duplication history of tandemly repeated genes. *Molecular Biology and Evolution*, 19:278–288, 2002.

[54] J. Robinson et al. Imgt/hla and imgt/mhc : sequence databases for the study of the major histocompatibility complex. *Nucleic Acids Research*, 31:311–314, 2003.

[55] O. Eulenstein, B. Mirkin et M. Vingron. Comparison of annotating duplication, tree mapping, and copying as methods to compare gene trees with species trees.

Mathematical hierarchies and biology ; DIMACS Series Discrete Math. Theoret. Comput. Sci., 37:71-93, 1997.

[56] K. Ezawa, K. Ikeo, T. Gojobori et N. Saitou. Evolutionary patterns of recently emerged animal duplogs. *Genome Biology and Evolution*, 3:1119–1135, 2011.

[57] K. Ezawa, S. Oota et N. Saitou. Genome-wide search of gene conversions in duplicated genes of mouse and rat. *Molecular Biology and Evolution*, 23:927–940, 2006.

[58] Gang Fang, Nitin Bhardwaj, Rebecca Robilotto et Mark B. Gerstein. Getting started in gene orthology and functional analysis. *PLoS Comput Biol*, 6(3):e1000703, 03 2010.

[59] J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Sunderland, 2003.

[60] G. Fertin, A. Labarre, I. Rusu, E. Tannier et S. Vialette. *Combinatorics of genome rearrangements*. The MIT Press, Cambridge, Massachusetts and London, England, 2009.

[61] W.-M. Fitch. Phylogenies constrained by cross-over process as illustrated by human hemoglobins and a thirteen-cycle, eleven amino-acid repeat in human apolipoprotein A-I. *Genetics*, 86:623–644, 1977.

[62] A. Force, M. Lynch, F.-B. Pickett, A. Amores, Y. Yan et J. Postlethwait. Preservation of duplicate genes by complementary, degenerative mutations. *Genetics*, 151: 1531–1545, 1999.

[63] A. C. Frank et J. R. Lobry. Oriloc : prediction of replication boundaries in unannotated bacterial chromosomes. *Bioinformatics*, 16(6):560–561, 2000.

[64] T. Gabaldon. Large-scale assignment of orthology : back to phylogenetics ? *Genome Biology*, 9:235, 2008.

[65] M. Gabrisko et S. Janecek. Characterization of maltase clusters in the genus drosophila. *Journal of Molecular Evolution*, 72:104- 118, 2011.

[66] Y. Gagnon, O. Tremblay Savard, D. Bertrand et N. El-Mabrouk. Adances on genome duplication distances. Dans *Comparative Genomics*, volume 6398 de *Lecture Notes in Computer Science*, pages 25–38. Springer Berlin Heidelberg, 2011.

[67] O. Gascuel, D. Bertrand et O. Elemento. Reconstructing the duplication history of tandemly repeated sequences. Dans O. Gascuel, éditeur, *Mathematics of Evolution and Phylogeny*, pages 205–235. Oxford, 2005.

[68] H. Gavranović et E. Tannier. Guided genome halving : probably optimal solutions provide good insights into the preduplication ancestral genome of Saccharomyces cerevisiae. Dans *Pacific Symposium on Biocomputing*, volume 15, pages 21 – 30, 2010.

[69] E.-N. Gilbert. Random graphs. *Annals of Mathematical Statistics*, 30:1141–1144, 1959.

[70] Joseph J. Gillespie, Alice R. Wattam, Stephen A. Cammer, Joseph L. Gabbard, Maulik P. Shukla, Oral Dalay, Timothy Driscoll, Deborah Hix, Shrinivasrao P. Mane, Chunhong Mao, Eric K. Nordberg, Mark Scott, Julie R. Schulman, Eric E. Snyder, Daniel E. Sullivan, Chunxia Wang, Andrew Warren, Kelly P. Williams, Tian Xue, Hyun Seung Yoo, Chengdong Zhang, Yan Zhang, Rebecca Will, Ronald W. Kenyon et Bruno W. Sobral. Patric : the comprehensive bacterial bioinformatics

resource with a focus on human pathogenic species. *Infection and Immunity*, 79 (11):4286–4298, 2011.

[71] M. Goodman, J. Czelusniak, G.-W. Moore, A.-E. Romero-Herrera et G. Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Zoology*, 28:132–163, 1979.

[72] M. Goodman, G. W. Moore et G. Matsuda. Darwinian evolution in the genealogy of haemoglobin. *Nature*, 253:603–608, 1975.

[73] A. J. E. Gordon et J. A. Halliday. Inversions with deletions and duplications. *Genetics*, 140:411–414, 1995.

[74] J.L. Gordon, K.P. Byrne et K.H. Wolfe. Additions, losses, and rearrangements on the evolutionary route from a reconstructed ancestor to the modern saccharomyces cerevisiae genome. *PloS Genetics*, 5(5):e1000485, 2009.

[75] P. Gorecki et J. Tiuryn. DLS-trees : a model of evolutionary scenarios. *Theoretical Computer Science*, 359:378–399, 2006.

[76] R. Guigó, I. Muchnik et T.-F. Smith. Reconstruction of ancient molecular phylogeny. *Molecular Phylogenetics and Evolution*, 6:189–213, 1996.

[77] M.-W. Hahn, M.-V. Han et S.-G. Han. Gene family evolution across 12 *drosophilia* genomes. *PLoS Genetics*, 3 :e197, 2007.

[78] S. Hannenhalli. Polynomial-time algorithm for computing translocation distance between genomes. Dans *LNCS*, volume 937, pages 162 – 176, 1995.

[79] S. Hannenhalli et P. A. Pevzner. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). *JACM*, 48:1 – 27, 1999.

[80] S. Hannenhalli et P.A. Pevzner. Transforming men into mice. Dans *Proceedings of the IEEE 36th Annual Symposium on Foundations of Computer Science*, pages 581 – 592, 1995.

[81] K. Hokamp, A. McLysaght et K.-H. Wolfe. The 2r hypothesis and the human genome sequence. *Journal of Structural and Functional Genomics*, 3:95–110, 2003.

[82] P.-W. Holland, J. Garcia-Fernandez, N.-A. Williams et A. Sidow. Gene duplications and the origins of vertebrate development. *Development*, Suppl.:125–133, 1994.

[83] P. Holloway, K. Swenson, D. Ardell et N. El-Mabrouk. Evolution of genome organization by duplication and loss : An alignment approach. Dans Benny Chor, éditeur, *Research in Computational Molecular Biology*, volume 7262 de *Lecture Notes in Computer Science*, pages 94–112, 2012.

[84] J. E. Hopcroft et R. M. Karp. An $n^{(5/2)}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

[85] R. Horton, L. Wilming et V. Rand et al. Gene map of the extended human mhc. *Nature Reviews Genetics*, 5(12):889–899, 2004.

[86] Catherine M. Houck, Frank P. Rinehart et Carl W. Schmid. A ubiquitous family of repeated {DNA} sequences in the human genome. *Journal of Molecular Biology*, 132(3):289 – 306, 1979.

[87] Jaime Huerta-Cepas, Hernan Dopazo, Joaquin Dopazo et Toni Gabaldon. The human phylome. *Genome Biology*, 8(6):R109, 2007. ISSN 1465-6906.

[88] A.L. Hughes et M. Nei. Pattern of nucleotide substitution at mhc class i loci reveals overdominant selection. *Nature*, 335:167–170, 1988.

[89] Y. Ina. New methods for estimating the numbers of synonymous and nonsynonymous substitutions. *Journal of Molecular Evolution*, 40:190–226, 1995.

[90] O. Jaillon, J.-M. Aury, F. Brunet et et al. (62 coauthors). Genome duplication in the teleost fish tetraodon nigroviridis reveals the early vertebrate proto-karyotype. *Nature*, 431:946–957, 2004.

[91] D. Jones, W. Taylor et J. Thornton. The rapid generation of mutation data matrices from protein sequences. *Computer Applications in the Biosciences*, 8:275–282, 1992.

[92] M. Kellis, B.-W. Birren et E.-S. Lander. Proof and evolutionary analysis of ancient genome duplication in the yeast saccharomyces cerevisiae. *Nature*, 428:617–624, 2004.

[93] Motoo Kimura. Was globin evolution very rapid in its early stages ? : A dubious case against the rate-constancy hypothesis. *Journal of Molecular Evolution*, 17: 110–113, 1981.

[94] N. Kohmura, K. Senzaki, S. Hamada, N. Kai, R. Yasuda, M. Watanabe, H. Ishii, M. Yasuda, M. Mishina et T. Yagi. Diversity revealed by a novel family of cadherins expressed in neurons at a synaptic complex. *Neuron*, 20:1137–1151, 1998.

[95] E. V. Koonin. Orthologs, paralogs, and evolutionary genomics. *Annual Review of Genetics*, 39:309–338, 2005.

[96] A. Kotzig. Moves without forbidden transitions in a graph. *Matematicky casopis*, 18:76 – 80, 1968.

[97] M. Lajoie, D. Bertrand et N. El-Mabrouk. Inferring the evolutionary history of gene clusters from phylogenetic and gene order data. *Molecular Biology and Evolution*, 27:761-772, 2010.

[98] M. Lajoie, D. Bertrand, N. El-Mabrouk et O. Gascuel. Duplication and inversion history of a tandemly repeated genes family. *Journal of Computational Biology*, 14 (4):462-478, 2007.

[99] R.-S. LaRue, S.-R. Jonsson, K.-A.-T. Silverstein, M. Lajoie, D. Bertrand, N. El-Mabrouk, I. Hötzel, V. Andresdottir, T.-P.-L. Smith et R.-S. Harris. The artiodactyl APOBEC3 innate immune repertoire shows evidence for a multi-functional domain organization that existed in the ancestor of placental mammals. *BMC Molecular Biology*, 9:104, 2008.

[100] G. Li, Z. Qi, X. Wang et B. Zhu. A linear-time algorithm for computing translocation distance between signed genomes. Dans *LNCS*, volume 3109, pages 323–332, 2004.

[101] Li Li, Christian J. Stoeckert et David S. Roos. Orthomcl : Identification of ortholog groups for eukaryotic genomes. *Genome Research*, 13(9):2178–2189, 2003.

[102] W. H. Li. Unbiased estimation of the rates of synonymous and nonsynonymous substitution. *Journal of Molecular Evolution*, 36:96–99, 1993.

[103] W. H. Li et T. Gojobori. Rapid evolution of goat and sheep globin genes following gene duplication. *Molecular Biology and Evolution*, 1(1):94–108, 1983.

[104] W. H. Li, C. I. Wu et C. C. Luo. A new method for estimating synonymous and nonsynonymous rates of nucleotide substitution considering the relative likelihood

of nucleotide and codon changes. *Molecular Biology and Evolution*, 2(2):150–174, 1985.

[105] Benjamin Linard, Julie Thompson, Olivier Poch et Odile Lecompte. Orthoinspector : comprehensive orthology analysis and visual exploration. *BMC Bioinformatics*, 12(1):11, 2011. ISSN 1471-2105.

[106] L.-G. Lundin. Evolution of the vertebrate genome as reflected in paralogous chromosomal regions in man and the house mouse. *Genomics*, 16:1–19, 1993.

[107] M. Lynch et J. S. Conery. The evolutionary fate and consequences of duplicate genes. *Science*, 290(5494):1151–1155, 2000.

[108] E. Lyons et M. Freeling. How to usefully compare homologous plant genes and chromosomes as dna sequences. *The Plant Journal*, 53(4):661–673, 2008.

[109] E. Lyons, B. Pedersen, J. Kane et M. Freeling. The value of nonmodel genomes and an example using synmap within coge to dissect the hexaploidy that predates the rosids. *Tropical Plant Biology*, 1:181–190, 2008.

[110] M. A. Lysak, A. Berr, A. Pecinka, R. Schmidt, K. McBreen et I. Schubert. Mechanisms of chromosome number reduction in arabidopsis thaliana and related brassicaceae species. *Proceedings of the National Academy of Sciences USA*, 103(13): 5224–5229, 2006.

[111] B. Ma, M. Li et L. Zhang. From gene trees to species trees. *SIAM Journal on Computing*, 30:729–752, 2000.

[112] J. Ma, A. Ratan, B.-J. Raney, B.-B. Suh, L. Zhang, W. Miller et D. Haussler. Dupcar : Reconstructing contiguous ancestral regions with duplications. *Journal of Computational Biology*, 15(8), 2008.

[113] K. Marhold et J. Lihova. Polyploidy, hybridization and reticulate evolution : lessons from the brassicaceae. *Plant Systematics and Evolution*, 259:143–174, 2006.

[114] M. Marron, K.-M. Swenson et B.-M.-E Moret. Genomic distances under deletions and insertions. Dans *Proc. 9th Int'l Combinatorics and Computing Conf. (COCOON'03)*, volume Lecture Notes in Computer Science 2697, pages 537-547. Springer Verlag, 2003.

[115] A. McLysaght, K. Hokamp et K.-H. Wolfe. Extensive genomic duplication during early chordate evolution. *Nature Genetics*, 31:200–204, 2002.

[116] W. Messier et C.-B. Stewart. Episodic adaptative evolution of primate lysozymes. *Nature*, 385:151–154, 1997.

[117] J. Messing, A.-K. Bharti, W.-M. Karlowski, H. Gundlach, H.-R. Kim, Y. Yu, F. Wei, G. Fuks, C.-A. Soderlund et K.-F. Mayer. Sequence composition and genome organization of maize. *Proceedings of the National Academy of Sciences USA*, 101: 14349- 14354, 2004.

[118] J. Mixtacki. Genome halving under dcj revisited. Dans *Proceedings of COCOON'08*, volume Lecture Notes in Computer Science 5092, pages 276–286. Springer Verlag, 2008.

[119] T. Miyata et T. Yasunaga. Molecular evolution of mrna : A method for estimating evolutionary rates of synonymous and amino acid substitutions from homologous nucleotide sequences and its application. *Journal of Molecular Evolution*, 16:23–36, 1980.

[120] K. Naruse, M. Tanaka, K. Mita, A. Shima, J. Postlethwait et H. Mitani. A me-

daka gene map : the trace of ancestral vertebrate proto-chromosomes revealed by comparative gene mapping. *Genome Research*, 14:820–828, 2004.

[121] N. L. Nehrt, W. T. Clark, P. Radivojac et M. W. Hahn. Testing the ortholog conjecture with comparative functional genomic data from mammals. *PLoS Comput Biol*, 7(6):e1002073, 06 2011.

[122] M. Nei et T. Gojobori. Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions. *Molecular Biology and Evolution*, 3 (5):418–426, 1986.

[123] J. Noonan, J. Grimwood, J. Schmutz, M. Dickson et R. Myers. Gene conversion and the evolution of protocadherin gene cluster diversity. *Genome Research*, 14: 354–366, 2004.

[124] R. A. Notebaart, M. A. Huynen, B. Teusink, R. J. Siezen et B. Snel. Correlation between sequence conservation and the genomic context after gene duplication. *Nucleic Acids Research*, 33(19):6164–6171, 2005.

[125] S. Ohno. *Evolution by gene duplication*. Springer, Berlin, 1970.

[126] E.-M. Ostertag et H.-H.-Jr. Kazazian. Biology of mammalian l1 retrotransposons. *Annual Review of Genetics*, 35:501–538, 2001.

[127] J.L. Boore P. Dehal. Two rounds of whole genome duplication in the ancestral vertebrate. *Plos Biology*, 3(10):e314, 2005.

[128] R.-D.-M. Page. Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Systematic Biology*, 43:58–77, 1994.

[129] R.-D.-M. Page. Genetree : comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14:819–820, 1998.

[130] R.-D.-M Page et M.-A. Charleston. Reconciled trees and incongruent gene and species trees. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 37:57–70, 1997.

[131] B. Papp, C. Paul et L.-D. Hurst. Dosage sensitivity and the evolution of gene families in yeast. *Nature*, 424:194–197, 2003.

[132] A.-H. Paterson, J.-E. Bowers et R. Bruggmann et al. The sorghum bicolor genome and the diversification of grasses. *Nature*, 457:551–556, 2009.

[133] P. Pevzner et G. Tesler. Transforming men into mice : the nadeau-taylor chromosomal breakage model revisited. Dans *Proceedings of the seventh annual international conference on Research in computational molecular biology RECOMB'03*, pages 247–256, 2003.

[134] P.A. Pevzner. Dna physical mapping and alternating eulerian cycles in colored graphs. *Algorithmica*, 13:77 – 105, 1995.

[135] A. Plaitakis, C. Spanaki, V. Mastorodemos et I. Zaganas. Study of structureŰfunction relationships in human glutamate dehydrogenases reveals novel molecular mechanisms for the regulation of the nerve tissue-specific (glud2) isoenzyme. *Neurochemistry International*, 43:401 – 410, 2003.

[136] M. Remm, C. E. V. Storm et E. L. L. Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of Molecular Biology*, 314:1041–1052, 2001.

[137] H.H. Rogers, C.M. Bergman et S. Griffiths-Jones. The evolution of tRNA genes in Drosophila. *Genome Biology and Evolution*, 2:467- 477, 2010.

[138] F. Ronquist et J.-P. Huelsenbeck. Mrbayes 3 : Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19:1572–1574, 2003.

[139] Claude-Alain H. Roten, Patrick Gamba, Jean-Luc Barblan et Dimitri Karamata. Comparative genometrics (cg) : a database dedicated to biometric comparisons of whole genomes. *Nucleic Acids Research*, 30(1):142–144, 2002.

[140] S. Rouquier, A. Blancher et D. Giorgi. The olfactory receptor gene repertoire in primates and mouse : Evidence for reduction of the functional fraction in primates. *Proc. Natl. Acad. Sci. USA*, 97(6):2870–2874, 2000.

[141] S. Rouquier, S. Taviaux, B.-J. Trask, V. Brand-Arpon, G. van den Engh, J. Demaille et D. Giorgi. Distribution of olfactory receptor genes in the human genome. *Nature Genetics*, 18:243–250, 2000.

[142] J. Salse, S. Bolot, M. Throude, V. Jouffe, B. Piegu, U.M. Quraishi, T. Calcagno, R. Cooke, M. Delseny et C. Feuillet. Identification and characterization of shared duplications between rice and wheat provide new insight into grass genome evolution. *The Plant Cell*, 20:11 – 24, 2008.

[143] D. Sankoff et M. Blanchette. The median problem for breakpoints in comparative genomics. Dans *Proceedings of the Third International Computing and Combinatorics Conference COCOON'97*, pages 251–263, 1997.

[144] D. Sankoff et M. Blanchette. Multiple genome rearrangement and breakpoint phylogeny. *Journal of computational biology*, 5:555–570, 1998.

[145] O. Tremblay Savard, D. Bertrand et N. El-Mabrouk. Evolution of orthologous tandemly arrayed gene clusters. *BMC Bioinformatics*, 18(Suppl 9):S2, 2011.

[146] O. Tremblay Savard, Y. Gagnon, D. Bertrand et N. El-Mabrouk. Genome halving and double distance with losses. *Journal of Computational Biology*, 18(9):1185–1199, 2011.

[147] E. Schildkraut, C. A. Miller et J. A. Nickoloff. Gene conversion and deletion frequencies during double-strand break repair in human cells are controlled by the distance between direct repeats. *Nucleic Acids Research*, 33:1574–1580, 2005.

[148] I. Schubert et M. A. Lysak. Interpretation of karyotype evolution should consider chromosome structural constraints. *Trends in Genetics*, 27:207–216, 2011.

[149] S. Schwartz, W.-J. Kent, A. Smit et *et al.* Human-mouse alignments with blastz. *Genome Research*, 13:103–107, 2003.

[150] V. Shoja et L. Zhang. A roadmap of tandemly arrayed genes in the genomes of human, mouse, and rat. *Molecular Biology and Evolution*, 23:2134–2141, 2006.

[151] D.E. Soltis, V.A. Albert, J. Leebens-Mack, C.D. Bell, A.H. Paterson, C. Zheng, D. Sankoff, C.W. dePamphilis, P.K Wall et P.S. Soltis. Polyploidy and angiosperm diversification. *American Journal of Botany*, 96:336 – 348, 2009.

[152] G. Song, L. Zhang, T. Vinar et W. Miller. Inferring the recent duplication history of a gene cluster. Dans F.D. Ciccarelli et I. Miklós, éditeurs, *Comparative Genomics*, volume 5817 de *Lecture Notes in Computer Science*. Springer, 2009.

[153] G. Song, L. Zhang, T. Vinar et W. Miller. Cage : Combinatorial analysis of gene-cluster evolution. *Journal of Computational Biology*, 17(9):1227–1242, 2010.

[154] E. L. Sonnhammer et E. V. Koonin. Orthology, paralogy and proposed classification for paralog subtypes. *Trends in Genetics*, 18(12):619 – 620, 2002. ISSN 0168-9525.

[155] J. Spring. Vertebrate evolution by interspecific hybridisation - are we polyploid ? *FEBS Letters*, 400:2–8, 1997.

[156] P. Stankiewicz et J. R. Lupski. Genome architecture, rearrangements and genomic disorders. *TRENDS in Genetics*, 18(2):74–82, 2002.

[157] A. H. Sturtevant. The linear arrangement of six sex-linked factors in drosophila, as shown by their mode of association. *Journal of Experimental Zoology*, 14:43–59, 1913.

[158] M. Tang, M.-S. Waterman et S. Yooseph. Zinc finger gene clusters and tandem gene duplication. Dans *Research in Molecular Biology (RECOMB 2001)*, pages 297–304, 2001.

[159] E. Tannier, C. Zheng et D. Sankoff. Multichromosomal median and halving problems under different genomic distances. *BMC Bioinformatics*, 10(120), 2009.

[160] G. Tesler. Efficient algorithms for multichromosomal genome rearrangements. *Journal of Computer and System Sciences*, 65:587 – 609, 2002.

[161] The Arabidopsis Genome Initiative. Analysis of the flowering plant *arabiopsis thaliana*. *Nature*, 408:796- 815, 2000.

[162] E.R.M. Tillier et R.A. Collins. Genome rearrangement by replication-directed translocation. *Nature Genetics*, 26, 2000.

[163] O. Tremblay-Savard et K. M. Swenson. A graph-theoretic approach for inparalog detection. *BMC Bioinformatics*, 13(Suppl 19):S16, 2012.

[164] A. van Hoof. Conserved functions of yeast genes support the duplication, degeneration and complementation model for gene duplication. *Genetics*, 171:1455 – 1461, 2005.

[165] R.-A. Veitia. Exploring the etiology of haploinsufficiency. *BioEssays*, 24:175 – 184, 2002.

[166] R.-A. Veitia. Gene dosage balance : deletions, duplications and dominance. *Trends in Genetics*, 21:33 – 35, 2005.

[167] R.-A. Veitia. Cellular reactions to gene dosage imbalance : genomic, transcriptomic and proteomic effects. *Trends in Genetics*, 24:390 – 397, 2008.

[168] Tomáš Vinař, Broňa Brejová, Giltae Song et Adam Siepel. Reconstructing histories of complex gene clusters on a phylogeny. *Journal of Computational Biology*, 17: 1267–1269, 2010.

[169] X. Wang, J.-A. Weiner, S. Levi, A.-M. Craig, A. Bradley et J.-R. Sanes. Gamma protocadherins are required for survival of spinal interneurons. *Neuron*, 36:843–854, 2002.

[170] I. Wapinski, A. Pfeffer, N. Friedman et A. Regev. Natural history and evolutionary principles of gene duplication in fungi. *Nature*, 449:54–61, 2007.

[171] R. Warren et D. Sankoff. Genome halving with double cut and join. *Journal of Bioinformatics and Computational Biology*, 7:357–371, 2009.

[172] G. Watterson, W. Ewens, T. Hall et A. Morgan. The chromosome inversion problem. *Journal of Theoretical Biology*, 99:1–7, 1982.

[173] J.-A. Weiner, X. Wang, J.-C. Tapia et J.-R. Sanes. Gamma protocadherins are required for synaptic development in the spinal cord. *PNAS*, 102:8–14, 2005.

[174] J. Wienberg, A. Jauch, H. J. Lüdecke, G. Senger, B. Horsthemke, U. Claussen, T. Cremer, N. Arnold et C. Lengauer. The origin of human chromosome 2 analyzed by comparative chromosome mapping with a dna microlibrary. *Chromosome Research*, 2:405–410, 1994.

[175] J. Wienberg, A. Jauch, R. Stanyon et T. Cremer. Molecular cytotaxonomy of primates by chromosomal in situ supression hybridization. *Genomics*, 8:347–350, 1990.

[176] M. Withers, L. Wernisch et M. Dos Reis. Archaeology and evolution of transfer RNA genes in the *escherichia coli* genome. *Bioinformatics*, 12:933-942, 2006.

[177] I.-G. Woods, C. Wilson, B. Friedlander, P. Chang, D.-K. Reyes, R. Nix, P.-D. Kelly, F. Chu, J.-H. Postlethwait et W.-S. Talbot. The zebrafish gene map defines ancestral vertebrate chromosomes. *Genome Research*, 15:1307–1314, 2005.

[178] Q. Wu. Comparative genomics and diversifying selection of the clustered vertebrate protocadherin genes. *Genetics*, 169:2179–2188, 2005.

[179] Q. Wu et T. Maniatis. A striking organization of a large family of human neural cadherin-like cell adhesion genes. *Cell*, 97:779–790, 1999.

[180] T. Yagi. Clustered protocadherin family. *Development, growth & differentiation*, 50:S131–S140, 2008.

[181] S. Yancopoulos, O. Attie et R. Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21:3340 – 3346, 2005.

[182] Z. Yang. Likelihood ratio tests for detecting positive selection and application to primate lysozyme evolution. *Molecular Biology and Evolution*, 15:568–573, 1998.

[183] Z. Yang et R. Nielsen. Estimating synonymous and nonsynonymous substitution rates under realistic evolutionary models. *Molecular Biology and Evolution*, 17: 32–43, 2000.

[184] J. Zhang, A.-M. Dean, F. Brunet et M. Long. Evolving protein functional diversity in new genes of drosophila. *Proceedings of the National Academy of Sciences USA*, 101(46):16246–16250, 2004.

[185] L. Zhang, B. Ma, L. Wang et Y. Xu. Greedy method for inferring tandem duplication history. *Bioinformatics*, 19:1497–1504, 2003.

[186] L.-X. Zhang. On Mirkin-Muchnik-Smith conjecture for comparing molecular phylogenies. *Journal of Computational Biology*, 4:177–188., 1997.

[187] Y. Zhang, G. Song, C.-H Hsu et W. Miller. Simultaneous history reconstruction for complex gene clusters in multiple species. Dans *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 162–173, 2009.

[188] Y. Zhang, G. Song, T. Vinar, E.-D. Green, A. Siepel et W. Miller. Reconstructing the evolutionary history of complex human gene clusters. Dans M.Vingron et L.Wong, éditeurs, *Research in Computational Molecular Biology. (RECOMB 2008)*, volume 4955 de *Lecture Notes in Computer Science*, pages 29–49. Springer, 2008.

[189] Y. Zhang, G. Song, T. Vinar, E.-D. Green, A. Siepel et W. Miller. Evolutionary history reconstruction for mammalian complex gene clusters. *Journal of Computational Biology*, 16:1051–1070, 2009.

[190] C. Zheng, Q. Zhu et D. Sankoff. Genome halving with an outgroup. *Evolutionary Bioinformatics*, 2:319–326, 2006.

[191] C. Zheng, Q. Zhu et D. Sankoff. Descendants of whole genome duplication within gene order phylogeny. *Journal of Computational Biology*, 15(8):947 – 964, 2008.

[192] Chunfang Zheng, Krister M. Swenson, Eric Lyons et David Sankoff. Omg ! orthologs in multiple genomes - competing graph-theoretical formulations. Dans *WABI*, volume 6833 de *Lecture Notes in Computer Science*, pages 364–375. Springer, 2011.

[193] Chunfang Zheng, Qian Zhu, Zaky Adam et David Sankoff. Guided genome halving : hardness, heuristics and the history of the hemiascomycetes. *Bioinformatics*, 24(13):i96–i104, 2008.

[194] L. Zhou, B. Huang, X. Meng, G. Wang, F. Wang, Z. Xu et R. Song. The amplification and evolution of orthologous 22-kDa α-prolamin tandemly arrayed genes in *coix*, sorghum and maize genomes. *Plant Molecular Biology*, 74:631- 643, 2010.

[195] C. M. Zmasek et S. R. Eddy. A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics*, 17:821–828, 2001.