THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Discipline ou spécialité :
Informatique
Spécialité Bioinformatique

Présentée et soutenue par :
Eric AUDEMARD

le : lundi 28 novembre 2011

Titre :

Détection des duplications en tandem au niveau nucléique à l'aide de la théorie des flots

---

Ecole doctorale :
Mathématiques Informatique Télécommunications (MITT)

Unité de recherche :
Unité INRA BIA (UR 875), Toulouse

Directeur(s) de Thèse :

Thomas Faraut & Thomas Schiex

Rapporteurs :

Sébastien Aubourg (INRA, Evry) Philippe Monget (INRA, Tours) Alain Viari (INRIA, Grenoble)

Autre(s) membre(s) du jury

Caludette Cayrol (Université Paul Sabatier), Présidente

Remerciements

En rédigeant cette dernière page de manuscrit, je suis bien obligé de reconnaître que cette thèse est le fruit d'un peu de recherche et de beaucoup d'aide reçue de nombreuses personnes que je tiens à remercier ici. J'espère n'oublier personne, et si tel était le cas, prévenez-moi, et je me ferai un plaisir de corriger cet oubli et de vous offrir une version dédicacée de ce manuscrit.

Je tiens tout d'abord à remercier mes directeurs de thèse, Thomas Faraut et Thomas Schiex (ou Thomas^2), pour leur encadrement continu et vigilant. J'ai pu apprécier et m'inspirer de leurs méthodes de travail et profiter de leurs connaissances tant en informatique qu'en biologie.

Merci également aux membres de mon jury et notamment aux trois rapporteurs de ce travail, Sébastien Aubourg, Philippe Monget et Alain Viari, qui en plus d'avoir relu et commenté ce manuscrit ont renvoyé des rapports complets qui m'ont aidé à préparer ma soutenance.

Parmi les nombreux permanents de l'unité BIA dans laquelle j'ai été accueilli, je tiens à remercier tout particulièrement, les membres de mon bureau (Aurélie Favier, Mahuna Akplogan et Ronan Trepos) ainsi que Céline Noirot et Gauthier Quesnel (le roi du C++) pour leur bonne humeur et leurs conseils toujours avisés. J'ai passé de très bons moments en votre compagnie et j'espère qu'on ne s'arrêtra pas là.

Je tiens évidemment à remercier ma famille et mes amis, pour leur soutien du début (et même avant) jusqu'à la fin de ce travail. Même si mes activités semblaient mystérieuses, vous savez maintenant que je travaillais. Malgré ma présence en pointillés, vous m'avez toujours soutenu et une très grande partie du travail que j'achève maintenant vous est due !

Enfin, je tiens à remercier l'école doctorale MITT qui m'a soutenu jusqu'au bout.

Sommaire

Préambule

L'apparition de nouvelles fonctions biologiques est un corollaire de la théorie de l'évolution et a fait à ce titre l'objet de nombreuses recherches. Le processus d'acquisition de nouvelles fonctions par l'intermédiaire de duplication de gènes, popularisé par l'ouvrage de Susumu Ohno "Evolution by gene duplication", a connu un regain d'intérêt depuis la révolution génomique. L'analyse des génomes entièrement séquencés a en effet révélé d'une part l'étendue du phénomène de duplication au sein des génomes de tous les règnes et d'autre part une dynamique, jusqu'alors insoupçonnée, de la contraction et de l'expansion des familles de duplications entre les différentes lignées évolutives. Les duplications chromosomiques constituent la matière première de ce processus de duplication de gènes, conduisant potentiellement à l'apparition d'une nouvelle fonction, et présentent donc un intérêt tout à fait particulier pour la compréhension de ce processus d'acquisition.

L'objectif de cette thèse est de tenter d'identifier au sein de séquences génomique la présence de duplications de segments chromosomiques ou duplications segmentales. Contrairement aux méthodes qui identifient les gènes dupliqués à partir de l'information des séquences protéiques fournies par l'annotation, nous exploitons uniquement l'information fournie par la séquence d'ADN. Cette approche permet d'une part de s'affranchir des problèmes d'annotation mais elle permet surtout d'aborder le problème de la duplication indépendamment de la nature fonctionnelle des segments dupliqués. Dans ce manuscrit, nous commençons par décrire brièvement, dans l'introduction, le phénomène de duplication au sein des génomes. Dans le 2^{e} chapitre nous étudions les méthodes qui détectent les duplications, puis nous décrirons la méthode développée lors de cette thèse, dans le 3^{e} chapitre. Enfin nous analyserons les résultats obtenus, avant de conclure sur ce travail de recherche.

Chapitre 1

Introduction

Sommaire

Dans ce chapitre d'introduction, la section **génome** présente les bases élémentaires de biologie moléculaire nécessaires à la compréhension de ce manuscrit. La complexité et la richesse des phénomènes biologiques ne seront que sommairement décrits. Ensuite, la section **évolution des génomes** décrit le contexte et les motivations scientifiques. Pour finir la section **détection ab initio des duplications en tandem** précise les objectifs que je me suis fixés pendant la thèse.

1.1 Le génome

Tout être vivant est constitué de cellules. La cellule contient l'ensemble de l'information génétique appelée génome qui caractérise l'individu et son espèce. La présence d'un noyau, compartiment spécifique renfermant le génome, permet de distinguer dans la classification du vivant les organismes *eucaryotes* des organismes *procaryotes*, qui en sont dépourvus. Le génome est conservé dans une molécule, l'acide désoxyribonucléique ou ADN.

1.1.1 La séquence d'ADN

La structure de l'ADN est connue depuis 1953, grâce à la célèbre publication de J.Watson et F. Crick (Watson et Crick, 1953). Elle se présente sous la forme d'une hélice composée de deux brins (Fig. 1.1 page ci-contre). Chaque brin est constitué d'un enchaînement linéaire et orienté d'éléments appelés nucléotides. Dans l'ADN il existe quatre types de nucléotides représentés par la première lettre de leurs noms : A, C, G et T (A. Kossel 1896).

Remarque 1 *Dans un brin d'ADN, un nucléotide est lié au précédent (resp. au suivant) par un atome de carbone en position* $5'$ *(resp.* $3'$*). Cette propriété est utilisée pour donner un sens à l'orientation d'un brin d'ADN :* $5' \rightarrow 3'$.

Les deux brins composant l'hélice de l'ADN génomique sont "antiparallèles", car ils sont parallèles mais leur orientation $5' \rightarrow 3'$ est opposée. Leur association est rendue possible grâce aux appariements entre leurs nucléotides respectifs, chaque nucléotide de l'un pouvant s'associer à un nucléotide lui faisant face. Les paires de nucléotides ainsi formées impliquent presque exclusivement soit un A avec un T, soit un G avec un C (Fig. 1.1 page suivante). Ce type d'appariement appelé "Watson-Crick" définit la complémentarité entre deux brins.

Définition 1 (La séquence d'ADN) *est une succession de lettres A, C, G et T identifiant les nucléotides successifs de l'un des brins de la double hélice dans le sens* $5' \rightarrow 3'$.

Remarque 2 *La séquence de l'autre brin se déduit sans ambiguïté par complémentarité (et inversion pour garder le sens* $5' \rightarrow 3'$*). Par exemple GATTACA a pour reverse complément TGTAATC.*

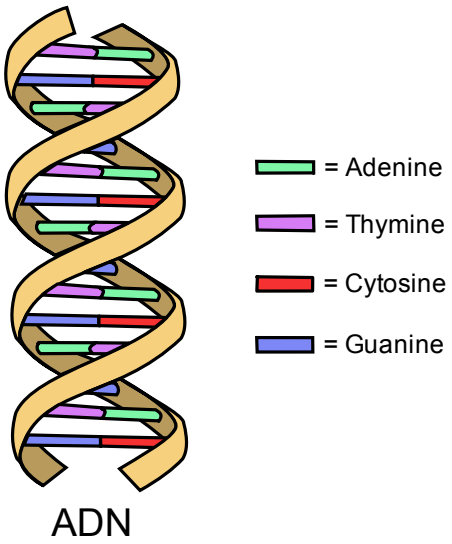


FIGURE 1.1: Structure schématique de la double hélice de l'ADN

Un nucléotide se nomme parfois *base nucléotidique*. C'est à partir de cette appellation qu'une unité de mesure de l'ADN, composé de deux brins, a été définie en paire de base (bp). Il faut aussi noter que chez les *eucaryotes* (et quelques procaryotes) la séquence d'ADN est scindée en plusieurs chromosomes, chaque chromosome étant composé d'une molécule d'ADN. Le nombre de chromosomes est variable et dépend de l'espèce, l'espèce humaine en a 22 paires plus 2 chromosomes sexuels et la certaines espèces de fougère peuvent en posséder jusqu'à 1200.

1.1.2 Les gènes et les séquences protéiques

La fonction fondamentale de tout génome est de maintenir et transmettre l'information génétique, portée par certaines régions dites *fonctionnelles*. Ces régions fonctionnelles sont classées en plusieurs catégories (ou sous-catégories), comme les gènes d'ARN, de miARN, les gènes codants pour les protéines et bien d'autres. Dans cette section nous allons nous focaliser sur les gènes codant pour des protéines.

Le terme protéine vient du grec ancien *prôtos* qui signifie premier ou essentiel. Une autre théorie, voudrait que le mot protéine fasse référence au dieu grec *Protée* qui pouvait changer de forme à volonté. Les protéines adoptent en effet de multiples formes et assurent de multiples fonctions essentielles au bon fonctionnement des cellules. Il est intéressant de noter que malgré l'importance des gènes dans le fonctionnement des cellules, les régions codantes ne concernent qu'une très petite partie de l'ADN. Par exemple, chez l'homme elles ne concernent que 3% de l'ADN.

Comme l'ADN, une protéine est composée d'un assemblage de plusieurs unités, qui sont ici des acides aminés. La fonction d'une protéine est déterminée par sa séquence d'acides aminés, elle-même étant déterminée par la séquence nucléotidique du gène correspondant (voir tableau 1.1).

Définition 2 (La séquence protéique) *est une succession de lettres, dans un alphabet de 20 lettres, identifiant les acides aminés qui composent la protéine.*

1^{re} position (5′)	2^{e} position T		2^{e} position C		2^{e} position A		2^{e} position G		3^{e} position (3′)
T	TTT	Phe:F	TCT	Ser:S	TAT	Tyr:Y	TGT	Cys:C	T
	TTC		TCC		TAC		TGC		C
	TTA	Leu:L	TCA		TAA	STOP	TGA	STOP	A
	TTG		TCG		TAG		TGG	Trp:W	G
C	CTT	Leu:L	CCT	Pro:P	CAT	His:H	CGT	Arg:R	T
	CTC		CCC		CAC		CGC		C
	CTA		CCA		CAA	Gln:Q	CGA		A
	CTG		CCG		CAG		CGG		G
A	ATT	Ile:I	ACT	Thr:T	AAT	Asn:N	AGT	Ser:S	T
	ATC		ACC		AAC		AGC		C
	ATA		ACA		AAA	Lys:K	AGA	Arg:R	A
	ATG	Met:M	ACG		AAG		AGG		G
G	GTT	Val:V	GCT	Ala:A	GAT	Asp:D	GGT	Gly:G	T
	GTC		GCC		GAC		GGC		C
	GTA		GCA		GAA	Glu:E	GGA		A
	GTG		GCG		GAG		GGG		G

TABLE 1.1: Tableau du code génétique.

L'expression génique est le mécanisme moléculaire qui assure la synthèse protéique à partir de la séquence d'ADN. Pour comprendre certaines caractéristiques de l'évolution des génomes il est nécessaire de décrire ce mécanisme. Ici seule l'expression génique eucaryote, qui concerne tous les êtres vivants à l'exception des bactéries, sera brièvement décrite.

Chez les eucaryotes, l'essentiel de la fabrication des protéines a lieu à l'extérieur du noyau cellulaire. Par conséquent, la séquence nucléotidique du gène doit y être transportée à l'aide de l'*ARN messager*. L'expression génique se décompose en trois étapes majeures comprenant la synthèse d'un précurseur d'ARN messager à partir de l'ADN génomique (transcription et épissage), sa maturation en ARN messager, et la construction de la protéine (traduction) à partir de l'ARN messager (Fig. 1.2 page suivante).

FIGURE 1.2: Schéma de l'expression génique eucaryote. Cette figure est aussi l'occasion d'introduire des thermes biologiques, comme le codon start composé de des trois nucléotides ATG, par exemple.

Nous venons de voir que les protéines étaient construites à partir de séquences nucléotidiques de l'ADN. Ainsi pour reproduire une protéine de façon identique de génération en génération, sa séquence doit rester identique au cours du temps. Une modification d'un seul nucléotide peut modifier la protéine et sa fonction ou empêcher sa traduction.

1.2 L'évolution des génomes

L'évolution des espèces et des génomes sont des mécanismes intimement liés. Cette évolution est envisagée comme la transformation d'individus au sein d'un groupe d'une même espèce. Le principe de base date de Darwin, les individus évoluent par l'action de la sélection naturelle. Une population évolue quand une sous-population d'individus possède une même "modification" (évolution). Ces modifications peuvent être de différentes natures ; elles se manifestent par exemple par une nouvelle version d'un gène ou par un nombre différent de copies d'un même gène. Je reviendrai sur ce point dans la sous-section 1.2.3 page 12.

Lorsque ces modifications s'accumulent dans une sous-population, elles peuvent mener à la constitution d'une barrière reproductive au sein d'une même espèce : une nouvelle espèce se crée. Ce phénomène est appelé la spéciation (Mayr (1942)).

Une fois la barrière d'espèce établie, les génomes des deux espèces sœurs vont évoluer indépendamment. Ainsi, les espèces soeurs partagent des caractéristiques chromosomiques héritées de l'ancêtre avec des différences, dues aux modification chromosomiques, qui se sont fixées dans chacune des sous-espèces depuis l'évènement de spéciation.

1.2.1 Les séquences homologues

En biologie de l'évolution, une homologie désigne un lien évolutif entre deux caractères observés chez deux individus (ou espèces) différent(e)s, provenant de l'héritage d'un ancêtre commun. Un caractère, pour un organisme vivant, est un de ses aspects anatomique, physiologique, moléculaire ou comportemental, qui peut-être analysé (par exemple : la présence de cheveux). Et l'ensemble des caractères observables d'un individu est appelé *phénotype*.

Remarque 3 *Il existe deux types d'homologies : l'homologie anatomique et l'homologie moléculaire. Ici nous ne traiterons que d'homologie moléculaire, à savoir deux régions de l'ADN (ou protéines) qui sont homologues.*

Définition 3 (Homologie) *Deux ou plusieurs séquences sont homologues si et seulement si elles dérivent d'une séquence ancestrale commune. Par extension, des gènes sont homologues s'ils dérivent d'un même gène ancestral.*

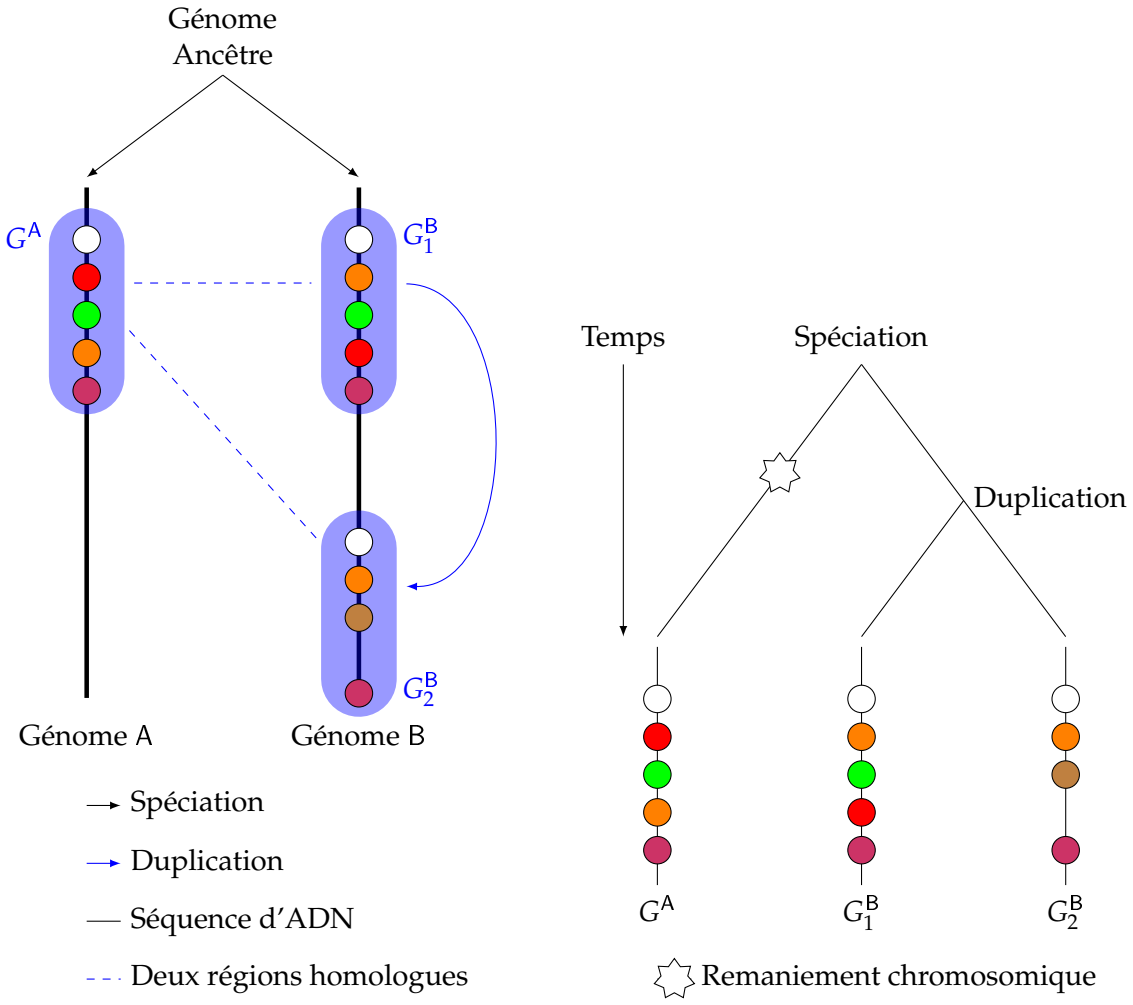


FIGURE 1.3: Schéma qui montre deux séquences génomiques partageant des caractères hérités d'un ancêtre, et l'évolution de ces caractères au cours du temps. On observe un remaniement chromosomique qui a inversé l'ordre des régions rouge verte et orange entre G^A et G_1^B, la modification d'une région verte en une région marron et la disparition de la région rouge sur G_2^B. C'est trois régions conservent cependant une **structure** similaire.

L'homologie se manifeste généralement par une similarité significative entre les séquences concernées. Et c'est précisément la recherche de similarité entre séquences qui permet d'établir une relation d'homologie. Il est admis qu'à partir d'un certain niveau de similarité deux séquences sont considérées comme homologues (voir section 2.1 page 16).

1.2.2 Comment évoluent les séquences ?

L'organisation actuelle des séquences est le résultat d'une histoire évolutive complexe, faite de modifications chromosomiques de différentes natures. L'évolution des chromosomes est un processus qui se déroule à différentes échelles. Aux modifications locales n'impliquant que quelques nucléotides, s'ajoutent des ré-arrangements microscopiques locaux pouvant impliquer de quelques milliers à quelques dizaines de milliers de nucléotides et des ré-arrangements de plus grande ampleur portant sur plusieurs millions de nucléotides voire des bras chromosomiques entiers (voir figure 1.3 page précédente).

Je classe les mécanismes d'évolutions dans deux catégories, ceux qui préservent le nombre de gènes et ceux qui modifient le nombre de gènes.

1.2.2.1 Mutations ne modifiant pas le nombre de gènes

Il existe deux mécanismes qui rentrent dans cette catégorie :

Les remaniements chromosomiques :

Les remaniements chromosomiques sont liés à des cassures suivies de recollements. Ces recollements surviennent immédiatement après les cassures, mais leur mise en œuvre peut aboutir à des reconstructions chromosomiques qui ne correspondent plus à la configuration initiale d'où l'apparition de chromosomes remaniés.

Ceci explique que des espèces qui partagent un grand nombre de gênes, voire même la totalité, peuvent présenter des organisations chromosomiques très différentes.

Les mutations nucléotidiques :

Il existe deux types de mutations nucléotidiques. La **substitution** qui se traduit par le changement d'un nucléotide par un autre, et l'**insertion** ou la **délétion** d'un ou plusieurs nucléotides.

Ces modifications mineures peuvent avoir de gros impacts sur le génome, si elles surviennent dans des régions codantes. Si une seule substitution modifie un codon start/stop ou un site accepteur/donneur d'épissage (voir figure 1.2 page 7), elle peut empêcher la transcription ou l'épissage et ainsi transformer une région qui était codante en région non-codante de type intergénique. Les mêmes conséquences sont à prévoir avec une

insertion/délétion de nucléotides non multiple de 3, ce qui décale la phase de lecture de la transcription.

Elles peuvent aussi n'avoir qu'un impact plus faible comme la modification/insertion ou la délétion d'un acide aminé, voir très faible avec la modification d'un codon sans changer l'acide aminé (substitution silencieuse) (voir tableau 1.1 page 6).

Définition 4 (substitution synonyme) *Une substitution synonyme désigne une substitution silencieuse qui touche un exon, sans changer la séquence de la protéine*

Remarque 4 *Les contraintes génomiques ne s'exercent pas de la même façon sur les régions codantes (considérablement contraintes) que sur les non-codantes. Ceci se traduit par une différence de la quantité de mutations accumulées au cours de l'évolution dans les régions codantes par rapport au reste du génome. Hughes et Yeager (1997) ont montré que le nombre de substitutions dans les introns était 3 fois supérieur aux substitutions non-synonymes des exons, chez le rat.*

1.2.2.2 Mutations modifiant le nombre de gènes : les duplications

Le phénomène de duplication de matériel génétique est l'une des forces majeures de l'évolution des génomes, de part sa récurrence. Comme en témoigne l'abondance des séquences répétées en tandem, des satellites [1] ou des séquences répétées dispersées dans le génome [2], qui peuvent représenter à eux seuls jusqu'à 50% des génomes de mammifères (Smit, 1999). On trouve également des duplications complètes de génomes, qui sont encore fréquentes chez les plantes et chez certains poissons.

Le phénomène de duplication présente également un intérêt particulier pour l'évolution fonctionnelle des génomes, car la duplication de gène est synonyme de naissance de gènes. Ce phénomène de duplication de gènes fut observé pour la première fois par Bridges chez la drosophile (Bridges, 1936). Cette capacité à modifier le répertoire génique d'un organisme a conduit Ohno (Ohno, 1970) à proposer que la duplication constituait l'une des principales forces évolutives des organismes supérieurs. En effet la duplication de gènes permet souvent à un organisme d'acquérir de nouvelles fonctions. La théorie proposée par Susumu Ohno stipule qu'à la suite d'une duplication, l'une des copies conserve la fonction du gène initial tandis que l'autre copie devient libre d'accumuler des mutations et de potentiellement coder pour une nouvelle protéine : un nouveau gène est né. Les projets de séquençage complet des génomes ont montré que la proportion de gènes dupliqués était comprise entre 30% et 50% dans les trois phylums(domaines) de la vie : bactéries, archæbactéries et eucaryotes (Zhang, 2003).

1. Les minisatellites ou microsatellites sont des séquences d'ADN formées par une répétition continue de motifs composés de quelques nucléotides ($\simeq$ 20)
2. *interspearsed repeats* en anglais

Définition 5 (Duplication en tandem) *La duplication est dite en tandem si les régions dupliquées sont adjacentes. Lorsque les régions correspondent à des gènes, on parle également de succession de gènes en tandem ou* Tandem Gene Arrays.

Les mécanismes moléculaires responsables de ces phénomènes de duplication sont encore mal compris même si l'on sait que les duplications en tandem résultent principalement de recombinaisons homologues inégales (Lynch, 2007), que les séquences dispersées sont principalement le fait de transpositions. Pour les gènes, la rétrotranscription d'ARN messager est également un mécanisme pouvant conduire à la duplication du gène parent (Kaessmann *et al.*, 2009). Les duplications complètes de génomes sont bien sûr de nature tout à fait différentes et se produisent pendant la méiose ou dans les premières mitoses de l'organisme. Quoi qu'il en soit, ces mécanismes moléculaires ne ciblent pas préférentiellement des séquences non-codantes ou des séquences codant pour des protéines. Ainsi les duplications peuvent impliquer tout type de régions qu'elles soient fonctionnelles ou non. Les exemples d'éléments fonctionnels dupliqués foisonnent dans la littérature Voinnet (2004); Nagaswamy et Fox (2003).

Remarque 5 *Suivant un usage assez répandu dans la littérature, nous dirons que l'on observe une duplication en tandem lorsqu'une région est répétée une ou plusieurs fois, éventuellement de façon approximative et à courte distance dans un génome.*

1.2.3 Contribution des duplications à l'évolution des génomes

L'un des enjeux majeur de la génétique est de comprendre la relation qui lie le génome au phénotype. À ce jour, plusieurs auteurs ont souligné le paradoxe apparent de la faible différence nucléotidique entre le génome de l'homme et celui du chimpanzé (Chimpanzee Sequencing Consortium, 2005) ($\simeq$ 1.2%) par rapport à leur différence phénotypique. Même si les modifications de l'expression des gènes contribuent très certainement à cette différence phénotypique, des études récentes (Norris et Whan, 2008; Marques-Bonet *et al.*, 2009) suggèrent que le mécanisme de duplication joue également un rôle important.

L'étude récente des génomes entièrement séquencés a mis en évidence une dynamique jusqu'alors sous-estimée de duplications des segments chromosomiques et plus particulièrement des duplications de gènes. Demuth *et al.* (Demuth *et al.*, 2006) ont montré que les catalogues de gènes du chimpanzé et de l'homme différaient d'au moins 6% (voir Tab. 1.2 page suivante).

Différentes études ont montré que le processus de duplication, qui participe au phénomène de naissance et de mort de gènes (renouvellement perpétuel de gènes), était présent chez tous les organismes à des degrés variables (le taux de duplication de gènes varie entre 0.01 et 0.002 (Lynch et Conery, 2000a) par gène et par million d'années). Intuitivement on imagine que ce taux est très faible par rapport aux taux de mutations des nucléotides. Hors, dans le chapitre 8 de *The origins of genome architecture* (Lynch, 2007),

	Polymorphisme	Divergence
Nucléotides	0.0009	0.0123
Nombre de copies	0.0038	0.064

TABLE 1.2: Polymorphisme et divergence au niveau nucléotidique et du nombre de copie de gène. Le polymorphisme correspond aux variations de la séquence d'ADN au sein d'une espèce, ici l'homme. La divergence correspond aux variations de la séquence d'ADN entre deux espèces, ici entre l'homme et le chimpanzé. Ces variations sont exprimées en base au niveau nucléotidique et en gènes au niveau des copies. Tableau issu de l'article de Schrider et Hahn (Schrider et Hahn, 2010).

Lynch montre qu'en moyenne le taux de duplications par gène est égal à 40% du taux de mutations par nucléotide. Ainsi le taux de duplication est seulement deux fois plus lent. Ce qui amène Lynch à suggérer que la relation entre les deux taux participe à l'équilibre entre le taux de naissance et de mort des gènes. (Friedman et L., 2004) soulignent par ailleurs que la majorité des duplications de gènes sont en tandem.

1.3 La détection ab initio des duplications en tandem

L'exploitation des données issues des grands projets de séquençage des génomes représente un enjeu majeur de la biologie moderne. Actuellement, l'étude comparative des génomes est fortement dépendante de l'annotation des génomes. L'objectif du travail présenté ici consiste à développer une nouvelle méthode pour détecter les duplications, en particulier en tandem, à partir de séquences d'ADN et ce malgré la poursuite des mécanismes d'évolution qui modifient le contenu de ces régions. Nous voulons aussi montrer que les régions détectées comme dupliquées en tandem sont riches en éléments fonctionnels comme les gènes.

La détection *ab initio* des duplications en tandem au sein d'un génome est un cas particulier de la recherche de régions homologues entre deux espèces. En conséquence, nous allons commencer par décrire et étudier ce problème dans le chapitre 2 à travers un état de l'art du domaine. A cette occasion, un nouveau formalisme sera développé permettant de réunir une grande majorité des méthodes existantes. Les outils existants, les méthodes utilisées et les performances correspondantes y sont présentés, ainsi que les principales limitations subsistantes.

C'est de cette analyse que découle l'approche entreprise dans cette thèse, qui consiste à développer une méthode plus générale pour reconstruire les régions dupliquées en tandem, présentée dans le chapitre 3. Nous commencerons par décrire le formalisme et la stratégie développée pour le résoudre. Notamment, nous verrons les contraintes de *cohérence globale* que les régions détectées doivent respecter pour représenter une histoire évolutive cohérente (voir section 3.1.2.2 page 54). Puis nous verrons comment le problème est résolu à l'aide de la théorie des flots.

Le chapitre 4 page 83 est consacré à la présentation et à l'analyse des résultats obtenus grâce à l'implémentation de cette nouvelle méthode dans le logiciel ReD (**Re**gion **D**upliquées). En particulier, nous analyserons les résultats de la détection *ab initio* des duplications en tandem au niveau nucléique obtenus sur le génome d'*Arabidopsis Thaliana*.

Chapitre 2

État de l'art de la recherche de régions dupliquées

Sommaire

La recherche d'une relation de parenté, c'est-à-dire d'une origine ancestrale commune, ou relation d'homologie, entre séquences (nucléiques ou protéiques) d'espèces différentes, ou d'une même espèce, est centrale à l'étude de l'évolution des gènes, des génomes et des espèces. Parmi ces relations d'homologie, on distingue généralement les relations d'orthologies, lorsque l'homologie entre deux séquences résulte d'un événement de spéciation, et la paralogie lorsqu'elle résulte d'une duplication. Ainsi notre travail est un cas particulier de la recherche d'homologie.

Une grande similarité entre deux séquences est un signal fort permettant de suggérer une relation de parenté. C'est donc tout naturellement que la comparaison de séquences est devenue synonyme de recherche d'homologie et que l'algorithmique du texte a fait son apparition en biologie. Les premières approches, connues sous le nom d'alignement de séquences, permettaient la comparaison de séquences courtes. Pour des séquences plus longues, comme les séquences génomiques, la poursuite des mécanismes d'évolution détruisent le signal de similarité de façon locale. L'objectif est alors de reconstruire de grandes régions globalement conservées, c'est à dire des régions composées d'une succession de séquences similaires et non similaires.

Nous décrivons dans ce chapitre le principe des méthodes de recherche d'homologie et plus précisément celles qui tentent d'identifier des relations d'homologie entre régions chromosomiques. Nous décrirons également les méthodes développées pour la détection des duplications en tandem.

2.1 Recherche de régions homologues

L'évolution des chromosomes est un processus qui se déroule à différentes échelles. Aux modifications locales n'impliquant que quelques nucléotides, s'ajoutent des ré-arrangements microscopiques locaux pouvant impliquer de quelques milliers à plusieurs dizaines de milliers de nucléotides et des ré-arrangements de plus grande ampleur portant sur plusieurs millions de nucléotides voire des bras chromosomiques entiers. Ces deux types de ré-arrangements, microscopiques et macroscopiques, n'ont pas la même dynamique, les seconds étant bien moins fréquents que les premiers (Lynch, 2007) (un génome de mammifère subit quelques dizaines de mutations par an et à peine un ré arrangement de grande ampleur par million d'années).

Pour illustrer l'évolution des génomes, on peut utiliser l'analogie d'un long texte organisé en chapitres. Les modifications microscopiques locales fréquentes porteraient sur l'orthographe des mots, leur substitution ou un ré arrangement local du texte, alors que les ré-arrangements macroscopiques déplaceraient de grandes sections de texte au sein d'un chapitre ou même entre chapitres. Notre problème consiste, à partir de l'observation de deux livres ayant évolué indépendamment, à identifier les régions de texte globalement similaires. La difficulté réside dans le fait que les modifications brouillent le signal évolutif de conservation.

L'évolution des génomes se déroule à différentes échelles, il est logique que la recherche de séquences homologues se déroule aussi à différentes échelles, schématisées dans la figure 2.1 :

- au niveau local pour trouver les **(sous-)séquences conservées** (comme les gènes) ;
- au niveau global pour trouver les **régions conservées**.

Au niveau local, l'objectif consiste à vérifier si les séquences se ressemblent ou non, à l'aide d'une **mesure de similarité**. Cette similarité est calculée à partir de la mise en correspondance des nucléotides entre deux séquences, appelée un alignement. Une fois cet alignement construit, il est possible de compter le nombre de nucléotides identiques. Sachant qu'il existe plusieurs alignements possibles, l'objectif est de construire l'alignement de similarité maximum afin d'identifier les (sous-)séquences conservées.

Au niveau global, l'objectif consiste, à partir des couples de sous-séquences conservées, à vérifier si un ensemble de ces couples partagent une **organisation similaire** sur les deux séquences, ce qui permet de ne pas être entravé par la perte locale de similarité.

Une organisation similaire est détectée à l'aide des relations qui lient deux couples de séquences conservées, souvent modélisées sous la forme d'un graphe. Ces relations peuvent être de différentes natures, comme la distance qui sépare deux couples, l'ordre sur les deux séquences ou encore l'orientation ; et l'objectif est de prolonger les alignements des sous-séquences en minimisant les changements d'ordre et en satisfaisant les contraintes de distance

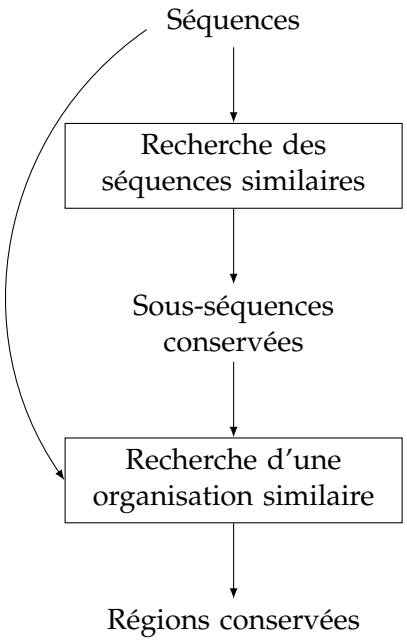


FIGURE 2.1: Schéma des principales étapes du processus de recherche d'homologies.

Nous allons, dans un premier temps, présenter brièvement les différentes méthodes qui mettent en évidence une relation de conservation par une forte similarité entre deux séquences. Puis, dans un second temps, nous étudierons les méthodes qui détectent la conservation des séquences en exploitant une éventuelle organisation similaire entre deux séquences. Enfin, nous proposerons une analyse générale de l'existant pour mettre en évidence les points que nous souhaitons améliorer.

2.2 Recherche d'homologie de séquences

2.2.1 Le dotplot

Le dotplot est l'une des plus anciennes méthodes utilisées en comparaison de séquences. Chaque axe d'un tableau rectangulaire représente une des séquences à comparer. Deux fenêtres de taille fixe parcourent les deux séquences. A chaque fois qu'un critère de similitude est satisfait (fenêtres identiques, pourcentage d'identité des séquences supérieur à un seuil fixé) un point est placé à la position correspondante du tableau rectangulaire (voir figure 2.2).

	G	A	T	T	A	C	A
G	*						
T			*	*			
T			*	*			
T			*	*			
A		*			*		*
G	*						
A		*			*		*

	G	A	T	T	A	C	A
G	*						
T		*					
T			*				
T				*			
A					*		
G						*	
A							*

FIGURE 2.2: Comparaison de deux séquences avec deux dotplots différents. A gauche, la taille des fenêtres est de 1 nucléotide et elles doivent être identiques pour satisfaire le critère de similarité. A droite, la taille est de 4 nucléotides et 75% d'identité est suffisant pour satisfaire le critère.

L'inconvénient du dotplot tient à sa nature qualitative. Nous allons voir dans la section suivante comment définir une mesure quantitative de la similarité entre deux séquences, à l'aide d'un alignement.

Le dotplot est aussi utilisé comme outil de visualisation graphique d'un ensemble d'alignements locaux entre deux séquences (voir figure 2.6 page 29).

2.2.2 L'alignement

Un alignement peut s'interpréter comme la représentation d'un travail d'édition consistant à appliquer le minimum d'opérations élémentaires d'édition permettant de trans-

former une séquence en une autre. Les deux opérations d'édition autorisées sont :

– la substitution : remplacement d'une lettre par une autre ;
– l'indel : insertion ou délétion d'une lettre.

Ces deux opérations d'édition peuvent s'interpréter comme des mutations nucléotidiques (section 1.2.2.1 page 10) et le travail d'édition comme une tentative de reconstruction de l'histoire évolutive en considérant ces trois types de mutations élémentaires (voir figure 2.3).

```
G A T T A C A
| * | | |   |
G T T T A – A
```

FIGURE 2.3: Alignement de deux séquences. Cet alignement contient : 5 positions identiques, marquées par des barres verticales ; 1 position de substitution, marquée par une étoile ; et 1 position d'indel.

2.2.2.1 Mesure de la qualité d'un alignement

La qualité d'un alignement peut se mesurer de différentes manières. On peut considérer une distance d'édition qui serait une mesure de la distance évolutive ou à l'inverse une mesure de similarité. Nous allons détailler ces deux types de mesure.

1. Distance d'édition : la distance d'édition correspond précisément au nombre minimum d'opérations d'édition nécessaire pour passer d'une séquence à l'autre. La distance entre deux séquences prend la forme suivante :

$$d(\mathsf{A},\mathsf{B}) = nbSub * P_s + nbIndel * P_i \tag{2.1}$$

   avec A et B les deux séquences, $nbSub$ le nombre de substitutions et $nbIndel$ le nombre d'indels. P_s et P_i permettent de pénaliser différemment les substitutions des indels.

2. Mesure de similarité ou score :

   Le score de similarité prend en compte la similarité des deux séquences alignées, en plus des différences. Ce score peut être calculé en fonction de la distance d'édition :

$$S(\mathsf{A},\mathsf{B}) = nbSim * R_s - d(\mathsf{A},\mathsf{B}) \tag{2.2}$$

   avec $nbSim$ le nombre de nucléotides ou d'acides aminés identiques et R_s la récompense attribuée à chacun.

   La distance d'édition et le score de similarité peuvent aussi être calculés par des formules plus respectueuses des réalités biologiques. Notamment, la suppression

de i nucléotides consécutifs s'explique par un seul évènement mutationnel (appelé *ouverture d'indel*) et doit être moins pénalisée que la suppression de i nucléotides éparpillés (qui demande i ouvertures d'indel). Pour un score de similarité cela donne par exemple cette formule :

$$S(\mathsf{A},\mathsf{B}) = nbSim * R_s - (nbSub * P_s + nbOuverture * P_o + nbExtension * P_e) \quad (2.3)$$

avec $nbOuverture$ le nombre d'ouvertures d'indel, $nbExtension$ le nombre de nucléotides supprimés ou ajoutés et P_o, P_e les pénalités associées.

2.2.2.2 L'alignement global

La construction d'un alignement global consiste à identifier le meilleur alignement possible entre deux séquences, celui qui minimise la distance d'édition $d(\mathsf{A},\mathsf{B})$ ou qui maximise le score $S(\mathsf{A},\mathsf{B})$. En pratique, le principe de score de similarité est le plus largement utilisé.

Ce problème bien connu est résolu par un algorithme de programmation dynamique avec une complexité temporelle en $O(nm)$, où n et m sont les longueurs des séquences A et B. Cet algorithme a été développé par Needleman et Wunsch (Needleman et Wunsch, 1970) en 1970.

2.2.2.3 L'alignement local

Dans la majorité des cas, les deux séquences à comparer n'ont pas une forte similarité sur toute leur longueur, mais elles possèdent plusieurs paires de sous-séquences très similaires. Un alignement global qui tente d'aligner, à la fois, les régions similaires et non similaires n'est pas très informatif. Il est préférable d'extraire et d'aligner seulement les paires de sous-séquences similaires. C'est ce qui est réalisé par l'alignement local.

Définition 6 (Alignement local) *Un alignement local est un alignement dont le score de similarité est localement maximum.*

Ce problème bien connu est résolu par de nombreux algorithmes de programmation dynamique, le premier étant celui de Waterman et Smith (1981) avec une complexité en $O(nm)$. En pratique cet algorithme est trop lent et sa complexité est trop élevée pour traiter un grand nombre de comparaisons. On a alors recours à des heuristiques [1], comme BLAST (Altschul *et al.*, 1990) (Basic Local Alignment Search Tool).

1. On appelle heuristique un algorithme qui ne garantit pas de trouver la solution optimale au problème posé, mais qui en général trouve des solutions assez proches.

Ce type d'alignement est souvent appliqué à l'ADN (séquences nucléiques), qui est composé à la fois de régions codantes et de régions non codantes. Les régions non codantes étant moins conservées au cours du temps (voir remarque 4 page 11), les alignements locaux alignent majoritairement des régions codantes. C'est notamment le cas des exons et des introns, quand on compare les séquences nucléiques de gènes.

Par la suite, ces alignements sont utilisés comme *ancres* par les méthodes de reconstruction de grandes régions dupliquées (voir définition 8 page 29).

2.2.2.4 Interprétation statistique du score de similarité

Les logiciels d'alignements tels que Blast (Altschul *et al.*, 1990) renvoient un grand nombre d'alignements locaux et il est nécessaire de faire le tri entre les alignements pertinents et les autres. Intuitivement, on comprend qu'en alignant deux séquences nous obtenons des alignements locaux de scores variables et que la probabilité de trouver un alignement parmi tous les autres dépend de son score et de la taille des séquences alignées.

Ainsi les logiciels réalisent un test statistique et calculent une *e-value* pour chaque alignement qui permettra de quantifier la probabilité que le score d'un alignement soit le fruit du hasard.

Définition 7 (e-value) *La* e-value *d'un alignement de score S est le nombre attendu d'alignements de score supérieur ou égal à S, sous l'hypothèse que les scores observés sont obtenus à partir de séquences aléatoires. En deux mots : c'est le nombre d'alignements attendus avec au moins ce score, du seul fait du hasard.*

2.2.3 L'alignement génomique

Avant l'arrivée des séquences complètes de génomes, la génomique comparative s'appuyait essentiellement sur la comparaison des gènes et plus précisément la comparaison de séquences protéiques. Avec l'arrivée des séquences complètes, la génomique comparative a pu s'étendre aux régions non-codantes. Plus généralement, la démarche de recherche d'homologie s'applique aux longues séquences que sont les séquences génomiques. L'objectif reste ici de pouvoir identifier les positions homologues entre deux séquences mais est bien plus ambitieuse dans le cadre de séquences génomiques qui sont par nature bien plus longues et plus variables que les séquences protéiques (Dewey et Pachter, 2006).

Afin d'aborder ce problème de recherche d'homologie au niveau nucléique sur de longues séquences génomiques de nouvelles méthodes de comparaison nucléique ont été développées, notamment celles qui essayent d'aligner deux génomes sur l'ensemble de leurs séquences, comme MUMmer (Delcher *et al.*, 1999), GLASS (Batzoglou *et al.*, 2000; Morgenstern, 2002) et Shuffle LAGAN (Brudno *et al.*, 2003). Ces méthodes d'alignement

de génomes deux à deux ont été généralisées à l'alignement simultané de plusieurs génomes, appelé alignement multiple (Myers et Miller, 1995), à l'aide des méthodes MGA (Hohl *et al.*, 2002), Mauve (Darling *et al.*, 2004) ou (Abouelhoda et Ohlebusch, 2005) par exemple.

L'ensemble de ces méthodes partage une stratégie en 5 étapes :

1. **Filtre :** Cette étape consiste à filtrer les séquences nucléiques pour supprimer les régions ayant peu d'intérêt, comme les régions de faible complexité ou les séquences répétées dans le génome qui sont susceptibles de perturber les étapes ultérieures du processus d'alignement.
2. **Graine :** Cette étape consiste à créer des alignements locaux de très bonne qualité, avec très peu de substitutions et d'indels. Ces alignements sont souvent appelés des *graines*, car certains de ces alignements serviront de base à la construction de l'alignement génomique.
   Du fait que certains de ces alignements serviront d'ancres, cette étape est très importante et les choix faits pour créer ces alignements influencent directement le résultat final.
3. **Chaînage :** Cette étape consiste à filtrer les graines pour sélectionner celles qui serviront à la construction de l'alignement génomique. Les graines choisies sont appelées des *ancres*.
   La sélection de ces ancres dépend des méthodes et des critères que ces méthodes souhaitent satisfaire. L'un de ces critères, peut exiger la sélection d'ancres colinéaires [2] ou non colinéraire (Brudno *et al.*, 2003).
   En général, la sélection des graines est réalisé par une méthode de *chaînage* comme par exemple celle consistant à identifier la chaîne de poids maximum dans le graphe des ancres (vpor ci-dessous), méthode connue sous le nom de maximum weighted chain (MWC) (Hohl *et al.*, 2002) (voir remarque 18 page 40).
4. **Récursivité :** Cette étape consiste à re-exécuter les étapes 2 et 3 (Graine et Ancre) un certain nombre de fois sur les régions des génomes qui ne sont pas utilisées dans des ancres. À chaque nouvelle exécution de ces étapes, on relâche les critères de qualité des alignements locaux, dans l'objectif de relier les extrémités de chaque ancre.
5. **L'alignement :** Cette étape consiste à relier les ancres, parfois à l'aide de grands indels, pour créer un alignement unique des deux séquences comparées.

Il est intéressant de remarquer que ces méthodes répondent à une problématique proche de notre objectif, elles cherchent à identifier des relations d'homologies au niveau nucléotidique entre régions chromosomiques, sans information d'annotation. Cependant, il existe une différence importante, elles ne cherchent à reconstruire qu'un unique alignement pour chaque région du génome, là ou notre problématique nécessite d'en construire plusieurs pour retrouver de multiples duplications d'une région du génome,

2. des ancres qui se suivent dans le même ordre sur les deux génomes

en particulier en tandem. En effet à chaque paire d'une duplication (en tandem) correspond deux régions conservées qu'il faut reconstruire, soit un alignement pour chaque paire.

Ainsi pour reconstruire les régions conservées issues de duplications en tandem, il est préférable de s'inspirer des méthodes qui détectent les régions homologues et les duplications segmentales. Ces méthodes ont la caractéristique de pouvoir créer plusieurs "alignements", appelés chaînes, à partir d'une région génomique, ce qui correspond précisément à nos besoins.

Remarque 6 *Naturellement, les méthodes d'alignement génomique et les méthodes de reconstruction de régions dupliquées ont un socle de notions et de problèmes communs non négligeable. Elles partagent en particulier les notions d'ancres, de chaînage et des problèmes de chevauchements qui seront expliqués et définis en détail dans les parties suivantes.*

Avant de décrire en détails les méthodes de reconstruction de régions homologues et de duplications segmentales (voir section 2.4 page suivante), qui serviront de base au développement de notre nouvelle méthode, nous mentionnons les méthodes existantes de détection de duplication en tandem.

2.3 Duplication en Tandem

Nous avons vu dans la section 1.2.3 page 12, que les duplications ont joué un rôle important dans l'évolution des génomes. À ce titre, l'étude des régions dupliquées présente un intérêt certain et de nombreux logiciels ont été développés pour les détecter.

Ces logiciels peuvent être classés en deux grandes catégories :

- La première détecte ces duplications à l'intérieur de séquences protéiques (Lynch et Conery, 2000b; Enright *et al.*, 2002; Li *et al.*, 2003; Shoja et Zhang, 2006; Jorda et Kajava, 2009). Ces méthodes dédiées à l'analyse des séquences courtes que sont les protéines ne sont bien entendu pas adaptées à la détection de duplications au sein des génomes.
- Le second a pour objectif de détecter de courtes régions nucléiques répétées en tandem. Ces répétitions sont la plupart du temps d'une taille inférieure à 500 pb, très répétées et très bien conservées et détectées par des logiciels tels que TRF (Benson, 1999), mReps (Kolpakov *et al.*, 2003), STAR (Delgrange et Rivals, 2004) ou la méthode de (Bailey *et al.*, 2001). Les dernières versions de TRF et mReps et la méthode de (Bailey *et al.*, 2001) sont capables d'aller jusqu'à des tailles de régions de 2kb, mais elles sont limitées à la détection de duplications très récentes, sans perte locale de similarité. Ces méthodes ne dépendent pas d'une éventuelle annotation mais sont incapables de détecter des régions dupliquées avec perte de similarité locale.

Je m'arrêterai ici pour la présentation des méthodes existantes. En effet, notre problématique est bien différente des problématiques abordées par ces méthodes. Notre objectif est de détecter des séquences nucléiques dupliquées de taille supérieure à 500 pb, sans s'appuyer sur une annotation et ce malgré une éventuelle perte locale de similarité. Ainsi, il sera possible de détecter des duplications plus anciennes contenant potentiellement des gènes mais aussi d'autres éléments fonctionnels.

Finalement, il s'agit de reconstruire de grandes régions dupliquées et à ce titre la stratégie utilisée est plus proche des méthodes de chaînage (voir section 2.4), même si les régions dupliquées en tandem sont en général plus courtes que celles issues de duplications segmentales.

Remarque 7 *Une méthode récente (Despons et al., 2010) vise à détecter les duplications en tandem à l'aide d'analyses basées sur l'utilisation de gènes codants pour des protéines*[3] *annotés mais avec un objectif qui se rapproche du nôtre. Elle tente, à l'aide des gènes de protéines, de détecter les gènes dupliqués en tandem mais aussi les pseudo-gènes issus de ces duplications. Cette recherche s'appuie sur une seconde analyse de similarité protéine/ADN, via l'utilisation locale de BLASTX. Cette méthode reste dépendante de l'annotation des génomes et ne détecte que les gènes codants des protéines et d'éventuels pseudo-gènes adjacents. Elle a été développée dans le cadre de l'analyse de génomes procaryotes ou eucaryotes unicellulaires (levures) et reste limitée, du fait de son principe même, à l'analyse de génomes compacts.*

2.4 Reconstruction de régions homologues et de duplications segmentales

Peu après la séparation de deux espèces ou d'une duplication, la conservation des génomes est telle qu'il est facile de les comparer (aligner) et d'identifier les longues séquences chromosomiques conservées. Lorsque la spéciation ou la duplication est plus ancienne, l'identification des régions dupliquées est rendue difficile par la poursuite des différents mécanismes évolutifs (mutations ponctuelles, ré-arrangements...). Ces modifications peuvent mener à la disparition locale de la similarité, à des changements d'orientation ou d'ordre entre éléments inclus dans la région (gènes, régulateurs...). Les seuls indices aisément identifiables permettant de retrouver ces régions homologues sont de courtes régions suffisamment similaires pour pouvoir être alignées (aussi appelées *ancres*), traces potentielles de l'homologie entre les deux régions. La densité de ces ancres dans une région, la succession d'une quantité importante d'ancres dans un ordre et/ou une orientation identique sont autant d'éléments attestant d'une organisation similaire permettant de suggérer l'existence d'une région homologue.

La figure 2.4 page 26 est un bon exemple pour comprendre les principales difficultés et les quelques notions essentielles. Les données du problème sont représentées sous

3. dit gènes de protéines par la suite

forme d'un dotplot. Il s'agit d'ancres obtenues en comparant au niveau protéique tous les gènes du chromosome 2 avec tous les gènes du chromosome 4 d'*Arabidopsis thaliana*.

Il faut savoir que la notion de régions dupliquées est assez informelle, et les mécanismes évolutifs sous-jacents mal évalués. C'est pourquoi une grande variété de critères et d'outils ont été proposés dans la littérature pour identifier de telles régions. Ainsi pour certaines méthodes une région homologue est une succession d'ancres dans le même ordre sur les deux génomes. Une autre définition tolèrera les changements d'ordre. Ainsi, il est difficile de comparer les méthodes sur leurs résultats, il est préférable de comparer leurs stratégies.

Ce problème de définition est illustré sur la figure par certaines régions mises en valeur comme des régions potentiellement homologues. Certaines ont une forte concentration d'ancres linéaires, tandis que d'autres demandent une analyse plus approfondie pour être discernées.

De plus, cet exemple met en évidence que la plupart des ancres ne participent pas à la reconstruction de régions homologues, à tel point que l'expression *"chercher une aiguille dans une botte de foin"* s'applique parfaitement à notre situation. Cette expression illustre la complexité du problème à résoudre et nous verrons que les méthodes actuelles ont choisi de le simplifier en modélisant qu'une partie des informations fournies par les ancres.

Dans les sections qui suivent, je propose de classer les méthodes en trois catégories, en fonction de leur stratégie de reconstruction :

1. La première utilise un graphe pour modéliser l'ensemble des solutions et sélectionne la meilleure, une solution étant un ensemble de chemins.
2. La deuxième utilise des tests statistiques pour vérifier si la succession d'un ensemble d'ancres est le résultat du hasard ou d'une région dupliquée.
3. La troisième ne crée pas de régions dupliquées, elle fait une sous-sélection d'ancres qui visuellement permet d'identifier les régions dupliquées.

Les deux dernières catégories sont marginales, mais elles possèdent quelques propriétés intéressantes que nous étudierons à la fin de cette section. Ainsi, je vais particulièrement m'intéresser à la première catégorie que je décris dans un formalisme commun.

2.4.1 Les méthodes de chaînage

Dans cette section je décris et compare sept méthodes : ADHoRe (Vandepoele *et al.*, 2002), GRIMM-Synteny (Pevzner et Tesler, 2003), FISH (Calabrese *et al.*, 2003), DiagHunter (Cannon *et al.*, 2003), DAGchainer (Haas *et al.*, 2004), SyMAP (Soderlund *et al.*, 2006), OSfinder (Hachiya *et al.*, 2009) ; que je réunis sous un seul formalisme schématisé dans la figure 2.5 page 27 et décomposé en deux étapes :

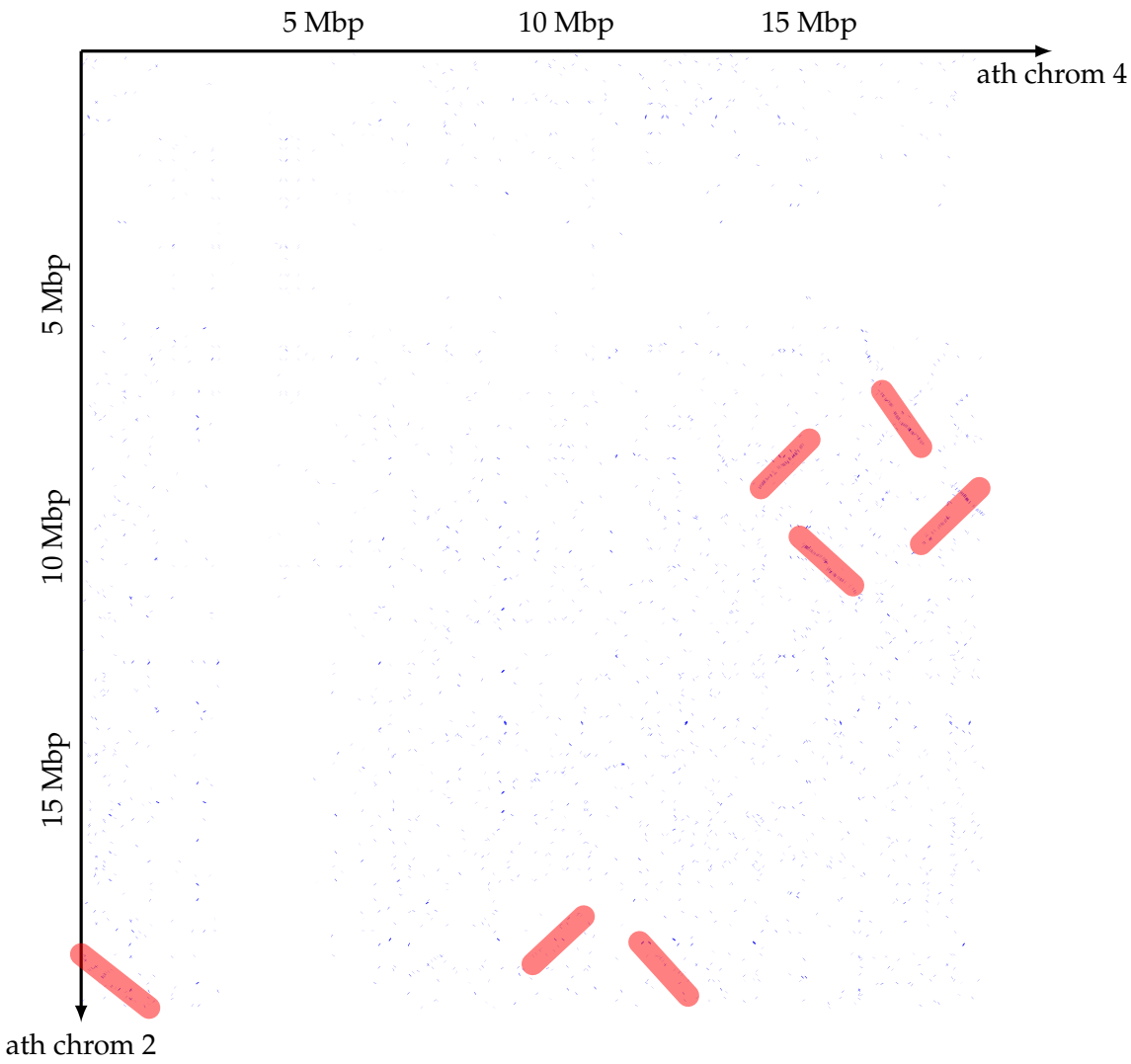


FIGURE 2.4: Un exemple réel de recherche de régions homologues au niveau protéique entre deux chromosomes d'*Arabidopsis thaliana*. Chaque trait bleu est une ancre et chaque trait rouge est une région potentiellement homologue.

- **Compatibilité des ancres :** A partir des *ancres* et des *séquences* fournies comme données, cette étape commence par pré-sélectionner tous les couples d'ancres qui peuvent appartenir à deux régions dupliquées. Ces couples sont établis notamment à l'aide de leurs positions sur les deux génomes, et ils sont modélisés sous la forme d'un graphe, appelé *graphe de relation*, où les ancres sont représentées par des sommets et les couples par des arcs. Un chemin dans ce graphe sélectionne une succession d'ancres et reconstruit deux régions dupliquées.
- **Chaînages des ancres :** A partir du *graphe de relation*, cette seconde étape a pour objectif d'étendre les alignements des ancres à l'aide des chemins, appelé *chaînes*, afin de reconstruire les régions homologues.

Remarque 8 *Par la suite, les* graphes de relation *sont visualisés à l'aide d'un dotplot, ce qui permet de représenter une chaîne comme une association des alignements qui la compose sous la forme d'une longue diagonale.*

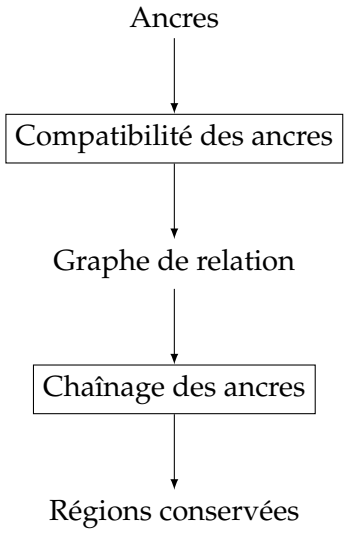


FIGURE 2.5: Schéma du formalisme commun à toutes les méthodes de chaînages.

Dans un premier temps, nous étudierons le formalisme qui réunit toutes ces méthodes. Puis, nous passerons à la phase d'*optimisation* de création des chaînes. C'est ici que les méthodes se différencient par les critères qu'elles optimisent et/ou par leurs stratégies.

2.4.1.1 Représentation des génomes : les séquences

Dans un souci de simplification, je considère que les génomes étudiés sont constitués chacun d'un seul chromosome (soit une seule séquence), les méthodes se généralisant facilement au cas multi-chromosomiques (voir remarque 13 page 32).

Les génomes A et B sont donc modélisés par deux séquences[4], de tailles respectives n et m. Un segment d'un génome est entièrement déterminé par la donnée de l'index de la première lettre et de celui de la dernière lettre, il peut donc être noté par un intervalle $I = [deb, fin]$, avec $1 \leq deb < fin$ ($fin \leq n$ pour un intervalle de A et $fin \leq m$ pour un intervalle de B). On munit ces intervalles d'une relation d'ordre, on note $I < I'$ le fait que l'intervalle I' soit situé après l'intervalle I (voir ci-dessous). La distance entre 2 intervalles, définie comme la distance séparant ses éléments les plus proches, sera notée $dist(I, I')$.

Il est important de noter qu'en pratique, les frontières des intervalles manipulés (deb, fin) sont rarement exactes et il arrive que deux intervalles se chevauchent. Une relation d'ordre strict[5] est alors pénalisante et il vaut mieux relâcher la contrainte d'ordre. Ainsi, on note $I < I'$ le fait que l'intervalle I' soit situé après l'intervalle I à condition qu'ils se chevauchent sur moins de p%.

Les logiciels utilisent deux types de séquences pour représenter les génomes :

1. **La séquence nucléique** : une séquence de nucléotides, chaque position correspond à un nucléotide.
2. **La séquence génique** : une séquence de gènes, chaque position correspond à un gène issu d'une annotation du génome (l'intergénique est supprimé, chaque gène est réduit à un point).

Sachant que les gènes sont les principales régions d'intérêt de l'ADN, représenter le génome par une succession de gènes est une heuristique astucieuse et logique pour limiter la quantité d'information à traiter. Elle n'est pourtant pas exempt de défauts. Tout d'abord, la distance (en nucléotides) inter-gènes n'est plus accessible, c'est pourtant une source d'information significative pour la détection de régions homologues.

Remarque 9 *Pour donner une idée de la quantité de données supprimées, chez l'homme la totalité des régions géniques correspond à moins de* 3% *de l'ADN.*

Ensuite, la séquence génique est directement construite à partir de l'annotation. Ainsi la qualité de l'annotation aura une influence directe sur les résultats et sans annotation il est impossible d'utiliser cette solution. Il faut aussi ajouter qu'il est impossible d'utiliser les similarités de séquences non annotées et donc d'étudier les similarités des régions fonctionnelles non-annotées.

2.4.1.2 Les ancres

Une ancre a définit une correspondance entre un segment de A et un segment de B. On note a^{A} et a^{B} les intervalles correspondants. Une ancre est susceptible de relier des éléments d'orientations opposées, on dira alors qu'elle est de polarité négative, sinon

4. les séquences du brin forward uniquement
5. n'acceptant pas les chevauchement

elle est de polarité positive. Enfin, la qualité de la similarité observée entre les deux régions est capturée dans un score ou une e-value (définition 7 page 21) (fourni par les outils d'alignement locaux).

Définition 8 (Une ancre) *Une ancre a_i est déterminée par la donnée de quatre composantes :*

- *un intervalle a_i^A sur le génome A*
- *un intervalle a_i^B sur le génome B*
- *une polarité notée $a_i.pol$*
- *un score noté $a_i.s$ et/ou une e-value noté $a_i.e$*

Remarque 10 *Pour simplifier les formules, on adopte la convention suivante, $a_i^A.deb < a_i^A.fin$ et $a_i^B.deb < a_i^B.fin$, même si la polarité de l'ancre est négative.*

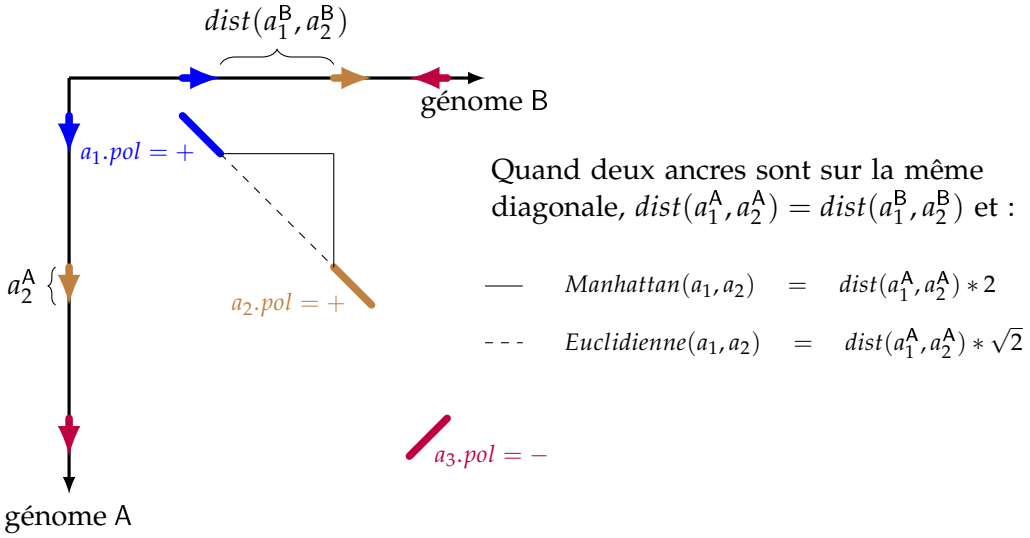


FIGURE 2.6: Dotplot qui visualise les ancres et leurs propriétés ainsi que les distances qui les séparent.

Il existe deux types d'ancres :

– **Les ancres protéiques :** Ces ancres ne concernent que des régions qui codent des protéines. Elles sont obtenues à l'aide de logiciels d'alignements protéiques tel que BLASTP (Altschul *et al.*, 1990).
– **Les ancres nucléiques :** Ces ancres concernent toutes les régions de l'ADN (géniques et intergénique). Elles sont obtenues à l'aide de logiciels d'alignements locaux nucléiques comme YASS (Noé et Kucherov, 2005) ou BLAST (Altschul *et al.*, 1990).

Remarque 11 *Les méthodes qui utilisent des ancres protéiques peuvent uniquement détecter des duplications contenant des protéines.*

2.4.1.3 L'analyse des ancres

Dans cette section, nous détaillons la première étape des algorithmes de chaînage. Comme souvent quand le problème à résoudre est compliqué, il vaut mieux commencer par des sous-problèmes faciles à résoudre. C'est la stratégie adoptée par les méthodes de chaînage, qui commencent par vérifier s'il est raisonnable de supposer que deux ancres font partie d'une même région ancestrale.

Définition 9 (Deux ancres chaînables) *Nous dirons que deux ancres a_i, $a_j \in \mathcal{A}$, sont compatibles ou* chaînables, *noté $a_i \prec a_j$, si les relations d'homologies qu'elles définissent chacune entre A et B sont compatibles avec une seule relation d'homologie entre A et B. En d'autres termes, si les intervalles associés sur A et B - a_i^A, a_j^A d'une part et a_i^B, a_j^B d'autre part - peuvent être considérés comme appartenant à une même région ancestrale.*

La définition de deux *ancres chaînables* dépend des mutations qui sont acceptées au sein de deux régions homologues. Sachant que plus on accepte de mutations plus le problème devient complexe à résoudre.

Détection des régions colinéaires : commençons par la définition la plus simple, celle qui ne modélise pas les mutations qui changent l'ordre et l'orientation des ancres. Pour respecter cette définition il faut que les deux ancres soient de même polarité. L'existence de deux ancres de polarité différente indique que le sens de lecture a changé quelque part entre les deux ancres, soit un changement d'orientation. En ce qui concerne l'ordre, il faut vérifier si la succession des deux ancres est correcte. La succession dépend de la polarité des ancres :

- Avec une polarité positive les ancres doivent se suivre dans le même ordre sur les deux génomes ;
- Avec une polarité négative les ancres doivent se suivre dans un ordre inversé sur les deux génomes.

Définition 10 ($\prec_\varnothing$) *On dit que $a_i \prec_\varnothing a_j$ lorsque $a_i.pol = a_j.pol$ et :*

- $a_i^A < a_j^A$ *et* $a_i^B < a_j^B$ *si* $a_i.pol = +$
- $a_i^A < a_j^A$ *et* $a_i^B > a_j^B$ *si* $a_i.pol = -$

Définition 11 (Ancres colinéaires) *J'appelle* ancres colinéaires, *une succession d'ancres reliées par une relation $\prec_\varnothing$.*

Définition 12 (Régions colinéaires) *J'appelle* régions colinéaires, *deux régions composées d'ancres colinéaires.*

Cette définition stricte de la composition de deux régions homologues a l'avantage de diminuer le nombre d'*ancres chaînables* n'appartenant pas à des régions homologues. En diminuant ce nombre on diminue les chances de se tromper dans l'étape d'*optimisation*.

Nous avons vu que la quantité d'information à traiter pouvait être considérable, diminuer le champ des possibilités est un avantage non négligeable. Cependant les chances de supprimer des *ancres chaînables* issues de régions homologues sont non nulles, ce qui peut empêcher de les délimiter correctement ou encore de les détecter (figure 2.7 page 33).

Changement d'orientation : pour limiter les défauts de $\prec_{\varnothing}$, une première amélioration consiste à accepter les changements d'orientations des ancres, ce qui *a priori* augmente la complexité du problème. Pour revenir à un problème plus simple DAGchainer (Haas *et al.*, 2004) décide de le résoudre en deux étapes indépendantes :

1. La première s'occupe des *ancres chaînables*, uniquement pour la reconstruction de régions dupliquées de polarité positive exclusivement (figure 2.8 page 34), soit des ancres qui sont dans le même ordre sur les deux génomes :
   Définition 13 ($\prec_{1+}$) *On dit que* $a_i \prec_{1+} a_j$ *lorsque* $a_i^A < a_j^A$ *et* $a_i^B < a_j^B$
2. La deuxième s'occupe des *ancres chaînables*, uniquement pour la reconstruction de régions dupliquées de polarité négative exclusivement (figure 2.8 page 34), soit des ancres qui sont dans un ordre opposé sur les deux génomes :
   Définition 14 ($\prec_{1-}$) *On dit que* $a_i \prec_{1-} a_j$ *lorsque* $a_i^A < a_j^A$ *et* $a_i^B > a_j^B$

Cette stratégie a l'avantage d'augmenter les possibilités tout en gardant des problèmes de taille réduite, donc simples à résoudre. Cependant, elle fait l'hypothèse que la recherche de régions homologues de polarité positive et négative sont deux problèmes indépendants, ce qui est un choix discutable (Voir section 3.1.2 page 53).

Changement d'ordre et d'orientation : une dernière relation accepte les changements d'ordre et d'orientation des ancres. Dans ce cas la notion d'*ancres chaînables* est une relation d'ordre sur un des deux génomes uniquement, ce qui laisse la possibilité de changer l'ordre des ancres sur le second.

Définition 15 ($\prec_2$) *On dit que* $a_i \prec_2 a_j$ *lorsque* $a_i^A < a_j^A$

Cette fois tous les mouvements évolutifs sont modélisés et c'est pour cette raison que ce type d'ordre est utilisé dans des outils comme (Pevzner et Tesler, 2003; Soderlund *et al.*, 2006). Cette relation correspond également à la notion de *1-monotonic conservation map* utilisée dans Shuffle-LAGAN (Brudno *et al.*, 2003) en alignement génomique pour capturer des alignements avec remaniements (translocations, inversions...).

Comme nous le verrons par la suite, il n'y a plus aucune garantie de qualité sur les ancres chaînées à partir de cette relation : n'importe quel ensemble d'ancres peut être ordonné sur le génome A. Ainsi, l'utilisation de cette relation nécessite de renforcer d'autres éléments qui interviennent dans la construction des régions dupliquées, comme les fonctions de scores (voir section 2.4.1.5 page 34) ou la contrainte de distance. Contrainte qui est décrite ci-dessous.

La contrainte de distance : cette contrainte est ajoutée à chaque définition de $\prec$. Elle a pour but de supprimer la relation entre deux *ancres chaînables* qui sont trop éloignées. Elle s'exprime sous cette forme :

$$dist(a_i, a_j) < distMax \tag{2.4}$$

Une nouvelle fois, il existe plusieurs stratégies pour calculer $dist(a_i, a_j)$. La plus simple utilise une distance de Manhattan : $Manhattan(a_i, a_j)$. Cette distance a la propriété de calculer une distance deux fois plus grande pour deux ancres séparées par n substitutions, par rapport à n indels (voir figure 2.6 page 29). L'association de deux ancres séparées par des substitutions étant préférable à l'association de deux ancres séparées par des indels, l'utilisation de cette distance semble peu adaptée à notre problématique.

Afin de ne pas pénaliser les ancres séparées par des substitutions, il est possible d'utiliser une fonction de distance qui dépend du génome g : $dist(a_i^g, a_j^g)$. De plus, sachant que la distance moyenne entre gènes varie selon les génomes, SyMAP (Soderlund *et al.*, 2006) ajoute une distance maximum qui dépend du génome: $distMax^g$. Cette dernière solution a l'avantage de s'adapter aux caractéristiques de chaque séquence, ce qui la rend plus robuste que les précédentes.

La dernière possibilité pénalise les indels par rapport aux substitutions (voir figure 2.10 page 37) :

$$dist_d(a_i, a_j) = 2 * max_g\{dist(a_i^g, a_j^g)\} - min_g\{dist(a_i^g, a_j^g)\} \text{ (Vandepoele } et\ al.\text{, 2002)} \tag{2.5}$$

Cette stratégie a l'avantage de favoriser la création de régions homologues de très bonne qualité (diagonale), mais en contrepartie elle a des difficultés pour détecter des régions homologues anciennes.

Remarque 12 *La distance entre deux ancres est aussi utilisée pour mesurer la qualité de deux* ancres chaînables *(voir les fonctions de score 2.4.1.5 page 34).*

Remarque 13 *Pour généraliser au cas multi-chromosomique, il faut ajouter aux définitions de $\prec$ la contrainte : $a_i^g.chrom = a_j^g.chrom$ avec $a_i^g.chrom$ le chromosome de a_i sur le génome g.*

2.4.1.4 Le graphe de relation

Nous avons vu dans la section précédente que les relations $\prec$ définissent un ordre partiel sur les ancres. Cet ordre se capture naturellement sous la forme d'un graphe appelé graphe de relation :

Définition 16 (Graphe de relation) *Pour un ensemble d'ancres donné $\mathcal{A}$, le graphe de relation $R = (\mathcal{A}, E)$ est un graphe orienté dont l'ensemble des sommets est $\mathcal{A}$ et qui contient un arc $(a_i, a_j) \in E$ si et seulement si $a_i \prec a_j$.*

La relation $\prec$ étant une relation d'ordre sur le génome A (définition 9 page 30), le graphe de relation R est sans circuit. Tout chemin dans ce graphe représente une succession d'ancres qui sera appelée une *chaîne* par la suite. Toute chaîne définit donc une paire de régions homologues potentielles.

Afin de différencier les chaînes, un score sera associé à chaque élément de R (arc et sommet) avec l'idée que plus le score est élevé, plus il est vraisemblable que l'élément appartienne à une paire de régions homologues. Mais avant de détailler les fonctions de score, nous allons nous attarder un peu sur cette notion de graphe de relation et détailler les quatre graphes de relation obtenus par les différentes relations $\prec$ sur un petit exemple.

Commençons par le graphe de relation obtenu avec $\prec_\varnothing$ (figure 2.7). Sur cet exemple, ce graphe ne contient aucun chemin qui relie toutes les ancres. Les deux régions dupliquées sont décomposées en deux chemins :

- Le premier, de polarité positive, ne contient que deux ancres. De plus, ces ancres sont assez éloignées, ce qui diminue sa qualité et ses chances d'être sélectionné.
- Le second, de polarité négative, contient trois ancres proches deux à deux. Ce chemin est a priori de bonne qualité et il sera sûrement sélectionné, mais il ne contient qu'une partie des deux régions dupliquées.

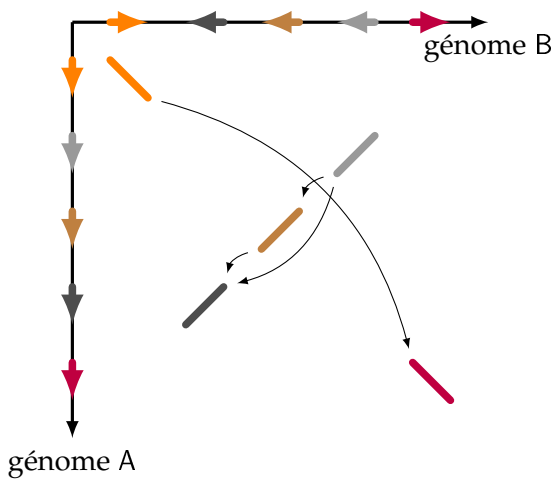


FIGURE 2.7: Graphe de relation obtenu avec la relation $\prec_\varnothing$.

Passons aux graphes de relation obtenus avec $\prec_{1+}$ et $\prec_{1-}$ (figure 2.8 page suivante). Encore une fois, il n'y a aucun chemin qui relie toutes les ancres des deux régions dupliquées. Cependant, le $graphe_+$ possède trois chemins plus ou moins équivalents qui délimitent correctement les régions dupliquées. Et contrairement à $\prec_\varnothing$, l'ajout d'une ancre de polarité différente augmente la qualité des chemins, ce qui permettra d'en sélectionner un. En ce qui concerne le $graphe_-$, le chemin de polarité négative constitué de trois ancres sera aussi sélectionné. Ainsi il y aura une ancre qui sera utilisée

dans deux chemins[6], ce qui n'est pas cohérent au niveau de l'histoire évolutive des séquences, nous y reviendrons dans la section 3.1.2 page 53.

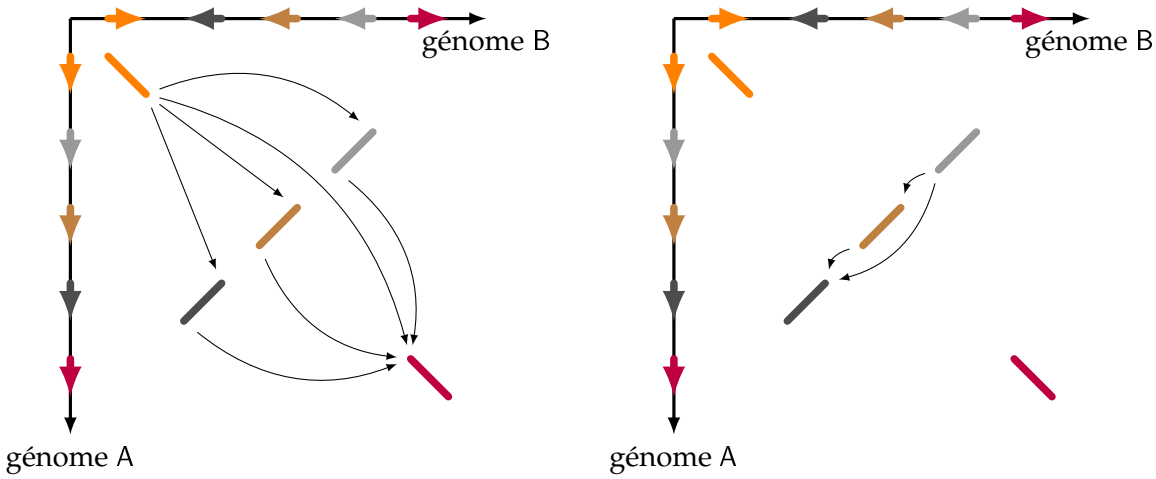


FIGURE 2.8: Graphes de relations obtenus avec les relations $\prec_{1+}$ et $\prec_{1-}$. J'appelle $graphe_+$ (resp. $graphe_-$) le graphe de gauche (resp. droite) obtenu avec $\prec_{1+}$ (resp. $\prec_{1-}$).

Au final, le graphe de relation obtenu avec $\prec_2$ (figure 2.9 page suivante) est le seul graphe à posséder un chemin qui passe par toutes les ancres des régions homologues pourvu que la contrainte de distance ne supprime pas l'arc $(orange, gris\ clair)$ et l'arc $(gris\ foncé, pourpre)$. Mais les nombreux arcs ajoutés ne facilitent pas la sélection de ce chemin, en comparaison avec les chemins des graphes précédents, et il le sera encore moins dans un cas réel[7].

2.4.1.5 Les fonctions de scores

Nous venons de voir comment construire le graphe de relation R. Il faut maintenant différencier les ancres et les chemins les plus intéressants *a priori*, à l'aide d'un score associé à chaque sommet ($S(a)$) et à chaque arc ($S(a_i \prec a_j)$) de R. Nous allons détailler les différentes stratégies utilisées et les fonctions de score qui leurs sont associées.

Score des sommets : en ce qui concerne la fonction de score des ancres (sommets), la solution la plus simple et la plus utilisée ne fait aucune différence entre les ancres, et un score constant est associé à tous les sommets $S_0(a) = cte$. C'est certainement la solution la plus raisonnable si la majorité des ancres appartiennent à des régions homologues. Dans ce cas de figure, la qualité de l'ancre est moins pertinente que sa position sur les génomes, information récupérée par $S(a_i \prec a_j)$.

6. un chemin dans $graphe_+$ et un chemin dans $graphe_-$
7. Essayez d'imaginer les graphes avec les données de la figures 2.4 page 26

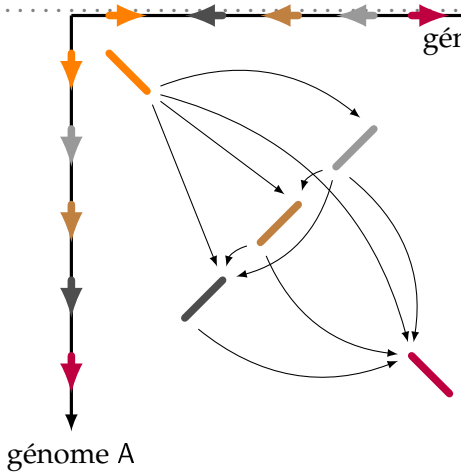


FIGURE 2.9: Graphe de relation obtenu avec la relation $\prec_2$.

Cependant l'exemple de la figure 2.4 page 26, nous montre que la majorité des ancres ne participent pas à des régions homologues. Il est alors logique de favoriser l'utilisation d'ancres de bonne qualité.

Les logiciels d'alignements fournissent une mesure de la qualité de chaque ancre, sous la forme d'une *e-value* (cf. définition 7 page 21) ou d'un score, qui est utilisé sous cette forme :

$$\begin{array}{rcl} S_e(a) & = & \min\{-log_{10}(a.e), MaxMatch\} \text{ (Haas } et\ al.\text{, 2004)} \\ S_s(a) & = & a.s \end{array} \quad (2.6)$$

avec $MaxMatch$ le score maximum d'une ancre afin de borner les log des *e-value* dont la représentation numérique est nulle.

Une autre solution consiste à s'appuyer sur un modèle probabiliste permettant d'associer deux probabilités à chaque ancre :

- $P(a|M^+_{ancre})$, la probabilité d'observer l'ancre a si les deux régions sont homologues ;
- $P(a|M^-_{ancre})$, la probabilité d'observer l'ancre a si les deux régions ne sont pas homologues :

$$S_p(a) = log\frac{P(a|M^+_{ancre})}{P(a|M^-_{ancre})} \text{ (Hachiya } et\ al.\text{, 2009)} \quad (2.7)$$

Le calcul des probabilités dépend de plusieurs paramètres, comme la longueur par exemple. Nous y reviendrons un peu plus tard dans la section optimisation 2.4.1.7.

Remarque 14 *Par la suite, afin de faciliter la compréhension du lecteur, je note :*

- $S(a)$ *le paramètre qui recueil la fonction de score utilisé ;*

- S_{cte} *la fonction de score* $S_0(a)$;
- S_{evalue} *la fonction de score* $S_e(a)$;
- S_{score} *la fonction de score* $S_s(a)$;
- S_{prob} *la fonction de score* $S_p(a)$;

Score des arcs : comme pour les scores des sommets, il existe plusieurs solutions pour mesurer la qualité de l'association de deux ancres $a_i \prec a_j$. La solution la plus évidente consiste à utiliser la distance qui sépare deux ancres. Cette distance peut se calculer de trois façons différentes, menant à plusieurs définitions du score d'un arc :

- La plus simple utilise une distance de Manhattan :

$$S_m(a_i \prec a_j) = -Manhattan(a_i, a_j) \tag{2.8}$$

  Cette distance a la propriété de calculer une distance deux fois plus grande pour deux ancres séparées par n substitutions, par rapport à n indels (voir figure 2.6 page 29). L'association de deux ancres séparées par des substitutions étant préférable à l'association de deux ancres séparées par des indels, l'utilisation de cette distance risque d'avoir un impact négatif sur les résultats.

- Afin de ne pas pénaliser l'association des ancres dont les positions relatives sont cohérentes avec une duplication simultanée (sans indels supplémentaires), on peut introduire le score S_n. Ainsi, pour deux ancres a_i et a_j séparées par $\Delta^A = (a_j^A.deb - a_i^A.fin)$ et $\Delta^B = (a_j^B.deb - a_i^B.fin)$, le score utilisée est :

$$S_n(a_i \prec a_j) = -\frac{\Delta^A + \Delta^B + |\Delta^A - \Delta^B|}{2} \text{ (Haas } et\ al.\text{, 2004)} \tag{2.9}$$

  Ce score s'appuie sur la distance euclidienne pour les horizontales ou les verticales sur un dotplot mais qui divise la distance euclidienne par $\sqrt{2}$ pour les diagonales (voir figure 2.6 page 29).

- La dernière solution pénalise les indels par rapport aux substitutions, pour favoriser les associations d'ancres dans la diagonale (voir figure 2.10 page ci-contre) à l'aide de $dist_d$ (équation 2.5 page 32, page 32):

$$S_d(a_i \prec a_j) = -dist_d(a_i, a_j) \text{ (Vandepoele } et\ al.\text{, 2002)} \tag{2.10}$$

Une autre stratégie consiste, comme pour le score des ancres, à s'appuyer sur un modèle probabiliste permettant d'associer deux probabilités à chaque arc $a_i \prec a_j$:

- $P((a_i, a_j)|M^+_{arc})$, la probabilité d'observer l'arc (a_i, a_j) si les deux régions sont homologues ;
- $P((a_i, a_j)|M^-_{arc})$, la probabilité d'observer l'arc (a_i, a_j) si les deux régions ne sont pas homologues :

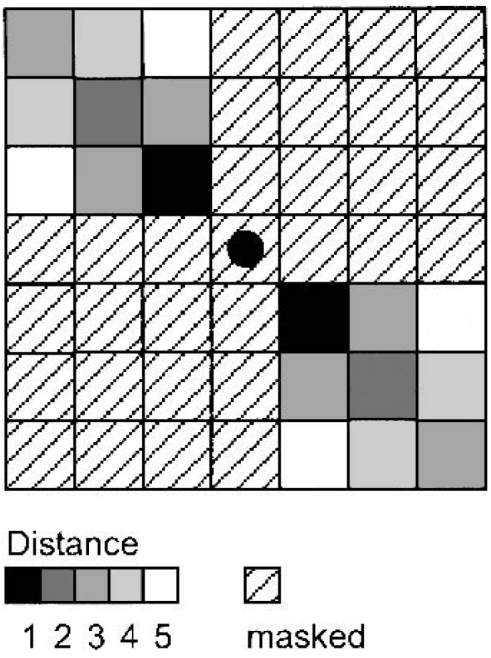


FIGURE 2.10: Matrice de distance de l'équation 2.5 page 32. Le cercle représente la position de l'ancre. Figure tirée de l'article d'ADHoRe (Vandepoele *et al.*, 2002)

$$S_p(a_i \prec a_j) = log\frac{P((a_i,a_j)|M^+_{arc})}{P((a_i,a_j)|M^-_{arc})} \text{ (Hachiya } et\ al.\text{, 2009)} \tag{2.11}$$

Remarque 15 *Par la suite, afin de faciliter la compréhension du lecteur, je note :*

- $S(a_i, a_j)$ *le paramètre qui recueil la fonction de score utilisé ;*
- S_{indel} *la fonction de score* $S_m(a_i \prec a_j)$ *qui favorise les insertions et les délétions;*
- S_{sub} *la fonction de score* $S_d(a_i \prec a_j)$ *qui favorise les substitutions;*
- S_{neutre} *la fonction de score* $S_n(a_i \prec a_j)$ *qui ne favorise ni l'un ni l'autre;*
- S_{vrai} *la fonction de score* $S_p(a_i \prec a_j)$ *qui favorise la vraisemblance des données (voir équation 2.14 page 41);*

2.4.1.6 Les chaînes

Avant de détailler la phase d'optimisation, la notion de chaîne doit être précisée. Une chaîne est un chemin dans le graphe de relation R. Par définition de la relation $\prec$, toute chaîne c traverse des ancres dont les intervalles se succèdent sur le génome A (mais pas nécessairement sur le génome B). A chaque chaîne c, et pour chaque génome $g \in \{A, B\}$, on peut associer un ensemble d'informations : le début de la chaîne $c^g.deb$, défini par la plus petite position d'une ancre dans c sur g, et sa fin $c^g.fin$ (définie à

partir de la plus grande position, on a donc $c^g.deb < c^g.fin$). Comme pour les ancres, une chaîne c définit donc deux intervalles sur chacun des génomes, notés c^A et c^B. Le nombre d'ancres dans la chaîne est noté $c.n$.

À partir de l'ensemble des positions de début et de fin de l'ensemble des ancres traversées par une chaîne, il est possible de calculer un coefficient de corrélation de Bravais-Pearson entre les coordonnées sur le génome A et sur le génome B. Ce coefficient, noté $c.corr$, renseigne sur le degré de dépendance linéaire entre les coordonnées et fournit une mesure de la conservation de l'organisation des ancres du segment sur les deux génomes. Ce coefficient est particulièrement utile pour vérifier la qualité des chaînes construites avec la relation $\prec_2$ et son signe permet de définir la polarité $c.pol$ d'une chaîne c. Enfin, le score d'une chaîne $c.s$ est défini comme la somme des scores des éléments (sommets et arcs) qui la composent.

Définition 17 (Une chaîne) *Une chaîne c_i est déterminée par un chemin dans R. Elle définit cinq composantes :*
- *un intervalle c_i^A sur le génome A*
- *un intervalle c_i^B sur le génome B*
- *un coefficient de corrélation de Bravais-Pearson $c_i.corr$*
- *une polarité notée $c_i.pol$*
- *un score noté $c_i.s$*

Remarque 16 *Comme pour les ancres (voir remarque 10 page 29), $c_i^A.deb < c_i^A.fin$ et $c_i^B.deb < c_i^B.fin$.*

2.4.1.7 Optimisation

Méthodes	Séquences	Distance	Graphe de relation	Score sommet
ADHoRe	géniques	$dist_d(a_i, a_j)$	$\prec_\varnothing$	S_{cte}
GRIMM-Synteny	nucléiques	$Manhattan(a_i, a_j)$	$\prec_2$	$\varnothing$
FISH	géniques	$Manhattan(a_i, a_j)$	$\prec_\varnothing$	S_{cte}
DiagHunter	géniques	$dist_d(a_i, a_j)$	$\prec_\varnothing$	S_{cte}
DAGchainer	nucléiques	$dist(a_i^g, a_j^g)$	$\prec_1$	S_{evalue}
SyMAP	nucléiques	$dist(a_i^g, a_j^g)$	$\prec_2$	S_{cte}
OSfinder	nucléiques	$Manhattan(a_i, a_j)$	$\prec_\varnothing$	S_{prob}

TABLE 2.1: Tableau qui récapitule les choix faits par les sept méthodes. Au niveau des ancres, toutes les méthodes ont été testées principalement avec des ancres protéiques, même s'il est mentionné que certaines acceptent les ancres nucléiques.

Une fois le graphe de relation R construit (voir le tableau 2.1) la reconstruction des régions dupliquées est résolue par deux types de méthodes que je classe en deux catégories :

– La première crée une région dupliquées par composante connexe. Elle est utilisée par les outils GRIMM-Synteny (Pevzner et Tesler, 2003) et DiagHunter (Cannon *et al.*, 2003);
– La seconde peut créer plusieurs régions similaires par composante connexe. Elle est utilisée par les outils FISH (Calabrese *et al.*, 2003), ADHoRe (Vandepoele *et al.*, 2002), DAGchainer (Haas *et al.*, 2004), SyMAP (Soderlund *et al.*, 2006) et OSfinder (Hachiya *et al.*, 2009).

Définition 18 (Composante connexe) *Un graphe non-orienté est connexe s'il existe un chemin entre tout couple de sommets. Quand on parle de connexité pour un graphe orienté, on considère non pas ce graphe mais le graphe non-orienté correspondant. Le parcours en profondeur (Ahuja et al., 1993) est un exemple d'algorithme pour déterminer les composantes connexes.*

Le chaînage par GRIMM-Synteny :

La méthode de GRIMM-Synteny consiste à chaîner toutes les ancres d'une composante connexe, dans le même ordre que leurs positions sur le génome A. Il n'y a aucune vérification de la qualité globale de la chaîne en dehors d'un critère de longueur.

$$c^g >= tailleMin$$

Au final GRIMM-Synteny reconstruit ces régions similaires à l'aide d'un unique critère, local à deux ancres : la distance maximum qui les séparent (voir équation 2.4 page 32). Ce critère est important mais il ne suffit pas à modéliser la complexité des évènements biologiques. L'exemple de la figure 2.4 page 26 nous montre que cette distance est variable selon les régions et au sein d'une même région similaire.

Évolution de la méthode par DiagHunter :

Comme GRIMM-Synteny, DiagHunter crée une chaîne par composante connexe. Mais contrairement à ce dernier, DiagHunter est capable de construire une chaîne sans utiliser toutes les ancres de la composante connexe. Pour cela il part de la première ancre d'une composante connexe et utilise une méthode itérative qui sélectionne l'ancre la plus proche de la dernière sélectionnée, à chaque étape.

Cette méthode associée à l'utilisation d'une fonction de distance, qui favorise l'association d'ancres dans la diagonale (voir figure 2.10 page 37), permet de limiter la sélection d'ancres avec un fort potentiel de ne pas appartenir à des régions homologues, puisque non nécessaire à la création des deux régions similaires. Cependant, la distance entre deux ancres reste le principal critère pour reconstruire les régions dupliquées, et la qualité globale de la chaîne n'est toujours pas considérée.

Les méthodes de la deuxième catégorie :

Contrairement à GRIMM et DiagHunter, les méthodes de la deuxième catégorie offrent la possibilité de commencer et de finir une chaîne sur n'importe quelle ancre de R. Cette caractéristique permet de créer plusieurs chaînes par composante connexe et ainsi d'être moins dépendante de la contrainte de distance (équation 2.4 page 32).

A l'exception d'ADHoRe (que nous verrons plus tard), le principe général est le suivant :

1. chercher la meilleure chaîne de R ;
2. supprimer de R les ancres utilisées dans la chaîne ;
3. recommencer jusqu'à ce que la meilleure chaîne soit de qualité insuffisante, inférieure à $scoreMin$.

La qualité globale d'une chaîne est mesurée à l'aide de son score *c.s*. Ainsi, à chaque étape, ces méthodes cherchent le chemin de score maximum dans R. Ce problème est bien connu en théorie des graphes et il est résolu par un algorithme de programmation dynamique, généralement connu sous le nom de *plus court chemin* (DAG-Shortest-Paths (Ahuja *et al.*, 1993)). En effet, maximiser le score d'un chemin dans un graphe sans circuit est équivalent à minimiser son coût, les deux notions étant liés :

$$score = -cout \tag{2.12}$$

Remarque 17 *L'utilisation de coûts (ou de scores) négatifs peut générer des circuits absorbants*[8] *qui font obstacle à la recherche du plus court chemin. Dans notre cas, le graphe de relation R étant par définition orienté et sans circuit, l'utilisation de coûts négatifs ne génère pas de circuits absorbants (Ahuja* et al.*, 1993; Gondran et Minoux, 2009).*

L'utilisation d'un score pour mesurer la qualité des chaînes est plus performante que l'utilisation binaire de la contrainte de distance. Cependant, elle demande de fixer plusieurs paramètres, qui dépendent des données, ce qui est souvent un obstacle pour les utilisateurs ou la source de mauvais résultats. Les paramètres ne sont pas identiques pour chaque méthode, mais on peut notamment citer : $tailleMin$, $scoreMin$, $distMax$. Afin de supprimer ce défaut, OSfinder propose une optimisation automatique des paramètres, que nous allons étudier maintenant.

Remarque 18 *Il est intéressant de noter que, dans le cas particulier où l'on utilise* $\prec_0$*, la recherche d'un plus court chemin dans le graphe de relations R est équivalente à l'algorithme de construction d'une chaîne de poids maximum non chevauchante (maximum weighted chain, MWC) telle que décrite dans (Hohl* et al.*, 2002) dans le cadre de l'alignement de génomes multiple (Multiple GWA).*

Optimisation automatique des paramètres :

Avant de procéder au chaînage des ancres par les trois étapes vues ci-dessus, OSfinder commence par optimiser les paramètres de son modèle probabiliste en maximisant la vraisemblance des données selon ce modèle. Cette technique est connue sous le nom d'"EM" (Espérance et Maximisation de Dempster *et al.* (1977)). Cette méthode a l'avantage de fournir une nouvelle source d'information globale, à partir de l'ensemble des données, qui complète celle fournie par les paires d'ancres (locales) modélisées dans R.

8. circuit de coût négatif

Pour cela OSfinder utilise un modèle probabiliste qui différencie les régions homologues des non homologues à l'aide d'un critère de longueur sur les ancres et les arcs :

- Plus une ancre est longue plus il est probable qu'elle appartienne à une région homologue ;
- Plus un arc est long plus il est probable qu'il n'appartienne pas à une région homologue.

Ainsi chaque ancre et arc possède deux probabilités modélisées à l'aide de quatre chaînes de Markov (figure 2.11) à deux états :

- *ext* qui représente le fait de rester dans le même type de région à l'étape suivante (homologue ou non)
- *fin* qui représente le fait de quitter la région à la fin de l'ancre ou de l'arc.

qui sont utilisées pour représenter les distributions géométriques de chacun des cas :

$$\begin{array}{rcl} P(a_i|M^+_{ancre}) & = & P(ext|M^+_{ancre})^{a_i.l-1} \times P(fin|M^+_{ancre}) \\ P(a_i|M^-_{ancre}) & = & P(ext|M^-_{ancre})^{a_i.l-1} \times P(fin|M^-_{ancre}) \\ P((a_i,a_j)|M^+_{arc}) & = & P(ext|M^+_{arc})^{distM(a_i,a_j)-1} \times P(fin|M^+_{arc}) \\ P((a_i,a_j)|M^-_{arc}) & = & P(ext|M^-_{arc})^{distM(a_i,a_j)-1} \times P(fin|M^-_{arc}) \end{array} \quad (2.13)$$

avec $distM(a_i, a_j)$ la distance de Manhattan entre a_i et a_j et $a_i.l$ la somme des intervalles des deux génomes.

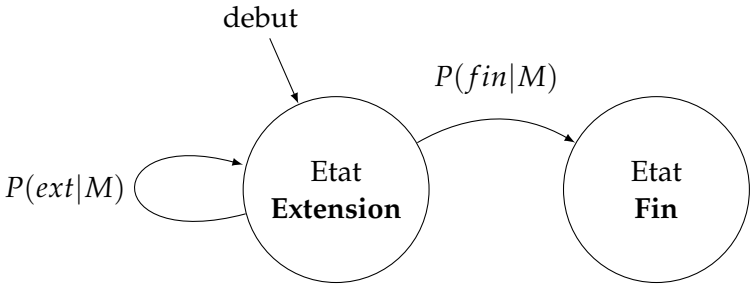


FIGURE 2.11: Modèle générique de chaîne de Markov utilisé par OSfinder.

Maintenant, supposons que nous disposions d'un ensemble de labels L qui définit :

- un ensemble d'ancres homologues $\mathcal{A}^+$ et un ensemble d'ancres non homologues $\mathcal{A}^-$;
- un ensemble d'arcs homologues E^+ et un ensemble d'arcs non homologues E^- ;

avec $\mathcal{A} = \mathcal{A}^+ \cup \mathcal{A}^-$ et $E = E^+ \cup E^-$. Cet ensemble de labels permet de calculer la vraisemblance du graphe de relation R, avec le produit de chaque élément qui le compose :

$$\begin{array}{rcl} P(R|M,L) & = & \prod_{a_i \in \mathcal{A}^+} P(a_i|M^+_{ancre}) \times \prod_{a_i \in \mathcal{A}^-} P(a_i|M^-_{ancre}) \\ & \times & \prod_{(a_i,a_j) \in E^+} P((a_i,a_j)|M^+_{arc}) \times \prod_{(a_i,a_j) \in E^-} P((a_i,a_j)|M^-_{arc}) \end{array} \quad (2.14)$$

A l'aide de cet ensemble de labels, il est possible de calculer la longueur moyenne des éléments pour chaque modèle et d'optimiser les paramètres des modèles afin de maximiser la probabilité de L : $\hat{M}_L = \text{argmax}_M P(G|M, L)$ à l'aide de ces quatre équations [9] :

$$\begin{array}{rclcrcl} P(fin|M^+_{ancre}) & = & \frac{|\mathcal{A}^+|}{\sum_{a_i \in \mathcal{A}^+} a_i.l} & & P(fin|M^-_{ancre}) & = & \frac{|\mathcal{A}^-|}{\sum_{a_i \in \mathcal{A}^-} a_i.l} \\ P(fin|M^+_{arc}) & = & \frac{|E^+|}{\sum_{(a_i,a_j) \in E^+} distM(a_i,a_j)} & & P(fin|M^-_{arc}) & = & \frac{|E^-|}{\sum_{(a_i,a_j) \in E^-} distM(a_i,a_j)} \end{array} \tag{2.15}$$

et d'en déduire les probabilités $P(ext|M)$, avec la propriété $P(ext|M) + P(fin|M) = 1$.

Maintenant, l'objectif consiste à trouver l'ensemble de labels qui maximise la vraisemblance des données. La méthode triviale énumère tous les ensembles de labels possibles. Cependant, cet ensemble est beaucoup trop grand $2^{(|\mathcal{A}|+|E|)}$ et OSfinder propose une recherche itérative plus intelligente qui trouve le maximum de vraisemblance à l'aide de $(|\mathcal{A}| + |E| - 2)$ labels. Dans le cas où cette recherche serait encore trop longue, OSfinder propose une méthode heuristique pour calculer un maximum local de la vraisemblance.

Une fois que l'apprentissage des paramètres est terminé, le graphe de relation R est valué à l'aide des équations 2.7 page 35, 2.11 page 37 et la construction de chaînes à l'aide de l'algorithme de plus court chemin peut commencer.

Le cas particulier d'ADHoRe :

ADHoRe fait aussi partie des logiciels qui peuvent créer plusieurs chaînes par composante connexe. Cependant il a choisi le coefficient de détermination r^2 comme critère pour mesurer la qualité globale des chaînes, au lieu du score $c.s$.

Définition 19 (coefficient de détermination) *Le coefficient de détermination (r^2) est un indicateur qui permet de juger la qualité d'une régression linéaire. Dans le cadre d'une régression linéaire simple (c'est le cas ici), r^2 est le carré du coefficient de corrélation de Bravais-Pearson.*

Cependant le calcul de r^2 n'est pas décomposable en une somme d'éléments distincts, contrairement à $c.s$, et la sélection des chemins par l'algorithme de plus court chemin n'est plus possible. ADHoRe est contraint d'utiliser une autre méthode de chaînage, dont le principe est le suivant :

1. Grouper les ancres d'une distance ≤ 3 gènes (figure 2.10 page 37) dans des clusters ;

9. Intuitivement on comprend qu'elles sont justes, pour en avoir la preuve j'invite les lecteurs à se reporter à l'article (Hachiya *et al.*, 2009)

2. Tester la linéarité des ancres de chaque cluster à l'aide du coefficient de détermination r^2. Si le test est négatif, le cluster est supprimé mais pas ses ancres qui peuvent encore être sélectionnées dans d'autres clusters ;
3. Ajouter les ancres aux clusters qui sont à distance ≤ 4 à condition que le test de linéarité soit valide ;
4. Grouper les clusters d'une distance ≤ 4 à condition que le test de linéarité soit valide ;
5. Recommencer les deux dernières étapes en incrémentant les seuils de distance de 1 jusqu'à $distMax$;
6. Transformer les clusters en chaînes.

Cette méthode a l'avantage de vérifier une caractéristique biologique des régions homologues (la linéarité des ancres) pendant la construction des chaînes. Tandis que l'algorithme de plus court chemin en est incapable, il s'appuie uniquement sur la qualité du graphe de relation et les fonctions de score qui lui sont données. Au mieux, un filtre est appliqué à la sortie.

2.4.1.8 Conclusion

Si on prend un peu de recul sur cette partie consacrée aux méthodes de chaînage, il est important de remarquer que malgré leurs différences ces méthodes se ressemblent énormément : construction d'un graphe de relation à l'aide des *ancres chaînables*, puis reconstruction des régions dupliquées par le chaînage des ancres.

On remarque aussi qu'avec l'évolution des méthodes, le processus de chaînage est de moins en moins dépendant de la distance qui sépare deux ancres successives. En effet cette distance est un critère local qui ne tient pas compte de "l'aspect global" des régions dupliquées reconstruites (GRIMM et Diaghunter) et c'est une évolution logique d'essayer de pondérer ce critère. Cette évolution a commencé par l'utilisation d'un algorithme de plus court chemin (FISH, DAGchainer, SyMAP et OSfinder) puis par l'optimisation des paramètres en maximisant la vraisemblance des données (OSfinder).

Définition 20 (Méthodes gloutonnes) *Une méthode gloutonne est une méthode qui résout un problème, étape par étape, à l'aide d'une suite de choix qui optimisent un critère local, dans l'espoir d'obtenir un résultat qui optimise un critère global.*

Cependant, dès lors que l'on passe d'une problématique d'identification de chaînes **indépendantes** à une problématique de construction d'un **ensemble** de chaînes qui ne partage pas d'ancres, le choix d'une chaîne optimale (obtenue par programmation dynamique) à chaque étape ne garantit pas l'obtention d'un ensemble de chaînes optimum, comme maximiser la somme des scores de l'ensemble des chaînes. Une telle approche sera, par la suite, qualifiée de gloutonne.

ADHoRe est un cas particulier. C'est la première méthode publiée, mais elle a l'avantage d'utiliser un critère global (le coefficient de détermination) pour juger de la qualité linéaire des chaînes. Cependant sa méthode de chaînage est totalement heuristique et ad-hoc et il n'y a aucune garantie sur les résultats obtenus.

Pour conclure, malgré la tentative d'ADHoRe d'utiliser un critère global ou les diverses évolutions apportées par les autres méthodes pour minimiser l'importance de la distance qui sépare deux *ancres chaînables*, les méthodes de chaînages restent gloutonnes.

2.4.2 Une approche statistique

Dans cette section je décris et compare trois logiciels : LineUp (Hampson *et al.*, 2003), CloseUp (Hampson *et al.*, 2005) et CHSMiner (Wang *et al.*, 2009) ; qui ont choisi de reconstruire les régions homologues à l'aide de tests statistiques permettant de vérifier si la similarité de deux régions est significative.

Afin d'être le plus exhaustif possible, ces logiciels reconstruisent un couple de régions potentiellement homologues r_i par ancre a_i. Elles utilisent un processus itératif qui essaye d'ajouter une nouvelle ancre à r_i validé par un test statistique, à chaque itération. Le test statistique est propre à chaque logiciel. LineUp et CloseUp utilisent des simulations de Monte Carlo pour calculer la probabilité de tirer n gènes homologues (ancres) parmi les m gènes de r. Cette technique peut être très coûteuse en temps et CHSMiner préfère utiliser une méthode analytique.

Dans le détail, ces trois méthodes demandent en entrée les mêmes données que celles utilisées par les logiciels de chaînage, soit un ensemble d'ancres protéiques $\mathcal{A}$ et deux séquences géniques A et B. Avec ces données, ils commencent par initialiser r_i avec l'ancre de départ, $r_i = a_i$ ce qui définit un intervalle initial sur chaque génome : r_i^{A} et r_i^{B}.

Ensuite, le principe consiste à parcourir la séquence A de gauche à droite à partir de la fin de r_i^{A}, jusqu'à la rencontre d'un gène utilisé dans une ancre a_j ou jusqu'à satisfaction d'un critère d'arrêt, ici une distance maximum. Si le gène rencontré est utilisé dans plusieurs ancres, l'ancre a_j choisie est la *plus proche* de r_i^{B}. La notion de plus proche dépend des méthodes et sera précisée plus bas.

Dans le cas où une ancre a_j est trouvée, une procédure est lancée pour vérifier si l'association $r_i \cup a_j$ respecte la distance maximum sur le génome B et satisfait un test statistique. Si l'association est validée, r_i est mise à jour $r_i = r_i \cup a_j$, et dans tous les cas le parcours de la séquence recommence jusqu'à la rencontre d'une nouvelle ancre ou la satisfaction du critère d'arrêt (voir figure 2.12 page suivante).

Dans le cas où la distance maximum est atteinte, la reconstruction de r_i est terminée et une autre ancre est choisie comme point de départ, jusqu'à ce que toutes les ancres

soient utilisées.

La méthode décrite ci-dessus, compare la séquence A à la séquence B. Cette comparaison est asymétrique et il est nécessaire ensuite de comparer la séquence B à la séquence A pour reconstruire les régions non détectées par la première comparaison (voir figure 2.12).

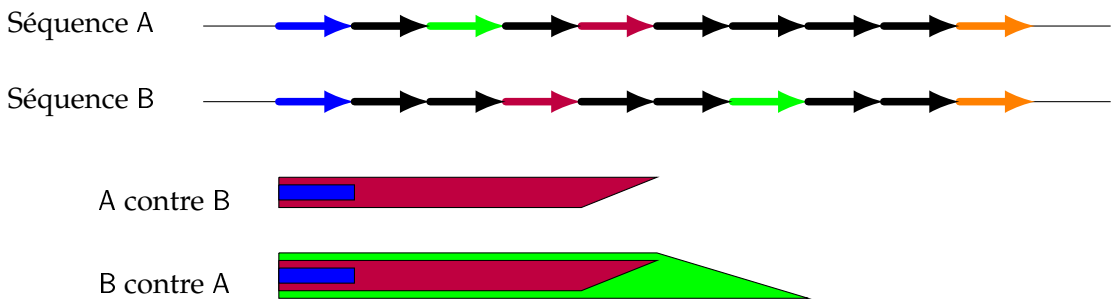


FIGURE 2.12: Exemple de reconstruction de régions homologues par les méthodes statistiques avec une distance max de 4. Deux gènes de la même couleur sont homologues sauf s'ils sont de couleur noire. En dessous des deux séquences, sont représentées les régions homologues reconstruites lors de la comparaison de la séquence A contre la séquence B puis l'inverse. Les différentes couleurs indiquent les frontières des régions homologues à chaque étape de sa reconstruction. Dans le cas où la séquence A est comparée à la B l'ancre verte n'est pas détectée car la distance avec le gène bleu est trop grande sur le génome B. Dans les deux cas la région n'est pas entièrement reconstruite car l'ancre la plus proche de l'ancre orange est à une distance de 5 sur la séquence A.

L'ancre la plus proche :

En ce qui concerne la notion de *plus proche*, LineUp est la plus ancienne des trois méthodes et comme pour les premières méthodes de chaînage, elle cherche à reconstruire des régions similaires ayant un ordre conservé (appelées colinéaires). En conséquence, seules les ancres qui se trouvent après l'intervalle r^B sont acceptées. CloseUp et CHSMiner acceptent les changements d'ordre. Ainsi, elles tiennent compte des ancres qui se trouvent après, mais aussi avant et au sein de l'intervalle r^B ; avec une distance nulle pour les ancres situées dans l'intervalle r^B.

2.4.3 Une approche globale du problème

Dans cette section, je décris une nouvelle méthode publiée en juin 2010 : EGM (Mahmood *et al.*, 2010). L'objectif est de retrouver les régions homologues à l'aide d'une sous-sélection de l'ensemble $\mathcal{A}$. Ce sous-ensemble sélectionne une ancre par gène en maximisant un score qui dépend de la similarité de l'ancre, du voisinage de l'ancre et de l'orientation des gènes. Cette méthode ne résout pas tout à fait le même problème que les méthodes précédentes. Elle se contente de faire une sélection d'ancres qui

doivent appartenir à des régions homologues, mais à aucun moment elle ne regroupe ses ancres pour reconstruire ses régions. Cependant elle a l'avantage de donner un résultat qui maximise un critère global.

EGM modélise l'ensemble des solutions sous la forme d'un graphe biparti où les sommets représentent les gènes utilisés dans des ancres et les arcs représentent la liaison entre deux gènes par ancre.

Définition 21 (Graphe biparti) *En théorie des graphes, un graphe est dit biparti s'il existe une partition de son ensemble de sommets en deux sous-ensembles U et V, telle que chaque arête ait une extrémité dans U et l'autre dans V.*

Ici l'ensemble U représente l'ensemble des gènes de la séquence A utilisés dans des ancres et V représente ceux de de la séquence B (voir figure 2.13). La sélection d'un sous-ensemble d'ancres ne possédant pas de gènes en commun est représentée par un couplage.

Définition 22 (Couplage) *En théorie des graphes, un couplage ou appariement d'un graphe est un ensemble d'arêtes de ce graphe qui n'ont pas de sommets en commun. Un couplage maximum est un couplage contenant le plus grand nombre possible d'arêtes. Un graphe peut posséder plusieurs couplages maximum.*

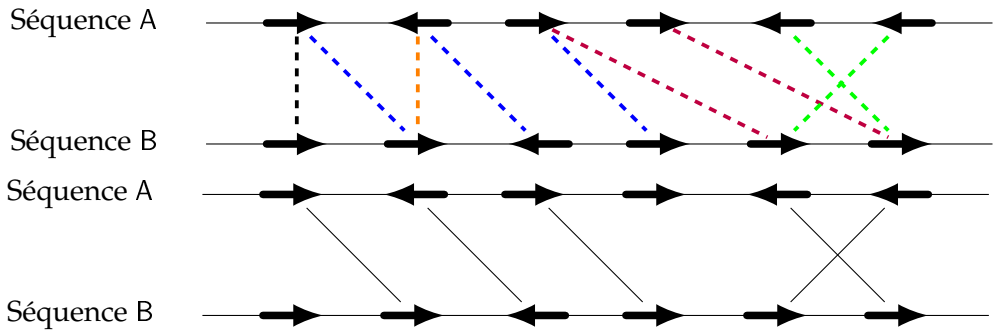


FIGURE 2.13: Exemple d'un graphe biparti et d'un couplage maximum. Sur le premier graphe les plus grands alignements locaux apparaissent sous différentes couleurs.

EGM cherche un couplage de score maximum [10]. Il reste cependant à donner un score à chaque arc. Pour cela il crée à partir de chaque arc le plus grand **alignement local d'ancres** : Les ancres doivent se suivre dans le même ordre sur les deux génomes (ou inversé si la polarité des ancres est négative) et de même polarité (voir figure 2.13 et section 2.2.2 page 18). Le score de l'arc est donné par la somme des pourcentages d'identité des ancres utilisées dans l'alignement.

Remarque 19 *Les arcs qui représentent les ancres d'un même alignement local d'ancres ont, par construction, un score identique.*

10. Un couplage maximum n'est pas forcement de score maximum

2.5 Analyse

2.5.1 Performance des méthodes existantes

2.5.1.1 Comment évaluer ?

À ce jour, il est délicat d'évaluer les performances des méthodes existantes. En effet, les régions homologues sont le résultat d'un ensemble de phénomènes biologiques encore mal compris et en conséquence il est possible de proposer plusieurs définitions des régions dupliquées. Chaque méthode propose son propre ensemble de critères à optimiser et vérifie que sa méthode donne effectivement de meilleurs résultats sur ces critères.

En conséquence, je propose d'évaluer les performances de ces méthodes à travers une analyse critique de leurs stratégies.

2.5.1.2 L'analyse critique

Composition d'une région homologue :

A l'exception de GRIMM-Synteny, les premières méthodes cherchent des régions homologues composées d'ancres colinéaires. C'est en effet la solution la plus simple et qui garantit de créer des régions de qualité. Cependant, cette caractéristique est trop exigeante et ne permet pas de détecter des régions homologues anciennes, les remaniements chromosomiques étant un mécanisme évolutif fréquent (Simillion *et al.*, 2004).

Le passage de $\prec_\varnothing$ à $\prec_2$ pour reconstruire des régions dupliquées non colinéaires, correspond de ce point de vue à l'évolution entre LineUp (Hampson *et al.*, 2003) et son amélioration ultérieure CloseUp (Hampson *et al.*, 2005), qui donne apparemment des résultats plus pertinents, en particulier lorsque les données sont de bonne qualité (utilisation d'ancres protéiques et non d'ancres nucléiques). Ce sont alors surtout les fonctions de score et non les contraintes de $\prec_2$ qui mènent à la construction de chaînes intéressantes.

En contrepartie les résultats donnés par les logiciels qui acceptent les changements d'ordre et d'orientation sont plus difficiles à justifier. Le risque de créer des régions homologues erronées augmente et il nécessaire de vérifier la qualité des chaînes à l'aide d'un coefficient de corrélation de Bravais-Pearson ou d'un examen visuel.

Stratégie de reconstruction :

A l'exception de EGM, toutes les méthodes utilisent des stratégies de reconstruction qui maximisent un critère local, comme le score d'une chaîne. Quand la reconstruction se limite à des régions homologues colinéaires, les méthodes gloutonnes ont peu de risque de créer de fausses régions homologues. Car les choix locaux sont limités par

une contrainte globale représentée par la colinéarité forcée des ancres. Ce qui n'est plus le cas avec les régions homologues non colinéaires, quand on accepte les changements d'ordres et d'orientations.

Afin de limiter la création de fausses régions homologues, des méthodes comme CloseUp utilisent un test statistique à chaque étape du processus de reconstruction. Néanmoins leurs méthodes restent gloutonnes et il est toujours possible de faire un mauvais choix à l'étape i, que le test statistique ne détectera qu'à l'étape $i + 1$ ou plus, soit trop tard pour être corrigé.

Pour finir, il est intéressant de noter que les dernières méthodes publiées utilisent des critères globaux. OSfinder, malgré le choix de créer des régions homologues colinéaires, utilise une méthode qui maximise la vraisemblance globale des données selon son modèle probabiliste, avant de construire son ensemble de chaînes par un processus itératif glouton. Récemment, une nouvelle étape été franchie avec EGM, qui utilise un critère global pour faire sa sélection d'ancres, mais il ne reconstruit pas les régions dupliquées.

2.5.2 Perspectives d'évolution

Évolution des méthodes de reconstruction

L'ensemble des régions homologues de deux séquences décrivent une histoire évolutive unique. Pourtant ces régions sont reconstruites une par une de façon indépendante par des procédures itératives gloutonnes. Il parait justifié de remplacer ces méthodes gloutonnes par des méthodes qui maximisent des critères globaux pour améliorer les performances des logiciels.

De plus, en l'absence de jeu de données où la vérité est connue, la création d'un ensemble de chaînes décrivant une histoire évolutive cohérente serait un plus pour justifier la qualité des résultats.

Évolution des besoins

Suite aux nouvelles technologies de séquençage, la quantité de séquences non annotées augmente régulièrement. L'annotation des séquences est un processus cher, compliqué, pas toujours fiable et qui prend du temps ; même quand il est réalisé par des algorithmes automatiques.

Une méthode efficace pour détecter les régions dupliquées à partir de la séquence d'ADN serait sûrement une aide précieuse pour analyser et/ou comparer les génomes et ainsi aider à comprendre leur évolution.

2.6 Conclusion

Cette étude de l'état de l'art de la recherche de régions dupliquées nous a permis de mettre en évidence les points suivants :

- Les méthodes de détection de duplication en tandem au niveau nucléique ont des difficultés avec une éventuelle perte locale de similarité et se limitent souvent à la détection de courtes séquences nucléiques dupliquées.
- La reconstruction de régions dupliquées est fortement dépendante de l'annotation des génomes.
- Dans les méthodes de chaînage, les chaînes sont construites indépendamment les unes des autres par des méthodes gloutonnes alors qu'elles appartiennent à une même histoire évolutive commune.
- Les méthodes qui acceptent de créer des régions dupliquées non colinéaires ne sont pas nombreuses et il est difficile de justifier leurs résultats. C'est certainement pour cela que DAGchainer, ADHoRe et depuis peu OSfinder restent les logiciels de référence.

Nous avons vu que la recherche de régions dupliquées non colinéaires était justifiée, pourtant il semblerait que les résultats ne soient pas suffisants. Le risque de créer de fausses régions dupliquées ou homologues et le manque de justification des résultats ne plaident pas en faveur de ces méthodes.

D'autre part, une méthode capable de détecter les régions dupliquées à l'aide de l'ADN uniquement (sans annotation) serait une source importante d'information pour nourrir l'annotation et/ou pour analyser les régions jouant un rôle important dans l'évolution des génomes.

Dans l'objectif de corriger ces deux points, nous allons développer une méthode de chaînage de reconstruction globale de régions dupliquées en tandem avec perte potentielle de similarité locale, qui permettra de prendre en compte la qualité globale d'une région (comme la linéarité des ancres) et la qualité globale de l'ensemble des régions (l'ensemble doit d'écrire une histoire évolutive cohérente) pendant la construction des régions.

Nous comparerons d'abord cette nouvelle méthode de chaînage avec les autres méthodes de sa catégorie, au niveau protéique, pour vérifier qu'elle obtient des résultats cohérents. Puis nous l'appliquerons à la recherche de duplications en tandem de grandes unités fonctionnelles (comme les gènes) au niveau ADN, et ainsi répondre à l'objectif que nous nous sommes fixé.

Cependant, la recherche de régions dupliquées aux niveau ADN est un problème encore plus complexe qu'au niveau protéique, notamment du fait de la quantité d'ancres en entrée. Se limiter aux duplications en tandem nous permet de limiter l'espace de recherche, ces duplications étant proches les unes des autres.

De plus, comme une grande majorité des régions dupliquées en tandem ne contient qu'une seule unité fonctionnelle, cela justifie de se limiter uniquement à la reconstruction de régions dupliquées colinéaires. Un changement d'ordre ou d'orientation de la séquence nucléotidique empêcherait certainement la traduction de cette séquence en protéine (voir figure 1.2 page 7).

Pour finir, nous avons vu dans l'introduction que les duplications en tandem participent de façon importante aux mécanismes d'évolution des génomes. Ainsi la détection de ces duplications au niveau ADN est en soi un résultat possédant un fort intérêt biologique.

Chapitre 3

Un modèle à base de graphes et sa résolution par recherche d'un flot

Sommaire

Ce chapitre est consacré à la présentation de la nouvelle méthode de chaînage développée durant ma thèse et qui a déjà fait l'objet d'une publication longue dans la conférence ROADEF (Audemard *et al.*, 2010a) et de plusieurs présentations (Audemard *et al.*, 2010c,b). Cette méthode a pour objectif de reconstruire un ensemble de régions dupliquées en tenant compte des duplications internes des génomes, en particulier dans le cas de la comparaison d'un génome contre lui-même. Cette méthode a aussi fait l'objet d'une implémentation dans le logiciel ReD (**Re**gions **D**upliquées).

Comme toutes les méthodes de chaînage, sa stratégie est d'associer des ancres pour reconstruire des régions dupliquées. Et à l'instar de ces méthodes, ReD utilise un graphe de relation pour modéliser l'ensemble des solutions, puis chaîne les ancres à l'aide de chemins dans ce graphe. A la différence des méthodes de chaînages existantes (comme MWC (Hohl *et al.*, 2002) ou DAGchainer (Haas *et al.*, 2004)), nous ne considérons pas le problème de la production d'une chaîne de score optimal, mais d'un ensemble de chaînes de score total optimal.

En effet, la détection de régions dupliquées dans un ou plusieurs génomes ne se limite pas à la production d'une région, mais à la production de plusieurs régions, nécessitant d'allouer de façon raisonnée, les ancres détectées, à chacune de ces régions.

Dans un premier temps, nous allons préciser le formalisme de ReD. Nous verrons ensuite les contraintes qu'une chaîne et qu'un ensemble de chaînes doivent respecter pour être considérés comme **cohérents** et ainsi représenter des couples de régions dupliquées. Nous aborderons ensuite le nouveau critère d'optimisation utilisé dans la reconstruction des chaînes. Ce critère est global à l'ensemble des chaînes, contrairement aux méthodes existantes qui utilisent un critère local sur les chaînes. Et nous verrons comment satisfaire ce nouveau critère et les contraintes de cohérence à l'aide de la théorie des flots.

3.1 Un modèle à base de graphe

3.1.1 Le graphe de relation

ReD fait partie des méthodes de chaînage, le formalisme de ce type de méthode est décrit dans la section 2.4.1 page 25. En conséquence, je décris ici uniquement les choix faits par ReD dans son formalisme et j'invite le lecteur à retourner à cette section pour plus d'explications.

ReD est capable d'utiliser des ancres protéiques ou nucléiques. Au niveau de la notion d'*ancres chaînables*, il est possible de choisir entre la relation $\prec_\varnothing$ et $\prec_2$ associées à une contrainte de distance qui dépend du génome :

$$dist(a_i^g, a_j^g) < distMax^g \tag{3.1}$$

Les génomes n'ayant pas une distance inter-gène moyenne identique (certains sont plus compacts que d'autres), le choix d'une distance maximum qui dépend du génome permet d'augmenter la robustesse de la méthode. De plus laisser le choix entre $\prec_\varnothing$ et $\prec_2$ permet d'utiliser ReD de façon optimum selon l'objectif visé. Nous avons vu que la relation $\prec_2$ très permissive, permet cependant de construire des chaînes constituées d'ancres non colinéaires. Si l'objectif est de reconstruire des régions colinéaires, comme dans la recherche de duplications en tandem, il vaut mieux utiliser $\prec_\varnothing$.

Toujours dans l'optique de pouvoir s'adapter à différentes situations, ReD offre le choix entre plusieurs fonctions de score :

- Pour les sommets : S_{cte}, S_{evalue} ou S_{score} (voir remarque 14 page 35);
- Pour les arcs : S_{indel} qui favorise les insertions ou les délétions, S_{sub} qui favorise les substitutions entre deux ancres, ou S_{neutre} qui ne favorise ni l'une ni l'autre (voir remarque 15 page 37).

avec une configuration par défaut :

- $\prec_\varnothing$, S_{score} et S_{sub}, pour la recherche de duplications en tandem ;
- et $\prec_2$, S_{score} et S_{neutre}, pour la recherche de grandes régions convervées.

Remarque 20 *La relation d'ordre $I < I'$ impose que deux ancres chaînables ne se chevauchent pas. En pratique, cela peut empêcher de chaîner des ancres proches et pertinentes mais un peu trop longues. Nous avons choisi de contourner le problème en réduisant chaque intervalle à son point central : $\frac{fin-deb}{2}$. Cette heuristique a l'avantage de la simplicité et de résoudre le problème de façon satisfaisante sans nécessiter de paramétrage supplémentaire.De fait, le chevauchement des intervalles dans la construction d'une chaîne est, en soi, une problématique (Uricaru et al., 2010).*

3.1.2 Cohérence et validité des chaînes

Nous avons vu dans le chapitre 2, que le graphe de relation R permet de représenter l'ensemble des solutions et qu'un chemin dans ce graphe, appelé une *chaîne*, représente deux régions dupliquées. Cependant tous les chemins de R ne sont pas des régions dupliquées pertinentes. Alors comment sélectionner ces chemins pour créer des chaînes pertinentes ?

Nous avons vu (voir figure 2.4 page 26), qu'il était difficile de justifier de la qualité des chaînes en l'absence de jeu de données où la vérité est connue. Pour cela nous allons définir des critères garants d'une qualité minimale qu'un chemin doit respecter pour être considéré comme une chaîne valide (tout comme ADHoRe calcule un coefficient de détermination pour construire ses chaînes, voir la définition 18 page 39).

En plus de ces conditions, nous allons imposer un ensemble de contraintes qui devront être vérifiées par l'ensemble des chaînes et qui devront être respectées lorsqu'elle décrivent une histoire évolutive cohérente. C'est-à-dire qu'une succession de mutations

doit pouvoir être à l'origine de cet ensemble de chaînes. Les outils existants cherchent parfois à satisfaire certaines de ces contraintes mais sans l'exprimer clairement ou de façon assez brutale, ne garantissant pas de façon stricte qu'elles soient satisfaites (voir l'analyse des résultats du tableau 4.3 page 89).

Nous verrons ensuite le cas particulier de la comparaison d'une séquence contre elle même, qui est soumise à d'autres contraintes de cohérence. Cette contrainte supplémentaire a pour but d'interdire qu'une sous-séquence du génome soit sélectionnée dans les deux régions représentées par une même chaîne.

3.1.2.1 Validité d'une chaîne

Tout comme pour les ancres, il est possible de définir un ensemble de propriétés destinées à capturer les chaînes qui sont les plus représentatives de paires de régions dupliquées :

- $minAncre$ le nombre minimum d'ancres d'une chaîne
- $minScore$ le score minimum d'une chaîne
- $minCorr$ le coefficient de corrélation minimum d'une chaîne

Ainsi, une chaîne c valide doit respecter une série de conditions qui visent à éliminer les chaînes trop petites (contenant peu d'ancres) ou de trop mauvaise qualité (de score faible ou avec un coefficient de corrélation trop faible) :

$$\begin{array}{rcl} c.s & \geq & minScore \\ c.n & \geq & minAncre \\ c.corr & \geq & minCorr \end{array} \tag{3.2}$$

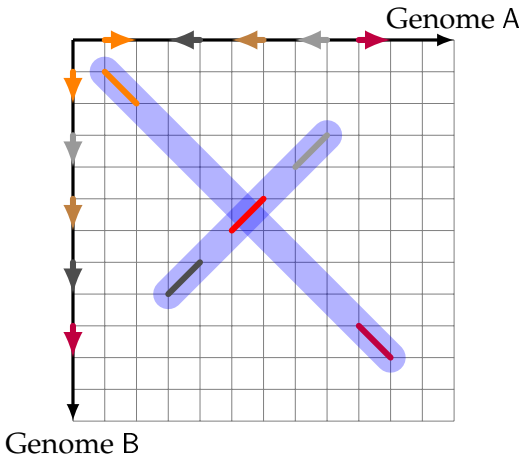
Ces conditions sont appelées par la suite conditions de *validité*.

Définition 23 (Chaîne valide) *On appelle* chaîne valide, *une chaîne qui respecte les trois conditions de l'équation 3.2.*

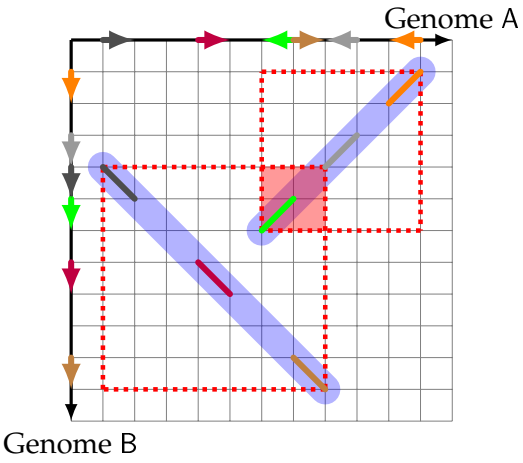
3.1.2.2 Cohérence d'un ensemble de chaînes

Le problème de recherche de régions dupliquées entre deux génomes pourrait alors se ramener à la production d'un ensemble $\mathcal{C}$ de chaînes tirées du graphe de relation R, qui respectent les conditions précédentes. Il est toutefois nécessaire de rajouter un ensemble de conditions liant entre elles les différentes chaînes apparaissant dans l'ensemble de chaînes $\mathcal{C}$ (voir figure 3.1 page ci-contre).

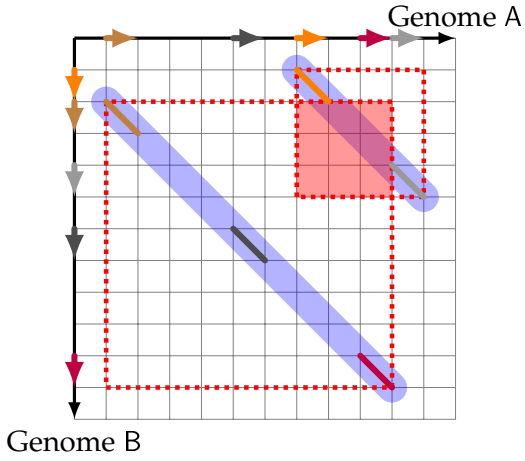
Nous précisons ici le postulat de biologie évolutive que nous utilisons pour formuler les conditions de cohérence d'un ensemble de chaînes. Nous rappelons tout d'abord qu'une relation d'homologie décrit une origine ancestrale commune.



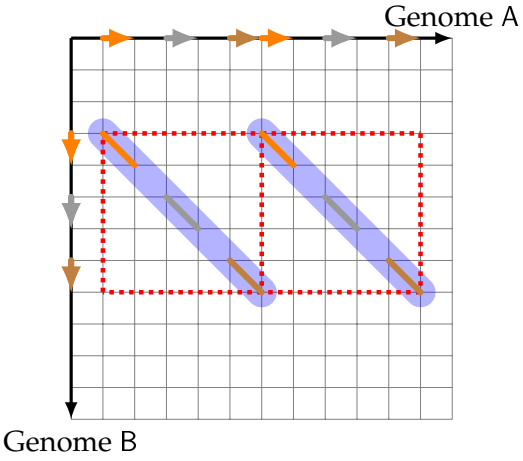
(a) Exemple de deux chaînes qui utilisent une même ancre.



(b) Exemple de deux chaînes qui se chevauchent sur les deux génomes, avec une ancre.



(c) Exemple de deux chaînes qui se chevauchent sur les deux génomes, qui montre qu'il n'est pas nécessaire d'avoir une ancre dans le "rectangle d'une chaîne" pour créer deux chaînes qui se chevauchent.



(d) Exemple qui représente une spéciation et une duplication en tandem sur le génome A.

FIGURE 3.1: Trois exemples d'ensembles de chaînes non cohérentes ((a), (b) et (c)) et un exemple cohérent (d). Les contraintes violées sont visibles en rouge.

Postulat 1 *Dans un ensemble de régions homologues, chaque région ne peut posséder qu'un lien de parenté, celui qui résulte de l'évènement (spéciation, duplication) qui a mené à son apparition.*

Dans notre formalisme, les relations d'homologie sont représentées par les chaînes et doivent donc avoir les mêmes propriétés. Dans ce formalisme, la propriété ci-dessus s'énonce de la manière suivante :

Soit deux chaînes c_i et $c_j \in \mathcal{C}$ avec $c_i \neq c_j$, ces deux chaînes ne peuvent relier une même paire de séquences génomiques s^A et s^B, car cela reviendrait à établir deux liens de parenté différents pour ces deux séquences.

Propriété 1 (Cohérence de chaînes) *Plus précisément, $\forall c_i, c_j \in \mathcal{C}$, $c_i \neq c_j$ et c_i^g l'intervalle de c_i sur le génome g, ces chaînes ne peuvent pas se chevaucher à la fois sur le génome A et sur le génome B. La duplication d'une région implique en effet la création d'une région qui n'est l'image que d'une unique origine.*

$$(c_i^A \cap c_j^A) = \varnothing \;\; ou \;\; (c_i^B \cap c_j^B) = \varnothing \tag{3.3}$$

En particulier, dans un ensemble cohérent de chaînes, une ancre ne peut participer qu'à une seule chaîne.

$$\forall c_i, c_j \in \mathcal{C}, \textbf{ Si } a \in c_i \text{ et } c_i \neq c_j \textbf{ alors } a \notin c_j \tag{3.4}$$

Ces deux conditions supplémentaires montrent que l'ensemble de chaînes qu'il faut produire n'est pas un ensemble de chaînes indépendantes mais qu'il forme un tout. Pourtant nous avons vu que les méthodes créent les chaînes indépendamment les unes des autres et le risque de créer un ensemble de chaînes incohérent est présent. Conscient de ce problème, des méthodes comme ADHoRe, SyMAP et OSfinder, ont décidé de supprimer les ancres situées dans le rectangle défini par les extrémités d'une chaîne. Cependant, comme le montre la figure 3.1(c) page précédente, cette stratégie n'est pas suffisante.

Il pourrait sembler au lecteur qu'il est incohérent d'interdire ici les chevauchements entre deux chaînes de $\mathcal{C}$ alors que nous avons, dans la remarque 20 page 53, autorisé de fait le chevauchement partiel de deux ancres successives dans une chaîne en les réduisant à deus points. Il s'agit en fait de situations très différentes, apparaissant soit à l'intérieur d'une chaîne soit entre deux chaînes. Si deux ancres successives d'une chaîne se chevauchent, cela indique qu'il est très simplement possible d'extraire localement de ces ancres des sous-ancres chaînables, non chevauchantes (cette stratégie est utilisée dans (Delcher *et al.*, 1999)). La chaîne ainsi obtenue définit toujours les mêmes intervalles sur A et B et a un score éventuellement affaibli. Si plusieurs chaînes se chevauchent, il devient impossible de considérer simultanément les duplications

de séquence que ces chaînes représentent. De délicats arbitrages entre ces différentes chaînes sont alors nécessaires. Dans notre algorithme, ces arbitrages sont résolus sur la base d'un score global et des propriétés que doivent satisfaire différentes chaînes qui représentent des évènements de duplication (en tandem).

3.1.3 Cohérence dans le cas d'un génome contre lui-même

Comparer un génome à lui-même pour rechercher des duplications internes, impose de vérifier qu'une région du génome ne participe pas plusieurs fois à une même chaîne : à la fois en tant que séquence du génome A et en tant que séquence du génome B. Ainsi une propriété supplémentaire est nécessaire.

Propriété 2 (Cohérence des chaînes de duplications) *Dans le cas de la comparaison d'un génome avec lui-même, $\forall c \in \mathcal{C}$, les deux intervalles c^g définis par c ne peuvent pas se chevaucher car le résultat d'une duplication ne peut chevaucher son origine (voir figure 3.2 page suivante) :*

$$c^A \cap c^B = \varnothing \tag{3.5}$$

Cette propriété considère comme incohérentes les chaînes qui couvriraient des régions dupliquées trois fois ou plus consécutivement (duplication en tandem). Cela ne rend pas pour autant problématique la détection des régions dupliquées en tandem. Mais ces régions seront détectables par plusieurs chaînes qui pourront chacune respecter les conditions de cohérence (voir Figure 3.2(d) page suivante).

De plus cette propriété, énoncée au niveau des chaînes, permet de construire une condition nécessaire au niveau des ancres : deux ancres qui mettent en relation des régions ne peuvent pas appartenir à une même chaîne si la simple chaîne c formée de ces deux ancres définit deux régions qui se chevauchent.

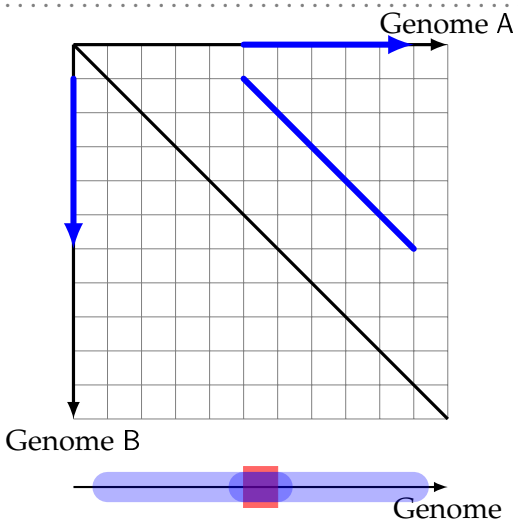
Soit a_i et $a_j \in \mathcal{A}$ deux ancres satisfaisant $a_i \prec a_j$ (définition 9 page 30) et soit c la chaîne formée des deux ancres a_i et a_j, alors :

$$c^{\mathsf{A}} \cap c^{\mathsf{B}} = \varnothing. \tag{3.6}$$

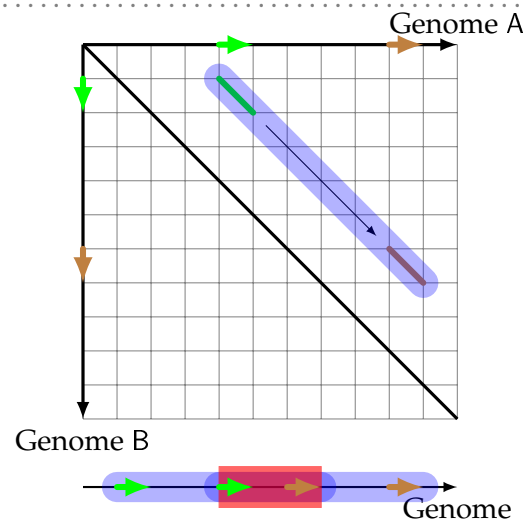
Dans le cas de reconstruction de duplications segmentales ou de duplications en tandem, cette condition sera donc directement mobilisée en supprimant dans R tout arc reliant deux ancres a_i et a_j violant cette propriété. Cette condition est nécessaire mais non suffisante (voir figure 3.2(b) page suivante et 3.2(c) page suivante).

Il est aussi possible d'imposer (condition nécessaire) que les ancres vérifient également cette propriété :

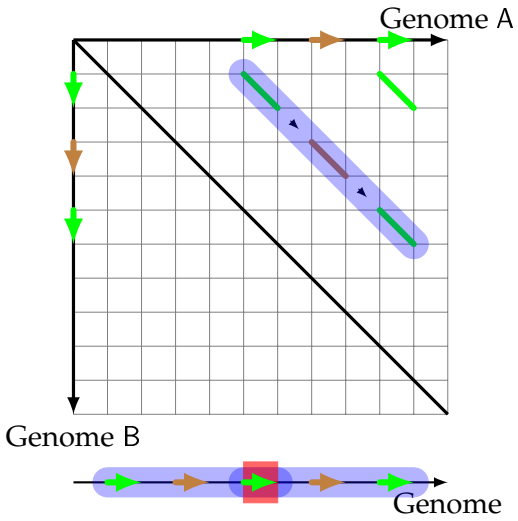
$$a^{\mathsf{A}} \cap a^{\mathsf{B}} = \varnothing.$$



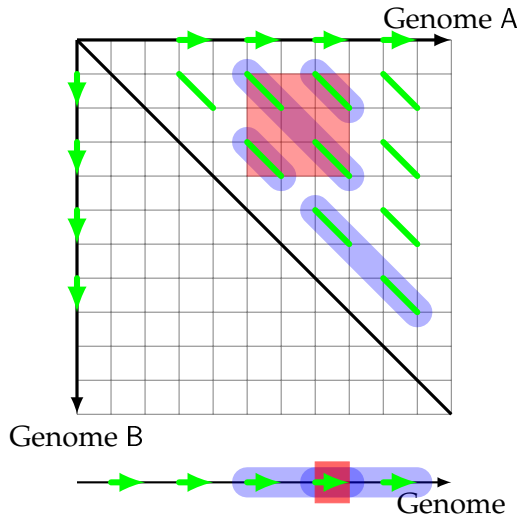
(a) Exemple d'une ancre qui viole la contrainte de chevauchement.



(b) Exemple d'un arc qui crée une situation où la contrainte de chevauchement sera forcément violée. Cet arc est inutile, il sera supprimé à la création du graphe.



(c) Exemple d'une chaîne non valide.



(d) Exemple qui montre que contraintes de cohérence des chaînes sont particulièrement utiles dans la détection des unités dupliquées en tandem.

FIGURE 3.2: Exemple de contraintes de chevauchements violées dans le cas particulier où un génome est comparé à lui-même. La diagonale noire représente la similarité des deux génomes sur toute leur longueur, elle est aussi l'axe de symétrie de la comparaison. Pour ne pas surcharger la figure, seule la symétrie supérieure est visible. De plus, sous chaque dotplot est représentée la projection du dotplot sur le génome comparé, ce qui permet de voir les zones d'incohérence en rouge.

Les logiciels d'alignements locaux ne font pas attention à cette contrainte et peuvent créer des ancres incohérentes (voir figure 3.2(a) page ci-contre). par (Benson, 1995) qui propose un algorithme exact de recherche d'un alignement non chevauchant en $O(n^2 \log^2 n)$, quadratique en espace. Cet algorithme, qui ne produit qu'un alignement, est inapplicable en l'état à l'échelle d'un chromosome. Mais ces ancres sont un vrai problème. Elles sont inutilisables, ce qui représente une perte d'information importante. Une autre solution consiste à les couper en deux ancres cohérentes (ou plus), cependant comment savoir où il faut couper : au milieu, au début, à la fin , ... toutes les solutions sont *a priori* équivalentes. Il vaut mieux intervenir en amont, soit pendant la création des ancres. C'est cette dernière solution qui a été choisie, en améliorant un logiciel d'alignement développé en interne (Glint, (Faraut et Courcelle, 2011)).

Remarque 21 *Les propriétés de cohérence d'un ensemble de chaînes sont particulièrement importantes dans le cadre de la recherche de duplications en tandem, comme nous le montre les figures 3.1(d) page 55 et 3.2 page ci-contre, du fait que les chaînes sont, par définition, très proches les unes des autres et proches de la diagonale du dotplot.*

3.1.4 Le problème cible

Une fois le graphe de relation R construit, le problème de recherche de duplications entre A et B se ramène donc à la recherche d'un ensemble cohérent de chaînes valides. La taille de cet ensemble est inconnue *a priori*.

Définition 24 (Ensemble de chaînes cohérent) *On appelle ensemble de chaînes cohérent, un ensemble de chaînes dont toutes les chaînes sont* valides *(voir définition 23 page 54), respectent la propriété 1 page 56 de cohérence des chaînes et la contrainte 3.1 page 52 de distance maximum entre ancres successives.*

Dans le cas ou un génome est comparé à lui-même, l'ensemble de chaînes doit de plus respecter la propriété 2 page 57 de cohérence des chaînes de duplications.

La contrainte (3.1 page 52), qui exige une distance maximale entre *ancres chaînables*, est immédiate à prendre en compte dans le graphe R et est donc satisfaite dans la majorité des outils existants, comme DAGchainer (Haas *et al.*, 2004). Les autres contraintes sont plus délicates à traiter. L'approche suivie par les méthodes existantes (excepté EGM) consiste à résoudre le problème de la construction d'un ensemble de chaînes en recherchant une chaîne optimale de façon répétée, de façon gloutonne et en exploitant les propriétés restantes pendant l'algorithme, voir *a posteriori*.

Cette approche gloutonne ne garantit naturellement pas de construire un ensemble de chaînes dont le score total, défini par la somme des scores des chaînes qu'il contient, est maximum car les chaînes ne sont pas indépendantes les unes des autres et la maximisation du score d'une chaîne peut se faire au détriment du score des autres chaînes de l'ensemble, comme le montre un simple exemple :

Exemple 1 *Soient quatre chaînes c_1, c_2, c_3 et c_4 de score $S(c_1) = 100$, $S(c_2) = -98$, $S(c_3) = 99$ et $S(c_4) = 1$. Si c_1, c_2, c_3 sont elles même chaînables, alors il existe une chaîne c de score $S(c) = 101$, issue de la concaténation de c_1, c_2 et c_3. L'approche gloutonne qui maximise le score de la chaîne courante donnera : $\mathcal{C} = \{c, c_4\}$ avec $S(\mathcal{C}) = 102$. Il existe naturellement un ensemble de chaînes de meilleur score $\mathcal{C}' = \{c_1, c_3\}$ avec un score $S(\mathcal{C}') = 199 \approx 2 \times S(\mathcal{C})$. L'approche gloutonne a de plus tendance à construire des chaînes qui peuvent contenir des sous segments (ici c_2) de très mauvaise qualité.*

Le problème que nous souhaitons résoudre peut donc s'exprimer formellement comme suit :

Problème 1 *Étant donné un entier d et un graphe de relation R construit à partir de la comparaison d'un ou deux génomes, trouver un ensemble de d chaînes cohérent (définition 24 page précédente) et dont la somme des scores des chaînes qu'il contient est maximal (parmi tous les ensembles de d chaînes cohérents).*

Nous n'avons pas cherché à déterminer la complexité de ce problème, mais la prise en compte simultanée des différentes contraintes et de la recherche de l'optimalité dans une situation naturellement combinatoire ne simplifie pas sa résolution informatique. L'algorithme que je propose dans la suite résout en fait un problème plus simple, consistant à déterminer un ensemble de chaînes cohérent et de score total optimal mais dont la cardinalité d n'est pas fixée `a priori` mais imposée essentiellement par la propriété de cohérence.

Pour résoudre efficacement ce problème fortement combinatoire (le simple nombre de chemins dans le graphe de relations peut croître de manière exponentielle avec la taille du graphe), j'exploite une relation forte entre flots dans un réseau (où toutes les capacités sont de 1) et ensembles de chemins dans R.

Par la suite, il sera montré que la recherche d'un ensemble de chemins ne partageant aucun sommet et de score maximum dans le graphe de relation peut se ramener à la recherche d'un flot de coût minimum dans un réseau dérivé de R. Avant de présenter cette solution, nous commençons par présenter quelques rappels minimaux sur la théorie des flots et ses notions essentielles (réseau, flot, valeur d'un flot, coût d'un flot...), suffisant pour notre propos. Le lecteur curieux pourra se reporter aux nombreuses monographies sur le sujet dont (Ahuja *et al.*, 1993; Gondran et Minoux, 2009; Jungnickel, 2007).

3.2 Rappel sur la théorie des flots

En général, les premiers travaux sur les flots sont attribués à Gustav Kirchhoff et à ses travaux dans le domaine de la physique sur les courants continus qui parcourent un circuit électrique. C'est d'ailleurs un très bon exemple pour illustrer la notion de flot (figure 3.3 page suivante). A partir de cette notion physique du flot, un modèle

général s'est vu développé et appliqué à un très grand nombre d'autres problèmes concrets : problèmes de transport (routiers, aériens, ...), structuration et dimensionnement optimaux des réseaux de communication, problème de gestion des stocks, d'ordonnancement, d'affectation et dans les mathématiques combinatoires (Ahuja *et al.*, 1993; Gondran et Minoux, 2009).

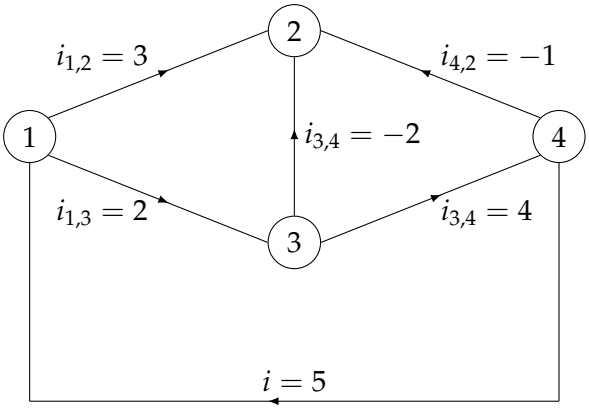


FIGURE 3.3: Exemple d'un flot dans un circuit électrique soumis à la loi des nœuds (ou sommets) : la somme des intensités des courants entrants dans un nœud est égale à la somme des intensités des courants qui sortent du nœud.

Dans tous ces problèmes, et dans beaucoup d'autres, il est question de déplacer des entités (électricité, objets, messages, personnes, ...) d'un point à un autre à travers un réseau et de le faire le plus efficacement possible, selon des critères propres à chacun des problèmes. Nous allons voir comment le problème se modélise sous forme mathématique.

3.2.1 Définition et propriétés

La théorie des flots s'articule autour de trois définitions :

1. la *capacité* : quantité maximum d'éléments qui peut être transportée directement entre deux sommets ;
2. le *réseau de transport* : ensemble des connections de transports directes entre sommets qui permet de relier globalement un point d'origine à une destination finale ;
3. le *flot* : quantité effective d'éléments transportée directement entre deux sommets.

Définition 25 (Capacité) *Soit $G = (S, A)$ un graphe orienté dans lequel S et A représentent respectivement l'ensemble des sommets et l'ensemble des arcs du graphe. Sur chaque arc $a = (i, j) \in A$, on définit une application, appelée capacité, $C : A \to \mathbb{N}$. Par la suite, nous noterons c_{ij} la capacité de l'arc (i, j).*

Définition 26 (Réseau de transport) *Un réseau de transport est un triplet $T = (S, A, C)$ dans lequel S et A représentent un graphe orienté, et C les capacités des arcs.*

Il existe dans S deux sommets particuliers **s** *et* **p** *(appelés respectivement source et puits) tels que pour tout sommet $i \in S$ il existe un chemin de* **s** *vers i et un chemin de i vers* **p**. *Le degré entrant (resp. sortant) de* **s** *(resp.* **p**) *est nul (voir figure 3.4(a) page ci-contre).*

Définition 27 (Flot) *Soit un réseau de transport $T = (S, A, C)$, un flot f sur T est une application $f : A \to \mathbb{N}$ (nous noterons f_{ij} la valeur du flot sur l'arc (i, j)) vérifiant :*
1. *La contrainte de capacité : $\forall (i, j) \in A, \quad 0 \leq f_{ij} \leq c_{ij}$*
2. *La contrainte de conservation (loi des nœuds de Kirchhoff) :*

$$\forall i \in S, i \notin \{\mathbf{s}, \mathbf{p}\} \qquad \sum_{(j,i)\in A} f_{ji} = \sum_{(i,j)\in A} f_{ij}$$

Dans la suite du document, si le réseau de transport sur lequel nous raisonnons a déjà été défini, nous parlerons d'un flot f sans nécessairement préciser le réseau de transport lequel le flot est défini.

Définition 28 (Débit) *Soit T un réseau de transport et un flot f sur T, on appelle* débit **d** $\in \mathbb{N}$, *la quantité de flot qui va de* **s** *à* **p**.

La contrainte de conservation du flot, permet de déduire un théorème :

Théorème 1 ((Ahuja *et al.*, 1993; Gondran et Minoux, 2009)) *La quantité du flot qui sort de la source* **s** *est égale à la quantité de flot qui entre dans le puits* **p**.

qui permet de calculer la valeur du débit **d** :

$$\mathbf{d} = \sum_{(\mathbf{s},j)\in A} f_{\mathbf{s}j} = \sum_{(j,\mathbf{p})\in A} f_{j\mathbf{p}}$$

3.2.2 A la recherche du flot maximum

Le problème de flot maximum consiste, étant donné un réseau de transport T, à trouver un flot maximisant **d**. Le débit de ce flot de débit maximum sera noté *debitmax*.

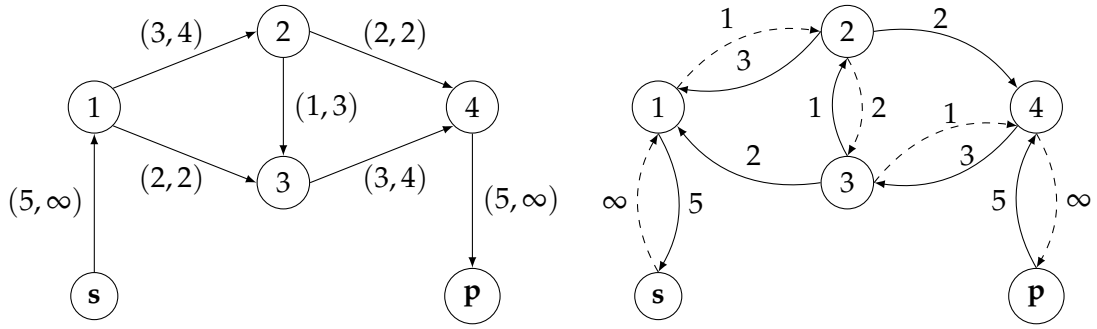
Remarque 22 *Dans un graphe, il peut exister plusieurs flots avec le même débit. Ici, il suffit de trouver un flot de débit maximum, et non pas tous les flots de débit maximum.*

La résolution de ce problème utilise les deux notions de *graphe d'écart* et de *chemin augmentant* (voir figure 3.4(b) page suivante).

Définition 29 (Graphe d'écart) *Soit un réseau de transport $T = (S, A, C)$ et un flot f sur T, le graphe d'écart associé $E(T, f) = (S, A^e, C^e)$ avec A^e et C^e respectivement l'ensemble des arcs et leurs capacités. On note $c^e_{ij} = C^e((i, j))$ la capacité de l'arc (i, j) dans $E(T, f)$. A^e, C^e sont construits à l'aide des deux règles suivantes :*

1. $(i, j) \in A^e$ *et* $c^e_{ij} = c_{ij} - f_{ij}$ ***si*** $(i, j) \in T$ *et* $f_{ij} < c_{ij}$.
2. $(j, i) \in A^e$ *et* $c^e_{ji} = f_{ij}$ ***si*** $(i, j) \in T$ *et* $f_{ij} > 0$.

Le graphe d'écart défini par un flot constitue une représentation des capacités qui sont encore disponibles sachant les capacités initiales et le flot lui-même. Un arc créé par la règle 1 porte une capacité réduite du fait du flot existant, un arc créé par la règle 2 représente le fait qu'il est aussi possible de diminuer le flot, créant ainsi une capacité positive sur l'arc inverse.



(a) Représente le graphe de transport T avec un flot f sur T d'un débit $\mathbf{d} = 5$. Le premier chiffre entre parenthèse correspond au flot qui passe sur cet arc ($f_{34} = 3$) et le deuxième la capacité ($c_{34} = 4$).

(b) Représente le graphe d'écart $E(T, f)$ construit à partir de f. Les arcs en pointillés forment un chemin augmentant qui se sature avec un flot de débit $d' = 1$.

FIGURE 3.4: Exemple d'un réseau de transport T avec un flot f sur T et le graphe d'écart $E(T, f)$.

Définition 30 (Chemin augmentant) *Dans un graphe d'écart $E(T, f)$ on appelle un chemin augmentant, un chemin qui va de* **s** *à* **p**.

Les capacités apparaissant dans le graphe d'écart étant toutes positives, un chemin augmentant indique une augmentation possible de débit. Il s'agit en fait d'une condition nécessaire (Ahuja *et al.*, 1993; Gondran et Minoux, 2009).

Théorème 2 ((Ahuja *et al.*, 1993; Gondran et Minoux, 2009)) *Soit un réseau de transport T, un flot f sur T est maximum si et seulement si le graphe d'écart $E(T, f)$ ne possède aucun chemin augmentant.*

Il existe plusieurs algorithmes qui résolvent ce problème de recherche d'un flot maximum. L'algorithme de Ford et Fulkerson (1962) est le premier à avoir résolu le problème avec une complexité $O(AS.debitmax)$. L'algorithme se décompose en 4 étapes, en partant d'un flot initial nul sur T (Soit T, $\forall i, j \in S\ f_{ij} = 0$) :

1. trouver un chemin augmentant μ quelconque, dans $E(T, f)$;
2. trouver le débit $\mathbf{d}'$ du flot qui *sature* μ (égal à la plus petite capacité du chemin augmentant), dans $E(T, f)$;
3. mettre à jour le flot f sur T (voir algorithme 1 page suivante) ;
4. recommencer tant qu'il existe un chemin augmentant dans $E(T, f)$.

Définition 31 (Algorithme polynomial) *Un algorithme est dit polynomial si le nombre d'opération est borné par un polynôme qui dépend de la taille des données.*

Algorithme 1 : Mise à jour de f

Données :

- $T = (S, A, C)$ un réseau de transport
- f un flot sur T de débit $\mathbf{d}$
- Un chemin augmentant μ et le débit qui le sature $\mathbf{d}'$

Résultat :

- f le flot sur T mis à jour

début
  $\mathbf{d} = \mathbf{d} + \mathbf{d}'$
  pour chaque $(i, j) \in \mu$ **faire**
    si *il existe* $(i, j) \in T$ **alors** $f_{ij} = f_{ij} + \mathbf{d}'$
    sinon $f_{ji} = f_{ji} - \mathbf{d}'$
  return f
fin

Remarque 23 *L'algorithme de Ford-Fulkerson n'est pas un algorithme polynomial car sa complexité dépend de la valeur de debitmax (voir figure 3.5).*

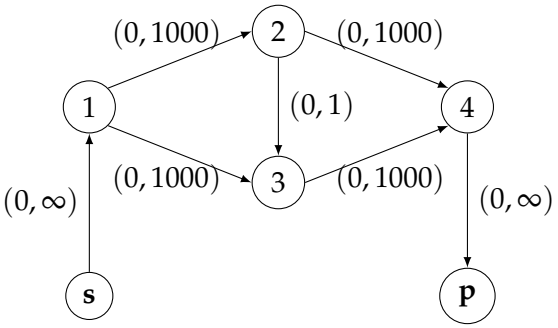


FIGURE 3.5: Exemple d'un réseau de transport T, où l'algorithme de Ford et Fulkerson (1962) n'est pas polynomial. Si l'algorithme utilise systématiquement un chemin augmentant dans $E(T, f)$ qui passe par la paire de sommets $\{2, 3\}$, le nombre d'itérations est proportionnel à la capacité c portée par les arcs $(1, 2), (1, 3, (2, 4), (3, 4)$ alors que la représentation de ces capacités est en $O(\log(c))$ en espace.

L'algorithme d'Edmonds et Karp (1972) en choisissant à chaque étape le chemin augmentant le plus court en nombre d'arcs, modifie la complexité de Ford-Fulkerson en $O(SA^2)$ et devient polynomial.

Mais le premier algorithme polynomial pour ce problème est en fait celui de Dinic (1970). L'histoire de cet algorithme publié dans une revue russe, et donc peu connu à l'époque dans le monde occidental, a été décrite en détail dans (Dinitz, 2006). Cet algorithme choisit de saturer tous les plus courts chemins à chaque étape et résout

le problème en $O(S^2A)$. L'implémentation de cet algorithme à l'aide d'un arbre dynamique (Sleator et Tarjan, 1983) améliore la complexité en $O(SA\ log\ S)$.

3.2.3 A la recherche d'un flot maximum de coût minimum

Le problème de flot de coût minimum est défini sur un *réseau de transport avec coût* :

Définition 32 (Coût unitaire) *Soit $T = (S, A, C)$ un réseau de transport, on définit une application pour chaque arc $(i, j) \in A$, appelée coût, $W : A \to \mathbb{R}$. Par la suite, nous noterons w_{ij} le coût de transport d'une unité de flot sur l'arc (i, j).*

Définition 33 (Réseau de transport avec coût) *Un réseau de transport avec coût est un quadruplet $T^c = (S, A, C, W)$ avec (S, A, C) un réseau de transport (définition 26 page 61) et W les coûts associés aux arcs.*

Soit T^c un graphe de transport avec coût, l'objectif est de trouver un flot de coût minimum parmi tous les flots de débit maximum.

Définition 34 (Coût d'un flot) *Soit un réseau T^c et un flot f sur T^c, on définit le coût du flot f comme $W(f) = \sum_{(i,j)\in A} w_{ij}.f_{ij}$.*

Ainsi soit $\mathbf{F}_{T^c}$, l'ensemble des flot de débit maximum sur un graphe de transport T^c donné, le problème de recherche qui nous intéresse consiste à trouver un flot :

$$f^* = arg \min_{f\in\mathbf{F}_{T^c}} W(f) \tag{3.7}$$

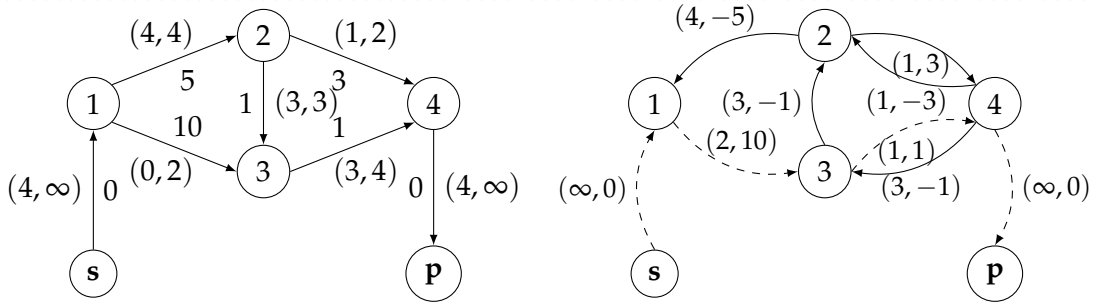
De nombreuses méthodes existent pour résoudre ce problème. Comme pour le flot maximum, la majorité de ces méthodes utilisent un graphe d'écart, qui tient compte des coûts du réseau de transport.

Définition 35 (Graphe d'écart avec coût) *Soit un réseau de transport $T^c = (S, A, C, W)$ avec coûts et f un flot sur T^c, le graphe d'écart avec coût associé $E(T^c, f) = (S, A^e, C^e, W^e)$ avec A^e, C^e et W^e respectivement l'ensemble des arcs, les capacités et les coûts associés. On note $c^e_{ij} = C^e((i, j))$ et $w^e_{ij} = W^e((i, j))$ respectivement la capacité et le coût de l'arc (i, j) dans $E(T^c, f)$ et A^e, C^e W^e sont construits à l'aide des deux règles suivantes :*

1. *$(i, j) \in A^e$, $c^e_{ij} = c_{ij} - f_{ij}$ et $w^e_{ij} = w_{ij}$ **si** $(i, j) \in A$ et $f_{ij} < c_{ij}$.*
2. *$(j, i) \in A^e$, $c^e_{ji} = f_{ij}$ et $w^e_{ji} = -w_{ij}$ **si** $(i, j) \in A$ et $f_{ij} > 0$.*

La transformation des coûts utilisée dans la règle 2 de la définition précédente découle du fait qu'augmenter le flot sur l'arc (j, i) revient en fait à diminuer le flot sur l'arc inverse (i, j) et donc le coût du flot.

Théorème 3 ((Ahuja *et al.*, 1993; Gondran et Minoux, 2009)) *Soit un réseau de transport avec coûts T^c, un flot f sur T^c est de coût minimum si et seulement si il n'existe pas de circuit de coût négatif dans le graphe d'écart $E(T^c, f)$.*



(a) Représente le graphe de transport T^c avec un flot f sur T^c de débit $\mathbf{d} = 4$ et de coût minimum $W(f) = 29$. Par rapport à un graphe de transport classique, le chiffre supplémentaire utilisé pour étiqueter les arcs correspond au coût de l'arc : $w_{34} = 1$.

(b) représente le graphe d'écart avec coût $E(T^c, f)$ construit à partir de f sur T^c. Le premier chiffre entre parenthèse correspond à la capacité de l'arc ($c^e_{34} = 1$) et le deuxième à son coût ($w^e_{43} = -1$). Les arcs en pointillés indiquent un chemin augmentant de coût minimum qui sature avec un flot de débit $\mathbf{d}' = \mathbf{1}$.

FIGURE 3.6: Exemple d'un réseau de transport avec coût T^c avec un flot f sur T^c et le graphe d'écart $E(T^c, f)$.

Les méthodes de flot maximum de coût minimum les plus optimisées sont celles qui utilisent le principe de *Canceling Negative Cycles* : l'objectif est de saturer les circuits de coûts négatifs dans le graphe d'écart. Ce principe a permis d'écrire des algorithmes polynomiaux, dont la complexité s'exprime seulement en nombre d'arcs et de sommets (Tardos, 1985; Goldberg et Tarjan, 1989; Wallacher et Zimmermann, 1992).

Je décris ci-dessous l'algorithme 2 page suivante de Busacker et Gowen (1961), qui résout le problème de flot maximum de coût minimum par une procédure itérative sur le débit. Cet algorithme itératif, capable de fournir un série de flots de coûts minimum et débits toujours croissants, sera en effet nécessaire pour résoudre notre problème de reconstruction de grandes régions dupliquées (voir section 3.3.2 page 70). Cette méthode est illustrée et expliquée à travers les figures 3.6 et 3.7 page ci-contre.

Théorème 4 ((Busacker et Gowen, 1961)) *Si le réseau de transport initial T^c ne contient pas de circuit de coût négatif, l'algorithme de Busacker et Gowen fourni à chaque itération, un flot sur T^c de coût minimum et de débit croissant à chaque itération.*

Remarque 24 *Dans le cas général, l'algorithme de Busacker et Gowen (BG) n'est pas polynomial. En effet sa complexité dépend de la valeur du débit du flot maximum, tout comme l'algorithme de Ford-Fulkerson. Nous verrons que dans notre cas, cela n'est pas problématique.*

Algorithme 2 : Algorithme de Busacker et Gowen

Données :

- $T^c = (S, A, C, W)$ un réseau de transport
- f un flot sur T^c de débit $\mathbf{d} = 0$

Résultat :

- f un flot maximum de coût minimum sur T^c

début

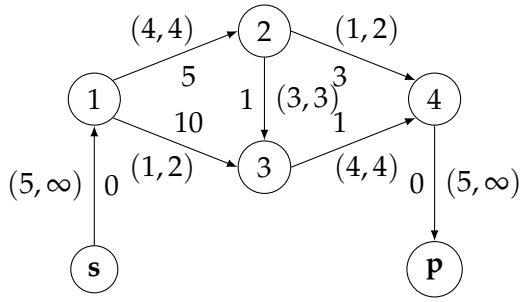
  tant que *il existe un chemin augmentant μ de coût minimum dans $E(T^c, f)$* **faire**

    Trouver le flot de débit $\mathbf{d}'$ qui sature μ

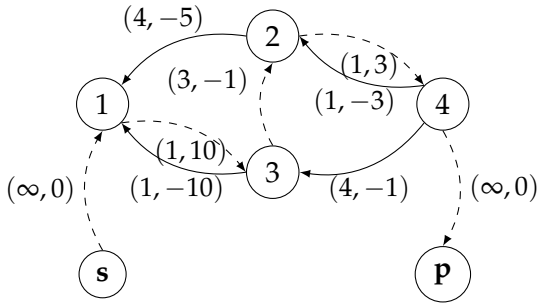
    Mettre à jour le flot f (algorithme 1 page 64)

  return f

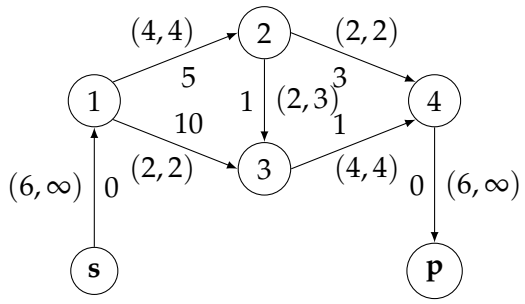
fin



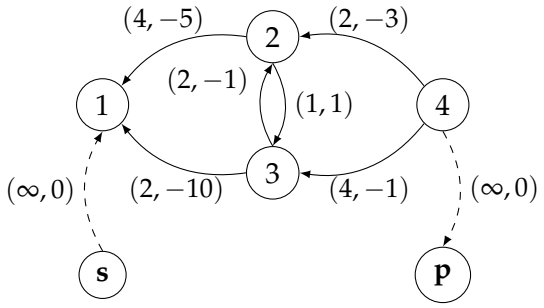
(a) Graphe de transport obtenu après saturation du chemin augmentant de coût minimum de l'étape 3.6(b).



(b) Graphe d'écart du graphe de transport de l'étape (a).



(c) Graphe de transport obtenu après saturation du chemin augmentant de coût minimum de l'étape (b).



(d) Graphe d'écart du graphe de transport de l'étape (c). Il n'a plus de chemin augmentant, l'algorithme s'arrête.

FIGURE 3.7: Exécution de l'algorithme de Busacker et Gowen à partir du flot de coût minimum de la figure 3.6 page précédente. Le flot maximum de coût minimum f est de débit $\mathbf{d} = 6$ et de coût $W(f) = 52$.

3.3 Résolution par recherche d'un flot

3.3.1 Du graphe de relation au réseau de relation

Nous allons voir ici, comment transformer le graphe de relation R introduit dans la section 3.1.1 page 52 en un réseau de transport avec coût T^R et prouver que la recherche d'un ensemble de chaînes ne partageant aucun ancre et de score optimum dans R est équivalent à la recherche d'un flot de coût minimum sur T^R.

3.3.1.1 Transformation

Avant toute chose, tous les scores du graphe R sont transformés en un coût de valeur opposée afin de ramener le problème de maximisation de score à un problème de minimisation de coût.

Les sommets du graphe de relation étant porteurs d'une information de coût, nous transformons, de façon classique, tout sommet a_i de R en une paire de sommets a_i et a'_i reliés par un arc allant de a_i à a'_i. Le coût $w_{ii'}$ associé à cet arc est égal à $-S(a_i)$.

Enfin, les sommets **s** et **p** sont ajoutés. **s** est relié à tous les sommets a_i de R par un arc de coût $w_{\mathbf{s}i} = 0$. De même, tous les sommets $a_{i'}$ ajoutés à l'étape précédente sont reliés à **p** par un arc de coût $w_{i'\mathbf{p}} = 0$. Ce qui permet de commencer et de finir un chemin sur n'importe quel sommet (ancre). Pour finir, toutes les capacités sont fixées à 1.

Un exemple d'une telle transformation est présenté dans le figure 3.8 page suivante.

Remarque 25 *Dans le cas précis où $\forall(i,j), c_{ij} = 1$, la présence d'un arc (i,j) dans le réseau de relation est suffisant pour indiquer qu'un flot de 1 peut aller du sommet i vers j.*

3.3.1.2 Équivalence entre chaînes et flot

Théorème 5 *Dans T^R, tout flot entier f sur T_R, de débit* **d**, *et de coût $W(f)$ définit un ensemble de* **d** *chaînes (chemins) $\mathcal{C}$ ne partageant aucun sommet (propriété (3.4 page 56)) dans R et de score $-W(f)$ et réciproquement.*

Preuve : Soit un flot entier de débit **d** sur T^R, dont tous les arcs ont une capacité de 1. Chaque sommet a_i de R étant représenté par un arc dans T^R, chaque sommet de T^R (en dehors de **s** et **p**) a soit un degré entrant égal à 1, soit un degré sortant égal à 1. Il reçoit donc une unité de flot d'un seul de ses arcs entrant et la diffuse sur un seul des arcs sortant. Globalement, le flot de valeur **d** est donc formé d'un ensemble de **d** chemins $(\mathbf{s}, a_i, a_{i'}, \ldots, \mathbf{p})$ ne partageant aucun arc entre eux (chemin dit simple). Les sommets a_i de R étant représentés par un arc dans T^R, après suppression des sommets $a_{i'}$, ces **d** chemins définissent **d** chaînes ne partageant aucune ancre entre eux dans R.

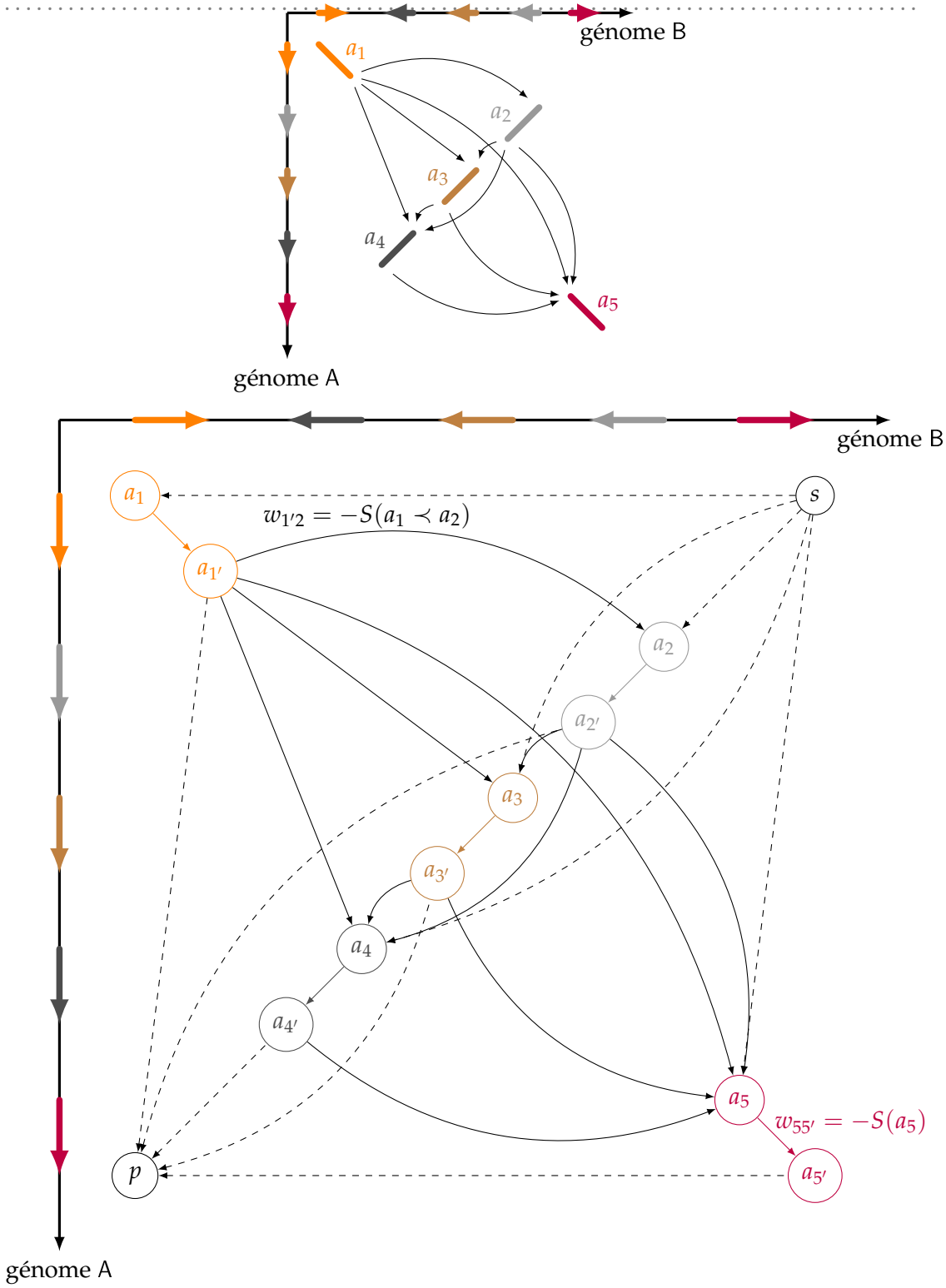


FIGURE 3.8: Illustration de la transformation d'un graphe de relation en réseau de relation. Pour ne pas surcharger le graphe, les capacités et le flot associé ne sont pas visibles (Toutes les capacités sont à 1 et flot est de débit $\mathbf{d} = 0$).

Réciproquement, pour toute chaîne $(a_i, \ldots, a_j)$ issue d'un ensemble de $\mathbf{d}$ chaînes ne partageant pas de sommets dans R, on peut construire un chemin $(\mathbf{s}, a_i, a_{i'}, \ldots, a_j, a_{j'}, \mathbf{p})$ dans T^R. Les chemins obtenus ne partagent aucun arc et l'ensemble de ces arcs définit donc un flot de valeur $\mathbf{d}$ sur T^R.

Dans les deux cas, le score de l'ensemble des chemins, égal à la somme des scores des arcs et sommets traversés dans R est bien égal à l'opposé du coût du flot dans T^R (le flot étant d'une unité sur tout arc porteur de flot). □

On constate donc qu'il est équivalent de chercher un flot de débit $\mathbf{d}$ et de coût minimum dans T^R et et chercher un ensemble de $\mathbf{d}$ chemins de score optimal et ne partageant aucun sommet (ancre) dans R.

Remarque 26 *Dans notre cas :*

- *l'algorithme 2 page 67 de Busacker et Gowen reste polynomial. En effet, par construction du réseau T^R, le débit maximum est égal au nombre d'ancres $\mathcal{A}$, soit au nombre de sommets divisé par 2 (hors source et puits), et la complexité de BG est polynomiale, en $O(A^2S)$.*
- *le flot maximum ne nous intéresse pas en pratique car il correspond à un flot de valeur $|\mathcal{A}|$, traversant chaque ancre a_i indépendamment. Un flot de débit $\mathbf{d}$ (non nécessairement maximum) et de coût minimum sera recherché.*
- *L'algorithme de Busacker et Gowen sature un chemin augmentant à chaque itération. Appliqué à un réseau de relation T^R dont toutes les capacités sont de 1, cet algorithme augmente donc le débit $\mathbf{d}$ de 1 à chaque itération, et permet ainsi de trouver un flot de coût minimum pour tout débit inférieur ou égal au flot maximum.*

3.3.2 Contraintes de cohérence des chaînes

Nous avons montré qu'un flot de débit $\mathbf{d}$ sur T^R définit un ensemble de $\mathbf{d}$ chaînes de R, mais l'ensemble de chaînes défini par le flot *maximum* n'est pas pertinent pour reconstruire l'ensemble des régions dupliquées. L'algorithme de Busacker et Gowen a l'avantage de produire une série de flots de coût minimum (théorème 4 page 66) et de débit augmenté de 1 à chaque itération. Il nous fournit donc, à chaque itération, un ensemble de $\mathbf{d}$ chaînes de score optimal. Mais combien de chaînes doivent être construites ? Et ces chaînes respectent-elles les contraintes de cohérence et de qualité (3.1.2 page 53) ?

3.3.2.1 Qualité des chaînes

Si le nombre de chaînes à construire est inconnu, nous avons cependant certaines exigences au niveau de la qualité des chaînes définies dans la propriété (3.2 page 54). Ceci va nous permettre de définir un critère d'arrêt.

Propriété 3 *Le coût des chemins améliorants construit par l'algorithme de Busacker et Gowen croît à chaque itération.*

Preuve : Notons f^i le flot sur le réseau T^R construit par l'algorithme de Busacker et Gowen à l'itération i. Le chemin améliorant construit dans le graphe d'écart $E(T^R, f^i)$, un des chemins les plus courts de ce graphe, est noté μ_i. Le coût d'un chemin ν quelconque dans $E(T^R, f^i)$ est noté $W_i(\nu)$.

On veut montrer que $\forall i$, le coût du chemin améliorant μ_i dans $E(T^R, f^i)$ croit avec i.

Dans $E(T^R, f^i)$, le chemin améliorant μ_{i-1} apparaît comme un chemin de **p** à **s**, dont le coût $W_i(\mu_{i-1})$ est l'opposé du coût de $W_{i-1}(\mu_{i-1})$ (dans le graphe d'écart précédent).

Soit μ_i un plus court chemin de **s** à **p** dans $E(T^R, f^i)$. Les deux chemins définis par μ_{i-1} et μ_i définissent un circuit dans $E(T^R, f^i)$.

Par définition de $E(T^R, f^i)$, toutes les capacités résiduelles sur les arcs, et donc dans le circuit, sont strictement positives. Il est donc possible d'augmenter le flot le long de ce circuit d'une valeur ϵ (strictement positive) sans changer le débit du flot. Si $W_i(\mu_i)$ est plus faible que $W_{i-1}(\mu_{i-1})$, alors cela permet d'obtenir un flot de même débit et de coût plus faible que f^i ce qui est en contradiction avec son optimalité. □

La propriété 3 permet de définir un critère d'arrêt sur le paramètre $minScore = -maxCout$:

tant que $W_i(\mu_i) \leq maxCout$ **faire**
    Chercher un chemin augmentant de coût minimum
    Augmenter le flot de 1 le long ce ce chemin
fin

Propriété 4 *Le critère d'arrêt $W_i(\mu_i) \leq maxCout$ garantit qu'il n'existe plus de chaîne de score $\geq minScore$, sans diminuer la qualité des chaînes existantes.*

Preuve : Notons $T^R(f^i)$ le réseau de transport T^R privé des arcs et des sommets utilisés par le flot de coût minimum f^i (ceci pour ne pas toucher aux chaînes existantes) et μ un chemin augmentant dans ce réseau (pour un flot nul).

Si le coût $W(\mu) > maxCout$ alors il n'existe plus de chaînes dans $T^R(f^i)$ de score $\geq minScore$. En effet, s'il existait une telle chaîne, il existerait un chemin ν (de **s** à **p**) de coût $W(\nu) < W(\mu)$ dans $T^R(f^i)$, ce qui n'est pas possible puisque μ est un chemin de coût minimum. □

Satisfaction des autres contraintes :

Les autres contraintes de validité définies par la propriété (3.2 page 54), exigeant que les chemins formés définissent des régions suffisamment "solides" (formées d'un nom-

bre suffisant d'ancres et raisonnablement linéaires), sont respectées à l'aide d'un filtre appliqué sur les chaînes. Le résultat de ce filtre donne un ensemble de chaînes appelé $\mathcal{C}$. Par la suite, nous noterons $\mathcal{C}_i$ l'ensemble de chaînes obtenu à l'itération i.

3.3.2.2 Cohérence globale des chaînes

En se focalisant sur la production d'un ensemble de chaînes optimal et non pas sur la production d'une chaîne optimale effectuée de façon itérative et gloutonne, l'approche par flot de coût minimum présente l'avantage d'intégrer :

- l'optimisation d'un score global ;
- la prise en compte de la contrainte (2.4 page 32) de proximité des ancres (via le graphe R) mais aussi la condition (3.4 page 56) de non partage des ancres entre chaînes (via les capacités unitaires) ;
- et la condition (3.6 page 57) (dans le cas des duplications segmentales, en éclaircissant le graphe R).

Elle n'empêche cependant pas de produire des chaînes violant les conditions de non chevauchement (3.3 page 56) et (3.5 page 57) (dans le cas des duplications segmentales). Dans la pratique, certains outils, comme OSFinder (Hachiya *et al.*, 2009), suppriment toutes les ancres intersectant le rectangle défini par la dernière chaîne créée dans un processus glouton (mais cette condition n'est cependant pas suffisante pour éviter totalement le problème : un arc peut encore chevaucher le rectangle, voir figure 3.1 page 55(C)). Dans DAGchainer (Haas *et al.*, 2004), la contrainte (3.3 page 56) est négligée et les chaînes qui ne respectent pas la condition (3.5 page 57) sont déclarées en *tandem* et supprimées. DAGchainer peut aussi violer la contrainte (3.4 page 56) en exécutant deux passes de son algorithme, la première (resp. la deuxième) pour trouver les chaînes de polarités positives (resp. négatives).

Dans notre cas, l'ensemble de chaînes $\mathcal{C}$ produit par l'algorithme peut contenir des chaînes qui violent les conditions (3.3 page 56) et (3.5 page 57) et qui sont classées comme "non cohérentes". Ces chaînes sont non cohérentes, mais contiennent au moins une sous-chaîne qui satisfait les conditions (3.3 page 56) et (3.5 page 57).

Propriété 5 *Soit $\mathcal{C}_i$, l'ensemble des i chaînes définies par le flot f_i sur T_R défini au début de l'itération i, S_c une sous-chaîne stricte d'une chaîne c de $\mathcal{C}_i$ et μ_i le chemin augmentant à l'itération i. Alors le score de la sous-chaîne $S_c = -W(S_c) \geq W(\mu_i)$.*

Preuve : Notons $(a_1, a'_1 \ldots, a_x, a'_x)$ les sommets qui composent le chemin représentant la chaîne c dans le réseau de transport T^R. Le sous-réseau du réseau T^R induit par les sommets de cette chaîne ainsi que les sommets **s** et **p** est visualisé dans la figure 3.9 page ci-contre (A). Par définition du réseau de transport T^R, la source **s** est reliée à chaque sommet a_i et chaque sommet a'_i est relié au puits **p**.

Par définition du graphe d'écart, tous les arcs de la chaîne c sont inversés dans le graphe d'écart courant $E(T^R, f^i)$, et leur coût est l'opposé du coût correspondant dans le réseau

T^R (voir définition 35 page 65). Le sous-graphe induit par les mêmes sommets dans le graphe d'écart est ainsi représenté dans la figure 3.9 (B).

La sous-chaîne S_c est définie par un premier sommet a_i et un dernier sommet a'_j (avec $i \neq 1$ ou $j \neq x$). Cette sous-chaîne définit un chemin $P(S_c) = (\mathbf{s}, a_j, a'_{j-1}, \ldots, a'_i, \mathbf{p})$ dans le graphe d'écart courant $E(T^R, f^i)$.

Le chemin augmentant μ_i étant un plus court chemin de $\mathbf{s}$ à $\mathbf{p}$ dans $E(T^R, f^i)$, son coût minore donc le coût du chemin $P(S_c)$: $W_i(\mu_i) \leq W_i(P(S_c))$.

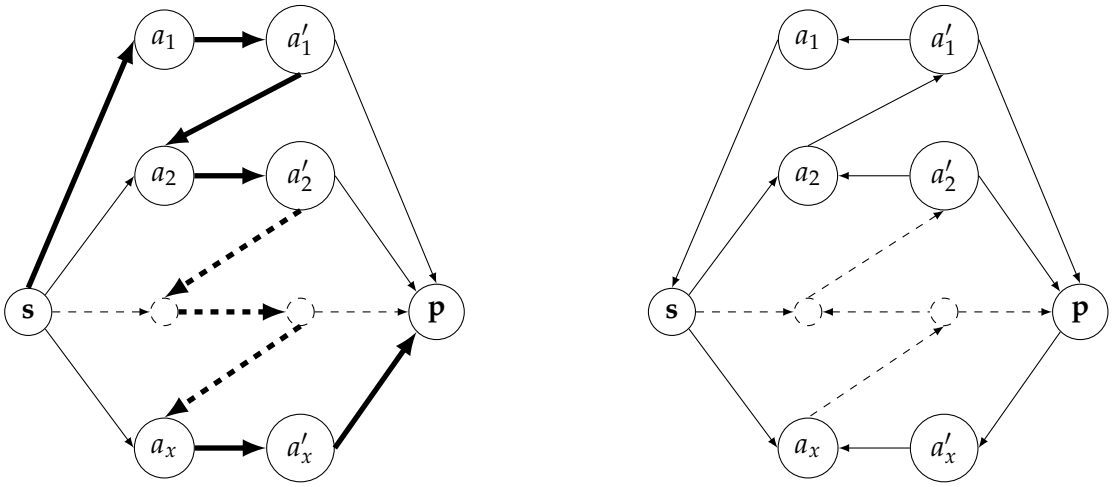


FIGURE 3.9: Illustration d'un réseau de relation (a) qui contient la chaîne c en gras et du graphe d'écart associé (b).

Dans le réseau de transport T^R original, le coût $W(S_c)$ de la sous-chaîne S_c, représentée par le chemin $(a_i, a'_i, \ldots, a_j, a'_j)$ est égal à la somme du coût de l'arc (a_i, a'_i), du coût de l'arc (a_j, a'_j) et de l'opposé de $W_i(P(S_c))$. On a donc :

$$W(S_c) = W((a_i, a'_i)) + W((a_j, a'_j)) - W_i(P(S_c))$$

Les arcs (a_i, a'_i) et (a_j, a'_j) représentant des ancres, leur score est positif et donc leur poids négatif dans T^R. De ce fait :

$$W(S_c) \leq -W_i(P(S_c)) \leq -W_i(\mu_i)$$

On a donc un bien un score de sous-chaîne S_c qui vérifie $-W(S_c) \geq W_i(\mu_i)$. □

Corollaire 1 *Comme $W_i(\mu_i)$ croit à chaque itération (propriété 3 page 71), on a donc une garantie de plus en plus forte sur la qualité des chaînes obtenues.*

Ainsi il suffit d'augmenter les itérations pour supprimer les sous-chaînes de mauvaise qualité jusqu'à ce que toutes les chaînes de $\mathcal{C}$ respectent les conditions (3.3 page 56) et (3.5 page 57). Ce qui modifie le critère d'arrêt, en :

Algorithme 3 : Algorithme de ReD

Données :

- $T^R = (S, A, C, W)$ un réseau de relation
- f un flot sur T^R de débit $\mathbf{d} = 0$

Résultat :

- $\mathcal{C}$ un ensemble de chemins (chaînes) *cohérent(e)s*

début

  tant que $\exists$ μ *de coût minimum dans* $E(T^R, f)$ ***et*** $W_i(\mu_i) \leq maxCout$ ***et*** $\mathcal{C}$ *non coherent* **faire**

    Mettre à jour le flot f (algorithme 1 page 64) ($\mathbf{d}' = 1$)

  return $\mathcal{C}$

fin

A la différence des méthodes gloutonnes employées traditionnellement dans le domaine, notre approche globale permet donc de produire de façon garantie un ensemble de chaînes cohérent et de score optimal, comme l'exige le problème 1 page 60 que nous cherchons à résoudre. La garantie d'optimalité est fournie par l'algorithme de flot de coût minimum utilisé et la cohérence par le critère d'arrêt employé. Cependant, l'algorithme polynomial que nous proposons n'est pas capable de produire un ensemble de cardinalité d arbitraire et ne résout donc pas le problème 1 page 60 dans toute sa généralité. La cardinalité d de l'ensemble de chaînes ne peut être imposée *a priori* dans notre algorithme mais est fixée de façon opérationnelle par le critère d'arrêt.

3.3.2.3 Composantes connexes du graphe de relation

En pratique, le graphe de relation R n'étant pas connexe, la transformation en réseau de relation (voir section 3.3.1 page 68) est appliquée à chaque composante connexe et nous obtenons p réseaux de transport, où p est le nombre de composantes connexes de R.

La notion de cohérence (voir section 3.1.2 page 53) d'un ensemble de chaînes est indépendante pour chaque composante connexe. De ce fait, l'algorithme modifié de Busacker et Gowen (1961) est appliqué indépendamment à chaque composante connexe.

L'application de l'algorithme à chaque composante connexe a une autre vertu, celle de diminuer le temps d'exécution. En effet la complexité de l'algorithme Busacker et Gowen est importante, elle s'exprime en $O(isr)$ avec i le flot maximum, s le nombre de sommets et r le nombre d'arcs du graphe. Même si dans notre cas i est bien inférieur à la

valeur du flot maximum, la taille du graphe peut suffire pour être à l'origine de temps d'exécution trop longs. Par exemple dans le cas de la recherche de duplications segmentales sur le génome d'*Arabidopsis thaliana*, le graphe de relations comporte 165977 ancres et 487932 arcs. Et le fait, d'appliquer l'algorithme sur chaque composante connexe plutôt que sur le graphe de relation diminue le temps d'exécution.

Si cela ne suffit pas, il est possible de demander à ReD de limiter le nombre d'arcs sortants de chaque sommet/ancre du graphe de relation, aux 15 meilleurs arcs (les arcs de scores les plus élevés).

3.3.3 Estimation des paramètres

Afin de rendre la méthode plus robuste et facile à paramétrer, ReD est capable d'estimer ces paramètres seul ou en limitant l'intervention des utilisateurs.

Tout d'abord, les paramètres $DistMax_{\mathsf{A}}, DistMax_{\mathsf{B}}$ peuvent être estimés à partir des données à l'aide de l'équation 3.8 ci-dessous. Soit D_g la somme des longueurs entre deux ancres consécutives sur le génome g :

$$DistMax_g = \frac{D_g}{\sqrt{2|\mathcal{A}|}} \tag{3.8}$$

$D_g/\sqrt{|\mathcal{A}|}$ constitue une approximation de la distance euclidienne moyenne entre deux ancres distribués aléatoirement sur la région bidimensionelle d'un dotplot (SyMAP (Soderlund *et al.*, 2006)). Le facteur $\sqrt{2}$ tient compte du fait que la distance sur un génome de deux ancres sur la même diagonale est égale à la distance euclidienne divisée $\sqrt{2}$ (voir figure 2.6 page 29).

Enfin, le score minimum d'un chemin *minScore* est une variable difficile à paramétrer *a priori*. En effet les scores des arcs et des sommets sont de différentes natures et il en existe différentes définitions possibles (S_e, S_s pour les sommets, S_m, S_e, S_d pour les arcs). Afin de faciliter le paramétrage de *minScore* quelques soient les formules choisies, ReD commence par normaliser les scores pour que le score moyen d'un arc, $\overline{arc}$ soit l'opposé du score moyen d'une ancre $\overline{ancre}$:

$$\overline{ancre} = -\overline{arc} \text{ avec } \overline{ancre} > 0$$

Ainsi l'espérance du score d'une chaîne est égale à $\overline{ancre}$ quelle que soit sa longueur, du fait qu'une chaîne est toujours constituée de x ancres et $x-1$ arcs. Cette propriété facilite le paramétrage de *minScore* via un cœfficient de pondération β :

$$minScore = \beta.\overline{ancre} \tag{3.9}$$

De cette façon, il suffit d'augmenter β pour favoriser la création de chaînes plus longues et de meilleure qualité.

3.4 Le *pipeline* "ReD Tandem"

La recherche de duplications en tandem est un cas particulier de la recherche de régions conservées. En effet, **une** duplication en tandem, c'est une région qui s'est dupliquée **plusieurs** fois. Chacune de ces régions [1] est appelée une *unité de duplication* et la région qui contient l'ensemble des unités de duplication est appelée *région dupliquée*.

Ainsi, pour détecter correctement une duplication en tandem, il faut détecter la région dupliquée ainsi que toutes les unités de duplication. Or une chaîne ne met en correspondance que deux régions, cela implique qu'il est nécessaire de trouver une chaîne pour chaque paires d'unités de duplication (voir figure 3.2(d) page 58).

La reconstruction directe de l'ensemble des chaînes d'une duplication en tandem par ReD est extrêmement rare. Le signal de conservation au niveau nucléique est faible et ReD ne reconstruit souvent qu'un sous-ensemble de ces chaînes. Pour essayer de récupérer les signaux manquants, nous avons défini un pipeline, appelé ReD Tandem, qui ajoute deux étapes pour traiter et filtrer les chaînes de la sortie de ReD. Par la suite, nous évaluerons ce pipeline dans la section 4.2 page 94 par la confrontation de sa sortie avec des données expertes récoltées à partir de l'annotation d'un génome.

La reconstruction des régions dupliquées et de leurs unités de duplication se fait donc en trois étapes :

1. Le chaînage : qui est une exécution de ReD pour chaîner les ancres nucléiques [2] pour reconstruire partiellement le signal des duplications en tandem.
2. Le grand rassemblement : qui analyse les chaînes construites par ReD et définit pour chaque région dupliquée, sa position sur le génome et une unité de duplication de référence, appelée plus loin *séquence de référence*.
3. The End : qui exécute une seconde fois ReD mais cette fois sur des ancres obtenues à l'aide d'un logiciel d'alignement ADN contre ADN de type TBLASTX entre une région dupliquée et sa séquence de référence.

Le logiciel qui exécute la succession de ces trois étapes, qui sont détaillées ci-dessous, est appelé ReD Tandem.

3.4.1 Le chaînage

Nous allons voir ici comment ReD est paramétré pour les duplications en tandem.

Tout d'abord, il faut commencer par analyser les ancres. Les unités de duplications sont proches les unes des autres par définition. Ainsi les ancres qui représentent deux régions sur des chromosomes différents, ou trop éloignées, sont inutiles. En conséquence

1. issues ou à l'origine de la duplication
2. Obtenues avec le logiciel Glint (Faraut et Courcelle, 2011), un logiciel d'alignement local du type BLAST, qui a l'avantage de vérifier que les séquences d'une ancre ne se chevauchent pas (voir équation 3.6 page 57).

avant d'exécuter ReD, un filtre est appliqué sur les ancres qui ne respectent pas l'une des deux contraintes :

$$\begin{array}{rcl} a^A.chrom & = & a^B.chrom \\ dist(a^A, a^B) & \leq & distDiag \end{array} \tag{3.10}$$

avec *distDiag* la distance maximum entre deux régions d'une même ancre. Cette distance correspond aussi à la distance verticale ou horizontale de l'ancre à la diagonale du dotplot, d'où son nom.

Ensuite, ReD est exécuté avec les paramètres suivants :

- Relation de chaînabilité : $\prec_\varnothing$. Comme l'on s'intéresse essentiellement aux duplications qui sont susceptibles de représenter un élément fonctionnel, on se limite volontairement à la reconstruction de régions composées d'ancres dans le même ordre (ou inversé) et de même polarité.
- Scores des ancres : $S_s(a_i)$. Glint (Faraut et Courcelle, 2011), le logiciel d'alignement local qui fournit les ancres, utilise un score d'alignement comme indice de qualité et pas de $e - value$. C'est donc ce score qui est utilisé pour pondérer les sommets. De plus, les régions du génome qui sont utilisées dans plusieurs ancres ont plus de chances d'appartenir à une duplication en tandem. En conséquence le score des ancres sera multiplié par un bonus qui est égal au pourcentage de fois que les régions d'une ancre sont utilisées dans d'autres ancres.
- Scores des arcs : $S_d(a_i \prec a_j)$. Pour les mêmes raisons que pour $\prec_\varnothing$, on favorise l'association d'ancres qui forment une diagonale.
- Nombre d'ancre minimum : $minAncre = 1$. Il est possible qu'une chaîne soit composée d'une seule ancre, si la duplication est très récente. Le paramètre $minAncre$ est fixé en conséquence.
- Longueur minimum des séquences : Notre objectif consiste à retrouver les séquences dupliquées en tandem de taille ≥ 500. Ainsi un filtre est appliqué à *posteriori* pour supprimer les chaînes qui possède au moins un intervalle, c^A ou c^B, de taille < 500 nucléotides.

Remarque 27 *Une séquence dupliquée une seule fois en tandem ne sera représentée que par une seule ancre. En conséquence, le score de cette ancre ne bénéficie pas de bonus et sera pénalisée par rapport à des régions dupliquées plusieurs fois (voir figure 4.4 page 99). Pour limiter cet effet, toutes les ancres non utilisées dans des chaînes et possédant deux intervalles (a^A ou a^B) de taille ≥ 500 nucléotides, sont transformées en chaînes avec un score $c.s = scoreMin$.*

Remarque 28 *La relation $\prec_\varnothing$ permet de construire des chaînes de polarité positive et négative. En conséquence, il est possible de détecter des duplications en tandem qui ont changé de brin d'ADN, ce qui est souvent le cas (Lajoie et al., 2007).*

3.4.2 Le grand rassemblement

Maintenant que ReD a reconstruit les chaînes, il nous faut les analyser pour trouver les régions dupliquées et leur position ainsi qu'une unité de duplication de référence (pour la suite du processus).

Pour cela, on commence par construire un nouveau graphe non orienté, à partir des chaînes reconstruites par ReD et des ancres non utilisées dans des chaînes:

- chaque ancre et chaque chaîne est un sommet du graphe.
- et une arête est ajoutée entre deux sommets si au moins un nucléotide est chevauché par une des deux régions de chaque sommet.

Les composantes connexes de ce graphe de chevauchement représentent chacune une région dupliquée, indiquant sa position sur le génome. Ces composantes sont extraites par un classique algorithme de parcours en profondeur (Cormen *et al.*, 2001) (Section 22.3: Depth-first search, pp. 540–549). Ces composantes peuvent se classer en deux catégories :

1. La première catégorie correspond aux composantes connexes qui n'ont aucune chaîne parmi leurs sommets. Une chaîne peut être constituée d'une seule ancre, ainsi si une composante connexe ne contient aucune chaîne, alors cela signifie que ReD n'a trouvé aucun signal de conservation suffisamment fort et il n'est pas nécessaire de traiter ces composantes. Elles sont supprimées.
2. La deuxième catégorie correspond aux composantes connexes qui possèdent au moins une chaîne parmi leurs sommets. Dans ce cas de figure, ReD a détecté une région dupliquée composée d'au moins deux unités de duplications. Ce signal capturé n'est, la plupart du temps, que partiel. En effet, comme nous le montre la figure 3.10 page suivante, toutes les chaînes ne sont pas nécessairement reconstruites. Toutes ces régions sont donc mises de côté pour une deuxième exécution de ReD localisée sur la région dupliquée afin d'essayer de reconstruire les unités manquantes.

3.4.2.1 Sélection de la séquence de référence :

Il s'agit ici d'identifier la plus petite unité de duplication. Cette unité caractérise la taille de la plus petite région ayant fait l'objet de plusieurs duplications en tandem. Pour cela, une séquence de référence est sélectionnée parmi toutes les régions représentées par les chaînes de la composante. Cette séquence de référence sélectionné correspond à l'intervalle le plus long de la chaîne de score maximum, c^A ou c^B.

Pour finir, une dernière vérification est réalisée afin de vérifier que la séquence de référence ne contient qu'une unité de duplication. En effet, il est toujours possible que la région soit composée de plus d'une unité de duplication. Notamment, si toutes les

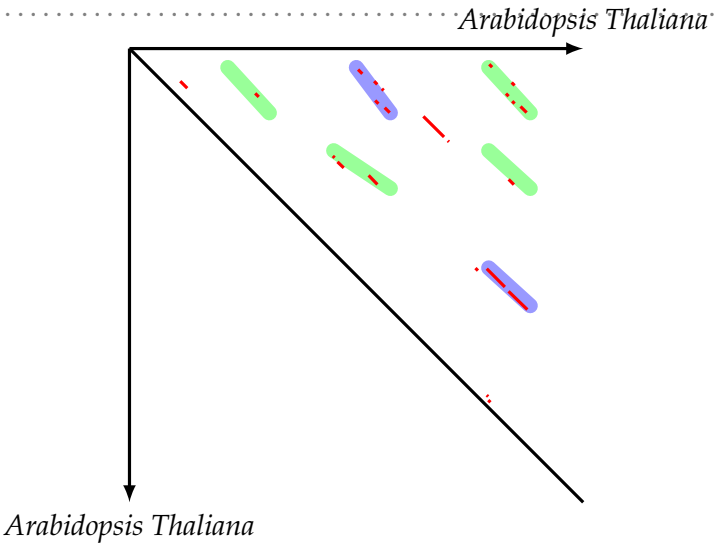


FIGURE 3.10: Exemple d'un signal incomplet de duplication en tandem sur le chromosome 2 d'*Arabidopsis thaliana* entre la position 18818000 et 18830000. Les diagonales en rouge représentent les ancres nucléiques, les diagonales en bleu représentent les chaînes construites à la première exécution de ReD et les diagonales en vert représentent les chaînes oubliées (non construites) par ReD.

chaînes de la composante sont suffisamment éloignées de la diagonale, dans ce cas la distance entre deux régions est très grande et peut contenir plusieurs unités.

Pour cela, la séquence de référence est alignée contre elle-même. Si au moins 50% des nucléotides de cette séquence sont utilisés dans des alignements locaux, c'est un signe qu'elle contient au moins deux unités de duplication et qu'il faut supprimer une partie de la séquence.

Dans ce cas, on sélectionne la plus petite région alignée car elle représente la plus petite région présente en double dans la séquence. La région de cet alignement la plus proche de l'extrémité de la séquence de référence est identifiée et on retire tous les nucléotides à partir de l'extrémité choisie jusqu'à la fin de la région identifiée.

Cette procédure de réduction de la séquence de référence est itérative et se poursuit tant qu'il y a au moins 50% des nucléotides utilisés dans des alignements locaux.

3.4.3 The End

Arrivé à cette étape, ReD Tandem possède, pour chaque région dupliquée, une séquence de référence. Maintenant l'objectif est d'aligner la séquence de référence sur la région dupliquée et d'exécuter ReD une seconde fois. avec une suite de k répétitions approchées de celle-ci a été abordé par (Fischetti *et al.*, 1993) dans un algorithme d'alignement de

séquences d'ADN global dédié.

Cependant, le signal de similarité au niveau nucléique (ancres nucléiques), n'est pas toujours suffisant pour reconstruire toutes les unités de duplications (avec ReD). Qu'en est-il du signal de conservation avec une traduction de la séquence d'ADN niveau protéique ?

Au niveau protéique, il est connu que le signal de similarité est mieux conservé. Notamment parce que plusieurs codons codent le même acide aminé (voir tableau 1.1 page 6). En conséquence, ReD Tandem utilise cette fois le logiciel d'alignement local TBLASTX (Altschul *et al.*, 1990), qui a la propriété de traduire [3] les séquences nucléiques en séquences protéiques avant d'aligner et de créer les ancres. Une approche similaire, s'appuyant sur les gènes de protéine annotés et BLASTX, est utilisée et forme le coeur de (Despons *et al.*, 2010).

Remarque 29 *Le logiciel TBLASTX est très coûteux en temps. Il faut commencer par traduire les séquences nucléiques en séquences protéiques avant de les aligner. Ce processus est beaucoup plus long que la construction "classique" d'ancres nucléiques. Avec les ordinateurs actuels, il est inenvisageable d'utiliser TBLASTX sur tout un génome, mais sur de petites régions (séquences) comme les* régions dupliquées, *c'est possible et efficace.*

Remarque 30 *Avant d'utiliser TBLASTX, les frontières des régions dupliquées sont élargies de $DistMax_g$ (voir section 4.2.1.1 page 94) de chaque coté pour éventuellement capturer des unités de duplications à l'extérieur de la région, non détectées au niveau nucléique.*

Une fois les ancres obtenues, ReD est exécuté une deuxième fois avec les mêmes paramètres. Le graphe est ici tellement simple à analyser qu'il suffit de donner un score identique à chaque ancre et d'utiliser l'algorithme de plus court chemin pour reconstruire les chaînes.

Pour finir, un filtre est appliqué sur les chaînes afin de ne garder que les chaînes qui représentent des unités de duplications de taille supérieure ou égale à 50% de la séquence de référence. Sinon, le risque de capturer de mauvaises chaînes est trop grand.

3.5 Conclusion

Nous avons vu dans un premier temps, l'avantage de considérer la recherche de duplications comme la création d'un ensemble de chaînes, plutôt que la création de chaînes indépendantes les unes des autres, à travers la notion de *cohérence* d'un ensemble de chaînes. Cette notion de cohérence est particulièrement intéressante pour justifier de la qualité des résultats obtenus, puisque qu'elle garantie que chaque région reconstruite, par l'ensemble des chaînes, ne possède qu'un lien de parenté qui résulte d'un évènement de spéciation ou de duplication à l'origine de son apparition.

3. dans les six phases.

Par la suite, nous avons défini notre problème comme étant la création d'un ensemble de chaînes de score optimum et cohérent et nous avons vu comment nous avons résolu ce problème en adaptant les algorithmes de la théorie des flots qui a ensuite été implémenté dans le logiciel ReD.

Enfin, pour atteindre notre objectif de détection de duplications en tandem au niveau nucléique, nous avons décrit le *pipeline* ReD Tandem, qui regroupe la succession d'étapes nécessaire afin d'analyser et d'identifier les régions dupliquées en tandem à partir des résultats de ReD.

Passons maintenant à l'application de cette nouvelle méthode et à l'analyse de ces résultats.

Chapitre 4

Application et résultats

Sommaire

Ce chapitre est consacré à la présentation et à l'analyse des résultats de la nouvelle méthode de chaînage ReD, décrite dans le chapitre 3.

Dans un premier temps, cette méthode est appliquée à la recherche de régions homologues et/ou de duplications segmentales, afin de montrer que le chaînage par la recherche d'un flot donne des résultats comparables aux précédentes méthodes avec l'avantage de fournir un ensemble de chaînes cohérent.

Cette vérification faite, nous avons appliqué le *pipeline* ReD Tandem à la recherche de duplications en tandem, dans une approche agnostique, exploitant uniquement des ancres détectées par alignement d'ADN du génome d'*Arabidopsis thaliana*. Pour évaluer les résultats obtenus, nous avons confronté ceux-ci à un jeu de référence de gènes de protéines dupliqués en tandem, construit en exploitant le protéome défini par l'annotation. Comme une part non négligeable des régions et unités construites par ReD ne correspondent pas à des gènes de protéines en tandem, nous avons complété cette analyse par une comparaison directe des régions et unités construites par ReD aux éléments fonctionnels contenus dans l'annotation du génome, incluant pseudo-gènes, gènes d'ARN...

4.1 Application à la recherche de duplications segmentales et de régions homologues en exploitant l'information protéique

L'objectif de cette section est de comparer la nouvelle méthode, implémentée dans le logiciel ReD, avec les méthodes existantes. Nous avons évalué notre algorithme dans le cas de la recherche de duplications segmentales internes au génome d'*Arabidopsis thaliana* et de régions homologues entre le génome de *Glycine max* et celui de *Medicago truncatula*. Cette recherche est réalisée sur la base des séquences protéiques fournies par l'annotation. Pour cette étude, nous avons récupéré les génomes et les annotations des versions :

- 7.1 d'*Arabidopsis thaliana* [1]
- 1.0 de *Glycine max* [2]
- 3.0 de *Medicago truncatula* [3]

Pour ces deux exemples, nous présentons d'abord une comparaison entre la méthode gloutonne et la méthode par flot de coût minimum, toutes choses étant égales par ailleurs (fonctions de scores, graphe de relation...). Cette comparaison sera l'occasion de montrer que les chaînes obtenues par l'optimisation globale de l'algorithme de flot

1. ftp://ftp.ncbi.nih.gov/genomes/Arabidopsis_thaliana
2. ftp://ftp.jgi-psf.org/pub/JGI_data/Glycine_max/Glyma1/assembly
3. http://medicagohapmap.org/

possèdent de meilleures propriétés que celles obtenues par un algorithme de plus court chemin, traditionnellement utilisé.

Puis dans un deuxième temps, sur les mêmes génomes, ReD sera comparé aux logiciels DAGchainer (Haas *et al.*, 2004) et OSfinder (Hachiya *et al.*, 2009). DAGchainer est un logiciel très cité dans les analyses génomiques. OSfinder est ajouté en tant que logiciel très récent. Ces comparaisons montreront que la prise en compte des contraintes de cohérence en amont dans le formalisme est un avantage pour fournir un ensemble de chaînes cohérent.

4.1.1 Paramètres et données utilisés à l'exécution de ReD

4.1.1.1 Les ancres

Tout d'abord pour alimenter les logiciels, il faut détecter les ancres. Pour cela nous avons utilisé le logiciel BLASTP (Altschul *et al.*, 1990) en comparant les protéines du génome A aux protéines du génome B. Les alignements doivent couvrir au moins 70% de chaque protéine et avoir une *e-value* (définition 7 page 21) de 10^{-5} au plus.

De plus, un filtre est ajouté pour supprimer toutes les paires de gènes qui possèdent au moins un gène présent dans plus de 50 paires de gènes. Ce filtre est appliqué non pas pour résoudre un problème d'explosion combinatoire, mais parce que ces gènes *répétés* sont susceptibles de biaiser la reconstruction des relations d'homologie (voir Filtre 1 page 22).

4.1.1.2 Les paramètres de ReD

Pour la recherche de régions homologues et de duplications segmentales, ReD est exécuté avec les paramètres suivants :

- $minAncre = 3$. Ce paramètre fixe le nombre minimum d'ancres pour qu'une chaîne soit considérée comme *valide* (voir équation 3.2 page 54). DAGchainer utilise la même valeur par défaut.
- $\beta = 4$. Ce paramètre fixe le score minimum d'une chaîne *valide* à 4 fois celui de l'espérance du score d'une chaîne quelconque dans le graphe de relation (équation 3.9 page 75).
- $minCorr = 0,8$. Ce paramètre fixe le coefficient de corrélation minimum pour qu'une chaîne soit considérée comme *valide*.
- $\prec = \prec_2$. La relation de chaînabilité $\prec_2$ permet à ReD d'accepter les changements d'ordres et d'orientations entre ancres consécutives d'une même chaîne (voir définition 15 page 31).

- le paramètre $DistMax_g$, qui fixe la distance maximum entre deux ancres *chaînables* (voir définition 9 page 30) sur le génome g, est calculé automatiquement (voir équation 3.8 page 75).
- $S(a) = S_{evalue}$. Ce paramètre fixe la fonction de score utilisée pour pondérer les ancres du graphe de relation. La fonction choisie est une fonction basée sur le logarithme de la e-value de l'alignement de chaque ancre (voir équation 2.6 page 35).

Notons que le paramètre $S(a_i, a_j)$, qui fixe la fonction de score utilisée pour pondérer les arc de transition entre ancres, n'est pas précisé à ce niveau. En effet, différentes valeurs de ce paramètre seront testées pendant l'analyse des résultats de ReD.

Les paramètres sont choisis (en particulier $\prec_2$) pour tester ReD, dans le cas le plus complexe et afin de montrer que malgré cette difficulté, ReD reconstruit des régions dupliquées de bonne qualité ainsi que de nouvelles régions, non détectées par les méthodes de reconstruction de régions colinéaires (interdisant les changement d'ordres ou d'orientation).

4.1.2 Comparaison avec la méthode gloutonne

L'approche par flot et l'approche gloutonne par plus court chemin ont été comparées sur le problème de la recherche d'un nombre de chaînes fixé. Dans notre approche, le nombre de chaînes correspond exactement au nombre d'itérations. Pour se placer dans un cadre comparable, les contraintes de validité et de cohérence (équations 3.2 page 54, 3.3 page 56 et 3.5 page 57) sont ignorées dans cette comparaison. Dans tous les cas, le graphe de relation R et les fonctions de score sont identiques. Les performances sont comparées à l'aide de trois critères :

1. le score de l'ensemble de chaînes produit ;
2. le coefficient de corrélation évaluant la linéarité des chaînes produites ;
3. la longueur totale des chaînes (somme des longueurs des intervalles couvrant les deux séquences génomiques).

Un score élevé, une forte linéarité et une large couverture des génomes sont préférables sachant que ces critères sont souvent antagonistes.

Les résultats présentés dans les tableaux 4.1 page suivante et 4.2 page 88 montrent que l'approche gloutonne fournit des scores inférieurs, comme cela était attendu, et que la différence entre les deux approches augmente avec le nombre de chaînes construites. L'approche par flot de coût minimum permet d'autre part de construire des chaînes plus longues (ajout d'ancres aux extrémités des chaînes) avec une meilleure linéarité : les choix locaux réalisés par l'approche gloutonne pénalisent la qualité globale des chaînes.

Ces tableaux montrent aussi que la fonction de score S_{sub} (équation 2.10 page 36), qui favorise les substitutions, est celle qui prédit les chaînes les plus longues pour la recherche

S_{indel}

Arabidopsis vs Arabidopsis		Score	Coef. de corrélation			Longueur en kb		
Méthode	Nombre de chaînes		min	moy	max	min	moy	max
Gloutonne	100	$19,7 \cdot 10^6$	0,16	0,91	1	363	3440	12130
	200	$25,5 \cdot 10^6$	0,01	0,84	1	49	2397	12130
	300	$29,7 \cdot 10^6$	0,01	0,79	1	49	2074	12130
Flot	100	$\mathbf{20,5 \cdot 10^6}$	0,18	**0,95**	1	560	**3756**	11624
	200	$\mathbf{26.8 \cdot 10^6}$	0,01	**0,87**	1	91	**2831**	11624
	300	$\mathbf{31,5 \cdot 10^6}$	0,01	**0,81**	1	58	**2425**	11624

S_{sub}

Arabidopsis vs Arabidopsis		Score	Coef. de corrélation			Longueur en kb		
Méthode	Nombre de chaînes		min	moy	max	min	moy	max
Gloutonne	100	$24,3 \cdot 10^6$	0,02	0,85	1	756	3720	12130
	200	$31,7 \cdot 10^6$	0,02	0,78	1	62	2800	12130
	300	$37,0 \cdot 10^6$	0,01	0,72	1	61	2388	12130
Flot	100	$\mathbf{25,1 \cdot 10^6}$	0,04	**0,89**	1	995	**4046**	1181
	200	$\mathbf{33.1 \cdot 10^6}$	0,02	**0,83**	1	329	**3222**	9907
	300	$\mathbf{38.7 \cdot 10^6}$	0,01	**0,75**	1	61	**2762**	11290

S_{neutre}

Arabidopsis vs Arabidopsis		Score	Coef. de corrélation			Longueur en kb		
Méthode	Nombre de chaînes		min	moy	max	min	moy	max
Gloutonne	100	$24,5 \cdot 10^6$	0,07	0,85	1	756	3744	12130
	200	$31,9 \cdot 10^6$	0,02	0,79	1	56	2769	12130
	300	$37,3 \cdot 10^6$	0,00	0,73	1	56	2386	12130
Flot	100	$\mathbf{25,2 \cdot 10^6}$	0,02	**0,89**	1	100	**4043**	11811
	200	$\mathbf{33,3 \cdot 10^6}$	0,00	**0,83**	1	560	**3217**	10074
	300	$\mathbf{39,0 \cdot 10^6}$	0,00	**0,76**	1	75	**2748**	10000

TABLE 4.1: Comparaison du score, du coefficient de corrélation et de la longueur des chaînes entre la méthode *gloutonne* et la méthode de *flot*, dans le cas de la recherche de duplications segmentales sur le génome *Arabidopsis thaliana*. Le premier tableau contient les résultats de ReD obtenus avec la fonction de transition S_{indel} qui favorise les insertions et les délétions entre ancres ; le deuxième avec S_{sub} qui favorise les substitutions entre ancres ; et le troisième avec la distance neutre S_{neutre}. Dans chaque tableau, un chiffre en gras indique un meilleur résultat dans une comparaison locale flot contre glouton.

S_{indel}

Glycine vs Medicago		Score	Coef. de corrélation			Longueur en kb		
Méthode	Nombre de chaînes		min	moy	max	min	moy	max
Gloutonne	100	$39,2 \cdot 10^6$	0,02	**0,82**	1	987	7380	24396
	200	$52,5 \cdot 10^6$	0,01	**0,77**	1	552	6258	24396
	300	$61,8 \cdot 10^6$	0,01	0,74	1	356	5533	24396
Flot	100	$\mathbf{39,9 \cdot 10^6}$	0,01	**0,82**	1	987	**7814**	24554
	200	$\mathbf{53,5 \cdot 10^6}$	0,01	**0,77**	1	552	**6638**	24554
	300	$\mathbf{62,9 \cdot 10^6}$	0,01	**0,75**	1	356	**5845**	24554

S_{sub}

Glycine vs Medicago		Score	Coef. de corrélation			Longueur en kb		
Méthode	Nombre de chaînes		min	moy	max	min	moy	max
Gloutonne	100	$51,7 \cdot 10^6$	0,04	0,84	1	1797	6301	24396
	200	$70,9 \cdot 10^6$	0,04	0,84	1	676	5052	24396
	300	$84,2 \cdot 10^6$	0,03	**0,82**	1	552	4420	24396
Flot	100	$\mathbf{52,3 \cdot 10^6}$	0,04	**0,85**	1	1797	**6692**	24409
	200	$\mathbf{71,9 \cdot 10^6}$	0,04	**0,85**	1	676	**5341**	24409
	300	$\mathbf{85,3 \cdot 10^6}$	0,03	**0,82**	1	552	**4641**	24409

S_{neutre}

Glycine vs Medicago		Score	Coef. de corrélation			Longueur en kb		
Méthode	Nombre de chaînes		min	moy	max	min	moy	max
Gloutonne	100	$53,6 \cdot 10^6$	0,02	0,80	1	1737	7915	24396
	200	$73,5 \cdot 10^6$	0,01	0,78	1	676	6525	24396
	300	$87,2 \cdot 10^6$	0,01	**0,77**	1	552	5731	24396
Flot	100	$\mathbf{54,3 \cdot 10^6}$	0,01	**0,82**	1	1839	**8329**	24409
	200	$\mathbf{74,7 \cdot 10^6}$	0,01	**0,79**	1	937	**6972**	24409
	300	$\mathbf{88,4 \cdot 10^6}$	0,01	**0,77**	1	552	**5983**	24409

TABLE 4.2: Comparaison du score, du coefficient de corrélation et de la longueur des chaînes entre la méthode *gloutonne* et la méthode de *flot*, dans le cas de la recherche de régions homologues entre le génome de *Glycine max* et et le génome de *Medicago truncatula*. Le premier tableau contient les résultats de ReD obtenus avec la fonction de score S_{indel} qui favorise les insertions et les délétions entre ancres ; le deuxième avec S_{sub} qui favorise les substitutions entre ancres ; et le troisième avec la fonction de score neutre S_{neutre}. Dans chaque tableau, un chiffre en gras indique un meilleur résultat dans une comparaison locale flot contre glouton.

de duplications segmentales sur le génome *Arabidopsis thaliana*. C'est par contre la fonction de score S_{indel} (équation 2.8 page 36), qui favorise les insertions et les délétions, qui est plus performante de ce point de vue dans la recherche de régions homologues entre le génome de *Glycine max* et et le génome de *Medicago truncatula*.

En effet, dans ce cas, le génome de *Medicago truncatula* est plus grand que celui de *Glycine max*, et les densités en gènes sont différentes. Ainsi les distances entre deux couples de gènes homologues consécutifs changent entre les deux génomes. Il n'est donc pas surprenant qu'une distance qui favorise les insertions et les délétions donne de meilleurs résultats.

Ces tableaux sont aussi l'occasion de remarquer que la distance S_{neutre} (équation 2.9 page 36), qui donne un score identique à n substitutions et n insertions ou délétions, est la plus robuste et donne des chaînes de longueurs comparables à la meilleure distance dans chaque cas.

4.1.3 Comparaison avec des logiciels existants

La comparaison avec des logiciels existants n'est pas évidente. Tout d'abord parce qu'il n'existe pas de jeu de données où les relations d'homologies entre régions sont avérées. Mais aussi car chaque logiciel intègre ses propres méthodes de filtrage pour les données en entrée, des fonctions de score différentes voire même un graphe de relation différent. Pour évaluer les logiciels, en dehors des mesures de longueur et de corrélation déjà réalisées, nous nous sommes appuyés également sur le caractère cohérent des ensembles de chaînes produits. En effet, d'après la propriété 1 page 56, les projections de deux chaînes sur les deux génomes ne doivent pas se chevaucher sur au moins l'un des deux génomes. Lorsque ce n'est pas le cas on dit que ces chaînes se chevauchent.

Les chaînes violant cette condition sont comptabilisées dans la colonne *Chevauche*. Dans le cas de la comparaison d'un génome contre lui-même, une relation d'homologie entre deux régions qui se chevauchent viole également la propriété 2 page 57. Dans ce cas, les chaînes sont proches de la diagonale du dotplot et elles sont comptabilisées dans la colonne *Diag*.

Arabidopsis vs Arabidopsis	Nombre de chaînes		
Logiciel	Total	Diag	Chevauche
DAGchainer	306	65	63
OSfinder	286	5	126
ReD	258	0	0

Glycine vs Medicago	Nombre de chaînes	
Logiciel	Total	Chevauche
DAGchainer	708	116
OSfinder	7864	2143
ReD	1198	0

TABLE 4.3: Ces tableaux montrent le nombre de chaînes créées et le nombre de chaînes incohérentes.

Le tableau 4.3 nous montre que seul ReD crée un ensemble de chaînes cohérent alors que la proportion de chaînes incohérentes créées par DAGchainer et OSfinder est com-

prise entre 10% et 50%. Il est naturellement possible de filtrer ces chaînes, mais le nombre de chaînes restantes diminue d'autant. Une alternative consiste à trouver un sous-ensemble de chaînes cohérent mais ce n'est pas un problème simple, les chaînes incompatibles s'excluant mutuellement. ReD règle ce problème en amont, dans son formalisme, et dans l'algorithme de flot associé.

De plus nous remarquons que le nombre de chaînes d'OSfinder est très élevé dans le tableau de droite. L'analyse comparative des longueurs de chaînes (tableau 4.4 page 90) et des dotplots (figure 4.1 page suivante et 4.2 page 92), nous montre que les résultats d'OSfinder sont très différents des résultats de ReD et de DAGchainer. Cette différence est sans doute due au fait qu'OSFinder a été développé pour détecter les régions homologues et testé sur des génomes de mammifères qui possèdent moins de duplications internes que les génomes de plantes utilisés.

Arabidopsis vs Arabidopsis	Longueur des chaînes en kb		
Logiciel	Min	Moy	Max
DAGchainer	155	25177	56362
OSfinder	6	4784	60433
ReD	295	25431	53025

Glycine vs Medicago	Longueur des chaînes en kb		
Logiciel	Min	Moy	Max
DAGchainer	56	42347	92669
OSfinder	13	2800	29837
ReD	64	41668	92669

TABLE 4.4: Ces tableaux montrent que les résultats de ReD et de DAGchainer, en termes de longueur, sont assez proches mais se distinguent nettement de ceux d'OSfinder.

Pour ce qui est de la comparaison entre ReD et DAGchainer, l'exemple illustré par la figure 4.2 page 92 nous montre que la prise en compte de la cohérence des chaînes permet une sélection plus claire des régions ancestrales communes. Cependant, la prise en compte de cette cohérence dans le critère d'arrêt mène à une fragmentation en plusieurs chaînes de deux couples de régions homologues (A et B). Cette figure est aussi l'occasion de montrer que ReD en acceptant les changements d'ordre et d'orientation, est seul capable de détecter de nouvelles chaînes non détectées par DAGchainer (C, D et E). Mais l'inverse est aussi vrai, DAGchainer possède des chaînes non détectées par ReD, cependant ces chaînes sont souvent très courtes et donc d'une qualité discutable (F, G et H).

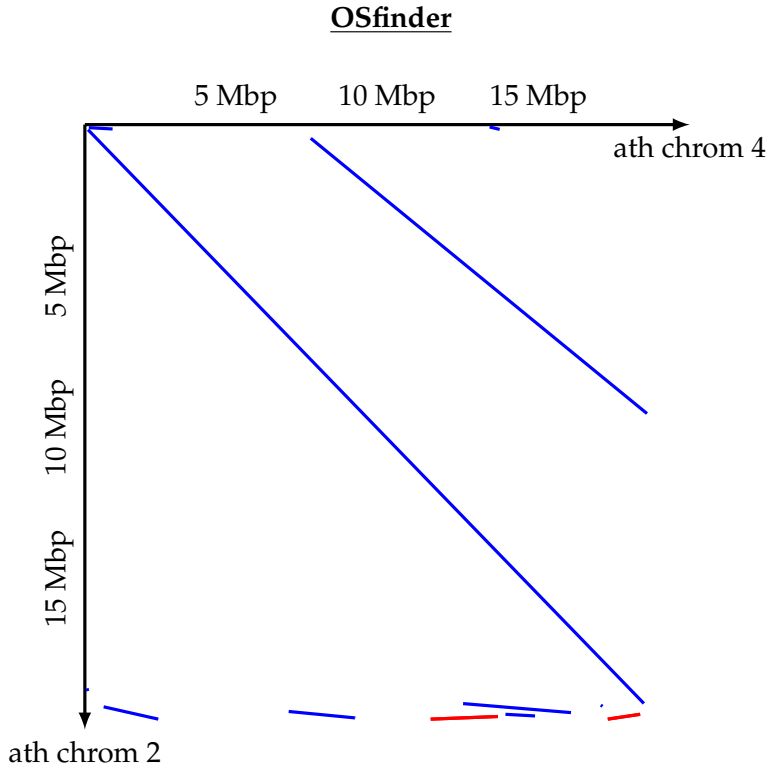


FIGURE 4.1: Dotplot des résultats d'OSfinder dans la recherche de duplications segmentales au niveau protéique, entre deux chromosomes d'*Arabidopsis thaliana*. Les chaînes de polarités positives sont représentées en bleu et celles de polarités négatives en rouge. Les ancres utilisées en données sont visibles dans la figure 2.4 page 26. Ce dotplot montrent qu'OSfinder crée des chaînes avec une pente proche de 0 et/ou qui se chevauchent sur les deux axes.

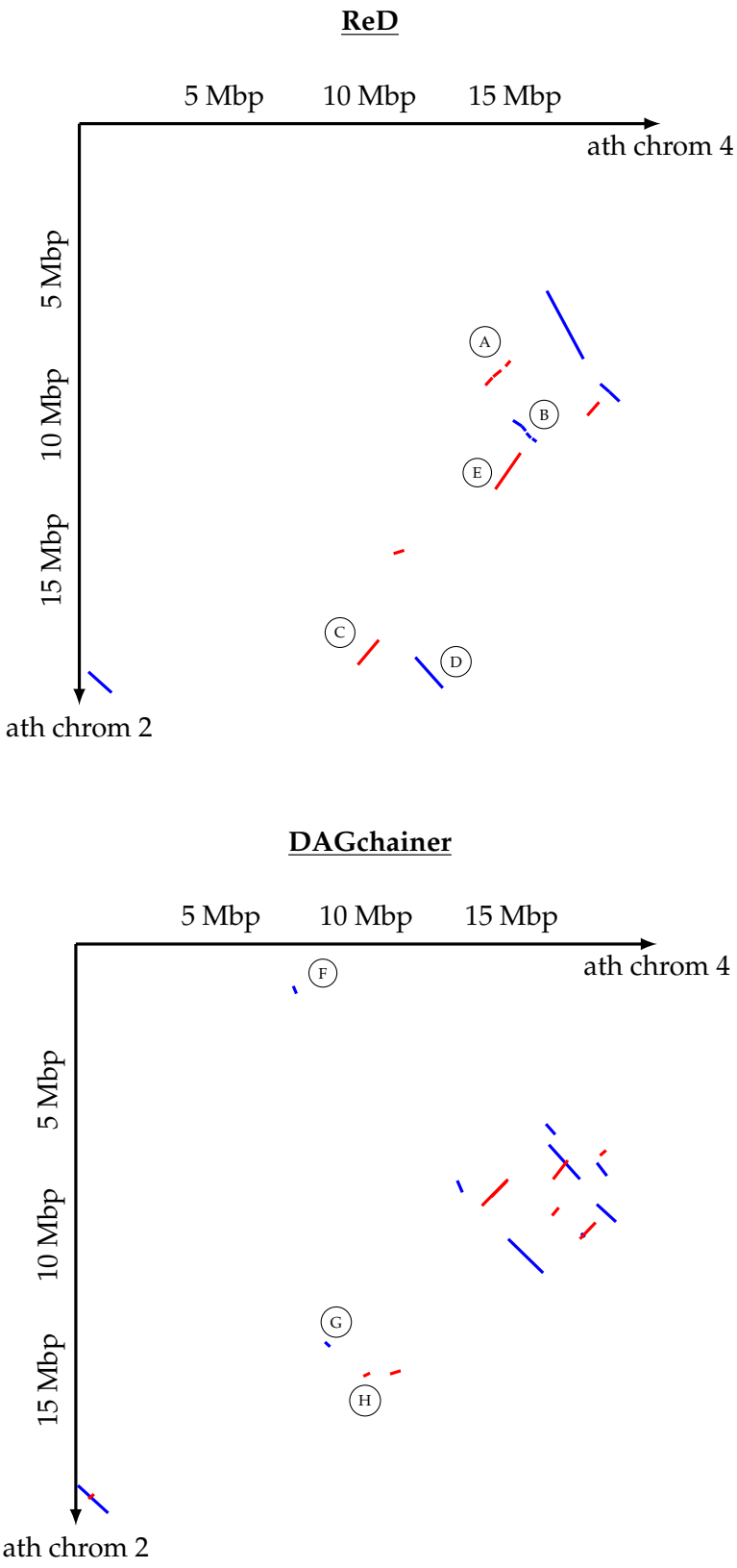


FIGURE 4.2: Comparaison des résultats de ReD et DAGchainer dans la recherche de duplications segmentales au niveau protéique, entre deux chromosomes d'*Arabidopsis thaliana*. Les chaînes de polarités positives sont représentées en bleu et celles de polarités négatives en rouge. Les ancres utilisées en données sont visibles dans la figure 2.4 page 26

4.1.4 Conclusion

Ces résultats ont montré que l'utilisation jointe d'une modélisation sous la forme d'un graphe orienté pondéré et d'un algorithme de recherche de flot de coût minimum dans un graphe de transport, permet d'identifier des régions homologues entre génomes avec des garanties en terme de score et de respect des contraintes qui dépassent celles des outils existants.

De plus le passage d'une méthode gloutonne à une méthode de recherche de flot, plus complexe, reste tout à fait confortable en terme de temps de calcul. Sur une machine récente simple cœur, l'implémentation actuelle de ReD, en C++, prend de l'ordre de 15 minutes pour s'exécuter sur les exemples ci-dessus. Et à la différence des outils existants, ReD permet d'activer ou non les contraintes optionnelles d'ordre et d'orientation présentées dans la section 2.4.1.3 page 30, et reste le seul outil prenant en compte les contraintes spécifiques de non chevauchement qui apparaissent dans la détection de duplications segmentales.

Cependant, la prise en compte des contraintes de non chevauchement dans le critère d'arrêt n'est pas toujours idéale et peut parfois mener à une fragmentation excessive des chaînes dans les génomes fortement dupliqués. Malgré cette caractéristique de ReD, nous avons montré que le nouveau formalisme et son implémentation répondent aux objectifs fixés :

- maximiser le score de l'ensemble des chaînes ;
- créer un ensemble de chaînes cohérent (propriété 1 page 56) ;

et que le résultat, obtenu par ces chaînes, détecte des régions homologues ou des duplications segmentales comparables à celles obtenues par DAGchainer.

Nous allons donc maintenant évaluer les performances de ReD dans un contexte plus difficile et encore peu exploré : l'exploitation d'ancres détectées directement au niveau de l'ADN. En effet au niveau ADN, le signal de similarité est largement plus bruité qu'au niveau des protéines (utilisé traditionnellement pour détecter les ancres). La détection de duplications en tandem (riches en régions fonctionnelles) à l'aide de la séquence d'ADN uniquement, permettrait une utilisation de ReD en amont du processus d'annotation des génomes. Cette *annotation* pourrait alors contribuer au processus de détection de gènes, ou d'autres éléments fonctionnels et fournir une vision non censurée des mécanismes de duplication locale.

La détection de régions dupliquées en tandem à partir de la seule information de la séquence d'ADN fait l'objet de la partie suivante.

4.2 Application à la recherche des duplications en tandem au niveau nucléique

L'objectif de cette section est de montrer que le pipeline ReD Tandem, fondé principalement sur la méthode de chaînage par flot, est capable de reconstruire de longues régions dupliquées en tandem à partir de la seule séquence d'ADN génomique. Pour cela, nous allons confronter les résultats obtenus sur le génome d'*Arabidopsis thaliana* à l'annotation de celui-ci fourni par TAIR10 [4]. Cette annotation sera exploitée pour construire un jeu de références de gènes de protéines dupliqués en tandem d'une part et sera ensuite exploitée pour directement évaluer les sorties de "ReD Tandem".

Ce génome a été choisi car il a l'avantage d'avoir été bien étudié et de posséder beaucoup d'éléments fonctionnels dupliqués en tandem, comme les gènes mais aussi des éléments transposables et des ARNs (Kane *et al.*, 2010), ce qui en fait une très bonne référence pour tester ReD Tandem.

Le jeu de données de référence de gènes de protéines dupliqués en tandem sera mobilisé pour montrer que ReD Tandem est capable de capturer effectivement un signal de duplication en tandem. Nous verrons que plus de 68% des gènes de protéines dupliquées en tandem sur le génome d'*Arabidopsis thaliana*, sont ainsi détectés par ReD Tandem et que les 31% non détectés sont principalement des duplications anciennes.

Ensuite, nous montrerons que, contrairement aux méthodes de détection de duplication en tandem existantes, qui utilisent les gènes de protéines annotés comme source d'information, les régions détectées par ReD tandem ne correspondent pas uniquement à des gènes de protéines. Pour cela nous montrerons que ReD Tandem détecte des régions riches en éléments fonctionnels divers et variés en nous appuyant sur l'annotation existante.

4.2.1 Préambule à l'analyse des résultats

4.2.1.1 Paramètres et données utilisés à l'exécution de ReD Tandem

ReD Tandem est exécuté sur le génome d'*Arabidopsis thaliana* (version : TAIR10) avec les paramètres suivants :

- $distDiag = 50000$. Ce paramètre définit la distance maximum acceptée entre les extrémités les plus proches des deux régions qui composent une ancre ainsi que la distance maximum acceptée entre deux unités consécutives de duplication détectées. 50kb est une distance raisonnable pour un génome compact comme celui d'*Arabidopsis thaliana*.

4. ftp://ftp.arabidopsis.org/home/tair/Genes/TAIR10_genome_release/

- $DistMax_g = 40000$. Ce paramètre, qui définit la distance maximum pour que deux ancres soit considérées comme chaînables, n'est pas fixé automatiquement ici. En effet, dans le cas de recherche de duplications en tandem, on ne compare pas l'ensemble d'un génome contre lui même, mais seulement les séquences à une distance maximum de $distDiag$.
- $\beta = 1$. Ce paramètre (défini par l'équation 3.9 page 75) est utilisé pour calculer le $scoreMin$: le score minimum d'une chaîne. Comme il est possible qu'une chaîne soit composée d'une seule ancre, β est fixé à 1.

4.2.1.2 Notations

Dans la suite, afin de différencier de façon claire le jeu de données de référence et les prédictions de ReD Tandem, nous appellerons :

- *gène tandem* un gène de protéine qui fait partie du jeu de données de référence.
- *unité de ReD* une unité de duplication prédite par ReD Tandem.
- *région tandem* la région qui contient l'ensemble des gènes de protéines d'une même famille du jeu de données de référence.
- *région de ReD* une région de duplication, formée de plusieurs *unités de ReD* et construite par ReD Tandem.

La comparaison des données de référence et des prédictions s'appuie sur la notion de détection. On dira ainsi :

- qu'un élément fonctionnel (référence) est *détecté*, s'il est chevauché sur au moins 70% de sa longueur par une *unité de ReD*.
- qu'une *région tandem* (référence) est *détectée*, si elle est chevauchée sur au moins 70% de sa longueur par une *région de ReD* et qu'au moins un de ses *gènes tandem* est *détecté*.

4.2.2 Les résultats de ReD Tandem en quelques chiffres

À partir de 60 021 ancres nucléiques fournies par Glint (Faraut et Courcelle, 2011), la première exécution de ReD reconstruit 10 290 chaînes composées de 29 817 ancres. Par la suite, l'analyse des ancres et des 10 290 chaînes permet de créer 1 779 composantes connexes composées d'au moins une chaîne. Enfin, à l'aide de ces composantes, ReD Tandem reconstruit :

- 1 718 *régions de ReD* qui couvrent 28.8% du génome (voir tableau 4.5 page suivante), avec une moyenne de 3.2 unités par régions.
- 5 477 *unités de ReD* dupliquées qui couvrent 10.6% du génome.

Arabidopsis	chrom 1	chrom 2	chrom 3	chrom 4	chrom 5	total
Régions de ReD	31.6%	25.7%	30.1%	31.3%	25.0%	28.8%
Unités de ReD	11.6%	9.3%	10.5%	11.7%	9.7%	10.6%

TABLE 4.5: Tableau qui résume le pourcentage de couverture du génome des *régions de ReD* et *unités de ReD*. On remarque que le pourcentage couvert par les unités de duplication reste assez faible et relativement uniforme sur chaque chromosome.

Avant d'analyser plus en détail les régions dupliquées et leurs unités, il est important de signaler que les régions dupliquées reconstruites dans la première exécution de ReD sont composées **en moyenne de** 2.9 **ancres**. Ce résultat montre que le chaînage est un mécanisme essentiel dans le fonctionnement de ReD Tandem pour reconstruire des régions avec perte de la similarité locale.

4.2.3 Comparaison à un jeu de référence, sensibilité

L'objectif de cette section est de montrer que ReD Tandem capture un signal de duplication en tandem, en calculant une sensibilité de détection sur une classe d'éléments fonctionnels bien connue. Pour cela, nous allons utiliser l'annotation des gènes de protéine de TAIR10 pour construire un jeu de données de référence de familles de gènes de protéines dupliquées en tandem. Ainsi il sera possible de comparer les régions et unités construites par ReD Tandem à ce jeu de données de référence, considéré comme un "Gold Standard".

4.2.3.1 Création du jeu de données de référence

À l'aide de l'annotation du génome, on récupère la plus longue protéine de chaque gène [5] de taille $\geq$ 500pb [6]. Toutes ces séquences protéiques sont comparées deux-à-deux en les alignant à l'aide du logiciel BLASTP (Altschul *et al.*, 1990). Parmi tous les alignements construits, seuls les alignements d'une $e - value \leq 10^{-5}$ et qui couvrent au moins 70% de chaque séquence sont conservés. De plus les alignements qui ne respectent pas les conditions de proximité de l'équation 3.10 page 77 sont filtrés, pour ne garder que les gènes de protéines similaires et suffisamment proches, considérés comme dupliqués en tandem.

À partir de ces alignements, un graphe est construit. Les sommets représentent les alignements et une arête est ajoutée entre deux sommets s'ils partagent un gène. Ainsi chaque composante connexe définit une famille de gènes considérée comme dupliquée en tandem.

5. Du fait du mécanisme d'épissage alternatif, un grand nombre de gènes codent pour plusieurs protéines.

6. ReD tandem détecte les unité de duplication de taille $\geq$ 500pb

L'ensemble de ces familles définit notre jeu de référence, utilisé pour évaluer la sensibilité de ReD Tandem.

4.2.3.2 Analyse de la sensibilité de ReD Tandem

À partir de ce jeu de référence, un calcul classique de sensibilité est réalisé. Rappelons que la sensibilité peut se calculer à différents niveaux, ici *gènes tandem* ou *régions tandem*, et est définie comme le pourcentage d'éléments du jeu de référence qui sont effectivement détectés. Les résultats de cette évaluation sont présentés dans le tableau 4.6.

Arabidopsis vs Arabidopsis	Total	Détectés	%
Gène tandem	3 694	2 505	67.8
Région tandem	1 361	930	68.3

TABLE 4.6: Ce tableau montre que plus de 67% des gènes tandem et des régions tandem sont détectés.

Avec son approche agnostique, basée uniquement sur des comparaisons au niveau ADN, les capacités prédictives de ReD Tandem ont de fortes chances d'être affectées par la distance évolutive entre les gènes dupliqués. Pour approfondir l'analyse précédente, nous avons voulu évaluer la capacité de ReD Tandem à détecter les duplications en tandem en fonction de l'âge des duplications. Pour cela, nous avons mesuré une distance évolutive entre chacune des protéines appartenant à une paire de *gènes tandem*. Cette distance est mesurée par le dS (Yang et Nielsen, 2000).

Définition 36 (dS) *C'est le nombre de substitutions silencieuses par site entre deux séquences homologues.*

Il existe plusieurs méthode pour estimer le dS. Pour cette étude c'est la méthode de Yang-Nielsen (Yang et Nielsen, 2000) implémentée dans le programme PAML (Yang, 1997), qui est utilisée.

La figure 4.3 page suivante présente l'histogramme du nombre de *gènes tandem* total d'une part et du nombre de gènes tandem *détectés* en fonction du dS. Cette figure permet d'observer que la majorité des duplications récentes sont effectivement détectées. Elle montre aussi qu'à partir d'une certaine distance évolutive, ReD n'est généralement plus capable de détecter des duplications. Ce n'est pas surprenant, plus la distance évolutive est grande, plus le signal de conservation nucléique est faible, et il est logique de ne plus rien détecter.

Globalement, l'histogramme de la figure 4.3 page suivante permet donc de mieux analyser la sensibilité de 67% au niveau *gènes tandem*. Les gènes de protéines manquants s'expliquent principalement par le manque de signal de conservation au niveau ADN

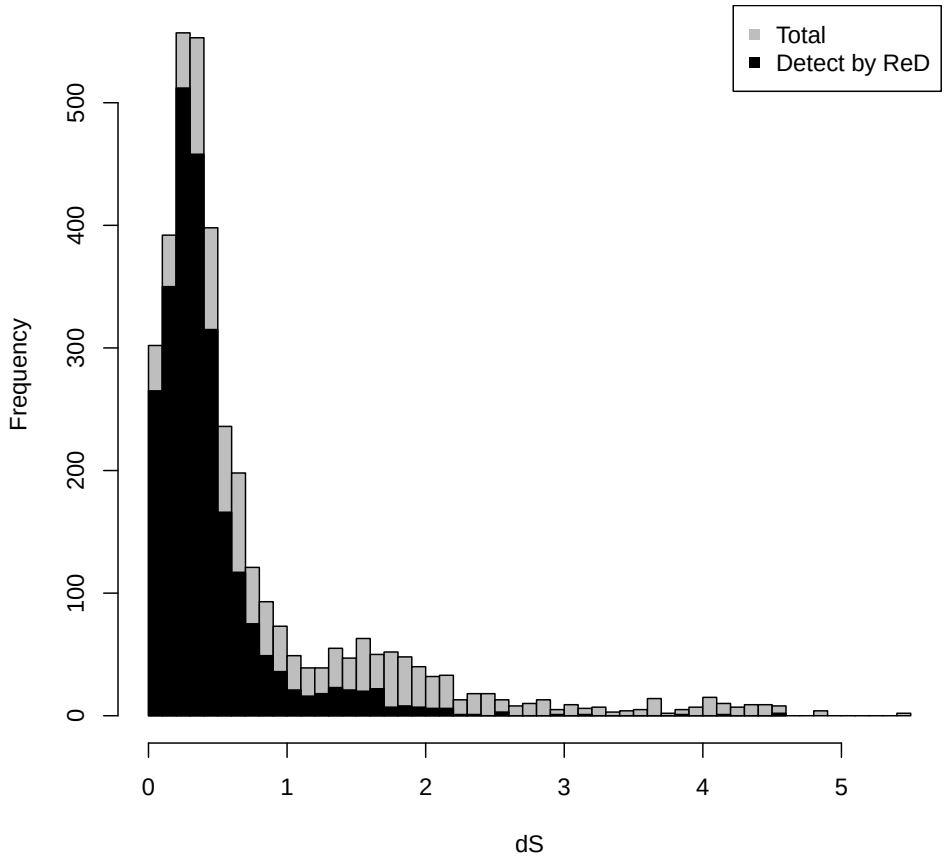


FIGURE 4.3: Pour chaque paire de *gènes tandem*, une distance évolutive est estimée par le *dS*. L'histogramme montre le nombre de paires de gènes tandem total (en gris) et détectés (en noir) en fonction du *dS*. À partir d'une certaine distance évolutive, le signal de conservation au niveau nucléique n'est plus assez fort pour ReD. Avant cette distance, l'histogramme montre que ReD détecte 80.1% des gènes dupliqués ayant un $dS \leq 1$.

quand la distance évolutive augmente. Si l'on restreint le calcul de sensibilité aux duplications suffisamment récentes, celles ayant un $dS \leq 1$ (resp. $dS \leq 0.5$), ReD Tandem détecte alors environ 79% (resp. 85%) des *gènes tandem*.

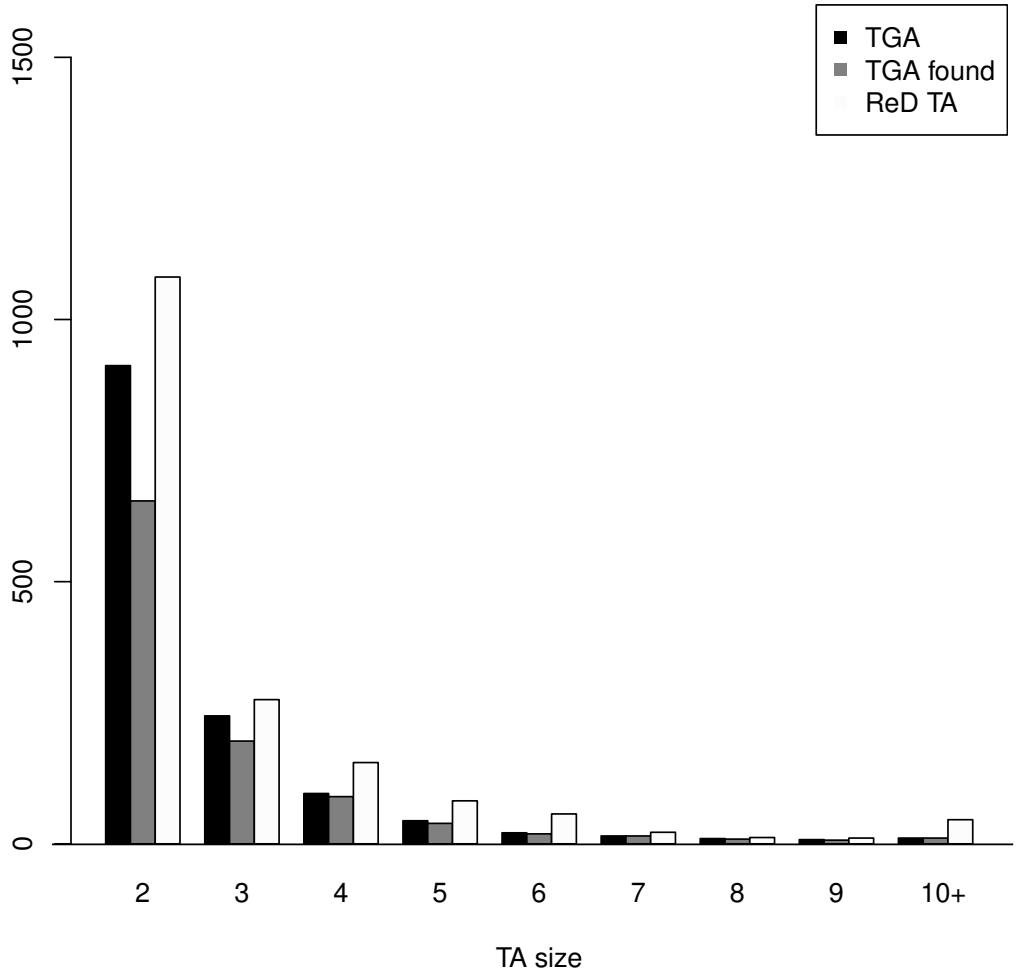


FIGURE 4.4: histogramme montrant les familles de *gènes tandem* (noir), les familles de *gènes tandem* détectées (gris) et les *unités de ReD* (blanc) en fonction de leurs tailles (nombre d'unités de duplication).

Un second facteur qui pourrait affecter le comportement de ReD est la taille des familles dupliquées en tandem. Une dernière analyse, illustrée par l'histogramme de la figure 4.4, montre que ReD Tandem a plus de mal à détecter les familles de *gènes tandem* de taille réduite (en particulier de taille 2). C'est certainement une conséquence du bonus qui est attribué aux ancres qui représentent des régions utilisées dans plusieurs ancres (voir section 3.4.1 page 76).

Pour résumer, alors que les *unités de ReD* ne couvrent que 10.6% du génome (tableau 4.5 page 96), elles détectent 67% des *gènes tandem*, et 79% des *gènes tandem* récemment du-

pliqués (avec un $dS \leq 1$). Un exemple de duplication de ce type avec la prédiction associée de ReD Tandem est présenté dans la figure 4.5.

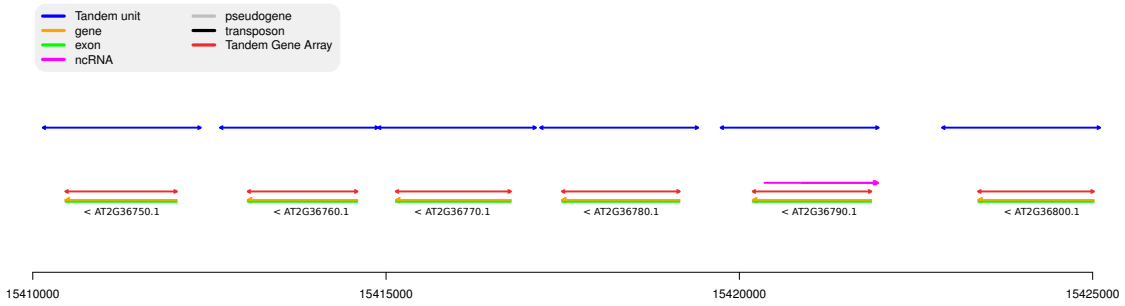


FIGURE 4.5: Chromosome 2:15 410 142-15 425 104. Un exemple typique d'une région dupliquée et de ses unités de duplication sur le chromosome 2. Ici nous avons une famille de 6 gènes de Glycosyltransferases (annotation TAIR10) parfaitement détectée. Il existe 666 *régions de ReD* similaires à celle-ci (régions de ReD dont *toutes* les unités détectent des *gènes tandem*).

Globalement, ces résultats indiquent que ReD Tandem est capable de détecter un signal de duplication en tandem quand il est encore observable au niveau de l'ADN.

4.2.4 Comparaison directe à l'annotation

Les gènes de protéines ne sont pas les seuls éléments fonctionnels à se dupliquer en tandem et ReD Tandem ne détecte pas que des régions et des unités qui correspondent à des *gènes* et des *régions tandem*. De ce fait, cette section a pour objectif d'analyser les résultats de ReD Tandem en les confrontant à l'ensemble des éléments fonctionnels présents dans l'annotation TAIR10 du génome d'*Arabidopsis thaliana*.

Pour commencer, le tableau 4.7 page suivante montre, pour chacune des catégories définies dans TAIR10, le nombre d'éléments fonctionnels présents dans l'annotation et effectivement *détectés*[7] par des *unités de ReD*. Ainsi, nous constatons que ces unités ne détectent pas uniquement des gènes de protéines.

Pour donner un peu de corps à cette table, on peut directement retourner aux prédictions de ReD Tandem. Une visualisation graphique de tous les résultats de ReD tandem est disponible à cette adresse :

- `http://snp.toulouse.inra.fr/~faraut/ReDTandem`

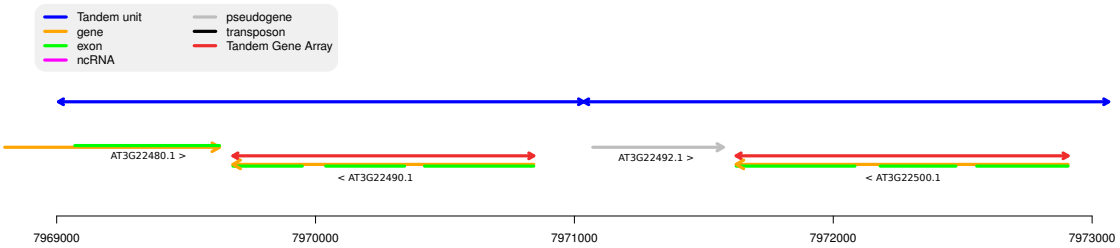
Pour donner un peu de chair aux chiffres précedents et vérifier ainsi que l'approche agnostique de ReD Tandem permet, par exemple, de détecter des duplications en tandem d'éléments fonctionnels non codants, nous considérons maintenant un ensemble d'exemples illustratifs.

7. soit chevauchés sur au moins 70% de leur longueur.

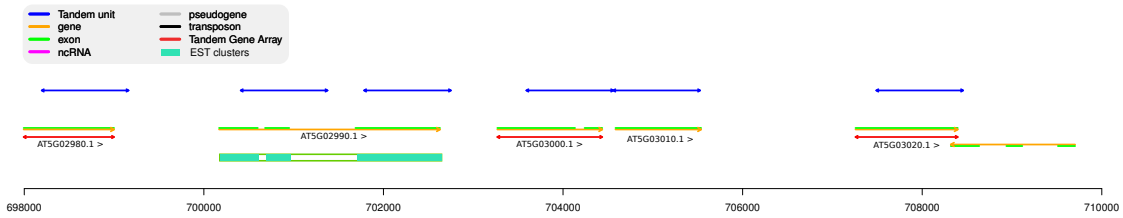
Arabidopsis vs Arabidopsis	Total	Détectés	Détectés (en %)
Gene	27 169	3 462	12.7
Trans. Element gene	3 899	118	3.0
Pseudogene	871	220	25.2
Unknow gene	23	3	13.0
pre-tRNA	631	120	19.0
miRNA	174	19	10.9
snoRNA	71	8	11.3
Other RNA	301	29	9.6

TABLE 4.7: Ce tableau montre, pour chacune des catégories présentes dans l'annotation TAIR10 du génome d'*Arabidopsis thaliana*, le nombre total d'éléments fonctionnels présent dans l'annotation ainsi que le nombre et le pourcentage d'éléments de ce type qui sont chevauchés sur au moins 70% de leurs longueurs par une *unité ReD* (détecté). Les valeurs relatives de ces pourcentages indiquent que certaines catégories semblent être plus souvent détectés en tandem, comme dans le cas des pre-tRNA, souvent organisés en cluster (Kane *et al.*, 2010).

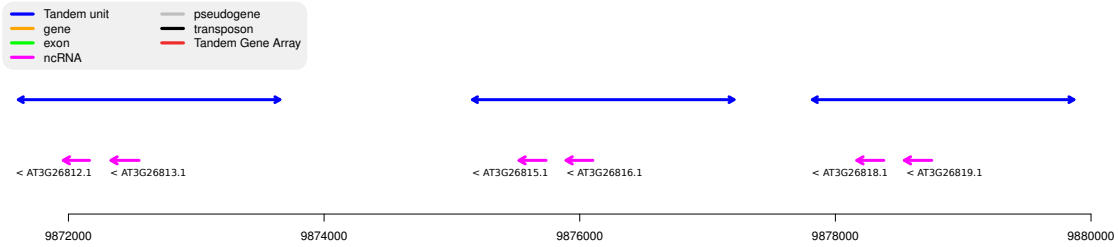
Pseudo-gènes : ReD reconstruit également des régions dupliquées mixant gènes de protéines et pseudo-gènes apparus suite à la perte de fonction codante causée par les mécanismes évolutifs. Un exemple de ce type est détaillé dans la région 7 969 007-7 973 064 du chromosome 3 qui suit. La duplication partielle du gène actif AT3G22480 a sans doute mené à l'apparition du pseudo-gène AT3G22492.



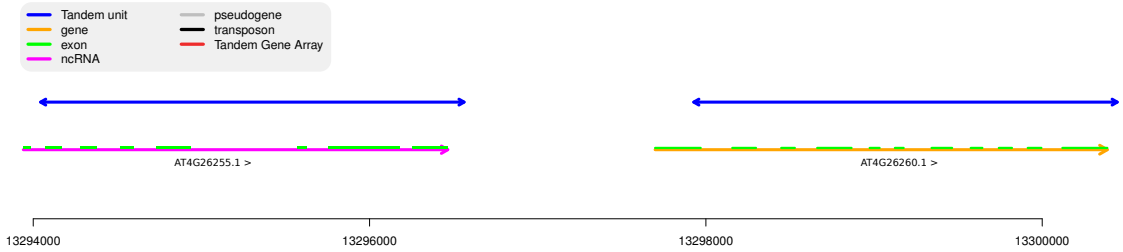
Fusion de gènes : il peut aussi arriver que dans une même région ReD, certaines unités ReD détectent parfaitement des *gènes tandem*, alors que d'autres unités ont fusionné pour créer un nouvel éléments fonctionnel, comme le montre la région 698 194-708 456 du chromosome 5 qui suit. Sur cette figure, nous avons représenté un alignement de clusters d'EST issu du site de Gramène confirmant l'annotation d'un gène unique. Tout comme les autres gènes de cette région tandem, ce gène "fusionné" est annoté comme une protéine de galactose oxidase/kelch repeat.



Gènes d'ARN : une duplication en tandem connue chez *A. thaliana* concerne une famille de trois tRNA, (tRNA^{Tyr}-tRNA^{Tyr}-tRNA^{Ser}) dupliqués 27 fois (Kane *et al.*, 2010), cette famille est détectée en grande partie au travers de 26 unités [8]. D'autres familles d'ARN (ici une paire de microARNs, dupliquée trois fois) sont détectées comme dupliquées en tandem, comme le montre la figure ci dessous, dans la région 9 871 608-9 879 866 du chromosome 3 :



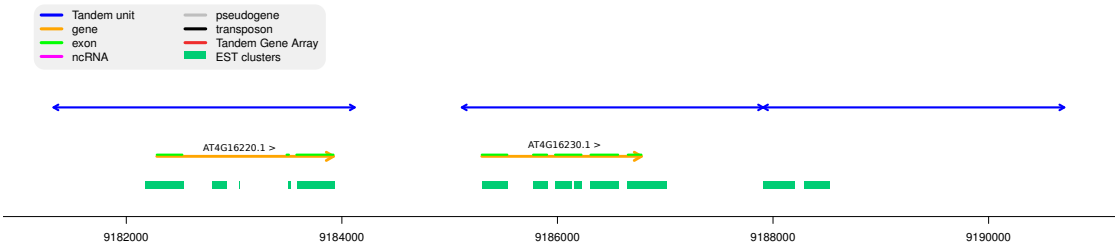
Gène d'ARN long (lncRNA) et gène de protéine : Le nombre important de gènes d'ARN non codant longs dans les génomes n'a été découvert que récement et on en sait assez peu sur leurs origines. Un des scénarios proposés situe l'origine de ces gènes dans la transformation de gènes de protéine en gène d'ARN long (Kaessmann, 2010). Dans la figure ci-dessous, on observe deux *unités de ReD* dans la région 13 294 044-13 300 446 du chromosome 4. L'une détecte un gène de protéine (annotée comme une myo-inositol oxygenase 4) et l'autre détecte un gène d'ARN long (sans fonction associée).



8. http://snp.toulouse.inra.fr/~faraut/ReDTandem/pdf/26/ath1-21268281-21308992.pdf

Selon (Kaessmann, 2010), une telle transformation a déjà été observée chez les mammifères (gène Xist (Duret *et al.*, 2006)) et chez la Drosophile. Cette région est un candidat naturel pour les plantes.

Unité de ReD orpheline : il arrive qu'une région ReD possède à la fois des unités qui chevauchent un élément fonctionnel et des unités qui ne chevauchent aucune annotation (appelé unité orpheline). Parmi les 5 573 *unités de ReD*, 1 438 sont orphelines. Un exemple d'une telle situation est visible dans la région 9 181 325-9 190 702 du chromosome 4 représentée ci-dessous. Les deux gènes de cette région sont annotés comme des "GDSL-like Lipase/Acylhydrolase superfamily protein".



Dans ce cas, il est assez vraisemblable de penser que cette région correspond à un gène ou pseudo-gène qui est absent de l'annotation. Dans cet exemple, cette hypothèse est appuyée par l'existence de similarités de cette région avec des clusters d'EST (similarités issus du site de Gramène et visualisés sur la figure), et d'une prédiction de FGENESH (non visualisée ici [9]).

Région de ReD orpheline : une région de ReD orpheline est une région qui ne possède que des unités orphelines. Parmi les 1 741 *régions de ReD*, 465 sont orphelines (voir tableau 4.8 page suivante). Notamment, la plus grande *région de ReD* en termes de nombre d'unités dupliquées (70 *unités* d'une taille d'environs 600 nucléotides) est une région orpheline [10]. Il est intéressant de remarquer que cette région a été récemment identifiée comme un *hot-spot* de CNV (Copy Number Variation [11]) (DeBolt, 2010).

Une analyse plus générale de ces régions orphelines a été menée pour voir s'il s'agit de régions de nature spécifique. Le résultat de cette analyse est donné dans le tableau 4.8 page suivante. Il apparaît que ces régions contiennent des unités sensiblement plus courtes que les autres.

Pour résumer, ces analyses montrent que ReD Tandem, à l'aide des ancres nucléiques, capture un signal de régions génomiques issues potentiellement de duplication en tan-

9. Voir http://www.gramene.org/Arabidopsis_thaliana/Location/View?db=core&r=4%3A9180387.2-9191639.8

10. Région 15 088 006-15 430 870 du chromosome 1, voir http://snp.toulouse.inra.fr/~faraut/ReDTandem/70.html

11. CNV : désigne une région dont le nombre de copies d'un même gène ou d'une sous-séquence du génome est variable entre les individus de la même espèce.

	Nb régions	Nb unités	Longueur moyenne des unités (pb)
Région ReD non orpheline	1 253	3 980	2 823
Région ReD orpheline	465	1 497	1 100

TABLE 4.8: Ce tableau donne le nombre de régions ReD orphelines et non orphelines. Pour chacun de ces deux types de régions, le nombre total d'unités de ces régions ainsi que la longueur moyenne de leurs unités par région est indiqué. Il est intéressant de remarquer que la longueur moyenne des unités est plus de deux fois plus longue dans les régions non orphelines, ce qui indique que les régions ReD orphelines sont sans doute, en partie, composées de régions de nature spécifique.

dem avec une spécificité de 67% au niveau unité des régions. Nous avons montré que ce pourcentage dépasse les 80% quand on s'intéresse au duplications récentes ($dS \leq 1$). De plus nous venons de voir que ces régions ne sont pas uniquement composées de gènes de protéines (voir tableau 4.7 page 101) et détectent nombre d'éléments que les "méthodes classiques" qui exploitent activement une annotation, sont incapables de détecter.

4.2.5 Conclusion

À l'aide de la construction d'un jeu de données de référence de gènes de protéines dupliqués en tandem, nous avons montré que ReD Tandem est capable de détecter des duplications en tandem à l'aide d'ancres nucléiques, sans exploiter d'annotation, avec de bons résultats dès lors que les distances évolutives ne sont pas trop grandes.

De plus, nous avons vu que le choix de détecter les duplications en tandem au niveau ADN, permet de détecter d'autres éléments fonctionnels comme des pseudo-gènes et des gènes d'ARN. Cependant, près de 26% des *unités de ReD* ne chevauchent aucun élément fonctionnel. Ces unités représentent potentiellement des séquences génomiques qui auraient échappées à l'annotation (Aubourg *et al.*, 2007).

Au niveau de l'implémentation, ReD Tandem est programmé en C++ (comme ReD) et malgré l'utilisation de TBLASTX, le temps d'exécution sur le génome d'*Arabidopsis thaliana* est d'environ 4 heures avec une machine récente simple cœur. Ce qui est acceptable pour un utilisateur.

Cependant, la méthode reste perfectible : un peu moins de 20% des gènes dupliqués récemment ne sont pas détectés. Certainement parce que la famille de ses gènes est de petite taille (2 ou 3 gènes). La cause de cette défaillance est probablement le bonus ajouté au score des ancres, qui est par ailleurs une aide précieuse (voire indispensable) pour détecter correctement les grandes familles de duplication. Une solution envisagée serait donc de dissocier la recherche de grandes familles de duplications, de la recherche de petites familles de duplications. Ainsi, il sera possible d'adapter les fonctions de score

aux spécificités de chaque cas.

Une autre étape de la méthode est à l'origine de défaillances, il s'agit de la sélection de l'unité de référence. Pour l'avoir expérimenté, nous savons que pour certaines *régions de ReD*, il est possible d'obtenir de meilleurs résultats avec un autre choix d'unité de référence. Intuitivement on comprend que choisir une seule unité, parmi toute celles créées par la première excécution de ReD, n'est pas optimum. Une piste envisagée serait d'utiliser plusieurs unités de références.

En résumé, la méthode proposée pour reconstruire des duplications en tandem semble prometteuse. Il faut maintenant passer à l'analyse de d'autres génomes pour confirmer.

Chapitre 5

Conclusion et perspectives

Ces dernières décennies ont vu la recherche scientifique bouleversée par l'explosion de la génomique et plus récemment par l'exploitation des données produites par le séquençage à haut débit. L'analyse et le traitement des séquences génomiques représentent un enjeu colossal dans bien des domaines et en particulier dans la compréhension de l'évolution des génomes. Une part majeure de cette tâche revient à la bioinformatique, concernant notamment l'étape d'annotation des séquences génomiques et la recherche de régions homologues et plus particulièrement dupliquées en interne.

L'objectif du travail réalisé dans le cadre de cette thèse consistait à développer une nouvelle méthode de recherche de régions dupliquées en tandem à partir de la séquence d'ADN uniquement. Dans l'objectif de mettre en lumière des régions ayant un fort potentiel informatif, puisque conservées, sans être dépendant des informations données par l'annotation.

Dans un premier temps, à travers une étude approfondie de l'état de l'art de la recherche d'homologie, nous avons pu mettre en évidence les limitations des méthodes existantes et proposer un axe de travail pour apporter une contribution originale au domaine de recherche :

- L'analyse de l'évolution des méthodes de reconstruction d'un ensemble de régions dupliquées révèle que, du fait de la complexité du problème, la reconstruction de ces régions se fait principalement à l'aide de choix indépendant pour chaque région, voire des choix locaux pour les plus anciennes. Et ce, bien que l'ensemble des régions dupliquées décrivent une histoire globale. Il est aussi important de remarquer que ces choix sont réalisés par un processus itératif glouton, comme l'exécution d'un algorithme de programmation dynamique (*le plus court chemin*).
- L'analyse des données utilisées pour reconstruire les régions dupliquées montre que les méthodes existantes dépendent fortement de l'annotation. Or, l'annotation est un processus complexe, long et coûteux et qui demande de nombreuses analyses et rectifications pour arriver à un état stable, c'est-à-dire sans erreur ou très peu. De plus, les régions annotées ne représentent qu'une petite partie du génome.

La synthèse qui se dégage de cette étude est qu'une méthode de reconstruction d'un ensemble de régions dupliquées doit, pour garantir une efficacité, faire des choix en connaissant l'environnement global pour reconstruire une histoire évolutive globalement cohérente. De plus, une méthode capable de reconstruire des régions dupliquées sans annotation aurait un sérieux avantage sur les autres pour découvrir de nouvelles régions dupliquées.

D'autres questions de recherche possibles apparaissent également concernant les mécanismes évolutifs, qui détériorent le signal de conservation, acceptés pendant la re-

construction des régions dupliquées : les changements d'ordre et d'orientation sont-ils acceptés dans une région dupliquées ? Si oui, comment empêcher la création de régions farfelues ?

Après avoir réuni les méthodes de reconstruction de type "chaînage" sous un formalisme commun, qui a servi de base au développement de la nouvelle méthode, nous nous sommes intéressés aux contraintes qu'un ensemble de régions dupliquées doit respecter pour décrire une histoire évolutive cohérente. Ces contraintes peuvent être résumées en une phrase : deux séquences génomiques ne peuvent avoir deux liens de parenté différents.

Ces contraintes sont capitales tout d'abord pour justifier des résultats obtenus dans un domaine ou l'on ne connaît pas la réalité, mais aussi comme source d'information supplémentaire. En effet, si un ensemble de régions n'est pas cohérent, c'est qu'au moins une erreur a été commise pendant la reconstruction.

Du coup la reconstruction d'un ensemble de régions dupliquées par des algorithmes gloutons est un handicap. En effet, le principe de ces algorithmes est de ne jamais revenir sur un choix fait précédemment et si un ensemble de régions est incohérent, il n'est plus possible de revenir en arrière pour corriger l'erreur.

En conséquence, nous avons développé un nouvel algorithme itératif de chaînage, non glouton, qui maximise un critère de score global. Pour cela, nous avons développé une approche originale basée sur la théorie des flots de coût minimum, qui possède la caractéristique de remettre en question tous les choix précédemment faits, à chaque nouvelle étape.

Avant d'aller plus loin dans notre objectif, nous avons testé cette nouvelle méthode dans la recherche d'un ensemble régions homologues, en général, et dans la recherche d'un ensemble de duplications segmentales, en particulier. La reconstruction de ces régions s'est faite à partir de similarités protéiques, ce qui nécessite une annotation préalable du/des génome(s). Ce cadre bien connu des méthodes existantes a permis de comparer les méthodes et de montrer que l'approche globale, à l'aide de flot, possède des caractéristiques intéressantes et est la seule à garantir un ensemble cohérent de régions.

Encouragé par ces résultats, nous avons testé notre méthode dans la reconstruction de régions dupliquées en tandem à partir de similarités nucléiques, ne nécessitant pas d'annotation.

Comme nous l'avons vu dans l'introduction, les duplications en tandem sont fréquentes comparées à d'autres évènements évolutifs, comme les mutations des nucléotides. Et il est fort probable qu'elles soient à l'origine des différences observées, inter et intra espèce, et du nombre de copies de chaque gène. Ainsi la reconstruction de régions issues de duplications en tandem au niveau nucléique, devrait être riche en gènes, mais pas uniquement.

De ce fait, après avoir adapté notre méthode à ce tout nouveau problème, très particulier, nous l'avons testé sur le génome d'*Arabidopsis thaliana*. C'est un génome de plante de référence notamment connu pour ses nombreuses duplications internes. Les résultats ont montré que parmi les régions détectées, environ 75% étaient des éléments fonctionnels (comme des gènes codant pour des protéines ou des ARNs par exemple) et les derniers 25% étaient des régions non connues. De plus, nous avons montré que 79% des gènes codants des protéines supposées dupliqués en tandem relativement récemment étaient parfaitement détectés par notre méthode.

Il reste cependant à montrer que la détection des régions dupliquées à l'aide d'information nucléique, en particulier issues de duplications en tandem, apporte une information significative par rapport à la recherche de régions dupliquées au niveau protéique.

Cette approche a mis en lumière des régions non connues d'un génome, dont l'intérêt reste à démontrer.

Ces régions permettent-elles de mieux comprendre les mécanismes d'évolution et leurs impacts sur les génomes ?

Une analyse approfondie des régions détectées doit permettre de répondre à ces questions. Notamment les régions détectées comme dupliquées mais avec une annotation différente doivent être riches en informations et en enseignements, remettant en question la qualité et la fiabilité de l'annotation.

Si tel est le cas, le fait de savoir que ces régions sont dupliquées permet-il de corriger l'erreur d'annotation ? Si oui, peut-on utiliser ces régions comme source d'informations dans les logiciels d'annotation et ainsi inverser le processus actuel qui consiste à annoter les génomes avant de chercher les régions dupliquées ?

Ce sont autant de nouvelles questions que posent les résultats obtenus au cours de cette thèse.

Bibliographie

ABOUELHODA, M. I. et OHLEBUSCH, E. : Chaining algorithms for multiple genome comparison. *J. Discrete Algorithms*, 3(2-4):321–341, 2005.

AHUJA, R. K., MAGNANTI, T. L. et ORLIN, J. B. : *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, February 1993. ISBN 013617549X. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/013617549X.

ALTSCHUL, S. F., GISH, W., MILLER, W., MYERS, E. W. et LIPMAN, D. J. : Basic local alignment search tool. *J Mol Biol*, 215(3):403–10, 1990. URL http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?cmd=prlinks&dbfrom=pubmed&retmode=ref&id=2231712.

AUBOURG, S., MARTIN-MAGNIETTE, M. L., BRUNAUD, V., TACONNAT, L., BITTON, F., BALZERGUE, S., JULLIEN, P. E., INGOUFF, M., THAREAU, V., SCHIEX, T., LECHARNY, A. et RENOU, J. P. : Analysis of CATMA transcriptome data identifies hundreds of novel functional genes and improves gene models in the Arabidopsis genome. *BMC Genomics*, 8:401, 2007. URL http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?cmd=prlinks&dbfrom=pubmed&retmode=ref&id=17980019.

AUDEMARD, E., T., F. et SCHIEX, T. : Détection de régions génomiques homologues par un algorithme de flot avec couts. *In Actes de ROADEF'2010*, pages 125–140, Toulouse, France, 2 2010a.

AUDEMARD, E., T., F. et SCHIEX, T. : Détection de régions génomiques homologues par un algorithme de flot avec couts. Présentation faite dans le cadre du workshop Gtseq2010, Janvier 2010b.

AUDEMARD, E., T., F. et SCHIEX, T. : Détection de régions homologues au niveau génique et de duplications en tandem au niveau adn par un algorithme de flot avec cout. Invité à présenter mes travaux dans la journée satellite de JOBIM2010 : Annotations des génomes et génomique comparée, Septembre 2010c.

BAILEY, J. A., YAVOR, A. M., MASSA, H. F., TRASK, B. J. et EICHLER, E. E. : Segmental duplications: organization and impact within the current human genome project assembly. *Genome Res*, 11(6):1005–17, 2001. URL http:

//eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?cmd=prlinks&dbfrom=pubmed&retmode=ref&id=11381028.

BATZOGLOU, S., PACHTER, L., MESIROV, J. P., BERGER, B. et LANDER, E. S. : Human and Mouse Gene Structure: Comparative Analysis and Application to Exon Prediction. *Genome Research*, 10(7):950–958, July 2000. URL http://dx.doi.org/10.1101/gr.10.7.950.

BENSON, G. : A space efficient algorithm for finding the best nonoverlapping alignment score. *Theoretical Computer Science*, 145(1-2):357–369, 1995.

BENSON, G. : Tandem repeats finder: a program to analyze DNA sequences. *Nucl. Acids Res.*, 27(2):573–580, January 1999. ISSN 0305-1048. URL http://dx.doi.org/10.1093/nar/27.2.573.

BRIDGES, C. B. : The bar "gene" a duplication. *Science*, 83(2148):210–211, February 1936. URL http://www.sciencemag.org/cgi/content/citation/83/2148/210.

BRUDNO, M., MALDE, S., POLIAKOV, A., DO, C. B., COURONNE, O., DUBCHAK, I. et BATZOGLOU, S. : Glocal alignment: finding rearrangements during alignment. *Bioinformatics*, 19 Suppl 1, 2003. ISSN 1367-4803. URL http://view.ncbi.nlm.nih.gov/pubmed/12855437.

BUSACKER, R. et GOWEN, P. : A procedure for determining minimal-cost network flow patterns. *In ORO Technical Report 15*, 1961.

CALABRESE, P. P., CHAKRAVARTY, S. et VISION, J. T. : Fast identification and statitistical evaluation of segmental homologies in comparative maps. *Bioinformatics*, 19:i74–i80, January 2003.

CANNON, S. B., KOZIK, A., CHAN, B., MICHELMORE, R. et YOUNG, N. D. : Diaghunter and genopix2d: programs for genomic comparisons, large-scale homology discovery and visualization. *Genome Biology*, pages 853–860, September 2003.

CORMEN, T. H., STEIN, C., RIVEST, R. L. et LEISERSON, C. E. : *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd édition, 2001. ISBN 0070131511.

DARLING, A. C. E., MAU, B., BLATTNER, F. R. et PERNA, N. T. : Mauve: Multiple Alignment of Conserved Genomic Sequence With Rearrangements. *Genome Research*, 14 (7):1394–1403, July 2004. ISSN 1088-9051. URL http://dx.doi.org/10.1101/gr.2289704.

DEBOLT, S. : Copy number variation shapes genome diversity in arabidopsis over immed iate family generational scales. *Genome Biology and Evolution*, 2:441, 2010.

DELCHER, A. L., KASIF, S., FLEISCHMANN, R. D., PETERSON, J., WHITE, O. et SALZBERG, S. L. : Alignment of whole genomes. *Nucleic acids research*, 27(11):2369–2376, June 1999. ISSN 0305-1048. URL http://dx.doi.org/10.1093/nar/27.11.2369.

DELGRANGE, O. et RIVALS, E. : STAR: an algorithm to Search for Tandem Approximate Repeats. *Bioinformatics*, 20(16):2812–2820, 2004. URL http://bioinformatics.oxfordjournals.org/content/20/16/2812.abstract.

DEMPSTER, A. P., LAIRD, N. M. et RUBIN, D. B. : Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246. URL http://web.mit.edu/6.435/www/Dempster77.pdf.

DEMUTH, J. P., BIE, T. D., STAJICH, J. E., CRISTIANINI, N. et HAHN, M. W. : The evolution of mammalian gene families. *PLoS ONE*, 1(1):e85+, December 2006. ISSN 1932-6203. URL http://dx.doi.org/10.1371/journal.pone.0000085.

DESPONS, L., BARET, P. V., FRANGEUL, L., LOUIS, V. L., DURRENS, P. et SOUCIET, J. L. : Genome-wide computational prediction of tandem gene arrays: application in yeasts. *BMC Genomics*, 11:56, 2010. URL http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?cmd=prlinks&dbfrom=pubmed&retmode=ref&id=20092627.

DEWEY, C. N. et PACHTER, L. : Evolution at the nucleotide level: the problem of multiple whole-genome alignment. *Hum Mol Genet*, 15 Spec No 1:R51–R56, Apr 2006. URL http://dx.doi.org/10.1093/hmg/ddl056.

DINIC, E. A. : Algorithm for solution of a problem of maximum flow in networks with power estimation. *Soviet Mathematics Doklady*, 11:1277–1280, 1970.

DINITZ, Y. : Dinitz' Algorithm: The Original Version and Even's Version. *In Theoretical Computer Science*, volume 3895 de *Lecture Notes in Computer Science*, chapitre 10, pages 218–240. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2006.

DURET, L., CHUREAU, C., SAMAIN, S., WEISSENBACH, J. et AVNER, P. : The xist rna gene evolved in eutherians by pseudogenization of a protein-coding gene. *Science*, 312(5780):1653, 2006.

EDMONDS, J. et KARP, R. M. : Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972. ISSN 0004-5411.

ENRIGHT, A. J., VAN DONGEN, S. et OUZOUNIS, C. A. : An efficient algorithm for large-scale detection of protein families. *Nucl. Acids Res.*, 30(7):1575–1584, April 2002. ISSN 1362-4962. URL http://dx.doi.org/10.1093/nar/30.7.1575.

FARAUT, T. et COURCELLE, E. : Glint, a versatile program for genome comparison. En préparation, 2011.

FISCHETTI, V., LANDAU, G., SCHMIDT, J. et SELLERS, P. : Identifying periodic occurrences of a template with applications to protein structure. *Information Processing Letters*, 45(1):11–18, 1993.

FORD, L. R. et FULKERSON, D. R. : *Flows in Networks*. Princeton University Press, 1962.

FRIEDMAN, R. et L., H. A. : Two patterns of genome organization in mammals: the chromosomal distribution of duplicate genes in human and mouse. *Mol. Biol. Evol.*, 21:1008–1013, February 2004.

GOLDBERG, A. V. et TARJAN, R. E. : Finding minimum-cost circulations by canceling negative cycles. *J. ACM*, 36(4):873–886, 1989. ISSN 0004-5411.

GONDRAN, M. et MINOUX, M. : *Graphes et algorithmes*. Lavoisier, 4ème édition, 2009.

HAAS, B. J., DELCHER, A. L., WORTMAN, J. R. et SALZBERG, S. L. : Dagchainer: a tool for mining segmental genome duplications and synteny. *Bioinformatics*, 20 (18):3643–3646, December 2004. ISSN 1367-4803. URL http://view.ncbi.nlm.nih.gov/pubmed/15247098.

HACHIYA, T., OSANA, Y., POPENDORF, K. et SAKAKIBARA, Y. : Accurate identification of orthologous segments among multiple genomes. *Bioinformatics*, 25, February 2009.

HAMPSON, S., MCLYSAGHT, A., GAUT, B. et BALDI, P. : Lineup: statistical detection of chromosomal homology with application to plant comparative genomics. *Genome research*, 13(5):999–1010, May 2003. ISSN 1088-9051. URL http://dx.doi.org/10.1101/gr.814403.

HAMPSON, S. E., GAUT, B. S. et BALDI, P. : Statistical detection of chromosomal homology using shared-gene density alone. *Bioinformatics*, 21(8):1339–1348, April 2005. ISSN 1367-4803. URL http://view.ncbi.nlm.nih.gov/pubmed/15585535.

HOHL, M., KURTZ, S. et OHLEBUSCH, E. : Efficient multiple genome alignment. *Bioinformatics*, 18(suppl_1):S312–320, July 2002. URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/18/suppl%_1/S312.

HUGHES, A. L. et YEAGER, M. : Comparative evolutionary rates of introns and exons in murine rodents. *Molecular Evolution*, 45:125–130, 1997.

JORDA, J. et KAJAVA, A. V. : T-reks. *Bioinformatics*, 25:2632–2638, October 2009. ISSN 1367-4803. URL http://dx.doi.org/10.1093/bioinformatics/btp482.

JUNGNICKEL, D. : *Graphs, Networks and Algorithms*. Springer, 3rd édition, 2007.

KAESSMANN, H. : Origins, evolution, and phenotypic impact of new genes. *Genome research*, 20(10):1313, 2010.

KAESSMANN, H., VINCKENBOSCH, N. et LONG, M. : RNA-based gene duplication: mechanistic and evolutionary insights. *Nat Rev Genet*, 10(1):19–31, 2009. URL http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?cmd=prlinks&dbfrom=pubmed&retmode=ref&id=19030023.

KANE, J., FREELING, M. et LYONS, E. : The evolution of a high copy gene array in arabidopsis. *Journal of Molecular Evolution*, 10(6):531–544, 2010. URL http://www.springerlink.com/content/v51q6rr7924153x7/.

KOLPAKOV, R., BANA, G. et KUCHEROV, G. : mreps: efficient and flexible detection of tandem repeats in DNA. *Nucl. Acids Res.*, 31(13):3672–3678, July 2003. URL http://dx.doi.org/10.1093/nar/gkg617.

LAJOIE, M., BERTRAND, D., EL-MABROUK, N. et GASCUEL, O. : Duplication and inversion history of a tandemly repeated genes family. *Journal of Computational Biology*, 14 (4):462–478, July 2007. ISSN 1088-9051. URL http://www.liebertonline.com/doi/abs/10.1089/cmb.2007.A007.

LI, W. H., GU, Z., CAVALCANTI, A. R. et NEKRUTENKO, A. : Detection of gene duplications and block duplications in eukaryotic genomes. *J Struct Funct Genomics*, 3:27–34, 2003. URL http://view.ncbi.nlm.nih.gov/pubmed/12836682.

LYNCH, M. : *The Origins of Genome Architecture*. W.H. Freeman & Company, March 2007. ISBN 0878934847. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0878934847.

LYNCH, M. et CONERY, J. S. : The evolutionary fate and consequences of duplicate genes. *Science*, 290(5494):1151–1155, November 2000a. ISSN 0036-8075. URL http://dx.doi.org/10.1126/science.290.5494.1151.

LYNCH, M. et CONERY, J. S. : The evolutionary fate and consequences of duplicate genes. *Science*, 290(5494):1151–1155, November 2000b. ISSN 0036-8075. URL http://dx.doi.org/10.1126/science.290.5494.1151.

MAHMOOD, K., KONAGURTHU, A. S., SONG, J., BUCKLE, A. M., WEBB, G. I. et WHISSTOCK, J. C. : Egm: encapsulated gene-by-gene matching to identify gene orthologs and homologous segments in genomes. *Bioinformatics*, 26(17):2076–2084, September 2010. URL http://dx.doi.org/10.1093/bioinformatics/btq339.

MARQUES-BONET, T., KIDD, J. M., VENTURA, M., GRAVES, T. A., CHENG, Z., HILLIER, L. W., JIANG, Z., BAKER, C., MALFAVON-BORJA, R., FULTON, L. A., ALKAN, C., AKSAY, G., GIRIRAJAN, S., SISWARA, P., CHEN, L., CARDONE, M. F., NAVARRO, A., MARDIS, E. R., WILSON, R. K. et EICHLER, E. E. : A burst of segmental duplications in the genome of the african great ape ancestor. *Nature*, 457(7231):877–881, February 2009. URL http://dx.doi.org/10.1038/nature07744.

MAYR, E. : *Systematics and the origin of species*. Columbia University Press, New York, 1942.

MORGENSTERN, B. : A simple and space-efficient fragment-chaining algorithm for alignment of dna and protein sequences. *Appl. Math. Lett.*, 15(1):11–16, 2002.

MYERS, G. et MILLER, W. : Chaining multiple-alignment fragments in sub-quadratic time. *In SODA*, pages 38–47, 1995.

NAGASWAMY, U. et FOX, G. E. : RNA ligation and the origin of tRNA. *Orig Life Evol Biosph*, 33(2):199–209, 2003. URL http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?cmd=prlinks&dbfrom=pubmed&retmode=ref&id=12967267.

NEEDLEMAN, S. B. et WUNSCH, C. D. : A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, March 1970. ISSN 0022-2836. URL http://view.ncbi.nlm.nih.gov/pubmed/5420325.

NOÉ, L. et KUCHEROV, G. : Yass: enhancing the sensitivity of dna similarity search. *Nucleic Acids Res*, 33(Web Server issue), July 2005. ISSN 1362-4962. URL http://view.ncbi.nlm.nih.gov/pubmed/15980530.

NORRIS, B. J. et WHAN, V. A. : A gene duplication affecting expression of the ovine asip gene is responsible for white and black sheep. *Genome Res*, pages 1282–1293, August 2008.

OHNO, S. : *Evolution by gene duplication*. Springer-Verlag, 1970. ISBN 0045750157. URL http://www.amazon.co.uk/exec/obidos/ASIN/0045750157/citeulike00-21.

PEVZNER, P. et TESLER, G. : Genome rearrangements in mammalian evolution: Lessons from human and mouse genomes. *Genome Research*, 13(1):37–45, January 2003. URL http://dx.doi.org/10.1101/gr.757503.

SCHRIDER, D. R. et HAHN, M. W. : Gene copy-number polymorphism in nature. *Proc Biol Sci*, 277(1698):3213–21, 2010. URL http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?cmd=prlinks&dbfrom=pubmed&retmode=ref&id=20591863.

SHOJA, V. et ZHANG, L. : A roadmap of tandemly arrayed genes in the genomes of human, mouse, and rat. *Molecular Biology and Evolution*, 23(11):2134–2141, November 2006. ISSN 0737-4038. URL http://dx.doi.org/10.1093/molbev/msl085.

SIMILLION, C., VANDEPOELE, K. et Van de PEER, Y. : Recent developments in computational approaches for uncovering genomic homology. *Bioessays*, 26(11):1225–1235, November 2004. ISSN 0265-9247. URL http://dx.doi.org/10.1002/bies.20127.

SLEATOR, D. D. et TARJAN, R. E. : A data structure for dynamic trees. *Journal of Computer and System Sciences*, 24:362–391, 1983.

SMIT, A. F. : Interspersed repeats and other mementos of transposable elements in mammalian genomes. *Curr Opin Genet Dev*, 9(6):657–63, 1999. URL http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?cmd=prlinks&dbfrom=pubmed&retmode=ref&id=10607616.

SODERLUND, C., NELSON, W. et et AL, A. S. : Symap: A system for discovering and viewing syntenic regions of fpc maps. *Genome research*, 16:1159–1168, December 2006.

TARDOS, E. : A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5(3):247–255, 1985. ISSN 0209-9683.

THE CHIMPANZEE SEQUENCING AND ANALYSIS CONSORTIUM : Initial sequence of the chimpanzee genome and comparison with the human genome. *Nature*, 437

(7055):69–87, September 2005. ISSN 0028-0836. URL http://dx.doi.org/10.1038/nature04072.

URICARU, R., MANCHERON, A. et RIVALS, E. : Novel definition and algorithm for chaining fragments with proportional overlaps. *In Proceedings of the 2010 international conference on Comparative genomics*, RECOMB-CG'10, pages 161–172, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-16180-4, 978-3-642-16180-3. URL http://portal.acm.org/citation.cfm?id=1927857.1927871.

VANDEPOELE, K., SAEYS, Y., SIMILLION, C., RAES, J. et VAN DE PEER, Y. : The automatic detection of homologous regions (adhore) and its application to microcolinearity between arabidopsis and rice. *Genome Res*, 12(11):1792–1801, November 2002. ISSN 1088-9051. URL http://dx.doi.org/10.1101/gr.400202.

VOINNET, O. : Shaping small RNAs in plants by gene duplication. *Nat Genet*, 36(12): 1245–6, 2004. URL http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?cmd=prlinks&dbfrom=pubmed&retmode=ref&id=15565102.

WALLACHER, C. et ZIMMERMANN, U. : A combinatorial interior point method for network flow problem. *Mathematical Programming*, 56:321–325, 1992.

WANG, Z., DING, G., YU, Z., LIU, L. et LI, Y. : Chsminer: a gui tool to identify chromosomal homologous segments. *Algorithms for Molecular Biology*, 4:2+, January 2009. ISSN 1748-7188. URL http://dx.doi.org/10.1186/1748-7188-4-2.

WATERMAN, M. S. et SMITH, T. F. : Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

WATSON, J. D. et CRICK, F. H. C. : A structure for deoxyribose nucleic acid. *Nature*, 171:737–738, 1953.

YANG, Z. : PAML: a program package for phylogenetic analysis by maximum likelihood. *CABIOS*, 13:555–556, 1997.

YANG, Z. et NIELSEN, R. : Estimating Synonymous and Nonsynonymous Substitution Rates Under Realistic Evolutionary Models. *Molecular Biology and Evolution*, 17(1): 32–43, January 2000. ISSN 0737-4038. URL http://mbe.oxfordjournals.org/cgi/content/abstract/17/1/32.

ZHANG, J. : Evolution by gene duplication: an update. *Trends in Ecology & Evolution*, 18 (6):292–298, June 2003. URL http://dx.doi.org/10.1016/S0169-5347(03)00033-8.

RÉSUMÉ : après un rappel des notions fondamentales de biologie moléculaire et plus particulièrement des duplications en tandem, la thèse présente un panorama des outils existants permettant de détecter des régions homologues à grande échelle, en se focalisant sur les méthodes de chaînage d'ancres. Le document introduit alors un formalisme général de modélisation basé sur les graphes. Une nouvelle méthode de chaînage, capable de produire un ensemble de chaînes de score optimal et satisfaisant des contraintes de cohérences assurant une interprétation aisée des résultats, est formulée en exploitant la théorie des flots de coût minimum. Cette méthode est évaluée sur des problèmes de détection de duplications segmentales chez les plantes puis intégrée dans un pipeline de détection de grande régions dupliquées en tandem directement à partir de la séquence génomique. Cet outil est évalué sur le génome de la plante modèle *Arabidopsis thaliana* et confronté à l'annotation du génome, montrant ses capacités à détecter des régions dupliquées impliquant des éléments non-codants.

MOTS-CLÉS : bioinformatique, théorie des flots, duplication, en tandem.

ABSTRACT : after a quick introduction to molecular biology and more specifically tandem duplications, the thesis presents an overview of existing tools for detecting large scale homologuous regions, witha focus on anchor chaining methods. The thesis introduces a new graph-based general modelling formalism. A new chaining method, which is able to produce an optimal set of chains that satisfies specific consistency constraints that aim at easier interpretation is described, using minimum cost flow theory. This method is evaluated on segmental duplications detection in plants and then integrated in a pipeline targeted at tandem duplication detection directly from DNA. This tool is evaluated on the *Arabidopsis thaliana* genome and compared to the annotation, showing that it is able to detect tandem duplicated regions involving non coding elements.