Modélisation du déséquilibre de liaison en génomique des populations par méthodes d'optimisation

Thomas Dias Alves

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE
Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES
Spécialité : **Modèles, méthodes et algorithmes en biologie, santé et environnement**
Arrêté ministériel : 25 mai 2016

Présentée par
Thomas DIAS ALVES

Thèse dirigée par **Michael BLUM**
et codirigée par **Julien MAIRAL**
préparée au sein du laboratoire **Techniques de l'Ingénierie Médicale et de la Complexité - Informatique, Mathématiques et Applications de Grenoble (TIMC-IMAG)**
et de l'école doctorale **"Ingénierie de la Santé, de la Cognition et Environnement" (EDISCE)**

Modélisation du déséquilibre de liaison en génomique des populations par méthodes d'optimisation.

Thèse soutenue publiquement le **18 décembre 2017**,
devant le jury composé de :
Emmanuelle GÉNIN
DR INSERM, INSERM Brest, Rapporteuse, Présidente
Julien CHIQUET
CR INRA, INRA Paris, Rapporteur
Franck PICARD
DR CNRS, Université de Lyon, Examinateur
Étienne PATIN
CR CNRS, Institut Pasteur, Examinateur
Bertrand SERVIN
CR INRA, INRA Toulouse, Examinateur
Michael BLUM
DR CNRS, UGA Grenoble, Directeur de thèse

et d'un membre invité :
Julien MAIRAL
CR Inria, Inria Grenoble

Titre : Modélisation du déséquilibre de liaison en génomique des populations par méthodes d'optimisation.

Résumé : Nous présentons un nouveau formalisme et des nouvelles méthodes pour modéliser le déséquilibre de liaison et tenir compte de la structure en haplotypes pour les données issues de la génomique des populations. La modélisation repose sur un problème d'optimisation avec contraintes qui est résolu avec un algorithme de programmation dynamique. Les méthodes établies ont toutes l'avantage d'avoir un coût algorithmique linéaire et donc de pouvoir traiter de grands jeux de données. Dans un premier temps, nous avons appliqué notre approche à l'étude des populations métisses et plus particulièrement au problème d'inférence des coefficients de métissage locaux. Notre méthode a été appliquée à des génotypes simulés de métissage humain ainsi qu'à des vrais génotypes obtenus dans des populations métisses de peupliers. Dans un second temps, nous avons développé notre formalisme d'optimisation pour traiter de l'inférence des haplotypes à partir des génotypes d'une population. L'ensemble de ces méthodes d'optimisation a été développé dans un module Python qui s'appelle `Loter`.

Mots-clés : optimisation, déséquilibre de liaison, haplotype, structure génétique des populations, programmation dynamique, métissage, phase des haplotypes

Title : Modeling the linkage disequilibrium in population genomics with optimization methods.

Abstract : We present a new formalism and new methods to model linkage disequilibrium and to account for haplotype structure of population genomics data. Modeling relies on an optimization problem with constraints that is solved using dynamic programming. The algorithmic cost of proposed methods is linear, which is a desirable property to process large datasets. First, we applied our framework to study admixed populations and perform local ancestry inference. Our method is applied to simulated genotypes of admixed human populations and to real genotypes from admixed Populus species. Second, we developed our optimization framework to perform haploptype phasing and imputation based on a population of genotypes. All optimization methods have been developed in a Python package called `Loter`.

Keywords : optimization, linkage disequilibrium, haplotype, genetic structure of populations, dynamic programming, admixture, phasing

Table des matières

Résumé

Objectifs : Face à la profusion des données massives, modéliser le déséquilibre de liaison et la structure des haplotypes est un enjeu important de la génomique des populations. L'objectif de cette thèse est de développer des modèles reposant sur le déséquilibre de liaison pour décomposer des individus comme une mosaïque de populations ancestrales. Ce type de décomposition est à la base de nombreuses méthodes d'estimation des coefficients de métissage locaux et des logiciels de reconstruction la phase d'haplotypes. Toutefois, les méthodes actuelles requièrent souvent de nombreux paramètres et ne passent pas à l'échelle des grands jeux de données.

Méthodes : Dans un premier temps, nous nous sommes intéressés à l'estimation des coefficients de métissage locaux. Pour des métissages récents, il est possible d'attribuer à chaque locus de chaque individu métisse une population de provenance parmi des populations sources du métissage. Cette décomposition des individus métisses est effectuée grâce à la minimisation d'un problème d'optimisation avec contraintes. Les résultats de plusieurs optimisations sont combinés, à l'image des méthodes d'ensemble, pour améliorer la précision et modéliser l'incertitude des paramètres. Notre méthode tient compte de l'incertitude de la phase des haplotypes grâce à un deuxième problème d'optimisation correcteur de phase. Dans un second temps, nous avons mis au point notre propre algorithme de reconstruction de la phase des haplotypes d'un ensemble d'individus. Cet algorithme est fondé sur le même formalisme d'optimisation avec contraintes et regroupe les haplotypes similaires dans une structure compacte.

Résultats : Ces méthodes ont été mises au point au sein du logiciel **Loter** et appliquées à l'estimation des coefficients d'ascendance sur des données simulées humaines et sur les données de peupliers *P. Trichocarpa* et *P. Balsamifera*. Loter obtient une meilleure précision que les méthodes actuelles telles que LAMP-LD et RFMix pour des métissages anciens. De plus nous avons testé notre algorithme de reconstruction de la phase des haplotypes sur le jeu de données humaines HapMap, pour lequel nous avons obtenu des résultats similaires à l'état de l'art mais avec une complexité algorithmique moindre.

Remerciements

La thèse est un long cheminement qui ne se fait pas seul. D'abord je remercie Michael Blum et Julien Mairal pour leur encadrement au-delà de mes espérances. J'ai beaucoup appris grâce à eux et au temps qu'ils m'ont consacré pour me relire, discuter et critiquer notre recherche.

Un énorme merci à toute ma famille : à Céline pour toute sa bienveillance et à mes parents Claudine et Pascal. Enfin une pensée à mes co-bureaux/collègues/amis Kévin, Keurcien, Thomas et Florian pour toutes nos discussions sur l'existence du hasard, Emacs et Python vs R (les deux pardis !). Les résultats de cette thèse sont aussi le fruit de toute la communauté scientifique et du logiciel libre. Je compte bien continuer à poser ma petite pierre dans ces domaines. Merci aussi à toi lecteur égaré qui pour une raison que j'ignore a décidé de lire cette thèse.

CHAPITRE 1

Introduction

As others have noted, population genetics as a field was theory rich and data poor for much of its history. Over the last five years this has changed beyond recognition.

Donnelly, Peter 2010

Grâce aux progrès technologiques et à l'avènement des données génomiques, la génétique des populations bénéficie de nouvelles ressources pour comprendre la structure des populations et les prédispositions à certaines maladies. Seulement une dizaine d'années nous sépare du premier séquençage du génome entier d'un individu, celui de James Watson, codécouvreur de la structure de l'ADN *Acide Désoxyribo Nucléique*, en 2007. Les nouvelles méthodes de séquençage, **NGS** *next-generation sequencing*, permettent aujourd'hui de caractériser un individu par des millions de marqueurs génétiques à faible coût. L'étude de la variabilité génétique au sein de populations naturelles possède en effet de vastes applications, de la théorie de l'évolution au conseil médical. Nous allons maintenant détailler ce que nous entendons par marqueurs génétiques dans le cadre de cette thèse et présenter succinctement les notions basiques du contexte dans lequel nous nous plaçons.

1.1 Données de polymorphismes génétiques

Le génome désigne l'ensemble des informations génétiques encodées dans l'ADN. Par exemple, le génome humain est composé de 22 chromosomes homologues et de

deux chromosomes sexuels X et Y. Les espèces constituées de paires de chromosomes homologues sont dites diploïdes et les gènes portés sur chacun des chromosomes peuvent différer. Dans ce cas, les gènes disposés sur les loci [1] d'un même chromosome constituent un haplotype (contraction de *haploid genotype*). À l'instar d'une cartographie de la variabilité du génome humain, une carte d'haplotypes communs a vu le jour grâce à l'International Hapmap Consortium (The International HapMap Consortium, 2005).

L'ADN est une molécule en double hélice constituée de paires de bases nucléiques : l'adénine (A), la cytosine (C), la thymine (T) et la guanine (G). Ces nucléotides varient seulement sur un sous-ensemble des positions du génome entre différents individus. Diverses opérations altèrent l'ADN et produisent ces variations au fur et à mesure des générations comme les substitutions ou les insertions de nucléotides et caractérisent de nombreux polymorphismes.

SNP *single nucleotide polymorphism*. Le polymorphisme nucléotidique (Figure 1.1) désigne la variation d'une seule paire de base entre différents individus et chaque variant doit avoir une fréquence supérieure à 1% dans la population.

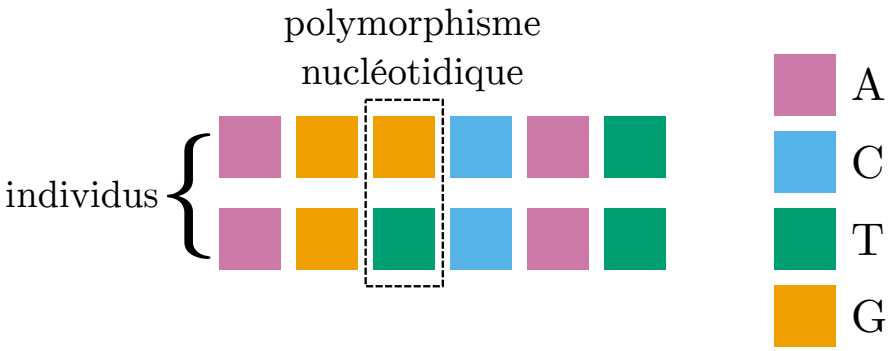

Figure 1.1 – Représentation de deux séquences de nucléotides avec une substitution seulement sur la troisième base azotée de G en T. Le polymorphime nucléotidique (**SNP**) est un des polymorphismes les plus simples et les plus fréquents (Brookes, 1999).

Modèle à infinité d'allèles. Pour la majorité des SNPs, seulement deux variants sont observés. En effet, dans le modèle à infinité d'allèles, la probabilité d'avoir plus d'une mutation sélectionnée pour une position du génome est considérée nulle. Ce modèle simplifie ainsi l'encodage des haplotypes et des génotypes (Figure 1.2).

1. Emplacement physique sur un chromosome.

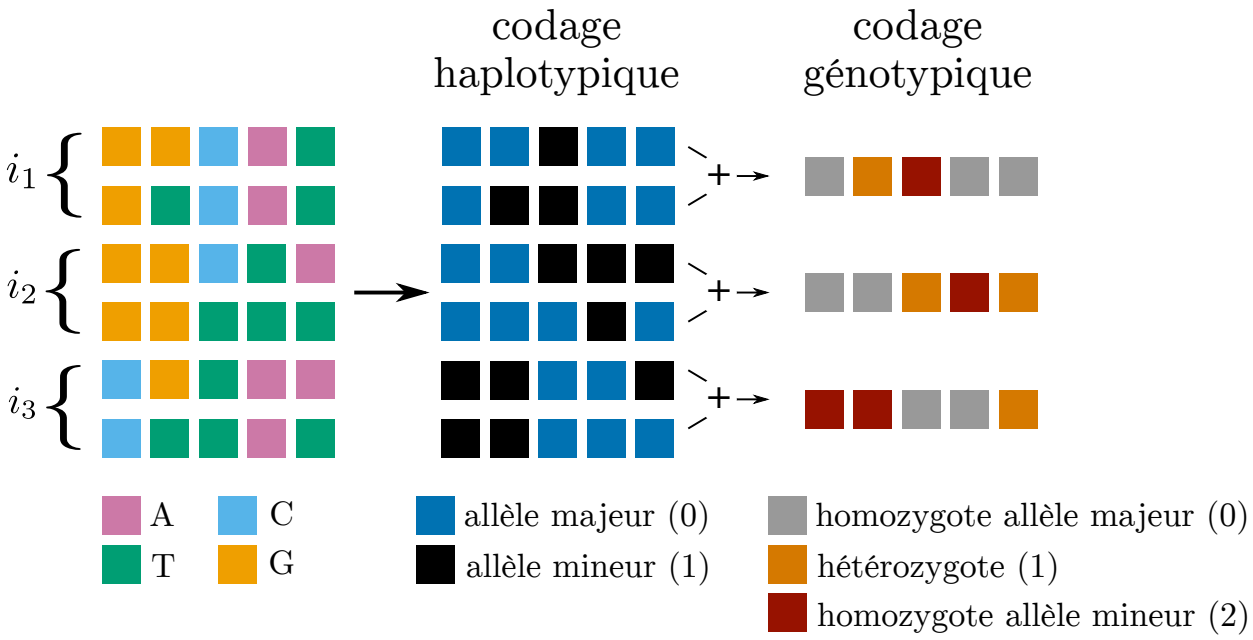

FIGURE 1.2 – Modèle à infinité d'allèles. En se restreignant seulement aux SNPs, on suppose qu'il n'existe que deux variants possibles pour chaque position. Pour chacun des haplotypes, le variant le plus fréquent (allèle majeur) est encodé par 0 et l'autre par 1. L'encodage pour les génotypes est obtenu par la somme de deux haplotypes.

Soit n le nombre d'individus et m le nombre de loci. Nos analyses se fondent donc sur des matrices de SNPs soit haplotypiques $H \in \{0,1\}^{2n \times m}$, soit génotypiques $G \in \{0,1,2\}^{n \times m}$. Chaque ligne de G désigne le génotype de l'individu i et chaque couple de lignes $(2i, 2i+1)$ représente les haplotypes du i-ième individu. La somme des haplotypes d'une paire est égale au génotype de l'individu.

Ces données génétiques sont transmises au fil des générations au moment de la reproduction pour certaines espèces. Les deux principaux phénomènes à l'origine des variations entre les haplotypes des parents et ceux de l'enfant sont les mutations et les recombinaisons (Figure 1.3).

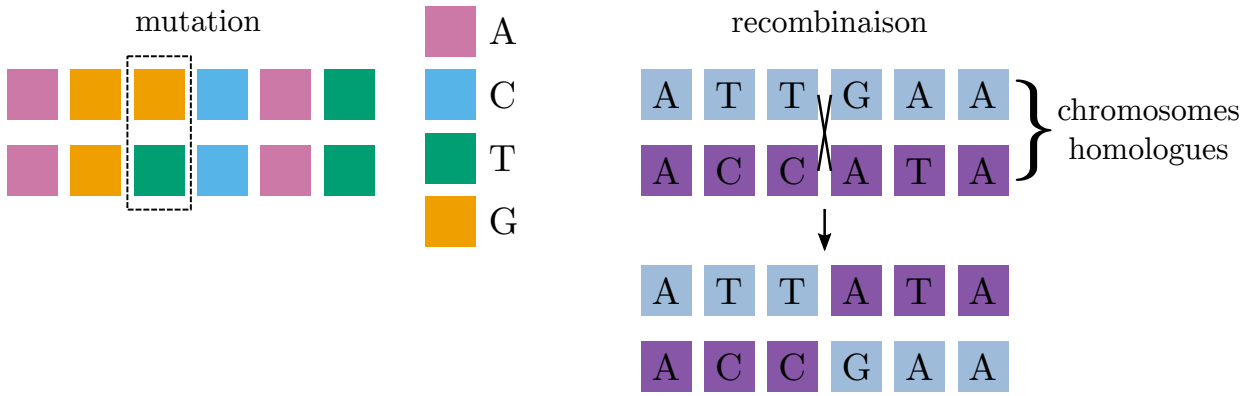

FIGURE 1.3 – Mutation et Recombinaison.

1.2 Déséquilibre de liaison

Le déséquilibre de liaison représente la non-indépendance des allèles sur des loci différents du génome. Autrement dit, le déséquilibre de liaison (noté LD *linkage disequibrilium*) indique que des allèles particuliers sur des sites différents peuvent apparaître ensemble sur le même haplotype plus souvent que par chance (WALL et PRITCHARD, 2003). Le LD peut être mesuré de plusieurs façons dont le r^2 (équation 1.1) (PRITCHARD et PRZEWORSKI, 2001). En notant π_A, π_a, π_B, π_b les fréquences alléliques des quatre allèles et π_{AB}, π_{Ab}, π_{aB}, π_{ab} la fréquence des haplotypes, alors r^2 est défini de la manière suivante :

Définition 2.1: Mesure du déséquilibre de liaison

$$D = \pi_{AB} - \pi_A \pi_B.$$
$$r^2 = \frac{D^2}{\pi_A \pi_a \pi_B \pi_b}. \tag{1.1}$$

Cette approche renseigne la liaison entre des paires de SNPs (Figure 1.4).

Le déséquilibre de liaison est produit, entre autres, par les phénomènes de métissage des populations, la dérive génétique et la sélection naturelle. Le phénomène de recombinaison joue un rôle important dans l'évolution du déséquilibre de liaison puisque les recombinaisons modifient directement des zones des haplotypes et donc π_{AB} dans l'équation 1.1. Les recombinaisons ont pour effet de ramener à l'équilibre de liaison en diversifiant les haplotypes de la population (Figure 1.3). Le taux de recombinaison et le déséquilibre de liaison sont donc liés (Long et Langley, 1999). Intuitivement, entre deux zones où le taux de recombinaison est élevé, le déséquilibre de liaison entre les allèles des zones sera faible. Le génome est donc organisé en blocs où le LD est plus fort, appelés *blocs de LD* ou *blocs haplotypiques* (Figure 1.4). Dans un bloc de LD les SNPs sont donc fortement corrélés, il est possible de choisir un seul représentant et de filtrer les autres. Cette approche appelée *LD pruning* est par exemple utilisée pour reconstruire la structure de populations avec l'Analyse en Composantes Principales (ACP) afin d'éviter de capturer trop de variance pour les SNPs d'un bloc de LD (Galinsky et al., 2015 ; Abdellaoui et al., 2013). À l'inverse, d'autres méthodes reposent sur la présence de LD comme les méthodes d'estimation de la phase des haplotypes (Scheet et Stephens, 2006 ; Browning et Browning, 2007) ou d'imputation (Stephens et Scheet, 2005 ; Jung et al., 2007). Le déséquilibre de liaison joue un rôle important pour les études d'association en établissant les liens entre les marqueurs mais aussi pour l'histoire des populations en estimant le métissage par exemple.

La modélisation de la diversité génétique permet de tester des hypothèses et de comprendre l'évolution d'une espèce. Une grande variété de théories et de modèles mathématiques a été développée en biologie pour estimer des quantités biologiques comme par exemple le taux de recombinaison. Au-délà de l'étude démographique et historique des populations, de vastes applications médicales découlent de ces modélisations. Tout modèle est basé sur des hypothèses simplificatrices. La connaissance de ces hypothèses permet de distinguer les idées fondamentales qui séparent ces modèles et de percevoir les cas d'application. De plus, chaque modèle fournit des résultats théoriques distincts avec des garanties différentes de temps de calcul. Ce compromis entre complexité

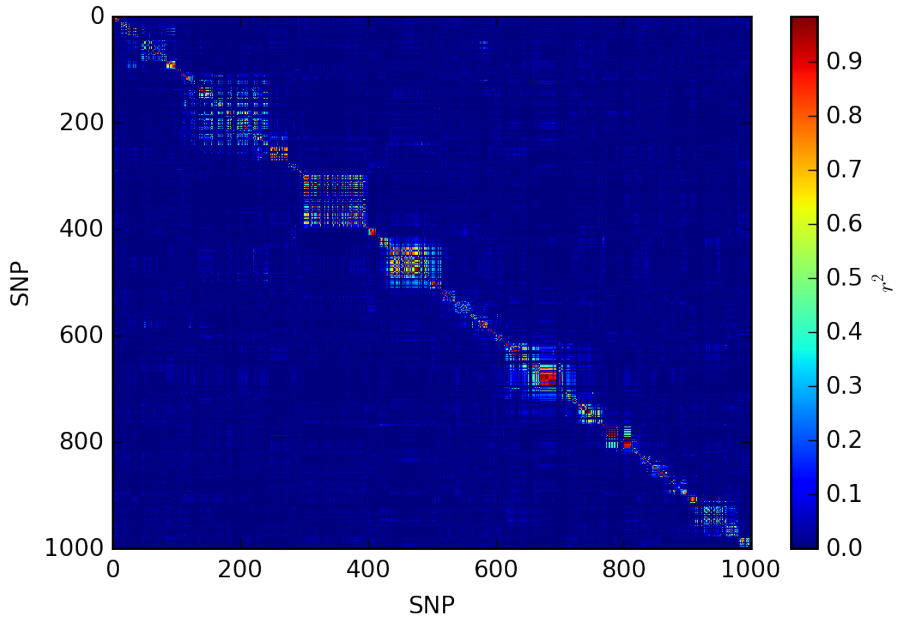

FIGURE 1.4 – Visualisation du déséquilibre de liaison sur les 1000 premiers SNPs de HapMap CEU en calculant le r^2. Les CEU sont les individus d'origine européenne du jeu de données HapMap.

algorithmique et complexité théorique du modèle joue un rôle primordial dans l'histoire de la modélisation en biologie ainsi que dans d'autres sciences. En effet, comme le souligne la citation de Peter Donnelly en tête de ce chapitre, ce domaine très riche en théories est confronté à des jeux de données de plus en plus massifs depuis plusieurs années.

Nous allons maintenant aborder les modèles en lien et à la base des thématiques de cette thèse dans l'ordre chronologique de leur création.

1.3 Modèle de Wright-Fisher

Un modèle fondamental a été introduit par Fisher (1930) et Wright (1931) pour modéliser l'évolution d'haplotypes au fil des générations. Ce modèle sur les haplotypes est transposable au cas des espèces diploïdes. Sa présentation est simplifiée en se restreignant au cas des haplotypes.

Le modèle de Wright-Fisher (Figure 1.5) considère une population constante de N individus haploïdes. Les individus à la génération suivante sont créés par tirage avec remise parmi les individus de la génération précédente. Vu dans l'autre sens (des parents vers les enfants), chaque individu donne naissance à un nombre d'enfants qui suit une loi binomiale de paramètre N le nombre d'épreuves et $\frac{1}{N}$ la probabilité

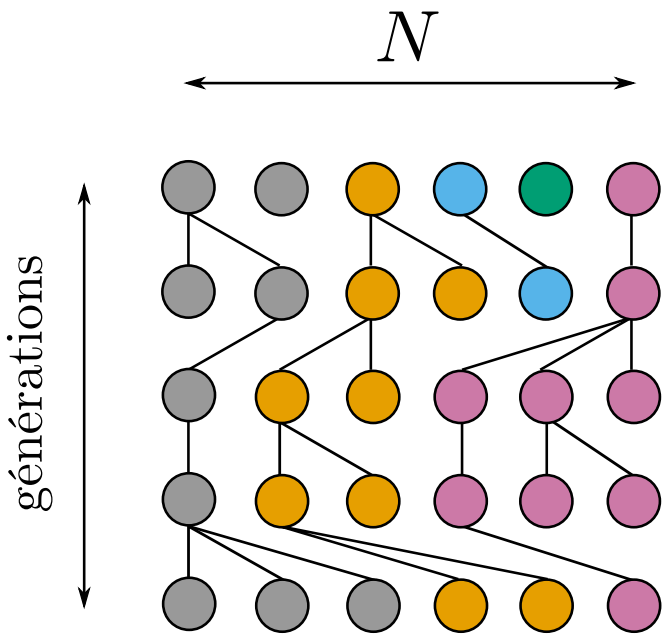

FIGURE 1.5 – Illustration du modèle de Wright-Fisher en considérant 5 couleurs de départ. Les couleurs symbolisent un trait héréditaire, un allèle par exemple. On aperçoit l'effet de la dérive génétique sur cette simulation pour 6 individus et 5 générations. On constate qu'il ne reste que trois couleurs à la fin.

de succès. Ces probabilités ne sont pas indépendantes puisque le nombre d'individus est fixe. Les N individus sont ainsi renouvelés, génération après génération, sur des périodes de temps non chevauchantes. Cette approche est dite "vision avant", *forward*, puisque les générations sont simulées dans le sens du temps. Cette version du modèle est la plus basique. Dans certaines variantes, N évolue au cours du temps et dans d'autres des recombinaisons sur les chromosomes ont lieu. Toutefois, ce modèle donne déjà lieu à bon nombre de résultats sur l'évolution génétique humaine et celle des autres espèces (DONNELLY et LESLIE, 2010). Il modélise simplement la généalogie d'un groupe d'individus et l'hérédité d'un ou plusieurs allèles.

La généalogie des populations s'intéresse plus particulièrement aux liens de parenté et à l'arbre phylogénétique construit par le processus de Wright-Fisher pour mieux comprendre les forces qui s'appliquent sur les populations telles que la dérive génétique, la sélection naturelle ou encore le métissage. La probabilité que deux individus aient un parent différent à la génération précédente est $1 - \frac{1}{N}$. Le nombre de générations τ nécessaire pour retrouver l'ancêtre commun de deux individus est appelé temps de coalescence et suit une loi géométrique,

$$\Pr(\tau = k) = \left(1 - \frac{1}{N}\right)^{k-1} \frac{1}{N}, \text{ avec } k \geq 1.$$

Zones de recombinaison. Les zones de recombinaison pour une génération sont définies par un processus de Poisson qui permet de modéliser l'apparition aléatoires d'évènements (Haldane, 1919). Un processus de Poisson est un processus de comptage, noté $N(t)$, comptant le nombre d'apparitions d'un événement dans l'intervalle $[0, t]$. Ce processus suit les deux propriétés suivantes :

Définition 3.1: Accroissements indépendants

pour tout t, t' et s tels que $t' > t > s \geq 0$,
$$\Pr\{N(t') - N(t) | N(s)\} = \Pr\{N(t') - N(t)\}$$

Définition 3.2: Accroissements stationnaires

pour tout s, $t \geq 0$,
$$N(s+t) - N(t) \stackrel{\mathcal{L}}{=} N(s)$$

La première propriété signifie que le nombre de recombinaisons sur une zone du chromosome est indépendant du nombre de recombinaisons sur des zones disjointes. La deuxième propriété impose que la loi du nombre de recombinaisons sur un intervalle ne dépend que de la longueur de l'intervalle.

Ce processus stochastique permet de simuler des points de recombinaison avec une densité constante λ le long des haplotypes.

1.4 Le coalescent de Kingman

Dans le cas où N tend vers l'infini, la probabilité d'avoir le même parent tend vers 0. L'unité de temps est donc modifiée en N générations pour contourner ce problème. Le nombre de générations τ nécessaire pour retrouver un ancêtre commun est donc transformé en T par la formule suivante,

$$k = \lfloor tN \rfloor, t \in \mathbb{R}_+$$

$$\tau = \lfloor TN \rfloor.$$

Le problème est en même temps rendu continu.

Une approximation de la loi géométrique est alors donnée par

$$\lim_{N \to \infty} p(T > t) = e^{-t},$$

et le temps de coalescence suit donc une loi exponentielle de paramètre 1. Le modèle de coalescent décrit les $N-1$ évènements de coalescence (Figure 1.6). En notant T_i le temps pour passer de i lignées à $i-1$ lignées, alors

$$p(T_i = t) = \binom{i}{2} e^{-\binom{i}{2}t}.$$

Kingman (1982) a démontré le lien entre ce modèle de coalescence et le modèle de Wright-Fisher lorsque N tend vers l'infini. Ce modèle reconstruit la généalogie dans le sens inverse (*backward*). Pour cela deux lignées sont choisies uniformément parmi les i lignées et se rejoignent au bout d'un temps T_i tiré selon la loi exponentielle de paramètre $\binom{i}{2}$. Une fois la généalogie construite, les mutations sont ajoutées (sens *forward*). Cela est valide seulement pour des marqueurs neutres c'est-à-dire des marqueurs pour lesquels le type n'a pas d'influence sur la généalogie. L'avantage du modèle de coalescence réside dans les simplifications exposées ci-dessus et sa capacité à simuler efficacement des jeux de données.

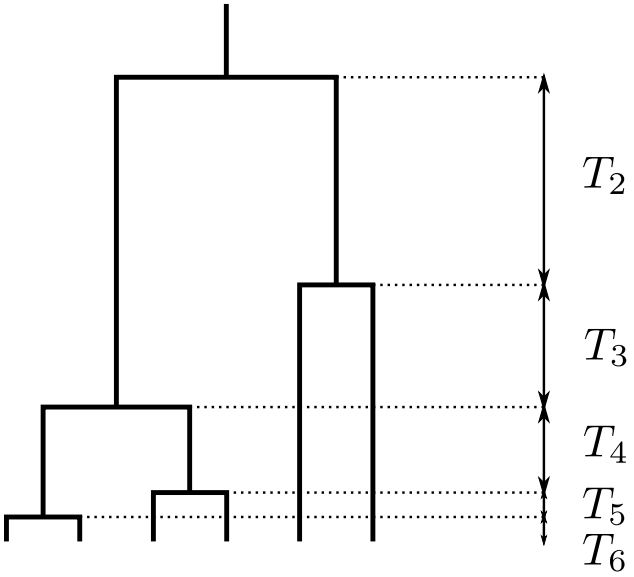

Figure 1.6 – Modèle de Kingman. Arbre de coalescence représentant les lignées d'une population. Les intersections sont les évènements de coalescence et apparaissent lorsqu'une nouvelle lignée est créée. $T_2, \ldots, T_6$ désignent les temps de coalescence et leur somme renseigne sur le temps nécessaire pour remonter à l'ancêtre commun des 6 individus, noté habituellement T_{MRCA}.

Les recombinaisons peuvent aussi être prises en compte dans la théorie de la coalescence mais nous ne discuterons pas de toutes ces modifications (McVean et Cardin, 2005 ; Hudson, 1983).

Si l'évolution des espèces est modélisée soit par le modèle *forward* de Wright-Fisher ou par le modèle *backward* de coalescence, l'inférence par le biais de ces modèles présente de nombreuses difficultés. Néanmoins ces modèles ont démontré leur efficacité pour la simulation de jeux de données (Hudson, 2002) et nous recourrons à leur utilisation pour les simulations.

Une des problèmatiques d'inférence est appelée **CSD** *Conditional Sampling Distribution*. Le but est de décrire la distribution de probabilité pour un ou plusieurs haplotypes supplémentaires sachant une population d'haplotypes. La population d'haplotypes doit permettre d'estimer les informations sur la population sous-jacente nécessaires pour connaître la probabilité d'observer le nouvel haplotype. Pour répondre aux exigences du CSD et de l'inférence via un modèle de génétique des populations, Li et Stephens (2003) ont mis au point un modèle novateur dans la lignée des modélisations proposées par Stephens et Donnelly (2000) et Fearnhead et Donnelly (2001).

1.5 Modèle de Li et Stephens

Le modèle de Li et Stephens, aussi appelé *copying model*, repose sur l'idée qu'un haplotype peut être vu comme une mosaïque des autres haplotypes. Soit $H_1, \ldots, H_n$ l'ensemble des haplotypes observés tels que $H_i \in \{0,1\}^m$, où m est le nombre de loci. Ces haplotypes proviennent soit de n individus haploïdes soit de $n/2$ individus diploïdes. Étant donné un ensemble Φ de paramètres du modèle, la probabilité $p(h_1, \ldots, h_n|\Phi)$ peut être décomposée en probabilités conditionnelles correspondant au problème CSD exposé précedemment,

$$p(H_1, \ldots, H_n|\Phi) = p(H_1|\Phi) \times p(H_2|H_1, \Phi) \times \ldots \times p(H_n|H_1, \ldots, H_{n-1}, \Phi).$$

Initialement, la première application du modèle de Li et Stephens était l'estimation du taux de recombinaison entre les SNPs noté ρ (Li et Stephens, 2003). Néanmoins, ces probabilités conditionnelles ne sont pas connues et doivent être approximées. Soit $\hat{\pi}$ l'approximation de la distribution conditionnelle telle que

$$p(H_1, \ldots, H_n|\Phi) \approx \hat{\pi}(H_1|\Phi) \times \hat{\pi}(H_2|H_1, \Phi) \times \ldots \times \hat{\pi}(H_n|H_1, \ldots, H_{n-1}, \Phi).$$

Ce modèle est appelé **PAC** *Product of Approximate Conditionnals*, la vraisemblance correspondante est notée L_{PAC} et le paramètre la maximisant est noté Φ_{PAC}. Tout l'enjeu consiste alors à proposer une approximation $\hat{\pi}$ qui soit appropriée. Li et Stephens (2003) ont relevé 5 points importants :

- Un nouvel haplotype a plus de chance de correspondre à un haplotype très fréquent que peu fréquent.

- La probabilité de découvrir un nouvel haplotype décroît lorsque le nombre d'haplotypes déjà vus augmente.
- La probabilité de découvrir un nouvel haplotype augmente en fonction du taux de mutation.
- Un nouvel haplotype doit soit être égal aux anciens haplotypes, soit leur ressembler.
- Les haplotypes se ressemblent sur des zones contiguës du fait des recombinaisons.

L'approximation de la probabilité conditionnelle d'observer un certain haplotype sachant une population d'haplotypes repose sur un modèle de Markov caché (HMM) $\{S, \Omega, A, B\}$.

- S : l'ensemble des états. Cela indique l'haplotype d'où l'on copie des marqueurs, $S = \{1, \ldots, n\}$.
- Ω : le vecteur $(\omega_i)_{1..n}$ des probabilités des états initiaux (voir équation 1.3).
- A : la matrice de transition. Les transitions reproduisent l'effet des recombinaisons (voir équation 1.2).
- B : la matrice des probabilités d'émission, elle permet de tenir compte des mutations (voir équation 1.4).

S_j représente donc l'haplotype duquel on copie au locus j. Alors, la probabilité de changer d'haplotype est liée au taux de recombinaison ρ et à la distance génétique d entre les marqueurs par la relation suivante :

$$\begin{aligned} A_{x,x'} &= \Pr(S_{j+1} = x' | S_j = x) \\ &= \begin{cases} 1 - \alpha + \frac{\alpha}{n}, & \text{si } x' = x, \text{ avec } \alpha = 1 - e^{-\frac{\rho d}{n}} \\ \frac{\alpha}{n}, & \text{sinon.} \end{cases} \end{aligned} \tag{1.2}$$

Lorsque ρ ou d augmente, la probabilité de changer d'état augmente. L'équation 1.2 suppose que le taux de recombinaison et la distance sont constants entre les SNPs. Il est possible d'adapter l'équation pour faire varier ces quantités selon la position j.

Quant à la probabilité d'être dans l'état initial i, elle est uniforme,

$$\omega_i = \frac{1}{n}. \tag{1.3}$$

La probabilité de copier l'allèle de l'haplotype doit tendre vers 1 lorsque $n \to \infty$ et tendre vers 1/2 lorsque $\theta \to \infty$, avec θ proportionel au nombre de mutations par méiose par locus.

$$\begin{aligned} B_{j,a} &= \Pr(H_{n+1,j} = a | S_j = X_j, H_1, \ldots, H_n) \\ &= \begin{cases} \frac{n}{n+\hat{\theta}} + \frac{1}{2} \times \frac{\hat{\theta}}{n+\hat{\theta}}, & \text{si } H_{S_j,j} = a \\ \frac{1}{2} \times \frac{\hat{\theta}}{n+\hat{\theta}}, & \text{sinon.} \end{cases} \end{aligned} \tag{1.4}$$

La probabilité conditionnelle d'obtenir l'haplotype h sachant les états cachés S et une famille d'haplotypes H est obtenue par produit des probabilités,

$$\begin{aligned} \Pr(h|S,H) &= \Pr(S)\prod_j \Pr(H_j|S_j,H) \\ &= \omega_{S_1}\prod_{j=2}^{m} A_{S_j,S_{j+1}} \times \prod_{j=1}^{m} B_j(h_j). \end{aligned} \tag{1.5}$$

.

L'algorithme *forward* permet d'intégrer sur l'espace des S et ainsi de calculer $\hat{\pi}$.

1.6 Problématiques de la thèse

Le modèle de Li et Stephens a ouvert la porte à l'analyse de grands jeux de données en tenant compte du LD et en modélisant les haplotypes. Nous partons des idées fondatrices du *copying* model et les adaptons à des problèmes d'optimisation qui s'affranchissent du paradigme probabiliste. L'objectif de cette thèse est d'établir des méthodes algorithmiques pour trois problèmes de la génétique des populations.

Estimation des coefficients de métissage locaux : En observant les haplotypes d'une population métisses ainsi que les haplotypes des populations "ancestrales", déterminer pour chaque locus des haplotypes de la population métisse quelle est la population ancestrale source.

Phasage populationnel : Retrouver les haplotypes d'une population à partir de leurs génotypes.

Imputation : Inférer les données manquantes de matrices de génotypes.

En effet, la structure de populations influence la formation de motifs dans les haplotypes des individus et notamment de blocs localisés d'haplotypes (Browning et Weir, 2010). Le LD (équation 1.1) va de pair avec les blocs d'haplotypes et renseigne donc sur la structure sous-jacente (Stumpf, 2002). Cette relation entre structure de populations, haplotypes et déséquilibre de liaison est au coeur de cette thèse (Figure 1.7). Le modèle de Li et Stephens illustre parfaitement ce lien entre la structure de populations S, les haplotypes H et le déséquilibre de liaison (hypothèse markovienne sur S). De même, ce modèle laisse percevoir les liens entre les trois problèmes. En effet, en estimant S, l'imputation peut se faire par $H_{S_j,j}$ et l'estimation du coefficient de métissage par la population d'où provient l'individu S_j.

Les contraintes modernes nécessitent des méthodes rapides et capables de traiter de grands jeux de données. En effet, l'analyse d'un jeu de données requiert souvent d'appliquer la méthode à de multiples reprises, en modifiant les paramètres ou en

effectuant divers prétraitements. L'objectif est donc d'établir des méthodes rapides relativement à l'état de l'art et simples d'utilisation, c'est-à-dire minimisant le nombre de paramètres requis.

Le délivrable de cette thèse est donc un ensemble de méthodes pour inférer les coefficients de métissage locaux, phaser les matrices de génotypes et imputer les données. Ces méthodes sont disponibles dans un module Python, nommé Loter, destiné à faciliter l'utilisation. De plus ces méthodes ont toutes une complexité algorithmique linéaire et sont parallélisées. L'analyse des données par le biais de Loter permet d'inspecter et de visualiser rapidement les résultats et de créer des pipelines d'analyse sans passer par des formats de données intermédiaires.

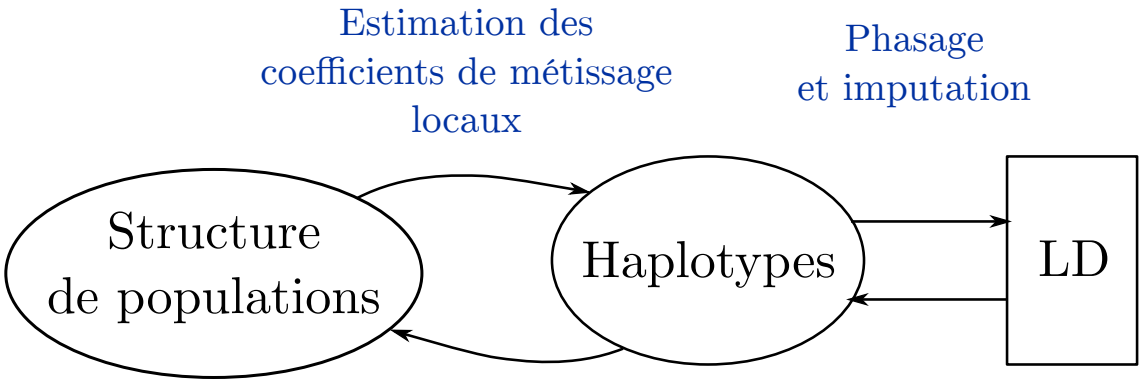

FIGURE 1.7 – Relation entre structure de populations, haplotypes et LD. La structure de populations, par la reproduction préférentielle et la dérive génétique, influence les motifs présents dans les haplotypes. Ces motifs engendrent des corrélations entre les loci d'un haplotype, cela se traduit par la présence de déséquilibre de liaison. Ces notions permettent d'appréhender le lien entre les problématiques de cette thèse : estimation des coefficients de métissage locaux, phasage et imputation. En effet, le phasage et l'imputation s'appuient sur le lien entre les haplotypes et le LD apparent. Tandis que l'estimation des coefficients de métissage locaux repose sur la dynamique entre la structure de populations et les informations génétiques contenues sur les haplotypes.

CHAPITRE 2

Estimation des coefficients de métissage locaux

La génétique étudie le vivant de l'échelle individuelle, "microscopique", à l'échelle "macroscopique" des espèces. Une échelle intermédiaire, celle des populations, regroupe les individus similaires d'une espèce, partageant un même territoire et se reproduisant. La structure génétique des populations est l'étude des similarités et des variations entre les individus grâce aux données génétiques. Pour les populations issues d'un métissage récent entre deux populations, les recombinaisons produisent une structure de populations qui localisée le long des chromosomes. En effet, les marqueurs des individus métisses proviendront tantôt d'une population, tantôt de l'autre (Figure 2.1). Estimer les coefficients de métissage locaux revient à déterminer la population de provenance pour chaque marqueur (Figure 2.2).

Formellement, nous notons $H \in \{0,1\}^{n,m}$ une matrice de haplotypes définis pour m marqueurs d'une population et $H_{\text{ref}} = \{H^i, i \in \{1,\ldots,k\}\}$ l'ensemble des matrices des k populations de référence. Soit $h \in \{0,1\}^m$ un haplotype métisse. Alors estimer les coefficients de métissage locaux de h consiste à calculer $Z \in \{1,\ldots,k\}^m$ tel que Z_j indique la population de provenance de h_j. Nous détaillerons ce formalisme dans la partie méthode.

De nombreuses populations, notamment humaines, sont des populations métisses. Leur étude est donc un enjeu majeur de la biologie. Dans ce chapitre, nous introduisons le problème d'estimation des coefficients de métissage locaux, nous présentons les méthodes existantes et nous exposerons la méthode que nous avons mise au point au sein du logiciel Loter ainsi que ses applications.

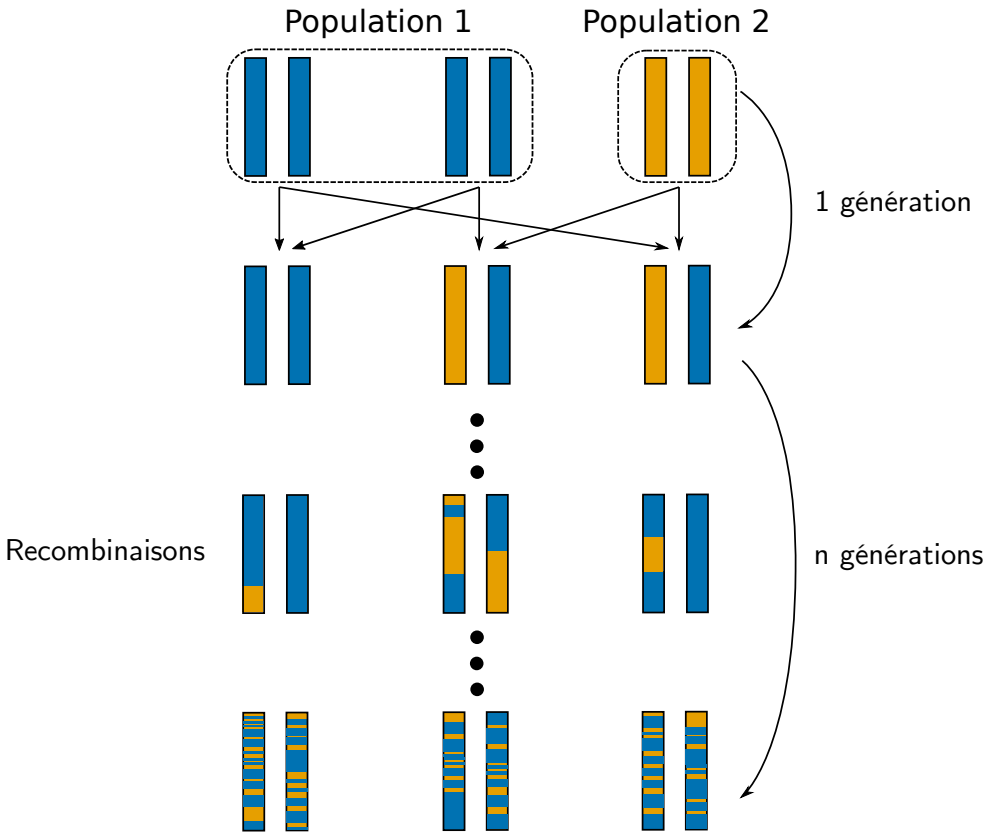

FIGURE 2.1 – Mosaïque d'haplotypes due au métissage de deux populations en bleu et en orange. Nous représentons le modèle de Wright-Fisher diploïde avec recombinaisons pour trois individus diploïdes appartenant à deux populations. Cette figure illustre l'importance d'une approche localisée pour comprendre le métissage récent.

2.1 État de l'art

Dans cette partie, nous présentons le phénomène du métissage et son impact sur la structure des données génétiques. Nous détaillons les approches pour estimer cette structure particulière et comment le déséquilibre de liaison a été intégré à ces méthodes. Enfin, nous décrivons des applications pratiques de l'estimation des coefficients de métissage locaux.

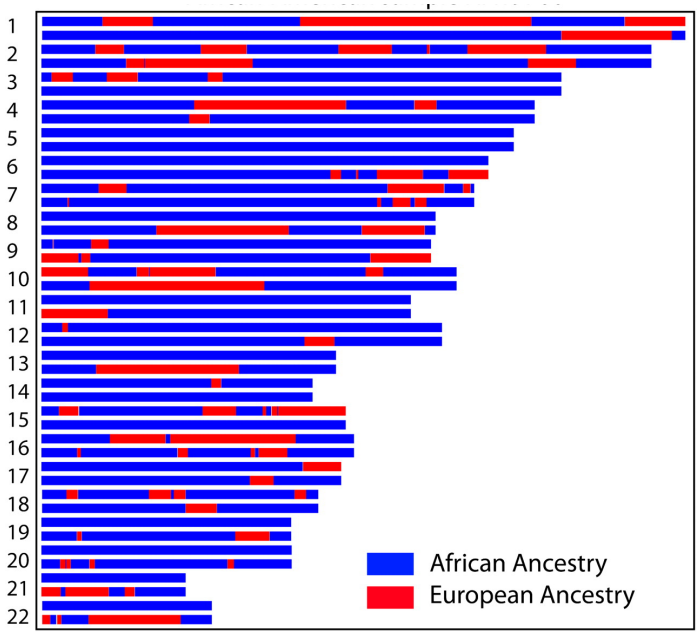

FIGURE 2.2 – Estimation des coefficients locaux de métissage d'un afro-américain. Image tirée de GRAVEL (2012) utilisant le logiciel PCAdmix. Les populations sources sont CEU et YRI de HapMap tout comme dans nos expériences.

2.1.1 Structure des données génétiques

De nombreuses espèces sont structurées génétiquement, c'est-à-dire que les génotypes des individus sont organisés en sous-groupes homogènes. Dans le cas d'une espèce non-structurée, la distribution des fréquences d'allèles et des haplotypes reste indépendante, dans une certaine mesure, de la répartition en sous-groupe choisie. Pour une espèce structurée, ces distributions varient entre les sous-groupes. Ces sous-groupes homogènes forment les populations de l'espèce.

Cette structure apparaît dès lors que des groupes d'individus se reproduisent de manière préférentielle et que des processus aléatoires ont effet sur ces populations. Ces processus peuvent être des processus neutres comme la dérive génétique et les événements démographiques ou des processus d'adaptation comme la sélection

naturelle. Pour illustrer ce propos sur la structure de populations, nous avons réprésenté graphiquement les matrices de génotypes $G \in \{0,1,2\}^{n \times m}$ du jeu de données HapMap (Figure 2.3) que nous présentons dans la partie **Données** 2.2.1.

La détection de la structure de populations est un sujet important en génétique des populations. La structure de populations explique en grande partie les variations génétiques entre les individus. L'Analyse en Composante Principale (ACP) illustre ce principe en projetant les individus selon les axes maximisant la variance. Novembre et al. (2008) applique l'ACP et montre que non seulement les populations sont séparées, mais qu'elles sont aussi positionnées globalement selon leur position géographique.

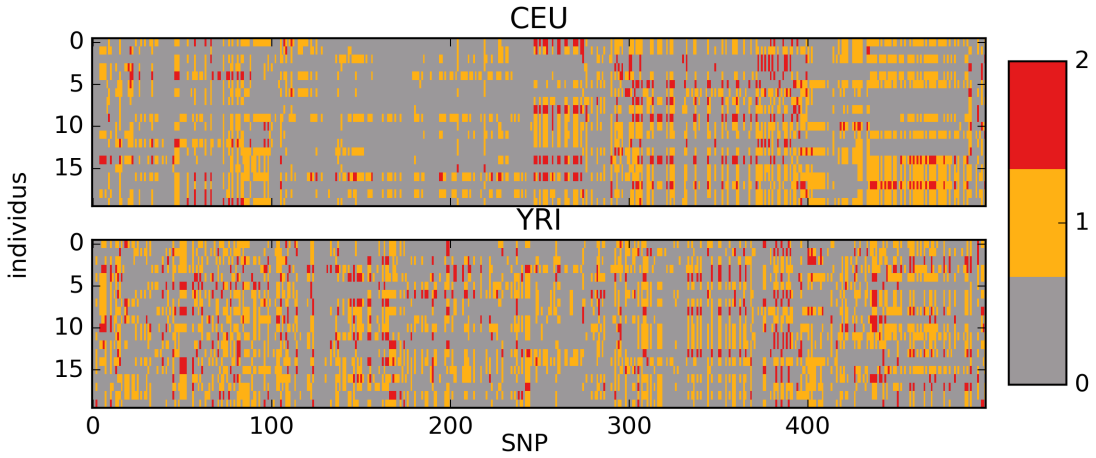

Figure 2.3 – Matrices de SNPs des populations CEU (européenne) et YRI (africaine) de HapMap.

Le métissage a lieu lorsque plusieurs populations ayant divergé séparement se reproduisent et se mélangent. Cela affecte alors à la fois la structure de populations et les données génétiques. Deux types de population sont à distinguer : les populations de référence, appelées aussi populations sources ou ancestrales et les populations métisses. L'étude du métissage et de la phylogénie peut se faire "globalement" ou "localement" vis-à-vis du génome. Du point de vue "global", l'objectif est d'estimer pour chaque individu ses coefficients de métissage, c'est-à-dire la probabilité d'appartenir à chaque population de référence. L'approche "locale" estime les coefficients de métissage (ou l'ascendance [1]) locus par locus. Dans ce contexte, le terme "local" fait réfèrence à une localité sur le génome et non pas une localité géographique (*local ancestry* en anglais). L'approche locale est justifiée par la présence de recombinaisons au fil des générations (Figure 2.1). Les loci le long du génome d'un individu proviennent de

1. Les coefficients de métissage, dans le cas "local", ne sont plus des proportions mais les indices des populations ancestrales desquelles sont copiés les marqueurs. Nous les appellerons soit coefficients de métissage locaux, soit coefficients d'ascendance.

différentes populations. Les haplotypes des individus métisses sont alors une mosaïque des haplotypes initiaux (Gravel, 2012).

Il faut noter que cette décomposition dépend du référentiel (populations ancestrales) et du temps T depuis le métissage. En effet, la taille des segments continus de coefficients d'ascendance dépend de T. Plus T est élevé, plus le phénomène de recombinaison a découpé les segments.

De plus, le problème d'estimation abordé ici repose essentiellement sur un métissage ponctuel entre des populations distinctes (Figure 2.1). Il est possible de détecter des situations plus complexes. La relation entre les populations étudiées peut être hiérarchique. Les méthodes basées sur les arbres phylogénétiques établissent des liens hiérarchiques entre les populations mais ne modélisent pas le métissage (Cavalli-Sforza et Edwards, 1967). Dans le cas d'un métissage de plusieurs populations dont certaines seraient elles-mêmes des métisses, la situation est plus complexe. Des approches hiérarchiques comme TreeMix ont été développées et possède aussi l'atout d'être multi-échelle Pickrell et Pritchard, 2012.

2.1.2 Modèle de métissage et déséquilibre de liaison

La densification en SNPs des jeux de données a permis de passer de modèle de métissage globaux à des modèles intégrant le LD et estimant localement la structure.

Modèle de métissage, *admixture model*, sans LD

Pour estimer la structure de populations, les premières approches telles que *structure* (Pritchard, Stephens et Donnelly, 2000) introduisent des coefficients de métissage q_k^i qui mesure la proportion de génome d'un individu i issue de la population k. Typiquement, pour un individu afro-américain, le vecteur de métissage $Q = (0.2, 0.8)$ signifie que le génome de l'individu est à 20% européen et à 80% africain vis-à-vis de ce modèle. Soit Z le vecteur multidimensionnel d'appartenance aux populations sources et A le vecteur des allèles observés d'un individu. Alors $Z_l^{(i,A_l)}$ représente la population de provenance de l'allèle A_l, sur un locus spécifique l, de l'individu i. Donc

$$\Pr\left(Z_l^{(i,A_l)} = k\right) = q_k^i.$$

Le modèle de *structure* fait donc l'hypothèse que les marqueurs sont en équilibre de liaison, les loci sont indépendants. De plus, l'inférence est réalisée grâce à une méthode de Monte-Carlo par chaînes de Markov *MCMC*, très lente à utiliser en pratique.

Falush, Stephens et Pritchard (2003) ont développé *structure v.2* et décrit trois types de LD qui permettent de comprendre les données de métissage (Figure 2.4). Le premier type de LD, *mixture LD*, est dû aux variations de Q et révèle le lien entre

les coefficients de métissage pour chaque locus et le coefficient de métissage global. Reprenons l'exemple d'un individu afro-américain tel que $Q = (0.2, 0.8)$, alors quelque soit la position, le SNP a plus de chance de provenir d'une population africaine. Cette forme de LD était donc déjà prise en compte par la version originale de *structure*. Le deuxième type de LD, *admixture LD*, établit que deux marqueurs voisins auront tendance à provenir de la même population. Enfin le troisième type de LD, *background LD*, permet de représenter la corrélation entre les allèles observés A, il décroît à des échelles plus courtes le long des chromosomes (Falush, Stephens et Pritchard, 2003).

Modèles de métissage avec LD

Le logiciel *structure v.2* a introduit un modèle de liaison, *linkage model*, pour tenir compte de la corrélation entre les variables Z_l (*admixture LD*). Les haplotypes sont alors hérités par morceaux, le coefficient de métissage est constant sur des blocs de chromosomes. La variable Z suit un processus de Markov,

$$\Pr(Z_{l+1} = k | Z_0, Z_1, \ldots, Z_l) = \Pr(Z_{l+1} = k | Z_l)$$

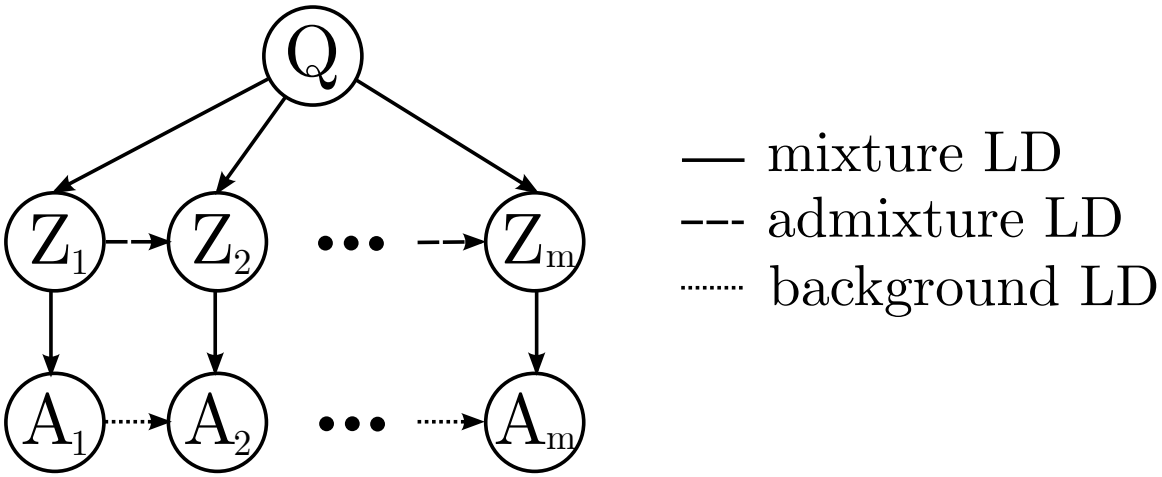

Figure 2.4 – Trois types de LD explicités par Falush, Stephens et Pritchard (2003). Q représente la variable aléatoire qui quantifie globalement les proportions de métissage. Chaque Z_i est une variable cachée indiquant la population de provenance du locus considéré. A_i représente les allèles observés. L'hypothèse markovienne impose une relation seulement entre les Z_i et Z_{i+1} ainsi qu'entre les A_i et A_{i+1}. Il est possible d'étendre ce modèle graphique au delà de l'hypothèse markovienne mais souvent au prix de la calculabilité.

Tang et al. (2006) tient compte du *background LD* à l'aide d'un *Markov Hidden*

Markov Model **MHMM** implémenté dans le logiciel SABER. Toutefois, cette méthode reste lente (Padhukasahasram, 2014).

D'autres méthodes sont apparues s'appuyant sur le modèle de Li et Stephens comme HAPMIX (Price et al., 2009), HAPAA (Sundquist et al., 2008) et LAMP-LD (Baran et al., 2012). HAPMIX étend le modèle de Li et Stephens pour inférer la population de provenance en supposant un évènement de métissage simple entre deux populations pop_1 et pop_2 au temps T. LAMP-LD et LAMP-HAP reprennent les concepts de HAPMIX mais proposent notamment deux modifications tout en restant dans le cadre du modèle de Li et Stephens. Tout d'abord, les recombinaisons se font entre les individus d'une même population le long d'une fenêtre. Une recombinaison entre deux populations différentes ne peut donc se faire qu'entre deux fenêtres. L'objectif est de contraindre les sauts et donc de lisser le résultat. Enfin, les haplotypes métisses ne sont pas décomposés selon les haplotypes de référence comme pour HAPMIX mais selon une structure de taille limitée k. Puisque ces méthodes, HAPMIX et LAMP-LD, ont une complexité quadratique vis-à-vis du nombre d'individus de référence, cette modification permet à LAMP-LD d'être un ordre de grandeur plus rapide que HAPMIX. HAPMIX et LAMP-LD sont parmi les logiciels les plus utilisés et les plus précis d'inférence des coefficients de métissage locaux (Liu et al., 2013 ; Hui et al., 2017). Nous nous comparons donc, en partie, à ces méthodes.

Des méthodes fondées en partie sur des algorithmes d'apprentissage automatique ont fait leur apparition : PCAdmix (Brisbin et al., 2012), SupportMix (Omberg et al., 2012) (SVM) ou RFMix (Maples et al., 2013) (forêts aléatoires). RFMix permet notamment de détecter du métissage de populations à des échelles géographiques très fines (Durand et al., 2014). RFMix est aussi inclus dans nos comparaisons. Nous présentons maintenant deux classifications qui illustrent ces changements de paradigme.

2.1.3 Modèles génératifs et discriminatifs

> One should solve the [classification] problem directly and never solve a more general problem as an intermediate step
>
> Vapnik, Vladimir N.

L'approche générale pour inférer les coefficients de métissage locaux était basée sur des modèles génératifs (Sankararaman et al., 2008 ; Price et al., 2009 ; Baran et al., 2012). Ces méthodes génératives modélisent d'abord la loi jointe des observations et

des paramètres cachés $\Pr(X, Y)$ ou de manière équivalente la vraisemblance $\Pr(X|Y)$ et la loi a priori $\Pr(Y)$. Par exemple, LAMP-LD et HAPMIX estiment $\Pr(X|Y)$ avec X les haplotypes et Y les classes via un HMM. Les classifieurs discriminatifs modélisent directement la loi a posteriori $\Pr(Y|X)$. Ng et Jordan (2002) comparent des paires de modèles génératifs/discriminatifs pour des modèles sous-jacents $\Pr(X, Y)$ communs. Pour les paires testées (Naives Bayes et Regression Logistique), ils concluent que les modèles discriminatifs ont une erreur asymptotique plus faible mais que les modèles génératifs convergent plus vite. De même, les classifieurs génératifs apprennent mieux que les classifieurs discriminatifs sur les jeux de données avec peu d'échantillons et inversement quand le nombre d'échantillons augmente. Cette analyse est reprise par RFMix (Maples et al., 2013) qui introduit une méthode discriminative pour effectuer le calcul des coefficients de métissage. Ainsi les méthodes discriminatives sont supposées se comporter mieux à l'heure où les jeux de données grandissent massivement. Il faut noter que d'autres méthodes discriminatives ont été publiées la même année que RFMix ou l'année précédente : EILA (Yang et al., 2013) et SupportMix (Brisbin et al., 2012). Néanmoins, Ng et Jordan (2002) ne généralisent pas leurs résultats pour tous les modèles, entre autres pour les modèles discriminatifs avec une pénalité comme le nôtre.

2.1.4 Modèle explicite vs Distance

Une autre classification des méthodes proposée par Pritchard, Stephens et Donnelly (2000) pour le clustering, valable pour le problème d'estimation des coefficients de métissage locaux (Padhukasahasram, 2014), sépare les méthodes *distance-based* et les méthodes *model-based*.

Les méthodes *distance-based*, non probabilistes (Bishop, 2006), établissent la répartition ou la classification selon une métrique soigneusement choisie. L'algorithme de *K-moyennes* ou l'algorithme des *k plus proches voisins* sont deux exemples de méthodes *distance-based*. Nous montrerons le lien entre nos méthodes et ces algorithmes.

Les méthodes *model-based*, probabilistes, supposent que les observations proviennent d'un modèle paramétrique. L'estimation de la classe est faite conjointement à l'estimation des clusters comme dans le modèle de mélange de gaussienne que l'on résout grâce à l'algorithme Espérance Maximisation, EM *Expectation-Maximization* (Bishop, 2006). Ainsi *structure* et HAPMIX sont fondées sur des modèles génétiques. Ces méthodes infèrent des paramètres potentiellement interprétables d'un point de vue biologique comme le taux de recombinaison.

Toutefois, ces nomenclatures ne sont ni exhaustives ni disjointes. Les approches récentes, comme RFMix, mêlent en réalité des concepts des deux types de catégories

en utilisant un classifieur localement discriminatif et en utilisant ces scores dans un module de raccordement et de lissage.

2.1.5 Utilité de l'estimation localisée des coefficients de métissage

Introgression adaptative

L'analyse des populations métisses requiert dans certaines situations de comprendre le métissage localement. La détection des transferts de variations génétiques par introgression entre les espèces est réalisée par des méthodes d'estimation des coefficients de métissage locaux. En effet, si un flux de gènes adaptatifs a lieu d'une espèce vers une autre alors la distribution des coefficients de métissage variera entre les régions liées aux gènes d'adaptation locale et le reste du génome. Suarez-Gonzalez et al., 2016 ont utilisé HAPMIX pour étudier l'introgression adaptatives d'espèces de peupliers *Populus trichocarpa* et *Populus balsamifera*. Les peupliers *Populus trichocarpa* et *Populus balsamifera* sont deux espèces soeurs ayant métissée en Amérique du Nord. L'espèce *P. Balsamifera* est la seule espèce adaptée au froid et présente dans les régions boréales au nord du Canada.

Suarez-Gonzalez et al. (2016) ont montré la présence d'une ascendance forte de *P. balsamifera* sur deux régions du chromosome 15 de métisses proches de *P. trichocarpa* (Figure 2.5).

Admixture Mapping

De nombreuses méthodes ont été développées pour corriger les tests d'association gène-maladie en détectant la structure de populations (Pritchard et al., 2000 ; Xu et Shete, 2005 ; Zhou et Stephens, 2012). Plutôt que de chercher des associations entre génotype et phénotype, il est aussi possible de détecter des associations entre l'ascendance et le phénotype, ce qu'on appelle *admixture mapping* (Seldin, 2007 ; Seldin, Pasaniuc et Price, 2011 ; Hoggart et al., 2004) (Figure 2.6).

L'*admixture mapping* repose sur la compréhension des différences des risques de maladie vis-à-vis de l'information du métissage et de l'ascendance locus par locus. Les logiciels typiques d'inférence des coefficients de métissage locaux (HAPMIX, LAMP-LD, PCAdmix, ...) sont utilisés dans ce type d'étude (Shriner, 2013). Par exemple, Freedman et al. (2006) et Bensen et al. (2013) ont utilisés l'*admixture mapping* pour identifier des allèles à risque pour le cancer de la prostate.

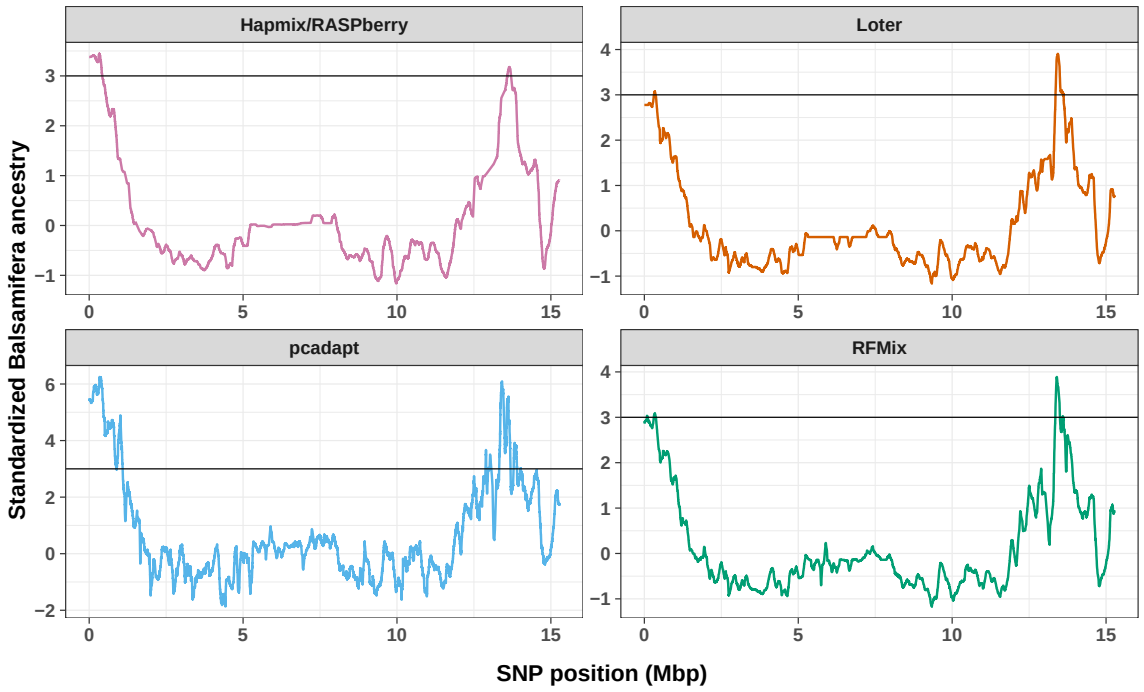

FIGURE 2.5 – Détection de région d'introgression pour de métisses de *P. trichocarpa* et *P. balsamifera*. Les zones d'introgression sont les régions possédant une ascendance *P. balsamifera* a plus de 3 s. d. (ligne noire). Comparaison de 3 méthodes d'estimation des coefficients de métissage locaux (le modèle d'HAPMIX implémenté dans le logiciel RASPberry, RFMix et Loter) et d'une méthode basée sur l'ACP (LUU, BAZIN et BLUM, 2017).

Démographie

Les logiciels d'estimation des coefficients de métissage locaux sont aussi utilisés pour étudier la structure de populations et les processus l'affectant tels que la sélection (TANG et al., 2007) et les migrations (BRYC et al., 2010). La compréhension du génome d'un individu comme une mosaïque de populations à des échelles fines facilitera la compréhension de l'histoire des populations et des personnes (DURAND et al., 2014 ; HELLENTHAL et al., 2014). Néanmoins, il convient d'établir précisément le modèle, les hypothèses et les biais des méthodes pour ne pas fausser les résultats. Par exemple, certaines méthodes prennent en paramètre le temps de métissage, l'estimation a posteriori de ce temps peut donc être biaisée par le paramètre en entrée.

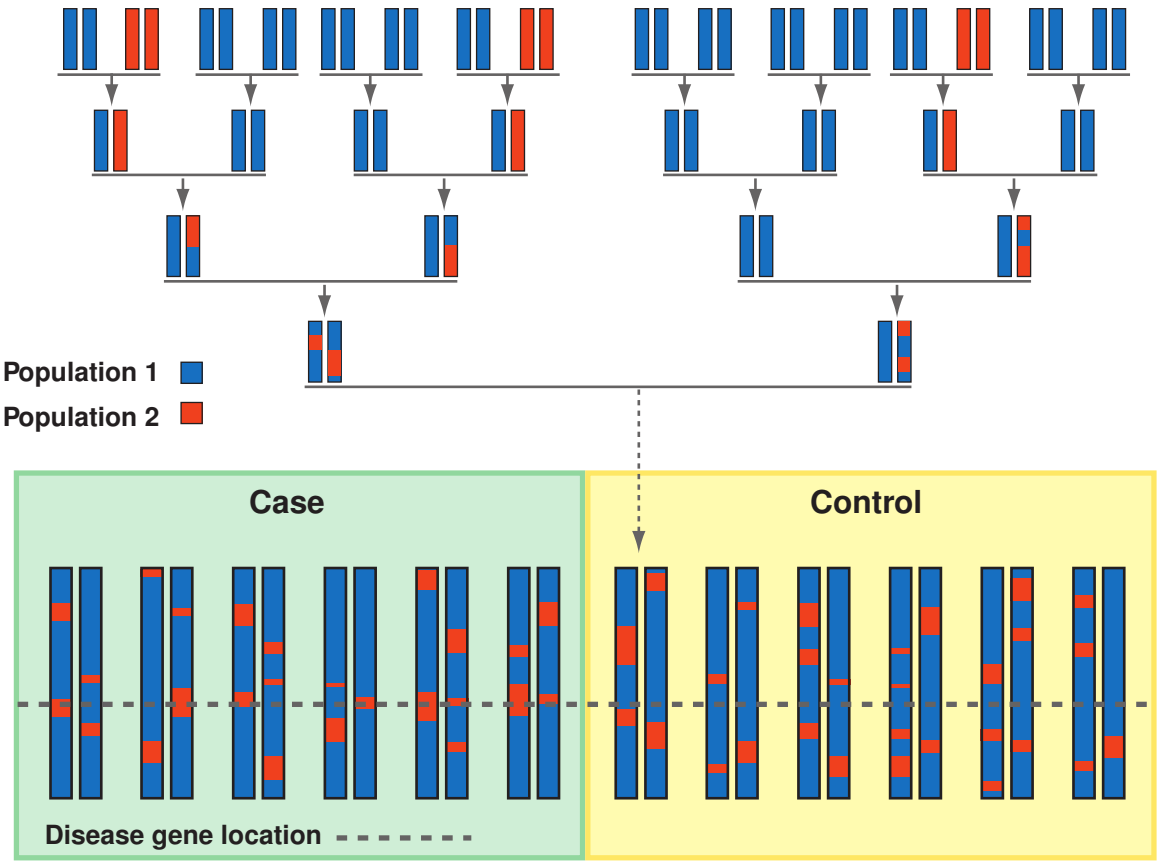

FIGURE 2.6 – Image tirée de DARVASI et SHIFMAN (2005) décrivant l'admixture mapping. Deux populations sont représentées en bleu et en rouge. La population rouge est supposée porteuse d'un allèle spécifique associée à la maladie et dont la fréquence est plus faible dans la population bleue. Les individus métisses sont échantillonnés selon la présence ou non de la maladie et divisés en cas et témoins. Les régions avec un excès d'ascendance "rouge", représentées en pointillés, seront donc associé à la maladie dans le cas de l'admixture mapping.

2.2 Données et Simulations

Nous présentons désormais les données sur lequelles se sont fondées nos expériences ainsi que les simulations utilisées pour évaluer la performance des méthodes.

2.2.1 Données : Populus et HapMap

Les mêmes jeux de données sont utilisés pour l'estimation des coefficients de métissage locaux, le phasage et l'imputation. Nos expériences ont donc été réalisées sur des données de deux natures : données humaines (HapMap3 THE INTERNATIONAL HAPMAP CONSORTIUM (2005)) et données sur des espèces de peupliers (Populus

Suarez-Gonzalez et al. (2016)).

HapMap permet de travailler avec des données phasées pour lesquelles de nombreux paramètres génétiques sont connus. Une carte génétique est disponible pour les données humaines. Pour HapMap, nous n'avons gardé que le premier chromosome, qui est le plus long avec 116415 marqueurs. Nous avons filtré 14231 positions dont la distance génétique n'était pas connue. De plus, HapMap3 release 2 est composé de trios (2 parents et 1 enfant) desquels nous n'avons gardé que les haplotypes de l'enfant. LAMP-LD étant limité à 50000 SNPs, nous avons donc simplement gardé les 50000 premiers SNPs dans les expériences incluant LAMP-LD.

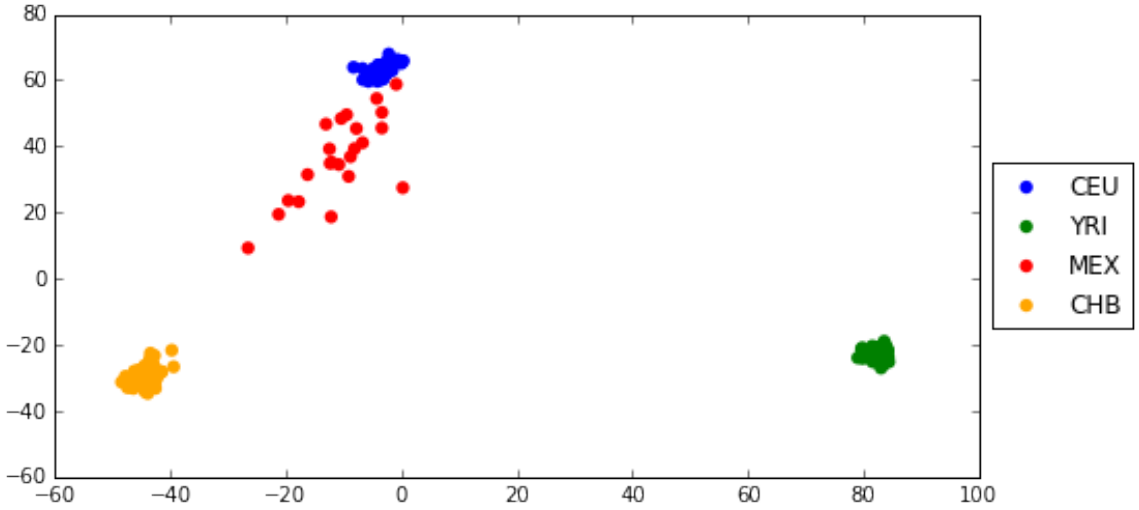

Figure 2.7 – Visualisation de quatre populations de HapMap, CEU, YRI, MEX et CHB sur les deux premiers axes de l'ACP.

Table 2.1 – Populations d'HapMap et leur nombre d'individus après filtrage.

population	CEU	YRI	MEX	CHB	JPT	TSI
nombre d'individu	44	50	23	84	86	88

Le jeu de données Populus est composé de 3 chromosomes : 6, 12 et 15. Nous avons utilisé le chromosome 6 pour l'étude de simulation, celui-ci est composé de 1418814 SNPs. Populus est composé de deux populations de référence, Populus balsamifera et Populus trichocarpa ainsi que de deux populations métisses. Ce jeu de données n'est pas phasé et contient $7,5\%$ de données manquantes. Le phasage et l'imputation ont été faits avec Beagle pour ne pas introduire de biais et comparer toutes les méthodes sur un pied d'égalité. Pour les simulations, nous gardons seulement les populations de référence balsamifera et Trichocarpa et générons des métisses.

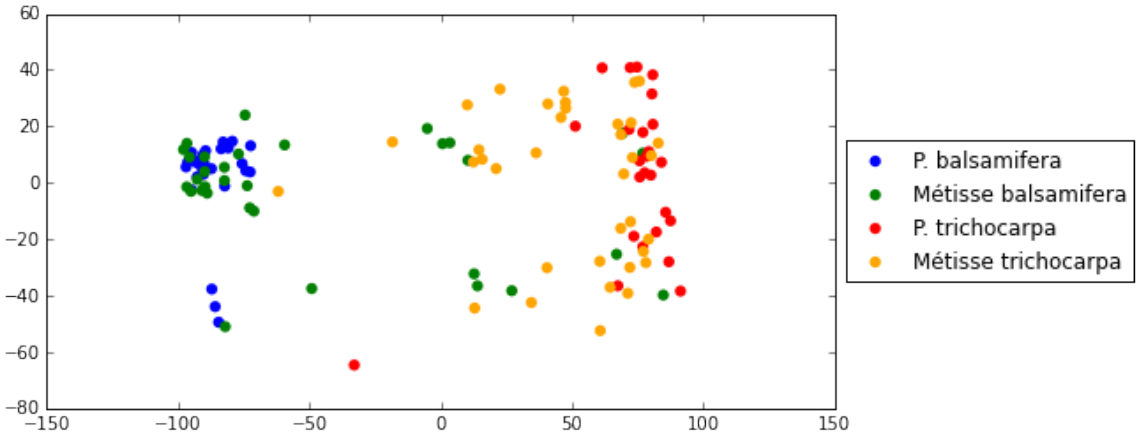

FIGURE 2.8 – Visualisation de quatre populations de peupliers sur les deux premiers axes de l'ACP.

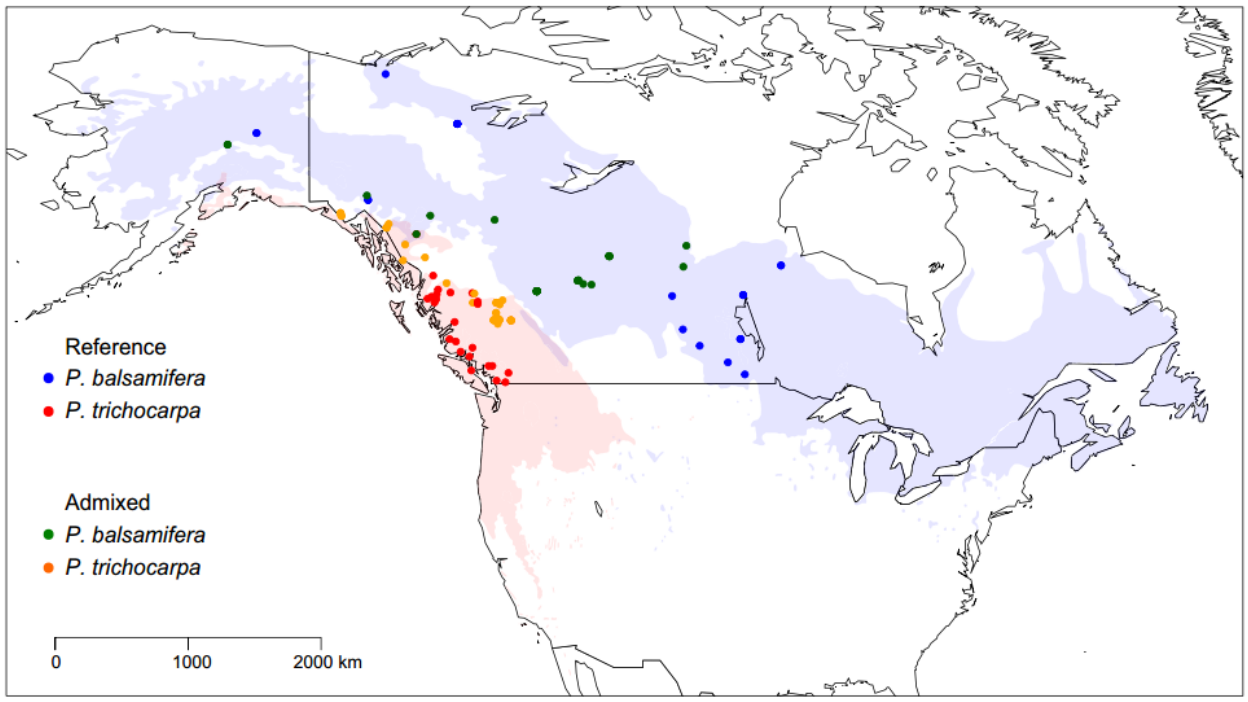

FIGURE 2.9 – Répartition géographique des peupliers du jeu de données Populus SUAREZ-GONZALEZ et al. (2016) REESE et LITTLE (1972).

Certains logiciels, ainsi que notre modèle de simulation, nécéssitent une carte des distances génétiques. Cette carte génétique est disponible pour les données humaines. Pour Populus, nous reprenons l'hypothèse faite par Suarez-Gonzalez et al. (2016) de 5 centiMorgans par million de paires de base. Le centiMorgan (cM) désigne l'unité de distance entre deux gènes. Cette distance mesure la liaison génétique comme la fréquence de recombinaisons à la méiose entre les positions. Ainsi, pour une distance de 1 cM entre deux gènes, on mesure en moyenne 1 recombinaison sur 100 méioses entre ces deux marqueurs génétiques.

La distance moyenne entre deux SNPs dans les données HapMap vaut $2.7.10^{-3}$ cM et $9.8.10^{-6}$ cM pour Populus. De ce fait, les simulations de métissages diffèrent beaucoup entre HapMap et Populus. Nous détaillons maintenant le principe des simulations.

2.2.2 Simulation du métissage

Pour tester l'efficacité des méthodes il faut des jeux de données contenant des populations ancestrales, des populations métisses et les labels de la provenance de chaque locus métisse par rapport aux populations de référence. Nous effectuons donc des simulations pour construire des populations métisses ainsi que la labellisation.

Le principal phénomène à modéliser pour le problème de l'estimation des coefficients de métissage locaux est la recombinaison. Les blocs de métissages sont en effet la conséquence d'un brassage dû aux recombinaisons auquel s'ajoutent d'autres phénomènes modifiant les génomes comme les mutations. Il s'avère que le processus de Poisson permet aussi de simuler l'effet des recombinaisons sur plusieurs générations. La superposition de n processus de Poisson de paramètre λ est un processus de Poisson de paramètre $n\lambda$. Liang et Nielsen (2014) et Gravel (2012) ont montré que l'hypothèse de segments indépendants et distribués exponentiellement était valide pour des temps de métissages moyens, de l'échelle de 10 - 200 générations [2] (Carmi, Xue et Pe'er, 2015). La probabilité d'apparition d'une recombinaison n'est pas uniforme le long du génome. Une carte génétique permet de prendre en compte de la distance génétique en cM entre les SNPs.

Les simulations suivent donc le modèle proposé par HAPMIX reposant sur un processus de Poisson (Price et al., 2009 ; Guan, 2014). Pour un haplotype d'un individu métisse, un coefficient de métissage global α est tiré selon une loi Beta de moyenne et variance μ et σ^2 afin de modéliser la variabilité du coefficient de métissage global au sein d'une population. Une recombinaison apparaît entre 2 SNPs distants de λ cM (centiMorgan) après n générations avec une probabilité de $1 - e^{n\lambda}$. A chaque

2. Le temps de métissage est naturellement exprimé en nombre de générations.

recombinaison, nous tirons selon une loi de Bernouilli de paramètre α la population de provenance de l'haplotype pour le bloc suivant. En conséquence, plus le métissage est ancien (nombre élévé de générations depuis le métissage) plus la longueur des blocs d'appartenance à une population est courte. Les jeux de données sont séparés en deux, une partie sert à construire les individus métisses tandis que l'autre échantillon est utilisé comme jeu d'entraînement pour estimer les coefficients de métissage locaux.

Enfin, pour simuler un métissage de k populations, nous tirons les coefficients de métissage globaux selon une loi de Dirichlet et tirons la longueur des segments selon une loi exponentielle de paramètre λ dépendant de la population. Afin de diminuer l'espace des paramètres de nos simulations, nous supposons que les $\lambda_1 = \lambda_2 = 2\lambda_3 = \ldots = 2^{k-2} * \lambda_k$. En pratique, nos simulations contiennent au plus trois populations.

FIGURE 2.10 – Exemple de simulation selon le modèle de HAPMIX indiquant le nombre de copies provenant de la deuxième population pour des individus diploïdes. Pour obtenir ce résultat, des coefficients de métissage des haplotypes métisses sont d'abord simulés puis sommés.

Pour les simulations avec deux populations, α est tiré selon la loi *beta* de moyenne 0.8 et d'écart type 0.1, des valeurs typiques pour les afro-américains (SMITH et al., 2004 ; PRICE et al., 2009).

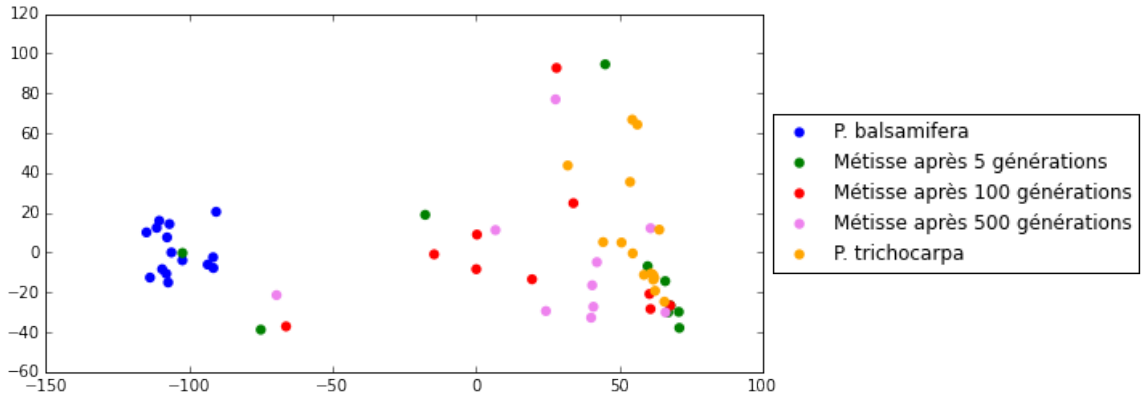

FIGURE 2.11 – Simulations de métisses en moyenne à 80% Trichocarpa et 20% Balsamifera pour différents nombre de générations depuis le métissage, représenté grâce à l'ACP.

2.3 Méthode

L'estimation des coefficients de métissage locaux est un problème de classification. Nous reprenons la notation en début de chapitre. Supposons que les jeux de données sont définis pour m loci. Soit $H^i \in \{0,1\}^{n_i,m}$ la matrice des n_i haplotypes de la population i et $H_{\text{ref}} = \{H^i, i \in \{1,\dots,k\}\}$ l'ensemble des matrices des k populations de référence. Un individu métisse est soit représenté par des haplotypes $h \in \{0,1\}^m$ soit par son génotypes $g \in \{0,1\}^m$. La résolution du problème consiste à trouver une application f qui pour chaque locus d'un individu métisse renvoie la population de provenance $Z \in \mathcal{Z} = \{1,\dots,K\}$. La population de provenance pour un individu diploïde est notée $D \in \mathcal{U} = \{\{y,y'\} : y,y' \in \mathcal{Z}\}$. De cette manière, à chaque locus l'information n'est pas ordonnée. Une méthode prend donc en entrée un individu métisse et renvoie en sortie les coefficients d'ascendance (noté Z et D selon la phase). Dès lors, trois cas de figures sont possibles.

— Entrée haploïde :

$$\begin{array}{rccc} f & : & \{0,1\}^m & \to & \mathcal{Z}^m \\ & & h & \mapsto & Z \end{array}$$

— Entrée diploïde, sortie non-phasée :

$$\begin{array}{rccc} f & : & \{0,1,2\}^m & \to & \mathcal{U}^m \\ & & g & \mapsto & D \end{array}$$

— Entrée diploïde, sortie phasée :

$$\begin{array}{rccc} f & : & \{0,1,2\}^m & \to & (\mathcal{Z}\times\mathcal{Z})^m \\ & & g & \mapsto & (Z,Z') \end{array}$$

Selon le type de sortie, deux erreurs sont possibles : l'erreur haploïde (Huang et al., 2009) pour les sorties phasées et l'erreur diploïde (Sankararaman et al., 2008) pour les sorties non-phasées [3]. Ces erreurs sont basées sur la distance de Hamming

$$d_H(x, y) = |\{i : x_i \neq y_i\}|$$

Pour une sortie phasée, nous définissons l'erreur haploïde ainsi

$$\begin{array}{rccc} e_{\text{hap}} & : & \mathcal{Z}^m \times \mathcal{Z}^m & \to & [0,1] \\ & & Z, Z' & \mapsto & \frac{d(Z,Z')}{m}. \end{array} \qquad \text{erreur haploïde}$$

Par exemple, si $K = 2$ et $m = 4$

$$\begin{aligned} Z &= (1,1,1,1) \\ Z' &= (2,2,1,1) \\ \text{alors } e_{\text{hap}}(Z, Z') &= \frac{2}{4} = 0.5 \end{aligned}$$

En revanche si la sortie n'est pas phasée, nous ne pouvons calculer que l'erreur diploïde, définie par

$$\begin{array}{rccc} e_{\text{dip}} & : & \mathcal{U}^m \times \mathcal{U}^m & \to & [0,1] \\ & & \{D, D'\} & \mapsto & \frac{d(D,D')}{m} \end{array} \qquad \text{erreur diploïde}$$

Par exemple, si $K = 2$ et $m = 4$

$$\begin{aligned} D &= (\{1,2\},\{1,2\},\{1,1\},\{1,1\}) \\ D' &= (\{1,2\},\{1,2\},\{1,1\},\{1,2\}) \\ \text{alors } e_{\text{dip}}(D, D') &= \frac{1}{4} = 0.25 \end{aligned}$$

Remarquons qu'il est possible d'avoir une sortie haploïde et de la convertir en sortie diploïde. Le D de l'exemple précédent représente l'information diploïde de l'ascendance de Z et Z' de l'exemple haploïde.

Le problème d'estimation des coefficients de métissage prend donc plusieurs formes selon la phase des données en entrée et en sortie. Ce problème est donc étroitement lié au phasage des haplotypes. Une des premières étapes courantes pour utiliser les méthodes d'estimation des coefficients de métissage est de phaser les individus ancestraux et potentiellement les individus métisses.

3. Les compléments à un de l'erreur haploïde et de l'erreur diploïde sont donc la précision haploïde et la précision diploïde respectivement.

Il s'agit de classification supervisée puisque les méthodes prennent aussi en entrée des ensembles d'haplotypes (H_{ref}) ou de génotypes des populations ancestrales pour lesquelles l'appartenance est donc connue.

Finalement, le problème d'estimation des coefficients d'ascendance locaux se fondent sur des données de métisses (phasées ou non) et au moins deux jeux de données de populations de référence, phasées dans la majeure partie des cas.

Considérons le cas le plus courant d'un métissage de deux populations avec en entrée des haplotypes. Il s'agit de construire une application f telle que :

$$\begin{array}{rccc} f & : & \{0,1\}^m & \to & \{pop1, pop2\}^m \\ & & h & \mapsto & Z \end{array}$$

Nous gardons le cadre du modèle de Li et Stephens en considérant qu'un haplotype peut être décomposé comme une mosaïque des autres haplotypes. Pour la problématique métissage, nous cherchons une décomposition parcimonieuse de l'haplotype métisse à l'aide des haplotypes des populations sources. C'est-à-dire une décomposition avec le moins de changements d'ascendance possibles. Soit $h \in \{0,1\}^m$ l'haplotype métisse et $H \in \{0,1\}^{n \times m}$ la matrice contenant les haplotypes de toutes les populations ancestrales. H_i^j représente donc le j-ième locus de l'individu i. S est le vecteur de dimension m, le nombre de loci, indiquant l'haplotype duquel on pioche. Enfin, λ représente le paramètre de régularisation. Dans le modèle de Li et Stephens, λ dépend du taux de mutation, du taux de recombinaison et éventuellement du nombre d'individus (Li et Stephens, 2003). Lorsque λ augmente, le nombre de sauts entre haplotypes diminue. Nous introduisons le problème d'optimisation suivant

$$\begin{array}{ll} \underset{S=(s_1,\ldots,s_n)}{\text{minimiser}} & \displaystyle\sum_{j=1}^{m} |h_j - H_j^{s_j}| + \lambda \sum_{j=1}^{m-1} 1_{s_j \neq s_{j+1}} \\ \text{avec} & s_j \in \{1, 2, \ldots, n\}. \end{array} \tag{2.1}$$

La résolution de ce problème d'optimisation est obtenue par une méthode de programmation dynamique. La solution optimale du problème s'appuie sur la solution optimale pour $m-1$ loci. Deux cas de figure sont possibles. Soit l'haplotype métisse n'a pas changé de sélection, $s_{m-1} = s_m$, alors :

$$\begin{array}{ll} \underset{S=(s_1,\ldots,s_{m-1})}{\text{minimiser}} & \displaystyle\sum_{j=1}^{m-1} |h_j - H_j^{s_j}| + \lambda \sum_{j=1}^{m-2} 1_{s_j \neq s_{j+1}} \\ & + |h_m - H_m^{s_{m-1}}|. \end{array} \tag{2.2}$$

Soit l'haplotype métisse a changé de sélection $s_{m-1} \neq s_m$:

$$\underset{S=(s_1,\ldots,s_{m-1})}{\text{minimiser}} \quad \sum_{j=1}^{m-1} |h_j - H_j^{s_j}| + \lambda \sum_{j=1}^{m-2} 1_{s_j \neq s_{j+1}} + |h_m - H_m^{s_m}| + \lambda. \tag{2.3}$$

La solution optimale du problème est obtenue en calculant le plus court chemin sur un graphe représentant tous les états possibles de S, c'est-à-dire un graphe de taille $n \times m$ (Figure 2.12). En effet, en notant (i, m) le noeud des chemins passant par l'individu i au locus m, le coût$_{(i,m)}$ du plus court chemin suit la formule récurrente suivante :

$$\text{coût}_{(i,m)} = \min\left(\text{coût}_{(i,m-1)}, \lambda + \min_{j\in\{1,\ldots,n\}}\{\text{coût}_{(j,m-1)}\}\right) + |h_m - H_m^i| \tag{2.4}$$

En stockant $\min_{j\in\{1,\ldots,n\}}\{\text{coût}_{(j,m-1)}\}$ à chaque étape, il suffit de calculer le minimum entre deux valeurs pour chaque noeud du graphe. Cet algorithme a donc un coût computationnel en $\mathcal{O}(n \times m)$ pour chaque individu métisse avec n le nombre d'individus de référence et m le nombre de loci. Le chemin obtenu S est converti en coefficients d'ascendance $z \in \{1, \ldots, K\}$ avec K le nombre de populations ancestrales. Il suffit d'attribuer à z_j la population de provenance de s_j.

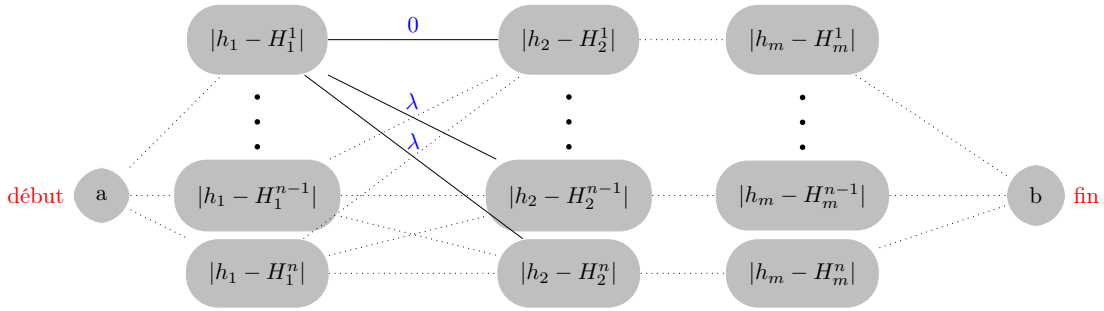

FIGURE 2.12 – Graphe de taille $n \times m$ qui optimise l'équation 2.1. Une valeur de S optimale est obtenue en trouvant le plus court chemin entre l'état a et b.

Algorithme 1 : Calcul de S pour des données haplotypiques

Data : $h \in \{0,1\}, H \in \{0,1\}^{n,m}, \lambda$
Result : $S \in \{1,\dots,n\}$
$\mathrm{G} \in \{1,\dots,n\}^{n,m}$, le graphe des chemins
$\mathrm{Coût} \in \mathbb{R}^{n,m}$, la matrice de coût pour chaque noeud
u, le coût minimum au rang précédent
v, l'indice du minimum au rang précédent
Initialisation :
$\mathrm{Coût}_{\bullet,1} \leftarrow \ell(H_{\bullet,1}, h_1)$
$u \leftarrow \min(\mathrm{Coût}_{\bullet,1})$
$v \leftarrow \mathrm{argmin}(\mathrm{Coût}_{\bullet,1})$
for $j \in \{2,\dots,m\}$ `// Forward`
do
    for $i \in \{1,\dots,n\}$ **do**
        if $Coût_{i,j-1} \leq \lambda + u$ **then**
            $\mathrm{Coût}_{i,j} \leftarrow \mathrm{Coût}_{i,j-1}$
            $\mathrm{G}_{i,j} \leftarrow i$
        else
            $\mathrm{Coût}_{i,j} \leftarrow \lambda + u$
            $\mathrm{G}_{i,j} \leftarrow v$
        $\mathrm{Coût}_{i,j} \leftarrow \mathrm{Coût}_{i,j} + \ell(H_{i,j}, h_j)$
    $u \leftarrow \min\{\mathrm{Coût}_{\bullet,j}\}$
    $v \leftarrow \mathrm{argmin}\{\mathrm{Coût}_{\bullet,j}\}$
for $j \in \{m-1,\dots,1\}$ `// Backward`
do
    $S_j \leftarrow v$
    $v \leftarrow G_{v,j}$

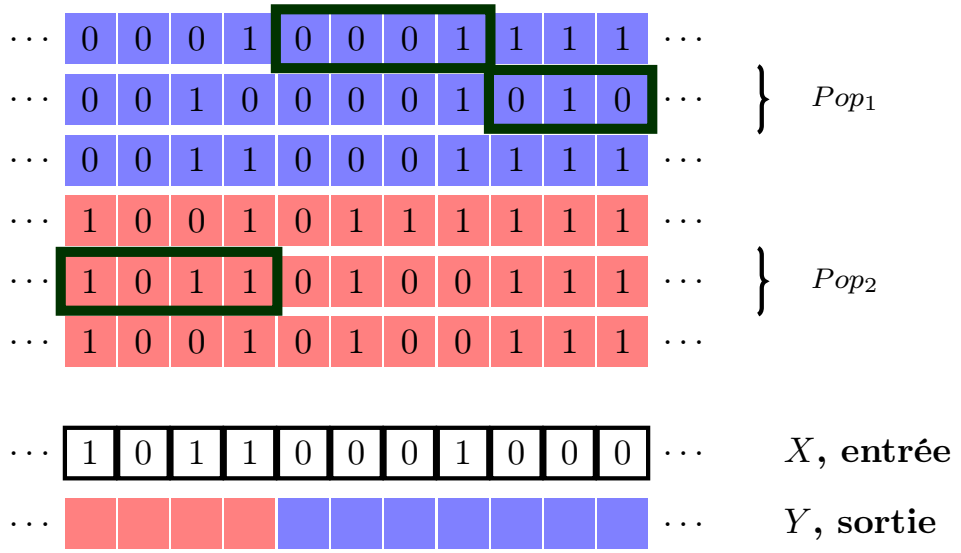

FIGURE 2.13 – Représentation graphique du problème d'inférence de Loter. On cherche le meilleur chemin parmi les haplotypes ancestraux en pénalisant les sauts par λ et les erreurs par 1. Dans cet exemple, le coût du chemin encadré vaut donc $2\lambda + 1$ puisque 2 sauts sont nécessaires et une valeur entre l'haplotype de référence et l'haplotype métisse est différente au 10ème SNP.

Théorême 3.1: Plus proche voisin

Lorsque $\lambda \to \infty$ cette méthode correspond à l'algorithme du plus proche voisin pour la norme ℓ_1. Avec $\ell_1(X) = \sum_i |X_i|$. :

$$\underset{s=\{1,\dots,n\}}{\text{minimiser}} \quad \sum_{j=1}^{m} |h_j - H_j^s| \tag{2.5}$$

Démonstration

Puisque les valeurs sont toutes comprises entre 0 et 1, on a :

$$\sum_{j=1}^{m} |h_j - H_j^{s_j}| \leq m, \quad \forall s_j \in \{1, \dots, n\}^m$$

Supposons que $S = (s_1, \dots, s_{m-1})$ soit tel qu'il existe s_i et s_{i+1} avec $s_i \neq s_{i+1}$. Alors

$$\sum_{j=1}^{m} |h_j - H_j^{s_j}| + \lambda \sum_{j=1}^{m-1} 1_{s_j \neq s_{j+1}} \geq \sum_{j=1}^{m} |h_j - H_j^{s_j}| + \lambda$$

Si $\lambda \to \infty$, alors en particulier $\lambda > m$.

$$\sum_{j=1}^{m} |h_j - H_j^{s_j}| + \lambda > m$$

Ceci n'est pas une solution acceptable puisque pour S tel que $\forall i \in \{0, \dots, n-1\}, s_i = s_{i+1}$.

$$\begin{aligned} \sum_{j=1}^{m} |h_j - H_j^{s_j}| + \lambda \sum_{j=1}^{m-1} 1_{s_j \neq s_{j+1}} &= \sum_{j=1}^{m} |h_j - H_j^{s_j}| \\ &\leq m \end{aligned}$$

On en déduit que le problème à minimiser est

$$\underset{s=\{1,\dots,n\}}{\text{minimiser}} \quad \sum_{j=1}^{m} |h_j - H_j^{s}|$$

s correspond à l'indice de l'haplotype qui est le plus proche. □

Théorême 3.2: Décomposition en sous-chaînes

Lorsque $0 < \lambda < 0.5$, cette méthode décompose l'haplotype métisse en un nombre minimum de segments ancestraux identiques au métisse.
Soit $\mathcal{J} = \{j : \exists i \in \{1, \dots, n\}, h_j = H_j^i\}$ l'ensemble des loci tel que l'haplotype métisse correspond à au moins un haplotype de H. Alors on minimise,

$$\begin{aligned} &\underset{S=(s_1,\dots,s_n)}{\text{minimiser}} && \lambda \sum_{j=1}^{m-1} 1_{s_j \neq s_{j+1}} \\ &\text{avec} && h_j = H_j^{s_j}, \forall j \in \mathcal{J}. \end{aligned} \tag{2.6}$$

Démonstration

Soit S' le chemin réalisant le minimum de l'équation 2.1 et c le coût associé,

$$S' = \underset{S=(s_1,\dots,s_n)}{\operatorname{argmin}} \sum_{j=1}^{m} |h_j - H_j^{s_j}| + \lambda \sum_{j=1}^{m-1} 1_{s_j \neq s_{j+1}}.$$

$$c = \min_{S=(s_1,\dots,s_n)} \sum_{j=1}^{m} |h_j - H_j^{s_j}| + \lambda \sum_{j=1}^{m-1} 1_{s_j \neq s_{j+1}}.$$

Il faut démontrer que le chemin S' ne passe que par des $H_j^{s'_j}$ égaux à h_j et que ce chemin fait le minimum de changements.

Supposons qu'il existe $j \in \mathcal{J}$, tel $h_j \neq H_j^{s'_j}$. Alors il existe i tel que $h_j = H_j^i$. Le coût associé au chemin $S^2 = (s'_1, \dots, s'_{j-1}, i, s'_{j+1}, \dots, s'_m)$ vaut

$$\begin{aligned} \sum_{j=1}^{m} |h_j - H_j^{s_j^2}| + \lambda \sum_{j=1}^{m-1} 1_{s_j^2 \neq s_{j+1}^2} &\leq \sum_{j=1}^{m} |h_j - H_j^{s_j^2}| - 1 + \lambda \sum_{j=1}^{m-1} 1_{s_j^2 \neq s_{j+1}^2} + 2\lambda \\ &\leq c - 1 + 2\lambda \\ &< c. \end{aligned}$$

Ce qui est absurde par définition de c. Alors nous avons démontré que S' est tel que

$$\sum_{j=1}^{m} |h_j - H_j^{s'_j}| = |\mathcal{J}|.$$

$$h_j = H_j^{s'_j}, \forall j \in J$$

On en conclut qu'il ne reste qu'à minimiser le nombre de sauts

$$\begin{aligned} &\underset{S=(s_1,\dots,s_n)}{\text{minimiser}} && \lambda \sum_{j=1}^{m-1} 1_{s_j \neq s_{j+1}} \\ &\text{avec} && h_j = H_j^{s_j}, \forall j \in \mathcal{J}. \end{aligned}$$

□

Ces théorèmes montrent l'importance du choix du paramètre de régularisation λ. Pour un λ très fort, nous obtenons une solution très lisse. En revanche, pour un λ faible les sauts peuvent être fréquents. Ce paramètre de régularisation est donc lié au paramètre de la loi exponentielle simulant la longueur des segments d'haplotype pour le métissage. Il convient donc de trouver une méthode robuste au choix de la pénalité ou alors de proposer une méthode pour l'estimer.

Remarquons tout d'abord que dans le cas de données haplotypiques phasées par un logiciel, les segments sont plus courts (Figure 2.14). En effet, chaque erreur de phasage sur un segment permute les coefficients de métissage. Le paramètre de lissage dépend donc aussi de la qualité du phasage des données.

2.3.1 Méthode d'ensemble et combinaison de modèles

> Diversity and independence are important because the best collective decisions are the product of disagreement and contest, not consensus or compromise.
>
> James Surowiecki

Les méthodes d'ensemble combinent les résultats de plusieurs modèles pour obtenir un meilleur résultat, en termes de biais et de variance. Ces procédures prédisent ou régressent à l'aide d'apprenants faibles. Ces méthodes ont connu un fort succès dans divers secteurs de l'apprentissage automatique (Ali, Tirumala et Sarrafzadeh, 2015 ; Chen et He, 2014) et sont utilisées dans des outils populaires d'analyse de données comme XGBoost (Chen et Guestrin, 2016).

Le *bagging* (*Bootstrap AGGregating* Breiman, 1996) est une des méthodes d'ensemble possible qui repose sur la méthode de bootstrap pour améliorer les prédictions. Le *bootstrap* (Efron et Tibshirani, 1993), approche la distribution d'un estimateur d'un échantillon de loi inconnue en rééchantillonant avec remise parmi les observations initiales. Cet échantillon est appelé échantillon *bootstrap*. Le *bagging* apprend donc sur les k échantillons *bootstrap* et ensuite vote pour obtenir un consensus. Dans notre cas, nous rééchantillonons donc les individus des populations de référence. Ainsi, nous tenons compte de l'incertitude sur les individus dans la population de référence.

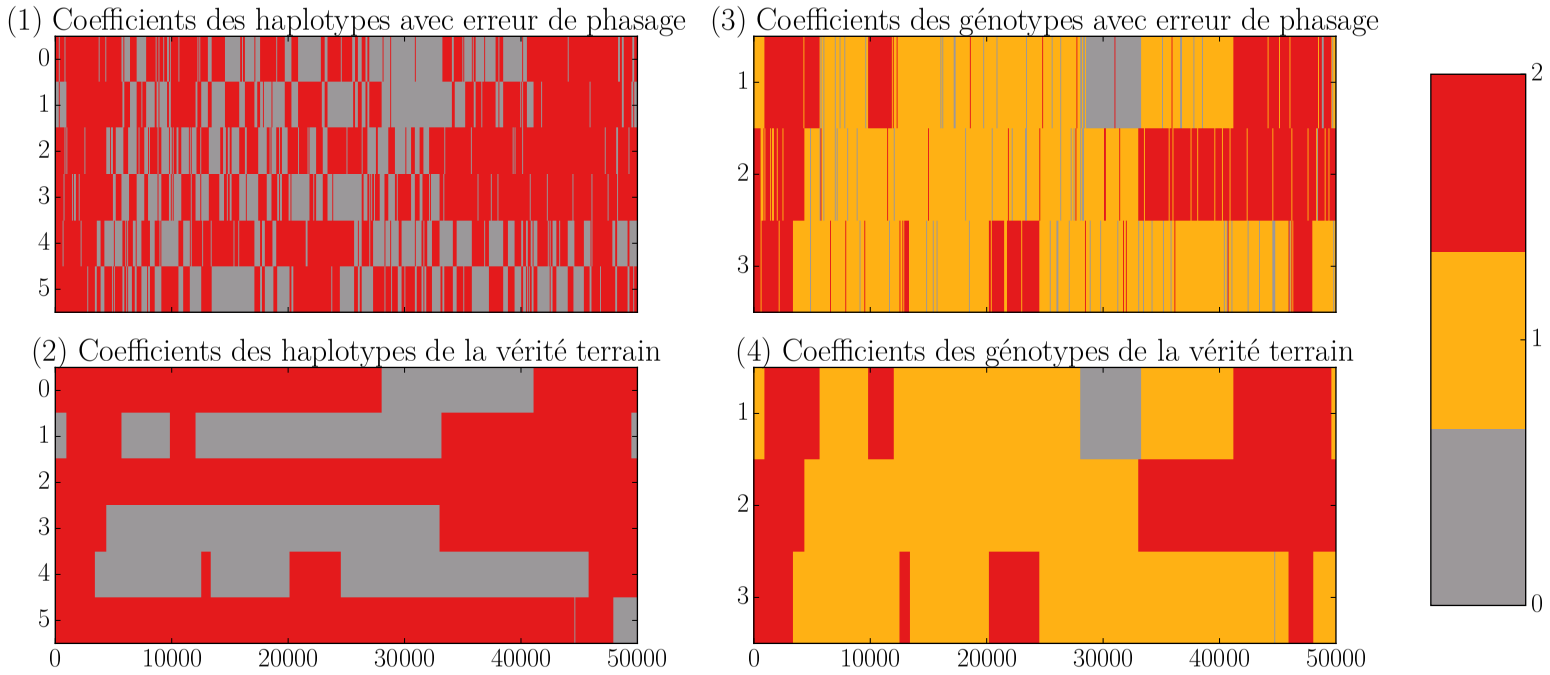

FIGURE 2.14 – Impact des erreurs de phasage sur l'estimation des coefficients de métissage locaux. La figure (1) représente les coefficients de métissage locaux estimés par Loter d'haplotypes phasés avec Beagle sans module correcteur de phase. (2) est la matrice des vrais coefficients d'ascendance pour les haplotypes simulés (avant phasage avec Beagle). (3) et (4) représentent l'information diploïde de (1) et (2) respectivement, c'est-à-dire l'information compactée à chaque locus en perdant la phase. Pour les méthodes prenant en entrée des haplotypes estimés avec des logiciels de phasage populationnel, les segments d'ascendance seront fragmentés du fait des erreurs de phasage. Cela illustre donc l'importance des modules correcteurs de phase pour ces méthodes.

De plus, pour tenir compte de l'incertitude sur le paramètre de régularisation λ, nous intégrons sur $\lambda \in [1.5, 5]$ avec un pas de 0.5 par défaut. En tout, l'algorithme calcule donc $k \times 8$ plus courts chemins dans le graphe. En pratique $k = 20$, donc le bagging vote sur 160 résultats.

Une méthode de validation croisée de Monte Carlo est disponible pour constater si l'intervalle $[1.5, 5]$ est adéquat. La validation croisée est une technique de sélection de modèle et de paramètre. La validation croisée de Monte Carlo consiste à apprendre le modèle sur un sous-échantillon aléatoire et indépendant de valeurs d'entraînement. Un certain pourcentage de valeurs de génotypes est masqué et le modèle infère ces données manquantes comme détaillé dans le chapitre 3 3. Le taux d'erreur d'imputation indique alors la validité du paramètre. En pratique, l'intervalle $[1.5, 5]$ est robuste et nous a permis de traiter tous les jeux de données. λ s'interprète en nombres de mauvaises copies acceptables. Le théorème 3.1 illustre comment un fort λ autorise les mauvaises copies mais interdit les recombinaisons alors qu'un λ faible (théorème 3.2) interdit les erreurs de copies. Donc $\lambda \in [1.5, 5]$ couvre déjà des taux d'erreurs du simple au triple.

2.3.2 Module correcteur de phase et d'erreur

Les erreurs rencontrées sont principalement de deux types. Les erreurs de phasage des haplotypes (Figure 2.14) tendent à découper les blocs d'ascendance en plus petits blocs et donc à fausser l'hypothèse de continuité. Dans le cas où les deux haplotypes proviennent de deux populations ancestrales différentes, il se peut que l'étape de phasage ait permuté des segments d'haplotypes sur de très courtes distances. À ce titre, les premières approches d'estimation des coefficients de métissage locaux comme HAPMIX et LAMP prenaient seulement des données non-phasées en entrée pour ne pas sous-estimer constamment la longueur des segments d'haplotype et pour que le paramètre du temps de métissage (cas de HAPMIX) soit valide. En présence d'erreurs de phase, les méthodes auront tendance à trop lisser le résultat puisqu'elles ne suspectent pas que les blocs ont été découpés. Le deuxième type d'erreur est dû, au contraire, à la détection erronée de petits fragments lorsque le paramètre de lissage est trop faible.

Nous proposons deux approches pour tenir compte de l'incertitude du phasage. La première approche est une version génotypique, c'est-à-dire qui prend simplement des génotypes en entrée. Nous n'évaluerons pas cette approche car de complexité quadratique par rapport au nombre d'haplotypes de référence. La deuxième approche est un module correcteur de phase qui permet de rétablir la phase des haplotypes.

Version génotypique

Notre modèle a été adapté pour les données génotypiques des métisses. Tout comme HAPMIX et LAMP-LD, nous cherchons désormais une paire de plus courts chemins dans le graphe. Soit $s, s' \in \{1, \ldots, n\}^m$ représentant les indices desquels les haplotypes h et h' copient. Nous ajoutons la contrainte que la somme des haplotypes est égale au génotype.

$$\begin{aligned} \underset{S,S',h,h'}{\text{minimiser}} \quad & \sum_{j=1}^{m} |h_j - H_j^{s_j}| + \sum_{j=1}^{m} |h'_j - H_j^{s'_j}| \\ & + \lambda \sum_{j=1}^{m-1} 1_{s_j \neq s_{j+1}} + \lambda \sum_{j=1}^{m-1} 1_{s'_j \neq s'_{j+1}} \\ \text{avec} \quad & s_j, s'_j \in \{1, 2, \ldots, n\}. \\ & h + h' = g \end{aligned} \tag{2.7}$$

Le principe d'optimalité de Bellman s'applique de la même manière que pour la version haplotypique.

Chercher une paire de plus courts chemins est équivalent à chercher le plus court chemin dans un graphe avec $n^2 \times m$ noeuds. En notant (i, j, m) le noeud des chemins passant par l'individu i et l'individu j au locus m, alors le coût $\text{coût}_{(i,j,m)}$ du plus court chemin suit la formule récurrente suivante :

$$\begin{aligned} \text{coût}_{(i,j,m)} = \quad \min \big(& \text{coût}_{(i,j,m-1)}, \\ & \lambda + \min_{k \in \{1,\ldots,n\}} \{\text{coût}_{(k,j,m-1)}\} \\ & \lambda + \min_{k \in \{1,\ldots,n\}} \{\text{coût}_{(i,k,m-1)}\} \\ & 2\lambda + \min_{k,k' \in \{1,\ldots,n\}} \{\text{coût}_{(k,k',m-1)}\}\big) \\ & + |h_m - H_m^i| \end{aligned} \tag{2.8}$$

Il faut alors stocker $2n - 1$ minima entre chaque locus. Cette fois-ci l'algorithme est donc en $\mathcal{O}(n^2 \times m)$ pour chaque individu métisse. La force de ce modèle réside dans sa capacité à estimer la phase grâce au génotype de l'individu et aux haplotypes de référence. Nous avons étendu ce modèle pour des données de référence non phasées. $\widetilde{H}$ désigne un phasage quelconque des haplotypes ancestraux. Nous ne pénalisons plus le changement d'haplotype si l'on change pour l'haplotype d'un même individu. Nous définissons la pénalité p entre deux sélections x et y par

$$p(x, y) = \begin{cases} 0 & \text{si } x \,/\, 2 = y \,/\, 2, \text{ (" / " désigne le quotient de la division euclidienne)} \\ \lambda & \text{sinon.} \end{cases}$$

Alors le problème pour traiter des génotypes pour les individus métisses et de référence devient :

$$\begin{aligned} \underset{S,S',h,h'}{\text{minimiser}} \quad & \sum_{j=1}^{m} |h_j - \widetilde{H}_j^{s_j}| + \sum_{j=1}^{m} |h'_j - \widetilde{H}_j^{s'_j}| \\ & + \sum_{j=1}^{m-1} p(s_j, s_{j+1}) + \sum_{j=1}^{m-1} p(s'_j, s_{j'+1}) \qquad (2.9) \\ \text{avec} \quad & s_j, s'_j \in \{1, 2, \ldots, n\}. \\ & h + h' = g \end{aligned}$$

Avec cette formulation, peu importe la phase de $\widetilde{H}$ puisqu'il est possible de transitionner entre les haplotypes d'un individu de référence une pénalité nulle.

Ces formulations ont l'avantage de résoudre le problème des erreurs de phasage mais leurs complexités algorithmiques ne permettent pas des analyses aussi rapides que pour les versions haplotypiques. Il est préférable de phaser une fois les données et ensuite d'appliquer les méthodes sur les haplotypes plutôt que de phaser au sein de la méthode pour chaque individu métisse.

Module correcteur de phase

Les problèmes d'erreur de phasage illustrés par la figure 2.14 s'interprètent comme un problème d'estimation de la phase. À partir des coefficients des génotypes estimés (graphique (3) de la Figure 2.14), on souhaite trouver la phase des coefficients de métissage pour les haplotypes (2). L'idée est alors d'utiliser l'algorithme 2.10 non pas sur les marqueurs génétiques mais sur les coefficients de métissage. Soit Z et Z' les vecteurs contenant les coefficients d'ascendance d'un individu. Notons D les coefficients ancestraux de cet individu diploïde, tel que $D_j = Z_j - 1 + Z'_j - 1$. D_j compte le nombre d'allèles au locus j qui proviennent de la deuxième population. Dans le cas d'un métissage à 2 populations, nous avons $Z \in \{1, 2\}^m$ et $D \in \{0, 1, 2\}^m$. Par exemple, si au locus 10, le premier individu a pour coefficient d'ascendance $Z_{10} = 1$ et $Z'_{10} = 1$ alors $D_{10} = 0$. La matrice des coefficients d'ascendance des populations de référence A découle naturellement,

$$A = \begin{pmatrix} 1 & \ldots & 1 \\ 2 & \ldots & 2 \end{pmatrix}$$

Alors le module correcteur de phase consiste à trouver une paire de chemins dans

Z qui reconstruisent D.

$$\begin{aligned} \underset{s,s',Z,Z'}{\text{minimiser}} \quad & \sum_{j=1}^{m} |Z_j - A_j^{s_j}| + \sum_{j=1}^{m} |Z'_j - A_j^{s'_j}| \\ & + \beta \sum_{j=1}^{m-1} 1_{s_j \neq s_{j+1}} + \beta \sum_{j=1}^{m-1} 1_{s'_j \neq s_{j'+1}} \qquad (2.10) \\ \text{avec} \quad & s_j, s'_j \in \{1, 2, \ldots, n\}. \\ & Z_j - 1 + Z' - 1 = D_j \end{aligned}$$

s et s' contiennent la solution corrigée du problème d'estimation des coefficients de métissage locaux pour chaque haplotype, notons $d_{corr} = s + s'$ la solution diploïde. Toutefois, cette méthode requiert elle aussi un paramètre de lissage noté β. β est sélectionné par dichotomie entre 0 et 500 tel que la différence entre d et d_{corr} soit inférieure à 90%.

Ce module peut se généraliser à k populations. A devient

$$A = \begin{pmatrix} 1 & \ldots & 1 \\ 2 & \ldots & 2 \\ \vdots & & \\ k & \ldots & k \end{pmatrix}$$

Mais désormais d n'est plus la somme des informations d'ascendance des haplotypes. d_j doit permettre de reconstruire les valeurs a_j et a'_j sans ordre. Néanmoins, pour $k = 3$, $\{a_j = 0, a'_j = 3\}$ et $\{a_j = 1, a'_j = 2\}$ donnerait $d_j = 3$. Pour adapter notre algorithme sans changer la dimensions des objets, il suffit de trouver une matrice symétrique E de taille $k \times k$ avec des valeurs différentes pour la partie triangulaire inférieure. Ainsi $d = E[a, a'] = E[a', a]$.

$$E = \begin{pmatrix} 0 & 1 & 3 & \cdots & \binom{k}{2} \\ 1 & 2 & 4 & \cdots & \binom{k}{2} + 1 \\ 3 & 4 & 5 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \binom{k}{2} & \binom{k}{2} + 1 & \cdots & \cdots & \binom{k}{2} + k - 1 \end{pmatrix}$$

De cette manière le module correcteur de phase peut être adapté au cas de plus de deux populations se métissant. La complexité est alors en $\mathcal{O}(k^2 \times m)$.

Correction du vote

La dernière manière de corriger les erreurs potentielles est de tenir compte du vote obtenu après *bagging* et intégration sur le paramètre λ. Pour de nombreux loci, le

vote peut se faire à 50/50. Dans ce cas, le consensus est probablement mauvais. Les votes pour lesquelles le ratio est inférieur à 75% sont considérés comme des valeurs manquantes. Ces valeurs sont imputées par la valeur non manquante du locus le plus proche.

2.3.3 Implémentation de la méthode

Les méthodes présentées dans ce chapitre ont été implémentées en C++ et intégrées dans un module Python.

Récapitulatif des paramètres de Loter :

— valeurs sur lesquelles intégrer λ

par défaut $[1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5]$

— nombre d'échantillons *bootstrap* pour le *bagging*

par défaut **20**

— ratio de vote pour le consensus

par défaut **75%**

— ratio de similarité pour la correction des erreurs de phase

par défaut **90%**

— nombre de *threads* pour le parallélisme

TABLE 2.2 – Différence entre les entrées et paramètres des méthodes (le vert correspond au mode le plus simple pour l'utilisateur). Cette table illustre les différences entre les méthodes et les modes de fonctionnement des logiciels.

	HAPMIX	LAMP-LD	RFMix	Loter
Métisses non phasés	oui	oui	non	les deux
Nombre de populations	2	2, 3 ou 5	≥ 2	≥ 2
Fenêtré	non	oui	oui	non
Carte génétique en cM	oui	non (positions physiques)	oui	non
Ancestraux non phasés	non	oui	non	non
Correction de la phase (cas haplotype)			oui	oui (pour 2 populations)
Implémentation parallélisée	non	non	oui	oui

2.4 Résultats

2.4.1 Estimation des coefficients de métissage locaux de matrices de SNPs

Pour évaluer l'efficacité de Loter, nous avons évalué la précision diploïde pour des métisses afro-américains, simulés à partir des populations européennes **CEU** et Yoruba **YRI** de HapMap. Le métissage de deux populations a aussi été évalué sur des données de peupliers, *Populus balsamifera* et *Populus trichocarpa*. Les métisses sont générés avec des coefficients de métissage globaux de 80% (YRI et *trichocarpa*) et 20% (CEU et *balsamifera*). Les temps de métissage entre les populations varient entre 5 et 500 générations. Toutefois, la distance génétique entre deux SNPs successifs diffèrent d'un facteur 100 entre Populus et HapMap. Les simulations sont donc de natures différentes, les coefficients d'ascendance seront plus lisses pour Populus puisque le taux de recombinaison est plus faible entre deux SNPs. Chaque jeu de données a été divisé en deux, une partie pour simuler les individus et l'autre pour l'apprentissage. De plus, sauf contre indication, chaque jeu de données résulte du phasage par Beagle (Browning et Browning, 2011) des populations ancestrales et de la population métisse simultanément. Les coefficients d'ascendance ont aussi été estimés avec HAPMIX, une des premières méthodes de l'état de l'art, LAMP-LD et RFMix, une méthode aux résultats très prometteurs et proposant une approche discriminative novatrice.

Néanmoins, HAPMIX a été supprimé de notre analyse comparative. En effet, les méthodes LAMP-LD, RFMix et Loter obtiennent des précisions diploïdes supérieures à 99% pour un nombre de générations depuis le métissage égal à 5. Alors que la précision de HAPMIX est de 56% (Figure 2.15). HAPMIX obtient des précisions diploïdes excellentes lorsque les paramètres et les données contiennent peu d'erreurs. Pour illustrer cette remarque, nous avons créé quatre jeux de données avec différentes configuration. D1 est la configuration la plus facile possible. D'une part, nous fournissons à HAPMIX le vrai nombre de générations depuis le métissage utilisé dans la simulation. D'autre part, le jeu de données contient les vrais haplotypes de HapMap et contient tous les individus (incluant ceux utilisés pour les simulations). Le jeu de données D2 ne contient pas les individus utilisés pour la simulation. Pour la configuration D3, nous ne donnons plus le temps de métissage en paramètre à HAPMIX. Enfin, la configuration D4 correspond à une configuration réaliste imposée dans les autres expériences : le temps de métissage est inconnu, les individus parents des simulations sont exclus et toutes les données sont phasées avec Beagle. La figure 2.15 indique que pour des données phasées avec Beagle, la précision diploïde de HAPMIX passe de 99% à 56%.

Cette expérience met en avant la sensibilité de HAPMIX vis-à-vis de la phase des

données. Bien que HAPMIX prenne en entrée des métisses non phasés, il est nécessaire que la phase des populations ancestrales soit connue. Toutefois, une expérience dans laquelle tous les paramètres sont connus et les données parfaitement phasées profiterait à HAPMIX. Mais dans ce cas de figure, les autres méthodes verraient aussi leurs précisions augmenter.

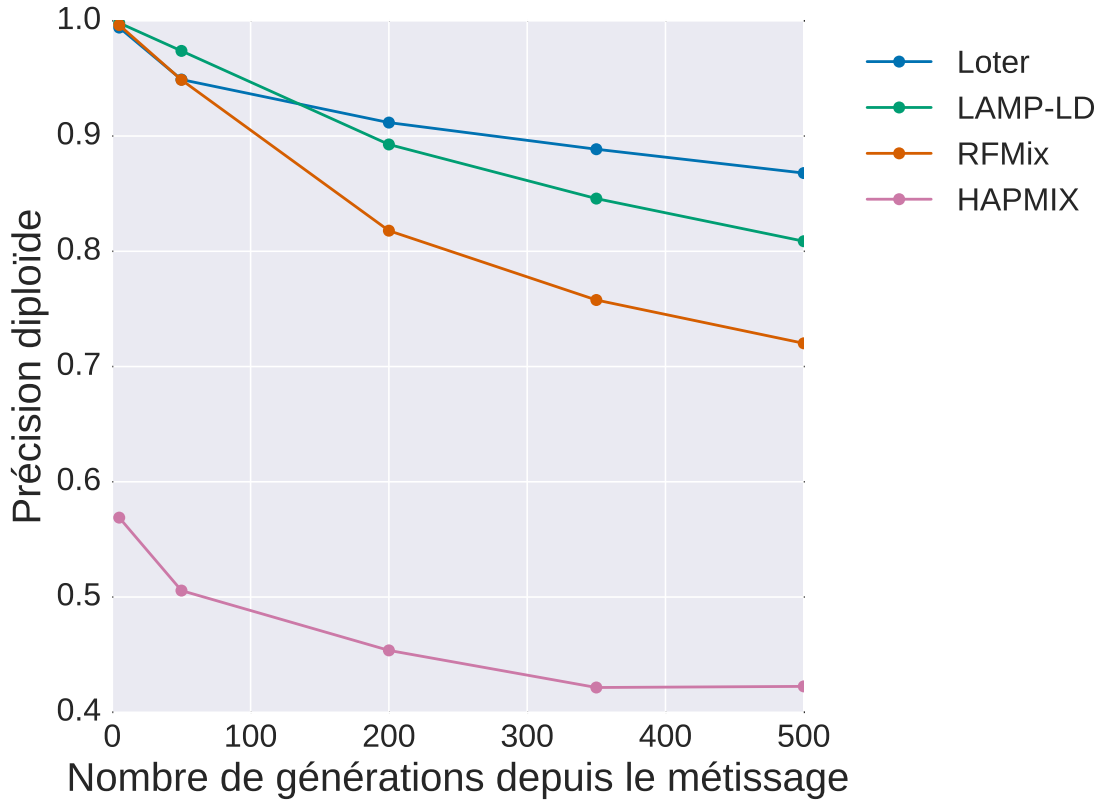

FIGURE 2.15 – Comparaison avec HAPMIX sur les données de métissage de CEU et YRI (HapMap). Ce graphique synthétise les résultats de quatre méthodes : HAPMIX, LAMP-LD, RFMix et Loter. Les résultats de HAPMIX étant nettement inférieurs aux autres méthodes, nous avons exclu HAPMIX du reste de notre analyse.

RFMix, LAMP-LD et Loter perdent en précision pour les métisses lorsque le temps depuis le métissage augmente (jeu de données CEU-YRI) (Figure 2.17). Les trois méthodes ont plus de 99% de précision pour 5 générations. La précision des méthodes décroit à 72%, 81% et 87% pour RFMix, LAMP-LD et Loter respectivement. Néanmoins, LAMP-LD réalise un meilleur score pour les temps de métissage faibles, entre 5 et 100 générations. Au-délà de 100 générations la précision de Loter l'emporte sur les

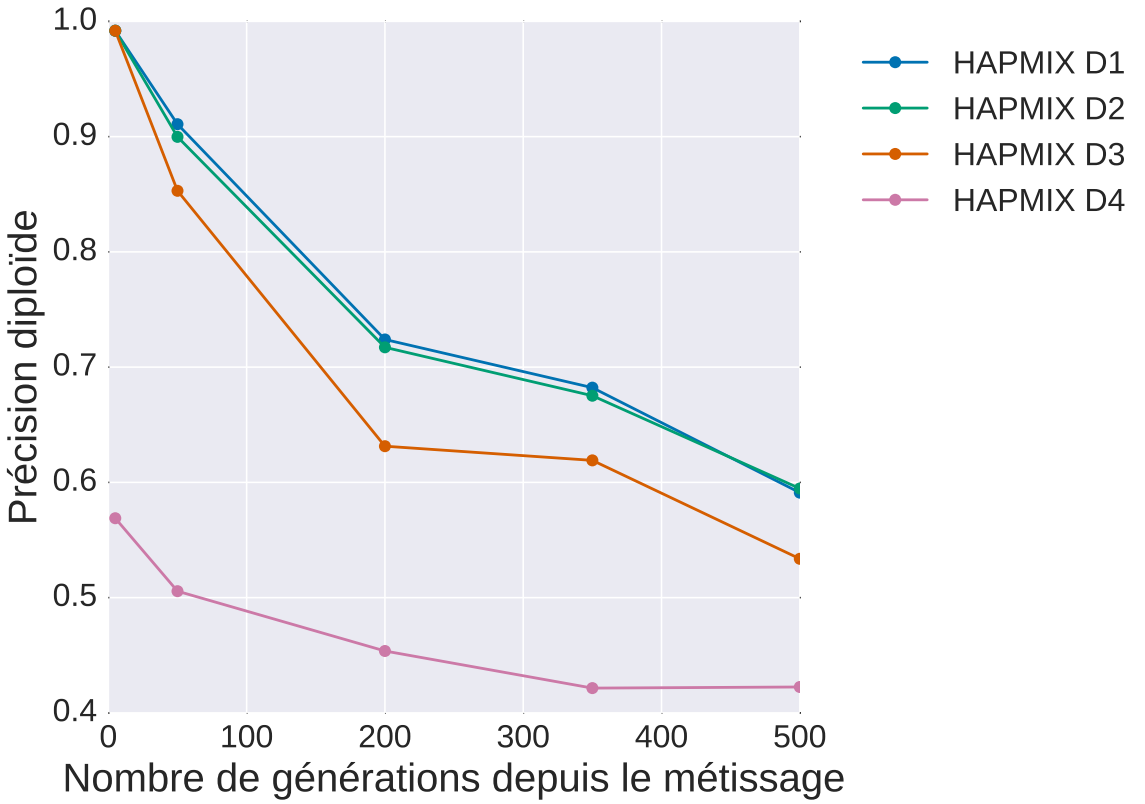

FIGURE 2.16 – Comparaison avec HAPMIX pour différents modes d'expérience. D1 : tous les individus, vraie phase et vrai temps de métissage (idéal). D2 : individus différents (entre les simulations et l'apprentissage) mais vraie phase. D3 : individus différents mais vraie phase et vrai temps de métissage. D4 : individus différents (entre train et test) et phasés avec Beagle.

autres méthodes. Néanmoins, RFMix prend en paramètre le nombre de générations depuis le métissage. Cette valeur vaut 8 par défaut. La méthode est naturellement mieux paramétrée pour les métissages récents. Nous avons noté que le paramètre du temps de métissage est bien plus robuste pour RFMix que pour HAPMIX. Nous avons donc répété l'expérience en donnant le vrai temps de métissage à RFMix. La précision de RFMix augmente mais n'égale toujours pas la précision de Loter pour les temps de métissage ancien (Figure 2.18). La précision de RFMix augmente dès 50 générations avec le vrai temps de métissage en paramètre, en passant de 95% de précision diploïde à 96% en moyenne sur 20 simulations. À 500 générations, la précision a gagné 10% (de 72% à 82%). Néanmoins, cela reste inférieur aux scores de Loter (87% de précision à 500 générations).

Le même type d'expérience a été réalisé sur les données de peupliers (Figure 2.19). Ce jeu de données permet de tester les méthodes non seulement sur des données d'une espèce différente, mais aussi de densité et de taille génétique très différentes (138 cM pour HapMap et 5 cM pour Populus). Les segments seront donc nettement plus denses en SNPs même pour 500 générations. Soulignons que nous fournissons à RFMix la carte génétique et à LAMP-LD la liste des positions physiques des marqueurs. Loter ne prend aucun paramètre d'échelle et est donc la seule méthode à ne pas connaître a priori cette différence entre les deux jeux de données. Pourtant, les performances de Loter varient entre 90% et 89% en moyenne sur 20 simulations. LAMP-LD gagne en moyenne 1% de précision sur l'ensemble des simulations. En revanche RFMix chute pour plus de 50 générations jusqu'à atteindre 65% de précision pour 500 générations. Cet échec de RFMix est dû au fenêtrage de la méthode. Malgré la carte génétique fournie en paramètre, RFMix découpe par défaut les génotypes en fenêtres de 0.2 cM. Ce paramètre réduit considérablement la précision maximale possible de RFMix sur Populus. RFMix découpe les génotypes en seulement 25 fenêtres. De plus, la méthode ralentit plus les SNPs dans une fenêtre sont nombreux, car les forêts aléatoires sont alors entraînées sur plus de données.

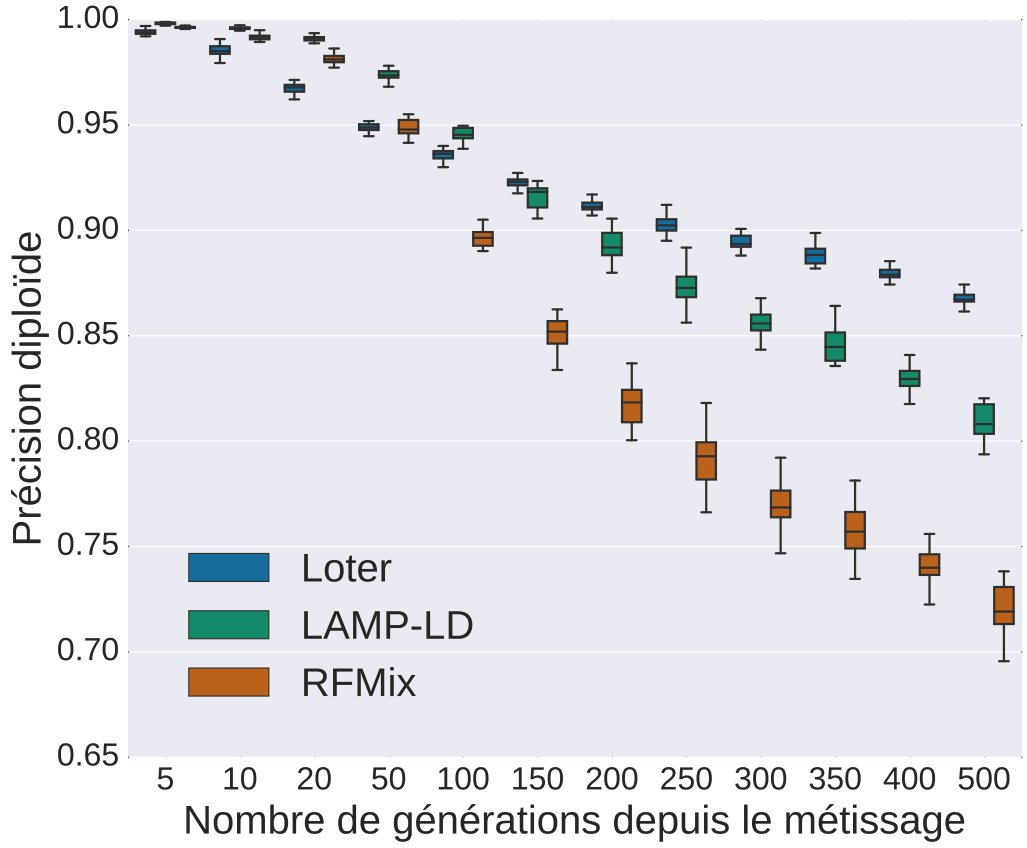

FIGURE 2.17 – Comparaison de Loter, LAMP-LD et RFMix sur des simulations de métisses entre CEU et YRI de HapMap pour un coefficient de métissage global de (0.20, 0.80). Les boites à moustache représentent la répartition des précisions diploïdes pour 20 simulations différentes par temps de métissage. Une simulation est constituée de 40 haplotypes européens, 40 haplotypes africains et de 48 haplotypes métisses.

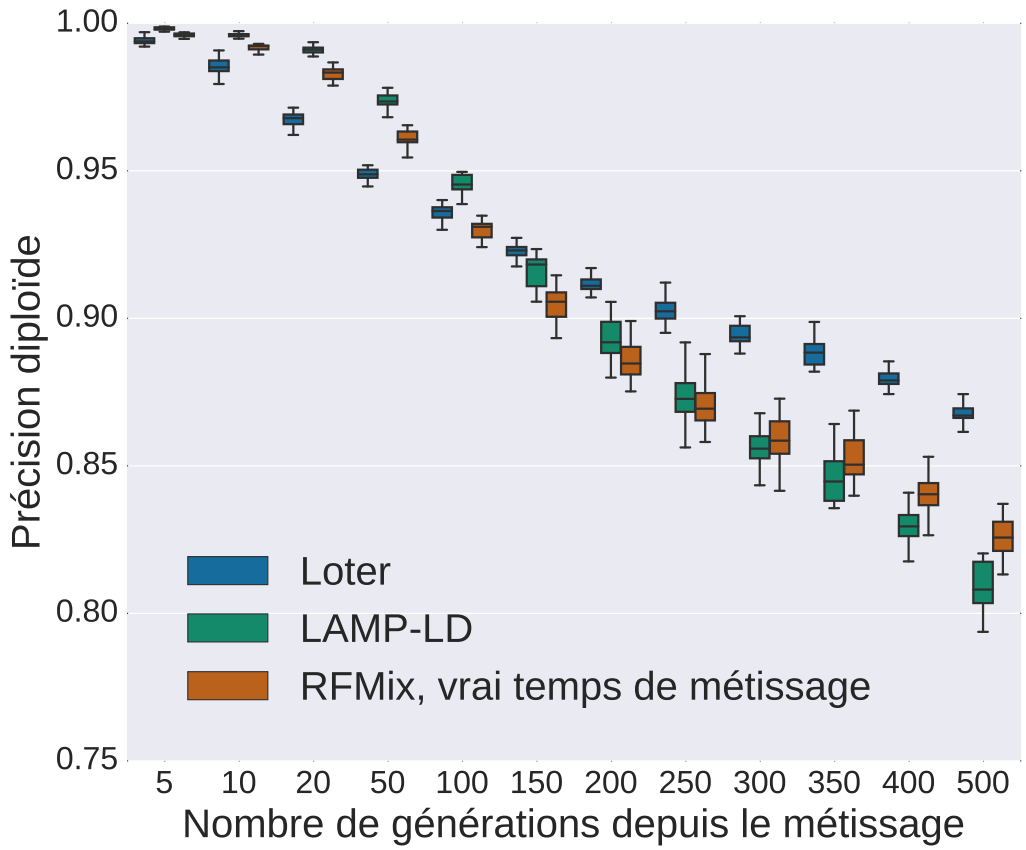

FIGURE 2.18 – Nous reprenons la même configuration d'expérience que pour la figure 2.17 en fournissant cette fois-ci le vrai nombre de générations depuis le métissage à RFMix.

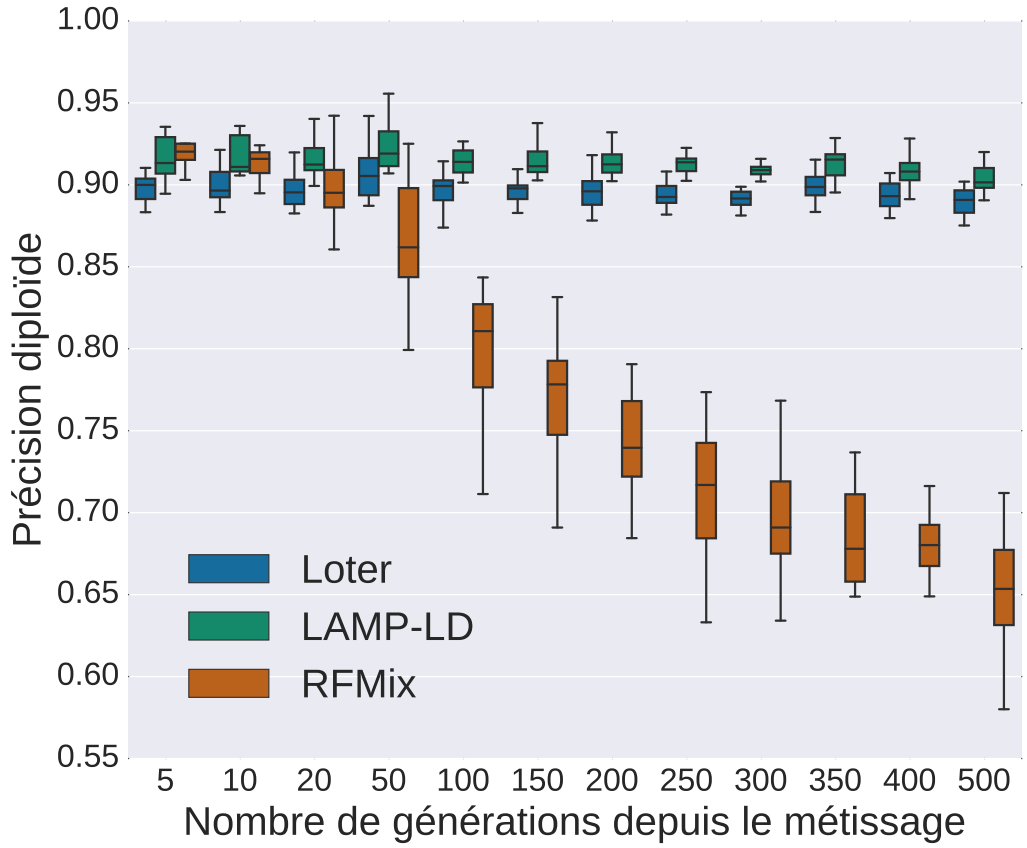

FIGURE 2.19 – Comparaison de Loter, LAMP-LD et RFMix sur des simulations de métisses entre les peupliers *trichocarpa* et *balsamifera* pour un coefficient de métissage global de (0.80, 0.20). Les 50000 premiers SNPs sont seulement distants de 5.2 cM. Pour chaque temps de métissage, 20 simulations de 25 individus métisses sont effectuées.

2.4.2 Moyennage et Bagging

Une étape importante pour améliorer la précision de Loter repose sur le moyennage des résultats de la méthode 2.1 en faisant varier le paramètre λ et les populations de référence. La figure 2.20 représente la précision de la méthode Loter pour $\lambda \in \{1, 2, 5, 10\}$. Les différents paramètres de régularisation optimisent différents nombres de générations. $\lambda = 10$ optimise la précision pour 5 générations (92.4%) alors que pour 500 générations depuis le métissage, ce paramètre est le moins bon avec seulement 83% de précision. Un λ élevé restreint les changements de coefficient d'ascendance, information a priori valable pour les faibles temps de métissage. Ainsi, les courbes associées aux différents paramètres s'intersectent et laissent penser que λ doit dépendre du temps de métissage pour résoudre correctement le problème. Néanmoins, en calculant le vecteur consensus des vecteurs associés à chaque λ, nous obtenons une solution meilleure en tout point. D'autre part, nous considérons que les populations de référence sont une donnée incertaine. Que dire si en observant d'autres individus des mêmes populations les coefficients de métissage changent drastiquement ? Le *bagging* permet de tenir compte de cette incertitude et d'augmenter la précision diploïde. En rééchantillonnant 20 fois les individus des populations de référence nous augmentons la précision de 1.2% en moyenne sur cette simulation. Dans toutes les expériences réalisées, le *bagging* augmente la précision de manière significative, entre autres pour les populations de référence parfois labellisées avec erreur.

La qualité de la phase des données impacte aussi la précision des méthodes et les paramètres de régularisation de celles-ci. Sur les simulations CEU/YRI, nous considérons la précision des méthodes sur les données phasées du jeu de données HapMap grâce à la méthode des trios (Figure 2.21) et sur les données phasées grâce à Beagle. Pour les données phasées avec Beagle, nous constatons que la précision est maximale pour $\lambda = 5$ indépendamment du temps de métissage. En pratique, le λ de Loter est intégré entre 1.5 et 5. Pour des données phasées avec des trios, la précision est globalement meilleure et on remarque que pour 5 générations depuis le métissage, le paramètre λ est un choix pertinent (précision supérieure à 98%) pour un large intervalle de $\lambda = 5$ à $\lambda = 100$.

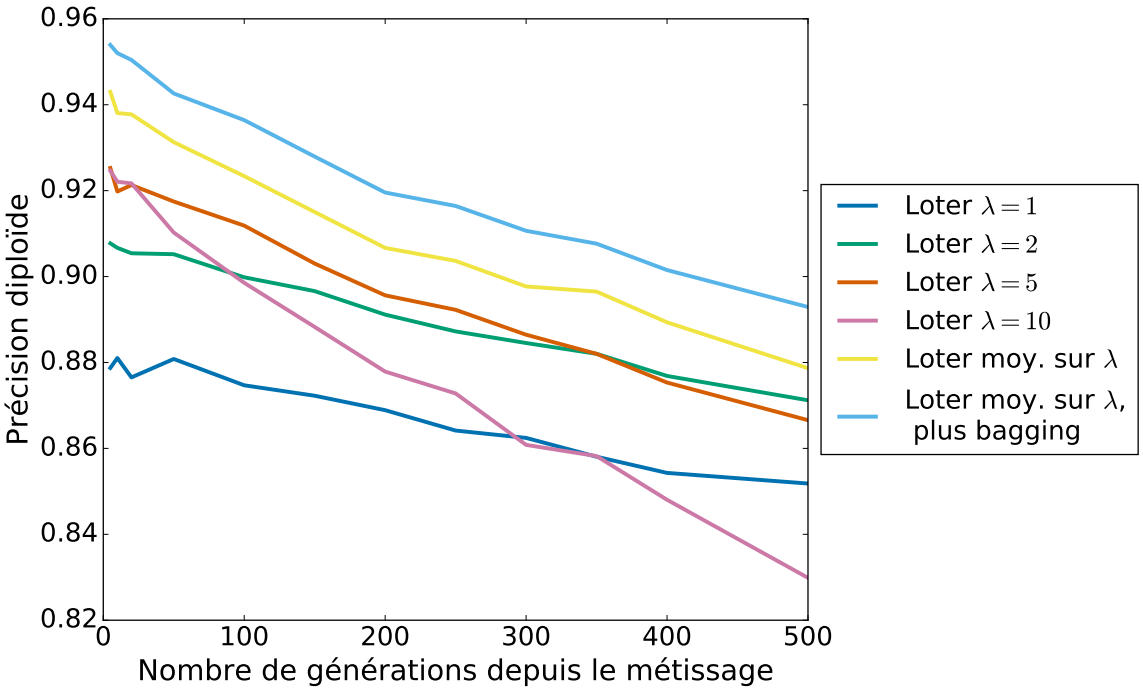

FIGURE 2.20 – Effet du moyennage et du *bagging* sur des simulations de métisses CEU et YRI. La précision diploïde est tracée en fonction du temps de métissage pour $\lambda \in \{1, 2, 5, 10\}$. Grâce aux quatre vecteurs de coefficients d'ascendance, un vecteur consensus est créé par vote. "Loter moy. sur λ" représente la précision diploïde de ce vote. Enfin, le *bagging* multiplie le nombre de votes (par 20 dans cette expérience) et permet d'améliorer la précision de notre modèle.

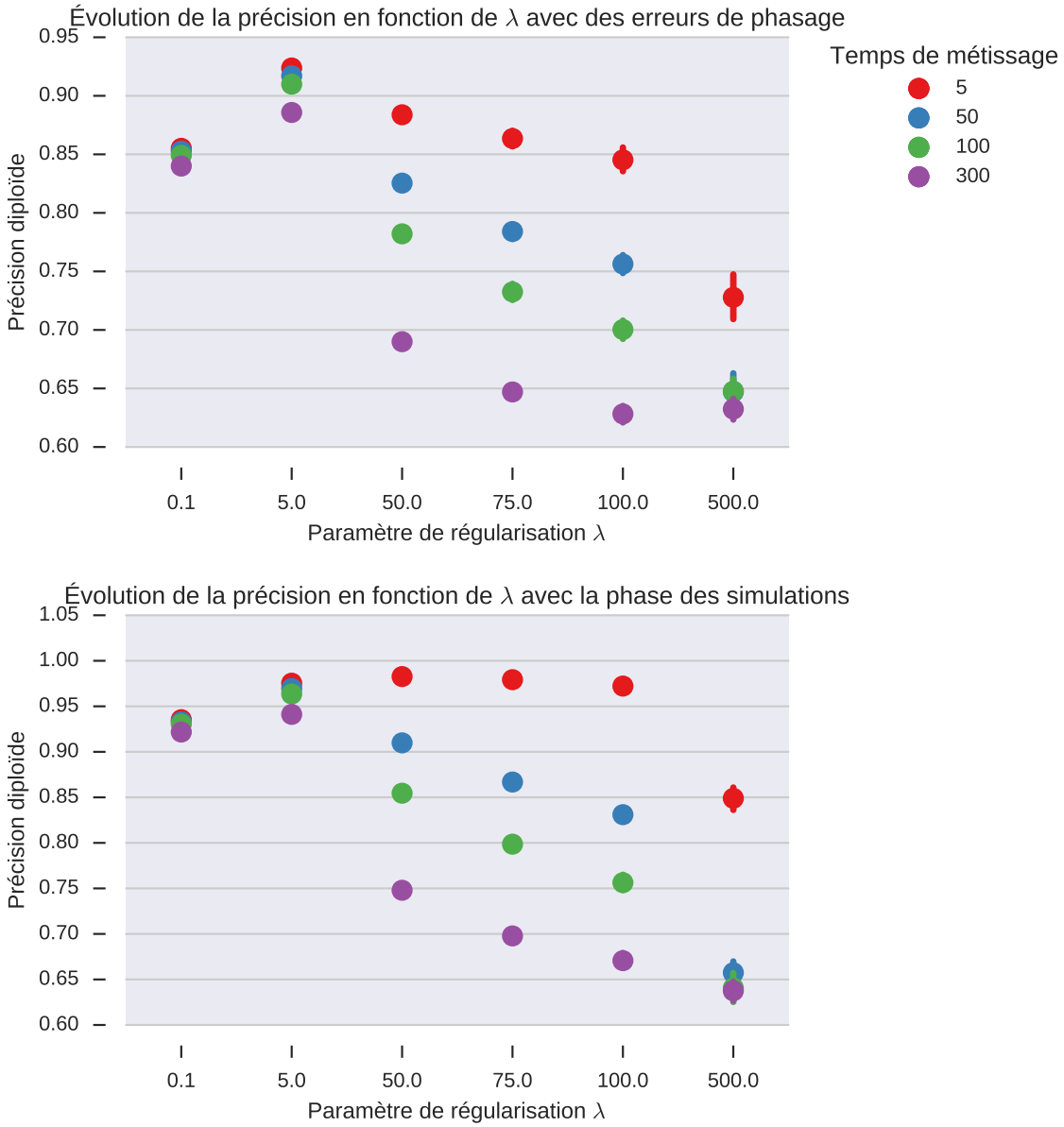

FIGURE 2.21 – Précision diploïde en fonction du temps de métissage et du paramètre λ pour deux types de données : les données phasées avec Beagle (figure supérieure) et les données avec la phase utilisée pour les simulations (figure inférieure). Pour chaque temps de métissage et chaque paramètre λ, 20 simulations sont effectuées. La précision de Loter, pour un λ donné, depend du temps de métissage et de la phase des données.

2.4.3 Validation croisée

En parallèle à l'intégration de la méthode sur le paramètre λ, nous avons développé une méthode d'évaluation du paramètre λ (Figure 2.22). 50% des SNPs sont masqués aléatoirement et sont inférés grâce aux populations de référence. Un λ correct doit minimiser l'erreur d'imputation. Néanmoins, la précision de l'imputation est strictement décroissante en fonction de λ. Ce critère semble simplement montrer que des valeurs faibles de λ sont optimales pour notre algorithme. En pratique, cette procédure n'est donc pas utilisée sur les résultats expérimentaux que nous présentons ici. Toutefois, la validation croisée peut être une première étape de l'analyse d'un jeu de données avec Loter pour estimer l'échelle de λ.

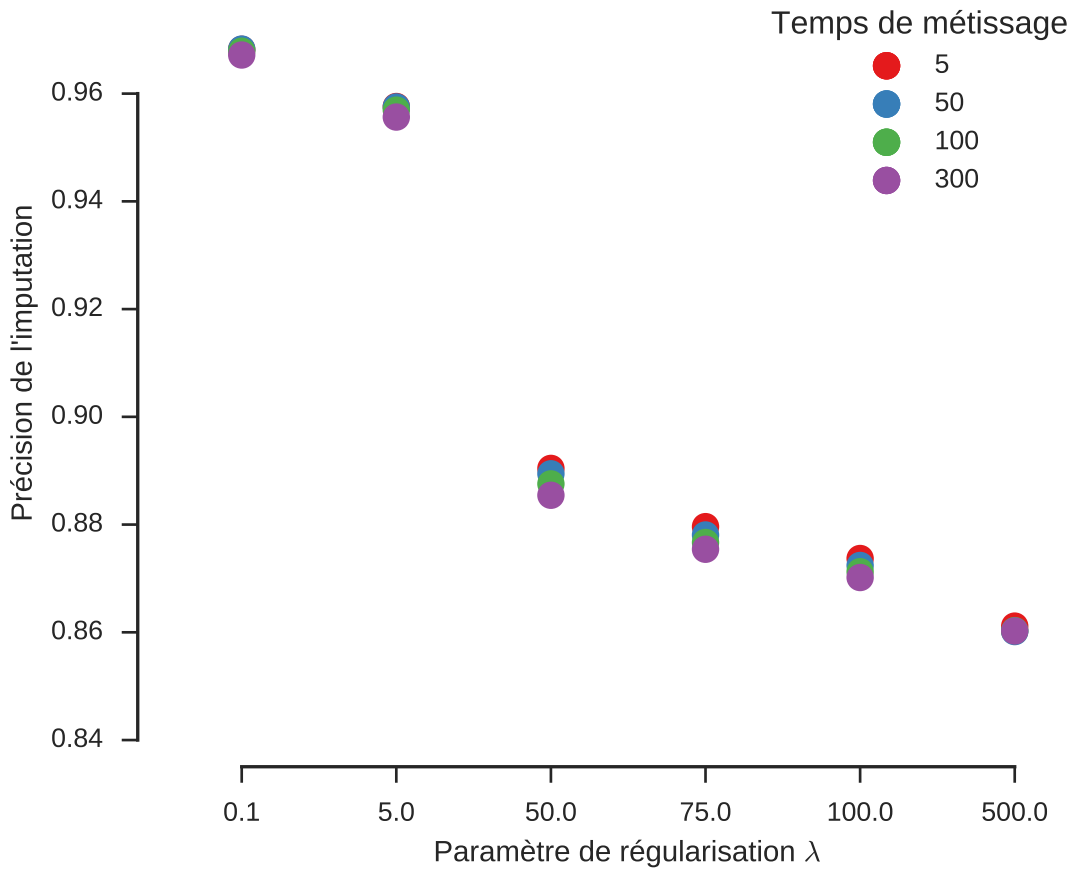

FIGURE 2.22 – Validation croisée pour déterminer une valeur de λ. Pour chaque jeu de données, 50% des valeurs sont masquées et imputées par Loter pour un λ fixé pour 4 temps de métissage différents.

2.4.4 Longueur des blocs

Le principal défaut de la précision diploïde pour comparer les méthodes est que cette mesure ne rend pas compte de la longueur des blocs. En effet, il est possible d'obtenir une précision diploïde très élévée tout en estimant mal la longueur des blocs. Cela est le cas lorsque la méthode se trompe de manière récurrente sur des zones très courtes. L'erreur induite par ces zones est alors négligeable mais les segments d'ascendance sont scindés en deux.

Nous avons donc simulé des métisses de peupliers *balsamifera* et *trichocarpa* et des métisses CEU/YRI pour trois temps de métissage : 10, 200 et 500 générations.

Pour Populus, les métisses sont à 80% *trichocarpa* et à 20% *balsamifera*. Les 500 000 premiers SNPs du chromosome 6 sont gardés pour nos analyses. Pour LAMP-LD, nous avons découpé en 10 les jeux de données puisque LAMP-LD n'accepte pas plus de 50 000 SNPs. Pour HapMap, les métisses sont à 80% YRI et à 20% CEU. Pour ce jeu de données, nous avons gardé les 50 000 premiers SNPs du chromosome 1. Nous étudions la longueur des blocs d'ascendance de la population minoritaire pour ces deux jeux de données. Dans la figure 2.23, nous constatons que pour les métissages récents avec les données d'HapMap, toutes les méthodes retournent des segments de tailles compatibles avec la vérité terrain. La médiane des longueurs des segments en centiMorgan est respectivement de 9.9 cM, 9.8 cM et 12.1 cM pour RFMix, LAMP-LD et Loter, contre 9.0 cM pour la simulation. Il s'agit de la seule des six configurations qui soit défavorable à Loter par rapport aux autres méthodes. Pour RFMix et Loter, ce sont principalement les modules correcteurs de phase qui corrigent la présence en très grand nsombre de segments courts.

La taille de fenêtre de RFMix de 0.2 cM empêche la méthode d'estimer des segments de taille inférieure à 0.2 cM. À l'inverse, LAMP-LD sous-estime la longueur des segments pour Populus à 10 générations depuis le métissage. Le découpage en fenêtre n'est toutefois pas la cause de ces erreurs puisque les fragments erronés ne sont pas localisés aux bords des fenêtres.

Nous avons évalué le temps de calcul des trois méthodes pour l'expérience sur Populus avec 500 000 SNPs (Table 2.3). Pour cette évaluation, 20 CPUs Intel Xeon 2.40 GHz sont utilisés.

Table 2.3 – Temps de calcul pour 500 000 SNPs

LAMP-LD	RFMix	Loter
58 min	28 min	6 min

2.4.5 Coefficients de métissage globaux

Les méthodes d'estimation des coefficients de métissage locaux fournissent l'ascendance à chaque locus. Pour convertir cette information en coefficient de métissage global, nous calculons la proportion d'ascendance pour chaque population de référence sur tous les loci. De cette manière toutes les méthodes d'inférence des coefficients de métissage locaux peuvent fournir une information globale. Cette approche a pour défaut de ne pas tenir compte de l'incertitude sur les coefficients. Pour Loter, nous pourrions notamment pondérer par le ratio de vote après *bagging* et variation de λ. La figure 2.24 représente les coefficients de métissage de deux populations métisses des peupliers *P. Trichocarpa* et *P. Balsamifera*. La population métisse au Sud est plus proche des peupliers *P. Trichocarpa* et celle au Nord est similaire aux peupliers *P. Balsamifera*. Les résultats 2.24 sont donc concordants avec la classification mise en place. Seul un individu (TNZA-4-1) supposé métisse proche de *P. Balsamifera*, est à 99% d'origine *P. Trichocarpa*. Néanmoins, pour la même coordonnée géographique (longitude : -130.47 et latitude : 567.0), des peupliers *P. Trichocarpa* sont présents. Cet individu a donc probablement été mal labellisé.

2.4.6 Métissage de plus de deux populations

Historiquement, les méthodes se sont restreintes au métissage de deux populations. Or l'étude des populations d'Amérique centrale a nécessité d'augmenter le nombre de populations de référence. Nous avons évalué la méthode sur un métissage des populations CHB (chinoise), CEU (européenne) et YRI (africaine). Pour cela, nous avons supposé la présence de deux paramètres des lois exponentielles, $1/T$ et $2/T$, pour la taille des segments d'ascendance. Nous attribuons à CEU le paramètre $1/T$ et aux deux autres populations le paramètre $2/T$. L'objectif est d'autoriser des tailles de segments différentes pour évaluer la capacité des méthodes à inférer des zones de longueur variable. Le paramètre T varie dans l'ensemble $\{5, 100, 200, 500\}$. Pour cette expérience, nous avons utilisé les haplotypes de HapMap et non pas ceux renvoyés par Beagle. Le précision diploïde de Loter sur ces simulations surpasse celle de RFMix et LAMP-LD à partir de $T = 100$ et $T = 200$ respectivement.

2.5 Conclusion

Le module Python "Loter" dédié à l'estimation des coefficients de métissage locaux ouvre la porte à la détection plus précise du métissage ancien. La méthode mise au point repose sur un nombre minimal de paramètres nécessaires. Les expériences effectuées ont toutes été réalisées avec la configuration par défaut où seules les matrices

d'haplotypes sont requises. De plus, l'algorithme est de complexité linéaire et est parallélisé par rapport au nombre d'individus métisses. L'implémentation de Loter est donc optimisée pour traiter de grands jeux de données, humains et non humains, pour lesquels les paramètres biologiques sont souvent inconnus.

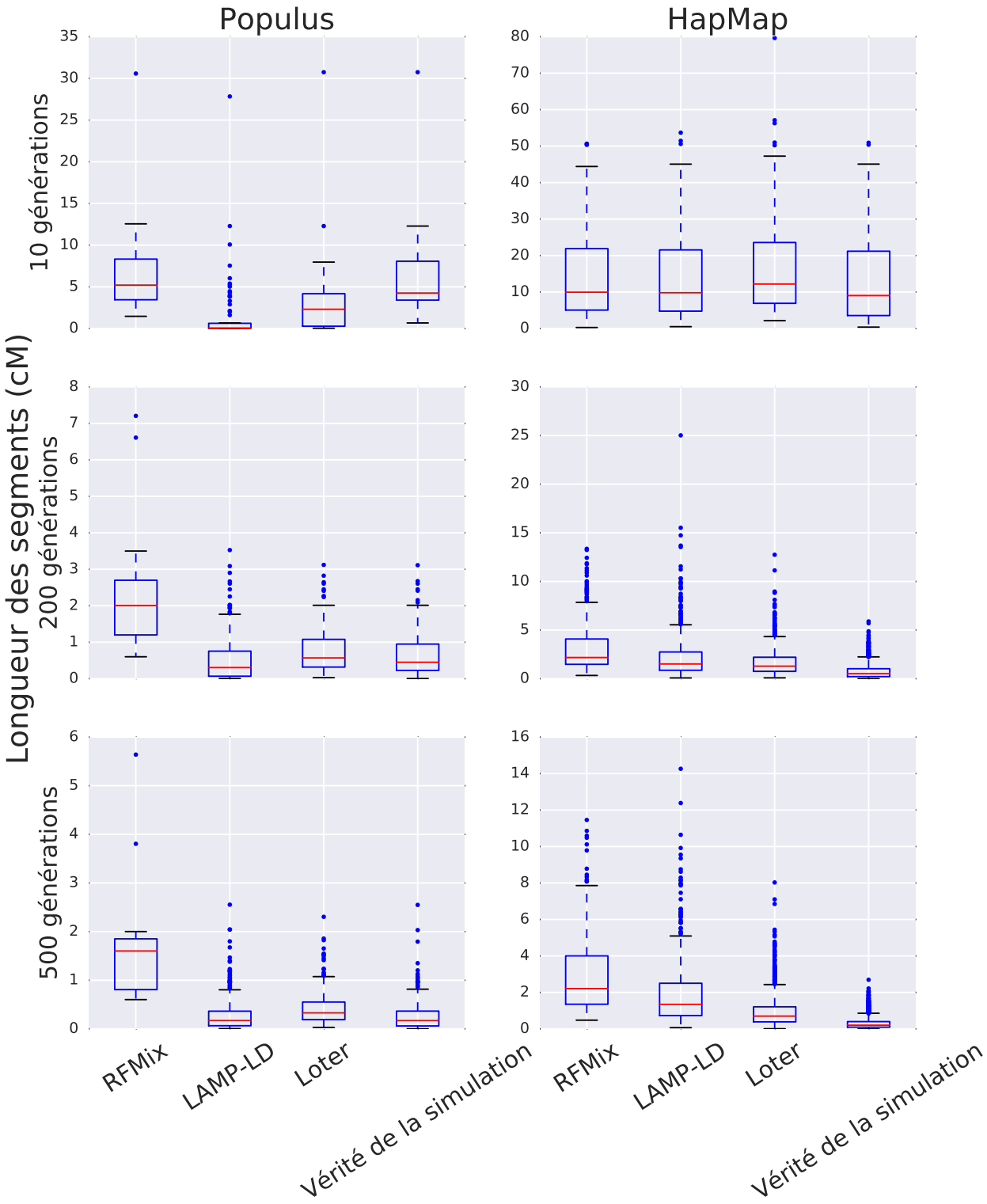

FIGURE 2.23 – Étude de la longueur des segments d'ascendance pour des simulations de métisses CEU/YRI et *trichocarpa*/*balsamifera* pour trois temps de métissage : 10, 200 et 500. Pour Populus, l'expérience est composée de 500000 SNPs pour 20 haplotypes métisses et 60 haplotypes de référence. Pour HapMap, l'expérience est composée de 50000 SNPs pour 48 haplotypes métisses et 80 haplotypes de référence.

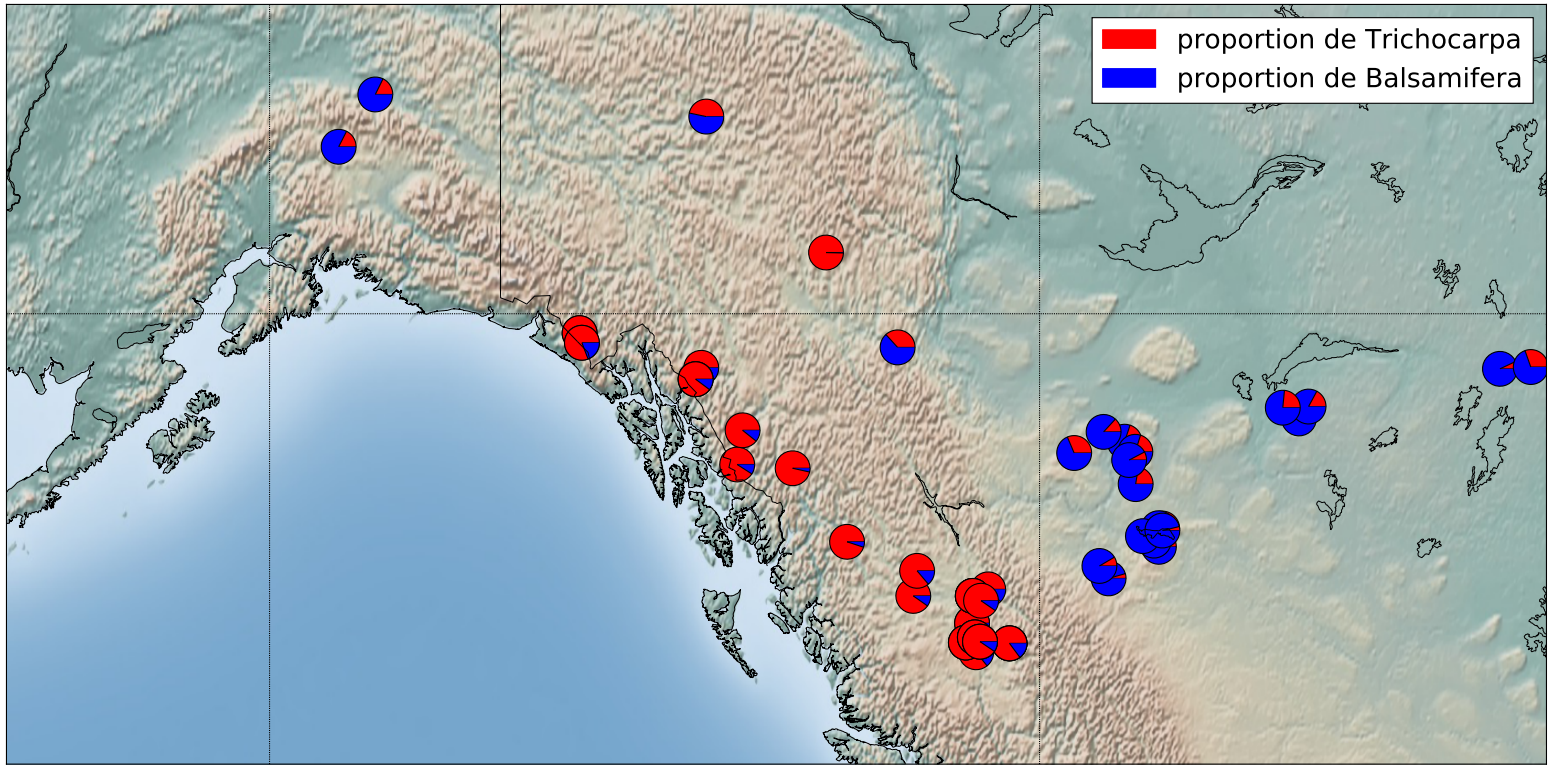

FIGURE 2.24 – Coefficients de métissage pour les deux populations métisses issues de Trichocarpa et Balsamifera. Les coefficients de métissage locaux ont été estimés grâce à Loter. Les coefficients de métissage globaux sont calculés en moyennant les coefficients locaux et affichés sur une carte de l'Amérique du Nord.

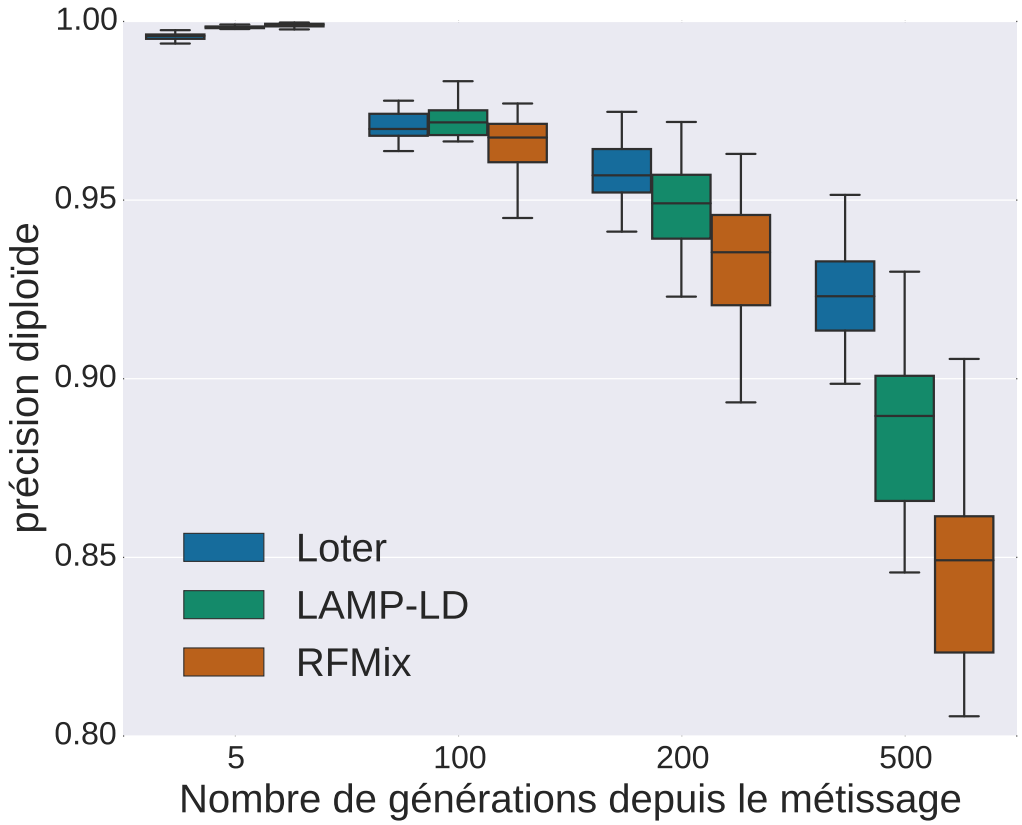

FIGURE 2.25 – Comparaison des méthodes RFMix, LAMP-LD et Loter sur des simulations de métisses CHB, CEU et YRI.

CHAPITRE 3

Phasage des haplotypes et imputation des données

Le phasage populationnel des haplotypes consiste à estimer les haplotypes à partir des génotypes d'une population. Le phasage est un outil important de l'analyse des données génomiques et de leur représentation. Les haplotypes apportent une information supplémentaire à celles des génotypes. En effet, ils contiennent l'information des génotypes ainsi que la phase, c'est-à-dire la répartition des variants sur les chromosomes homologues. Les haplotypes sont les marqueurs hérités ensemble entre les individus (sauf lors de recombinaisons). Dans ce chapitre, nous présentons une méthode pour reconstruire la phase de manière rapide et précise. Similairement au logiciel fastPHASE, la méthode proposée repose sur un modèle de clustering localisé des haplotypes. La phase des haplotypes est estimée en moyennant les solutions d'un problème d'optimisation avec contraintes. L'avantage de cette approche réside dans sa complexité algorithmique linéaire par rapport au nombre de clusters alors que la complexité de fastPHASE croît quadratiquement. Bien que la complexité soit inférieure à fastPHASE, les erreurs de phasage sont similaires à Beagle et fastPHASE pour le jeu de données HapMap.

3.1 État de l'art

Dans cette partie, nous présentons la problématique du phasage et plus particulièrement le problème du phasage populationnel, autrement dit comment reconstruire les haplotypes à partir des génotypes d'une population d'individus. Ensuite, nous décrirons les approches historiques et actuelles qui proposent de résoudre cette problématique.

3.1.1 Problématique du phasage

Le phasage est l'opération reconstruisant les haplotypes sous-jacents d'un génotype. En codant par 0 l'allèle majeur (le plus fréquent dans la population) et par 1 l'allèle mineur (le moins fréquent dans la population), alors les haplotypes sont des vecteurs binaires. La littérature du phasage emploie deux manières d'encoder les génotypes d'individus diploïdes. La première consiste à coder par 2 les loci hétérozygotes (Xing et al., 2006 ; Kalpakis et Namjoshi, 2005). Le tableau ci-dessous indique comment coder une paire de marqueurs codés par 0 et 1.

$\oplus$	0	1
0	0	2
1	2	1

La deuxième consiste à compter le nombre d'allèles majeurs. Nous gardons pour la suite cette notation additive (Scheet et Stephens, 2006).

$+$	0	1
0	0	1
1	1	2

Pour un individu diploïde, de génotype $g \in \{0,1,2\}^m$ où m est le nombre de loci, reconstruire sa phase haplotypique consiste à trouver ses haplotypes réels tels que $h, h' \in \{0,1\}^m$ et $h + h' = g$. Nous nous restreignons donc au cas de marqueurs bialléliques et d'individus diploïdes.

Pour les loci homozygotes de g, tels que $g_j = 0$ ou $g_j = 2$ (pour $j \in \{1,\ldots,m\}$), la phase est triviale puisque $h_j = 0$ et $h'_j = 0$ ou $h_j = 1$ et $h'_j = 1$ sont les seules solutions respectives au locus j. Toute la difficulté de l'estimation du phasage réside dans l'estimation de h et h' aux loci hétérozygotes, tels que $g_j = 1$ [1]. Deux solutions sont possibles, $h_j = 0$ et $h'_j = 1$ ou $h_j = 1$ et $h'_j = 0$.

Notons $\mathcal{I}_g = \{i : g_i = 1\}$, l'ensemble des positions hétérozygotes de g et $m_h(g) = \mathbf{card}(\mathcal{I}_g)$, le nombre de sites hétérozygotes. Alors pour un génotype g doté d'au moins un site hétérozygote, il existe $2^{m_h(g)-1}$ paires d'haplotypes qui reconstuisent g. Nous appellerons diplotype une paire d'haplotypes représentant le génotype d'un individu diploïde. Que ce soit pour $m_h(g) = 0$ (aucun site hétérozygote, cas trivial exposé précédement) ou $m_h(g) = 1$ (un site hétérozygote), la paire d'haplotypes est unique. Le nombre d'haplotypes augmente exponentiellement avec le nombre de positions hétérozygotes.

1. Ces positions sont donc aussi appelées positions “ambigües”.

Phasage expérimental et à partir de trios

La phase des haplotypes est une information physiquement présente sur les chromosomes. Il est donc légitime de comprendre en premier lieu les méthodes expérimentales et leurs limitations avant d'aborder les méthodes statistiques et algorithmiques.

Par exemple, les haplotypes sont connus pour les chromosomes non homologues, comme les chromosomes sexuels X/Y. Néanmoins, cela ne permet pas d'accéder à l'information haplotypique des autres chromosomes. Les chromosomes sexuels sont donc notamment utilisés pour constituer des vérités terrains et comparer les méthodes (Delaneau, Marchini et Zagury, 2011). Une autre méthode consiste à séquencer des trios (Roach et al., 2011), c'est-à-dire deux parents et un enfant.

21022	génotype maternel
11020	génotype paternel
21021	génotype de l'enfant

Alors en supposant qu'aucune recombinaison n'a eu lieu, les haplotypes de l'enfant sont :

	1•011	copie maternelle
+	1•010	copie paternelle
	21021	génotype de l'enfant

Il reste une ambiguïté (notée •) pour le cas de figure où les trois individus sont hétérozygotes à une position donnée. D'autres configurations familiales peuvent aussi réduire l'incertitude de la phase (Roach et al., 2011).

L'estimation de la phase peut aussi se faire directement à partir des informations de séquençage. Les techniques de séquençage des génomes procèdent en lisant des fragments de nucléotides [2]. Si un fragment comprend plus de deux sites hétérozygotes du génome d'un individu alors la phase pour chacune de ces positions est connue. En effet, la lecture du séquenceur contient directement les haplotypes du fragment. La longueur des fragments dépend des technologies. Les techniques Sanger peuvent créer des *reads* de plus de 700 paires de base alors que les *reads* des séquenceurs dernières générations sont typiquement plus courts (Hert, Fredlake et Barron, 2008). Le séquençage est un domaine toujours en plein essor, sur les aspects du coût, de la vitesse et de la quantité de données. Il est tout à fait envisageable que naissent des techniques de séquençage compétitives où la phase des haplotypes sera connue.

Dans la plupart des cas de figure, les trios et autres méthodes expérimentales sont indisponibles pour de multiples raisons. La bio-informatique a alors recours à des méthodes algorithmiques et statistiques pour reconstruire les haplotypes à partir d'une

2. Plus souvent appelés *reads*.

population ou d'une famille d'individus. Nous présentons maintenant un aperçu de ces méthodes et de leur histoire.

3.1.2 Historique des méthodes

Puisque $2^{m_h(g)-1}$ paires d'haplotypes reconstuisent g, il n'est pas difficile d'exhiber des paires d'haplotypes compatibles avec les génotypes. Il est nécessaire de fixer des critères sur la population d'haplotypes reconstruite à partir des n génotypes. Nous notons $G \in \{0,1,2\}^{n\times m}$ la matrice de ces n génotypes et $H \in \{0,1\}^{2n\times m}$ la matrice des $2n$ haplotypes correspondants. Nous dirons que $g \in G$ si, et seulement s'il existe une ligne de G égale à g (de même pour h et H).

La règle de Clark

La règle de Clark est la première méthode historique de phasage des haplotypes (Clark, 1990). Cette méthode gloutonne procède par soustraction. On reconstruit d'abord les haplotypes pour lesquels aucune ambiguïté n'est présente (avec au plus 1 site hétérozygote). Ces haplotypes sont ensuite utilisés pour reconstruire la phase des autres génotypes.

En effet, soit h un des haplotypes déjà reconstruit et g un génotype non phasé. Nous noterons $\mathcal{H} = \{0,1\}^m$, l'ensemble des haplotypes, et noterons $g \to h$ le fait que h est compatible avec g, c'est-à-dire que $g - h \in \mathcal{H}$. Si $h' = g - h \in \mathcal{H}$ alors g est phasé et h' est ajouté à l'ensemble des haplotypes reconstruits. La méthode est itérée jusqu'à la résolution complète du problème ou l'impossibilité d'estimer de nouveaux haplotypes. Un des défauts de cette méthode est qu'elle dépend de l'ordre de résolution et qu'elle n'est pas adaptée aux grands jeux de données dans lesquels les individus ont de nombreux sites hétérozygotes, auquel cas aucune initialisation n'est envisageable.

Principe du maximum de parcimonie

Introduit par Hubbel (2000) pour le phasage, le principe du maximum de parcimonie repose sur l'hypothèse qu'un nombre minimal d'haplotypes distincts reconstruisent les génotypes d'une population naturelle. Cette hypothèse est fondée sur l'analyse des haplotypes de larges populations (The International HapMap Consortium, 2005). Hubbel (2000) a démontré que ce problème est NP-difficile[3]. Ce principe a été utilisé par les nombreuses techniques nommées **HPP** *Haplotype phasing problem with Pure Parsimony*. Ces techniques regroupent des méthodes d'optimisation linéaire en nombres entiers (Bertolazzi et al., 2008 ; Brown et Harrower, 2004), des

3. Problème au moins aussi difficile que tous les problèmes de décision pour lesquels les solutions peuvent êtres vérifiées en temps polynomial.

méthodes de séparation et d'évaluation *branch and bound* (Wang et Xu, 2003) et des heuristiques (Wang et Yang, 2011).

Le modèle de parcimonie pure de Hubbel (**TIP**) se décrit ainsi :

Soit $\mathcal{H}_G \subseteq \mathcal{H}$ l'ensemble des haplotypes reconstruisant G

$$\mathcal{H}_G = \{h : \exists g \in G, g \to h \in \mathcal{H}\}.$$

À chaque élément de cet ensemble, nous associons une variable indicatrice x_h.

Soit $\mathcal{H}_G^2 = \{(h, h') : h \leq h', h, h' \in \mathcal{H}_G\}$, l'ensemble de toutes les paires ordonnées d'une quelconque manière d'haplotypes de $\mathcal{H}_G$. $y_{h,h'}$ indique la présence ou non de h et h' dans $\mathcal{H}_G$.

La première contrainte est de ne sélectionner qu'une seule paire d'haplotypes valides. Seulement un $y_{h,h'}$ est égal à 1 tandis que les autres sont nuls.

$$\sum_{(h,h') \in \mathcal{H}_g^2} y_{h,h'} = 1, \forall g \in G. \tag{3.1}$$

De plus, $y_{h,h'}$, x_h et x'_h sont des variables indicatrices des haplotypes et doivent être synchronisées. Si $y_{h,h'} = 1$ alors on a $x_h = 1$ et $x'_h = 1$. Cela permet de rajouter des contraintes pour chaque paire d'haplotypes,

$$\begin{aligned} \forall (h, h') \in \mathcal{H}_G^2, \quad & y_{h,h'} - x_h \leq 0 \\ & y_{h,h'} - x_{h'} \leq 0. \end{aligned} \tag{3.2}$$

Puisque les variables x_h et $y_{h,h'}$ sont binaires, nous avons

$$\begin{aligned} \forall h \in \mathcal{H}_G, \quad & x_h \in \{0, 1\} \\ \forall (h, h') \in \mathcal{H}_G^2 \quad & y_{h,h'} \in \{0, 1\}. \end{aligned} \tag{3.3}$$

Les contraintes des équations 3.1, 3.2 et 3.3 définissent l'ensemble de recherche. Nous savons que l'objectif final est de minimiser le nombre d'haplotypes pour reconstruire G, il faut donc minmiser

$$\text{minimiser} \quad \sum_{h \in \mathcal{H}_G} x_h.$$

Néanmoins ce type d'approche est très coûteux en temps et en mémoire. Une reformulation du problème, appelée k-HPP, a été proposée par Kalpakis et Namjoshi (2005) sous la forme d'un produit de matrices. Soit $S \in \{0, 1, 2\}^{n \times k}$ une matrice de sélection telle que chaque ligne est composée d'entiers positifs dont la somme vaut deux. Cela signifie qu'un génotype est la sélection de deux haplotypes parfois identiques. Alors nous cherchons S et H tels que

$$G = S \cdot H.$$

De plus, S doit avoir un nombre maximal de colonnes à zéro, c'est-à-dire d'haplotypes jamais sélectionnés. Le problème HPP est équivalent au problème $2n$-HPP et une solution approchée est obtenue par optimisation semi-définie positive. Néanmoins ces approches ne modélisent pas les recombinaisons et sont donc limitées à des jeux de données de petites tailles.

Approche EM

Les modèles multinomiales, souvent appelés modèles EM *Expectation Maximization*, supposent que tous les haplotypes sont équiprobables (Excoffier et Slatkin, 1995).

Sous l'hypothèse d'une reproduction aléatoire, la probabilité d'observer le génotype g_i est donnée par la somme des probabilités de chaque paire d'haplotypes reconstruisant ce génotype,

$$P_i = \Pr(g_i) = \sum_{(h,h')\in\mathcal{H}^2_{g_i}} \Pr\{(h,h')\}.$$

Les probabilités d'observer les paires sont définies par les fréquences haplotypiques notées p_h. Sous l'hypothèse d'un équilibre de Hardy-Weinberg, la probabilité des diplotypes s'écrit

$$\Pr\{(h,h')\} = \begin{cases} p_h^2 & \text{si } h = h' \\ 2p_h p_{h'} & \text{sinon.} \end{cases}$$

Soit l le nombre de génotypes distincts et n_i les comptes des génotypes g_i tels que $\sum_{i=1}^{l} n_i = n$. Ainsi, la probabilité d'observer G conditionnellement aux n fréquences P_i suit une loi multinomiale.

$$\Pr(G|P_1,\ldots,P_n) = \frac{n!}{n_1!\ldots n_l!}\prod_{i=1}^{n} P_i$$

Le but final est de maximiser la vraisemblance des paramètres

$$\mathcal{L}(p_{h_1},\ldots,p_{h_{|\mathcal{H}_G|}}) = \alpha\prod_{i=1}^{n}\sum_{(h,h')\in\mathcal{H}^2_{g_i}} \Pr\{(h,h')\}.$$

L'algorithme EM optimise la vraisemblance de manière alternée. L'étape d'Espérance calcule la probabilité des diplotypes. Puis l'étape de Maximisation met à jour les fréquences des haplotypes. Cette méthode est itérée jusqu'à convergence de l'algorithme. Ces approches ne passent pas à l'échelle de jeux de données de taille moyenne. Des méthodes de partitionnement (Partition Ligation EM) ont été introduite pour accélérer la méthode mais s'avèrent moins précises (Qin, Niu et Liu, 2002).

Modèles génétiques

Un phasage plus précis peut être obtenu un ajoutant a priori sur la distribution des haplotypes $\Pr(H|G)$. Le logiciel PHASE (Stephens, Smith et Donnelly, 2001) estime les haplotypes grâce à un échantillonnage de Gibbs et en modélisant $\pi(h|H)$, la probabilité conditionnelle d'observer un nouvel haplotype h sachant un ensemble H d'haplotypes, avec une approximation du coalescent (voir 1.4). HAPLOTYPER (Halperin et Eskin, 2004) propose une méthode bloc par bloc reposant sur une modélisation différente du prior.

Le logiciel PHASE était historiquement considéré comme une référence dans le phasage et a été notamment utilisé pour construire les vérités terrains des premières versions de HapMap. Il fut ensuite remplacé par Impute++ (Howie, Donnelly et Marchini, 2009).

Ce type d'approche est encore une fois très lent et adapté seulement aux petits jeux de données. Des modèles pour regrouper localement les haplotypes ont été élaborés afin de détecter des motifs complexes de LD. Ces approches reposent notamment sur des modèles de Markov cachés (Scheet et Stephens, 2006 ; Browning et Browning, 2007) pour modéliser la probabilité d'observer un certain haplotype sachant les autres haplotypes et le génotype, notée $\Pr(H_i|H_{-i}, G)$. Cela nous ramène au problème du CSD *Conditional Sampling Distribution* pour lequel Li et Stephens (voir 1.5) ont proposé une modélisation efficace. Néanmoins, l'approche standard implique un HMM avec $m \times (2n)^2$ états. Une des différences principales entre les méthodes modernes comme fastPHASE (Scheet et Stephens, 2006) , Beagle (Browning et Browning, 2011) ou SHAPEIT (Delaneau, Marchini et Zagury, 2011) repose donc sur leur manière de représenter H_{-i} pour compresser l'information et diminuer la complexité algorithmique. La force supplémentaire de toutes ces méthodes est la prise compte naturelle des recombinaisons grâce au HMM.

Le principal défi de l'inférence des haplotypes repose sur la capacité des méthodes à pouvoir traiter des données massives. Nous expliquons maintenant la méthode que nous avons mis au point pour répondre à ce besoin. Le concept de notre méthode découle du modèle d'haplotypes regroupés en clusters locaux de fastPHASE (Scheet et Stephens, 2006). Cependant la complexité algorithmique de notre méthode croît linéairement par rapport au nombre de clusters alors que celle de fastPHASE augmente quadratiquement.

3.2 Matériel et méthodes

Nous proposons dans cette partie un nouveau formalisme pour reconstruire la phase de jeux de données de SNPs avec une complexité linéaire.

3.2.1 Problème d'optimisation

Tout d'abord, nous rappelons et introduisons les notations nécessaires pour notre modèle. Soit $G \in \{0,1,2\}^{n\times m}$ la matrice des n génotypes avec m marqueurs. L'objectif du phasage est de construire $H \in \{0,1\}^{2n\times m}$, la matrice des haplotypes.

$M_{i,j}$ (ou M^i_j) désigne le j-ième élément de la i-ième ligne.

M_i désigne la i-ième ligne.

H est une matrice solution du problème si et seulement si

$$\forall i \in \{1,\dots,n\}, G_i = H_{2i} + H_{2i+1}.$$

(H_{2i}, H_{2i+1}) est un diplotype de l'individu i.

Soit $A \in [0,1]^{K\times m}$ la matrice représentant K clusters qui permettent de reconstruire les $2n$ haplotypes. Notons $S \in \{1,\dots,k\}^{2n\times m}$, la matrice indiquant de quel élément de A les haplotypes de H proviennent.

Nous notons $\mathcal{S}_{k,j} = \{i : S_{i,j} = k\}$, l'ensemble des indices des haplotypes qui ont pioché dans le cluster k au locus j. Notre objectif est de reconstituer chaque haplotype comme une segmentation des clusters. Chaque haplotype peut donc changer de cluster de provenance au fil des loci. Nous formulons le problème d'optimisation suivant :

$$\begin{aligned}
\underset{S,A,H}{\text{minimiser}} \quad & f(A,H,S) = \sum_{j=1}^{m}\sum_{k=1}^{K}\sum_{i\in\mathcal{S}_{k,j}} (H_{i,j} - A_{k,j})^2 + \lambda \sum_{j=1}^{m-1}\sum_{i=1}^{2n} 1_{S_{i,j}\neq S_{i,j+1}} \\
\text{avec} \quad & a_{i,j} \in [0,1] \\
& H_{i,j} \in \{0,1\} \\
& S_{i,j} \in \{1,2,\dots,k\} \\
& H_{2i} + H_{2i+1} = G_i, \forall i \in \{1,\dots,n\}
\end{aligned} \tag{3.4}$$

Le premier terme, l'attache aux données, impose aux haplotypes de ressembler au cluster duquel ils proviennent. Le second terme, la pénalité, régularise la solution en empêchant l'haplotype de changer de provenance trop fréquemment.

Pour résoudre ce problème d'optimisation, nous procédons par optimisation alternée. Nous optimisons successivement sur A, H et S. Dans le cadre de l'optimisation alternée, le problème est séparé pour chaque haplotype et notamment entre les deux haplotypes d'un même individu. C'est une des principales raisons pour laquelle notre formulation

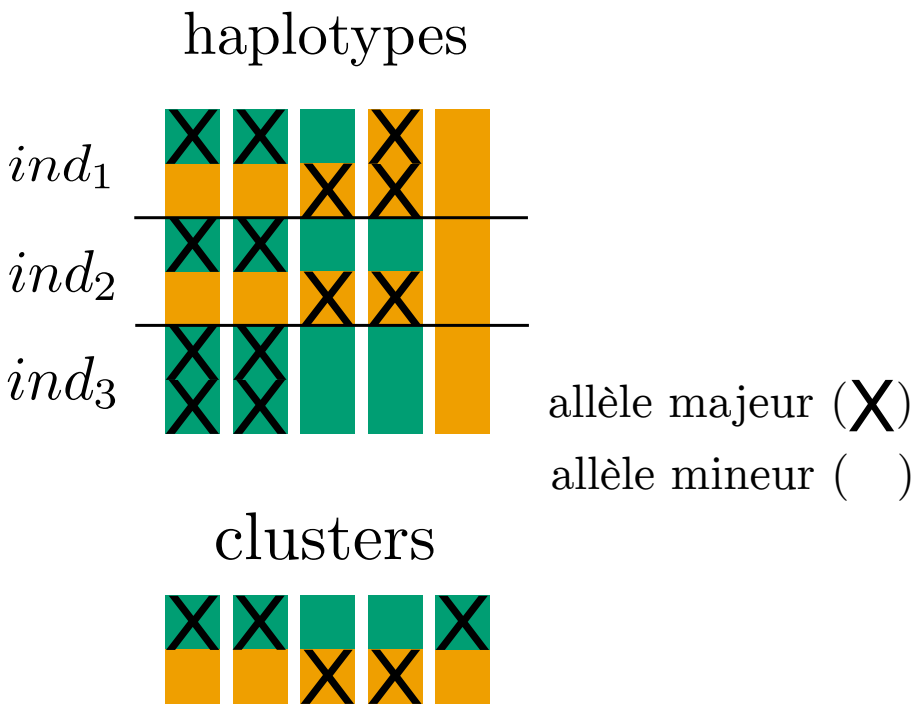

FIGURE 3.1 – Illustration du modèle de phasage. Les clusters représentent la matrice A. Chaque colonne représente un SNP et la présence ou non de l'allèle est signifiée par une croix. Trois individus (donc 6 haplotypes) sont phasés, ce qui correspond à la matrice H. La couleur pour chaque SNP et chaque haplotype désigne le cluster de provenance, c'est-à-dire S.

a une complexité algorithmique linéaire. Nous avons toutefois aussi développé une version de complexité quadratique qui optimise de manière alternée sur A et sur (S, H) de manière jointe.

Optimisation de A

En fixant H et S, le problème d'optimisation devient (en omettant les contraintes)

$$\underset{A}{\text{minimiser}} \quad f_{H,S}(A) = \sum_{j=1}^{m} \sum_{k=1}^{K} \sum_{i \in \mathcal{S}_{k,j}} (H_{i,j} - A_{k,j})^2. \tag{3.5}$$

Chaque coefficient $A_{k,j}$ dépend de variables $H_{i,j}$ spécifiques. Le coefficient est donc minimisé indépendamment

$$\underset{A_{k,j}}{\text{minimiser}} \quad f^{k,j}_{H,S}(A_{k,j}) = \sum_{i \in \mathcal{S}_{k,j}} (H_{i,j} - A_{k,j})^2. \tag{3.6}$$

La valeur optimale de $A_{k,j}$ est la moyenne des valeurs des haplotypes qui ont

sélectionné le cluster k

$$A_{k,j} = \sum_{i \in \mathcal{S}_{k,j}} \frac{H_{i,j}}{|\mathcal{S}_{k,j}|}. \tag{3.7}$$

Nous pondérons cette moyenne par w pour les $H_{i,j}$ reconstruisant un génotype homozygote. L'information obtenue aux sites homozygotes est en effet plus sûre et donc plus importante.

Optimisation de H

Pour estimer H, c'est-à-dire les haplotypes, nous fixons les valeurs de A et S. Nous savons donc de quel cluster les haplotypes proviennent.

$$\begin{aligned} \underset{H}{\text{minimiser}} \quad & f_{A,S}(A) = \sum_{j=1}^{m} \sum_{k=1}^{K} \sum_{i \in \mathcal{S}_{k,j}} (H_{i,j} - A_{k,j})^2 \\ \text{avec} \quad & H_{i,j} \in \{0,1\} \\ & H_{2i} + H_{2i+1} = G_i, \forall i \in \{1,\dots,n\} \end{aligned} \tag{3.8}$$

Chaque coefficient $H_{2i,j}$ et $H_{2i+1,j}$ n'intervient que dans un seul terme de cette somme. Soit i, un individu particulier, alors l'optimisation du coefficient est réalisée par

$$\begin{aligned} \underset{H_{2i,j},H_{2i+1,j}}{\text{minimiser}} \quad & f^{i,j}_{A,S}(H_{2i,j}) = (H_{2i,j} - A_{S_{2i,j},j})^2 + (H_{2i+1,j} - A_{S_{2i+1,j},j})^2 \\ \text{avec} \quad & H_{2i,j} \in \{0,1\} \\ & H_{2i+1,j} \in \{0,1\} \\ & H_{2i,j} + H_{2i+1,j} = G_{i,j}. \end{aligned} \tag{3.9}$$

La valeurs optimale de $H_{i,j}$ pour des SNPs hétérozygotes est obtenue en comparant les valeurs de $A_{S_{2i,j},j}$ et $A_{S_{2i+1,j},j}$.

$$\begin{aligned} H_{2i,j} &= \begin{cases} 1 & \text{si } A_{S_{2i,j},j} \geq A_{S_{2i+1,j},j} \\ 0 & \text{sinon} \end{cases} \\ H_{2i+1,j} &= 1 - H_{2i,j}. \end{aligned}$$

Néanmoins, en suivant cette procédure, l'optimisation tombe systématiquement plus rapidement dans des minima locaux. C'est pourquoi, dans l'approche implémentée, nous tirons aléatoirement la valeur de $H_{2i,j}$ avec la probabilité suivante

$$\begin{aligned} \Pr(H_{2i,j} = 1) &= \frac{A_{S_{2i,j},j}(1 - A_{S_{2i+1,j},j})}{A_{S_{2i,j},j}(1 - A_{S_{2i+1,j},j}) + (1 - A_{S_{2i,j},j})A_{S_{2i+1,j},j}} \\ H_{2i+1,j} &= 1 - H_{2i,j} \end{aligned}$$

Cette approche stochastique de la mise à jour de H corrige notamment les mauvaises initialisations aléatoires et modélise l'incertitude lorsque $A_{S_{2i,j},j}$ et $A_{S_{2i+1,j},j}$ ne sont ni 0 ni 1. Par exemple, si $A_{S_{2i,j},j} = 0.51$ et $A_{S_{2i+1,j},j} = 0.5$ alors la version déterministe accordera toujours l'allèle mineur au même haplotype contrairement à la mise à jour stochastique.

Optimisation de S

Nous pouvons séparer l'optimisation de S pour chaque $i \in \{1, \dots, 2n\}$ et en notant $s = S_i$, le problème d'optimisation de s est

$$\begin{aligned} \underset{s}{\text{minimiser}} \quad & f^i_{A,H} = \sum_{j=1}^{m} (H_{i,j} - A_{s_j,j})^2 + \lambda \sum_{j=1}^{m-1} 1_{s_j \neq s_{j+1}} \\ \text{avec} \quad & s_j \in \{1, 2, \dots, k\} \end{aligned} \tag{3.10}$$

Le s optimal est le plus court chemin dans un graphe représentant tous les s possibles et le coût associé (Figure 3.2). Le principe de cette résolution est le même que pour la méthode 2.3 d'estimation des coefficients de métissage locaux.

$$\begin{aligned} \text{coût}_{(k,m)} = \quad & \min \left(\text{coût}_{(k,m-1)}, \lambda + \min_{k' \in \{1,\dots,K\}} \{\text{coût}_{(k',m-1)}\} \right) \\ & + (H_{i,m} - A_{k,m})^2 \end{aligned} \tag{3.11}$$

Cette formulation récursive est résolue par programmation dynamique. Le coût algorithmique de cette étape est donc en $\mathcal{O}(K \times m \times n)$, notamment parce que le graphe ne contient que deux transitions possibles : 0 et λ.

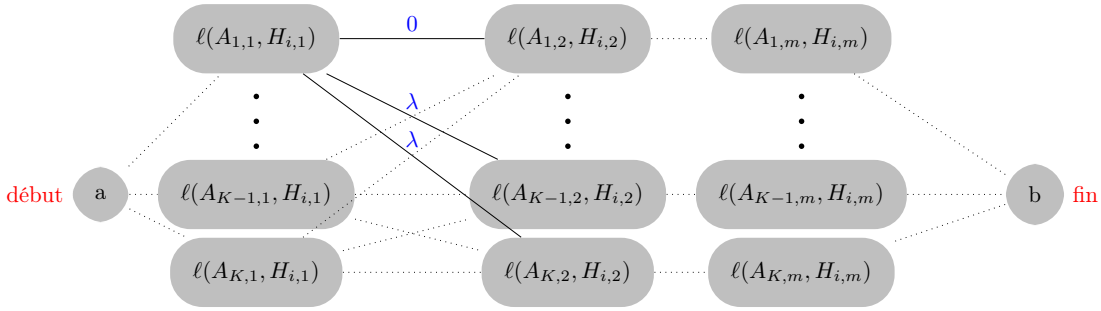

FIGURE 3.2 – Graphe construit pour optimiser S. La fonction de perte ℓ ici choisie est $\ell(x, y) = (x - y)^2$. Les noeuds contiennent le terme d'attache aux données et les arrêtes sont pondérées par la pénalité. Un chemin dans le graphe coûte au total la somme des termes dans les noeuds et sur les arrêtes. Ce graphe peut être adapté pour d'autres fonctions de perte ℓ.

L'algorithme alterne donc entre l'optimisation sur S, A et H. Au départ de l'algorithme nous optimisons S, il faut donc fixer des valeurs pour A et H.

Initialisation

L'initialisation de H est aléatoire et uniforme pour chaque locus hétérozygote. Pour $A_{k,j}$, nous tirons un individu aléatoirement. Si l'individu est hétérozygote au locus j alors nous tirons $A_{k,j}$ uniformément entre 0 et 1, sinon nous initialisons $A_{k,j} = G_{i,j}/2$ puisque $G_{i,j}$ vaut soit 0, soit 2 et que l'haplotype est connu pour ce locus.

Paramètres de la méthode

Notre méthode contient deux paramètres : K et λ.

K désigne le nombre de clusters de l'algorithme. La valeur de K est donc logiquement un entier inférieur au nombre d'haplotypes à reconstruire ($K \leq 2n$). Lorsque $K = 2n$, alors la solution optimale est que pour chaque individu i

$$\begin{aligned}
&\text{Soit } H, H' \in \mathcal{H}_{G_i} \\
&A_{2i} = H \\
&A_{2i+1} = H' \\
&S_{2i,j} = 2i, \forall j \in \{1, \ldots, m\} \\
&S_{2i+1,j} = 2i + 1, \forall j \in \{1, \ldots, m\}.
\end{aligned}$$

En effet, $f(A, H, S)$ vaut zéro puisque le terme d'attache aux données est nul et que les haplotypes piochent constamment dans le même cluster.

Lorsque $K = 1$, A est de dimension $1 \times m$ et pour chaque $A_{1,j}$ nous avons

$$A_{1,j} = \frac{\sum_{i=1}^{n} G_{i,j}}{2n}.$$

$A_{1,j}$ contient la fréquence allélique au locus j. Ces deux cas extrêmes du choix de K montrent l'importance de la valeur de K. Néanmoins, ce paramètre a l'avantage d'être une valeur entière entre 1 et $2n$. Dans le cas de fastPHASE, le paramètre est choisi dans l'ensemble $\{5, 10, 15\}$. La figure 3.3 illustre l'impact du choix du paramètre K pour un jeu de données simulé comme un mélange de 5 haplotypes. Avec un K trop faible, des haplotypes ne sont pas du tout représentés sur certaines régions. Avec un K trop élevé, nous surapprenons, c'est-à dire que chaque cluster représentera un seul haplotype et que les haplotypes reconstruits seront alors quelconques.

Le paramètre λ est la pénalité de notre modèle. Nos expérimentations ont montré qu'un λ dans la même gamme que pour l'estimation des coefficients de métissage locaux pouvait être choisi. Par défaut λ est fixé à 2. Les deux paramètres de notre modèles peuvent être estimés en masquant des données et en évaluant l'imputation du modèle.

Théorême 2.1: K-moyennes

Lorsque $\lambda \to \infty$ cette méthode correspond à l'algorithme des K-moyennes pour H fixé. Avec $\|x\|_2^2 = \sum_i x_i^2$, la norme euclidienne.

$$\begin{aligned} \underset{\mathcal{S}}{\text{minimiser}} \quad & f(\mathcal{S}) = \sum_{k=1}^{K} \sum_{i \in \mathcal{S}_k} \|H_i - A_k\|_2^2 \\ \text{avec} \quad & \mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_K\} \\ & \text{une partition des } 2n \text{ haplotypes en } K \text{ clusters.} \end{aligned} \tag{3.12}$$

Démonstration

Soit $\lambda > 2nm$, alors la solution est telle que $\forall i \in \{1, \ldots, 2n\}, \forall j \in \{1, \ldots, m-1\}, S_{i,j} = S_{i,j+1}$ (voir démonstration plus proche voisin). Pour H fixé,

$$\begin{aligned} \underset{S,A}{\text{minimiser}} \quad & f(A, S) = \sum_{j=1}^{m} \sum_{k=1}^{K} \sum_{i \in \mathcal{S}_{k,j}} (H_{i,j} - A_{k,j})^2 \\ \text{avec} \quad & a_{i,j} \in [0, 1] \\ & S_{i,j} \in \{1, 2, \ldots, k\}. \end{aligned} \tag{3.13}$$

Or $\mathcal{S}_{k,j} = \mathcal{S}_{k,i}, \forall i, j \in \{1, \ldots, m\}$. Nous posons donc $\mathcal{S}_k = \mathcal{S}_{k,0}$. En intervertissant les signes sommes et en utilisant la norme euclidienne

$$\begin{aligned} f(A, S) &= \sum_{j=1}^{m} \sum_{k=1}^{K} \sum_{i \in \mathcal{S}_{k,j}} (H_{i,j} - A_{k,j})^2 \\ &= \sum_{k=1}^{K} \sum_{i \in \mathcal{S}_k} \sum_{j=1}^{m} (H_{i,j} - A_{k,j})^2 \\ &= \sum_{k=1}^{K} \sum_{i \in \mathcal{S}_k} \|H_{i,j} - A_{k,j}\|_2^2. \end{aligned} \tag{3.14}$$

Nous retrouvons ainsi l'algorithme des K-moyennes. □

3.2.2 Erreurs de phasage et consensus

Une fois que la matrice H est estimée, nous comparons la qualité du phasage en calculant le nombre de permutations nécessaire pour retrouver la vérité terrain.

Nous définissons maintenant la mesure d'erreur de phasage entre deux paires haplotypes $H = \{h^1, h^2\}$ et $H = \{h'^1, h'^2\}$ telles que $h^1 + h^2 = h'^1 + h'^2 = g$ où g est le génotype connu.

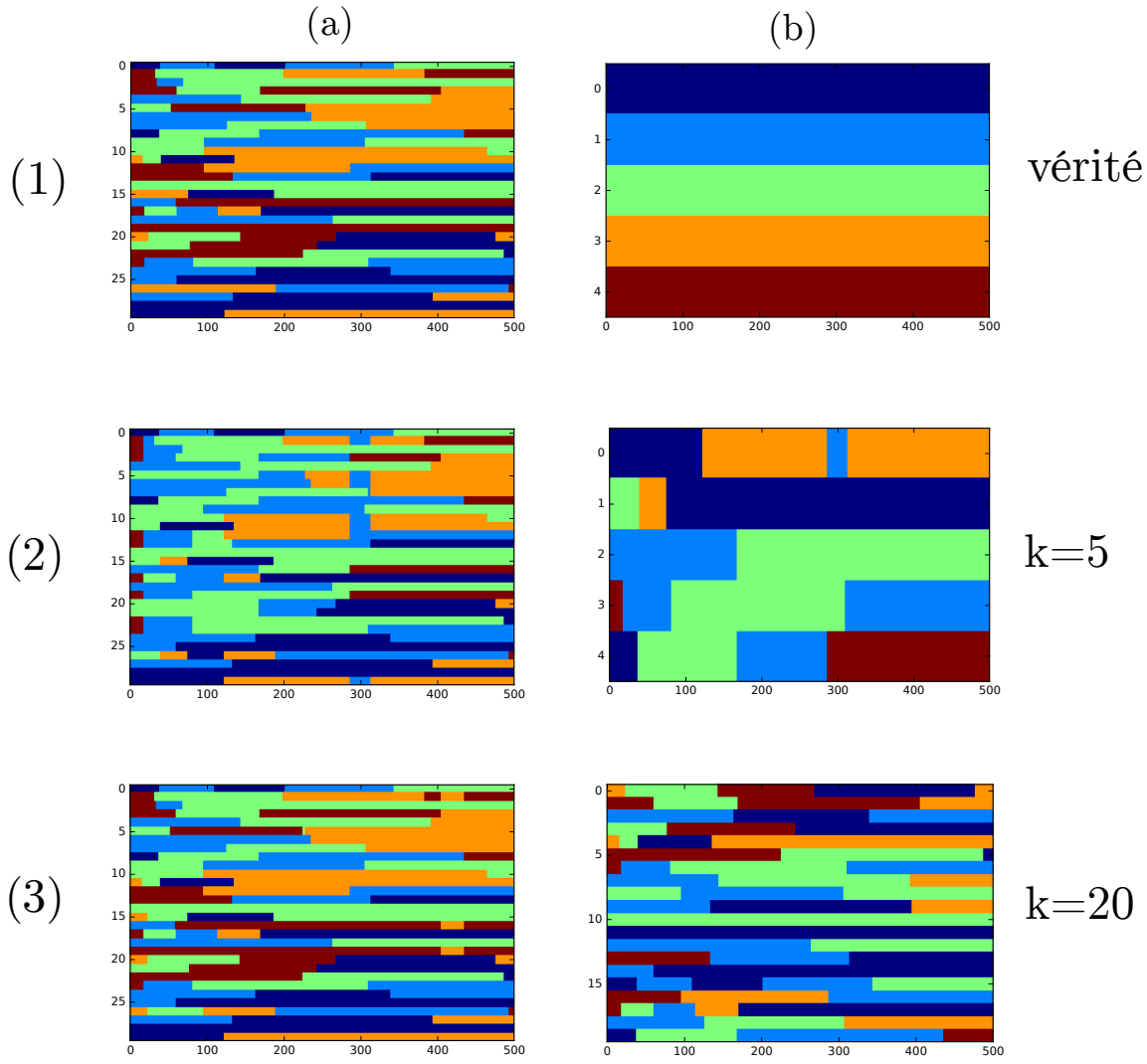

FIGURE 3.3 – Impact de paramètre K sur l'estimation de S et de A. (1) représente la vérité terrain avec (a) le jeu de données et (b) les clusters. (2) et (3) montrent les haplotypes reconstruits et les clusters estimés.

Tout d'abord, nous définissons la distance de Hamming qui compte le nombre de différences entre deux vecteurs.

Distance de Hamming

$$d_H(s,t) = |\{i \in \{1,...,m\} : s_i \neq t_i\}|$$

La séquence de permutations d'une paire d'haplotypes est la séquence indiquant si une permutation est présente entre deux positions hétérozygotes, c'est-à-dire si l'allèle mineur n'est plus sur le même haplotype.

Séquence de permutations

$$\begin{aligned} &h^1, h^2 \in \{0,1\}^m \\ &I = \{i \mid h^1_i \neq h^2_i\}, \text{ avec } i_1 < i_2 < ... < i_{m'} \\ &s(h^1,h^2) = s(h^2,h^1) = s \in \{0,1\}^{m'-1} \\ &s_j = \begin{cases} 0, \ h^1_{i_j} = h^1_{i_{j+1}} \text{ et } h^2_{i_j} = h^2_{i_{j+1}} \\ 1, \ h^1_{i_j} \neq h^1_{i_{j+1}} \text{ et } h^2_{i_j} \neq h^2_{i_{j+1}} \end{cases} \end{aligned} \tag{3.15}$$

La distance de permutation se définit donc comme la distance de Hamming des séquences de permutations

Distance de permutation

$$d_s(H,H') = d_s(\{h^1,h^2\},\{h'^1,h'^2\}) = d_H(s(h^1,h^2),s(h'^1,h'^2)).$$

Enfin, l'erreur de phasage est le ratio entre le nombre de permutations entre H et H' et le nombre total de permutations possibles.

Erreur de phasage

$$e_{\text{phasage}}(h_1,h_2) = \frac{d_s(h_1,h_2)}{m'-1}$$

Haplotype consensus

En pratique, l'algorithme estime la phase à plusieurs reprises pour différentes initialisations aléatoires. Nous estimons donc l paires d'haplotypes $\{H_1, H_2, \ldots, H_l\}$. L'haplotype consensus est l'haplotype qui minimise l'erreur de phasage vis-à-vis des l haplotypes estimés.

$$H_{\text{consensus}} = \underset{H \in \{0,1\}^{n,m}}{\operatorname{argmin}} \sum_l e_{\text{phasage}}(H,H_l) \tag{3.16}$$

Pour chaque paire de positions hétérozygotes successives, il faut prendre la décision majoritaire entre permuter ou ne pas permuter. Cette façon de construire un haplotype

consensus ne fonctionne que pour des paires d'haplotypes reconstruisant le même génotype. De plus, ce moyennage ne permet pas d'estimer ni A ni S directement pour les haplotypes consensus. Il est nécessaire d'exécuter la méthode à nouveau en fixant H avec $H_{\text{consensus}}$.

3.3 Imputation des données

Les valeurs manquantes de génotypes sont inférées (imputées) dans de nombreux cas grâce à la présence de LD et de blocs d'haplotypes (Howie, Donnelly et Marchini, 2009). Notre formalisme permet aussi de prendre en compte les données manquantes dans une matrice de SNPs. Une matrice de génotypes est désormais définie par $G \in \{0, 1, 2, NaN\}$, où NaN désigne une valeur manquante. Par exemple, le jeu de données des peupliers *Populus* compte 7.5% de valeurs manquantes (Figure 3.4).

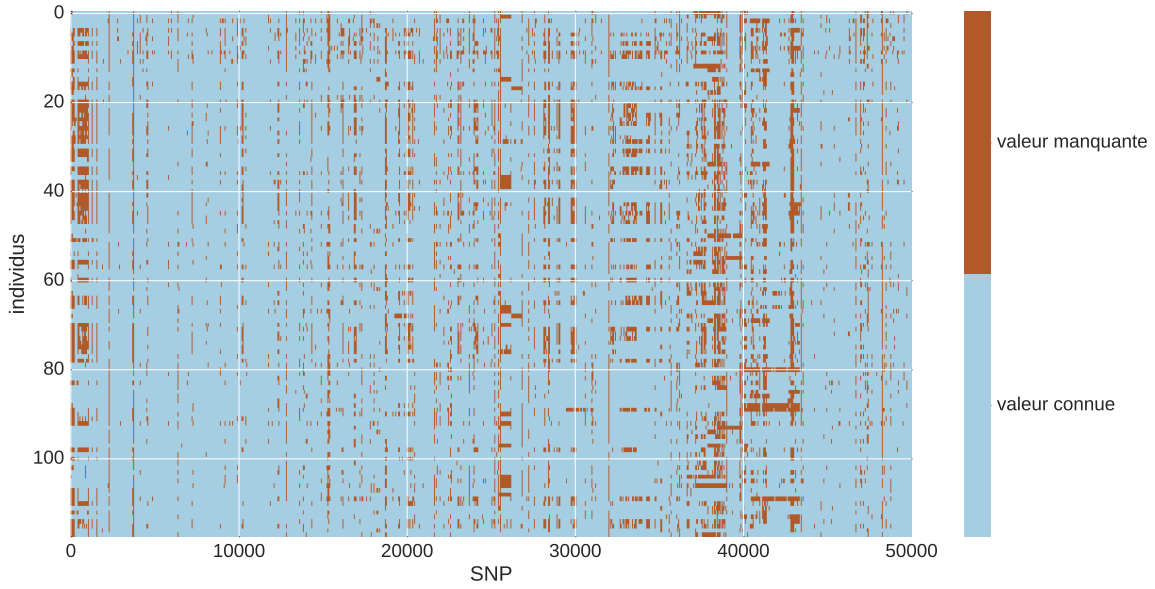

Figure 3.4 – Valeurs manquantes du jeu de données Populus (Suarez-Gonzalez et al., 2016). Environ 7.5% des données sont manquantes, mais leur répartition n'est pas uniforme.

Supposons que le deuxième SNP soit manquant pour l'individu i, $G_{i,2} = NaN$. Pour nos deux modèles, phasage et estimation des coefficients d'ascendance locaux, nous considérons qu'il ne faut alors pas changer de cluster pour ce SNP (Figure 3.5). En effet, le terme d'attache aux données est inconnu pour ce locus. Pour minimiser notre fonction, il faut que $S_{i,1} = S_{i,2}$. Contrairement à tous nos algorithmes, cette opération dépend du sens des marqueurs : les mêmes séquences de SNPs observées de

m à 1 produiraient un résultat différent. C'est pourquoi l'optimisation est effectuée sur la matrice G et la matrice d'ordre inversé G'.

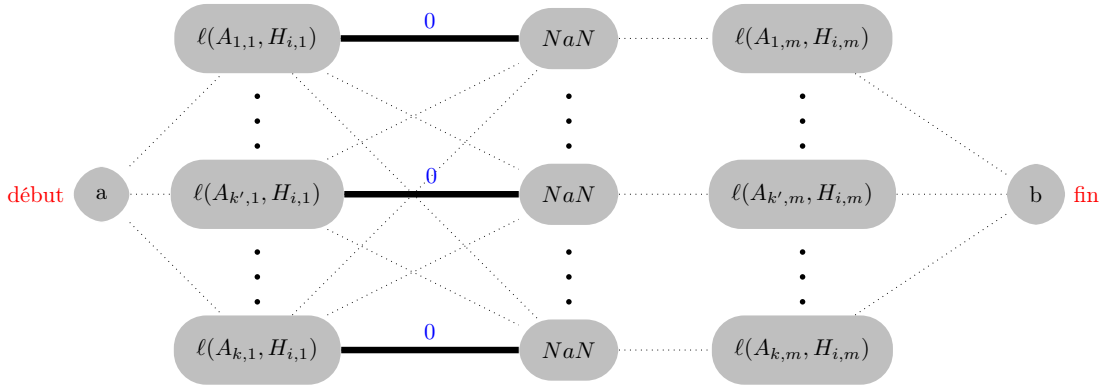

FIGURE 3.5 – Adaptation des graphes de phasage et d'estimation des coefficients de métissage locaux aux valeurs manquantes.

3.4 Résultats

Dans cette partie, nous présentons les résultats sur notre méthode de phasage et sur l'imputation qui en découle.

3.4.1 Reconstruction des haplotypes

La méthode Loter a été comparée pour le phasage à trois méthodes : fastPHASE, Beagle et SHAPEIT. Pour fastPHASE, nous avons eu recours à la version 1.4. Pour Beagle, nous avons utilisé la version 4.1. Pour SHAPEIT, la version "v2" a été utilisée. Ces trois méthodes ont été employées avec leurs paramètres par défaut. Pour ce comparatif du phasage, nous avons choisi les données de HapMap3.r2.b36 [4]. Ce jeu de données contient au total onze populations. Cinq de ces populations contiennent des trios, c'est-à-dire les séquençages des deux parents et de l'enfant. Parmi ces populations avec trios, nous gardons notamment la population CEU, européens du Nord, la population MEX, mexicains, et la population YRI, Yoruba d'Afrique.

La méthode des trios permet au jeu HapMap de reconstruire de façon déterministique le phasage de 80% des sites hétérozygotes et donc au total de 94% des marqueurs pour les individus trios. Ce pourcentage diminue à 85% pour les duos (un parent et un enfant), c'est pourquoi nous n'avons pas retenu ces individus, pour ne pas

4. ftp://ftp.ncbi.nlm.nih.gov/hapmap/phasing/2009-02_phaseIII/HapMap3_r2/

TABLE 3.1 – Nombre d'individus dans chaque population

CEU	MEX	YRI
44	23	50

introduire de biais dans l'analyse. Notons que la phase des sites restants est obtenue avec l'algorithme de phasage IMPUTE v2 (HOWIE, DONNELLY et MARCHINI, 2009).

Les erreurs de phasage des quatre méthodes sont similaires. Toutefois, Loter obtient de moins bons résultats sur la population Yoruba YRI (Figure 3.6).

La figure 3.7 souligne l'importance de la mise à jour stochastique de H. Nous avons constaté que la fonction objectif atteignait des minimas locaux plus rapidement dans le cas de la mise à jour déterministe de H. Ce point a été particulièrement important pour que notre méthode soit comparable à l'état de l'art.

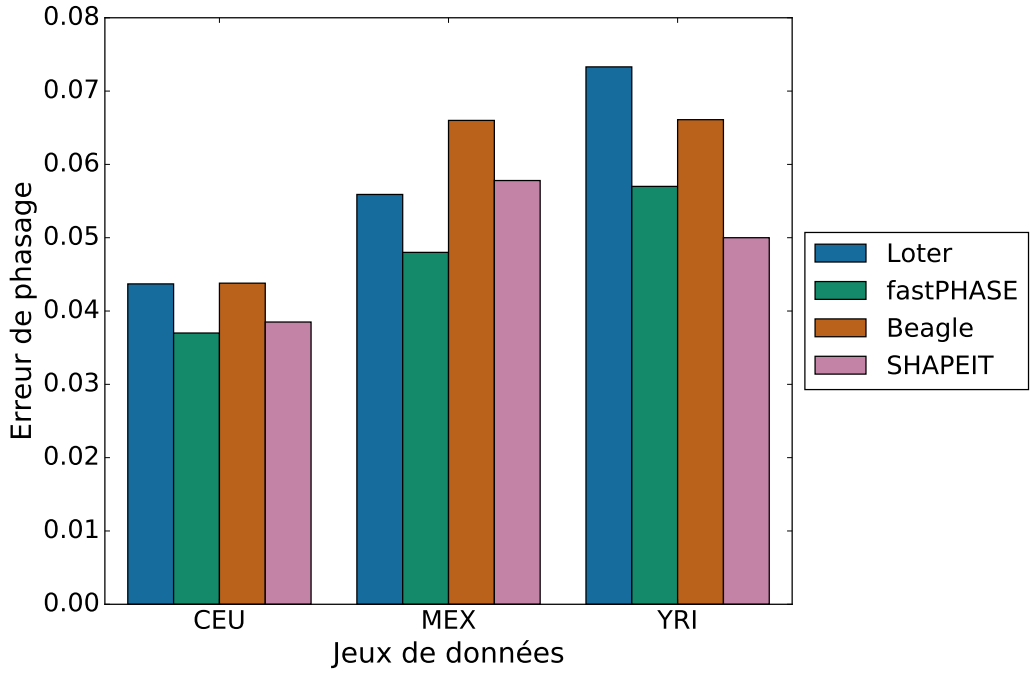

FIGURE 3.6 – Comparaison des méthodes de phasage fastPHASE, Beagle, SHAPEIT et Loter. Trois populations de HapMap sont utilisées : CEU, MEX et YRI.

Le logiciel Loter pour le phasage a besoin de deux paramètres K et λ. Tout comme pour fastPHASE (SCHEET et STEPHENS, 2006), ces paramètres sont estimés grâce à l'erreur d'imputation. En effet, nous masquons 10% des SNPs de la matrice de génotypes G et imputons les valeurs manquantes. Le nombre de SNPs correctement inférés indique la qualité des paramètres. La figure 3.8 illustre le choix de K pour le jeu de données CEU. L'erreur d'imputation est minimale entre $K = 14$ et $K = 22$.

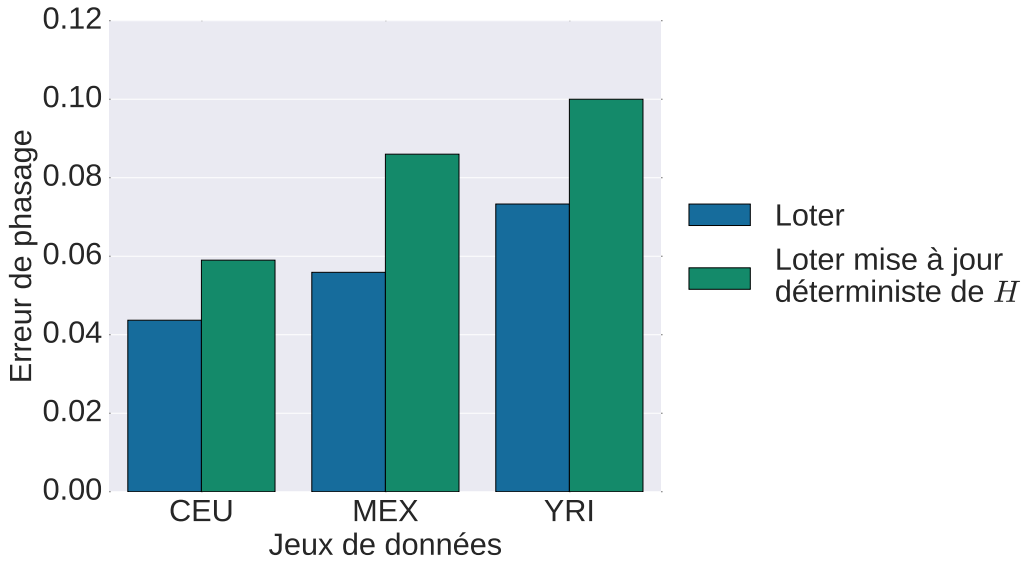

FIGURE 3.7 – Comparaison de notre méthode de phasage avec mise à jour déterministe des haplotypes H et la mise à jour stochastique.

Notre méthode est robuste à des choix de K trop grand. Nous avons constaté que dans ces cas, certains clusters n'étaient pas utilisés, c'est-à-dire qu'aucun haplotype ne provenait d'eux. Dans tous les cas, K ne devrait jamais dépasser le nombre de génotypes et a fortiori le nombre d'haplotypes.

Nous procédons de la même manière pour le choix de λ. La figure 3.9 illustre le choix d'une pénalité égale à deux. Le paramètre λ est donc similaire à celui de la méthode d'estimation des coefficients de métissage locaux.

La précision de notre algorithme dépend du nombre d'estimations (Figure 3.10) et s'améliore grandement avec le nombre d'estimations. Pour une seule estimation, l'erreur de phasage est de 18%. Au-delà de 100 estimations, l'erreur est inférieure à 4.5%. Par défaut, l'algorithme est appliqué 100 fois puis un consensus est calculé.

3.4.2 Imputation

L'imputation a été évaluée sur le jeu de donnée CEU de HapMap. Nous avons simplement comparé les méthodes fastPHASE, Beagle et Loter sur ce problème.

Nous avons masqué de manière aléatoire un certain pourcentage des données. Ces valeurs manquantes sont ensuite inférées par les différentes méthodes. La figure 3.11 illustre que l'erreur d'imputation augmente avec le taux de données manquantes. En effet, l'inférence est plus difficile lorsque les méthodes ont moins d'information. Loter est la méthode la moins précise lorsque moins de 40% des données sont manquantes. Au-delà, Loter est la méthode la plus efficace. Notre méthode convient donc mieux

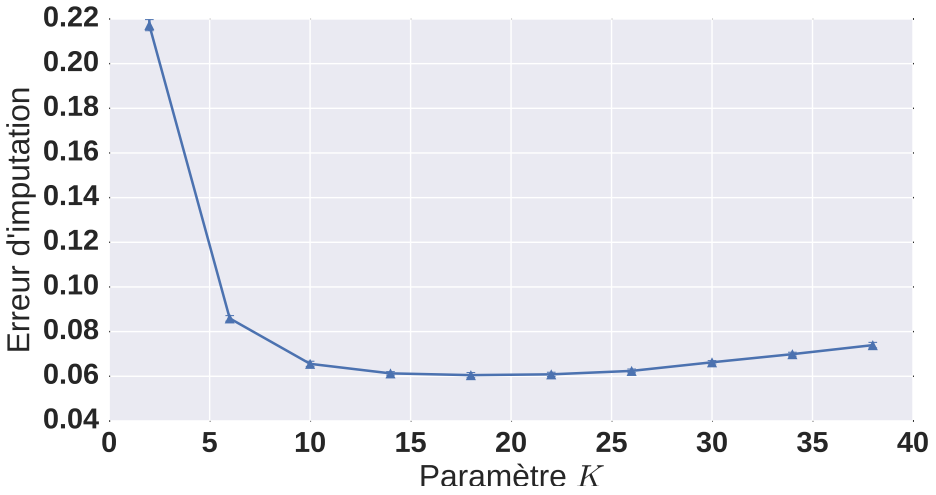

FIGURE 3.8 – Validation croisée de Monte Carlo pour estimer K. Nous masquons 10% de SNPs de la matrice des génotypes CEU de manière aléatoire, cinq fois pour des valeurs de K entre 2 et 40. L'erreur d'imputation est calculée pour chaque valeur de K et chaque échantillonnage des valeurs manquantes. Les barres d'erreur sont indiscernables car l'erreur d'imputation varie peu quelque soit les 10% de données masquées.

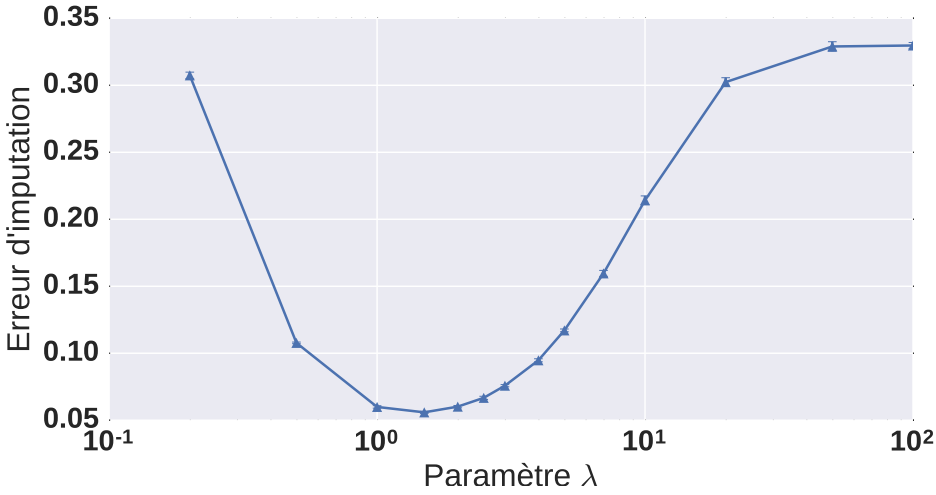

FIGURE 3.9 – Validation croisée de Monte Carlo pour estimer λ. Nous masquons cinq fois 10% des SNPs de la matrice CEU de manière aléatoire et calculons l'erreur d'imputation associée au paramètre λ utilisé. λ varie dans l'ensemble $\{0.2, 0.5, 1, 1.5, 2, 2.5, 3, 4, 5, 7, 10, 20, 50, 100\}$.

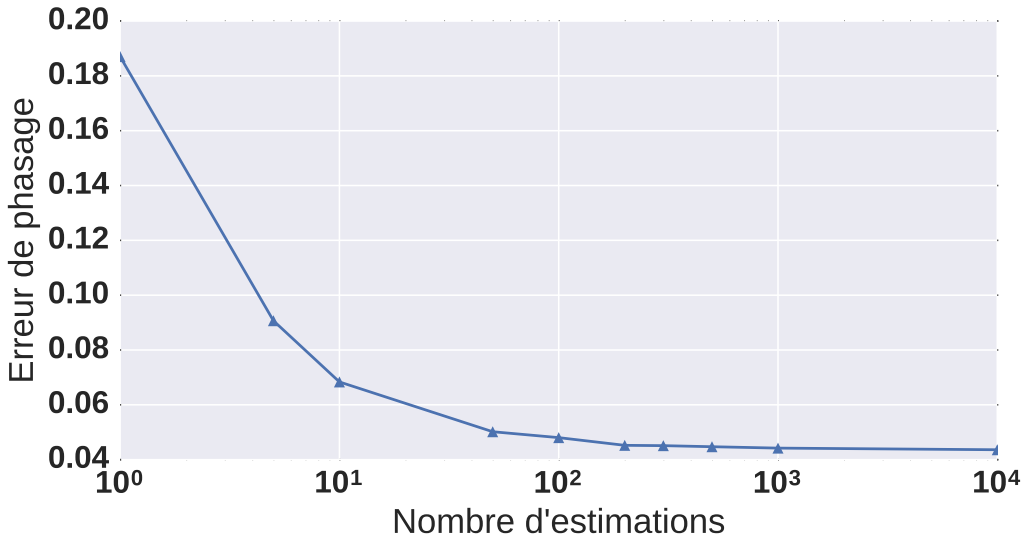

FIGURE 3.10 – Notre algorithme estime à plusieurs reprises la phase des CEU pour différentes initialisations. Un haplotype consensus est ensuite créé. L'erreur de phasage est évaluée en fonction du nombre d'estimations sur une échelle logarithmique.

aux jeux de données avec des taux élevés de valeurs manquantes.

3.5 Conclusion

Bien que notre méthode ne permette pas d'augmenter la précision du phasage, son principal intérêt est sa complexité linéaire par rapport au nombre de clusters. Le chapitre sur l'estimation des coefficients de métissage locaux a montré que les erreurs de phasage peuvent être prises en compte par les méthodes en aval dans le pipeline d'analyse. Dans ce type de pipeline, la rapidité de la méthode est donc aussi de mise.

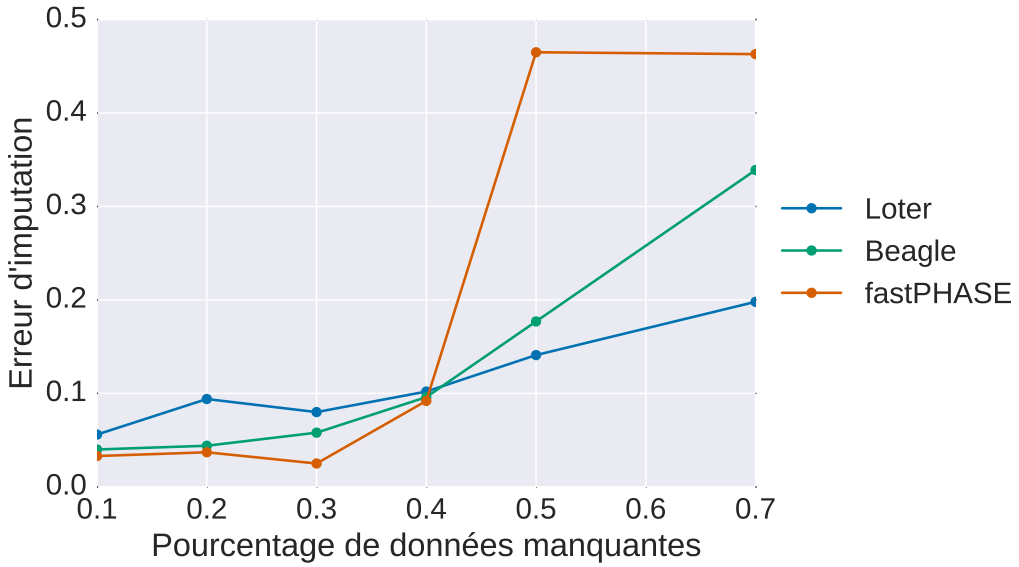

FIGURE 3.11 – Évolution de l'erreur d'imputation en fonction du pourcentage de données manquantes sur les CEU d'HapMap. Comparatif de trois méthodes : fastPHASE, Beagle et Loter.

CHAPITRE 4

Perspectives et Discussion

La modélisation du déséquilibre de liaison avec nos formulations permet de traiter différents problèmes de la génétique des populations comme l'inférence des coefficients de métissage locaux et le phasage. Notre logiciel, Loter, traite ces problèmes avec une complexité algorithmique linéaire et une précision équivalente aux autres méthodes pour le phasage et équivalente ou supérieure pour l'estimation des coefficients d'ascendance locaux. Nous discutons dans cette partie des modifications et des perspectives qui permettraient d'améliorer la vitesse de nos méthodes et leur efficacité, ainsi que les ouvertures qui découlent de nos travaux.

4.1 Évaluation des méthodes discriminatives

Les méthodes discriminatives récentes d'inférence des coefficients de métissage locaux montrent l'importance des mesures de comparaison des segments d'haplotypes. En effet, ces méthodes découpent le problème d'inférence sur des fenêtres non chevauchantes et construisent des classifieurs fenêtre par fenêtre. De cette manière, il est possible de puiser dans toute la littérature des problèmes de classification. À l'image du rapport de Arivazhagan, Kim et Yuan (2015), nous nous sommes questionnés sur l'efficacité des différentes méthodes de l'apprentissage automatique pour le problème particulier des coefficients de métissage locaux. Néanmoins, le rapport de Arivazhagan, Kim et Yuan (2015) ne compare que trois méthodes sans faire varier le temps de métissage : SVM, plus proche voisin et forêts aléatoires. RFMix (Maples et al., 2013) et Ancestry Decomposition (Durand et al., 2014) sont deux exemples de réussite d'intégration de classifieur localisé.

Nous avons donc cherché à déterminer la performance de sept modèles de classification sur les simulations de métissage entre la population CEU et YRI de HapMap, qui est le cas de figure le plus étudié pour les modèles d'estimation des coefficients d'ascendance locaux. Nous avons entraîné chaque méthode sur des fenêtres de tailles variables dans l'ensemble $\{20, 50, 100, 200, 500\}$ pour quatre temps de métissage différents : 5, 100, 200 et 500. Chaque méthode de classification ne peut renvoyer qu'une seule provenance par fenêtre. Or, sur une fenêtre, les coefficients d'ascendance de la simulation ont pu changer. Pour rendre compte de cette erreur inhérente aux méthodes "fenêtrées", nous affichons les scores d'une méthode "oracle" connaissant la vérité et décidant sur la fenêtre de l'ascendance majoritaire. L'oracle donne donc le meilleur score pour une méthode fondée sur des fenêtres.

Le score choisi pour comparer les méthodes est la précision diploïde qui a été déjà utilisée pour nos résultats dans le chapitre sur l'estimation des coefficients de métissage locaux.

Présentation des méthodes comparées

Nous présentons maintenant succintement les méthodes comparées dans cette étude. Nous n'avons pas implémenté ces méthodes, si ce n'est le noyau **K-mer**. Le package *scikit-learn* (Pedregosa et al., 2011) a été utilisé, ainsi que les paramètres par défaut.

SVM. Les machines à vecteurs de support, aussi appelées séparateurs à vaste marge, résolvent le problème de classification comme un problème d'optimisation quadratique. L'objectif est de déterminer l'hyperplan qui maximise la *marge*, c'est-à-dire la distance entre l'hyperplan et les échantillons les plus proches (Boser, Guyon et Vapnik, 1992).

Le SVM est adaptable aux problèmes non-linéaires grâce aux méthodes à noyaux. À notre connaissance, Durand et al. (2014) sont les premiers à proposer une méthode à noyaux en plus du SVM. Ils introduisent le noyau appelé **K-mer** (Figure 4.1) qui représente les données initiales par toutes les sous-chaînes de tailles possibles à chaque position. Soit $\mathcal{A} = \{a_1, ..., a_l\}$ un alphabet de taille l qui code des chaînes de taille m qui appartiennent à l'espace $\mathcal{X} = \mathcal{A}^m$. Le noyau est défini par la fonction $\Phi : \mathcal{X} \to \{0, 1\}^d$ avec $d = \sum_{i=1}^{m}(m+1-i)|\mathcal{A}|^i$. En effet, il existe $|\mathcal{A}|^m$ sous-chaînes de taille m, $2|\mathcal{A}|^{m-1}$ sous-chaînes de taille $m-1$ et ainsi de suite.

Pour calculer le produit scalaire dans l'espace des K-mer, il faut donc compter le nombre de sous-chaînes communes pour chaque position. Pour cela il suffit de trouver les plus longues sous-chaînes communes, c'est-à-dire l'écart de position entre deux paires de caractères différents entre les chaînes. Sachant les plus longues sous-chaînes communes, on en déduit toutes les sous-chaînes imbriquées. Soit L la longueur d'une

sous-chaîne commune, alors on peut compter $\frac{L(L+1)}{2}$ sous-chaînes communes imbriquées. Ce noyau permet donc de mettre en évidence l'importance de la structure locale. Nous testons ce noyau avec le SVM et la méthode KNN.

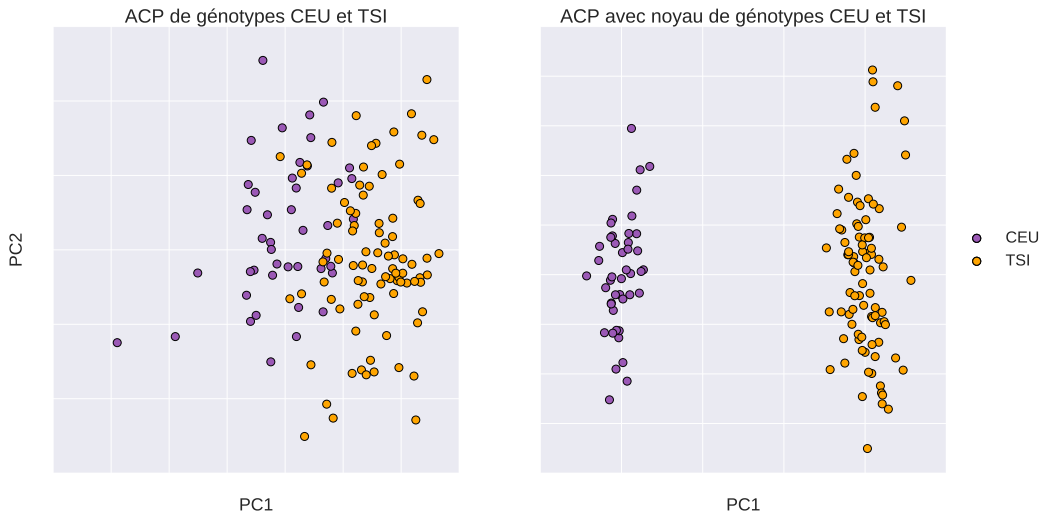

FIGURE 4.1 – Pour visualiser l'importance du noyau K-mer, nous comparons l'ACP avec et sans noyau K-mer sur les individus CEU (européens du nord) et TSI (toscans d'Italie) de HapMap.

KNN. La méthode de classification des plus proches voisins classe les éléments en entrée selon un vote sur les échantillons connus les plus proches. Le paramètre K du nombre de voisins pour voter a été fixé à 3 pour ces expériences.

Forêts aléatoires. Les forêts aléatoires sont un cas de méthodes d'ensemble qui entraînent des arbres décisionnels sur des sous-ensembles de données. Le *bagging* permet d'améliorer la décision en corrigeant notamment le surapprentissage. RFMix (MAPLES et al., 2013) est une méthode qui s'appuie sur des forêts aléatoires.

Gradient Boosted Trees Les forêts aléatoires sont souvent comparées aux techniques de *Gradient Boosted Trees*. Ces techniques de *boosting* apprennent aussi grâce à des arbres décisionnels très simples et pondèrent leur apprentissage en fonction de leurs erreurs. Nous avons utilisé l'implémentation de XGBoost (CHEN et HE, 2014) pour ce comparatif.

Enfin, nous avons inclus un modèle que nous appelons **Li et Stephens** qui est une variante de notre modèle autorisant des recombinaisons au sein de la population et des erreurs dues aux mutations. Nous calculons le plus court chemin dans chaque graphe des populations et comparons ces chemins.

Résultats du comparatif

Le but de cette expérience est avant tout d'ouvrir la discussion. La figure 4.2 synthétise les résultats de cette expérience. Nous constatons que pour le SVM et la méthode KNN, l'ajout d'un noyau augmente considérablement la précision des méthodes. La précision maximale du SVM avec K-mer est de 97% (fenêtre de taille 200 et temps de métissage égal à 5) contre 92% pour le SVM linéaire (fenêtre de taille 500 et temps de métissage égal à 5). De même, pour la méthode KNN, la précision maximale avec noyau est de 95% contre 86% sans noyau. Le modèle de Li et Stephens est la deuxième meilleure méthode en terme de précision maximale avec 96% de précision (fenêtre de taille 200 et temps de métissage égal à 5).

Remarquons que la taille de la fenêtre impacte grandement la qualité des résultats dans cette expérience. Il faut prendre ce point en compte lors de l'analyse de jeux de données avec des méthodes "fenêtrées" comme RFMix. Nous avons d'ailleurs constaté sur Populus que la fenêtre de 0.2 cM dégradait énormément la qualité des résultats.

4.2 Changement de distance

La détection des ruptures est l'analyse des points pour lesquels une propriété est modifiée. Le modèle de Li et Stephens et notre modèle sont aussi des problèmes de segmentation pour lesquels nous cherchons les points de changement d'haplotype.

Supposons que nous avons une séquence $h = (h_1, \ldots, h_m)$ de taille m. Soit l le nombre de points de rupture et $\tau_1 < \ldots < \tau_l$ ces points de rupture tels que $\tau = (\tau_0, \ldots, \tau_{l+1})$ avec $\tau_0 = 0$ et $\tau_{l+1} = m$. Nous notons $h_{[i]} = (h_{\tau_{i-1}+1}, \ldots, \tau_i)$, h limité à l'intervalle entre $\tau_{i-1} + 1$ et τ_i. Le but est alors de trouver une segmentation de h qui minimise :

$$\sum_{i=1}^{l+1} \ell(h_{[i]}) + \lambda l \tag{4.1}$$

La fonction ℓ est une fonction de perte et λ est la pénalité associée à chaque segment.

Maintenant, considérons le cas de figure particulier où nous connaissons K points notés $y^1, \ldots, y^K$ et où la fonction de perte est définie par

$$\ell(x_{[i]}) = \min_{k=1,\ldots,K} d(x_{[i]}, y^k_{[i]}) \tag{4.2}$$

La fonction d est une distance quelconque entre deux éléments. Il s'agit donc de trouver une segmentation telle que le vecteur h est le plus proche d'un y sur le segment

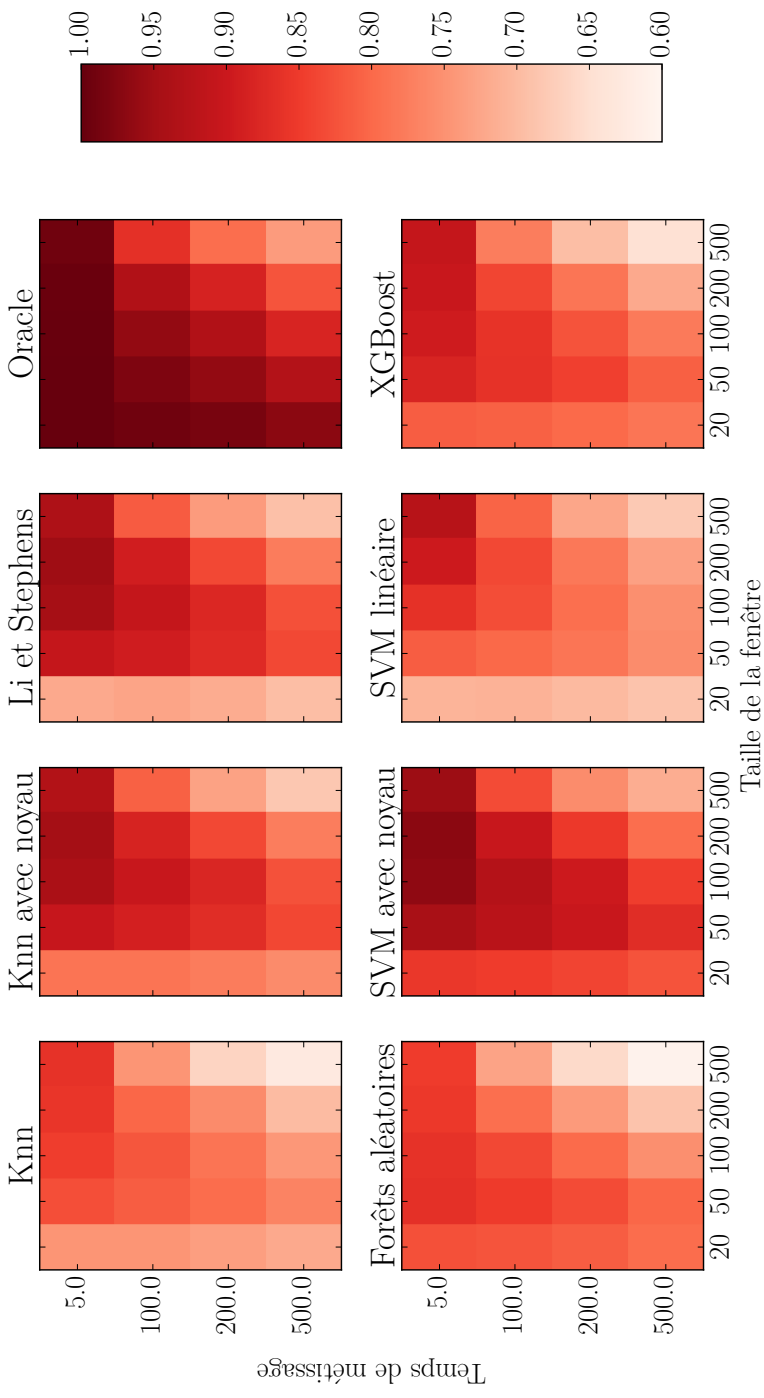

FIGURE 4.2 – Comparaison de différents classifieurs : KNN, KNN avec noyau K-mer, SVM, SVM avec noyau K-mer, Forêts aléatoires, XGBoost, le modèle avec recombinaison appelé "Li et Stephens" et l'oracle. La précision diploïde de chacun de ces modèles est évaluée sur des simulations de métisses entre les populations CEU et YRI de HapMap. Les modèles sont évalués pour différentes tailles de fenêtres et différents temps de métissage.

considéré et telle que le nombre de segments est minimisé. Cette formulation généralise notamment l'équation 2.1 pour l'estimation des coefficients de métissage locaux.

Pour cela nous posons

$$\begin{aligned} & d(x,y) = \sum_i |x_i - y_i| \\ & h \in \{0,1\},\ \text{l'haplotype métisse} \\ & H \in \{0,1\}^{K,m},\ \text{avec } H_i = y^i \\ & \alpha_i = \underset{k=1,\dots,K}{\operatorname{argmin}}\, d(x_{[i]}, y^k_{[i]}) \\ & s \text{ qui contient } \alpha_i \text{ répété } \tau_i - \tau_{i-1} \forall i \end{aligned} \tag{4.3}$$

$$\begin{aligned} \min_\tau \sum_{i=1}^{l+1} \ell(h_{[i]}) + \lambda l &= \min_\tau \sum_{i=1}^{l+1} \min_{k=1,\dots,K} d(h_{[i]}, y^k_{[i]}) + \lambda l \\ &= \min_\tau \sum_{i=1}^{l+1} \min_{k=1,\dots,K} \left\{ \sum_{j=\tau_{i-1}+1}^{\tau i} |h_j - y^k_j| \right\} + \lambda l \\ &= \min_\tau \sum_{i=1}^{l+1} \sum_{j=\tau_{i-1}+1}^{\tau_i} |h_j - H_{\alpha_i, j}| + \lambda l \end{aligned} \tag{4.4}$$

Puisque τ indique les points de changement et α_i la provenance, nous pouvons utiliser la notation s qui indique la provenance et les points de changement à chaque différence de valeurs consécutives.

$$\min_\tau \sum_{i=1}^{l+1} \ell(h_{[i]}) + \lambda l = \min_s \sum_{j=1}^{m} |h_j - H_{s_j, j}| + \lambda \sum_{j=1}^{m-1} 1_{s_i \neq s_{i+1}} \tag{4.5}$$

En plus de retrouver les résultats précédents nous pouvons donc utiliser cette formulation pour résoudre le problème avec d'autres fonctions d quelconques. Néanmoins, pour une fonction d quelconque la programmation dynamique se fait sur la position du dernier changement et ces algorithmes sont a priori de complexité quadratique par rapport au nombre de SNPs.

Au regard de la partie sur la comparaison des classifieurs 4.1, ce changement de distance possible répond aux exigences d'adapter la fonction de comparaison.

4.3 Modification du graphe

Les méthodes discriminatives telles que RFMix ont l'avantage de pouvoir traiter des populations très ressemblantes telles que les populations CEU, européens du Nord échantillonés en Utah et TSI, toscans d'Italie, de HapMap. Pour pouvoir traiter des

populations très similaires, nous proposons une nouvelle formulation pour laquelle la pénalité de changer de provenance est différente selon si l'on reste dans la même population ou si l'on en change.

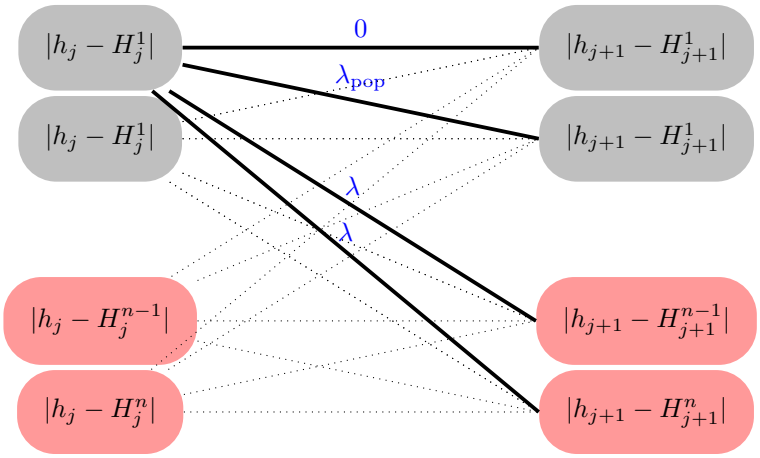

FIGURE 4.3 – Illustration du graphe avec différentes pénalités. Ces pénalités dépendent de l'information sur les populations ou sur les familles. Deux populations sont représentées en rouge et en gris.

Cette modification du problème d'optimisation et donc du graphe favorise les recombinaisons au sein d'un groupe d'haplotypes avec un attribut commun, qui peut être une information de population ou de famille. Néanmoins, la complexité algorithmique de nos méthodes croît linéairement avec le nombre de poids différents dans le graphe. Dans le cas où les poids sur les arrêtes sont tous différents, notre méthode devient quadratique par rapport au nombre d'individus de référence.

4.4 Transformée de Burrows-Wheeler

La transformée de Burrows-Wheeler (BURROWS et WHEELER, 1994) (BWT) est un algorithme de compression de données. Cette transformation réorganise les données et permet d'effectuer des opérations sur les chaînes de caractères de manière très efficace comme par exemple la recherche de la plus longue sous-chaîne commune. Ces propriétés de comparaison de chaînes de caractères ont naturellement trouvé leur utilité pour le séquençage de l'ADN où des algorithmes fondés sur la BWT permettent d'améliorer l'alignement des séquences d'ADN (LANGMEAD et al., 2009). DURBIN, 2014 a exhibé une variante positionnelle de cette transformée pour faire des comparaisons d'haplotypes de manière voisine aux nombreuses méthodes reposant sur des comparaisons probabilistes et des HMMs. Durbin conclut que sa méthode

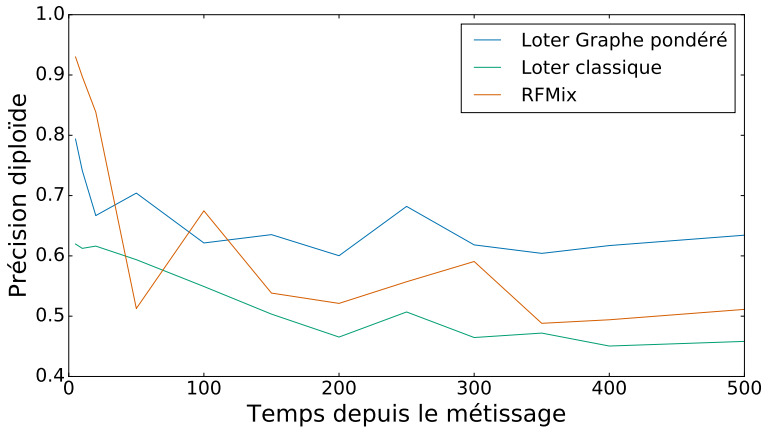

FIGURE 4.4 – Résultats préliminaires sur des simulations de métisses CEU et TSI. "Loter Graphe pondéré" correspond à une seule évaluation par la méthode avec $\lambda_{pop} = 2$ et $\lambda = 40$. En revanche, RFMix et Loter sont appliquées de manière classique, avec 160 évaluations pour Loter notamment.

pourrait être adaptée pour des logiciels compressant les données tels que Beagle et permettrait, contrairement à SHAPEIT, de traiter le problème du phasage sans approximation. Cette idée a récemment été reprise par LUNTER (2016) qui démontre l'équivalence de la BWT et de la variante positionnelle de Durbin. De plus, LUNTER (2016) propose un algorithme calculant l'haplotype le plus probable avec le modèle de Li et Stephens grâce à la BWT. L'auteur estime que la complexité algorithmique de l'étape de programmation dynamique ne dépend plus de la taille de la population d'haplotypes de référence. Même si cela reste une approximation, ce type de méthode très prometteuse peut tout à fait être adaptée à notre algorithme. En effet, l'étape de programmation dynamique de nos méthodes est similaire à celle de LUNTER (2016). Notons que la variante positionnelle de la BWT est à la base du logiciel de phasage avec références Eagle2 (LOH et al., 2016).

4.5 Développement logiciel

Parmi les logiciels testés au cours de la thèse (HAPMIX, RFMix, LAMP-LD, Beagle, etc.), aucun n'est disponible via un langage interprété tel que R ou Python. Pour l'ensemble de ces logiciels, nous avons dû créer des "wrappers", c'est-à-dire des modules Python, pour pouvoir les exécuter de manière répétée en faisant varier les paramètres. Pour le développement de Loter, nous avons suivi l'architecture logicielle de XGBoost (CHEN et GUESTRIN, 2016). Le coeur des algorithmes est développé en C++ pour

combiner rapidité et extensibilité. Une surcouche C permet un *binding*[1] avec une majorité de langages. En effet, la plupart des langages supportent mieux la connexion à une bibliothèque C que C++. Enfin, nous avons développé un module Python qui permet d'interagir avec la bibliothèque et de réaliser facilement les opérations. L'utilisateur peut plus facilement faire son analyse de données en R ou en Python. L'expérimentateur module à son gré l'algorithme pour tester des variantes et itérer sur les prochaines versions. À l'heure actuelle, seul le module Python a été développé, mais notre architecture logicielle permettrait de créer un module équivalent en R par exemple.

Nous avons choisi d'intégrer nos deux méthodes (phasage et estimation des coefficients de métissage locaux) au sein du même module Python : Loter.

4.6 Conclusion

De nombreuses perspectives sont possibles pour le formalisme d'optimisation que nous avons établi. La transformée de Burrows-Wheeler est très certainement une des prochaines étapes d'amélioration algorithmique de nos méthodes. Cette transformation permettrait aussi de traiter des jeux de données d'échelle encore plus grande. Pour l'estimation des coefficients de métissage locaux, redéfinir la distance et imposer plus de contraintes au graphe semblent prometteurs pour s'attaquer à l'analyse de populations métisses plus complexes. Enfin pour le phasage et l'imputation, la structure compressée des haplotypes semble un point majeur et un axe de progression de notre méthode.

1. Connexion logicielle entre un langage et une bibliothèque dans un autre langage.

Glossaire

$\mathcal{H} = \{0,1\}^m$, l'ensemble de tous les haplotypes possibles. 66, 67

$\mathcal{H}_G$ l'ensemble de tous les haplotypes possibles reconstruisant G. 67

$\mathcal{H}_G^2$ l'ensemble de toutes les paires ordonnées d'haplotypes possibles reconstruisant G. 67

$G \in \{0,1,2\}^{n \times m}$, matrice de n génotypes définis sur m loci. 66

$g \to h \equiv g - h \in \mathcal{H}$, g et h sont compatibles. 66, 67

$h \in \{0,1\}^m$, représentant un haplotype défini sur m loci. 32

$H \in \{0,1\}^{2n \times m}$, matrice de $2n$ haplotypes définis sur m loci. 32, 66

ACP Analyse en Composantes Principales. 5, 18

cM centiMorgan, unité de distance entre deux gènes. 28

EM Algorithme Espérance Maximisation, Expectation Maximization en anglais. 22, 68

HMM Modèle de Markov caché, Hidden Markov Model en anglais. 11, 22, 69

KNN K plus proches voisins, K Nearest Neighbours en anglais. 87

LD Déséquilibre de liaison, Linkage Disequilibrium en anglais. 4, 19

SVM Machine à vecteurs de support, Support Vector Machine en anglais. 86

Bibliographie

The International HapMap Consortium (2005). "A haplotype map of the human genome". In : *Nature* 437.7063, 1299–1320. ISSN : 1476-4679. DOI : 10.1038/nature04226. URL : http://dx.doi.org/10.1038/nature04226.

Brookes, Anthony J. (1999). "The essence of SNPs". In : *Gene* 234.2, 177–186. ISSN : 0378-1119. DOI : 10.1016/s0378-1119(99)00219-x. URL : http://dx.doi.org/10.1016/s0378-1119(99)00219-x.

Wall, Jeffrey D. et Jonathan K. Pritchard (2003). "Haplotype blocks and linkage disequilibrium in the human genome". In : *Nature Reviews Genetics* 4.8, 587–597. ISSN : 1471-0064. DOI : 10.1038/nrg1123. URL : http://dx.doi.org/10.1038/nrg1123.

Pritchard, Jonathan K. et Molly Przeworski (2001). "Linkage Disequilibrium in Humans : Models and Data". In : *The American Journal of Human Genetics* 69.1, 1–14. ISSN : 0002-9297. DOI : 10.1086/321275. URL : http://dx.doi.org/10.1086/321275.

Long, Anthony D et Charles H Langley (1999). "The power of association studies to detect the contribution of candidate genetic loci to variation in complex traits". In : *Genome Research* 9.8, p. 720–731.

Galinsky, Kevin J et al. (2015). "Fast principal components analysis reveals convergent evolution of ADH1B gene in Europe and East Asia". In : DOI : 10.1101/018143. URL : http://dx.doi.org/10.1101/018143.

Abdellaoui, Abdel et al. (2013). "Population structure, migration, and diversifying selection in the Netherlands". In : *European Journal of Human Genetics* 21.11, 1277–1285. ISSN : 1476-5438. DOI : 10.1038/ejhg.2013.48. URL : http://dx.doi.org/10.1038/ejhg.2013.48.

Scheet, Paul et Matthew Stephens (2006). "A Fast and Flexible Statistical Model for Large-Scale Population Genotype Data : Applications To Inferring Missing Genotypes and Haplotypic Phase". In : *The American Journal of Human Genetics* 78.4, p. 629–644. DOI : 10.1086/502802. URL : https://doi.org/10.1086/502802.

Browning, Sharon R. et Brian L. Browning (2007). "Rapid and Accurate Haplotype Phasing and Missing-Data Inference for Whole-Genome Association Studies By Use of Localized Haplotype Clustering". In : *The American Journal of Human Genetics* 81.5, 1084–1097. ISSN : 0002-9297. DOI : 10.1086/521987. URL : http://dx.doi.org/10.1086/521987.

Stephens, Matthew et Paul Scheet (2005). "Accounting for Decay of Linkage Disequilibrium in Haplotype Inference and Missing-Data Imputation". In : *The American Journal of Human Genetics* 76.3, 449–462. ISSN : 0002-9297. DOI : 10.1086/428594. URL : http://dx.doi.org/10.1086/428594.

Jung, Ho-Youl et al. (2007). "New methods for imputation of missing genotype using linkage disequilibrium and haplotype information". In : *Information Sciences* 177.3, 804–814. ISSN : 0020-0255. DOI : 10.1016/j.ins.2006.07.017. URL : http://dx.doi.org/10.1016/j.ins.2006.07.017.

Donnelly, Peter et Stephen Leslie (2010). "The coalescent and its descendants". In : sous la dir. de N. H. Bingham et C. M.Editors Goldie, 204–237.

Haldane, JBS (1919). "The combination of linkage values and the calculation of distances between the loci of linked factors". In : *J Genet* 8.29, p. 299–309.

Kingman, J. F. C. (1982). "On the Genealogy of Large Populations". In : *Journal of Applied Probability* 19, p. 27. ISSN : 0021-9002. DOI : 10.2307/3213548. URL : http://dx.doi.org/10.2307/3213548.

McVean, G. A. T. et N. J. Cardin (2005). "Approximating the coalescent with recombination". In : *Philosophical Transactions of the Royal Society B : Biological Sciences* 360.1459, 1387–1393. ISSN : 1471-2970. DOI : 10.1098/rstb.2005.1673. URL : http://dx.doi.org/10.1098/rstb.2005.1673.

Hudson, Richard R (1983). "Properties of a neutral allele model with intragenic recombination". In : *Theoretical population biology* 23.2, p. 183–201.

Hudson, R. R. (2002). "Generating samples under a Wright-Fisher neutral model of genetic variation". In : *Bioinformatics* 18.2, 337–338. ISSN : 1460-2059. DOI : 10.1093/bioinformatics/18.2.337. URL : http://dx.doi.org/10.1093/bioinformatics/18.2.337.

Li, Na et Matthew Stephens (2003). "Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data". In : *Genetics* 165.4, p. 2213–2233.

Stephens, Matthew et Peter Donnelly (2000). "Inference in molecular population genetics". In : *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 62.4, p. 605–635.

Fearnhead, Paul et Peter Donnelly (2001). "Estimating recombination rates from population genetic data". In : *Genetics* 159.3, p. 1299–1318.

Browning, S. R. et B. S. Weir (2010). "Population Structure With Localized Haplotype Clusters". In : *Genetics* 185.4, 1337–1344. ISSN : 0016-6731. DOI : 10.1534/genetics.110.116681. URL : http://dx.doi.org/10.1534/genetics.110.116681.

Stumpf, Michael P.H. (2002). "Haplotype diversity and the block structure of linkage disequilibrium". In : *Trends in Genetics* 18.5, 226–228. ISSN : 0168-9525. DOI :

10.1016/s0168-9525(02)02641-0. URL : http://dx.doi.org/10.1016/s0168-9525(02)02641-0.

Gravel, S. (2012). "Population Genetics Models of Local Ancestry". In : *Genetics* 191.2, 607–619. ISSN : 0016-6731. DOI : 10.1534/genetics.112.139808. URL : http://dx.doi.org/10.1534/genetics.112.139808.

Novembre, John et al. (2008). "Genes mirror geography within Europe". In : *Nature* 456.7218, p. 98.

Cavalli-Sforza, L. L. et A. W. F. Edwards (1967). "Phylogenetic Analysis : Models and Estimation Procedures". In : *Evolution* 21.3, p. 550. ISSN : 0014-3820. DOI : 10.2307/2406616. URL : http://dx.doi.org/10.2307/2406616.

Pickrell, Joseph K. et Jonathan K. Pritchard (2012). "Inference of Population Splits and Mixtures from Genome-Wide Allele Frequency Data". In : *PLoS Genetics* 8.11. Sous la dir. d'HuaEditor Tang, e1002967. ISSN : 1553-7404. DOI : 10.1371/journal.pgen.1002967. URL : http://dx.doi.org/10.1371/journal.pgen.1002967.

Pritchard, Jonathan K., Matthew Stephens et Peter Donnelly (2000). "Inference of Population Structure Using Multilocus Genotype Data". In : *Genetics* 155.2, p. 945–959. ISSN : 0016-6731. eprint : http://www.genetics.org/content/155/2/945.full.pdf. URL : http://www.genetics.org/content/155/2/945.

Falush, Daniel, Matthew Stephens et Jonathan K. Pritchard (2003). "Inference of Population Structure Using Multilocus Genotype Data : Linked Loci and Correlated Allele Frequencies". In : *Genetics* 164.4, p. 1567–1587. ISSN : 0016-6731. eprint : http://www.genetics.org/content/164/4/1567.full.pdf. URL : http://www.genetics.org/content/164/4/1567.

Tang, Hua et al. (2006). "Reconstructing Genetic Ancestry Blocks in Admixed Individuals". In : *The American Journal of Human Genetics* 79.1, 1–12. ISSN : 0002-9297. DOI : 10.1086/504302. URL : http://dx.doi.org/10.1086/504302.

Padhukasahasram, Badri (2014). "Inferring ancestry from population genomic data and its applications". In : *Frontiers in Genetics* 5. ISSN : 1664-8021. DOI : 10.3389/fgene.2014.00204. URL : http://dx.doi.org/10.3389/fgene.2014.00204.

Price, Alkes L. et al. (2009). "Sensitive Detection of Chromosomal Segments of Distinct Ancestry in Admixed Populations". In : *PLoS Genetics* 5.6. Sous la dir. de Jonathan K.Editor Pritchard, e1000519. ISSN : 1553-7404. DOI : 10.1371/journal.pgen.1000519. URL : http://dx.doi.org/10.1371/journal.pgen.1000519.

Sundquist, A. et al. (2008). "Effect of genetic divergence in identifying ancestral origin using HAPAA". In : *Genome Research* 18.4, 676–682. ISSN : 1088-9051. DOI : 10.1101/gr.072850.107. URL : http://dx.doi.org/10.1101/gr.072850.107.

Baran, Y. et al. (2012). "Fast and accurate inference of local ancestry in Latino populations". In : *Bioinformatics* 28.10, 1359–1367. ISSN : 1460-2059. DOI : 10.1093/bioinformatics/bts144. URL : http://dx.doi.org/10.1093/bioinformatics/bts144.

Liu, Yushi et al. (2013). "Softwares and methods for estimating genetic ancestry in human populations". In : *Human Genomics* 7.1, p. 1. ISSN : 1479-7364. DOI : 10.1186/1479-7364-7-1. URL : http://dx.doi.org/10.1186/1479-7364-7-1.

Hui, Daniel et al. (2017). “LAIT : a local ancestry inference toolkit”. In : *BMC Genetics* 18.1. ISSN : 1471-2156. DOI : 10.1186/s12863-017-0546-y. URL : http://dx.doi.org/10.1186/s12863-017-0546-y.

Brisbin, Abra et al. (2012). “PCAdmix : Principal Components-Based Assignment of Ancestry Along Each Chromosome in Individuals with Admixed Ancestry from Two or More Populations”. In : *Human Biology* 84.4, 343–364. ISSN : 1534-6617. DOI : 10.3378/027.084.0401. URL : http://dx.doi.org/10.3378/027.084.0401.

Omberg, Larsson et al. (2012). “Inferring genome-wide patterns of admixture in Qataris using fifty-five ancestral populations”. In : *BMC Genetics* 13.1, p. 49. ISSN : 1471-2156. DOI : 10.1186/1471-2156-13-49. URL : http://dx.doi.org/10.1186/1471-2156-13-49.

Maples, BrianK. et al. (2013). “RFMix : A Discriminative Modeling Approach for Rapid and Robust Local-Ancestry Inference”. In : *The American Journal of Human Genetics* 93.2, 278–288. ISSN : 0002-9297. DOI : 10.1016/j.ajhg.2013.06.020. URL : http://dx.doi.org/10.1016/j.ajhg.2013.06.020.

Durand, Eric Y et al. (2014). “Ancestry Composition : A Novel, Efficient Pipeline for Ancestry Deconvolution”. In : *bioRxiv*. DOI : 10.1101/010512. eprint : http://www.biorxiv.org/content/early/2014/10/18/010512.full.pdf. URL : http://www.biorxiv.org/content/early/2014/10/18/010512.

Sankararaman, Sriram et al. (2008). “Estimating Local Ancestry in Admixed Populations”. In : *The American Journal of Human Genetics* 82.2, 290–303. ISSN : 0002-9297. DOI : 10.1016/j.ajhg.2007.09.022. URL : http://dx.doi.org/10.1016/j.ajhg.2007.09.022.

Ng, Andrew Y. et Michael I. Jordan (2002). “On Discriminative vs. Generative Classifiers : A comparison of logistic regression and naive Bayes”. In : sous la dir. de T. G. Dietterich, S. Becker et Z. Ghahramani, p. 841–848. URL : http://papers.nips.cc/paper/2020-on-discriminative-vs-generative-classifiers-a-comparison-of-logistic-regression-and-naive-bayes.pdf.

Yang, J. J. et al. (2013). “Efficient inference of local ancestry”. In : *Bioinformatics* 29.21, 2750–2756. ISSN : 1460-2059. DOI : 10.1093/bioinformatics/btt488. URL : http://dx.doi.org/10.1093/bioinformatics/btt488.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA : Springer-Verlag New York, Inc. ISBN : 0387310738.

Suarez-Gonzalez, Adriana et al. (2016). “Genomic and functional approaches reveal a case of adaptive introgression fromPopulus balsamifera(balsam poplar) in P. trichocarpa(black cottonwood)”. In : *Molecular Ecology* 25.11, 2427–2442. ISSN : 0962-1083. DOI : 10.1111/mec.13539. URL : http://dx.doi.org/10.1111/mec.13539.

Luu, Keurcien, Eric Bazin et Michael G. B. Blum (2017). “pcadapt : an R package to perform genome scans for selection based on principal component analysis”. In : *Molecular Ecology Resources* 17.1, p. 67–77. ISSN : 1755-0998. DOI : 10.1111/1755-0998.12592. URL : http://dx.doi.org/10.1111/1755-0998.12592.

Pritchard, Jonathan K. et al. (2000). "Association Mapping in Structured Populations". In : *The American Journal of Human Genetics* 67.1, 170–181. issn : 0002-9297. doi : 10.1086/302959. url : http://dx.doi.org/10.1086/302959.

Xu, Hongyan et Sanjay Shete (2005). "Effects of population structure on genetic association studies". In : *BMC Genetics* 6.Suppl 1, S109. issn : 1471-2156. doi : 10.1186/1471-2156-6-s1-s109. url : http://dx.doi.org/10.1186/1471-2156-6-s1-s109.

Zhou, Xiang et Matthew Stephens (2012). "Genome-wide efficient mixed-model analysis for association studies". In : *Nature Genetics* 44.7, 821–824. issn : 1546-1718. doi : 10.1038/ng.2310. url : http://dx.doi.org/10.1038/ng.2310.

Seldin, Michael F (2007). "Admixture mapping as a tool in gene discovery". In : *Current Opinion in Genetics Development* 17.3, 177–181. issn : 0959-437X. doi : 10.1016/j.gde.2007.03.002. url : http://dx.doi.org/10.1016/j.gde.2007.03.002.

Seldin, Michael F., Bogdan Pasaniuc et Alkes L. Price (2011). "New approaches to disease mapping in admixed populations". In : *Nature Reviews Genetics* 12.8, 523–528. issn : 1471-0064. doi : 10.1038/nrg3002. url : http://dx.doi.org/10.1038/nrg3002.

Hoggart, C.J. et al. (2004). "Design and Analysis of Admixture Mapping Studies". In : *The American Journal of Human Genetics* 74.5, 965–978. issn : 0002-9297. doi : 10.1086/420855. url : http://dx.doi.org/10.1086/420855.

Shriner, Daniel (2013). "Overview of Admixture Mapping". In : *Current Protocols in Human Genetics*. doi : 10.1002/0471142905.hg0123s76. url : http://dx.doi.org/10.1002/0471142905.hg0123s76.

Freedman, M. L. et al. (2006). "Admixture mapping identifies 8q24 as a prostate cancer risk locus in African-American men". In : *Proceedings of the National Academy of Sciences* 103.38, 14068–14073. issn : 1091-6490. doi : 10.1073/pnas.0605832103. url : http://dx.doi.org/10.1073/pnas.0605832103.

Bensen, Jeannette T. et al. (2013). "Admixture mapping of prostate cancer in African Americans participating in the North Carolina-Louisiana Prostate Cancer Project (PCaP)". In : *The Prostate* 74.1, 1–9. issn : 0270-4137. doi : 10.1002/pros.22722. url : http://dx.doi.org/10.1002/pros.22722.

Darvasi, Ariel et Sagiv Shifman (2005). "The beauty of admixture". In : *Nature Genetics* 37.2, 118–119. issn : 1061-4036. doi : 10.1038/ng0205-118. url : http://dx.doi.org/10.1038/ng0205-118.

Tang, Hua et al. (2007). "Recent Genetic Selection in the Ancestral Admixture of Puerto Ricans". In : *The American Journal of Human Genetics* 81.3, 626–633. issn : 0002-9297. doi : 10.1086/520769. url : http://dx.doi.org/10.1086/520769.

Bryc, K. et al. (2010). "Genome-wide patterns of population structure and admixture among Hispanic/Latino populations". In : *Proceedings of the National Academy of Sciences* 107.Supplement$_2$, 8954–8961. issn : 1091-6490. doi : 10.1073/pnas.0914618107. url : http://dx.doi.org/10.1073/pnas.0914618107.

Hellenthal, G. et al. (2014). "A Genetic Atlas of Human Admixture History". In : *Science* 343.6172, 747–751. issn : 1095-9203. doi : 10.1126/science.1243518. url : http://dx.doi.org/10.1126/science.1243518.

Reese, William D. et Elbert L. Little (1972). “Atlas of United States Trees. Volume 1. Conifers and Important Hardwoods”. In : *The Bryologist* 75.1, p. 120. issn : 0007-2745. doi : 10.2307/3241543. url : http://dx.doi.org/10.2307/3241543.

Liang, M. et R. Nielsen (2014). “The Lengths of Admixture Tracts”. In : *Genetics* 197.3, 953–967. issn : 0016-6731. doi : 10.1534/genetics.114.162362. url : http://dx.doi.org/10.1534/genetics.114.162362.

Carmi, S., J. Xue et I. Pe’er (2015). “A note on the distribution of admixture segment lengths and ancestry proportions under pulse and two-wave admixture models”. In : *ArXiv e-prints*. arXiv : 1509.05904 [q-bio.PE].

Guan, Y. (2014). “Detecting Structure of Haplotypes and Local Ancestry”. In : *Genetics* 196.3, 625–642. issn : 0016-6731. doi : 10.1534/genetics.113.160697. url : http://dx.doi.org/10.1534/genetics.113.160697.

Smith, Michael W et al. (2004). “A high-density admixture map for disease gene discovery in African Americans”. In : *The American Journal of Human Genetics* 74.5, p. 1001–1013.

Huang, Lucy et al. (2009). “Genotype-Imputation Accuracy across Worldwide Human Populations”. In : *The American Journal of Human Genetics* 84.2, 235–250. issn : 0002-9297. doi : 10.1016/j.ajhg.2009.01.013. url : http://dx.doi.org/10.1016/j.ajhg.2009.01.013.

Ali, Shahid, Sreenivas Sremath Tirumala et Abdolhossein Sarrafzadeh (2015). “Ensemble learning methods for decision making : Status and future prospects”. In : *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*. doi : 10.1109/icmlc.2015.7340924. url : http://dx.doi.org/10.1109/icmlc.2015.7340924.

Chen, Tianqi et Tong He (2014). “Higgs Boson Discovery with Boosted Trees”. In : *Proceedings of the 2014 International Conference on High-Energy Physics and Machine Learning - Volume 42*. HEPML’14. JMLR.org, p. 69–80. url : http://dl.acm.org/citation.cfm?id=2996850.2996854.

Chen, Tianqi et Carlos Guestrin (2016). “XGBoost : A Scalable Tree Boosting System”. In : *CoRR* abs/1603.02754. url : http://arxiv.org/abs/1603.02754.

Breiman, Leo (1996). “Bagging predictors”. In : *Machine Learning* 24.2, 123–140. issn : 1573-0565. doi : 10.1007/bf00058655. url : http://dx.doi.org/10.1007/bf00058655.

Efron, Bradley et Robert J. Tibshirani (1993). “Introduction”. In : *An Introduction to the Bootstrap*, 1–9. doi : 10.1007/978-1-4899-4541-9_1. url : http://dx.doi.org/10.1007/978-1-4899-4541-9_1.

Browning, Sharon R. et Brian L. Browning (2011). “Haplotype phasing : existing methods and new developments”. In : *Nature Reviews Genetics* 12.10, 703–714. issn : 1471-0064. doi : 10.1038/nrg3054. url : http://dx.doi.org/10.1038/nrg3054.

Xing, Eric P. et al. (2006). “Bayesian multi-population haplotype inference via a hierarchical dirichlet process mixture”. In : *Proceedings of the 23rd international conference on Machine learning - ICML ’06*. doi : 10.1145/1143844.1143976. url : http://dx.doi.org/10.1145/1143844.1143976.

Kalpakis, K. et P. Namjoshi (2005). “Haplotype Phasing Using Semidefinite Programming”. In : *Fifth IEEE Symposium on Bioinformatics and Bioengineering*

(BIBE'05). DOI : 10.1109/bibe.2005.36. URL : http://dx.doi.org/10.1109/bibe.2005.36.

DELANEAU, Olivier, Jonathan MARCHINI et Jean-François ZAGURY (2011). "A linear complexity phasing method for thousands of genomes". In : *Nature Methods* 9.2, 179–181. ISSN : 1548-7105. DOI : 10.1038/nmeth.1785. URL : http://dx.doi.org/10.1038/nmeth.1785.

ROACH, JaredC. et al. (2011). "Chromosomal Haplotypes by Genetic Phasing of Human Families". In : *The American Journal of Human Genetics* 89.3, 382–397. ISSN : 0002-9297. DOI : 10.1016/j.ajhg.2011.07.023. URL : http://dx.doi.org/10.1016/j.ajhg.2011.07.023.

HERT, Daniel G., Christopher P. FREDLAKE et Annelise E. BARRON (2008). "Advantages and limitations of next-generation sequencing technologies : A comparison of electrophoresis and non-electrophoresis methods". In : *ELECTROPHORESIS* 29.23, 4618–4626. ISSN : 1522-2683. DOI : 10.1002/elps.200800456. URL : http://dx.doi.org/10.1002/elps.200800456.

CLARK, A.G. (1990). "Inference of Haplotypes from PCR-amplified Samples of Diploid Populations". In : 7, p. 111–22.

HUBBEL, E. (2000). "Finding a Maximum Parsimony Solution to Haplotype Phase Is NP-Hard". In :

BERTOLAZZI, Paola et al. (2008). "Solving haplotyping inference parsimony problem using a new basic polynomial formulation". In : *Computers Mathematics with Applications* 55.5, 900–911. ISSN : 0898-1221. DOI : 10.1016/j.camwa.2006.12.095. URL : http://dx.doi.org/10.1016/j.camwa.2006.12.095.

BROWN, Daniel G. et Ian M. HARROWER (2004). "A New Integer Programming Formulation for the Pure Parsimony Problem in Haplotype Analysis". In : *Algorithms in Bioinformatics*, 254–265. ISSN : 1611-3349. DOI : 10.1007/978-3-540-30219-3_22. URL : http://dx.doi.org/10.1007/978-3-540-30219-3_22.

WANG, L. et Y. XU (2003). "Haplotype inference by maximum parsimony". In : *Bioinformatics* 19.14, 1773–1780. ISSN : 1460-2059. DOI : 10.1093/bioinformatics/btg239. URL : http://dx.doi.org/10.1093/bioinformatics/btg239.

WANG, I-Lin et Hui-E YANG (2011). "Haplotyping populations by pure parsimony based on compatible genotypes and greedy heuristics". In : *Applied Mathematics and Computation* 217.23, 9798–9809. ISSN : 0096-3003. DOI : 10.1016/j.amc.2011.04.073. URL : http://dx.doi.org/10.1016/j.amc.2011.04.073.

EXCOFFIER, L. et M. SLATKIN (1995). In : *Molecular Biology and Evolution*. DOI : 10.1093/oxfordjournals.molbev.a040269. URL : http://dx.doi.org/10.1093/oxfordjournals.molbev.a040269.

QIN, Zhaohui S., Tianhua NIU et Jun S. LIU (2002). "Partition-Ligation–Expectation-Maximization Algorithm for Haplotype Inference with Single-Nucleotide Polymorphisms". In : *The American Journal of Human Genetics* 71.5, 1242–1247. ISSN : 0002-9297. DOI : 10.1086/344207. URL : http://dx.doi.org/10.1086/344207.

STEPHENS, Matthew, Nicholas J. SMITH et Peter DONNELLY (2001). "A New Statistical Method for Haplotype Reconstruction from Population Data". In : *The American Journal of Human Genetics* 68.4, 978–989. ISSN : 0002-9297. DOI : 10.1086/319501. URL : http://dx.doi.org/10.1086/319501.

Halperin, E. et E. Eskin (2004). “Haplotype reconstruction from genotype data using Imperfect Phylogeny”. In : *Bioinformatics* 20.12, 1842–1849. issn : 1460-2059. doi : 10.1093/bioinformatics/bth149. url : http://dx.doi.org/10.1093/bioinformatics/bth149.

Howie, Bryan N., Peter Donnelly et Jonathan Marchini (2009). “A Flexible and Accurate Genotype Imputation Method for the Next Generation of Genome-Wide Association Studies”. In : *PLoS Genetics* 5.6. Sous la dir. de Nicholas J.Editor Schork, e1000529. issn : 1553-7404. doi : 10.1371/journal.pgen.1000529. url : http://dx.doi.org/10.1371/journal.pgen.1000529.

Arivazhagan, N., H. J. Kim et E. Yuan (2015). *Local Ancestry Inference in Admixed Populations*. url : http://cs229.stanford.edu/proj2015/290_report.pdf.

Pedregosa, F. et al. (2011). “Scikit-learn : Machine Learning in Python”. In : *Journal of Machine Learning Research* 12, p. 2825–2830.

Boser, Bernhard E., Isabelle M. Guyon et Vladimir N. Vapnik (1992). “A training algorithm for optimal margin classifiers”. In : *Proceedings of the fifth annual workshop on Computational learning theory - COLT ’92*. doi : 10.1145/130385.130401. url : http://dx.doi.org/10.1145/130385.130401.

Burrows, M. et D. J. Wheeler (1994). *A block-sorting lossless data compression algorithm*. Rapp. tech.

Langmead, Ben et al. (2009). “Ultrafast and memory-efficient alignment of short DNA sequences to the human genome”. In : *Genome Biology* 10.3, R25. issn : 1465-6906. doi : 10.1186/gb-2009-10-3-r25. url : http://dx.doi.org/10.1186/gb-2009-10-3-r25.

Durbin, R. (2014). “Efficient haplotype matching and storage using the positional Burrows-Wheeler transform (PBWT)”. In : *Bioinformatics* 30.9, 1266–1272. issn : 1460-2059. doi : 10.1093/bioinformatics/btu014. url : http://dx.doi.org/10.1093/bioinformatics/btu014.

Lunter, Gerton (2016). “Fast haplotype matching in very large cohorts using the Li and Stephens model”. In : *bioRxiv*. doi : 10.1101/048280. eprint : https://www.biorxiv.org/content/early/2016/04/12/048280.full.pdf. url : https://www.biorxiv.org/content/early/2016/04/12/048280.

Loh, Po-Ru et al. (2016). “Reference-based phasing using the Haplotype Reference Consortium panel”. In : *Nature Genetics* 48.11, 1443–1448. issn : 1546-1718. doi : 10.1038/ng.3679. url : http://dx.doi.org/10.1038/ng.3679.

Liste des travaux

Articles

Dias Alves, Thomas, Julien Mairal et Michael G. B. Blum (in prep.). "Loter : A software to infer local ancestry for a wide species."
Luu, Keurcien, **Thomas Dias Alves** et Michael G. B. Blum (in prep.). "Scanning genomes for adaptive introgression using principal component analysis."

Conférences et Posters

Dias Alves, Thomas, Julien Mairal, Michael G. B. Blum (2016). "A new haplotype phasing algorithm for large genomic data." *RECOMB Genetics*.
Dias Alves, Thomas, Michael G. B. Blum, Julien Mairal (2016). "Une méthode d'optimisation pour la reconstruction des haplotypes" *JOBIM 2016*.

Site internet

2015 - Michael G. B. Blum, Kévin Caye, **Thomas Dias Alves**, Keurcien Luu[2] (ordre alphabétique). "Software and Statistical Methods for Population Genetics"

Logiciels

`Loter` : Logiciel d'inférence des coefficients de métissage locaux et de reconstruction des haplotypes (*Python package*).
`aede` : Logiciel de simulation de population métisse selon le modèle présenté par HAPMIX (Price et al., 2009) et généralisé à plusieurs populations (*Python package*).
`loc` : Un module Python pour généraliser le graphe de `Loter` mais sans la rapidité de calcul.

2. https://ssmpg-challenge.imag.fr/home/credits/

Annexe A

Loter: A software to infer local ancestry for a wide range of species

Thomas Dias-Alves ,[1] Julien Mairal,[2] Michael G.B. Blum,[*,1]

[1] Univ. Grenoble Alpes, CNRS, TIMC-IMAG UMR 5525, France

[2] Univ. Grenoble Alpes, Inria Grenoble, Team Thoth, LJK UMR 5224, France

Running Head: A software to infer local ancestry

Keywords: Corresponding author: Michael Blum

Laboratoire TIMC-IMAG, Faculté de Médecine, 38706 La Tronche, France

Phone +33 4 56 52 00 65

Email: michael.blum@univ-grenoble-alpes.fr

Abstract

Admixture between populations provides opportunity to study biological adaptation and phenotypic variation. Studies of admixture processes and admixture mapping rely on local ancestry inference for admixed individuals, which consists of computing at each loci the number of copies that originate from ancestral source populations. Various software exist for local ancestry inference and they are tuned to provide accurate results on human data. Here, we introduce the software Loter that does not require any biological parameters besides haplotype data in order to make local ancestry inference available for a wide range of species. Using simulations of admixture based on human and Populus haplotypes, we compared the performance of Loter to HAPMIX, LAMP-LD and of RFMIX. HAPMIX is the only software severely impacted by imperfect haplotype reconstruction. LAMP- LD and RFMIX provide very accurate results for recent admixture time typically smaller than 20 generations. However, when admixture become more ancient than 150 generations with simulated human data, Loter is the most accurate software and it is less impacted by increasing admixture time than LAMP-LD and RFMIX. For simulations of Populus admixture, both Loter and LAMP-LD are robust to increasing admixture times by contrast to RFMIX. When comparing length of ancestry tract, Loter and LAMP-LD provide results whose accuracy is again more robust to increasing admixture times. We applied Loter to admixed individuals that result from admixture between two Populus species and we used length of ancestry tracts to find that admixture took place around 100 generations ago.

Introduction

Admixture or hybridization between populations is a natural experiment that provides opportunity to map genomic regions involved in phenotypic variation and biological adaptation (Buerkle and Lexer 2008; Payseur and Rieseberg 2016). Mapping can rely on Local Ancestry Inference (LAI) of admixed individuals, which consists of computing at a given locus the number of copies that originates from the ancestral source populations. LAI makes use of haplotypes from putative source populations and process haplotypes or genotypes from admixed population to infer local ancestry of admixed individuals. Figure 1 shows local ancestry of 4 simulated Populus individuals resulting from admixture between 2 Populus species (Suarez-Gonzalez et al. 2016). Sequence and dense genotype data are now generated for a wide range of species besides humans for which LAI is relevant. LAI can be used to study patterns of introgression (Hufford et al. 2013; Suarez-Gonzalez et al. 2016; Medugorac et al. 2017), to map genes involved in reproductive isolation (Corbett-Detig and Nielsen 2017) and phenotypic variation (Lindtke et al. 2013; vonHoldt et al. 2016), and to decipher past admixture processes (Brandvain et al. 2014; Liu et al. 2014). Although various LAI software have been developed, they have been mainly tuned to human data set in order to map disease-associated variants (Patterson et al. 2004; Seldin et al. 2011).

We introduce the software Loter for Local Ancestry Inference, which does not require specifications of statistical or biological parameters in order to make LAI inference available for a wide range of species. Several software for LAI have been developed including HAPMIX, LAMP-LD and RFMIX (Price et al. 2009; Baran et al. 2012; Maples et al. 2013). However, they require various parameters to be specified, which can hamper practical use of LAI software. HAPMIX requires specifications of several biological parameters that might be difficult to obtain such as genetic map, recombination and mutation rate, average ancestry coefficient, and the average number of generations since admixture (Price et al. 2009). LAMP-LD requires a physical map and statistical parameters, which are the number of hidden states in the hidden Markov Model and a window size where local ancestry is assumed to be constant (Baran et al. 2012). Default parameter values can be provided when using LAMP-LD. RFMIX requires statistical and biological parameters, which are a genetic map, a window size (in centimorgan) where local ancestry is assumed to be constant and the average number of generations since admixture (Maples et al. 2013). Except for the genetic map, default parameters values can also be provided when using RFMIX (Maples et al. 2013). There are other differences between LAI software that are provided in Table 1. Except for LAMP-LD that uses statistical parameters only, RFMIX and HAPMIX require biological information such as genetic map, which can be difficult to provide for non-model species.

The software Loter is based on a modeling principle similar to HAPMIX and that was originally introduced by Li and Stephens (2003). The copying model assumes that given a collection of 'parental' haplotypes from the putative source populations, haplotypes from admixed individuals are modeled as a mosaic of existing parental haplotypes (Price et al. 2009) (Figure 2). The main difference with HAPMIX is that Loter is not based on a probabilistic formula-

tion, which requires several parameters to be specified. Instead, Loter is based on an optimization problem parametrized with a single regularization value λ that penalizes switches between parental haplotypes. Solutions of the optimization problem are found using dynamic programming, which is linear with respect to the number of markers and the number of individuals from the source populations. Inference of local ancestry is based on a combination of bagging and of model averaging where the optimal solution is found by averaging results obtained for different values of the regularization parameter λ.

We compare Loter to HAPMIX, LAMP-LD and RFMIX using diploid accuracy, which is an error measure analogous to imputation error for LAI (Sankararaman et al. 2008). We consider the example of admixture of two Populus species in North America to simulate admixed individuals (Figure 1) (Suarez-Gonzalez et al. 2016). We evaluate to what extent diploid accuracy of LAI software different is affected by the number of generations since admixture. We additionally evaluate to what extent length of ancestry tracts are accurately inferred by the different LAI software. Lengths of ancestry tracts is a biological information that is used to date and reconstruct admixture events and that should consequently be accurately inferred for reliable demographic reconstruction (Gravel 2012; Ni et al. 2016; Corbett-Detig and Nielsen 2017; Xue et al. 2017). We repeat the same admixture experiment using human genotypes from HAPMAP 3 where we consider admixture between Europeans (CEU) and Africans (YRI) (International HapMap 3 Consortium et al. 2010). We additionally consider a 3-way admixture scenario between Chinese (CHB), Europeans (CEU) and Africans (YRI) from HAPMAP 3. Finally we apply Loter to admixed sequenced Populus individuals. Based on genome resequencing data from chromosome 6, we estimate admixture time using length of reconstructed ancestry tracts.

New Approaches

We describe the optimization problem, which accounts that haplotypes from admixed individuals are described as a mosaic of haplotypes originating from the source populations (Figure 2). We assume that there are n individuals in the source populations resulting in $2n$ haplotypes denoted by $(H_1, \cdots, H_{2n})$. The value (0 or 1) of the i^{th} haplotype at the j^{th} SNP is denoted H_i^j. Haplotypes can be obtained from genotypes using computational phasing software such as fastPHASE or Beagle (Scheet and Stephens 2006; Browning and Browning 2007). A vector $(s_1, \ldots, s_p)$ describes the sequence of haplotype labels from which the haplotype h of an admixed individual can be approximated (Figure 2). For the j^{th} SNP in the data set, $s_j = k$ if haplotype h results from a copy of haplotype H_k. The optimization problem consists of minimizing the following cost function

$$C(s_1, \ldots, s_p) = \sum_{j=1}^{p} |h^j - H_{s_j}^j| + \lambda \sum_{j=1}^{p-1} \mathbb{1}_{s_j \neq s_{j+1}}, \quad (1)$$

where $(s_1, \ldots, s_p) \in \{1, \cdots, 2n\}^p$. The first term in equation (1) is a loss function that is a sum over loci of a $\{0, 1\}$-valued function equals to 1 if haplotype h is different from the copied haplotype and to 0 otherwise. The second term is a regularization term that is equal to the regularization parameter λ times the number of switches between parental haplotypes. A solution to minimize equation (1) can be found using dynamic programming and is provided in the Materials and Methods section. Once a solution has been provided about the sequence $(s_1, \ldots, s_p)$ of parental haplotypes, local ancestry values can be deduced automatically from this sequence because each parental haplotype belong to one of the source populations (Figure 2). The formulation described in equation (1) is valid for $K = 2$ or more ancestral source populations.

The optimization problem described in equation (1) is parametrized by a regularization parameter λ. Large values of λ strongly penalize switches between parental haplotypes such that solutions have long chunks of constant values of local ancestry. To avoid the difficult choice of λ, solutions for local ancestry are averaged over different values of λ. For each value of λ, we consider a bagging technique where 20 different solutions are found based on 20 different datasets generated using bootstrap with resampling (Breiman 1996). We consider $\lambda = 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5$ resulting in $160 = 8 \times 20$ different solutions and the final choice for local ancestry is obtained using a majority rule. When the most frequent vote has less that 75% of the votes, ancestry is imputed using local ancestry values of the closest SNPs with a preference for the SNP on the left in case of ambiguity. Finally, an additional smoothing procedure is considered in order to account for switch errors when phasing admixed individuals (see Materials and Methods).

Results

Simulated human admixed individuals

We consider simulated admixed individuals resulting from admixture between Africans (YRI population in HAPMAP 3) and Europeans (CEU population in HAPMAP 3). Accuracy obtained with several LAI software varies depending on time since admixture occurs (Figure 3). For recent admixture where admixture occurred 5 generations ago, LAMP-LD and RFMix obtain the best result with median diploid accuracies of 99.6% and 99.8% whereas median diploid accuracy of Loter is equal to 99.3%. For ancient admixture where admixture occurred 500 generations ago, Loter obtains the largest median diploid accuracy of 86.7% followed by LAMP-LD with a diploid accuracy of 80.6% and RFMix with a diploid accuracy of 72.0%. For the smallest admixture times ($\mu \leq 20$ generations), diploid accuracy obtained with the three software is larger than 95% and RFMix and LAMP-LD outperform LOTER. By contrast, for the largest admixture times ($\mu \geq 150$ generations), Loter is the most accurate software for LAI (Figure 3).

We evaluate the benefit of bagging and of averaging local ancestry values obtained with different values of the regularization parameter, which are implemented by default in Loter. First, the choice of the regularization parameter

matters since diploid accuracy depends on the choice of the regularization parameter (Supplementary Figure SI1). As expected, smallest values of λ provide the best result for ancient admixture event; $\lambda = 2$ is optimal when $\mu = 400$ and $\mu = 500$ generations, and $\lambda = 5$ is optimal otherwise. Second, for all values of the admixture times, averaging local ancestry values obtained with different values of the regularization parameter λ instead of considering a single value improves inference (Supplementary Figure SI1). Last, averaging over bootstrap replicates (bagging) further improve local ancestry inference (Supplementary Figure SI1).

We additionally evaluate the diploid accuracy of HAPMIX, which is another LAI software based on the copying model. Compared to the three other LAI software, its diploid accuracy is the smallest with values of diploid accuracy ranging from 42% to 57% (Supplementary Figure SI2). To identify the main factor that determines HAPMIX diploid accuracy, we consider several sets of haplotypes from the source populations to perform LAI. The fact that haplotypes used for LAI are not exact but can be computationally phased using Beagle causes the severe reduction of diploid accuracy obtained with HAPMIX (Supplementary Figure SI3). When considering haplotypes reconstructed with Beagle instead of true haplotypes, diploid accuracy was reduced by an average of 32%.

Additionally, we compare diploid accuracy of Loter, RFMix and LAMP-LD on data simulated under a 3-way admixture model (Supplementary Figure SI4). As for 2-way admixture model, we find that RFMix and LAMP-LD have larger diploid accuracies than Loter for recent admixture and smaller ones for ancient admixture. When admixture took place 5 generations ago, the diploid accuracy of RFMix is of 99.8%, the accuracy of LAMP-LD is of 99.8%, and the accuracy of Loter is of 99.5%. By contrast, when admixture took place 500 generations ago, the diploid accuracy of RFMix is of 84.6%, the accuracy of LAMP-LD is of 89.1%, and the accuracy of Loter is of 92.3%.

Simulated Populus admixed individuals

We simulate individuals that result from admixture between *Populus trichocarpa*, which is adapted to relatively humid, moist, and mild conditions west of the Rocky Mountains and *Populus balsamifera*, which is a boreal species (Suarez-Gonzalez et al. 2016). As for human data, we compare Loter, RFMix, and LAMP-LD using diploid accuracy as a criterion for comparison. Again, the diploid accuracy of RFMix decreases with increasing admixture time. It ranges from a diploid accuracy of 92.0% when admixture occurs 5 generations ago to a diploid accuracy of 65.3% when admixture occurs 500 generations. By contrast to the simulations of human data, we find that the diploid accuracy of Loter and of LAMP-LD does not change with admixture time. For LAMP-LD, diploid accuracies ranges from 91.3% to 90.1% and for Loter it ranges from 89.9% to 89.0% when admixture increases from 5 generations to 500 generations.

Length of ancestry tracts

For simulated admixed Populus individuals, we additionally compare the length of *P. balsamifera* reconstructed ancestry tracts to the length of true ancestry tracts (Figure 5). When admixture took place 10 generations ago in Populus, RFMIX provides a distribution of ancestry tracts that is closer to the true distribution. For Populus simulations, the true median length of ancestry tract is of 4.26 cM and RFMIX finds 5.20 cM whereas LAMP-LD and Loter find median lengths of 0.05 and 2.31 cM respectively. Both LAMP-LD and Loter return spurious chunks of local ancestry that are of small lengths and that contribute to reduce the mean length of local ancestry (SI Figure SI6). Additionally, several long blocks of ancestry chunks are cut into smaller pieces when using Loter or LAMP-LD (Supplementary Figure **??**). When admixture took place 10 generations ago in the human simulations, both Loter and RFMIX provide the most accurate results; the true median length of ancestry tract is of 9.03 cM and RFMIX, Loter, and LAMP-LD reconstruct ancestry tracts of median length 9.99 cM, 12.20 cM and 9.02 cM respectively.

When admixture took place 200 or of 500 generations, length of ancestry chunks are more accurately reconstructed with Loter and with Lamp-LD than with RFMIX (Figure 5). For both human and Populus simulated data, RFMix, by contrast to Loter and LAMP-LD, reconstructs ancestry tracts that are too long compared to true ancestry tracts when admixture is larger or equal than 200 generations. When admixture took place 200 generations ago, true median ancestry tracts is of 1 cM or less whereas RFMix reconstructs tracts of 2 cM or more. For Populus simulations, the true median length of ancestry tract is of 0.45 cM when admixture took place 200 generations ago, and RFMIX, Loter, and LAMP-LD find respectively 2.00 cM, 0.56 cM and 0.30 cM (Supplementary Figure SI6). When admixture took place 500 generations ago, the true median length of ancestry tract is of 0.17 cM , and RFMIX, Loter, and LAMP-LD find respectively 1.60 cM, 0.33 cM and 0.17 cM.

Application of Loter to admixed Populus individuals

We applied Loter to 36 individuals that are admixed between *P. balsamifera* and *P. trichocarpa*. When averaging local ancestry coefficients, we find that admixed individuals have on average 87% of *P. trichocarpa* ancestry and 13% of *P. balsamifera* ancestry . We find that the median length of *P. balsamifera* ancestry tracts is equal to 0.76 cM and the first and third quartiles are equal to 0.25 cM and 1.47 cM. We also perform simulations of admixed individuals based on true genotypes from *P. balsamifera* and *trichocarpa* individuals. When admixture time varies from 10 to 500 generations; median *P. balsamifera* ancestry tracts vary from 2.3 cM to 0.32 cM, the first quartile varies from 0.28 cM to 0.19 cM, and the third quartile varies from 4.18 cM to 0.55 cM (Figure 6). Of the six admixture times we considered ($\mu \in \{10, 50, 100, 200, 300, 500\}$) in the simulations, we find that $\mu = 100$ generations provide the most similar distribution of *P. balsamifera* ancestry tracts; when $\mu = 100$ generations, the three quartiles of *P. balsamifera* ancestry tracts are equal to 0.21 cM, 0.78 cM, and 1.29 cM.

Discussion

As dense genotype or sequencing data become more affordable, local ancestry inference provides an opportunity for admixture mapping and for deciphering admixture processes as well. We have introduced the software Loter in order to make local ancestry available for a wide range of species for which biological parameters such as admixture times or recombination rates are not available. The regularization parameter λ, which controls smoothing, depends in a complicated manner on several biological and statistical parameters including mutation rate, recombination rates. To avoid the difficult choice of the regularization parameter, Loter implements an averaging procedure where we average solutions for different values of the regularization parameter. Because of model averaging, Loter does not require parameter tuning which can make it easy to apply from a users's point of view.

We compared Loter to other local ancestry software: HAPMIX, RFMIX and LAMP-LD. We found that the diploid accuracy obtained with HAPMIX is reduced by 32% in average when haplotypes are not known perfectly using trio-phasing but only reconstructed using phasing software such as Beagle. By contrast, RFMIX, LAMP-LD, and Loter are robust to imperfect haplotype reconstruction and that is the reason why only RFMIX, LAMP-LD, and Loter were further considered in software comparisons. When admixture took place 5 generations ago, RFMIX and of LAMP-LD provide the largest diploid accuracies but all three software were found to provide an accurate reconstruction of local ancestry with diploid accuracies always larger than 99% for the simulations of Afro-American admixed haplotypes and larger than 89.9% for the simulations of Populus individuals. Compared to RFMIX and LAMP-LD, results obtained with Loter are more robust with respect to the time since admixture occurred. For simulated human data, the diploid accuracy of Loter for human data decreases to 87% for the most ancient admixture times of 500 generations whereas it decreases to 72% and 81% when using RFMIX and LAMP-LD (Thomas to complete). For Populus data, diploid accuracy does not depend on admixture times when using Loter and LAMP-LD whereas it is severely impacted when using RFMIX. The fact that diploid accuracy is not impacted by the considered range of admixture times is encouraging and suggests that local ancestry inference is possible for admixture that occurred hundreds of generation ago when SNP density is large enough as for Populus data; the mean distance between two SNPs is of $9.8.10^{-6}$ cM for Populus data whereas it is of $2.7.10^{-3}$ cM for the human data.

The reason why diploid accuracy may be impacted by admixture time is related to the statistical smoothing procedure. For instance, RFMIX has been tuned to genotypes resulting from recent admixture that occurred 10 generations ago or less such as admixture between African and Europeans (Gravel 2012; Bryc et al. 2015). When using default parameters of RFMIX for data resulting from ancient admixture events, reconstructed ancestry tracts are inadequately long (Figure 5 and Supplementary Figure SI6) and over-smoothing can affect diploid accuracy. Even when providing true admixture time to RFMIX, diploid accuracy of RFMIX remains more impacted by admixture time possibly because of the default of 0.2 cM for window size (Supplementary Figure SI7). When using HAPMIX, a choice of

window lengths or of the time since admixture should also be provided However, choice of admixture time can be very difficult and impacts biological results. For instance, the length of ancestry tracts found with HAPMIX depends on the choice on the provided value for admixture time (Patin et al. 2014). The model averaging procedure implemented in Loter has the advantage to avoid to put a strong prior on a particular length of ancestry tract. In addition, model averaging improves parameter inference, which has already been observed when phasing genotypes using a statistical model of Linkage Disequilibrium (Scheet and Stephens 2006).

The simulation results show that accuracy obtained with Loter and LAMP-LD is less sensitive to admixture times compared to the accuracy obtained with RFMix. LAMP-LD is more accurate that Loter for recent admixture times when using human data and for all values of admixture times when considering the Populus data. However, LAMP-LD has limitations for large-scale NGS data that contains a large number of molecular markers. It is limited to run on 50,000 SNPS and it can be computer intensive. To perform local ancestry inference for 500,000 SNPs and 20 admixed Populus individuals, the running time is of 28 minutes using RFMIX, 58 minutes with LAMP-LD and of 6 minutes using LAMP-LD when using 20 Intel Xeon processors of 2.40 GhZ. However, although there are differences between local ancestry inference, using different local ancestry inference software can be a wise strategy to provide evidence for an association or a selection signal (Zhou et al. 2016). We expect that providing a parameter-free and rapid software for local ancestry inference will make more accessible genomic studies about admixture processes.

Materials and Methods

Dynamic Programming

The optimization problem of equation (1) is solved using dynamic programming.The solution of the problem with p SNPs can be derived from the solution with $(p-1)$ SNPs. Two configurations are possible. Either the admixed haplotype copies from the same haplotype at the $(p-1)^{\text{th}}$ and p^{th} SNP and

$$C(s_1, \ldots, s_p) = C(s_1, \ldots, s_{p-1}) + |h^p - H^p_{s_{p-1}}|, \tag{2}$$

or it uses different template haplotypes and

$$C(s_1, \ldots, s_p) = C(s_1, \ldots, s_{p-1}) + |h^p - H^p_{s_p}| + \lambda. \tag{3}$$

The optimal solution is then found by computing the shortest path on a graph displayed in Figure SI5. To find the shortest path, dynamic programming computes at each node a quantity $Q(i,j)$ that corresponds to the optimal solution for the first j SNPs and when the template haplotype at SNP j is the i^{th} haplotype $s_j = i$. The quantity $Q(i,j)$ is

updated as followed

$$Q(i,j) = \min\left(Q(i,j-1), \min_{i' \in \{1,\ldots,n\}}\{Q(i',j-1)\} + \lambda\right) + |h^j - H_i^j| \quad (4)$$

Because we store the value of $\min_{i' \in \{1,\ldots,n\}}\{Q(i',j-1)\}$ at locus $j-1$, the value of $Q(i,j)$ can be computed as a minimum between 2 values. For each admixed haplotype, the complexity of this algorithm is therefore $\mathcal{O}(n \times p)$ where n is the number of individuals in the ancestral populations and p is the number of SNPs. The path $(s_1, \ldots, s_p)$ is then converted to an haploid ancestry sequence $a = (a_1, \ldots, a_p) \in (1, \ldots, K)^p$ where a_j is the population of origin of the s_j^{th} haplotype.

Accounting for phase errors

Reconstructed haplotypes from an admixed population may contain switch errors (Browning and Browning 2011). As considered in RFMix, it is possible to redistribute ancestry chunks among the two haplotypes from the same individual in order to correct for switch errors. For now, the software Loter accounts for phase error when there are 2 ancestral populations only. Once local ancestry values for each of the 2 haplotypes have been found after solving equation (1), we compute the sum of the 2 haplotypic local ancestries resulting in diploid local ancestry values $d = (d^1, \ldots, d^p) \in \{0,1,2\}^p$ as returned by the software HAPMIX. Local ancestry values are then reconstructed using an internal *ancestry phasing* algorithm. The phasing procedure considers that the 2 haplotypic local ancestry sequences are a mosaic of two possible ancestry sequences $A_1 = (0, \ldots, 0)$ and $A_2 = (1, \ldots, 1)$ corresponding to the two possible ancestral populations. Two vectors $(s_1, \ldots, s_p) \in \{0,1\}^p$ and $(s'_1, \ldots, s'_p) \in \{0,1\}^p$ describe the sequence of labels, $a \in \{0,1\}^p$ and $a' \in \{0,1\}^p$ are the haploid local ancestry values, and Θ is a compound parameter equal to $(s_1, \ldots, s_p, s'_1, \ldots, s'_p, a, a')$. The ancestry phasing algorithm consists of minimizing the following cost function

$$C'(\Theta) = \sum_{j=1}^{p} |a^j - A^j_{s_j}| + \sum_{j=1}^{p} |a'^j - A^j_{s_j}| + \lambda \sum_{j=1}^{p-1} \mathbb{1}_{s_j \neq s_{j+1}} + \lambda \sum_{j=1}^{p-1} \mathbb{1}_{s'_j \neq s'_{j+1}}, \quad (5)$$

subject to the constraint that the sum of the haploid local ancestries a and a' is equal to the diploid local ancestry d. The solution for Θ is found using dynamic programming. For each admixed individual, the complexity of this algorithm is $\mathcal{O}(p)$. Once a solution has been found, haplotypic local ancestry, which has been corrected for phasing errors, consists of the two sequences $(A^1_{s_1}, \ldots, A^p_{s_p})$ and $(A^1_{s'_1}, \ldots, A^p_{s'_p})$.

Admixture between Populus species

We simulate admixed individuals by constructing their genomes from a mosaic of real *P. balsamifera* and *P. trichocarpa* individuals (Suarez-Gonzalez et al. 2016). We consider a probabilistic model that has been used to simulate admixed

individuals and to evaluate the performances of HAPMIX (Price et al. 2009). Simulations are based on 20 haplotypes (first 50,000 SNPs) from chromosome 6 from the species *P. balsamifera* (balsam poplar) and of 20 haplotypes from the species *P. trichocarpa* (black cottonwood) (Suarez-Gonzalez et al. 2016). Haplotypes were obtained from genotypes using Beagle (Browning and Browning 2007). The *P. trichocarpa* ancestry α_i of a simulated admixed individual is drawn randomly according to a Beta distribution of mean 0.8 and of variance 0.1. At the first marker, the haplotype of an admixed individual i is assumed to originate from *P. trichocarpa* with a probability α_i and from *P. balsamifera* otherwise. For each simulated haplotype, we associate one *P. balsamifera* haplotype and one *P. trichocarpa* haplotype. For a given admixed individual, haplotypes are exclusively copied from these two source haplotypes that are chosen at random. The length (measured in Morgans) of an ancestry chunk is drawn according to an exponential distribution of rate μ generations. In the simulations, we consider values for μ ranging from 5 to 500 generations. The species origin of the new ancestry tract is again determined using the $(\alpha_i, 1-\alpha_i)$ admixture coefficients and the copying process for haplotype is repeated as before. To reconstruct local ancestry of simulated admixed individuals, we consider 30 haplotypes from *P. balsamifera* and 30 haplotypes from *P. trichocarpa* that were not used when simulating admixed individuals. Haplotypes were again phased using the software Beagle. To evaluate diploid accuracy for a given value of μ, we consider 20 sets of simulations consisting of 20 admixed haplotypes each.

Admixture between human populations

When simulating admixed individuals between Yorubans (YRI) and Europeans (CEU) from HAPMAP, we consider the copying process mentioned before. For simulations, we consider true haplotypes based on trio phasing. For inference, we consider haplotypes reconstructed using Beagle based on genotypes that are not used for simulations. A total of 48 Yoruban haplotypes and of 48 European haplotypes are considered to simulate 48 Afro-American haplotypes. We consider 40 European haplotypes and 40 African haplotypes, which were obtained with Beagle, to perform local ancestry inference. To evaluate diploid accuracy for a given value of μ, we consider 20 sets of simulations consisting of 48 admixed haplotypes each.

We consider an additional set of simulations where 3 populations admixed. Admixture is assumed to occur between Chinese (CHB), Europeans (CEU) and Africans (YRI). The exponential distribution for European chunks is of rate $\mu \in \{5, 100, 200, 500\}$ generations and the exponential distribution for African and Asian chunks is equal to $\mu/2$ generations. By contrast to the 2-way admixture models, we use trio-phased haplotypes for inference and not haplotypes reconstructed with Beagle.

Admixed Populus individuals

We consider 36 individuals that are admixed between *P. balsamifera* and *P. trichocarpa* (Suarez-Gonzalez et al. 2016) and use Beagle to phase them. We use the first $500,000$ SNPs of chromosome 6 to reconstruct ancestry tracts. We simulate 16 admixed individuals based on 20 genotypes from *P. balsamifera* and *P. trichocarpa* species. Instead of considering true ancestry ancestry tracts, we rather replicate the same pipeline as for real data such as the bias of ancestry tract reconstruction should be same for data and simulations. We phase simulated individuals using Beagle and reconstruct ancestry tracts using Loter.

Acknowledgments

Authors acknowledge Grenoble Alpes Data Institute, supported by the French National Research Agency under the "Investissements d'avenir" program (ANR-15-IDEX-02) and the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

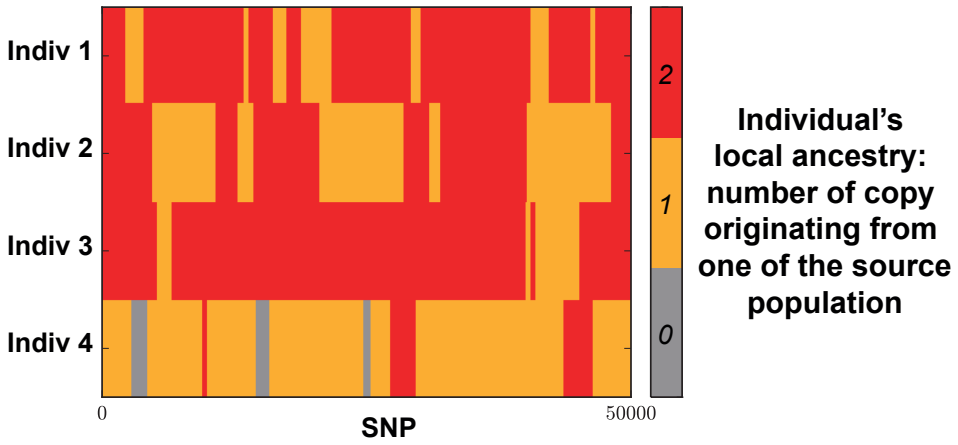

Figure 1: Example of local ancestry decomposition for 4 simulated Populus individuals resulting from admixture between 2 Populus species, which are *Populus trichocarpa* and *Populus balsamifera* (Suarez-Gonzalez et al. 2016). For an admixed individual, local ancestry at a given locus corresponds to the number of copies that has been inherited from the species *P. trichocarpa*. LAI software require haplotypes from putative source populations and process haplotypes or genotypes from admixed population to return local ancestry of admixed individuals. Details of the simulations are described in the Materials and Methods section.

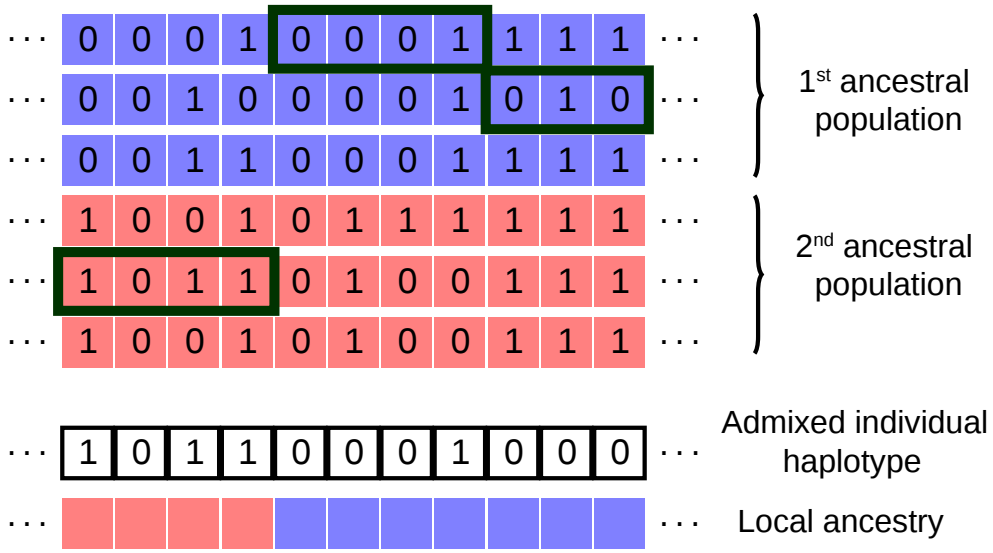

Figure 2: Graphical description of Local Ancestry Inference as implemented in the software Loter. Given a collection of parental haplotypes from the source populations depicted in blue and red, Loter assumes that an haplotype of an admixed individuals is modeled as a mosaic of existing parental haplotypes. In this example, the loss function in equation (1) is equal to 1 because of a single mismatch between parental and admixed haplotype located at the next-to-last position and the regularization term is equal to 2λ because there are 2 switches between parental haplotypes. The displayed solution corresponds to the mathematical solution $(s_1, \ldots, s_{11}) = (5, 5, 5, 5, 1, 1, 1, 1, 2, 2, 2)$ where haplotypes are numbered from top to bottom, and $s_j = k$ if the admixed haplotype results from a copy of the k^{th} parental haplotype at the j^{th} SNP.

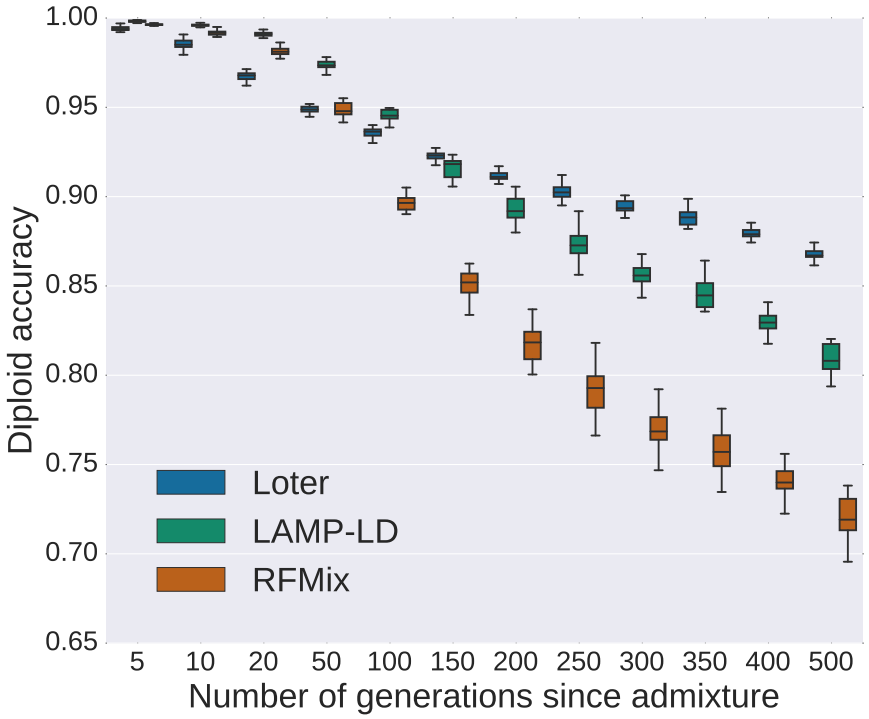

Figure 3: Diploid accuracy obtained with LAMP-LD, Loter, and RFMix for simulated admixed human individuals as a function of the time since admixture occurred. Admixed individuals are simulated by constructing their genomes from a mosaic of true African (YRI) and European (CEU) haplotypes (International HapMap 3 Consortium et al. 2010). For performing simulations, true haplotypes are obtained using trio information. For local ancestry inference, haplotypes are obtained with Beagle using individuals that are not used for simulating admixed individuals. For each value of the number of generations since admixture, 20 sets of 48 admixed individuals are generated. Boxplots show the distribution of the 20 values for the mean diploid accuracy.

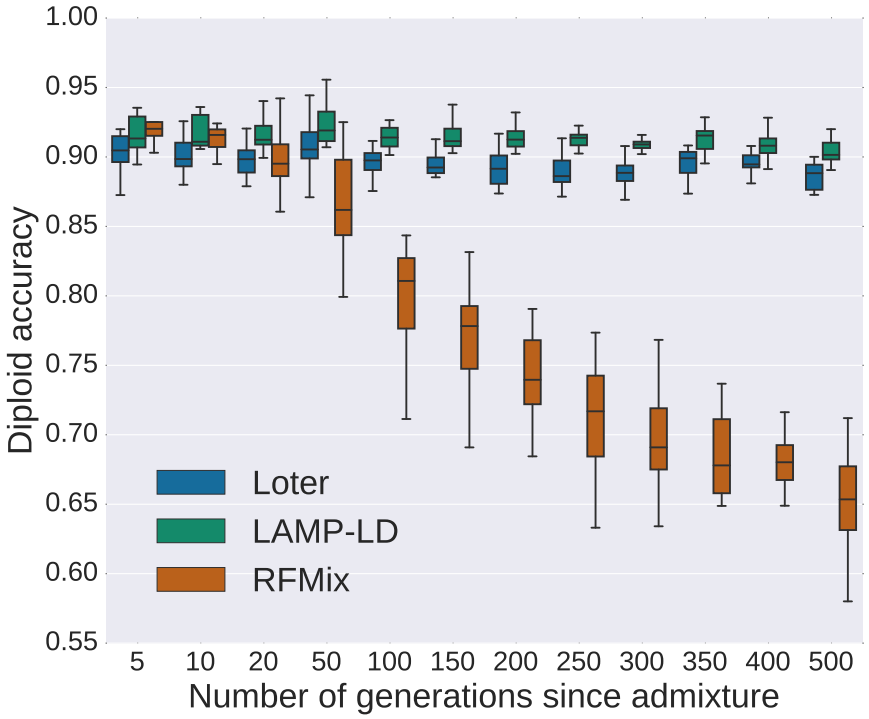

Figure 4: Diploid accuracy obtained with LAMP-LD, Loter, and RFMix for simulated admixed Populus individuals as a function of the time since admixture occurred. Admixed individuals are simulated by constructing their genomes from a mosaic of *Populus trichocarpa* and *Populus balsamifera* individuals. Individuals are phased using Beagle and two different sets of individuals are used for performing simulations and inference. For each value of the number of generations since admixture, 20 sets of 20 admixed individuals are generated. Boxplots show the distribution of the 20 values for the mean diploid accuracy.

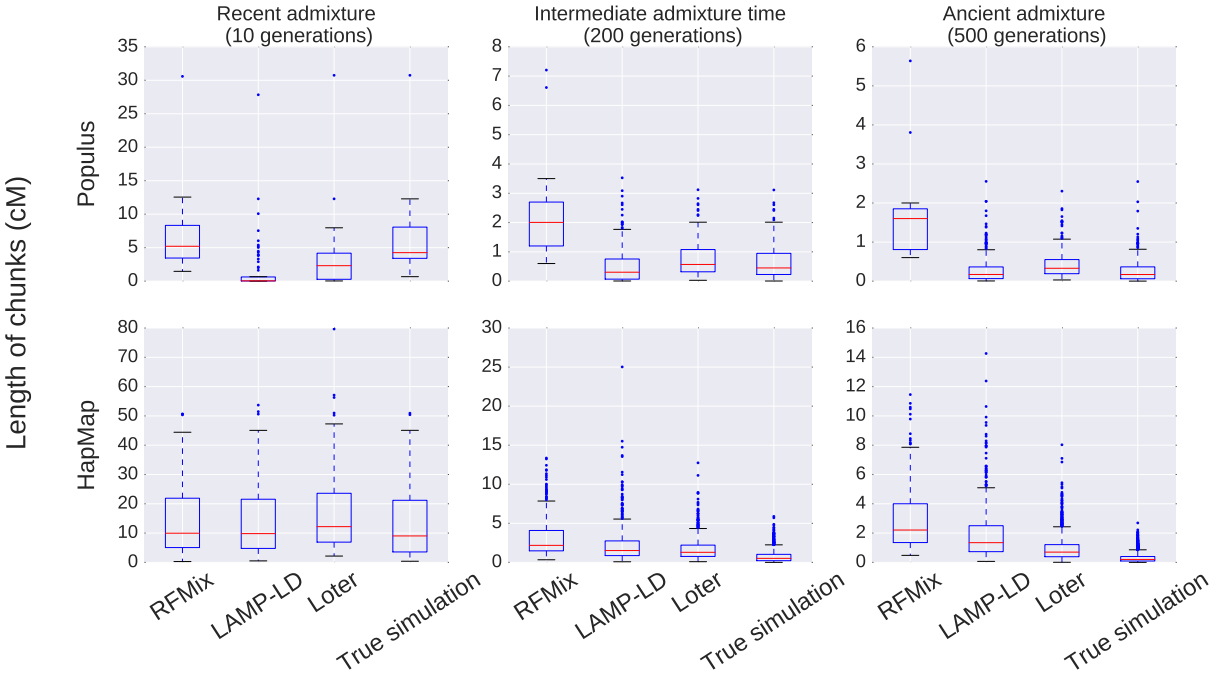

Figure 5: Distribution of the length of ancestry chunks for simulated data. For Populus data, we consider the first 500,000 SNPs of chromosome 6 and for human data, we consider the first 50,000 SNPs of chromosome 1. When considering Populus data, we run 10 times LAMP-LD on non-overlapping sets of SNPs in order to avoid the limitation of 50,000 SNPs of LAMP-LD.

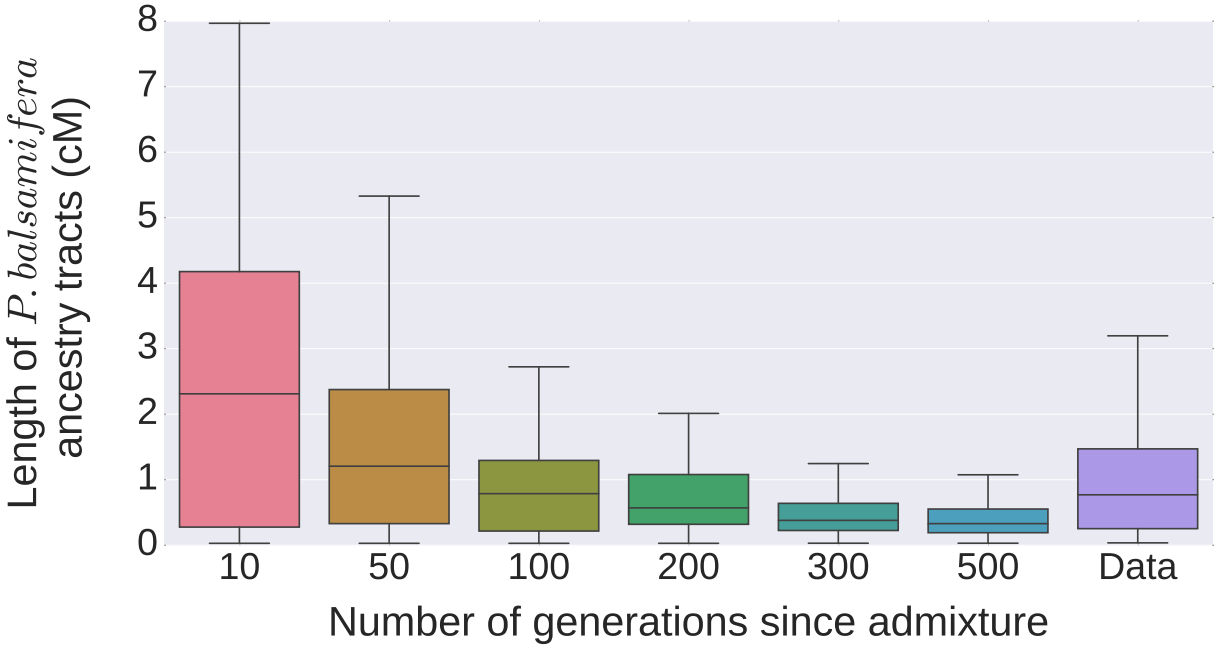

Figure 6: Distribution of the length of *P. balsamifera* ancestry tracts. The data consist of genotypes of admixed individuals between *P. balsamifera* and *P. trichocarpa*. For the simulations, we replicate the same pipeline as for local inference with real data, which consist of using Beagle to phase genotypes and Loter to reconstruct ancestry tracts.

	HAPMIX	LAMP-LD	RFMix	Loter
Phasing of admixed individuals	No	No	Yes	Yes
Number of ancestral pop.	2	$2, 3, 5$	≥ 2	≥ 2
Genetic map required (in cM)	Yes	No (Physical position)	Yes	No (Ordered SNPs)
Limitation of the number of SNPS	No	50,000	No	No
Phasing error correction	No	Not required	Yes	Yes for 2 ancestral pop.
Parallel implementation	No	No	Yes	Yes
Admixture time required	Yes	No	Yes	No
Other biological param. required	Yes	No	No	No

Table 1: Differences between several LAI software. The abbreviation param. stands for parameter and pop. for population. Other biological parameters required by HAPMIX are recombination and mutation rates.

References

Baran, Y., Pasaniuc, B., Sankararaman, S., Torgerson, D. G., Gignoux, C., Eng, C., Rodriguez-Cintron, W., Chapela, R., Ford, J. G., Avila, P. C. et al. 2012. Fast and accurate inference of local ancestry in Latino populations. Bioinformatics **28**:1359–1367.

Brandvain, Y., Kenney, A. M., Flagel, L., Coop, G. and Sweigart, A. L. 2014. Speciation and Introgression between *mimulus nasutus* and *mimulus guttatus*. PLoS Genet **10**:e1004410.

Breiman, L. 1996. Bagging predictors. Machine learning **24**:123–140.

Browning, S. R. and Browning, B. L. 2007. Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. The American Journal of Human Genetics **81**:1084–1097.

———. 2011. Haplotype phasing: existing methods and new developments. Nature Reviews Genetics **12**:703–714.

Bryc, K., Durand, E., Macpherson, J. M., Reich, D. and Mountain, J. 2015. The genetic ancestry of African, Latino, and European Americans across the United States. American Journal of Human Genetics **96**:37–53.

Buerkle, C. A. and Lexer, C. 2008. Admixture as the basis for genetic mapping. Trends in ecology & evolution **23**:686–694.

Corbett-Detig, R. and Nielsen, R. 2017. A hidden Markov model approach for simultaneously estimating local ancestry and admixture time using next generation sequence data in samples of arbitrary ploidy. PLoS genetics **13**:e1006529.

Gravel, S. 2012. Population genetics models of local ancestry. Genetics **191**:607–619.

Hufford, M. B., Lubinksy, P., Pyhäjärvi, T., Devengenzo, M. T., Ellstrand, N. C. and Ross-Ibarra, J. 2013. The genomic signature of crop-wild introgression in maize. PLoS Genet **9**:e1003477.

International HapMap 3 Consortium et al. 2010. Integrating common and rare genetic variation in diverse human populations. Nature **467**:52–58.

Li, N. and Stephens, M. 2003. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. Genetics **165**:2213–2233.

Lindtke, D., Gonzalez-Martinez, S., Macaya-Sanz, D. and Lexer, C. 2013. Admixture mapping of quantitative traits in Populus hybrid zones: power and limitations. Heredity **111**:474–485.

Liu, S., Lorenzen, E. D., Fumagalli, M., Li, B., Harris, K., Xiong, Z., Zhou, L., Korneliussen, T. S., Somel, M., Babbitt, C. et al. 2014. Population genomics reveal recent speciation and rapid evolutionary adaptation in polar bears. Cell **157**:785–794.

Maples, B. K., Gravel, S., Kenny, E. E. and Bustamante, C. D. 2013. RFMix: a discriminative modeling approach for rapid and robust local-ancestry inference. The American Journal of Human Genetics **93**:278–288.

Medugorac, I., Graf, A., Grohs, C., Rothammer, S., Zagdsuren, Y., Gladyr, E., Zinovieva, N., Barbieri, J., Seichter, D., Russ, I. et al. 2017. Whole-genome analysis of introgressive hybridization and characterization of the bovine legacy of Mongolian yaks. Nature Genetics **49**:470–475.

Ni, X., Yang, X., Guo, W., Yuan, K., Zhou, Y., Ma, Z. and Xu, S. 2016. Length distribution of ancestral tracks under a general admixture model and its applications in population history inference. Scientific reports **6**.

Patin, E., Siddle, K. J., Laval, G., Quach, H., Harmant, C., Becker, N., Froment, A., Régnault, B., Lemée, L., Gravel, S. et al. 2014. The impact of agricultural emergence on the genetic history of african rainforest hunter-gatherers and agriculturalists. Nature communications **5**:3163.

Patterson, N., Hattangadi, N., Lane, B., Lohmueller, K. E., Hafler, D. A., Oksenberg, J. R., Hauser, S. L., Smith, M. W., O'Brien, S. J., Altshuler, D. et al. 2004. Methods for high-density admixture mapping of disease genes. The American Journal of Human Genetics **74**:979–1000.

Payseur, B. A. and Rieseberg, L. H. 2016. A genomic perspective on hybridization and speciation. Molecular ecology **25**:2337–2360.

Price, A. L., Tandon, A., Patterson, N., Barnes, K. C., Rafaels, N., Ruczinski, I., Beaty, T. H., Mathias, R., Reich, D. and Myers, S. 2009. Sensitive detection of chromosomal segments of distinct ancestry in admixed populations. PLoS Genet **5**:e1000519.

Sankararaman, S., Sridhar, S., Kimmel, G. and Halperin, E. 2008. Estimating local ancestry in admixed populations. The American Journal of Human Genetics **82**:290–303.

Scheet, P. and Stephens, M. 2006. A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. The American Journal of Human Genetics **78**:629–644.

Seldin, M. F., Pasaniuc, B. and Price, A. L. 2011. New approaches to disease mapping in admixed populations. Nature Reviews Genetics **12**:523–528.

Suarez-Gonzalez, A., Hefer, C. A., Christe, C., Corea, O., Lexer, C., Cronk, Q. C. and Douglas, C. J. 2016. Genomic and functional approaches reveal a case of adaptive introgression from *populus balsamifera* (balsam poplar) in *p. trichocarpa* (black cottonwood). Molecular ecology **25**:2427–2442.

vonHoldt, B. M., Kays, R., Pollinger, J. P. and Wayne, R. K. 2016. Admixture mapping identifies introgressed genomic regions in North American canids. Molecular ecology **25**:2443–2453.

Xue, J., Lencz, T., Darvasi, A., Pe'er, I. and Carmi, S. 2017. The time and place of european admixture in Ashkenazi Jewish history. PLoS genetics **13**:e1006644.

Zhou, Q., Zhao, L. and Guan, Y. 2016. Strong selection at mhc in mexicans since admixture. PLoS genetics **12**:e1005847.

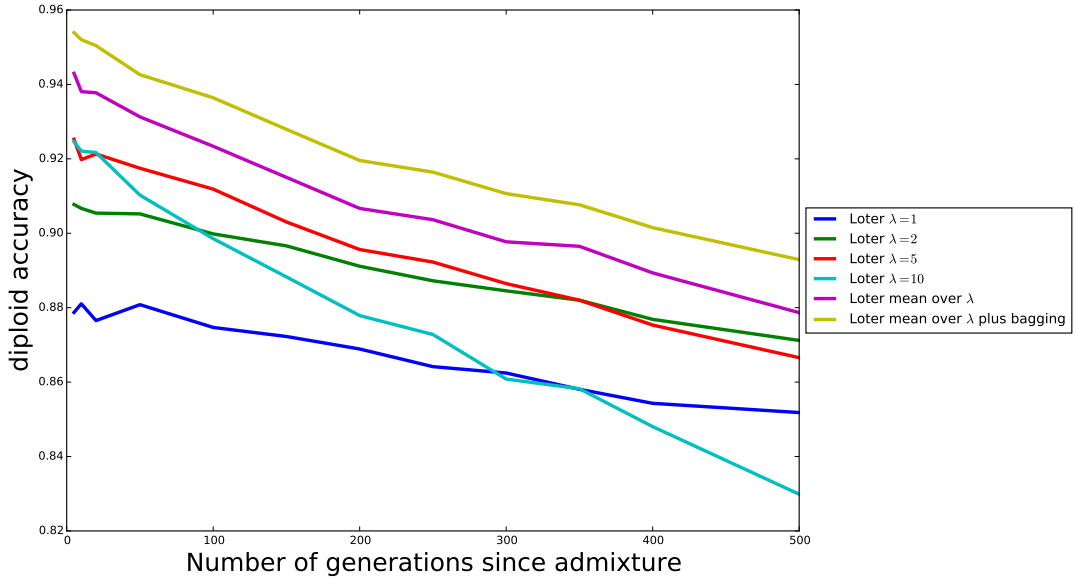

Figure SI1: Diploid accuracy obtained with Loter is improved when bagging and when averaging over multiple values of the regularization parameter. Admixed individuals were simulated by constructing their genomes from a mosaic of true African (YRI) and European (CEU) haplotypes (International HapMap 3 Consortium et al. 2010) (Figure 3). Diploid accuracies are evaluated for twelve different values of the admixture time corresponding to $5, 10, 20, 50, 100, 150, 200, 250, 300, 350, 400$ and 500 generations.

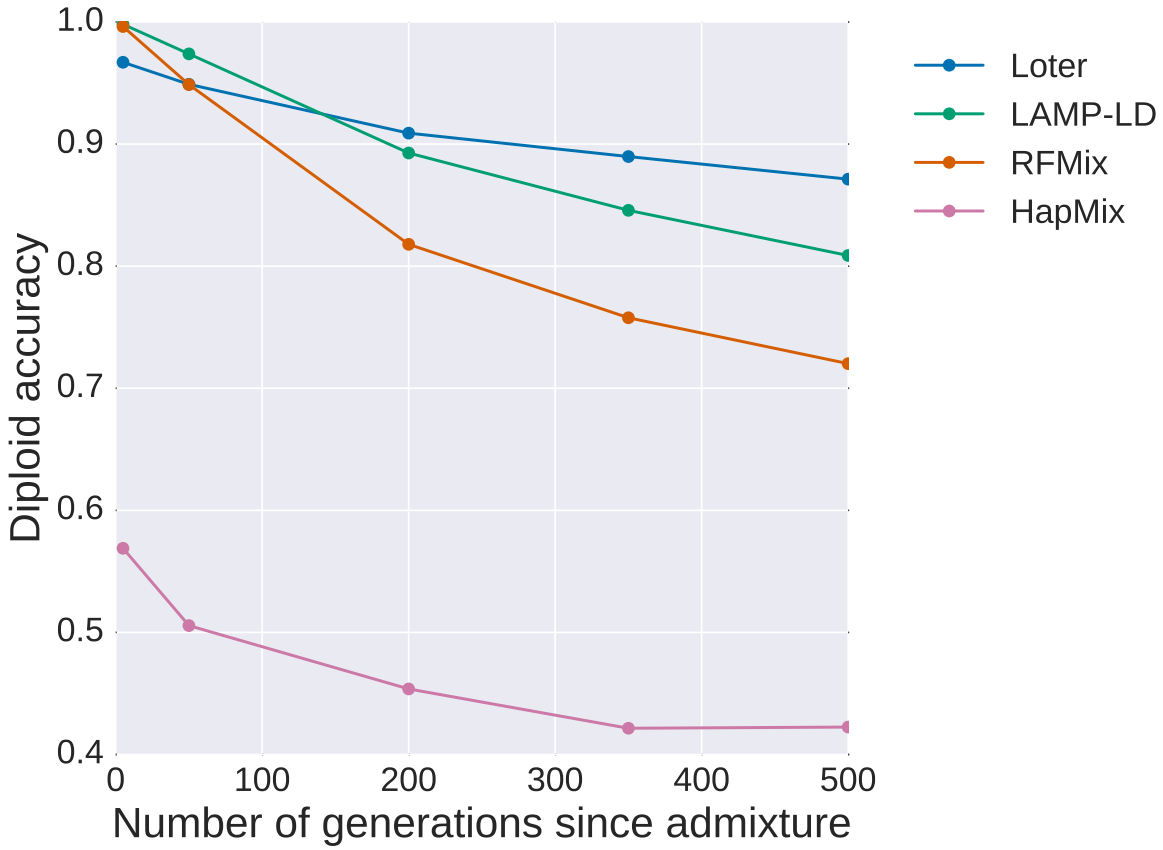

Figure SI2: Diploid accuracy obtained with LAMP-LD, Loter, and RFMix for simulated human individuals as a function of the time since admixture occured. Admixed individuals are simulated by constructing their genomes from a mosaic of true African (YRI) and European (CEU) haplotypes (International HapMap 3 Consortium et al. 2010) (Figure 3). For each admixture time, HAPMIX is evaluated using a single simulation of 48 admixed individuals. Other software, which run faster, are evaluated based on the mean diploid accuracy obtained with 20 simulated sets of 48 admixed individuals.

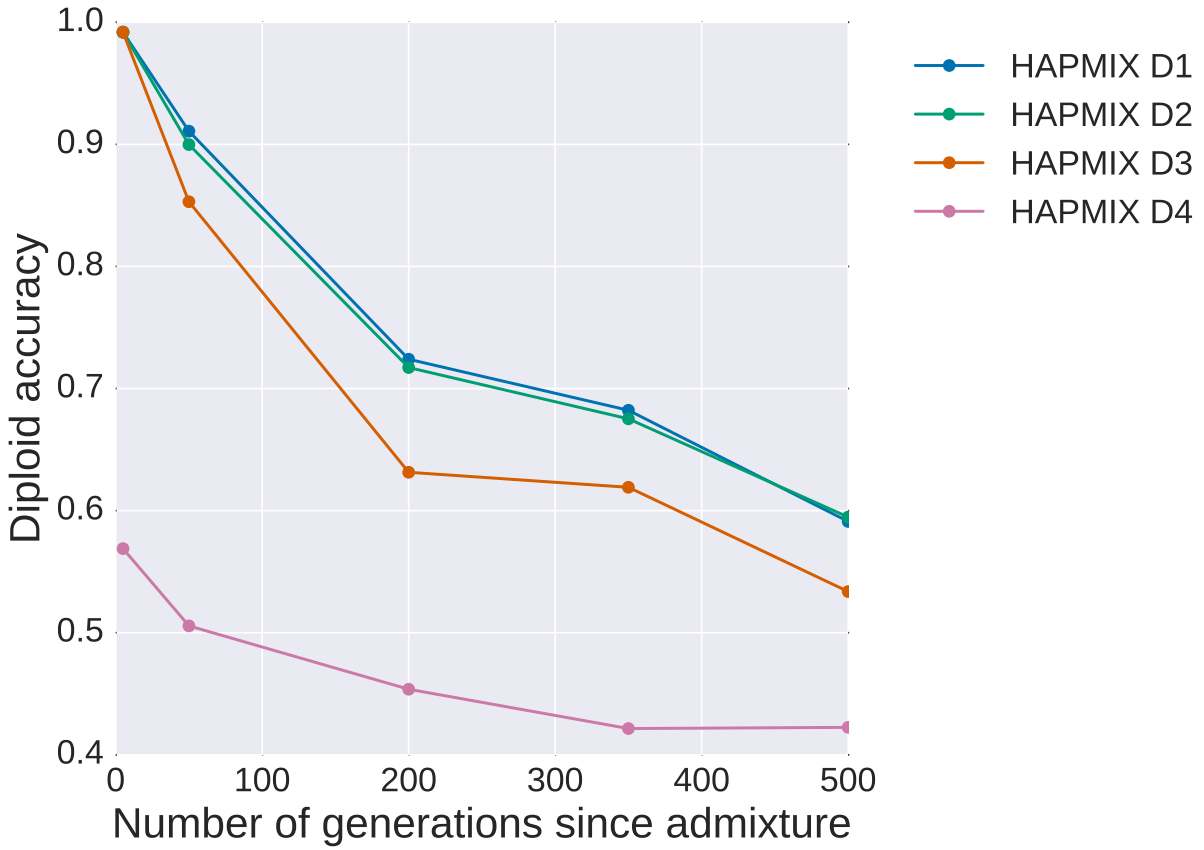

Figure SI3: Diploid accuracy obtained with HAPMIX using four different haplotype sets for LAI. The diploid accuracy of HAPMIX is severely reduced when considering reconstructed haplotypes of admixed individuals instead of true haplotypes. Admixed individuals are simulated by constructing their genomes from a mosaic of true African (YRI) and European (CEU) haplotypes (International HapMap 3 Consortium et al. 2010) (Figure 3). In set D1, we consider the same true haplotypes (trio-phased) for simulations and inference. In set D2, we consider different haplotypes for simulations and inference but haplotypes are all trio-phased and admixture time is assumed to be known. The set D3 is the same as D2 except that admixture time is unknown and assume to be equal to 6 generations. In set D4, haplotypes used for inference are not true haplotypes but have been reconstructed with Beagle.

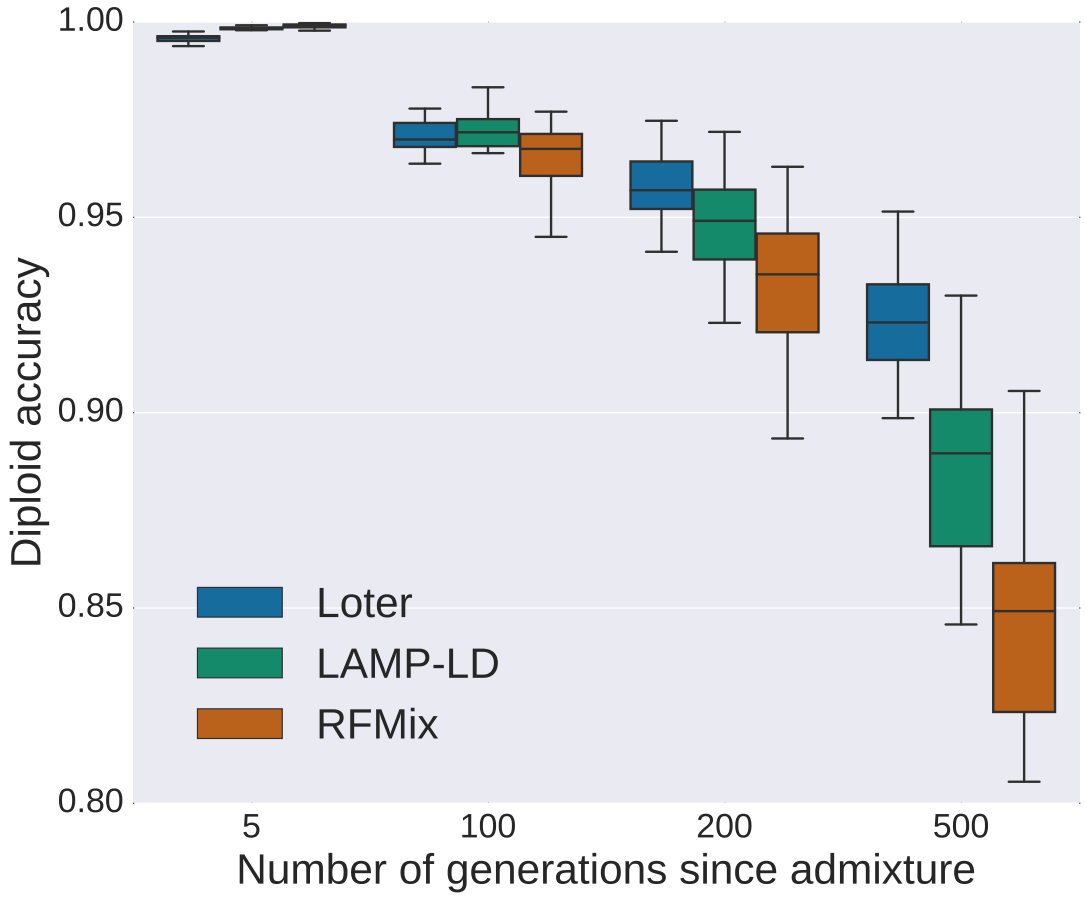

Figure SI4: Diploid accuracy obtained under a 3-way admixture model with LAMP-LD, Loter, and RFMix for simulated admixed human individuals as a function of the time since admixture occurred. Admixed individuals are simulated by constructing their genomes from a mosaic of true African (YRI), European (CEU),and Chines haplotypes (International HapMap 3 Consortium et al. 2010). For performing simulations, true haplotypes are obtained using trio information. For local ancestry inference, haplotypes are also reconstructed using trio-based inference and are different from haplotypes used for simulations. For each value of the number of generations since admixture, 20 sets of 20 admixed individuals are generated. Boxplots show the distribution of the 20 values for the mean diploid accuracy.

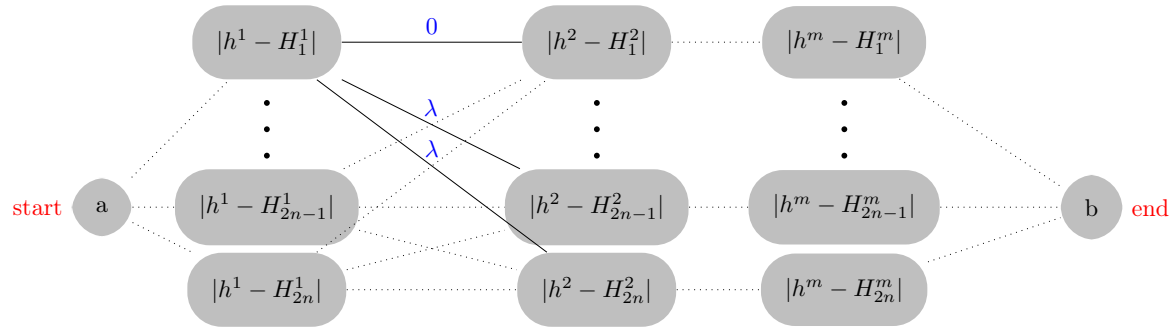

Figure SI5: Graph that represents the optimization problem of equation (1). An optimal solution for $(s_1, \ldots, s_p)$ is found by finding the shortest path from node a to node b.

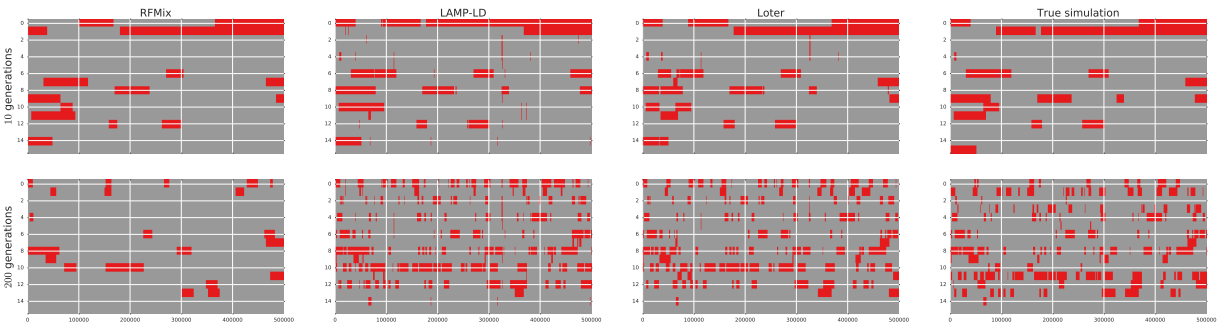

Figure SI6: Ancestry tracts for 20 simulated admixed Populus individuals. Grey chunks correspond to *P. trichocarpa* chunks and red chunks correspond to *P. balsamifera* chunks. Two rows correspond to the two haplotypes of a single individual. Ancestry switches between haplotypes are caused by haplotype phasing using Beagle. The presence of spurious and small ancestry chunks contribute to excessively decrease the median length of ancestry chunks in LAMP-LD and Loter.

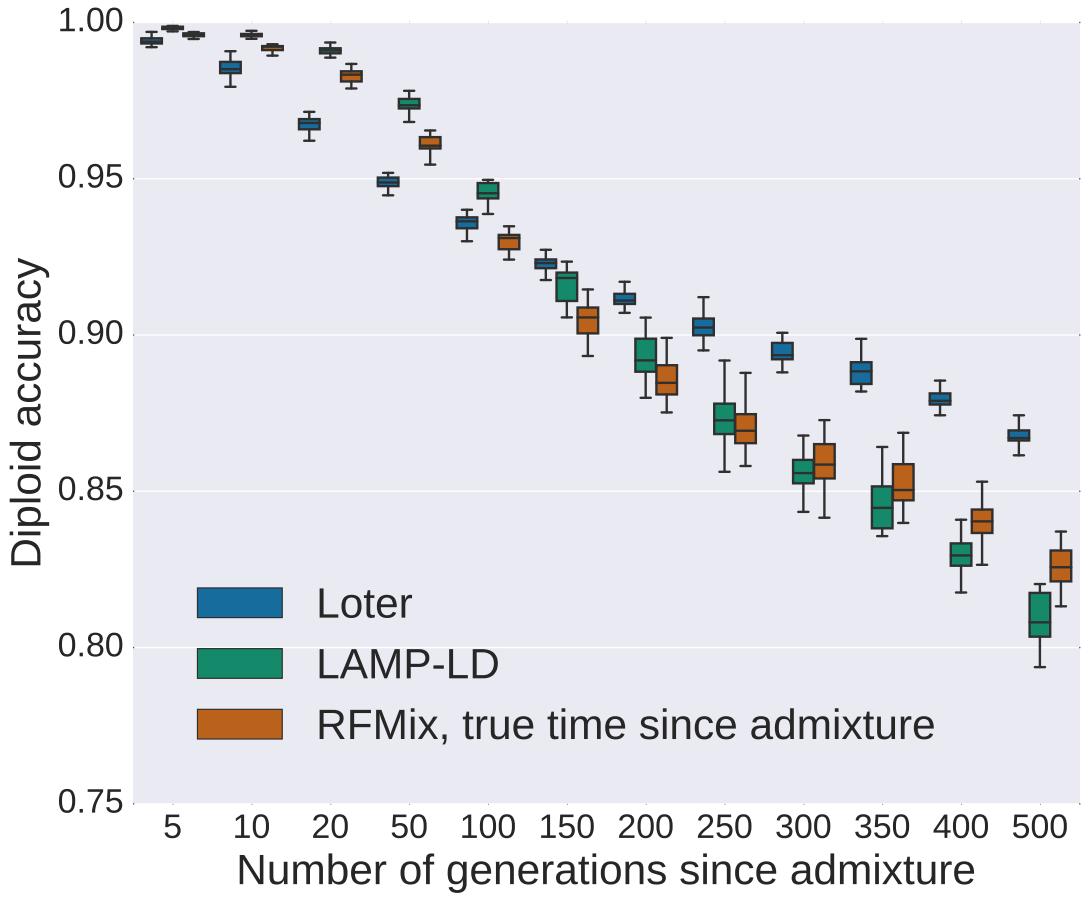

Figure SI7: Diploid accuracy obtained with LAMP-LD, Loter, and RFMix for simulated admixed Populus individuals as a function of the time since admixture occurred when true values of the time since admixture are provided to RFMIX. Admixed individuals are simulated by constructing their genomes from a mosaic of *Populus trichocarpa* and *Populus balsamifera* individuals. Individuals are phased using Beagle and two different sets of individuals are used for performing simulations and inference. For each value of the number of generations since admixture, 20 sets of 20 admixed individuals are generated. Boxplots show the distribution of the 20 values for the mean diploid accuracy.

Titre : Modélisation du déséquilibre de liaison en génomique des populations par méthodes d'optimisation.

Résumé : Nous présentons un nouveau formalisme et des nouvelles méthodes pour modéliser le déséquilibre de liaison et tenir compte de la structure en haplotypes pour les données issues de la génomique des populations. La modélisation repose sur un problème d'optimisation avec contraintes qui est résolue avec un algorithme de programmation dynamique. Les méthodes établies ont toutes l'avantage d'avoir un coût algorithmique linéaire et donc de pouvoir traiter de grands jeux de données. Dans un premier temps, nous avons appliqué notre approche à l'étude des populations métisses et plus particulièrement au problème d'inférence des coefficients de métissage locaux. Notre méthode a été appliquée à des génotypes simulés de métissage humain ainsi qu'à des vrais génotypes obtenus dans des populations métisses de peupliers. Dans un second temps, nous avons développé notre formalisme d'optimisation pour traiter de l'inférence des haplotypes à partir des génotypes d'une population. L'ensemble de ces méthodes d'optimisation a été développé dans un module Python qui s'appelle `Loter`.

Mots-clés : optimisation, déséquilibre de liaison, haplotype, structure génétique des populations, programmation dynamique, métissage, phase des haplotypes

Title : Modeling the linkage disequilibrium in population genomics with optimization methods.

Abstract : We present a new formalism and new methods to model linkage disequilibrium and to account for haplotype structure of population genomics data. Modeling relies on an optimization problem with constraints that is solved using dynamic programming. The algorithmic cost of proposed methods is linear, which is a desirable property to process large datasets. First, we applied our framework to study admixed populations and perform local ancestry inference. Our method is applied to simulated genotypes of admixed human populations and to real genotypes from admixed Populus species. Second, we developed our optimization framework to perform haploptype phasing and imputation based on a population of genotypes. All optimization methods have been developed in a Python package called `Loter`.

Keywords : optimization, linkage disequilibrium, haplotype, genetic structure of populations, dynamic programming, admixture, phasing