Algorithmes Pour l'Estimation des Coefficients de Métissage dans des Populations Continues Spatialement

Kevin Caye [1] & Olivier Michel [2] & Olivier François [3]

[1] *Centre National de la Recherche Scientifique, Université Grenoble-Alpes, TIMC-IMAG UMR 5525, Grenoble 38042, France, kevin.caye@imag.fr*

[2] *Centre National de la Recherche Scientifique, Universit e Grenoble-Alpes, GIPSA-lab UMR 5216, Grenoble 38042, olivier.michel@gipsa-lab.grenoble-inp.fr*

[2] *Centre National de la Recherche Scientifique, Université Grenoble-Alpes, TIMC-IMAG UMR 5525, Grenoble 38042, France, olivier.francois@imag.fr*

Résumé. Cet article présente une méthode qui permet l'estimation de coefficients de métissage individuels à partir de données génétiques et spatiales. Le modèle présenté repose sur une factorisation de matrice sous contraintes. L'information spatiale est prise en compte grâce à une régularisation fondée sur un graphe. Cette approche permet d'analyser des jeux de données génétiques pouvant atteindre la centaine d'individus et le million de gènes. Notre méthode d'estimation s'appuie sur un algorithme de descente par bloc de coordonnées. Afin de démontrer son utilité nous le comparons à un autre algorithme de programmation quadratique alterné. Alors que les deux algorithmes obtiennent des erreurs statistiques similaires, le premier algorithme s'exécute 100 fois plus vite pour des jeux de données comportant 500 individus et 1000 gènes.

Mots-clés. Génétique des Populations, factorisation matricielle, régularisation sur Graphe, descente par bloc de Coordonnées

Abstract. This paper presents a method to estimate individual admixture coefficients from genetic and spatial data. The model is based on a constrained matrix factorization approach. The spatial information is taken into account through regularization, based on a graph. This approach allows analyzing genetic data sets of up to a hundred individuals and a million genes. Our estimation method is based on a block-coordinate descent algorithm. To demonstrate the utility of our approach we compare it an alternating quadratic programming algorithm. While both algorithms obtain similar statistical errors, the algorithm proposed in this article runs up to 100 times faster on simulated datasets with 500 individuals and 1000 genes.

Keywords. Population genetics, matrix factorisation, graph based regularization, bloc-coordinate descent

1 Introduction

Une étape importante pour la compréhension de l'histoire d'une population à l'aide de données génétiques est l'étude de la structure génétique de cette population. Les méthodes qui utilisent des données issues du séquençage de l'ADN de plusieurs individus reposent généralement sur des modèles statistiques [1, 8, 10]. Dans le cas d'une espèce naturelle la variation géographique explique une grande partie de la variation génétique. C'est pourquoi certaines méthodes prennent explicitement en compte les données géographiques dans leurs modèles [3, 5, 7].

Afin d'étudier la structure de populations continues spatialement, nous présentons un modèle s'appuyant sur la factorisation matricielle sous contraintes et régularisée à l'aide d'un graphe porteur de l'information spatiale. Nous montrons ensuite comment il est possible de reformuler le problème afin d'utiliser un algorithme des moindres carrés alternés projetés (MCPA) pour estimer les facteurs. L'intérêt de l'algorithme des moindres carrés alternés est d'être efficace même sur des jeux de données pouvant atteindre la centaine d'individus et le million de variables. Nous mettons en évidence ce résultat en comparant les performances de notre algorithme à un algorithme d'optimisation quadratique alterné (OQA).

2 Algorithmes d'estimation des coefficients de métissage

On considère un échantillon de n individus, et pour chaque individu, L variants nucléotidiques correspondant à des emplacements physiques distincts (locus) sur les chromosomes. En supposant que seulement deux variants sont présents par locus et en choisissant un variant de référence, les données sont stockées dans une matrice de génotype G, où $G_{i,\ell}$ est le nombre de fois que le variant dérivé est observé pour l'individu i au locus ℓ. Si on note d le nombre de copies de chaque chromosome de l'espèce alors $G_{i,\ell} \in \{0, ..., d\}$. Par exemple, pour une espèce diploïde on a $G_{i,\ell} = 0, 1$ ou 2.

On suppose que les individus sont issus du métissage de K populations ancestrales, la formule des probabilité totale donne ainsi

$$P(G_{i,\ell} = j) = \sum_{k=1}^{K} Q_{i,k} f_{k,\ell}(j). \tag{1}$$

Dans l'équation (1), nous notons $Q_{i,k}$ la proportion du génome de l'individu i qui vient de la population ancestrale k. Cette grandeur est appelée coefficient de métissage. Nous notons $f_{k,\ell}(j)$ la fréquence du génome où le variant alternatif apparaît j fois au locus ℓ dans la population ancestrale k. Nous pouvons écrire l'équation (1) de la manière suivante

$$P = QF^T, \tag{2}$$

où $P = P(G_{i,\ell} = j)$ est une matrice de taille $n \times (d+1)L$, $Q = (Q_{i,k})$ est une matrice de taille $n \times K$ et $F = (f_{k,\ell}(j))$ est une matrice de taille $(d+1)L \times K$. Afin d'estimer les matrices Q et F nous introduisons une nouvelle matrice $X \in \mathbb{R}^{n \times (d+1)L}$ correspondant à un codage binaire de G. En plus des données génétiques, nous disposons de la position géographique de chaque individu. Afin de modéliser la continuité spatiale des coefficients de métissage individuel, nous utilisons un graphe pondéré par une matrice de poids W construite à partir des données géographiques. En s'inspirant du travail de [2] et [8], le problème d'estimation peut être formulé de la manière suivante

$$\begin{aligned} \min_{Q,F} \quad & ||X - QF^T||^2 + \lambda Tr(Q^T \Lambda Q) \\ \text{tel que} \quad & Q \succeq 0,\ F \succeq 0 \\ & \sum_{k=1}^{K} Q_{i,k} = 1,\ \forall i \in \{1, ..., n\} \\ & \sum_{j=0}^{d} f_{k,\ell}(j) = 1,\ \forall \ell \in \{1, ..., L\}, \end{aligned} \tag{3}$$

où la matrice Λ est la matrice laplacienne du graphe spatial tel que $\Lambda = D - W$. La matrice D est diagonale tel que $D_{i,i} = \sum_l W_{i,l}$. Les poids du graphe spatial et le paramètre λ permettent de régler la continuité spatiale des coefficients de métissage Q.

Nous présentons deux algorithmes permettant de minimiser le problème non convexe présenté dans l'équation (3). Le problème d'optimisation devient convexe par rapport à chacune des variables Q et F quand l'autre est fixée. Une approche permettant de ce problème de factorisation matricielle est d'alterner deux étapes d'optimisation quadratique (OQ) [4, 9]. On notera cet algorithme de OQ alterné OQA. L'étape de minimisation selon F s'écrit alors comme ceci

$$\min_{f \in \Delta_F} -2{c_F}^T f + f^T D_F f, \tag{4}$$

où Δ_F est le polyèdre convexe formé par les contraintes sur F (voir équation (3)). De plus, on note $f = \mathrm{Vec(F)}$ où Vec est la notation vectorielle de F. Enfin, on a $D_F = \mathrm{Id_{(d+1)L}} \otimes \mathrm{Q^T Q}$ et $c_F = \mathrm{Vec(Q^T X)}$ où $\otimes$ désigne le produit de Kronecker. De même, il est possible d'écrire le problème OQ selon Q comme ceci

$$\min_{q \in \Delta_Q} -2{c_Q}^T q + q^T D_Q q, \tag{5}$$

où Δ_Q est le polyèdre convexe formé par les contraintes sur Q (voir équation (3)), $q = \mathrm{Vec(Q^T)}$, $D_Q = \mathrm{Id_n} \otimes \mathrm{F^T F} + \lambda \Lambda \otimes \mathrm{Id_K}$ et $c_Q = \mathrm{Vec(G^T X^T)}$. Cette méthode converge vers un point stationnaire du problème d'optimisation [6].

Pour accroître l'efficacité numérique de l'algorithme, notre approche consiste à alterner une étape d'optimisation non contrainte suivie d'une projection sur le polyèdre

des contraintes. On notera cet algorithme MCPA (pour moindres carrés projeté alterné). L'étape d'optimisation selon F se résume alors à résoudre le problème suivant

$$\min_{F \in \mathbb{R}^{(d+1)L \times K}} ||X - QF^T||^2. \tag{6}$$

Le problème (6) peut se résoudre en le décomposant en $(d+1)L$ problèmes de moindres carrés. L'étape d'optimisation selon Q se fait en passant par une base de vecteurs propres de la matrice Laplacienne Λ. Notons R et D les matrices des vecteurs et des valeurs propres de Λ tels que $\Lambda = R^T D R$. Pour trouver Q, nous projetons les lignes de X sur la base des vecteurs propres $X_R = RX$. Pour chaque ligne i de Q, nous résolvons alors le problème suivant

$$\min_{q \in \mathbb{R}^K} ||(X_{R,i})^T - F^T q||^2 + \lambda \mu_i ||q||^2 \tag{7}$$

où μ_i est la $i$ème valeur propre de Λ et $X_{R,i}$ la $i$ème ligne de X_R.

3 Comparaison de l'erreur d'estimation et de la complexité des algorithmes OQA et MCPA

Afin de comparer les algorithmes OQA et MCPA, les poids du graphe spatial ont été fixés à $W_{i,j} = \exp(-||S_i - S_j||^2/\sigma^2)$ où S_i est la position spatiale de l'individu i et σ est égal à 5% de la distance spatiale moyenne entre les individus. Le paramètre λ est choisi tel que $\lambda = nL(d+1)/nK\mu_{max}$ où μ_{max} est la plus grande valeur propre de la matrice laplacienne Λ.

Pour simuler des jeux de données, nous avons échantillonné les positions géographiques des individus selon une loi normale, la moyenne μ_k de chaque groupe étant prise au hasard. Afin d'avoir une continuité spatiale, la matrice Q est calculée telle que $Q_{i,k} \propto e^{-||S_i - \mu_k||}$. La matrice F est prise au hasard dans l'espace des contraintes (voir équation 3). La Figure 1 montre que les erreurs d'estimation des matrices F et Q sont comparables pour les deux algorithmes sur ces simulations.

Un jeu de données composé de 170 individus européens de l'espèce modèle *Arabidopsis thaliana* séquencés en $216k$ locus a été utilisé pour comparer la vitesse de convergence numérique des algorithmes OQA et MCPA. Afin de mesurer la vitesse de convergence pour différentes valeurs du nombre de populations ancestrales K, l'erreur résiduelle normalisée a été évaluée pour chaque itération k (Figure 2 A). Cette figure montre que les deux algorithmes atteignent un point stationnaire à la même vitesse.

Enfin, pour comparer la vitesse d'exécution des deux algorithmes le temps moyen par itération a été mesuré sur plusieurs jeux de données simulées de la même façon que précédemment (Figures 2 B et C). On constate que la croissance du temps d'exécution par itération en fonction du nombre de locus L est similaire pour les deux algorithmes. Le temps d'exécution par itération en fonction du nombre d'individus n croit plus vite

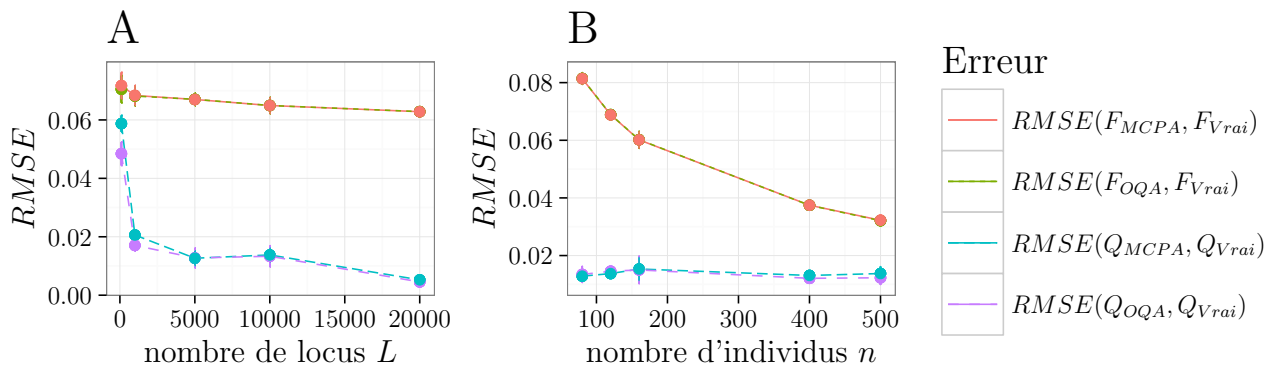

FIGURE 1 – Erreur moyenne d'estimation sur des simulations. Chaque expérience est répétée 5 fois avec $K = 4$ à la fois pour la simulation et en paramètre de l'algorithme. Le graphique A représente l'erreur en fonction du nombre de locus L et avec $n = 120$. Le graphique B représente l'erreur en fonction du nombre d'individus n et avec $L = 5000$.

pour OQA que pour MCPA. Pour $n = 500$ OQA prend 100 secondes par itération alors que MCPA prend moins de 1 seconde.

En conclusion, la convergence des deux algorithmes est similaire sur les jeux de données utilisés dans cet article (Figure 2 A). Il en est de même pour l'erreur d'estimation (Figure 1). Le résultat de cette analyse est que MCPA est significativement plus efficace que OQA pour tous les jeux de données considérés.

Références

[1] D. H. ALEXANDER *et al.* : Fast model-based estimation of ancestry in unrelated individuals. *Genome Research*, 19:1655–1664, 2009.

[2] Deng CAI *et al.* : Graph Regularized Non-Negative Matrix Factorization for Data Representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

[3] Kevin CAYE, Timo M. DEIST, Helena MARTINS, Olivier MICHEL et Olivier FRANCOIS : TESS3 : Fast Inference of Spatial Population Structure and Genome Scans for Selection. *Molecular Ecology Resources*, pages n/a–n/a, 2015.

[4] Yuansi CHEN *et al.* : Fast and Robust Archetypal Analysis for Representation Learning. *Arxiv*, 2014.

[5] Jukka CORANDER *et al.* : Bayesian spatial modeling of genetic population structure. *Computational Statistics*, 23(1):111–129, 2008.

[6] DIMITRI P. BERTSEKAS : *Nonlinear Programming*. Athena scientific, 1999.

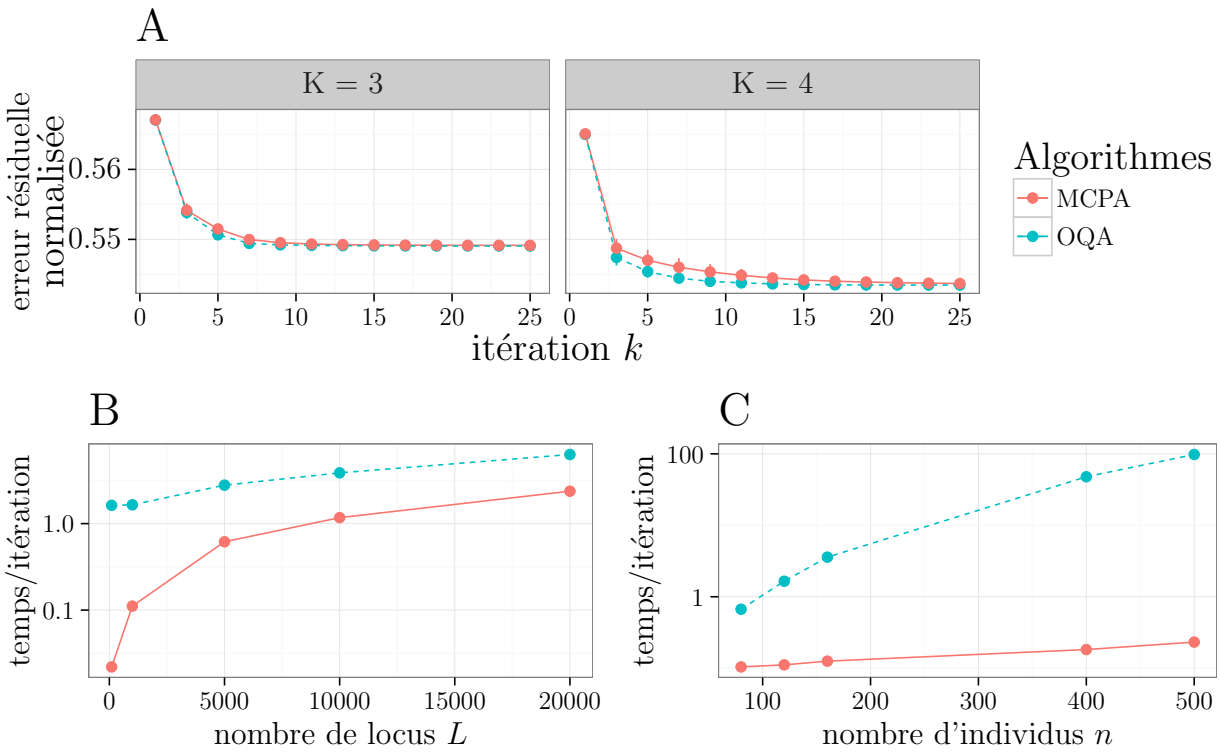

FIGURE 2 – Le graphique A représente l'erreur résiduelle normalisée en fonction du nombre d'itération pour un jeu de données réelles où $n = 170$ et $L = 216.10^3$. Les graphiques B et C représentent les temps par itérations sur des simulations, $n = 120$ pour B et $L = 1000$ pour C. Enfin $K = 4$ pour B et C.

[7] E. DURAND *et al.* : Spatial Inference of Admixture Proportions and Secondary Contact Zones. *Molecular Biology and Evolution*, 26(9):1963–1973, 2009.

[8] E. FRICHOT *et al.* : Fast and Efficient Estimation of Individual Ancestry Coefficients. *Genetics*, 196(4):973–983, 2014.

[9] Jingu KIM *et al.* : Fast Nonnegative Matrix Factorization : An Active-Set-Like Method and Comparisons. *SIAM Journal on Scientific Computing*, 33(6):3261–3281, 2011.

[10] Jonathan K PRITCHARD *et al.* : Inference of Population Structure Using Multilocus Genotype Data. *Genetics*, 155(2):945–959, 2000.