Fine mapping of antibiotic resistance determinants

Magali Jaillard Dancette

N° d'ordre NNT : 2018LYSE1282

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
l'Université Claude Bernard Lyon 1

École Doctorale ED 341
Evolution, Ecosystèmes, Microbiologie, Modélisation (E2M2)

Spécialité de doctorat : Biomath-Bioinfo-Génomique évolutive

Soutenue publiquement le 12/12/2018, par :
Magali Jaillard Dancette

Vers une cartographie fine des polymorphismes liés à la résistance aux antimicrobiens.

Devant le jury composé de :

Patricia Doublet, Professeure, CIRI, Université de Lyon	Présidente
Elizabeth Purdom, Professeure associée, University of California, USA	Rapporteure
Daniel Wilson, Professeur associé, University of Oxford, UK	Rapporteur
Hélène Touzet, Directrice de recherche CNRS, CRIStAL, Université de Lille	Examinatrice
Pierre Peterlongo, Chargé de recherche INRIA, INRIA/Irisa, Rennes	Examinateur
Franck Picard, Directeur de recherche CNRS, LBBE, Univérsité Lyon 1	Directeur de thèse
Laurent Jacob, Chargé de recherche CNRS, LBBE, Univérsité Lyon 1	Co-directeur de thèse
Pierre Mahé, bioMérieux	Invité

Staphylococcus saprophyticus
Aquarelle Magali Jaillard

Laboratoire d'accueil

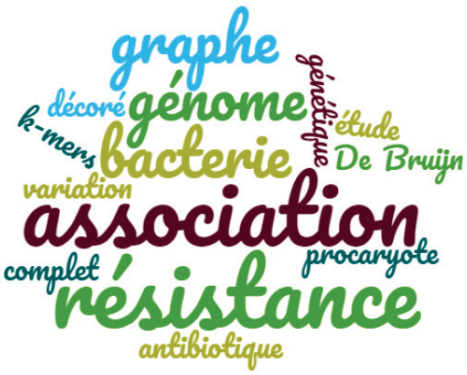
Université Lyon 1
UMR5558 - BIOMÉTRIE ET BIOLOGIE EVOLUTIVE
43, Boulevard du 11 Novembre 1918
69100 VILLEURBANNE
France

Vers une cartographie fine des polymorphismes liés à la résistance aux antimicrobiens.

Mieux comprendre les mécanismes de la résistance aux antibiotiques est un enjeu important dans la lutte contre les maladies infectieuses, qui fait face à la propagation de bactéries multi-résistantes. Les études d'association à l'échelle des génomes sont des outils puissants pour explorer les polymorphismes liés aux variations phénotypiques dans une population. Leur cadre méthodologique est très documenté pour les eucaryotes, mais leur application aux bactéries est très récente. Durant cette thèse, j'ai cherché à rendre ces outils mieux adaptés aux génomes plastiques des bactéries, principalement en travaillant sur la représentation des variations génétiques. En effet, parce que les bactéries ont la capacité à échanger du matériel génétique avec leur environnement, leurs génomes peuvent être trop différents au sein d'une espèce pour être alignés contre une référence. La description des variations par des fragments de séquence de longueur k, les k-mers, offre la flexibilité nécessaire mais ne permet pas une interprétation directe des résultats obtenus. La méthode mise au point teste l'association de ces k-mers avec le phénotype, et s'appuie sur un graphe de De Bruijn pour permettre la visualisation du contexte génomique des k-mers identifiés par le test, sous forme de graphes. Cette vue synthétique renseigne sur la nature de la séquence identifiée: il peut par exemple s'agir de polymorphisme local dans un gène ou de l'acquisition d'un gène dans un plasmide. Le type de variant représenté dans un graphe peut être prédit avec une bonne performance à partir de descripteurs du graphe, rendant plus opérationnelles les approches par k-mers pour l'étude des génomes bactériens.

Mots-clés

études d'association à l'échelle des génomes complets, antibiorésistance, graphes de De Bruijn, variations génomiques, k-mers, graphe décoré, génétique des procaryotes, génomes bactériens



Fine mapping of antibiotic resistance determinants

The emergence and spread of multi-drug resistance has become a major worldwide public health concern, calling for better understanding of the underlying resistance mechanisms. Genome-wide association studies are powerful tools to finely map the genetic polymorphism linked to the phenotypic variability observed in a population. However well documented for eukaryotic genome analysis, these studies were only recently applied to prokaryota. Through this PhD project, I searched how to better adapt these tools to the highly plastic bacterial genomes, mainly by working on the representation of the genetic variations in these genomes. Indeed, because the bacteria have the faculty to acquire genetic material by a means other than direct inheritance from a parent cell, their genomes can differ too much within a species to be aligned against a reference. A representation using sequence fragments of length k – the so-called k-mers – offers the required flexibility but generates redundancy and does not allow for a direct interpretation of the identified associations. The method we set up tests the association of these k-mers with the phenotype, and takes advantage of a De Bruijn graph (DBG) built over all genomes to remove the local redundancy of k-mers, and offer a visualisation of the genomic context of the k-mers identified by the test. This synthetic view as DBG subgraphs informs on the nature of the identified sequence: *e.g.* local polymorphism in a gene or gene acquired through a plasmid. The type of variant can be predicted correctly in 96% of the cases from descriptors of the subgraphs, providing a tractable framework for k-mer-based association studies.

Key words

genome-wide association study, GWAS, antibiotic resistance, De Bruijn graph, DBG, genome variation, k-mers, decorated graph, prokaryotic genetics, bacterial genome

Glossaire

Terme/abréviation	**Définition**
ADN	molécule d'acide désoxyribonucléique, support de l'information génétique
aa	acide aminé, unité moléculaire composant les protéines
nucléotides, nt	unité moléculaire composant l'ADN (A,C,G,T) et l'ARN (A,C,G,U).
(M)pb	(million de) paire(s) de bases, unité de mesure de la longueur d'une séquence nucléique. Une paire de bases faire référence à deux nucléotides complémentaires dans la molécule d'ADN double-brin
séquenceur nouvelle génération	famille de technologies haut débit permettant de déterminer la composition d'une séquence nucléique
lecture (read)	donnée générée par un séquenceur: séquence inférée représentant un fragment de la séquence complète
k-mer	toute séquence nucléique de k nucléotides
contig	fragment de séquence produit par l'assemblage de lectures
unitig	assemblage de séquences pour lesquelles il n'y a pas d'ambiguïté de choix; nœuds d'un graphe de De Bruijn compacté
génotype	information génétique spécifique d'un individu
phénotype	caractère observable d'un individu
SNP	modification ponctuelle d'un nucléotide dans une séquence relativement à une autre
indel	insertion ou suppression locale de nucléotides dans une séquence relativement à une autre
épistasie	interaction entre deux positions du génome (par exemple entre gènes ou entre mutations)
homoplasie	génotype similaire obtenu dans des populations indépendantes sous les mêmes contraintes
plasticité du génome	nature altérable des génomes bactériens qui ont la capacité à échanger du matériel génétique
core génome	partie du génome partagée au sein d'une population
génome accessoire	partie du génome qui n'est pas partagée au sein d'une population
élément génétique mobile	élément capable de se déplacer au sein et entre les génomes
transfert de gène horizontal	acquisition de matériel génétique par d'autres moyen que la reproduction asexuée
souche	variant d'un microorganisme; des souches distinctes diffèrent par leurs génomes
CMI	concentration minimal inhibitrice, plus faible concentration d'antibiotique inhibant la croissance bactérienne après une nuit d'incubation
CLSI	Clinical & Laboratory Standards Institute, institut américain proposant des standards pour la mesure et l'interprétation des CMI
EUCAST	European Committee on Antimicrobial Susceptibility Testing, institut européen proposant des standards pour la mesure et l'interprétation des CMI
MDR	souche muti-résistante (2 antibiotiques)
XDR	souche extrêmement résistante (plus de 2 antibiotiques)
antimicrobien	agent tuant des microorganismes ou inhibant leur croissance
antibiotique	antimicrobien agissant spécifiquement sur les bactéries
résistome	collection des déterminants génétiques décrits dans la littérature pour conférer la résistance
RWAS	étude d'association à l'échelle du résistome
GWAS	étude d'association à l'échelle du génome
LD	déséquilibre de liaison, corrélation entre variants génétiques
DBG/cDBG/pan-cDBG	graphe de De Bruijn / graphe de De Bruijn compacté / graphe de De Bruijn compacté construit à l'échelle d'une population

Glossary

Term/abbreviation	Definition
DNA	deoxyribo-nucleic acid molecule, support of the genetic information
aa	amino-acid, molecule unit composing the proteins
nucleotides, nt	molecule unit composing the DNA (A,C,G,T) and RNA (A,C,G,U).
bp	base pairs, unit of measure of a nucleic sequence. A base pair relates to two complementary nucleotides in a double strand DNA molecule.
NGS	next-generation sequencing, high throughput technology used to determine the composition of nucleic sequences
read	raw inferred sequence, product of a sequencer representing a fragment of the complete sequence,
k-mer	any nucleic sequence fragment of length k
contig	sequence fragment produced from the assembly of reads, in which the order of bases is known to a high confidence level.
unitig	assembly of fragments for which there are no competing choices in terms of internal overlaps; nodes of a compacted De Bruijn graph
genotype	genetic information specific to an individual
phenotype	observable character trait of an individual
SNP	single nucleotide polymorphisme, punctual change in the nucleic sequence
indel	local insertion or deletion in the nucleic sequence
epistasy	interaction between two potitions (e.g. genes or mutations) in the genome
homoplasy	similar genotype obtained in independant population under the same constraints
genome plasticity	alterable nature of bacterial genomes that enables the exchange of genetic material.
core genome	the part of the genome shared within a population
accessory genome	the part of the genome not shared within a population
MGE	mobile genetic element, genetic entity capable of moving within and between genomes
HGT	horizontal gene transfer, acquisition of genetic material by a means other than asexual reproduction.
strain	variant of a microorganism; distinct strains differ by their genomes
MIC	minimal inhibitory concentration, lowest concentration of antibiotic that inhibit the growth after overnight incubation.
CLSI	Clinical & Laboratory Standards Institute
EUCAST	European Committee on Antimicrobial Susceptibility Testing
MDR	multi-drug resistant
XDR	extensively drug resistant
antimicrobial	agent that kills microorganisms or inhibits their growth
antibiotics	particular antimicrobial killing specifically bacteria
resistome	set of variants described on the literature to be involved in the antibiotic resistance
RWAS	resitome-wide association study
GWAS	genome-wide association study
LD	linkage disequilibrium, correlation between variants
DBG/cDBG/pan-cDBG	de Bruijn graph / compacted DBG / cDBG built over a population of genomes

Lutte contre la résistance antimicrobienne: une pierre à l'édifice

Une résistance mondiale
Contexte général de l'antibiorésistance

Il y a 90 ans, Alexander Fleming découvrait la pénicilline et ouvrait une voie thérapeutique nouvelle et efficace de lutte contre les maladies infectieuses: celle des antibiotiques. Mais sous la pression sélective de ces molécules mortelles pour elles, les bactéries ont naturellement évolué, illustrant au passage la théorie de l'évolution de Darwin: s'adapter ou disparaître. Rapidement les premières souches bactériennes insensibles à la pénicilline ont été observées et dès 1945, Fleming mettait en garde contre un usage inapproprié des antibiotiques: en augmentant la pression de sélection, on allait rendre les bactéries plus résistantes et les antibiotiques moins efficaces.
Après plusieurs décennies d'âge d'or de ces médicaments – qui se sont au cours des années rendus indispensables à la médecine moderne – la course contre la montre a démarré: chaque nouvelle molécule antibiotique introduite a vu l'apparition d'une résistance quelques années plus tard. Depuis les années 1990, la recherche stagne et aucune nouvelle famille de molécule n'a été découverte. La course est-elle perdue d'avance ?
En 2014, l'Organisation Mondiale de la Santé tire pour la première fois la sonnette d'alarme dans un rapport sur l'émergence des résistances aux antimicrobiens: la proportion de souches microbiennes ayant développé des mécanismes pour résister à un ou plusieurs antimicrobiens croît dangereusement, en particulier dans les pays à faible revenu [212]. Une étude effectuée sur la période 2000 à 2015 met en lien la consommation croissante d'antibiotiques et l'augmentation des résistances [103].
On parle aujourd'hui d'ère post-antibiotique, et les prévisions sont actuellement très pessimistes: la propagation des souches bactériennes résistantes et multi-résistantes à l'échelle du globe remet en cause notre arsenal antibiotique. Les soins les plus anodins de la chirurgie, ou de banales infections urinaires pourront être mortels sans antibiotique. Il est estimé qu'en 2050, il y aura 10 millions de morts par an causés par une maladie infectieuse, soit plus que par le cancer [154].
L'accélération et la propagation de l'antibiorésistance constitue aujourd'hui l'une des plus graves menaces pesant sur la santé mondiale. Bien qu'il ait été observé qu'une baisse de la consommation d'antibiotique puisse entraîner une baisse de la prévalence des souches résistantes [53, 211], beaucoup d'études sont pessimistes quant à la réversibilité des résistances [7, 84, 89] et il est à craindre que nous ne puissions éradiquer le phénomène. Il reste à espérer que l'on arrive à trouver un équilibre entre notre usage des antibiotiques et l'émergence et la propagation des résistances.

Un large éventail d'armes
Mécanismes de résistance et leurs bases génétiques

En fonction de l'espèce à laquelle elles appartiennent, et en fonction du mode d'action de l'antibiotique, les bactéries agissent de différentes manières pour se défendre [24]. Elles peuvent mettre en place des mécanismes pour agir sur les sites actifs de l'antibiotique, modifier sa cible, réduire la perméabilité de la membrane – et ainsi la pénétration de l'antibiotique, ou permettre l'efflux de l'antibiotique hors de la membrane (Figure a.).
Certaines résistances sont intrinsèques. Par exemple, la double paroi des bactéries à Gram négatif leur confère une résistance structurelle à la vancomycin: la molécule ne passe pas la membrane. De la même manière, les pompes à efflux sont naturellement présentes chez de nombreuses espèces. Cependant une grande partie des mécanismes de résistance est acquise par des modifications au niveau du génome des bactéries.

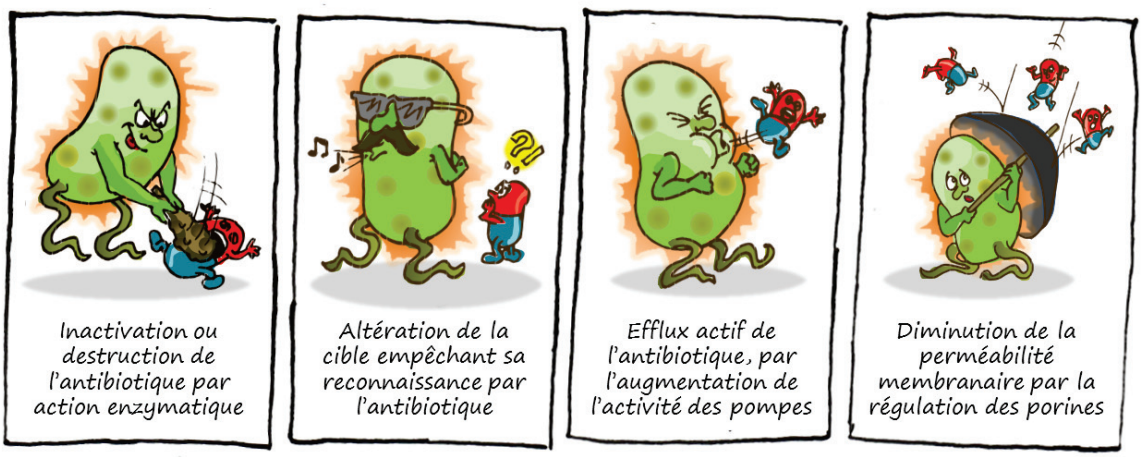


Figure a.: **Mécanismes de résistance** Vue de l'auteure.

Ces modifications génétiques ont plusieurs origines [4]. Elles peuvent être liées à l'acquisition d'un ou plusieurs gènes qui peuvent conférer la résistance. En effet, la plupart des bactéries ont la possibilité d'échanger et intégrer du matériel génétique provenant de l'extérieur [189]. On parle alors de transfert horizontal (par opposition à l'apport 'vertical' de leur reproduction asexuée). Il peut également être la conséquence de mutations ponctuelles locales. Il peut par exemple s'agir de SNPs (acronyme anglais de *single nucleotide polymorphism*, permutation ponctuelle d'un nucléotide) ou d'indels (insertion ou délétion ponctuelle d'un nucléotide) dans les gènes ou les régions intergéniques.

Parmi les mécanismes les mieux décrits, la modification de la séquence du gène ciblé par l'antibiotique peut être très efficace. C'est le mécanisme le plus fréquemment observé de résistance aux fluoroquinolones, antibiotiques qui ciblent les gènes acteurs de la réplication de l'ADN. Dans le cas de la résistance aux bêta-lactamines, qui agissent sur la paroi cellulaire, le mode d'action privilégié chez les entérobactéries est l'utilisation d'une enzyme capable d'inactiver l'antibiotique, naturellement présente ou acquise par transfert horizontal.

D'autres mécanismes plus complexes impliquent de multiples facteurs, la combinaison ou l'interaction de plusieurs gènes, des systèmes de régulation de l'expression des gènes, etc., et leur compréhension demeure souvent incomplète [89, 112]. Les pompes à efflux, largement répandues au sein de l'espèce *Pseudomonas aeruginosa*, rejettent activement les composés toxiques hors de la membrane, et peuvent être efficaces contre différentes familles d'antibiotiques. C'est le cas également des porines qui régulent la perméabilité de la cellule et qui ne sont pas spécifiques d'une famille d'antibiotiques. Porines et pompes sont naturellement présentes chez *P. aeruginosa* qui, en présence d'antibiotique, emploie des voies complexes pour réguler ou co-réguler leur expression. L'acquisition de mutations dans les gènes régulant ces voies peuvent avoir un fort effet sur le niveau de résistance [124, 137]. La formation de biofilms est un autre mécanisme de résistance indirect dont toutes les bases ne sont pas encore bien comprises [132, 171].

Toutes les bactéries n'ont pas accès au même arsenal. *Mycobacterium tuberculosis*, une espèce très clonale, n'acquiert pas de nouveau gène par transfert horizontal: la résistance est principalement liée à des mutations ponctuelles dans les gènes ou leurs régions promotrices. D'autres espèces au contraire échangent facilement du matériel génétique: le gène *mcr*-1 permettant la résistance à la colistine a été récemment identifié chez *Escherichia coli*, puis un transfert aux espèces *Klebsiella pneumoniae* et *P. aeruginosa* a été mis en

évidence [126].
La métaphore de la course aux armements que j'emploie est à nuancer lorsque l'on replace les déterminants de la résistance au sein de voies métaboliques plus globales et de réseaux de régulation, et plus généralement lorsque l'on considère leur part dans la physiologie des bactéries [138], mais elle offre un cadre général facile à appréhender dans lequel ancrer la présentation de mes travaux de thèse.

Mieux connaître son adversaire
Enjeux de la thèse, présentation de l'état de l'art

Ces mécanismes d'action sont étudiés depuis l'observation des premières résistances dans les années 1940. Des sites comme card.mcmaster.ca [141] ou megares.meglab.org [111] recensent aujourd'hui une grande partie des régions des génomes bactériens connues pour, ou suspectées d'être impliquées dans la résistance aux antibiotiques. Les mécanismes de résistance évoluant et se diversifiant, notre connaissance doit sans cesse être améliorée.
Dans ce contexte, le projet de thèse vise à développer des méthodes et outils permettant de compléter et améliorer cette connaissance des déterminants génétiques de l'antibiorésistance.
Au-delà de la thèse, les fruits de ce travail pourront par exemple servir à affiner la base de données d'un système de diagnostic, dans le but d'optimiser l'antibiothérapie d'un patient [56, 157].
L'enjeu est donc d'identifier de nouvelles régions du génome impliquées dans la résistance aux antibiotiques, de les annoter, et de quantifier leur impact sur le phénotype. Les études d'association entre phénotype (l'antibiorésistance exprimée par la bactérie) et génotype (l'information contenue dans son génome) offrent un cadre de choix pour compléter cette cartographie génétique. En 2006, Daniel Falush ouvre la voie de ces études aux bactéries – elles n'étaient jusqu'alors appliquées qu'aux organismes eucaryotes – en mettant en lumière les obstacles à surmonter pour les rendre efficaces: en particulier pouvoir estimer correctement les fortes structures des populations bactériennes, et cibler le plus grand nombre de variants génétiques, notamment ceux impliqués dans la régulation des gènes [63].
Ce type d'étude chez les bactéries n'a réellement vu son essor qu'en 2012. Entre 2012 et 2015, année du démarrage de la thèse, la majorité de ces études [3, 43, 110, 202] étaient basées sur des approches développées pour les organismes eucaryotes: recherche de SNPs par rapport à un génome de référence et modèle d'association prenant en compte la structure de la population [170, 190]. Une étude cependant proposait un test de convergence phylogénétique dédié aux bactéries [64], tandis qu'une autre proposait de travailler avec des k-mers (tous les fragments de k nucléotides contenus dans la séquence du génome) plutôt qu'avec un génome de référence [183]. A partir de 2015, plusieurs études ont suivi la voie des k-mers, tant pour la recherche de nouveaux marqueurs [61, 117] que pour la prédiction du phénotype [58].
Une synthèse de ces études réalisée en début de thèse a été intégrée dans un article publié en 2017 [59].

Mesure des forces ennemies
Phénotype: mesure de l'antibiorésistance

La mesure internationalement utilisée pour quantifier le phénotype que nous étudions – le niveau de résistance d'une souche bactérienne – est la concentration minimale inhibitrice (CMI). Elle se définie comme la concentration la plus faible d'antibiotique inhibant la croissance bactérienne après une nuit d'incubation [8]. La donnée brute peut être une concentration (ex: $8\mu g/mL$) ou une classe de concentrations (ex: $\geqslant 4\mu g/mL$).

Des organismes comme le CLSI (Clinical & Laboratory Standards Institute) et l'EUCAST (European Committee on Antimicrobial Susceptibility Testing) fournissent des standards de mesure et d'interprétation de ces CMI, et notamment des seuils pour définir si la bactérie est résistante, intermédiaire ou sensible à un antibiotique. Cette information est directement utilisée par les cliniciens pour adapter leur antibiothérapie.

Il existe plusieurs méthodes de mesure de la CMI, parmi lesquelles des méthodes dites de référence, comme la microdilution, ou des systèmes automatisés permettant un débit d'analyse élevé, comme le système Vitek®2 vendu par bioMérieux [177].

Selon l'antibiotique et l'espèce étudiée, les mesures effectuées avec une méthode ne sont pas toujours reproductibles [71] et peuvent varier entre méthodes [6, 85]. Un phénotype brut (CMI) est plus riche à exploiter car on peut identifier des variations du trait phénotypique au sein d'une population de souches sensibles (respectivement résistantes). Cependant l'utilisation d'un phénotype simplifié (binaire: sensible contre non-sensible) est largement répandu car il constitue l'information utilisée par les cliniciens.

Sur le terrain, des opportunités
Génotype: issu du séquençage haut-débit

Le génotype d'une souche bactérienne est représenté par les variations particulières de son génome par rapport à un génome de référence ou par rapport à tous les génomes d'une population. La donnée première est la séquence de son génome complet obtenue par séquençage.

Aujourd'hui trois techniques de séquençage haut-débit sont dominantes: la détection optique d'étiquettes fluorescentes spécifiques fixées aux nucléotides lors de leur synthèse (principe des séquenceurs Illumina® [17]), la détection de différences de potentiels spécifiques de chaque nucléotide, liées à l'émission d'ions hydrogènes lors de la synthèse de l'ADN (principe des séquenceurs Ion Torrent™ [178]) et la détection des fluctuations d'un courant électrique spécifique de la séquence d'ADN qui traverse un nanopore (principe des séquenceurs Oxford Nanopore [31]).

Ces technologies produisent toutes une très grande quantité de fragments de la séquence du génome, appelées lectures (ou plus fréquemment *reads*, le terme anglais). Il existe une grande variabilité de la longueur et du taux d'erreur de ces reads selon la technologie [172]. L'obtention de la séquence du génome se fait ensuite par l'assemblage de ces reads en plus longs fragments, les contigs [21, 160], puis éventuellement les contigs peuvent être re-assemblés jusqu'à retrouver la séquence d'un chromosome ou d'un plasmide complet [72, 104]. Enfin, les séquences des génomes peuvent être annotées avec les positions de domaines fonctionnels comme les gènes ou les régions régulatrices [155, 181].

Depuis son avènement il y a plus d'une dizaines d'année, le séquençage haut-débit s'est progressivement imposé en microbiologie [107, 129]. Il est aujourd'hui possible d'obtenir le génome assemblé et annoté de microorganismes en un temps et un coût relativement bas [56]. Ce progrès technique a permis de construire des panels de souches bactériennes pour lesquelles le génome séquencé et une mesure phénotypique sont disponibles.

Les premières études d'associations bactériennes incluaient plusieurs dizaines de génomes (50 en 2012 [219], entre 60 et 160 l'année suivante [64, 183, 187]), puis à partir de 2014, des panels de plusieurs centaines voire plusieurs milliers de souches ont commencé à être étudiés, notamment pour les espèces *Streptococcus pneumoniae* [43], *M. tuberculosis* [61, 202] et *Staphylococcus aureus* [61, 79].

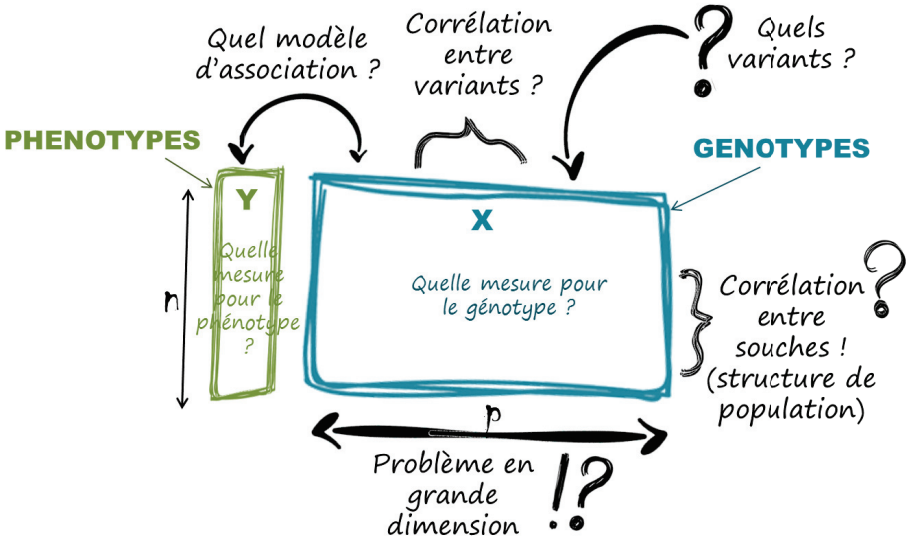


Figure b.: **Mise en place d'une étude d'association.** Pour estimer la relation entre le phénotype (vecteur Y, donnant pour chacune des n souches une mesure de l'antibiorésistance) et le génotype (matrice X décrivant pour chaque souche l'ensemble des p variations au niveau de son génome), plusieurs questions se posent.

Notre arsenal

Cadre méthodologique des études génotype-phénotype, et questions associées

Les études d'association génotype-phénotype, bien que récentes chez les procaryotes, ont une longue histoire chez les eucaryotes et en particulier en génétique humaine [33]. Notre travail s'appuie sur ces développements et cherche des pistes pour les rendre plus adaptés aux études chez les procaryotes. Nous nous sommes restreints ici au cadre majoritairement répandu des modèles de régression. La régression permet de modéliser une relation entre variables, ici entre un phénotype et des variations génétiques. Le modèle est estimé à partir d'un échantillon de la population de n individus. Ces modèles peuvent, avec les mêmes données d'entrée, répondre à deux objectifs de travail: la prédiction du phénotype et la recherche de marqueurs.

Dans le premier cas, l'objectif est l'optimisation des performances de prédiction en généralisation, c'est-à-dire sur d'autres jeux de données que celui utilisé pour construire le modèle. Dans ce cas, l'interprétation des variables utilisées pour le modèle et de leur coefficient, est secondaire.

Dans le second cas, celui qui nous intéresse, l'objectif est la recherche de variables significativement associées au phénotype. L'identification de ces variables offre des pistes de travail en laboratoire pour mieux comprendre le phénotype. Leur interprétation est donc au premier plan. La significativité de l'association de chaque variable est estimée par une procédure de test d'hypothèse: on teste l'hypothèse nulle, H_0, que le coefficient de la régression associé à la variable est égal à zéro (soit que la variable n'est pas associée au phénotype). La p-value calculée lors du test exprime la probabilité que l'hypothèse nulle soit rejetée par erreur (le test conclut que le coefficient est non-nul alors que ce n'est pas le cas), et est une mesure reconnue de la significativité d'une association.

Il s'agit donc d'identifier des associations significatives à partir de plusieurs centaines de milliers voire des millions de variables explicatives. Plusieurs questions se posent lors de la mise en place d'une telle étude (Figure b.).

- **Quels variants pour décrire le génotype?** Dans la littérature des études d'association chez les humains, l'unité de mesure de la variation génétique est le SNP [1, 33, 92]. Cette unité n'est pas satisfaisante pour les études chez les bactéries [175]. En effet, une grande part de la variabilité génomique bactérienne réside dans les transferts horizontaux. Il en résulte des génomes qui n'ont pas tous la même composition au sein d'une espèce et les SNPs seuls ne peuvent parvenir à décrire la diversité de cette partie du génome non partagée, dite accessoire [109]. Parmi les premières études chez les bactéries, il a été proposé de tester la présence des gènes en plus de celle des SNPs, et également de tester la présence de k-mers [61, 183]. La question de la description du génotype a été au cœur de la thèse, et nous nous sommes notamment intéressés aux approches basées sur les k-mers pour nos développements.

- **Quelle mesure pour le génotype ?** Les GWAS humains utilisent, pour encoder les SNPs sur les deux allèles présents chez chaque individu, les valeurs 0 (homozygote avec l'allèle majoritaire), 1 (hétérozygote) ou 2 (homozygote avec l'allèle minoritaire). Ici encore, la transposition ne peut être directe, les génomes bactériens étant haploïdes (présence d'un seul allèle par individu). Cependant, au niveau de la population bactérienne complète, plus de 2 allèles peuvent être observés pour un SNP. L'encodage peut alors se faire en utilisant plusieurs variables, décrivant chaque allèle. Ensuite, la mesure du génotype peut se faire de manière binaire (observation ou non du génotype dans la souche), comme c'est le cas dans les études bactériennes précédemment citées, ou par comptage (nombre d'observations). Le comptage peut en effet renseigner sur le nombre de copies de plasmides ou de gènes répétés. Cependant la séquence assemblée ne donne que partiellement l'information du nombre de copies car une partie des répétitions a déjà été supprimée [163], alors qu'un comptage normalisé au niveau des reads peut en donner une meilleure estimation [145].

- **Quelle mesure pour le phénotype ?** Le phénotype de l'antibiorésistance, lorsqu'il est donné en CMI, peut être représenté par l'appartenance à une catégorie ordonnée de concentration (cf. Figure S1.5 en supplément). On peut également utiliser une valeur transformée de la CMI: son logarithme [62] ou une valeur binaire issue des seuils CLSI ou EUCAST. Les études d'association précédemment citées utilisent toutes un phénotype binaire (sensible/résistant ou sensible/non-sensible). Cette question, ainsi que la suivante, ont été abordées en début de thèse.

- **Quel modèle d'association ?** Dans le cadre des régressions linéaires généralisées que l'on s'est donné, le modèle dépend directement de l'hypothèse faite sur le phénotype: un modèle ordinal sera adapté aux données catégorielles ordonnées [142], un modèle linéaire aux données numériques continues [185], un modèle logistique aux données binaires [49].

- **Problème en grande dimension.** Le nombre de covariables (les p variants génétiques) est très supérieur – de plusieurs ordres – au nombre d'individus (les n souches bactériennes). Cette grande dimension rend impossible l'utilisation des modèles de régression multivariés standards.

  L'approche la plus répandue dans les études d'association est de tester indépendamment chaque variant décrit dans la matrice X, en réalisant p régressions. Elle a l'avantage de pouvoir être facilement mise en place et de permettre un calcul aisé de p-values pour chaque variant. Mais ces régressions marginales sont des modèles souvent très simples ne prenant pas en compte les distributions jointes de groupes de variants [35, 120]. Cette approche est par ailleurs très sensible à la structure

de la population, si cette dernière n'est pas inclue dans le modèle. Toutes les études d'associations chez les bactéries citées précédemment et ayant pour objectif la recherche de nouveaux marqueurs testent les variants un à un, mais en prenant en compte la structure de la population dans le modèle. C'est également ce que nous avons implémenté dans nos travaux. Réaliser un grand nombre de tests soulève par ailleurs une problématique de tests multiples: la probabilité de rejeter l'hypothèse H_0 par erreur au moins une fois, parmi les p tests, n'est pas égale à la p-value individuelle de chaque test, et cette p-value nécessite d'être ajustée [14, 78, 150]. Une manière d'adapter la régression jointe à la grande dimension est par l'utilisation des régressions pénalisées [215]. Ces méthodes permettent de contrôler un compromis entre biais (ajustement du modèle) et variance (généralisation à d'autres données), en régulant la complexité du modèle par une pénalité. Si la pénalisation permet de résoudre le problème d'estimation des coefficients du modèle, il ne fournit pas de solution générale pour le problème de test d'hypothèse, car la distribution nulle des coefficients estimés n'est pas connue. Ce type d'approche permet cependant d'adapter le modèle aux jeux de données en s'appuyant sur un apriori inspiré de la biologie du problème, en développant des pénalités dédiées, par exemple, à la corrélation entre variants.

- **Corrélation entre variants (déséquilibre de liaison).** Le déséquilibre de liaison (LD pour le terme anglais *linkage disequilibrium*) fait référence à une corrélation entre variants au sein des génomes d'une population. Dans les génomes humains, le LD est structuré par blocs le long du génome. Ces blocs sont causés par la recombinaison entre paires de chromosomes lors de la méiose. De ce fait, la notion de LD est étroitement liée à la proximité entre les variants. Plusieurs pénalités ont été spécifiquement proposées pour prendre en compte ce LD dans les régressions [54, 125]. L'idée générale de ces méthodes et d'attribuer des coefficients proches aux variants proches et/ou de sélectionner des groupes de variants voisins corrélés.

  Cependant, dans les génomes bactériens, sans recombinaison systématique, le LD n'est pas structuré de la même manière. La corrélation observée entre les variants dans les génomes bactériens est fortement liée à la structure de la population: les variants sont transmis ensemble aux individus, et se répandent dans la population. Ainsi, le LD n'est pas limité aux variants proches, et des variants distants dans le génome peuvent être très corrélés [61]. Des pénalités supposant une proximité ne sont pas pertinentes; des pénalités sélectionnant des groupes corrélés sans contrainte sur la position pourraient être intéressantes dans ce contexte [227], mais n'ont pas été testées dans ce travail. Corriger la structure de la population permet en effet de prendre en compte une grande partie des corrélations entre variants [61].

- **Corrélation entre individus (structure de population).** La prise en compte d'une éventuelle structure génétique au sein du panel de souches est essentielle pour éviter toute association confondante [61, 63]. En effet, la reproduction clonale des bactéries rend les génomes fortement corrélés par clades, ce qui constitue une source de confusion augmentant le risque d'identifier de fausses associations. De nombreuses méthodes ont été développées pour inclure la structure de population dans les études d'association chez les eucaryotes [55, 169, 182]. A partir de 2006, deux méthodes dominent: utiliser les composantes principales de la matrice des variants pour l'ajustement des variables ou comme covariables à effet fixe dans le modèle [168] (mentionnés ci-après modèles à effet fixe), et l'utilisation des modèles mixtes, dans lesquels on suppose un effet fixe pour le variant testé et un effet aléatoire pour la matrice des composantes principales [61, 218, 225].

La diversité des études d'associations ne se limite pas à ces questions, et beaucoup n'ont pas été abordées dans ces travaux, comme :

- la prise en compte simultanée de plusieurs phénotypes (résistance à plusieurs antibiotiques) afin prendre en compte les corrélations existantes entre les phénotypes (co-résistance) [226],
- les approches multi-omiques, pour chercher des variables explicatives au-delà de la séquences du génome (données épigénétiques, transcriptomiques, protéomiques...) [28],
- la prise en compte d'interactions entre variants (épistasie) [143],
- la détection des variants rares et plus généralement de l'*héritabilité manquante* [134],
- etc.

Première offensive
Le résistome de P. aeruginosa

Dans une première étude, démarrée par d'autres collègues avant ma thèse, nous nous sommes intéressés à l'ensemble des déterminants génétiques de la résistance, déjà décrits pour l'espèce *P. aeruginosa*. Ces déterminants forment le résistome de l'espèce. *P. aeruginosa* est une espèce bactérienne particulièrement adaptable à son environnement et qui est à l'origine de nombreuses infections nosocomiales. Son adaptabilité s'explique par la structure riche et mouvante de son génome – on parle de plasticité du génome. Ce génome, de taille relativement élevée (entre 5,8 à 7,6 Mpb), contient une très grande variété de gènes lui apportant une large palettes de fonctionnalités (motilité, adhésion, création de biofilm, virulence, etc.). Il contient également, avec ceux des autres espèces du genre *Pseudomonas*, la plus grande proportion de gènes de régulation chez les bactéries [109]. En plus des réseaux très complexes de régulation que la bactérie est capable de mettre en place, notamment pour ajuster l'action des pompes à efflux et des porines, elle a une très grande facilité à échanger du matériel génétique qu'elle intègre ensuite à son génome, via plusieurs mécanismes dont les plasmides, les ICE (en anglais *integrative and conjugative elements*), les transposons ou les intégrons [109].
L'étude mise en place dans ce contexte regroupait 672 souches de *P. aeruginosa* issues de trois collections et avait deux pans: l'un descriptif, l'autre quantitatif. La description du résistome a été faite en annotant dans chaque génome la présence des déterminants connus (gène complet ou mutation ponctuelle) (Figure 1.1). L'annotation des déterminants par leurs mécanismes sous-jacents a ensuite confirmé que les gènes impliqués dans l'efflux et l'influx sont naturellement présents au sein de l'espèce (forte prévalence), alors que les gènes permettant l'inactivation de l'antibiotique sont principalement acquis (faible prévalence). Par ailleurs, une partie des gènes accessoires a été trouvée dans des intégrons. Une analyse descriptive des intégrons dans l'espèce, par annotation des intégrases et des gènes trouvés en amont des intégrases, a mis en évidence la grande diversité de ces éléments, dont certains se sont largement répandus, comme In51 portant le gène *aadA6* et qui est disséminé dans différents phylogroupes (Figure 1.2).
Le pan quantitatif incluait 9 études d'association modélisant, pour 9 antibiotiques, le phénotype exprimé en catégories ordonnées de CMI à l'aide d'un modèle ordinal. Les variables explicatives étaient le comptage des gènes du résistome, et des mutations ponctuelles dans ces gènes. Une matrice décrivant les mutations dans les *core* gènes, les gènes partagés par tous les individus au sein de l'espèce, par opposition aux gènes accessoires, a été utilisée pour estimer la structure de la population. Enfin, des q-values ont été calculées,

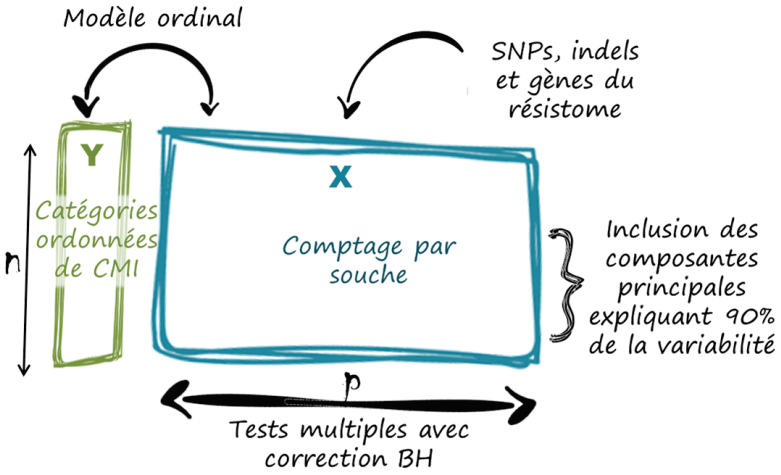


Figure c.: **Étude d'association à l'échelle du *résistome*.** Le phénotype, exprimé en catégories ordonnées de CMI, est modélisé à l'aide d'un modèle ordinal. Les variables explicatives sont le comptage des gènes du *résistome*, et des mutations ponctuelles dans ces gènes. La structure de la population est prise en compte en incluant les composantes principales de la matrice des core SNPs.

qui sont les valeurs transformées des p-values des tests multiples par la méthode Benjamini–Hochberg (BH) [14] et qui permettent un contrôle du taux de faux positifs (Figure c.). Dans la procédure de test, les déterminants connus pour un antibiotique particulier ont été inclus au préalable dans le modèle: la recherche de nouveaux loci au sein du résistome intègre déjà leur effet. Les résultats étaient mitigés selon les antibiotiques: pour certains antibiotiques, aucun déterminant attendu n'a présenté d'association significative, et de nouveaux loci n'ont été trouvés que pour 4 antibiotiques (Figure 1.3). Cela peut être imputé dans certains cas au déséquilibre des phénotypes dans le panel, à la variabilité des mesures phénotypiques [71], ou à la qualité des assemblages (Figure S1.6). Dans le cas de la levofloxacin, il est vraisemblable qu'aucun nouveau marqueur n'a été trouvé car les marqueurs existants sont très déterministes du phénotype.
Cette étude, publiée en 2017 dans la revue *International journal of antimicrobial agents* [98], a cependant permis de mettre en évidence la part importante du génome accessoire dans la résistance. Elle a également motivé le travail sur la description du génotype, en recherchant des méthodes qui permettraient de fouiller le génome accessoire de manière efficace et sans omettre les régions non-codantes qui peuvent être impliquées dans la régulation de l'expression des gènes.

Fouiller le moindre recoin
Représentation des variants par les graphes de De Bruijn

Dans cette optique, nous nous sommes intéressés aux k-mers, qui offrent une grande flexibilité pour décrire les génomes: toute séquence donnée en entrée sera fragmentée en k-mers et ajoutée à la collection de tous les k-mers décrivant tous les génomes. Cela inclut toutes les régions accessoires et non-codantes. Les k-mers peuvent aussi bien porter l'information d'une mutation ponctuelle que de l'insertion d'un gène ou d'un ré-arrangement dans le génome (Figure 2.9). Cependant le nombre de k-mers uniques observés dans une population entière devient vite très grand: plus de 60 millions de 41-mers uniques sont observés dans les 282 génomes d'un panel restreint de *P. aeruginosa* sur lequel nous nous sommes focalisés pour nos développements (Figure 2.11), et du fait de leur description par chevauchement (Figure 2.8), l'information qu'ils portent est très redondante. De plus,

leur interprétation n'est pas directe, des analyses complémentaires sont nécessaires pour retrouver à quelle région du génome les k-mers d'intérêt appartiennent, et s'ils représentent un SNP ou un autre type de variation.

Pour aider dans leur interprétation, nous avons voulu garder la trace de l'ordre dans lequel les k-mers ont été vus dans les séquences. Cela correspond à tracer une arête de chaque k-mer au suivant. Cette représentation du chevauchement entre chaînes de caractères a déjà été décrite: il s'agit des graphes de De Bruijn [52], utilisés en bioinformatique depuis que Pevzner *et. al* les a introduits, en 2001, pour l'assemblage *de novo* des reads [163]. Même si l'ordre des k-mers n'est pas totalement conservé (un k-mer répété dans un génome est représenté par un unique nœud dans le graphe et génère ainsi une boucle [163]), nous avons choisi d'explorer l'utilisation des graphes de De Bruijn dans le cadre des études d'association chez les bactéries.

Ces graphes sont utilisés comme outils pour répondre à un objectif soit d'assemblage (ABySS [21], IDBA [160]), soit, depuis 2010, d'identification de variants (KisSNP [162], KisSplice [180], DiscoSNP++ [198], Cortex [93]). Alors que les graphes sont jusqu'alors utilisés comme outils mathématiques, Iqbal *et. al* introduit en 2012 les graphes colorés, utilisés pour visualiser des variations génétiques [93]. Dans un graphe coloré, construit à l'échelle d'une population, chaque couleur représente un individu. Dans cet esprit, nous avons cherché comment enrichir un graphe de De Bruijn, représentant des variations génétiques, avec l'information phénotypique – cet enrichissement est mentionné dans la communauté des graphes comme une décoration [30].

Une preuve de concept a été réalisée en créant un graphe de De Bruijn à partir de toutes les séquences du gène *gyrA* (cible connue des fluoroquinolones, dont la levofloxacine) extraites du panel complet des génomes de *P. aeruginosa*, puis en le décorant d'une part en appliquant aux nœuds un gradient de couleur en fonction de la proportion de souches résistantes à la levofloxacine dans lesquelles la séquence représentée par le nœud était présente, et d'autre part en utilisant la fréquence d'observation de chaque séquence dans le panel complet pour déterminer la taille du nœud lui correspondant. Le graphe présenté dans la Figure 2.3 qui en a résulté nous a encouragé à exploiter cette couche d'information supplémentaire: en effet, les mutations de la région QRDR (*quinolone resistance determining region*) connues pour conférer la résistance à la levofloxacine apparaissent très clairement avec cette visualisation: les nœuds rouges représentent les allèles très présents dans les souches résistantes et les nœuds bleus ceux très présents dans les souches sensibles.

Stratégie d'attaque
Mise en place d'une étude d'association, et de son interprétation

Le lien entre le graphe de De Buijn et l'étude d'association n'est pas direct, car encore faut-il déterminer quelles seront les entités testées à partir du graphe. Dans l'exemple de *gyrA*, présenté précédemment, la région QRDR correspond à un sous-ensemble de nœuds du graphe – autrement dit à un sous-graphe. L'identification de tels sous-graphes décorés permettrait ainsi de pouvoir décrire des régions du génome associées au phénotype.

Les approches d'identification de variants par les graphes de De Bruijn ont privilégié la recherche de bulles (KisSNP, KisSplice, Cortex, DicoSNP++) ou de chemins divergents (Cortex). En effet, les bulles formées dans un graphe de De Bruijn capturent les variations entre séquences, comme illustré en Figure d.. Une bulle est flanquées de deux séquences de k nucléotides, et est définie par une paire de chemins variables. L'ouverture d'une bulle forme un chemin divergent et peut être utilisé pour identifier des bulles complexes.

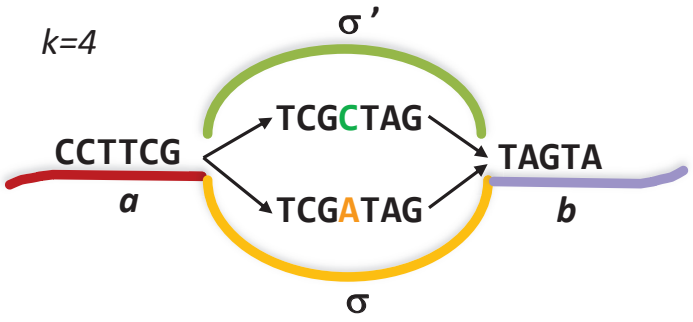


Figure d.: **Définition d'une bulle, cas d'un SNP isolé.** Une bulle dans un graphe de De Bruijn est définie par une paire de motifs $a\sigma b$ et $a\sigma' b$, où a et b sont des séquences d'au moins k nucléotides et σ et σ' des séquences qui ne partagent aucun k-mer. Dans cet exemple, $k = 4$ et σ et σ' sont des chemins représentant un SNP isolé.

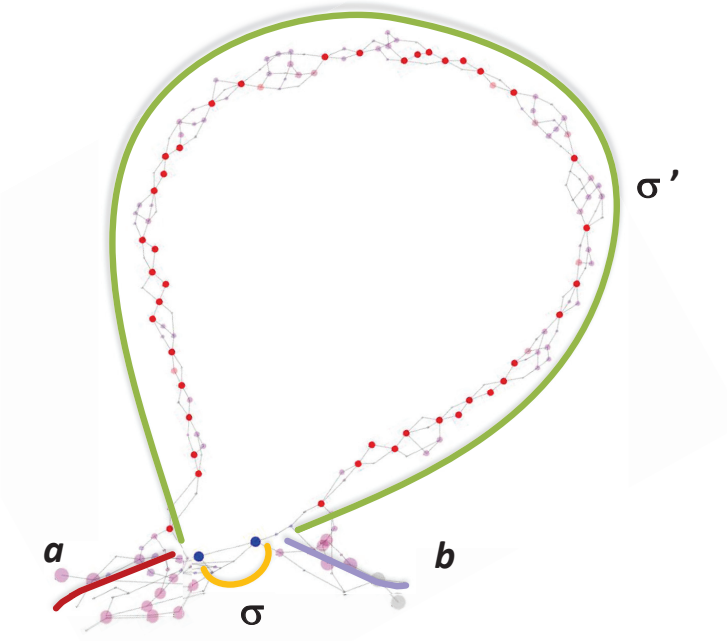


Figure e.: **Bulle complexe formée par l'insertion d'un gène accessoire polymorphe dans un site conservé.** Cet exemple a été généré avec DBGWAS, à partir d'une simulation de l'insertion d'un gène, suivant la méthode décrite en Figure S2.1, et d'un phénotype associé généré suivant le modèle proposé dans l'Équation 4.1. La couleur des nœuds est déduite de leur effet estimé sur le phénotype.

Les bulles complexes se forment lorsque qu'une nouvelle variation est rencontrée alors que la bulle ne s'est pas encore refermée: la bulle devient alors branchante. C'est le cas par exemple si deux SNPs sont distants de moins de k nucléotides (Figure 2.1D).
La Figure e. montre le graphe de De Bruijn obtenu pour l'insertion d'un gène de résistance dans un site conservé. La séquence du gène inséré est variable au sein de la population : les mutations dans le gène génèrent une succession de petites bulles, dont certaines sont branchantes. L'insertion dans son ensemble pourrait être résumée en une grande bulle: les séquences a et b de part et d'autre, le chemin σ représentant les individus dans la population qui n'ont pas l'insertion et le chemin σ' les individus ayant l'insertion. Cependant, il existe, du fait du polymorphisme du gène inséré, un très grand nombre de chemins possibles pour σ'. En nous appuyant sur l'outil KisSplice, nous avons testé une

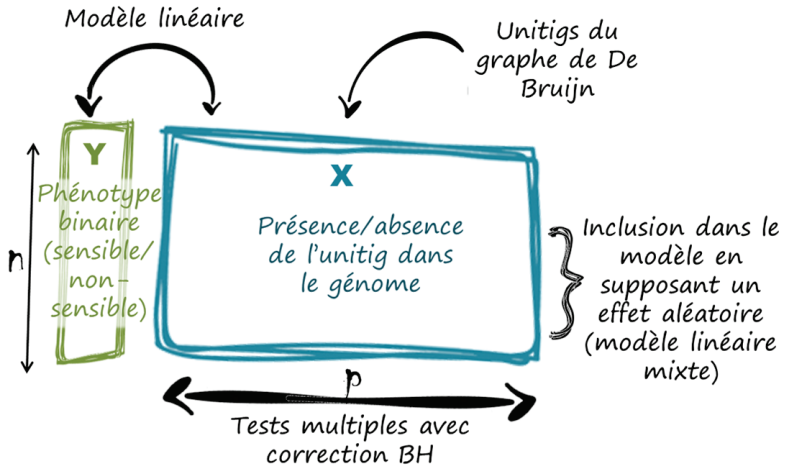


Figure f.: **Étude d'association utilisant les nœuds (unitigs) du graphe de De Bruijn.** Le phénotype binaire est modélisé à l'aide d'un modèle linéaire mixte. Les variables explicatives sont la présence/absence des séquences représentées par les unitigs dans les génomes. La structure de la population est prise en compte par le modèles mixte, pour lequel la matrice de parenté est estimée à partir de la matrice des unitigs.

approche utilisant les bulles, et effectivement constaté l'explosion combinatoire du nombre de chemins possibles pour représenter une insertion, qui elle-même, contient des SNPs. Chaque chemin décrit représente un des variants du gène qui n'est présent que dans un faible nombre d'individus: trop rarement en pratique pour être puissamment détecté avec un test.

Cependant, certains nœuds du graphes (en rouge dans la Figure e.) représentent des régions du gènes qui sont assez conservées pour contenir l'information de son insertion au niveau de la population. Ainsi, même s'ils ne représentaient pas l'entité finale souhaitée – le sous-graphe décoré – nous nous sommes intéressés à la description des génotypes par la présence ou l'absence des nœuds du graphes (appelés unitigs), sans considérer les arrêtes au moment du test. Nous avons choisi le cadre des tests multiples, et avons pris en compte la multiplicité des tests par la méthodes BH. Nous avons testé plusieurs manières de modéliser le phénotype, et comme les trois modèles testés – ordinal, linéaire et logistique – présentaient des performances de détection similaires (Figure 2.7), nous avons choisi d'utiliser le modèle linéaire, méthode la plus rapide et qui ne présentaient pas de problèmes de convergence. Pour prendre en compte la structure de la population, nous avons comparé, sur la base de simulations et de données réelles, un modèle sans correction, un modèle à effet fixe et un modèle mixte. Le modèle mixte présentait le meilleur compromis en fonction de l'existence ou non de d'effet confondant lié à la structure de la population. Ce modèle a en plus l'avantage de prendre en compte une distribution jointe des covariables. En effet, lorsqu'une variable est testée en lui supposant un effet fixe, les autres variables sont également représentées dans le modèles par les composantes principales, sous la forme d'effets aléatoires [35].

Tester des motifs uniques de présence/absence d'unitigs revient à réaliser les mêmes tests qu'avec des k-mers: la présence de l'un dans un génome induit nécessaire la présence de l'autre, les unitigs étant des assemblages locaux de k-mers. Du fait de cet assemblage, un motif unique de présence/absence correspond à moins d'unitigs (1,6 M d'unitigs sont générés pour les 282 génomes avec $k = 41$) que de k-mers (>60 M), et les unitigs représentent des séquences plus longues (Figure 2.11). Mais l'interprétation des unitigs invididuellement n'est pas non plus directe.

Cependant tester au niveau d'un sous-graphe n'est pas trivial. Alors que la présence ou l'absence d'un unitig est clairement définie dans un génome, il n'est pas évident de déterminer comment mesurer la présence d'un sous-graphe dans un génome. L'utilisation de méthodes pénalisées peut permettre de sélectionner des groupes de nœuds proches dans le graphes, en pénalisant les différences d'effet estimés pour des nœuds proches [119, 133]. Mais ces méthodes ne sélectionnent pas non plus explicitement des sous-graphes, potentiellement branchants, et ne pourraient pas permettre de contrôle du taux de faux positifs au niveau des sous-graphes. De plus elles ne sont pas toujours directement applicables aux graphes de De Bruijn, qui contiennent des cycles.
Bien que le test mis en place se fasse au niveau des unitigs, nous avons montré que son interprétation pouvait se faire au niveau des sous-graphes. En effet la contribution originale principale de la thèse a été le post-traitement de cette analyse, s'appuyant sur le graphe de De Bruijn construit initialement, et qui a permis de re-créer a posteriori des sous-graphes décorés interprétables, de la même manière que celui construit dans l'exemple du gène *gyrA*. L'analyse à grande échelle attribue une q-value à chaque nœud du graphe. Le sous-ensemble des nœuds avec les plus faibles q-values est replacé dans le graphe initial et le voisinage de ces nœuds est extrait. Le voisinage est défini comme tous les nœuds distants d'au plus *ne* arrêtes. Le sous-graphe induit par les nœuds les plus associés et leurs nœuds voisins est composé de plusieurs composantes connexes, et nous avons fait l'hypothèse que ces composantes connexes pouvaient représenter des régions génomiques distinctes (Figures S2.8 et S2.9). En effet, si deux nœuds proches dans le graphes sont sélectionnés ensemble, leurs voisinages se chevauchent et ils se retrouvent dans la même composante connexe (Figure 3.6). Pour plus de simplicité, nous avons nommé chacune de ces composantes un *sous-graphe*. Ces sous-graphes décorés offrent un cadre d'interprétation plus aisé:

- les polymorphismes locaux sont représentés par des nœuds bleus et des nœuds rouges, comme présenté dans la Figure 2.3,
- les insertions (resp. délétions) de régions sont représentées par un sous-graphe linéaire constitué majoritairement de nœuds rouges (resp. bleus) (Figure 3.3D,E),
- l'annotation des nœuds avec le nom de domaines fonctionnels informe si le polymorphisme est dans la séquence du gène, dans sa région promotrice (Figure 3.3C), ou dans une région intergénique.

De plus, le regroupement des nœuds fortement associés dans une même entité (le sous-graphe) permet de synthétiser l'information et de prendre indirectement en compte le LD local. La présentation des résultats se fait par sous-graphe – qui dans la pratique représente en effet très souvent une région particulière du génome – et non pas par entité testée. Cette représentation en sous-graphe permet également de visualiser le contexte de l'insertion d'un gène causal (cassette SCC*mec* portant le gène *mecA*, Figure 3.3D, plasmide portant le gène *ermC*, Figure 3.3E). Par ailleurs, le fait d'annoter l'ensemble des nœuds du sous-graphes et non pas seul le nœud fortement associé permet de consolider l'annotation fonctionnelle.

Déploiement à grande échelle
Développement, utilisation et publication de l'outil DBGWAS

Devant l'intérêt suscité par cette représentation des résultats, et en collaboration avec l'équipe LBBE/Erable Inria, nous avons développé un outil computationnellement efficace pour mettre en œuvre notre méthode en trois étapes : la première étape, qui construit le

graphe de De Bruijn à partir des assemblages, puis la matrice X des variants (présence/absence des unitigs du graphes dans les génomes) s'appuie sur la librairie C++ GATB [57]. La deuxième étape effectue les tests d'association à l'aide de modèles linéaires mixtes et s'appuie sur la librairie R bugwas [61]. Enfin, la troisième étape, qui effectue le post-traitement et affiche les sous-graphes décorés et ordonnés par q-value croissante, utilise la librairie javascript cytoscape.js [67] (Figure 3.2). Nous avons étoffé la sortie graphique et élargi les analyses à deux autres espèces présentant des structures de génomes différentes: *M. tuberculosis*, espèce très clonale et *S. aureus*, sujette aux transferts horizontaux.
L'outil créé, DBGWAS, a été comparé à deux autres méthodes publiées après 2016 et basées sur des k-mers pour tester des associations. Cette étude a montré que l'implémentation de DBGWAS était très compétitive. Nous présentons DBGWAS et son utilisation pour l'analyse de l'antibiorésistance pour les trois espèces bactériennes dans un article accepté pour publication en 2018 dans PLOS Genetics [99].

Plus loin vers l'interprétation
Prédiction des labels des graphes de DBGWAS

La topologie de la plupart des sous-graphes décorés peut facilement être associée à une catégorie de variant génétique (SNP, insertion d'élément mobile (MGE, pour *mobile genetic element*)), en particulier pour les espèces *S. aureus* et *M. tuberculosis*. Cependant, certains graphes obtenus pour *P. aeruginosa* et notamment pour l'étude de sa résistance à l'amikacine sont très branchants et moins facilement interprétables. Pour aider à leur lecture, et pour fournir aux utilisateurs de DBGWAS une aide générale à la lecture des sous-graphes, nous avons développé une méthode de prédiction de la catégorie des variant associés aux sous-graphes. Nous nous sommes focalisés sur les labels simples: polymorphisme local (label LP) et insertion ou délétion de MGE (label MGE), et avons conçu un ensemble de descripteurs des sous-graphes pour entrainer un moteur d'apprentissage. Nous l'avons alimenté avec des sous-graphes labellisés LP ou MGE, obtenus dans l'analyse précédemment évoquée sur les trois espèces bactériennes, et avec des sous-graphes simulés, représentant spécifiquement les deux catégories de variants. Une approche classique par validation croisée a permis de sélectionner le meilleur modèle parmi six méthodes très répandues: trois régressions pénalisées, deux machines à vecteur de support, et des forêts aléatoires. Les forêts aléatoires ont fourni les meilleures performances en validation croisée, et leurs performances sur un jeu de test indépendant étaient en moyenne de 97% sur données simulées et 92% sur données réelles. Le modèle sélectionné a été utilisé pour prédire les labels des sous-graphes étudiés dans le Chapitre 3, et pour étudier l'intérêt de diminuer la taille du voisinage dans la construction des sous-graphes. Ce modèle pourra être inclu à DBGWAS comme aide à l'interprétation.

La lutte continue, ne déposons pas les armes
Discussion et conclusions

Dans ce travail de thèse, nous nous sommes intéressés à l'application des études d'association aux génomes bactériens, pour l'étude de l'antibiorésistance.
Nous avons testé différentes manières de modéliser le phénotype, sans observer de différences significatives en termes d'identification des marqueurs existants. Nous avons évalué la meilleure manière de prendre en compte les structures de population dans une étude chez *P. aeruginosa* et avons confirmé le potentiel des modèles linéaires mixtes qui offraient dans nos expériences le meilleur compromis pour différentes configurations de structures de population. Afin d'élargir l'espace de recherche des associations, nous avons cherché comment améliorer la représentation des variations génétiques dans les génomes

bactériens particulièrement plastiques. Les k-mers offrent la flexibilité nécessaire mais sont très redondants et ne permettent pas une interprétation directe des résultats obtenus. Les bulles d'un graphe de De Bruijn capturent des variations simples mais ne permettent pas d'identifier des insertions polymorphes.
La méthode que nous avons mise au point teste la présence ou l'absence des séquences représentées par les nœuds d'un graphe de De Bruijn construit à partir de tous les génomes de la population étudiée. Nous avons concentré nos efforts sur le post-traitement de cette analyse, en proposant une visualisation du contexte génomique des nœuds fortement associés au phénotype. Cette vue synthétique permet d'appréhender la séquence d'intérêt dans son ensemble et renseigne sur sa nature: polymorphisme local dans un gène, acquisition d'un gène par un plasmide, etc. Nous avons publié un outil clé en main implémentant cette méthode, qui peut être étendu pour apporter plus de fonctionnalités : détection automatique de la nature de la séquence identifiée par le test, flexibilité dans le choix du test d'association, analyse du déséquilibre de liaison au sein et entre sous-graphes, définition d'un contrôle du taux de faux positifs au niveau des sous-graphes, etc.
La méthode proposée peut être appliqué à d'autres phénotypes et aux eucaryotes, en adaptant notamment la mesure du génotype. La visualisation que nous avons proposée peut également s'appliquer à l'interprétation des modèles de prédiction basés sur les k-mers, notamment pour la prédiction de l'antibiorésistance à partir d'un génotype.
La lutte contre la (multi) résistance aux antibiotiques sera longue et difficile. Chaque petit pas peut compter.

Avant-propos et remerciements

Je crois que la recherche est un chemin. Ce qu'on trouve au passage nous conduit vers de nouvelles questions et de nouvelles pistes de recherche. La thèse a cela d'intéressant et d'utile qu'elle a une fin. À ce moment, il faut refermer les portes entrouvertes, parfois rebrousser chemin, quitter les impasses, et partager la route parcourue.

Beaucoup de personnes ont contribué de près ou de loin à la réalisation de ce travail, par leur soutien, leurs conseils, les discussions, leurs réalisations. Je tiens à les remercier de m'avoir aidée à avancer et à progresser.
Jean-Baptiste merci pour ta confiance, je ne serai pas là aujourd'hui si tu n'avais cru en moi pour réaliser ce projet. Franck, merci de m'avoir accueillie au sein de ton équipe, j'ai eu beaucoup de plaisir à y travailler. Merci de ton soutien, de ta disponibilité et de tes conseils. Laurent, tu es un directeur tip top, j'envie tes prochains thésards ! Merci de ta patience quand il s'agissait de m'expliquer, me ré-expliquer et me re-ré-expliquer les bases des statistiques en grandes dimensions – je crois que ça n'a pas été vain. Merci pour les discussions toujours fructueuses devant ton tableau, pour ton implication. Leandro, sem ti este projeto não teria sido o mesmo. Acho que podemos orgulhar-nos do que realizámos. Obrigada pelo trabalho incrível que fizeste, por todas as horas passadas no desenvolvimento, e depois no benchmark e na escrita. Maud, merci pour ta rigueur et ton exigence, ta disponibilité et tes conseils. On a encore de beaux projets à réaliser. Vincent, après chaque discussion avec toi, toutes pertinentes, j'ai eu l'impression de faire un bond en avant ! La confrontation des idées permet d'aller plus loin. Alex, het was geen gewonnen strijd, maar het is me gelukt om jou te overtuigen van de voordelen van onze methode! Bedankt voor je openheid en veeleisendheid, het heeft ervoor gezorgd dat ik mezelf kon overtreffen. Ik reken op jou om reclame te maken voor DBGWAS! Pierre, c'est toujours un plaisir d'échanger avec toi, de chercher, de comprendre. Merci pour toutes tes explications ! Dany, I really enjoyed my visit in Oxford to get a deeper knowledge of bugwas. Thank you for welcoming me in your team, it was a great experience. A tous mes collègues de bioMérieux pour leurs encouragements sur la dernière ligne droite. Ça compte énormément. Audrey, Coralie (quelle formidable idée de nous avoir (re)mis au dessin), Meriem, Nathalie, Aurélien, Thierry, Stéphane, Stéphanie, Marie, Ghislaine (pour me traîner malgré tout au bad, promis on y va lundi!), Nadège, les bioMaths, Bertrand, Gaël, JF, Thomas, Mohamed, Sophie, Philippine, Françoise, Arthur, Guillaume, Bruno, Romain, Brice (on reparlera graphes bientôt j'espère !), Anthommy, Gaspard... et à tous ceux de la fac, Anne-Laure merci de m'avoir remise aux stats, Aurélie voisine de bureau attentive et discrète, Anna tu m'as fait sortir de mon trou ;), Jean-Pierre, Manolo, Dexiong, l'équipe du pôle info, vous m'avez bien aidée, Héloïse, Marie, Christine, Amandine, Laurent, Ghislain, Florian, Philippe, Hélène, Diego, Thibault, Anouk, Maud... 13h forever !
Sylvain merci pour ton soutien inconditionnel depuis des années. Il m'est si précieux. A mes enfants, vous m'avez apporté tellement plus d'énergie que vous m'en avez pris ;) Merci à vous et à ma grande et belle famille, (grands-)(beaux-)parents, (belles-)sœurs et beaux-frères qui m'avez encouragée dans l'aventure dans laquelle je me suis lancée. Anne-Claire, Joana, les Bimettes, Ronath, Roshub, l'amitié donne des ailes ! A Lysa, Noémie, Virginie, et les parents marcyllois et de la PIEP, merci pour ce réseau de solidarité si précieux dans l'organisation du quotidien. Merci à l'équipe Metal de MATEIS pour son accueil à midi et ses encouragements autour d'un bon verre.

Je remercie le comité de pilotage pour son écoute, ses conseils, ses propositions de piste de travail. Merci à tous les membres du jury d'avoir accepté de vous investir dans ce travail.

à propos de l'antibio-résistance:

"Il n'y a probablement pas de molécule chimiothérapeutique à laquelle,
dans des circonstances appropriées, la bactérie ne peut pas réagir
en acquérant d'une certaine manière une résistance."
– Alexander Fleming, 1946

"La bataille sera excessivement longue et dure,
l'approche nécessairement sans frontière et multi-disciplinaire."
– Alain Mérieux, 2017

à propos du travail d'écriture:

"Avant donc que d'écrire, apprenez à penser.
Ce que l'on conçoit bien s'énonce clairement,
Et les mots pour le dire arrivent aisément.
Hâtez-vous lentement, et sans perdre courage,
Vingt fois sur le métier remettez votre ouvrage,
Polissez-le sans cesse, et le repolissez,
Ajoutez quelquefois, et souvent effacez."
– Nicolas Boileau, L'Art poétique

À Sylvain.
À Louis, Alban & Camille, mes enfants chéris.
À mes parents et à mes grands-parents, continuelles sources d'inspiration.

Contents

Introduction

Antibiotic resistance

Ninety years ago, Alexander Fleming discovered penicillin and opened a path to a new and effective type of treatment against infectious diseases: the application of antibiotics. However under the selective pressure of these molecules the bacteria population evolved rapidly, illustrating Darwin's theory of evolution: adapt or disappear. Within a few years, the first bacterial strains not susceptible to penicillin were observed and, in 1945, Fleming warned against misuse of antibiotics: by increasing the selective pressure, the bacteria would become more resistant and the antibiotics less effective.
After several decades of successful use of these drugs – which became indispensable to modern medicine – the race against time started: every new antibiotic introduced led to new resistant strains a few years later. Since the 1990s, no new antibiotic family has been discovered. Is the race lost?
In 2014, the World Health Organisation published an alarming report on the global emergence of antimicrobial resistance: the proportion of microbial strains that had developed resistance mechanisms to one or more antimicrobials was growing dangerously, especially in low-income countries [212]. A study carried out over the 2000-2015 period linked the growing consumption of antibiotics to the increase of multi-drug resistance [103].
Today, experts refer to a post-antibiotic era, and the forecasts are very pessimistic: the worldwide spread of resistant and multi-resistant bacterial strains challenges our antibiotic arsenal. Trivial surgeries or banal urinary tract infections could be fatal without antibiotics. It is estimated that in 2050, there may be 10 million deaths per year caused by infectious diseases, more than cancer victims [154].
Today the acceleration and spread of antimicrobial resistance is one of the most serious threats to global health. Although in some situations the decrease of antibiotic consumption has been related to the decrease of the prevalence of resistant strains [53, 211], many studies are pessimistic about the reversibility of resistance [7, 84, 89] and it is highly probable that we will not be able to eradicate the phenomenon.

Antibiotic actions and resistance mechanisms

Antibiotics act on different aspects of the bacteria life, and either destroy cells or inhibit their growth. For instance, penicillin, methicillin, cefepime, meropenem and other β-lactams, as well as fosfomycin and glycopeptides perturb correct synthesis of the cell wall. While β-lactams inactivate the penicillin-binding proteins (coded by the *pbp* genes) which catalyse the cross-linking of the peptidoglycan, a major component of the cell wall, glycopeptides such as vancomycin block the *pbp* target, also preventing cross-linking. On the other hand, antibiotics of the quinolone family, such as ciprofloxacin, levofloxacin or ofloxacin, block the DNA replication by preventing DNA unwinding: they inhibit the topoisomerase proteins (coded by *gyrA* and *gyrB* or *parC* and *parE* genes) which participate in DNA winding. As a last example, amikacin, kanamycin, streptomycin, gentamicin

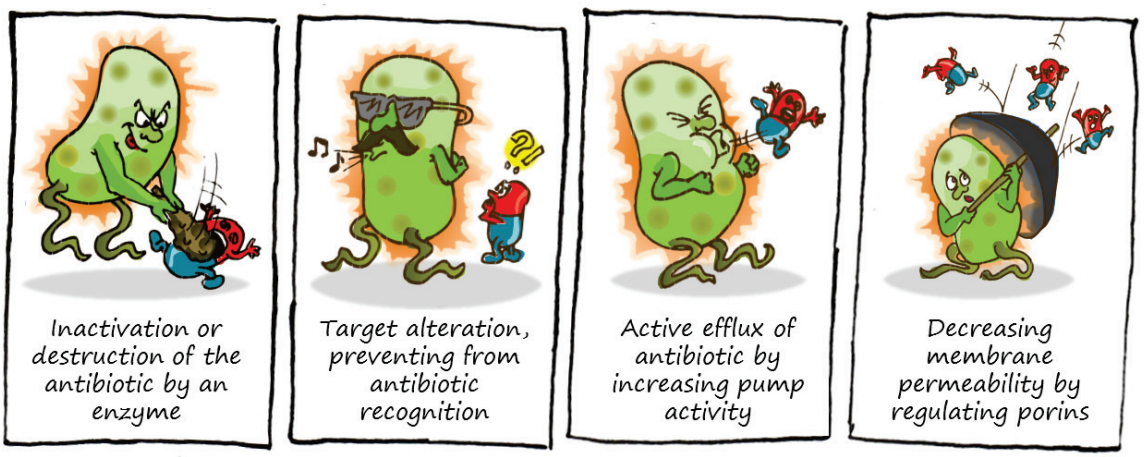


Figure 1: **Mechanisms of resistance** Author's view.

and other aminoglycosides, as well as fusidic acids and chloramphenicols inhibit protein synthesis by blocking the ribosome activity [203].
Depending on the antibiotic family, and on the species they belong to, the bacteria act in different ways to resist [24]. Resistant organisms can inhibit the antibiotic activity, modify the antibiotic target so it is not recognised any more, reduce the membrane permeability – and thus the penetration of the antibiotic in the cell, or allow enhanced efflux of the antibiotic (Figure 1). For a given drug, the preferred mechanism depends on the bacterial species. In the case of β-lactam resistance, for instance, bacteria of the Enterobacteriaceae family mostly degrade the β-lactam antibiotic with an intrinsic or acquired β-lactamase enzyme, while bacteria belonging to the Staphylococcaceae and Streptococcaceae families rather modify the *pbp* target by replacement (*mecA* gene acquired by *Staphylococcus aureus*) or recombination (*pbp2x* mosaic genes in *Streptococcus pneumoniae*) [43, 130].
Some resistances are intrinsic. For example, the cell membrane of Gram negative bacteria provides them with a structural resistance to vancomycin: this membrane is impermeable to the large glycopeptide molecule. In the same way, efflux pumps are naturally present in many Gram negative species. However, many resistance mechanisms are due to changes in the bacterial genome.

Acquisition of resistance: local polymorphism and horizontal gene transfer

Various genome modifications may confer antibiotic resistance [4]. Resistance can result from the acquisition of one or more resistance genes by horizontal gene transfer (HGT) – as opposed to the 'vertical' contribution of asexual reproduction – or be the consequence of local point mutations, such as SNPs (single nucleotide polymorphism, punctual permutation of nucleotide) or indels (insertion or deletion of one or more nucleotides), which may become fixed in the population under antibiotic selective pressure [192].
Highly clonal species such as *Mycobacterium tuberculosis* do not acquire new genes by HGT: in these species, resistance is mainly related to mutations in the genes or their promoter regions. Other species may have a great propensity to exchange genetic material [189]: as an example, the *mcr*-1 gene conferring resistance to colistin has recently been identified in the *Escherichia coli* species, then a transfer of the gene to *Klebsiella pneumoniae* and *Pseudomonas aeruginosa* species was discovered [126]. Three mechanisms have been widely described for horizontal transfer of genes [109, 189, 194]. First,

natural transformation allows competent bacteria to integrate extracellular 'naked' DNA – possibly remaining from a decomposing cell or from the active excretion of DNA by another cell. Second, conjugation, or conjugative transfer, refers to the transfer of genetic material by cell-to-cell contact, often through a pilus linking a donor cell to a recipient cell. Third, transduction involves DNA transfer via a bacteriophage: the bacterial virus infects a first host and integrates into its chromosome. It is then excised possibly with a part of the first host genome. This extra non-phage DNA can then be transferred to a second host after a new infection cycle. Alternative mechanisms have been recently described, among them, the extracellular vesicle-mediated transfer facilitates the exchange of large genetic cargo [194].

All these HGT mechanisms allow for the transfer of mobile genetic elements (MGE) carrying resistance genes and thus play an important role in antibiotic resistance dissemination. Plasmids are circular MGE sequences capable of replication which are acquired by conjugation. They contain transposable elements (transposons, insertion sequences) and integrons, which carry gene cargo. Integrons are not mobile by themselves but are usually carried by MGE. They integrate genes upstream their integrase [109]. Integrative and conjugative elements (ICE), acquired by conjugation and able to integrate the host chromosome via site-specific recombination, are MGE found frequently in *P. aeruginosa* genome. Material acquired by transduction is referred to prophages or phage-like elements. All this dispensable genetic material, differentially acquired within the species, constitutes the species accessory genome, as opposed to the *core* genome, which is shared within the species [109]. The ubiquitous species *P. aeruginosa* makes frequent use of HGT to adapt to its environment, and has a particularly large accessory genome.

The resistance mechanisms described in the previous section may have several genetic basis. For instance, mechanisms inhibiting the antibiotic activity may be caused by the acquisition via HGT of an antibiotic-inactivating enzyme. Such enzyme may acquire SNPs and indels in its active site, which would increase its affinity with the antibiotic [113] and thus increase the bacteria resistance level. Mechanisms modifying the antibiotic target may be caused by the acquisition of SNPs and indels in the target sequence – often in a particular, polymorphic region of the gene – for instance in the quinolone resistance determining region (QRDR) of the *gyrA* gene [90], or can be caused by the HGT acquisition of a gene coding for a protein providing the same function as the target, however presenting a different conformation not recognised by the antibiotic. For instance the *mecA* gene, acquired within a MGE cassette, replaces the methicillin target in *S. aureus*. Genes coding for efflux pumps may be acquired by HGT. SNPs and indels acquired in the genes or gene promoters involved in regulation may improve the efflux pump activity by optimising their regulation network [124].

Efflux pumps, which are widely used by *P. aeruginosa* strains, actively reject the toxic compounds outside the cell, and can be effective against different families of antibiotics. Porins regulate the permeability of the cells and are not specific to one antibiotic family. Porins and pumps are naturally present in *P. aeruginosa* which, in the presence of antibiotic, use complex networks to regulate or co-regulate their expression [124]. Some aspects of the genetic basis of such complex mechanisms, which may involve regulatory systems of gene expression, or multiple factors including combination or interaction of several genes, remain unclear [89, 112]. The formation of biofilms is another complex mechanism that is not yet fully understood [132, 171].

Towards a better understanding of resistance mechanisms and their genetic basis

Resistance mechanisms have been studied since the observation of the first resistances in the 1940s. Websites such as `card.mcmaster.ca` [141] or `megares.meglab.org` [111] try to gather all bacterial genome regions known or suspected to be involved in antibiotic resistance. However, a part of the phenotype variability remains unexplained. Moreover, the resistance mechanisms evolve and diversify, requiring the constant improvement of our knowledge.
In this context, my thesis project aims at developing methods and tools to complete and improve the knowledge of the genetic determinants of antibiotic resistance. Beyond the thesis, the products of this work could be used to refine the database of a diagnostic system, in order to optimise the antibiotherapy of a patient [56, 157].
The challenge is, therefore, to identify the regions of the genome involved in the antibiotic resistance, annotate them, and quantify their impact on the phenotype. Association studies, linking a phenotype (the antimicrobial resistance) and a genotype (the information contained in the genomes) provide a suitable framework to achieve this genetic mapping. In 2006, Daniel Falush paved the way for these studies in bacteria – before that they were mostly applied to eukaryotic organisms – by enumerating major obstacles to first overcome. Genome-wide association studies (GWAS) could revolutionise practices in microbiology if good solutions are found in particular to estimating correctly the bacterial population structures, and addressing most genomic variations, in particular those related to gene regulation. [63].
GWAS in bacteria really took off in 2012. Between 2012 and 2015, the majority of these studies [3, 43, 110, 202] were based on approaches developed for eukaryotic organisms: first, a search for SNPs relatively to a reference genome and then, an association model taking into account the structure of the population [170, 190]. One study, however, proposed a phylogenetic convergence test dedicated to bacteria [64], while another proposed to describe the genomic variation with k-mers (all the fragments of k nucleotides contained in the sequence of the genome) rather than with a reference genome [183]. From 2015, other studies have followed with the use of k-mers, either to predict the phenotype [58] or to discover new markers [61, 117].
At the beginning of my thesis in 2015, I contributed to a literature review on antimicrobial susceptibility testing, by providing a summary description of these bacterial association studies [59].

High throughput sequencing: an opportunity for bacterial GWAS

The genotype of a bacterial strain is represented by the particular genetic variations in its genome, by comparison to a reference genome or to a population of genomes. The primary data for its computation are a full genome sequence obtained by sequencing.
There are currently three major high throughput sequencing technologies: the optical detection, during DNA synthesis, of specific fluorescent labels fixed to the nucleotides (principle of the Illumina® sequencers [17]), the detection of potential differences specific to each nucleotide, which are related to the emission of hydrogen ions during the synthesis of DNA (principle of the Ion Torrent™ sequencers [178]), and the detection of the fluctuations of a specific electrical current of the DNA sequence that passes through a nanopore (principle of the Oxford Nanopore sequencers [31]).
These technologies all produce a very large number of inferred sequences from DNA fragments, called reads. Read length and error rate vary greatly across technologies [172].

The genome sequence is obtained by assembling the reads into longer fragments, called contigs [21, 160]. The contigs can possibly be reassembled until obtaining a single sequence describing a complete biological entity such as a chromosome or a plasmid [72, 104]. Finally, genome sequences can be annotated with the positions of functional domains, such as genes and regulatory regions [155, 181].
Since its advent a dozen years ago, high throughput sequencing technologies have progressively imposed themselves in the microbiology field [107, 129]. It is now possible to obtain the assembled and annotated genome of several microorganisms simultaneously in a relatively short time span and at low cost [56]. This technical progress made it possible to build panels of bacterial strains for which the whole genome sequence, and a collection of phenotypic measures are available.
Early bacterial association studies included a couple of dozens of genomes at most (50 in 2012 [219], between 60 and 160 the following year [64, 183, 187]), then, from 2014, panels of several hundreds or even several thousands of strains were studied, especially for *S. pneumoniae* [43], *M. tuberculosis* [61, 202] and *S. aureus* [61, 79].

Methodological framework of genotype to phenotype studies

Although genotype to phenotype association studies are recent in prokaryotes, they have a long history in eukaryotes and especially in human genetics [33]. Our work is based on these developments and aims at making them more suitable for prokaryotic studies. We have restricted our scope to association models based on regressions [185]. Regressions allow to model the relationship between variables, here between a phenotype and genetic variation. This model is estimated from n observations. Let us consider for instance a linear relationship between the phenotype y_i and the p genetic variants represented in X_i, for bacterial strain i:

$$y_i = \beta^\top X_i + \varepsilon_i, \quad i = 1, \ldots, n, \tag{1}$$

with $\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, $\sigma^2 > 0$, the random error and β the effects of the genetic variants on the phenotype. The objective is to provide estimates $\hat{\beta}$ of the coefficients β minimising the error $\varepsilon_i = y_i - \hat{\beta}^\top X_i$. A common approach is to minimise the sum of squared errors over the n observed samples.
The regression can then be written as a minimisation problem:

$$\min_\beta ||y - X\beta||^2 = \min_\beta \sum_{i=1}^{n} L(y_i, f(X_i)) \tag{2}$$

where $L(y_i, f(X_i))$ is a loss function, and in the particular case of the linear regression, $f(X_i) = \beta^\top X_i$. An analytical solution of the linear problem is given by the normal equation:

$$\hat{\beta} = (X^\top X)^{-1} X^\top y. \tag{3}$$

Regression models can, with the same input data, meet two objectives: prediction of a phenotype value, or statistical inference aiming at discovering markers and estimating their effect.
In the first case, the objective is to optimise the generalisation performance of the phenotype prediction, *i.e.* to predict on other datasets than the one used to build the model. In this case, the interpretation of the variables used for the model and their coefficients, is secondary.

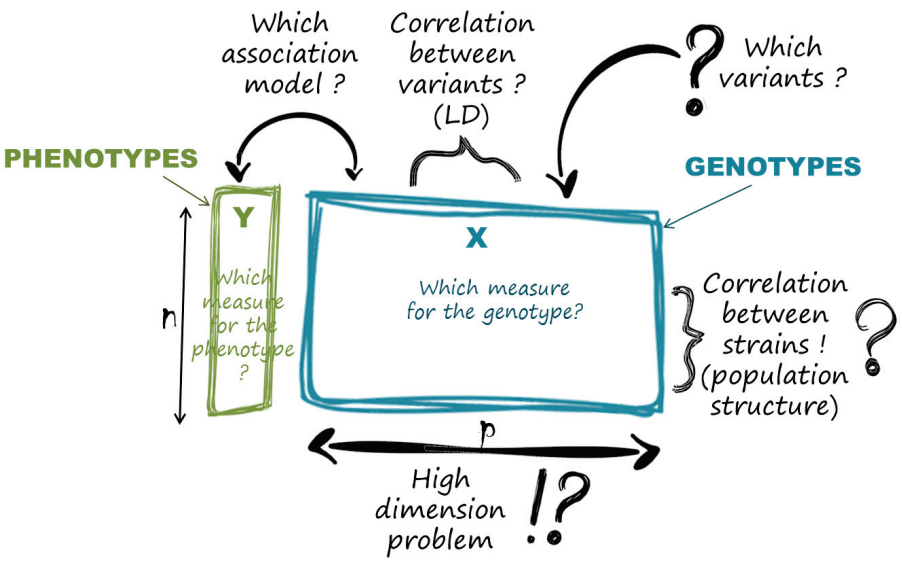


Figure 2: **Setting up an association study.** Several questions need to be addressed in order to estimate the relationship between the phenotype (Y vector, providing for each of the n strains a measure of the antimicrobial resistance) and the genotype (X matrix, describing for each strain the set of p variations in its genome).

In the second case – our case here – the objective is to identify the variables which are associated with the phenotype. This produces hypotheses which can then be experimentally validated, to better understand the phenotype. The interpretation of the identified variables is therefore of great importance. The significance of the association of the variables to the phenotype is assessed using a hypothesis testing procedure: under the null hypothesis H_0, the coefficients of the regression, describing each variable effect, are equal to zero. The p-value computed by the test expresses the probability that the null hypothesis is rejected by mistake (the test concludes that the coefficient is non-zero while this is not the case), and is a common measure of the significance of an association.
In this context, the challenge of an association study is therefore to identify significant associations from several thousands or even millions of explanatory variables. Several questions arise when setting up such a study (Figure 2).

Measuring and modelling the phenotype

The international measure for quantifying our studied phenotype – the level of resistance of a bacterial strain – is the minimal inhibitory concentration (MIC). It is defined as the lowest concentration of antibiotic inhibiting the bacterial growth after a night of incubation [8]. The raw MIC data can be a concentration (*e.g.* $8\mu g/mL$) or a concentration class (*e.g.* $\geqslant 4\mu g/mL$).
Institutes such as the CLSI (Clinical & Laboratory Standards Institute) or the EUCAST (European Committee on Antimicrobial Susceptibility Testing) provide standards for measuring and interpreting these MICs, including thresholds to define if the bacteria are resistant, intermediate or susceptible to an antibiotic. This information is directly used by clinicians to adapt their antibiotherapy. There are several methods for measuring the MIC, including reference methods such as microdilution, or automated systems allowing high throughput analyses, such as the bioMérieux system Vitek®2 [177].
The antibiotic resistance phenotype could be classified as 'simple' relatively to other phe-

notypes such as the virulence. Indeed, its definition is standardised and there are reference methods for its quantitative measurement. Moreover we can expect strong phenotype-genotype associations for the genuine resistance variants with a straight causality, *e.g.* mutations acquired in *gyrA* QRDR [90] or the acquisition of *mecA* gene [130].
The phenotype, when given as a MIC, can be represented by its membership to an ordered category of concentrations (cf. Supplementary Figure S1.5). It can also be expressed by a transformed value of the MIC: its logarithm [62] or a binary value computed from the CLSI or EUCAST thresholds. The previously cited bacterial GWAS studying the antibiotic resistance all used a binary phenotype (susceptible *versus* resistant or susceptible *versus* non-susceptible).
Depending on the antibiotic and on the studied species, the measures made with a given method are not always reproducible [71] and can vary between methods [6, 85]. A raw MIC is richer to exploit: variations in the phenotypic trait within a population of susceptible (resp. resistant) strains can be identified, however the use of a simplified phenotype (*e.g.* a binary status: susceptible *versus* non-susceptible) is widespread as this is the tractable information for the clinicians.
The measure chosen for the phenotype then drives the choice of the association model: an ordinal model will be adapted to ordered categorical data [142], a linear model will be used for continuous numerical data [185], a logistic model for binary data [49].
In the following manuscript, different ways to measure and model the phenotype were tested and discussed in Chapters 1 and 2.

Measuring the genotype

Most human GWAS use values in $\{0; 1; 2\}$ to encode the SNPs on the two alleles present in each diploid individual: 0 encodes homozygous alleles carrying the major allele seen in the population, 1 encodes heterozygous alleles, and 2 encodes homozygous alleles with the minor allele [33, 88].
The transposition of this encoding is not straightforward as bacterial genomes are haploid (presence of only one allele per individual). Moreover, at the level of the complete bacterial population, more than two alleles can be observed for a given SNP. Encoding can in this case be done using several binary variables, describing each allele.
More generally – beyond a particular genotype description such as SNPs – the genotype can be measured with a binary value (presence or absence of the genotype in the strain), or with a count value (number of observations within each individual). Counts can indeed provide a valuable information on the number of copies of genetic elements, such as plasmids or repeated genes. While the read depth coverage can provide a good basis for the estimation of the sequence copy numbers [145, 217], the assembled sequences only provide a partial information, as a part of the repetitions are suppressed in the *de novo* assembly process [163].
All previously cited bacterial GWAS use binary values to measure the genotype. The best way to measure the genotype was not explicitly addressed in this thesis, however the study presented in Chapter 1 used a count value computed from sequence assemblies while the method presented in Chapters 2 and 3 used binary vectors of genotype presence/absence.

Description of the genotype by genetic variants

In the human GWAS literature, genetic variations are described at the the SNP level [1, 33, 92]. The International Haplotype Map (HapMap) project [92] reduced the number of SNPs to investigate by proposing tag SNPs, which are markers of specific combinations of SNPs always found together in blocks: the haplotypes.

SNP-level studies are not appropriate for bacterial GWAS. Indeed, a large part of the bacterial genomic variability lies in HGT [109], and as a consequence genomes belonging to a given species do not all share the same composition in terms of genes and intergenic regions. A description based only on SNPs cannot describe the diversity of this accessory genome [175]. A majority of the first bacterial GWAS used methods imported from human GWAS [170]. These methods call local polymorphisms against a reference genome. As the use of a reference genome reaches limits to describe the variability of the accessory genome, some studies proposed to test the presence/absence of genes in addition to the SNPs, or to test the presence/absence of k-mers [61, 183].

K-mers offer the required flexibility. Indeed any k-mer, *i.e.* any sub-sequence of k nucleotides, observed in the genomes can be used to represent the genotypes in the GWAS: each genome is described by a binary (or count) vector indicating whether it contains each k-mer. All these sequence descriptors can carry information for instance on SNPs and indels, on genes which are differentially present in the population, or on rearrangements. However, they present several drawbacks such as their high number, high correlation, high level of redundancy (by construction using a sliding window of step 1, a k-mer overlaps at least two other k-mers with an overlap length of $k-1$ nt), and contrary to SNP and gene-based analysis, they do not offer a straightforward interpretation. In particular, it is not specified on which gene they were found, or if they represent a SNP or another type of variation.

The description of the genotype was a central question in this thesis, and in the works presented in Chapters 2 and 3, we chose to use k-mers for their flexibility. However, we searched how to take advantage of the overlaps to help in their interpretation. Indeed, we wanted to keep track of the order in which the k-mers were seen in the sequences, by drawing an edge from each k-mer to the following one. This representation of overlaps between strings was already described as De Bruijn graphs (DBG) [52]. Even though the order of the k-mers is not totally conserved in a DBG – repeated k-mers are collapsed into a single node and generate loops [163], we chose to use the DBG framework to build the list of variants used in the GWAS.

In De Bruijn's early description, the graph includes all possible strings of length k. The number of nodes in the graph is constant for a given alphabet of size N and length k, and equals N^k. For instance, enumerating all 25-mers with an alphabet containing 4 nucleotides $\mathcal{A} = \{A, C, G, T\}$ would result to 4^{25} nodes. In 2001, Pevzner introduced the application of DBGs in bioinformatics and adapted the original graph description by retaining only the k-mers observed in the input nucleotide sequences. This results to a number of nodes in the graph always lower than the sum of all input sequences' length. The first application in bioinformatics was for *de novo* assembly of reads (EULER [163], Velvet [221], ABySS [21], IDBA [160]). From 2010, DBGs have been used for variant detection as well (*e.g.* KisSNP [162], KisSplice [180], Cortex [93], DiscoSNP++ [198]).

More formally a DBG is a graph representing all the k-mers observed in the input sequences as nodes, and all possible $(k-1)$-overlaps between the k-mers as edges: an edge is drawn from a k-mer to another if the $(k-1)$-suffix of the k-mer equals the $(k-1)$-prefix of the other. DBGs can be compacted by first using a unique node to store a k-mer sequence and its reverse complement, and second by merging linear paths (path of nodes of in-degree and out-degree 1) [180]. Cortex, KisSplice, and DiscoSNP++ methods, allowing for reference-free variant calling, are based on the enumeration of bubbles in a compacted DBG (cDBG). Let a and b be sequences of more than k nucleotides and σ and σ' sequences not sharing any k-mer. Any pair of patterns $a\sigma b$ and $a\sigma' b$ represents a variation (σ versus σ'), and generates a bubble in the DBG, where a and b are switching nodes and σ and σ' are the two paths of the bubble.

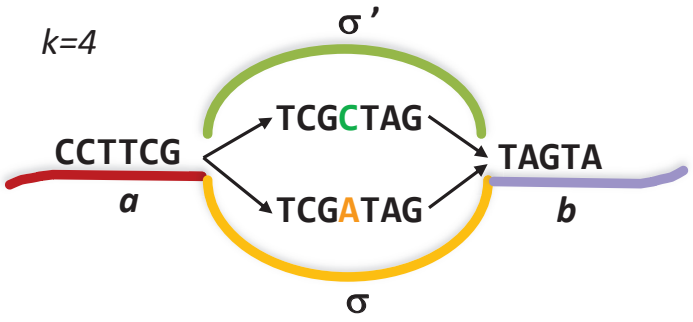


Figure 3: **Definition of a bubble.** A bubble in a cDBG is defined by a pair of patterns $a\sigma b$ and $a\sigma' b$, where a and b are sequences of more than k nucleotides and σ and σ' sequences not sharing any k-mer. In this example, $k = 4$ and σ and σ' paths represent an isolated SNP.

DiscoSNP++ specifically detects bubbles corresponding to isolated SNPs. Such bubbles can be recognised as their paths only differ by their central nucleotide – the SNP – and are of length exactly $2k - 1$ [198]. Cortex detects non-branching bubbles as well as path divergences which may correspond to complex bubbles [93]. KisSplice was designed to identify variations in transcriptomes. It enumerates bubbles and classifies them as SNPs, indels or alternative splicing events, depending on the path lengths of the bubble. Bubbles with more than 5 branches are discarded, as they often correspond to artifactual bubbles caused by repeated regions [180]. More generally highly branching bubbles are difficult to enumerate and each method introduced different solutions to deal with them, often resulting in a partial enumeration.

High dimension problem

In a GWAS, the number of covariates (the p genetic variants describing the genotype) is much higher – by several orders of magnitude – than the number of individuals (the n bacterial strains). This high dimension makes it impossible to use standard multivariate regression models, in particular because the normal equation (Eq. 3) cannot be solved.
The most common approach in association studies is to test independently each variant described in X, and thus compute p regressions. All previously cited bacterial GWAS used this approach, and this is also the strategy we implemented in the thesis. This strategy has indeed the advantage of being easily tractable and allows for a computation of a p-value for each variant using standard methods. However these p marginal regressions are often too simple models which do not account for the joint distribution of groups of variants [35, 120]. This approach is in particular very sensitive to the structure of the population, if it is not included in the model. However in practice the models adjust for the population structure, and such marginal models can be written, when considering a linear relationship between the phenotype and the j-th variable, as:

$$y_i = \beta_j X_{i,j} + W_i \alpha + \varepsilon_{i,j}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, p, \tag{4}$$

with $\varepsilon_{i,j} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, $\sigma^2 > 0$, the random error, $W \in \mathbb{R}^l$ a factor representing the population structure, and $\alpha \in \mathbb{R}^l$ the effect of this population structure on the phenotype. Testing all variants one by one also retains the so-called multiple testing problem: the probability to reject at least once the H_0 hypothesis by mistake, among the p tests, is not equal to the individual p-value of each test. Several methods were developed to compute an adjusted value from the p-values [78, 150]. Among them, the Bonferroni correction provides an adjusted family-wise error rate (FWER) [26] and the Benjamini-Hochberg

(BH) procedure allows to control the false discovery rate (FDR) [14]. These methods assume that the multiple tests are independent, which is not the case in practice because of the correlation between the tested genetic variants. To the best of my knowledge, there is currently no better alternative for positive dependency. In case of negative dependency, Benjamini-Yekutieli retains a control of the FDR while BH does not [15].
Multiple univariate test tends to produce small effect sizes because the phenotype variability is usually not explained by a single genetic variant. Moreover they can lead to more false positives because any variant correlated to a causal variant may show an association when tested alone [35].
A way to adapt joint regression to high dimension is to use a penalised regression [215]. Penalised regressions are models regulating the model coefficients and allowing to control a trade-off between the model bias (model adjustment) and variance (generalisation to other data). This trade-off is explicitly expressed in the model, which is composed of a loss function (model adjustment) and a penalty (regularisation of the model complexity), as shown in Eq. 5.

$$\min_f \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda\Omega(f), \tag{5}$$

where $L(y, f(x))$ is the loss function and $\lambda\Omega(f)$ the penalty. Ω measures the complexity of the model and λ is a tuning parameter for this regularisation. A low penalty can lead to a complex model which is too specific of the training data (overfitting), while a high penalty would lead to a simple model offering a higher generalisation to other data, but would lose in adjustment. A common way to regulate this trade-off is to penalise the $l1$-norm of the coefficients [191]. This penalty, called lasso, shrinks the coefficients and provide sparse models as it leads to estimators in which many coefficients are exactly 0. The ridge penalty, penalising the $l2$-norm of the coefficients [87], shrinks the coefficients and provide close coefficient estimates to variables which are close in Euclidean norm.
A ridge penalty applied to a linear model can we written as the following minimisation problem:

$$\min_\beta ||y - X\beta||^2 + \lambda||\beta||_2^2, \tag{6}$$

for which a closed-form solution exists:

$$\hat{\beta} = (X^\top X + \lambda I)^{-1} X^\top y. \tag{7}$$

Beyond these two standard penalties, the construction of dedicated penalties, adapted to particularities of the data, such as the correlation between variants (see next section), opens a large field of applications. However, such methods often require a prior knowledge on the data and the non-trivial tuning of the regulation hyperparameter. Moreover there is no general solution to quantify the significance of the association, as the null distribution of the estimated coefficients is unknown. Moreover as the penalty reduces the variance of the estimators by introducing a bias, the estimated coefficients are biased [77]. Inference with penalised methods is an active research field [50, 128].
Combined strategies have been proposed, such as a two-step procedure which first select a set of variants using a sparse regression such as the lasso, and then infer p-values on the subset. This is referred as post-selection inference: the data used for the inference is also used for selection, and the resulting statistics are over-optimistic if this issue is not properly taken into account [18, 25]. Methods based on multiple sample splitting allow for the calculation of statistical significance in a joint model, aggregated across multiple

splits [35, 144]. Mixed models have been described to offer a possibility of inference with joint analysis of variants, even though each variant is tested individually. In these models, the tested variant is included in the model as a fixed effect variable, while all variants are represented by random effects [218, 225]. These models are detailed in the section **Correlation between individuals**: they indeed also allow to adjust for population structure as they integrate all variants in the model.

Correlation between variants (linkage disequilibrium)

Linkage disequilibrium (LD) refers to the correlation between variants across genomes in a population. The squared Pearson coefficient of correlation is a common measure of the LD between two variants, which integrates the notion of disequilibrium D, defined as the deviation of the observed haplotype frequency from the frequency expected under equilibrium [184].
In human genomes, the LD is structured in blocks along the genome. These blocks are a consequence of the recombination between pairs of chromosomes during the meiosis. Thereby, the notion of LD is closely related to the proximity between variants.
Several penalties have been specifically proposed to take into account this LD in penalised regressions [54, 125]. The general idea of these methods is to assign close coefficients to close variants and/or to select groups of close correlated variants. It was also proposed to use hierarchical inference, which clusters close SNPs with a high LD measure, to assess the statistical significance of groups of close SNPs [35].
However in bacterial genomes not subject to systematic recombination, the correlation between variants is not structured in the same way. The correlation between variants observed in bacterial genomes is strongly related to the population structure: variants are transmitted together to individuals, and spread together in the population. As a consequence, the LD is not confined to close variants: distant variants in the genome can be very correlated [61]. Penalties assuming a proximity along the genome sequence become irrelevant. Penalties selecting correlated groups without any constraint on the position could be interesting in this context [227], but have not been addressed in this thesis. Because LD and population structure are strongly related, adjusting for population structure allows to take into account a part of the correlations between the variants [61].

Correlation between individuals (population structure)

From the human GWAS literature we know that taking into account a possible genetic structure within the population is essential to avoid confounding associations [55, 169, 182, 207, 226]. This is even more important for bacterial GWAS: because of their clonal reproduction, bacterial genomes are strongly correlated by clades, which increases the risk of identifying false associations [61, 63, 117]. Spurious associations happen when there is a population effect on the observed phenotype: some clades are enriched or depleted in resistant strains. For instance, if a mutation (A) causing resistance was acquired by the ancestor of a clade and transmitted to its descendants, the clade is enriched in resistant individuals. Any other mutation (B), not related to the resistance, which is acquired and transmitted within the descendants of this clade may be assigned a higher effect than other causal mutations (C) involving for instance fewer individuals (Figure 4).
Many methods have been developed to account for the population structure in human GWAS [55, 169, 182]. From 2006, two families of methods have been widely used: the first uses the principal components (PC) of the variant matrix, for variable adjustment, or as fixed-effect covariates in the model [168]. The second uses mixed models, in which the population structure is treated as a random effect: $u_i \sim \mathcal{N}(0, \sigma_a^2 K)$, where K is a

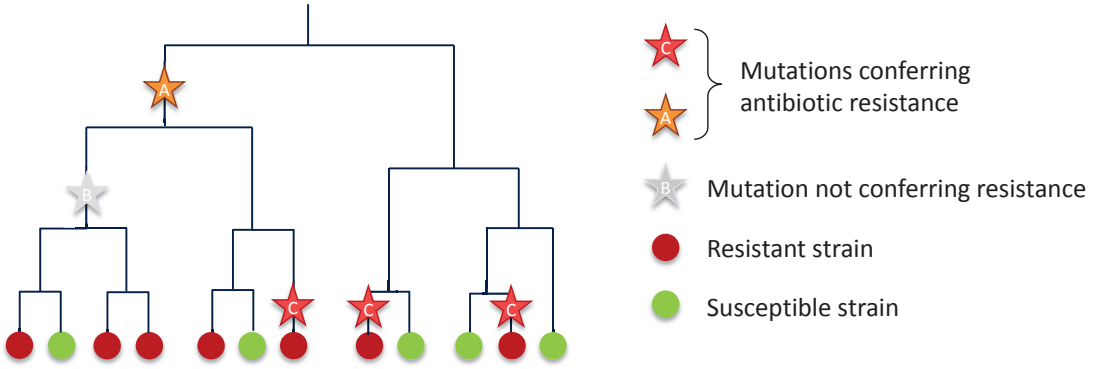


Figure 4: **Spurious association caused by population structure.** Mutations A (orange star) and C (red stars) are causal for the phenotype. Mutation B (grey star) is not causal but acquired in a clade enriched in resistant strains (dark red circles) and may be assign a high effect when ignoring the effect of all other mutations.

kinship matrix representing the relatedness between the individuals [61]. When assuming a linear relationship between the phenotype and the j-th variable for the i-th sample, the linear mixed model (LMM) is formulated as:

$$y_i = \beta_j X_{i,j} + u_i + \varepsilon_{i,j}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, p, \tag{8}$$

Consider $X = U\Lambda V^\top$, the singular value decomposition (SVD) of X, where U is the matrix of the n PCs, ordered by decreasing singular values, Λ is a diagonal matrix of the singular values and V stores the contribution of each variant to the PCs. Using the SVD, we can specify the marginal model presented in Eq. 4 for each of the two families of methods.

The first method considers the first PCs, which are the first q columns of U: in Eq. 4, $W = U_q$ and α represents the fixed effects of the PCs.

In the second method, the LMM can also be formulated using Eq. 4 when we consider the kinship matrix K as a simple function of the genetic variants: $K = XX^\top$. This matrix can thus be expressed relatively to the PCs, by considering the SVD of X [61, 88]. Indeed $K = XX^\top$ leads to $K = WW^\top$ with $W = U\Lambda$, and thus $u \sim \mathcal{N}(0, \sigma_a^2 K)$ is equivalent – by affine transformation of multivariate normal distribution – to $W\alpha$, with $\alpha \sim \mathcal{N}(0, \sigma_a^2)$. While only the first PCs are integrated in the fixed-effect model – providing a high-level estimation of the population structure, the mixed model allows for a finer resolution of the relatedness between individuals as all PCs are represented in the model [88]. However, as we assume a normal distribution centred around 0 with variance σ_a^2 for the random effects, their estimations are shrunk towards 0.

Buzdugan *et. al* classifies the mixed model as an hybrid between marginal and joint analysis: the variants are tested one by one, however in each test, all variants are represented in the model through the random effect [35].

Mixed models enable a computation of p-values for each variant as the null distribution of the estimated coefficients is known. From Eq. 4 and with $\beta = 0$ under H_0, the null model is expressed as:

$$y_i = W_i\alpha + \varepsilon_i, \quad i = 1, \ldots, n, \tag{9}$$

with $W = U\Lambda$, $\alpha \sim \mathcal{N}(0, \sigma_a^2)$ and $\varepsilon \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$. This random model can be expressed as a ridge regression, when considering $\lambda = \frac{\sigma^2}{\sigma_a^2}$ [23, 61] :

$$\min_{\alpha} ||y - W\alpha||^2 + \lambda||\alpha||_2^2, \quad \lambda = \frac{\sigma^2}{\sigma_a^2} \tag{10}$$

This formulation describes explicitly the shrinkage of the estimates, and the variance in the population structure defines the strength of the shrinkage.
Earle *et. al* pointed out that correcting for the population structure may mask causal population-stratified variants. To address this issue, their method, bugwas, in addition to testing regular variant effects with an LMM, also tests the lineage effects, where the lineages are approximated by the PCs. To do so, they use the random model defined in Eqs.9 and 10 and use the ridge closed-form solution provided in Eq. 7 to compute the mean and variance of α, which represent the effects of the PCs on the phenotype [61].
The variant matrix which is used to represent the population structure is not always the full genotype matrix X. In their k-mer-based approach, Earle *et. al* do not use the matrix of presence/absence of the k-mers which are tested for association with the phenotype, but instead use a restricted matrix of biallelic SNPs, while Lees *et. al* use a random subsample of their k-mer matrix in the SEER approach [117]. We did not investigate the impact of estimating the population structure using the accessory genome information in addition to core variations, however a previous analysis showed encouraging results for the use of k-mers in population genomics [188].
In Chapter 1 of this thesis, we present a study adding the PCs of a matrix of core SNPs as covariates with a fixed effect. In Chapter 2, we evaluate fixed and mixed effect models with simulated and real data. In this case we used $X = U\Lambda V^\top$, where X is the full genotype matrix.

Outline of the Chapters

Chapter 1 presents a study of *P. aeruginosa* resistance published in the International journal of antimicrobial agents, which focuses on the determinants described in antibiotic resistance-dedicated websites. It includes a descriptive analysis of the repartition of these determinants in a panel of 672 strains, and a genotype to phenotype association study, identifying the SNP and resistance gene presence/absence associated to the MIC obtained for 9 antibiotics.
Chapter 2 introduces a representation of the genetic variants based on a single DBG built from all genome sequences of the considered strain panel. It discusses how to make this representation amenable for a GWAS and evaluates several possibilities of phenotype modelling and population structure adjustment. It introduces a post-analysis based on DBG subgraphs which are enriched with metadata and allow an interpretation of the results in terms of SNPs or MGE insertion.The produced method is described in a pre-print integrated to this Chapter.
Chapter 3 integrates a manuscript accepted for publication in PLOS Genetics, which presents the software we built from the method described in Chapter 2: DBGWAS (for DBG and GWAS). It includes a detailed presentation of the results obtained by the software on three species: *P. aeruginosa*, *S. aureus* and *M. tuberculosis* for which tens of antibiotic resistance phenotypes were available. It compares DBGWAS efficiency to other GWAS methods.
Chapter 4 reports an unpublished work aiming at automatically predicting the type of the variants (SNP or MGE) reported by DBGWAS. It presents the construction of training and test datasets from real and simulated data, and the machine learning strategy used to select the model providing the best prediction performances.

Chapter 1

The resistome of *Pseudomonas aeruginosa*

1.1 Preamble

The work presented in this Chapter was started by other colleagues before my PhD. Jean-Baptiste Veyrieras designed the study, and in particular the association testing strategy presented in the following manuscript, using an ordinal regression to model the phenotype measured by the antibiotic minimum inhibitory concentration (MIC), and integrating the population structure and the effect of the genuine determinants.
The variants tested in this study were antibiotic resistance (AR) gene presence and polymorphism. These AR genes were gathered from the *P. aeruginosa* literature and from dedicated databases [97, 141]. This collection of *P. aeruginosa* AR genes represents the species' *resistome*, and defines the scope of the study. The resistome-wide association study (RWAS) described below allows to finely estimate the effect of the variations in the resistome for several antibiotics.
We extended the study with a descriptive analysis of the resistome, highlighting the important part of the accessory genome in the antibiotic resitance in *P. aeruginosa* species, and with an analysis of the diversity of the integrons found in the panel. Indeed, integrons are key genetic elements regarding the adaptability of bacteria to their environment. They allow the capture and dissemination of new genes, thanks to a specific recombinase gene, the integrase, found at the extremity of the integron cassette structure [140] and play an important role in the acquisition of accessory resistance genes.

The manuscript below was published in 2017 under the following reference:
Jaillard M, van Belkum A, Cady KC, Creely D, Shortridge D, Blanc B, Barbu EM, Dunne WM, Zambardi G, Enright M, Mugnier N, Le Priol C, Schicklin S, Guigon G, and Veyrieras J-B (2017). Correlation between phenotypic antibiotic susceptibility and the resistome in *Pseudomonas aeruginosa*. *International journal of antimicrobial agents*, **50**(2), 210–218.

1.2 Manuscript published in IJAA (2017)

"Correlation between phenotypic antibiotic susceptibility and the resistome in Pseudomonas aeruginosa"

Abstract

Genetic determinants of antibiotic resistance (AR) have been extensively investigated. High-throughput sequencing allows for the assessment of the relationship between genotype and phenotype. A panel of 672 *Pseudomonas aeruginosa* strains was analysed, including representatives of globally disseminated multidrug-resistant and extensively drug-resistant clones; genomes and multiple antibiograms were available. This panel was annotated for AR gene presence and polymorphism, defining a resistome in which integrons were included. Integrons were present in > 70 distinct cassettes, with In5 being the most prevalent. Some cassettes closely associated with clonal complexes, whereas others spread across the phylogenetic diversity, highlighting the importance of horizontal transfer. A resistome-wide association study (RWAS) was performed for clinically relevant antibiotics by correlating the variability in minimum inhibitory concentration (MIC) values with resistome data. Resistome annotation identified 147 loci associated with AR. These loci consisted mainly of acquired genomic elements and intrinsic genes. The RWAS allowed for correct identification of resistance mechanisms for meropenem, amikacin, levofloxacin and cefepime, and added 46 novel mutations. Among these, 29 were variants of the *oprD* gene associated with variation in meropenem MIC. Using genomic and MIC data, phenotypic AR was successfully correlated with molecular determinants at the whole-genome sequence level.

Introduction

Pseudomonas aeruginosa easily integrates exogenous DNA and, under selective antibiotic pressure, efficiently manifests resistance traits [124]. The emergence of resistance most likely occurs in isolates with enhanced virulence, during infection and treatment, allowing *P. aeruginosa* to evolve resistance to antibiotic-mediated killing [45]. Despite significant genetic variability within the species, antibiotic-resistant clones that spread globally have been identified [208]. Among them, multidrug-resistant (MDR) and extensively drug-resistant (XDR) clones are common, a phenotype that often relates to integron expansion. These pandemic clones include the highly prevalent sequence types ST235, ST111, ST348 and ST175 [36].

Progress is needed in the field of antibiotic resistance (AR) and antibiotic susceptibility testing [199]. Genome sequencing of *P. aeruginosa* has identified molecular markers for resistance to amikacin, meropenem and levofloxacin[105]. In addition, the International *Pseudomonas aeruginosa* Consortium has published a resistome recognising 73 AR genes in 389 isolates and highlighting the importance of the accessory genome [68]. CRISP-Cas-mediated immunity does not appear to be directly blocking acquisition of resistance elements [200]. Hence, additional phenotype–genotype association studies for *P. aeruginosa* are needed.

In other pathogens, studies have suggested that genomic antibiograms can be as good as phenotypic ones. The first study involving mixed bacterial species documented 99.7% concordance between genotypes and phenotypes [219]. Work focusing on *Staphylococcus aureus*, *Escherichia coli*, *Klebsiella pneumoniae* and *Campylobacter spp.* has expanded these

findings [79, 187, 224]. Optimum concordance was observed for clonal bacterial species such as *Mycobacterium tuberculosis* [202]. A recent resistome analysis of *P. aeruginosa* correlated meropenem resistance with outer membrane protein OprD polymorphism [164]. Genomic approaches hold promise for the development of future antibiotic susceptibility testing systems for routine use in clinical microbiology laboratories, although major knowledge gaps still need to be filled.
Here we present a resistome characterisation, integron dynamics and resistome-wide association study (RWAS) for 672 *P. aeruginosa* strains with complete genomes and minimum inhibitory concentration (MIC) values for various anti-*P. aeruginosa* antibiotics.

Materials and methods

Description of the strain panel

Genomic sequences of 672 *P. aeruginosa* strains were analysed. Strains were obtained from three collections: the bioMérieux collection ($n = 219$) [200]; the Kos collection ($n = 390$) [105]; and the Pirnay collection ($n = 63$) [164]. Genomic multilocus sequence typing (gMLST) and phylogenetic comparisons were performed. Reference AR data based upon broth dilution assays and VITEK®2 tests were available for several antibiotics (Supplementary Figure S1.5). More detailed information on all strains can be obtained from previously published work [200]. The genome sequences and antibiograms of 390 publicly available *P. aeruginosa* isolates (Kos collection) were added [105]. In this panel, resistance data covered meropenem, amikacin, levofloxacin and colistin.

Resistome

The resistome of the 672 *P. aeruginosa* isolates was obtained by annotating each genome assembly using an in-house, pan-bacterial AR sequence database. This database contains 2545 relevant, nonredundant reference sequences categorised into 569 AR genes. The database contained at least one sequence for each previously reported AR gene, including genes involved in susceptibility and/or resistance in *P. aeruginosa* [124]. Each genome was thus run through the database via BLASTn v.2.2.28+ for AR genes with at least 80% identity and 80% coverage. Overlapping alignments belonging to the same AR gene were clustered into a single hit if the overlap covered $> 10\%$ of the aligned reference sequences. Only the best hit, defined as the alignment with the highest percentage identity times the reference sequence coverage, was kept to infer the haplotype state of the gene. Note that for a given isolate and for a given AR gene, several distinct annotations can be reported when hits appeared at distinct genomic locations (e.g. duplicated genes). However, precise plasmid copy number definition was impossible using this approach.

Resistome genotyping

The 672 isolates were genotyped for allele counts both at the 'locus' and 'variant' levels. By allele counts at the locus level, the number of copies of a given AR gene (e.g. bla_{TEM}, *mexX*, etc.) was measured irrespective of the number of alleles of the gene. Regarding the allele counts at the variant level, all of the detected sequences of a given AR gene were aligned using MAFFT v.6.861 with high-precision mode [101] in order to identify both single nucleotide polymorphisms (SNPs) including tri- and quadri-allelic sites and indels. Finally, for a given isolate, the number of occurrences of each SNP and indel alleles was counted and this was used as the corresponding allele count at the variant level (see Supplementary Table S1.1).

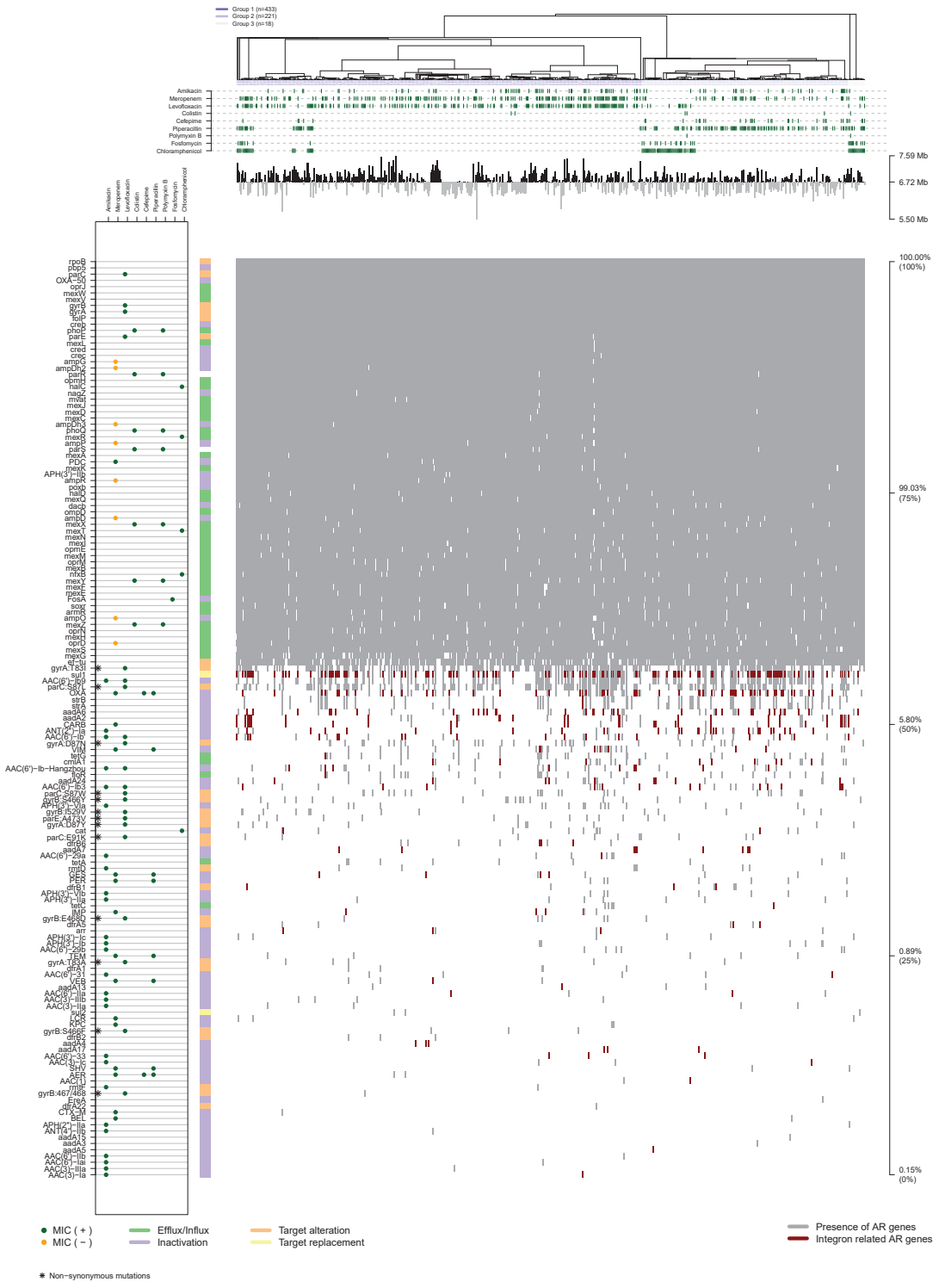


Figure 1.1: **Overview of the phenotypic, genomic and resistome data of 672 *Pseudomonas aeruginosa* isolates included in the present analysis.** The phylogenetic tree is inferred from core gene content and depicts the three major groups of *P. aeruginosa* in shades of purple. Depicted below the tree in dark green bars are the isolates that were found to be resistant to one or more of the nine antibiotics analysed in this study. Below, strains with draft genomes larger or smaller than the median are depicted as black or grey bars, respectively. The left panel provides the names of genes and non-synonymous mutations constituting the resistome for the nine antibiotics. Different antibiotic resistance mechanisms are colour coded (bar at the right of the determinant list). The right panel reviews resistance gene content on a per isolate basis, with grey shading denoting the presence of a given resistance gene or allele. The percentage of strains harbouring given resistance genes or alleles is shown on the far right, allowing for easy discrimination of core and accessory elements. The resistome structure illustrates that antibiotic inactivation genes are more likely to belong to the accessory resistome than efflux genes. Note that resistance genes embedded in integrons are colour coded as dark red bars. Additional columns with global information (e.g. plasmid content, percentage GC, etc.) can be added if needed.

Integron analyses

Integrase genes were detected by aligning (tBLASTn) the assembled genome sequences with the protein sequences of three integrases (IntI1, IntI2 and IntI3) as derived from a targeted UniProt search (sequences are provided in Supplementary File S1). Contrary to Intl1, for which we were able to retrieve 36 sequences, only 3 sequences were found for IntI3 and 1 sequence for IntI2. A 70% identity and 70% protein coverage cut-off was applied for the alignments. Then, recognisable and annotated AR genes upstream of the integrase start codon were searched for and the physical distance between the start codon and the $5'$ end of the resistance element was reported. All AR elements within a 10 kb window were included.

Resistome-wide association study (RWAS)

First, the strength of association between established genotypes linked to increases or decreases in the MIC (generally beyond the resistance breakpoint) was assessed. Second, novel candidate genotypes associated with additional variability in MIC values were looked for, taking the effect of known genotypes into account. Fig. 1.1 provides a literature-supported overview of the causal genes or variants thereof known to increase/decrease the MIC for each antibiotic in the association study.

Population structure

Results may be inflated by the presence of cryptic correlations between population structure and the MIC status of the strains. This could be due to population-wide linkage disequilibrium (LD) between causal mutations and genetic structure, or to a sampling bias leading to over-representation of related strains. Confounding effects related to population structure were restrained by computing the principal components (PCs) from the core–SNP genotype matrix and using them as covariates in the statistical association framework. To avoid extensive correlations that could mask the effect of a core gene on resistance, genotypes associated with polymorphisms in, for example, *gyrA*, *gyrB*, *parC*, *parE*, *folP* etc., were removed [166].

Associative modelling

Ordinal regression was used to develop the core statistical RWAS framework [142]. This provides the benefit of adequately accounting for the ordinal nature of the MIC values and facilitating inclusion of covariates. For each antibiotic, the PC was defined to include as covariates for population structure control, using a forward selection procedure [86]. This allows to build Z, the $n \times k$ matrix of PCs, where n is the number of isolates and k is the number of retained PCs. For each antibiotic, an optimal set of unequivocal genotypes was selected using a backward elimination procedure based on the Akaike information criterion (AIC) [2, 86]. An optimal although conservative subset of genuine genotypes associated with MIC variability was selected, while preserving statistical power for detecting new associations. This step led to a matrix U, of size $n \times p$, of established genotype counts where p is the number of retained genotypes. Novel candidate genotypes were evaluated for locus and variant allelic counts independently for each candidate genotype x. If we denote Y_i as the MIC values for isolate $i = 1, \ldots, n$ assuming that Y_i can fall into $j = 1, \ldots, J$ ordered categories (from the lowest tested antibiotic concentration to the highest), we then considered the following cumulative link model for each candidate $P(Y_i \leqslant j) =$

$\pi_{i1} + \ldots + \pi_{ij}$, where π_{ij} denotes the probability that the i^{th} observation falls in the j^{th} MIC category. The cumulative logit regression model is then:

$$\begin{aligned} logit[P(Y_i \leqslant j)] &= log[\frac{P(Y_i \leqslant j)}{1 - P(Y_i \leqslant j)}] \\ &= \theta_j - (Z\alpha + U\beta + x\gamma) \end{aligned} \tag{1.1}$$

where θ_j stands for the MIC category intercepts, α is the vector of k regression parameters for the PC scores capturing the population structure, β is the vector of p regression parameters for the known causal genotypes, and γ is the regression parameter for the candidate genotype x.
To compute the P-value of the association between candidate genotypes and the MIC variability, a standard likelihood-ratio test was used [209]. This test compares the log-likelihood of the model under the null hypothesis (*H0*: the genotype has no effect on MIC variability; $\gamma = 0$) versus the alternative hypothesis (*H1*: the genotype has an effect on MIC variability; $\gamma \neq 0$). If we denote $L0(\theta, \alpha, \beta, Y)$ and $L1(\theta, \alpha, \beta, \gamma, Y)$ the log-likelihoods under each hypothesis, we can derive the likelihood-ratio test formula: $-2 \times [L1 - L0] \sim \chi^2$ and compute the association P-value for each individual genotype. Significant associations were reported by controlling the false discovery rate for each antibiotic at 5% using the Benjamini–Hochberg procedure [14]. The R package ordinal (R Development Core Team) was used to implement the cumulative logit regression framework.

Results

Resistome analysis

Nine antibiotics belonging to six drug families were studied: aminoglycosides (amikacin); β-lactams (meropenem, cefepime and piperacillin); quinolones (levofloxacin); polypeptides (colistin and polymyxin B); fosfomycins (fosfomycin); and phenicols (chloramphenicol). Resistome analysis identified 147 loci previously believed or shown to cause resistance (see Fig. 1.1 for a complete data review). Whilst most of these were associated with acquired or accessory genomic elements, 45% are intrinsic and reside within the conserved core *P. aeruginosa* genome (Fig. 1.1, right panel). The 147 AR elements cover five mechanisms that neutralise antibiotic action, involving both intrinsic and acquired resistance traits. Resistance related to drug efflux and porins was confirmed to be mainly intrinsic, whereas resistance related to antibiotic inactivation was acquired (Fig. 1.1). Note that significant numbers of major pandemic clones ST235 ($n = 67$), ST111 ($n = 58$) and ST175 ($n = 11$) were included [200].

Integron analyses

Integrase sequence hits were found in one-third of all isolates ($n = 229$) and an AR gene was identified upstream of two-thirds of these ($n = 163$). Only class 1 integrons were detected, which carry a variety of resistance genes, particularly targeting phenicol, aminoglycoside or β-lactam antibiotics [124]. Thirty-seven different resistance genes were found co-localised with the integrase. bla_{OXA} and *sul1* were found to be the most frequent AR genes. Of the 163 integrons carrying AR genes, only 17 did not carry *sul1* genes. They encode aminoglycoside-modifying enzymes including adenylyl transferases (ANT) and acetyl transferases (AAC) [76, 124]. These gene families were enriched in *P. aeruginosa* integrons, as shown by the Poisson distribution tests (Fig. 1.2A). Enrichment was defined as a higher proportion of a gene found in an integron compared with other genome locations.

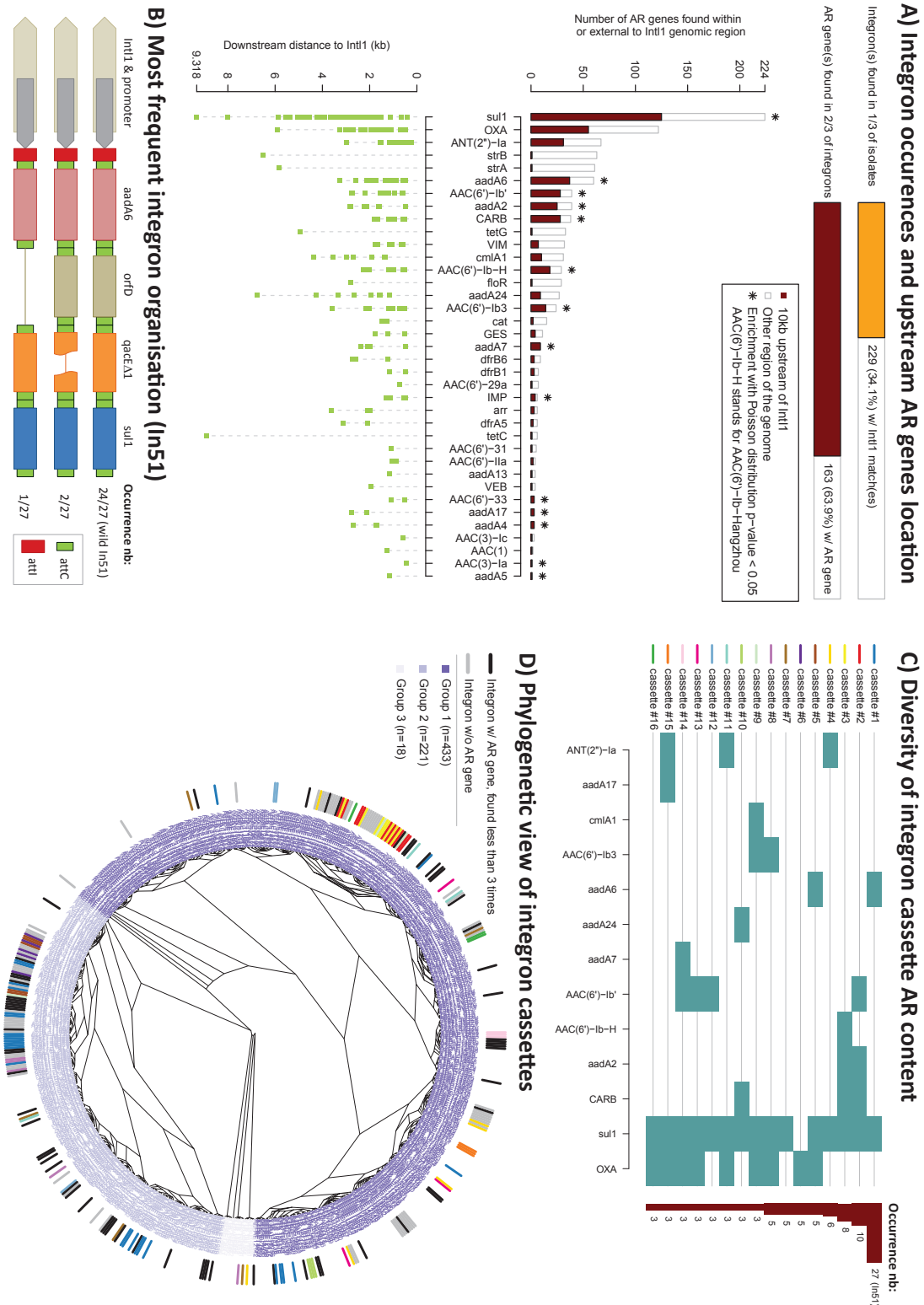


Figure 1.2: **Class 1 integron composition analysis for *Pseudomonas aeruginosa*.** Since only Intl1 sequences were detected in our panel, only class 1 integrons are reported. (A) Depicts the integron frequency (34.1%) in the isolates and assesses the antibiotic resistance (AR) content of putative integrons by searching for resistance genes up to 10 kb upstream of the Intl1 motif. The resistance genes were tested for enrichment in integrons compared with alternative genome locations. (B) Further identifies the order of the genes and their individual frequency of occurrence for the main integron cassette, In51. (C) Depicts the different configurations (in terms of AR gene composition) of the most frequent Intl1 integrons, with the most frequent one ($n = 27$) containing *sul1* and *aadA6* AR genes only. (D) Shows their presence in the different clonal *P. aeruginosa* clusters as present in the phylogenetic tree. Note that some integron types cluster according to bacterial sequence type (ST), whereas others are more widely scattered throughout the phylogenetic tree.

Seventy-two different AR gene combinations were identified within the integrons. The most prevalent combination occurred 27 times and was identified as the In51 integron [76], containing, ordered upstream of the integrase, *aadA6*, *orfD*, *qacE*Δ1 and *sul1* genes. Three isolates carried a truncated form of In51 with either a deletion within the antiseptic resistance gene *qacE*Δ1 or complete deletion of the *orfD* gene (Fig. 1.2B). The fact that certain AR gene combinations were found more than three times (Fig. 1.2C) and were variably distributed along the phylogenetic tree highlighted the great diversity of integron cassettes (Fig. 1.2C,D). Integrons specific to small clades could be identified, such as those carrying AR cassettes #2, #3, #14 or #15, as well as integrons that spread along all phylogroups [cassettes #1 (In51), #4 or #7] (see Fig. 1.2C). Obviously, such analyses identify physically-associated resistance traits and generate evidence of horizontal transfer.

Correlating genotypes and phenotypes through RWAS

Association between resistance gene presence or the existence of polymorphisms in resistance genes does not always directly correlate with phenotypic resistance data. Here were describe an approach that helps find such correlates.

General considerations

A detailed study was carried out based on ordinal regression, associating each of the nine antibiotic phenotypes (MIC values) with the strain genotypes [allele counts at locus (Fig. 1.3A) and variant (Fig. 1.2B) levels]. The data revealed both known and new associations (Fig. 1.3). Approximately 28% of the known loci were not found to be associated because their minor allele occurrence was one or two, which is clearly too low to perform any useful statistics. This involves 21 genes and some mutations in *gyrA* and *gyrB* genes. The test did not have sufficient statistical power to detect variants with such extreme allele frequencies. A very low prevalence of a particular phenotypic resistance status also prevents identification of significant associations. This prevalence was below 3% for polymyxin B, colistin and chloramphenicol (Supplementary Table S1.2), which may explain the absence of significant associations between known genotypes and MIC values for these three antibiotics.

Colistin and fosfomycin resistance markers

Sufficient samples were available for colistin ($n = 524$) to identify new associations between genotypes and MIC variations within the sensitive strains. These variations mostly involve low MIC values, below the resistance breakpoint value of 8 μg/mL. First, the absence of the *mexS* gene, which is located upstream the *mexEF–oprN* efflux genes and affects their expression, in 24 isolates yields an increased susceptibility: 50% of these isolates exhibit an MIC < 0.5 μg/mL [152]. Second, a deletion of the 5'end of the *ampD* gene observed in 12 non-clonal isolates appears to be related to an increased MIC value: 60% of these isolates have an MIC $\geqslant 2$ μg/mL in comparison with 3% for the 512 isolates with the non-deleted version of *ampD* (Supplementary Fig. S1.1). This is likely to be an indirect association.

No association was found between *fosA* (present in 108 among 113 strains) and fosfomycin MIC. The limited sample size ($n = 113$) together with the experimental variability of fosfomycin testing may explain this lack of correlation [71]. Recent data showed that variation in the glycerol-3-phosphate permease gene *glpT* is correlated with changes in colistin susceptibility [37]. At the time of analysis this marker was missing from the database.

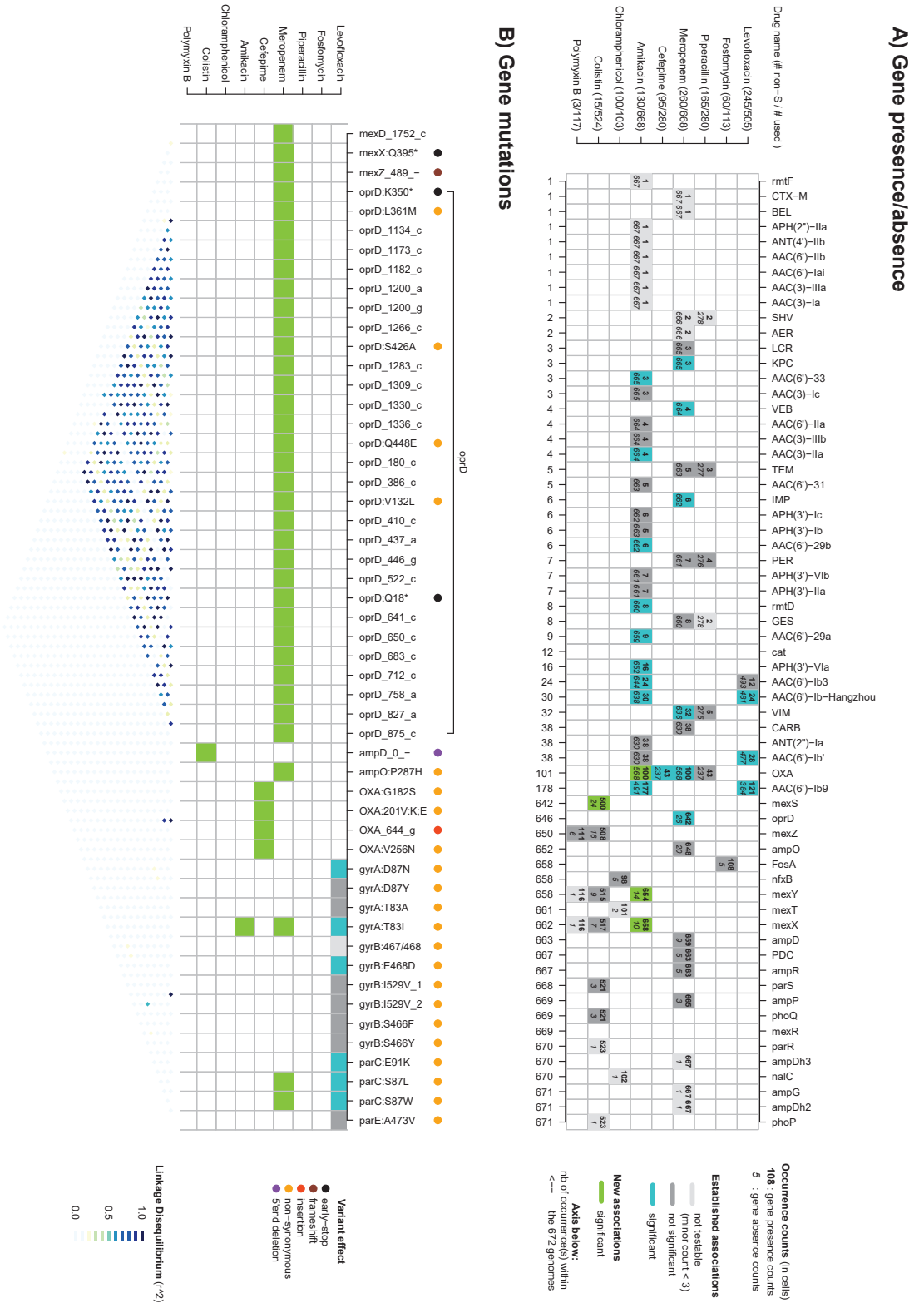


Figure 1.3: **Synoptic representation of the resistome-wide association study (RWAS) results.** Only genes and mutations either known to be associated or subsequently found to be associated with resistance against the nine tested antibiotics are displayed. (A) Provides the results for the presence versus absence of genes; and (B) focuses on specific mutations within genes. Genes and mutations are reported in columns and antibiotics in rows. Note that rows were ordered by decreasing phenotype prevalence, and in (A) columns were ordered by increasing allele frequency. Green cells indicate that a significant new association has been detected. Blue cells represent a known determinant significantly associated with minimum inhibitory concentration (MIC) variability. Conversely, grey cells indicate known determinants not significantly associated with MIC variability. Cells with extreme allele frequencies are shown in light grey. In (A), presence and absence gene counts in each drug sub-panel are noted in the cells (presence counts are in bold and above, absence counts in italic and below). Finally, the nature of the mutation is colour coded on the top of (B). The bottom panel represents the extent of linkage disequilibrium between mutations.

Piperacillin and cefepime resistance markers

No significant association was observed for piperacillin for any of the β-lactamase genes expected to impact resistance. However, we would have expected a high prevalence of these β-lactamases as there are 165 strains non-susceptible to piperacillin, and yet only two to five occurrences were found for bla_{SHV}, bla_{TEM}, bla_{PER}, bla_{GES} and bla_{VIM}. The bla_{OXA} gene was the most prevalent established gene (43 occurrences found among the 280 isolates), but an association was not confirmed. As β-lactamases are often carried by plasmids, one hypothesis is that some plasmids were poorly sequenced and were thus missed [40]. Susceptibility to cefepime was associated with the presence of bla_{OXA}. Associations with bla_{OXA} mutations were found, which suggests that specific alleles of bla_{OXA} correlate with higher MIC values, in particular mutations related to haplotypes bla_{OXA-31} and $bla_{OXA-224}$ (Supplementary Fig. S1.2) [9].

Levofloxacin resistance markers

Most of the known resistance genotypes for levofloxacin were found to be significantly associated with the MIC values, except the presence of AAC(6′)-Ib3, and some mutations in *gyrA* and *gyrB* with low allele frequencies. The two common non-synonymous mutations *gyrA*:T83I (38% of the isolates) and *parC*:S87L (26% of the isolates) show the highest association scores with P-values of 1.29×10^{-61} and 3.52×10^{-40}, respectively. No additional association was detected.

Amikacin resistance markers

Regarding amikacin, 9 of the 26 established genes (mostly the AAC family) were found to be statistically significantly associated with elevated MICs. Among the 17 established genes for which no association was found, 7 showed extremely low allele frequencies. Regarding associations, isolates lacking the efflux pump genes *mexX* (10 isolates) and *mexY* (14 isolates) tend to exhibit lower MIC values: in both cases ca. 50% of the *mexX*−*mexY*-deleted isolates have an MIC $\leqslant 0.5$ μg/mL (Supplementary Fig. S1.3). Since the expressions of *mexXY*−*oprM* (aminoglycoside efflux pump) and *oprD* are inversely correlated, one would expect a higher OprD expression in such strain whence the MIC would increase. Surprisingly, the *gyrA*:T83I genotype is strongly associated with amikacin MIC values, suggesting either a joint selective pressure for resistance to levofloxacin and amikacin or a sampling bias in our panel where closely related MDR isolates could be over-represented. The association with the presence of the bla_{OXA} gene is also likely an artefact since bla_{OXA} genes are a proxy of resistance cassettes linked to class 1 integrons.

Meropenem resistance markers

Of the 23 known genotypes for meropenem resistance, 6 showed a statistically significant association (bla_{VIM}, bla_{VEB}, bla_{OXA}, bla_{KPC} and bla_{IMP} β-lactamases and *oprD* gene presence or variations). Again,seven established genes were not found to be associated owing to extreme allele frequencies. The absence of the porin gene *oprD* (ca. 4% of the tested isolates) was confirmed as significantly associated with increase in MIC values: 75% of the isolates lacking *oprD* showed an MIC $\geqslant 8$ μg/mL whilst only 25% of isolates with *oprD* reached such a value (Supplementary Fig. S1.4). Regarding new associations for meropenem, *gyrA* and *parC* non-synonymous mutations related to levofloxacin resistance were identified. Assumptions similar to those for amikacin resistance associated with *gyrA* mutation may apply here. Interestingly, several mutations within the porin gene

oprD showed a significant association with meropenem MIC variability. Most of these mutations are in strong linkage-disequilibrium (LD) (Fig. 1.3B). When looking at LD measures, we found that both mutations leading to early stop codons (*oprD*:Q18* and *oprD*:K350*) were linked together and in low LD with other oprD mutations, and both led to an increase in the MIC values. Conversely, two linked synonymous mutations (*oprD*_180_*c* and *oprD*_758_*a*), also in low LD with other *oprD* mutations, appear to be associated with a decrease in MIC value. The fact that these mutations do not result in changes to the protein sequence may suggest that they are in LD with a non-identified causal mutation elsewhere. In examining efflux pump genes associated with meropenem resistance, a frameshift mutation was found within *mexZ* (*mexZ*_489_− in Fig. 1.3) carried by 12 isolates, among which 10 had a MIC > 8 μg/mL. Such a change in the reading frame is expected to alter the protein function; however, this particular frameshift mutation correlates with an increase in MIC value [139]. In the same way, but to a lesser extent, the early stop mutation found in *mexX* (*mexX*:Q395*) observed in eight isolates correlated with a slight increase in the MIC value. Conversely, the synonymous *mexD* mutation (*mexD*_1752_*c*) observed in six isolates was found to be associated with a decrease in MIC value. All these results in efflux pump genes must be treated with caution since the mutation frequencies are relatively low. Finally, a non-synonymous mutation was identified in *ampO* (*ampO*:P287H in Fig. 1.3), yielding a decrease in MIC values for the six isolates carrying this genotype. All association results are listed in Supplementary Tables S1.3 and S1.4.

Discussion

Genome sequences help detect and define existing and new resistance traits (e.g. [43]). Large databases that contain more or less complete inventories of genetic factors known to be involved in AR resistance have been developed. The format of such databases may differ and some of them are accompanied by specific software packages that facilitate searches for resistance genes in genome sequences or metagenomic data sets (e.g. [29, 141, 179, 205, 220]). These tools are new and need validation for the different human-specific bacterial pathogens. Here we focused on *P. aeruginosa* and its MDR and XDR clones as a model bacterial species and defined the genomic polymorphisms associated with antibiotic (multi-)resistance. In particular, the *oprD* gene and polymorphisms therein were confirmed as important meropenem AR determinants. Globally, the analyses confirmed most expected correlations if genotypes occur in sufficient frequency among phenotypically defined groups of isolates in the strain panel. The new statistical framework allowed us to explore novel candidate associations, such as the two mutations in *oprD* leading to an early stop codon, which clearly associated with an increase in meropenem MIC values.

Intrinsic resistance is a frequent phenotype for *P. aeruginosa*, which is dependent on a significant number of variables including colony morphology and metabolism [149, 205]. It correlates with multiple genomic markers, which begs for additional insight in these markers. Major markers are the efflux pumps and AmpC β-lactamases. Differences in the expression of these genes are very relevant to changes in resistance levels and are often due to changes in regulatory genes or regions. Here we have been focusing primarily on intragenic markers and we realise that in future versions of the database covering the resistance genes and their mutations, we need to include significantly more markers including, for instance, the *glpT* gene [37]. Recently, the first genome sequences of strains belonging to the *P. aeruginosa* MDR clone ST111 were reported [210]. The authors suggested that mapping of the polymorphism they detected in a variety of well-known

AR genes could be used for optimisation of antibiotic choice. Similar data for ST235 became available and genomic islands 1 and 2 were shown to be important in the dynamics of resistance development. Even more recently, a study of ST175 was published [37]. Class 1 integrons were identified as important evolutionary denominators facilitating resistance flexibility [46, 214]. A study described the successful detection of presumptive ceftazidime resistance markers in 88% of all *P. aeruginosa* genomes studied [106]. Here we introduced a method that allows monitoring of resistance acquisition but can also genomically categorise strains based on MIC levels, even below clinically relevant resistance. In the context of elevated LD and its drive towards 'statistical fog', we were still able to exploit the sequence information and to identify mutations that correlate well with the MIC distribution. This strategy permits prediction of the emergence and evolution of newresistance haplotypes on the basis of genome sequences and solid phenotypic AR data. It may also help understand the mechanisms of resistance development, including genetic control and regulatory feedback loops.

The RWAS approach allowed for the identification of new resistance factors, such as the *mexXY* efflux pump for amikacin, and RWAS highlights the importance of the allelic form of accessory genes: in meropenem resistance, not only *oprD* presence was expectedly correlated with MIC variation, but also polymorphism in the *oprD* gene was associated with this resistance. Likewise, bla_{OXA} allelic forms were also found to be important in cefepime resistance.

This study also showed some limitations in terms of the panel of isolates included for multidrug testing. Indeed, when working with nine drugs using a single panel, it is difficult to guarantee a high strain number for all phenotypes involved and lack of extreme allele frequencies of genotypes that prevent recognition of causal associations (Fig. 1.3). Also, there is a continuous need for database updating.

We demonstrate the usefulness of the bioinformatics pipeline developed for cataloguing full resistance gene content, the characterisation of integron composition and the value of RWAS for the detection of (new) resistance markers. Additional analyses using phenotypically well-defined and genetically diverse as well as ST-identical strains are needed to further validate the findings presented here. Subsequently, we envision transitioning towards pan genome-wide association studies (GWAS) without the strict need for a reference genome, which would otherwise be quite restrictive for highly plastic species such as *P. aeruginosa*. Our methods may further detail genotype–phenotype associations but validation is required before sequence-based resistance prediction can be used routinely.

1.3 Concluding remarks

This study was essential to acquire a good knowledge of the resistance mechanisms and their related genetic determinants in *P. aeruginosa*. In particular, it highlighted the rich and plastic structure of the *P. aeruginosa* genome and the importance of its accessory genome (at least 2 Mbp as the genome lengths are distributed between 5.8 and 7.6 Mbp within the species). The resistance was shown to be highly carried by determinants in the accessory genome. The resistome description confirmed that the genes involved in efflux pumps and porins are naturally present within the species, while genes allowing inactivation of the antibiotics are mainly acquired. A significant part of the accessory genes was found related to integrons. A descriptive analysis of these integrons highlighted the great diversity of these elements, illustrating here again the species genome plasticity. As discussed in the article, the panel used in the analysis showed some limitations in its power to detect some genuine variants. The inclusion of strains in a panel dedicated to such association study is however not trivial: strains should be representative of all groups in the species, and have balanced phenotypes, ideally for several antibiotics. Moreover, the quality of both the phenotype and genotype measures are essential. In an attempt to overcome some limits of the panel used in the study, we performed additional high-level descriptive analyses of the assembly quality, as it it the raw material for the genotype computation. As described in the Methods section of the article, the 'full' panel was composed of three collections: Kos, Pirnay and bioMérieux. From these supplementary analyses, we noticed that the assemblies from the Kos collection missed a significant part of core genes (Supplementary Figure S1.6A). We attributed this to the highest number of non-canonical bases in these assemblies (Supplementary Figure S1.6B). Hence, we decided to focus our developments only on a reduced panel composed of the Pirnay and bioMérieux collections.

This study, at the resistome scale, provided new candidates only to four phenotypes among the nine, and – by design – only among the genes belonging to *P. aeruginosa* resistome. Note that in the case of the levofloxacin however, it is likely that no new marker was found because the genuine markers are very deterministic of the phenotype. This however motivated us to work at a larger scale, and to take into account the particularly plastic nature of *P. aeruginosa* genomes, by describing genome-wide variations without the use of any prior reference such as collection of (pan) genes or a reference genome, and without omitting the noncoding regions that may be involved in the regulation of gene expression.

Chapter 2

Using cDBG in bacterial GWAS: why and how ?

2.1 Preamble

Improving GWAS approaches to explore and learn about the antibiotic resistance in bacteria is a multi-facet task, as described in the Introduction. Among all, we had at heart to improve the list of variants on which computing the association study. In Chapter 1, we focused the search to a list of resistance determinant genes selected from the literature. While this allowed a quantitative description of the resistome of *P. aeruginosa*, this choice prevented from retrieving new candidate markers outside this gene collection.

As presented in the Introduction, even though the k-mers – the collection of all subsequences of k nucleotides seen in the input genome sequences – seemed a good alternative to a targeted-variant-based description, we wanted to go further and chose to work on compacted De Bruijn graphs (cDBGs) instead. Once our work focused on these graphs, we started a collaboration with Vincent Lacroix (from LBBE/Erable Inria team) and Leandro Lima (a PhD student under his supervision). Their team in particular developed a tool, KisSplice [180], which builds DBGs to describe the alternative splicing events from RNA-Seq experiments. All the work presented in this Chapter is based on cDBGs built with this tool, using the levofloxacin and amikacin resistance phenotypes, and with the reduced panel of *P. aeruginosa* introduced in Chapter 1, Concluding remarks (except for two representations using the full panel).

In this Chapter, we present our work on describing the genetic variants in a bacterial GWAS using the cDBG representation, and to which extent they can improve the k-mer approach. Before presenting the manuscript released on bioRχiv in 2017, we begin the Chapter by the presentation of unpublished preliminary work which drove the choices made to build the method described in the manuscript. This includes an introduction to the cDBG topology, how cDBGs can be *decorated* (enriched with metadata) to highlight side information. We then present different ideas regarding the entities we could extract from the cDBGs and test for association with the phenotype, and a comparison of several ways to model the phenotype. Part of this preliminary work was presented as a poster in 2015 at the *Statistical analysis of massive genomic data* conference in Evry, and at talks in 2016 at the *Statistical Methods for Post Genomic Data* (SMPGD) conference in Lille and *Journées Ouvertes en Biologie, Informatique et Mathématiques* (JOBIM) conference in Lyon.

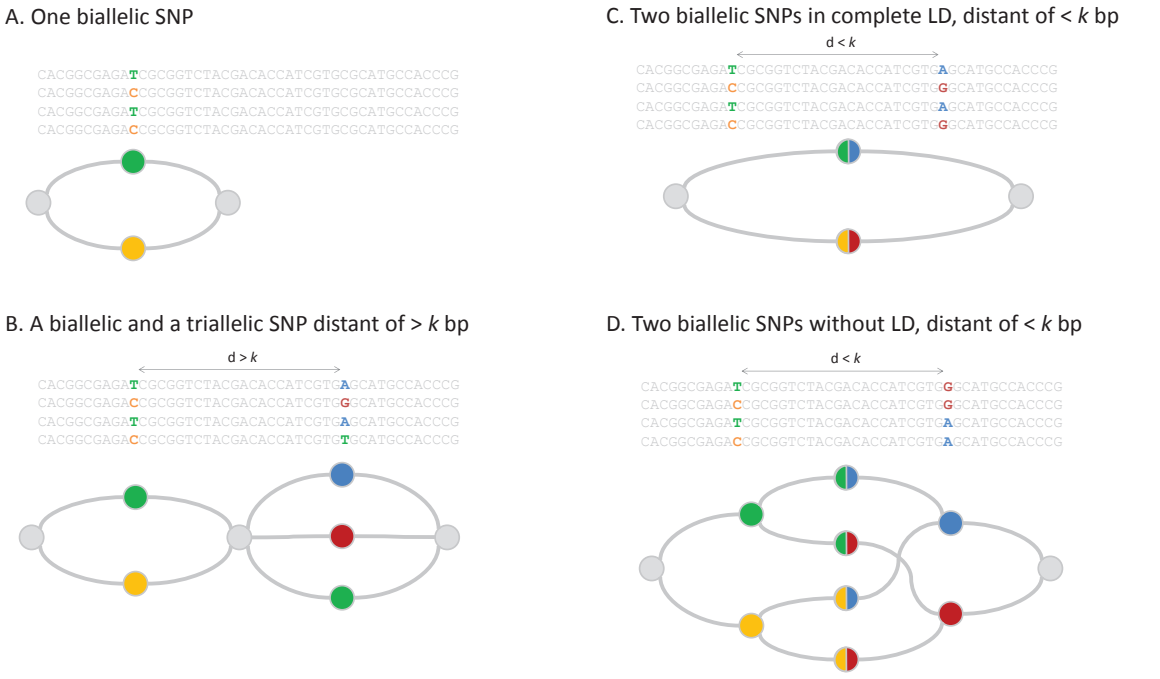


Figure 2.1: **Examples of cDBGs obtained with different configurations of SNPs.** (A) A single biallelic SNP is represented by a simple bubble. Each path of the bubble represents an allele. (B) Two SNPs form two distinct bubbles if they are distant of at least k nt. Multiallelic SNPs generate bubbles with more that two paths. (C) Two SNPs form a single bubble if they are distant of less than k nt and if they are in complete LD. (D) Independent close SNPs generate complex bubbles.

2.1.1 Bubbles and loops in the cDBG

The representation of the variation among a set of sequences, such as bacterial genomes, with a DBG not only keeps track of the links between the k-mers, but also offers a powerful tool to compress the information without loss. As illustrated in Figure 2.8 of the manuscript, the DBG shows a linear path when no variation is observed after a given k-mer: this k-mer determines the presence of the next one. All k-mers in a linear path are thus equivalent for an association test as they are all present in the same set of genomes. In the compacted De Bruijn graph (cDBG), any linear path is replaced by a single node, called a *unitig*, which describes the complete sequence of the replaced path.

In the simple example provided in Figure 2.8, the 4 unitigs forming a bubble in the cDBG carry the same information as the 11 k-mers of the full DBG. In this example, a biallelic SNP led to a simple bubble in the cDBG, whose paths represent sequences of exactly $2k-1$ nt [162]. This case is also illustrated in the more schematic panel A of Figure 2.1, while other panels of the Figure provide other examples of the local structure of the cDBG obtained for situations involving one or two SNP positions. In these examples, the topology of the graph depends on the number of alleles observed at the mutated position, on the distance d between the SNPs, and on the linkage disequilibrium (LD) between the loci.

The maximal number of outgoing edges for a unitig in a cDBG is given by the length of the alphabet [52], here, the four canonical nucleotides $\mathcal{A} = \{A, C, G, T\}$. The number of outgoing edges correspond to the number of alleles seen at a given position p defined by the k-mer at position $p-k$ in the dataset. As an example, Figure 2.1B represents two

SNPs, one biallelic, involving two nucleotides (T or C) which is represented by a bubble with two paths, and a triallelic SNPs represented by a bubble with three paths (A, T or G).
When the two SNP positions are distant of $d > k$ nucleotides (Figure 2.1B), each SNP is represented by a bubble, and both bubbles are separated by a unitig representing the more than k nucleotides in common within the sequences. In Figure 2.1C, on the contrary the two positions are separated by less than k nt: the bubble, opened with the first SNP, cannot close before the next one. In this example, only two haplotypes (TA and CG) are represented in the sequences: the cDBG shows a simple bubble topology where each path (of length $2k + d$) represents one of the haplotypes.
In Figure 2.1D, as in Figure 2.1C, there are two SNP positions separated by less than k nt, however the alleles are not correlated between the two positions: the four possible haplotypes (TA, TG, CA, and CG) are observed, which results in a more complex topology.
More generally, regions which are poorly conserved within all genomes, or present in multiple copies in some genomes can rapidly present highly branching structures, as opposed to conserved regions such as housekeeping genes.

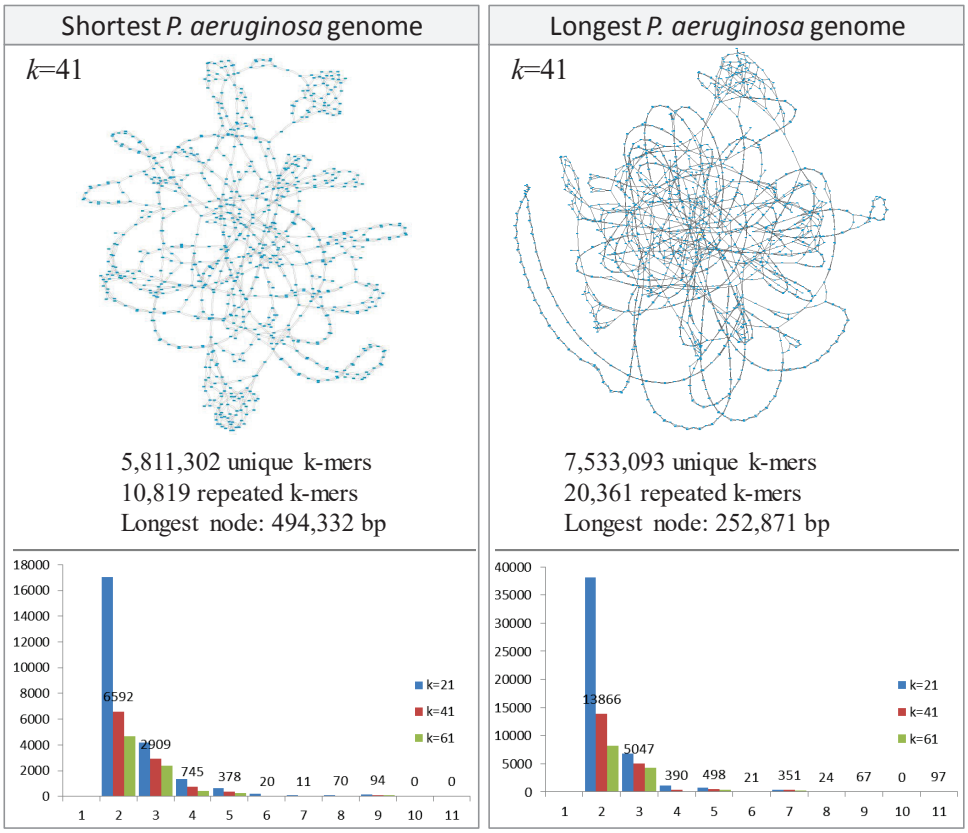


Figure 2.2: **Representation of the sequence repeats of length ≥ 41 in a single genome.** We used the cDBG framework to visualise the loops formed by the repeats in the shortest (left) and longest (right) genomes within our *P. aeruginosa* panel, with $k = 41$. The histograms below represent the number of times each repeat was seen (X-axis), and the number of repeated k-mers (Y-axis), for 3 values of k: 21, 41 and 61.

The choice of the k value used to construct the cDBG drives the variant resolution level: as presented in Figure 2.1B, a small k value can describe individual positions while a large k value can describe haplotypes (Figure 2.1C). The notion of 'small' and 'large' values depends on the variability context, quantifiable by the frequency of the mutations observed

at the genome population. The choice of k also drives the k-mer sequence specificity and thus the number of repeated k-mers, hence the graph topology, as repeated k-mers form loops in the cDBG. This is illustrated in the manuscript Figure S2.4: the gene *gyrA* is represented by six cDBGs for a population of 665 *P. aeruginosa* strains. We used different values of k from 11 to 61 and observed numerous loops due to unspecific 11-mer repeats at $k = 11$. We discuss the choice of the k value in the Method section of the following pre-print.
A cDBG can also be built on a single genome in order to analyse its repeats. Indeed, within a single genome, forks only happen when a k-mer is repeated in the genome. Figure 2.2 shows the graph topology obtained for the shortest (5.84Mb) and longest (7.59Mb) genomes from our *P. aeruginosa* panel, with $k = 41$. The longest unitig correspond to the longest sequence found in the genome with no repeat of length 41 nt or more. The histograms below the graphs represent the number of repeats found for 3 values of k, and the number of times the repeats were seen. For instance, 97 k-mers were found 11 times in the longest genome, with $k = 41$.
The information provided at a single-genome-level can also be of help in the choice a the best k value. This is the strategy used by kSNP [73], a tool to build k-mer-based phylogenies: they define the best k value as the one producing 1% of repeated k-mers.
Even though the repeated k-mer counts can be used to assess the presence of multiple copies of genes in the genome, we did not integrate this information in the remainder of the work. Indeed, we worked on genome assemblies, which already compress the repeats [163].

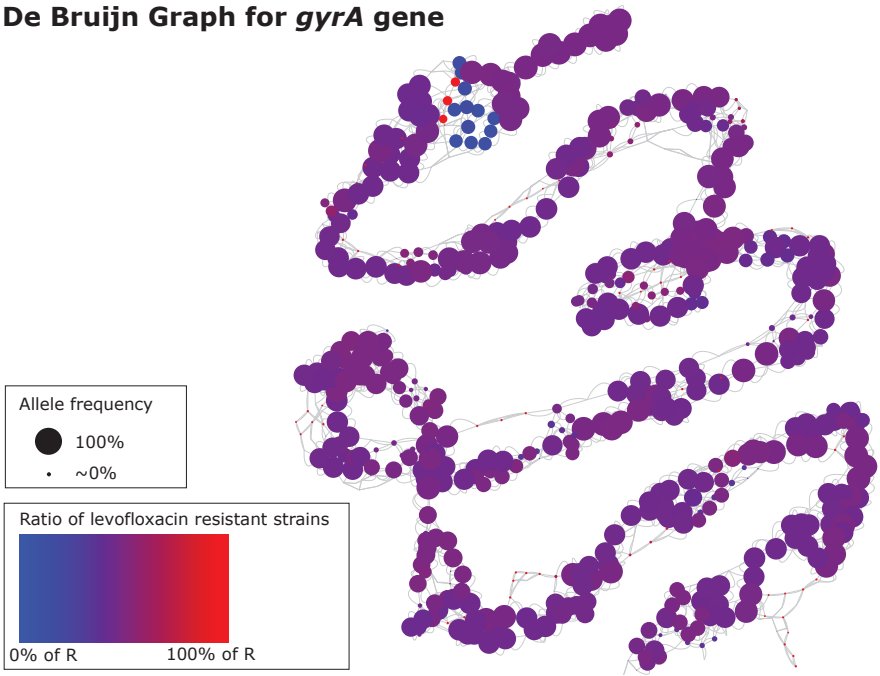


Figure 2.3: **Coloured cDBG of the *gyrA* gene from 665 strains of *P. aeruginosa*, built with $k = 41$.** This graph representation is generated by decorating a cDBG with two metadata: the allele frequency is used for the unitig size and the percent of non-susceptible strains is used for the unitig colour.

2.1.2 Adding decorations, for a better comprehension

Metadata can be added to the cDBG to represent unitig properties (size, color, etc.) and offer a more comprehensive representation [93, 151], beyond the graph topology. We

tested the feasibility of using such decorated graphs in the context of a genotype-to-phenotype study. We chose for this purpose to work at the gene instead of the genome level, and we selected the *gyrA* gene, a target of antibiotics of the quinolone family such as the levofloxacin. This gene, present in all strains, codes for a gyrase involved in the vital replication function. Under high selection pressure, some strains acquire mutations in a particular region of the gene, known as the quinolone resistance-determining region (QRDR), which alter the antibiotic target recognition and confer resistance to levofloxacin [90, 130]. Moreover the association study described in Chapter 1 showed that mutations in *gyrA* gene had the highest effect on the levofloxacin MIC.
We extracted *gyrA* gene from the genomes by homology search, then, from the *gyrA* gene sequences, built a unique cDBG using the KisSplice software. Each unitig sequence was remapped on the *gyrA* gene sequences to identify to which strain it belonged to and deduce a proportion of non-susceptible strains per unitig. We used this proportion to set a colour scale from blue to red, and applied it to the unitigs. We used the unitig allele frequencies – as defined by the proportion of strains mapping the sequence represented by the unitig – to set the diameter of the unitigs.

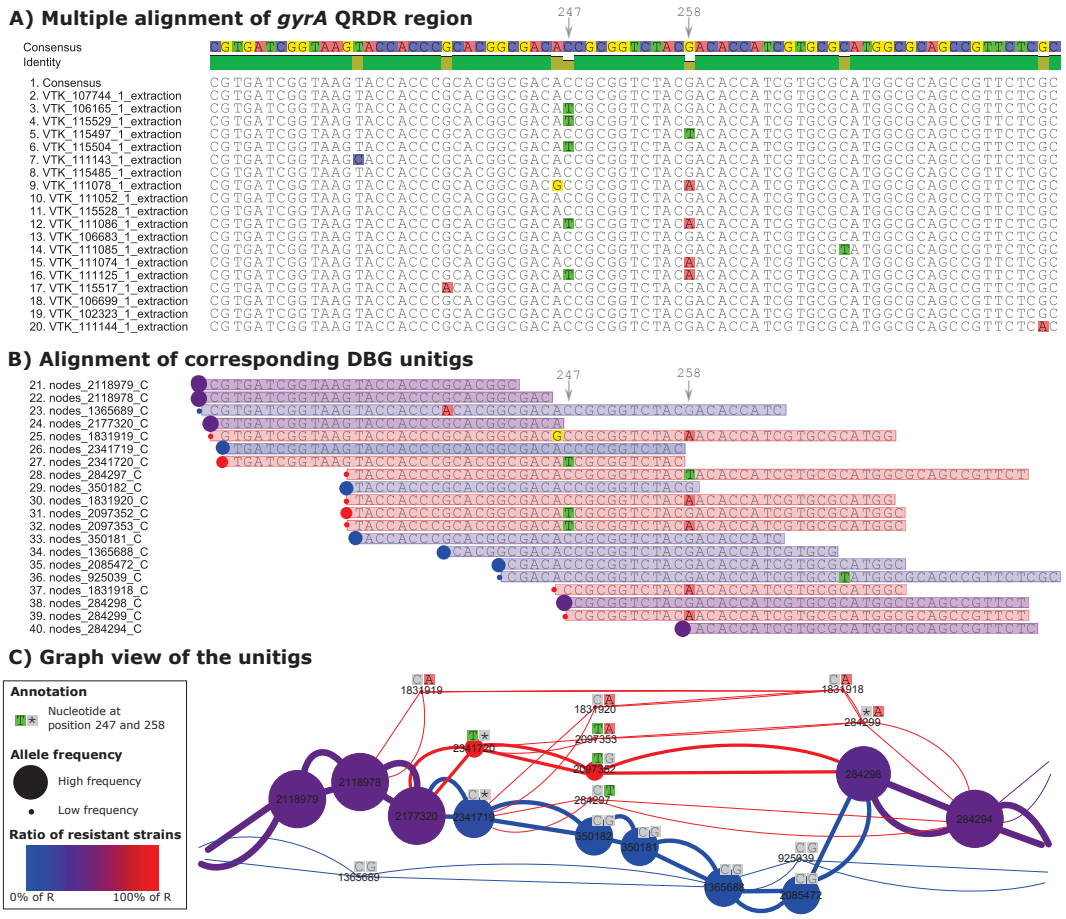


Figure 2.4: ***gyrA* QRDR region: from a multiple alignment to the cDBG.** (A) Sequences corresponding to *gyrA* QRDR were extracted from the genomes and a small number of representatives of each variant was kept for the multiple alignment. Mutations at nucleotide positions 247 (corresp. to the *aa* position 83) and 258 (*aa* 87)are both known to confer resistance to quinolones. (B) unitigs from the DBG are mapped against the alignment. Each mutation at any position creates a fork pattern and thus impacts the beginning and end of the unitigs. (C) Graphical view of the unitigs mapped on this region. Thanks to associated metadata, prevalent paths conferring resistance are highlighted, such as sequences having a base 'T' at position 247.

The resulting topology, shown in Figure 2.3, presents a globally linear structure, following

the gene sequence from one extremity to the other. Furthermore, this enriched representation allows for an obvious visualisation of the QRDR: in this region, the unitigs are specific to the antibiotic resistance status and appear either blue (susceptible strains) or red (non-susceptible strains), as opposed to the other unitigs of the graph, which appear purple (seen in both susceptible and non-susceptible strains). Retrieving the sequences corresponding to these unitigs would allow to find the expected associated mutations easily.
The mapping of features from the graph to the unitig sequences, and then from the unitig sequences to a multiple alignment of the QRDR allowed to better understand the link between a representation with a cDBG and with a multiple alignment. As a result, cDBGs summarise the information contained in a multiple alignment, by offering a local view of each variant, as illustrated in Figure 2.4. Moreover the decorated cDBG offers an easier way to highlight the most common haplotypes: frequent paths are represented as large circles while rare variants as small circles. Using this mapping, SNPs on nucleotide position 247 (corresponding to the *aa* position 83) and 258 (*aa* position 87) are retrieved (red paths) and we get an additional information on their prevalence: the mutation C $\rightarrow$T in position 247 is more prevalent than any mutation on position 258.
This preliminary work revealed that cDBGs were an appealing framework to represent genetic variations in bacterial GWAS. Indeed while a bubble represents a divergence between two genotypes, the cDBG built from all input genome assemblies provides a direct description of the variations at any position in the genomes. Encouraged by the *gyrA* example, we searched strategies to identify such coloured patterns at the complete genome level.

2.1.3 How to use the cDBG for a GWAS?

A visual inspection of a *pan-cDBG* (the cDBG built from all genomes, *i.e.* at the pan-genome level) was not a realistic approach for identifying genotype-to-phenotype associations, when the number of unitigs reached the millions. This identification had to be obtained computationally. The first natural choice was to use bubbles, as it is the solution implemented in DBG-based variant search [93, 162, 180].
Bubbles are very efficient to capture SNPs and indels in a population, but other types of variations are also expected to be related to the phenotype, such as the insertion of resistance genes, which are themselves polymorphic. Figure 2.5 shows the pan-cDBG obtained for a simulated insertion of a resistance gene into a conserved site. The sequence of the inserted gene is variable within the population: SNPs and indels in the gene generate a succession of small bubbles, some of which are branching. The insertion as a whole could be represented by a large bubble: using the same notation as in Figure 3, we represented, over the graph figure, both flanking sequences a and b on each side of the bubble, path σ representing the part of the population which does not have the insertion and path σ' representing the insertion. However, as the inserted sequence is polymorphic, a very large number of possible paths describing σ' exists. Moreover, in practice, accessory resistance genes are not always inserted in a single insertion site: the presence of two conserved switching nodes a and b for a given gene insertion is not guaranteed.
Using KisSplice, which is able to identify such branching bubbles, we observed the combinatorial explosion in the number of possible paths produced to represent a polymorphic insertion, and thus the limited ability of bubbles to describe a gene insertion, on a toy example illustrated in Supplementary Figure S2.3. This example represents the simple insertion of *aac*(6') variants between two genes – always the same pair. KisSplice identified numerous bubbles corresponding to the long insertion (flagged as alternative splicing

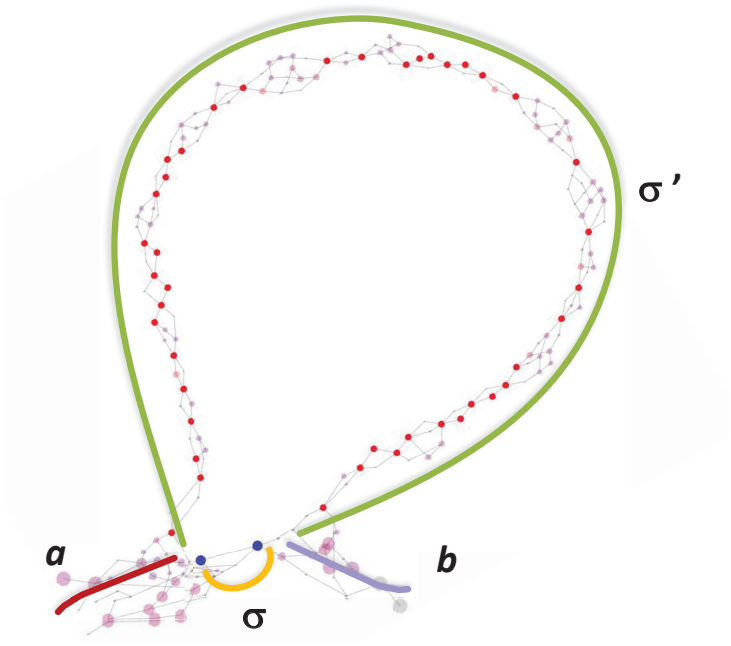


Figure 2.5: **Complex bubble formed by the insertion of an accessory gene polymorphic in a preserved site.** This example was generated with DBGWAS, from a simulation of a gene insertion, as described in Figure S2.1, and with a phenotype generated as described in Eq. 4.1.

events) and each bubble path described a particular variant of the gene. Due to the high variability of the inserted gene, each variant was only observed with a very low count: 1 or 2 observations per path.

We went a bit further in the bubble approach evaluation by building a naive pairwise score to identify the most discriminating bubbles in the context of levofloxacin resistance:

$$score_i = |(R_{upper,i} - S_{upper,i}) - (R_{lower,i} - S_{lower,i})|, \tag{2.1}$$

where $R_{upper,i}$ and $S_{upper,i}$ are respectively the non-susceptible and susceptible genome counts in the upper path and $R_{lower,i}$ and $S_{lower,i}$ the genome counts in the lower path of the i-th bubble. This naive approach did not correct for the population structure and did not provide any control over the level of false discovery rate of the procedure. Moreover it did not take into account the number of individuals per bubble, and was thus biased by the bubble path prevalence.

For this evaluation, we built synthetic genomes associated to real phenotypes: we built these genomes from the 282 genomes of the reduced panel by concatenating their core genes, as described in Supplementary Figure S2.1, A to C. We added to the genomes, as described in Supplementary Figure S2.1D, the sequences of all the levofloxacin genuine variants found significant in the RWAS study presented in Chapter 1 (Supplementary Table S1.3): *gyrA*, *gyrB*, *parC*, *parE*, and the *aac*(6') variants. We could expect high bubble score values for bubbles from these genes. We used the real binarised levofloxacin phenotype to split the genomes between two conditions, non-susceptible and susceptible, and provide them as input to KisSplice.

The highest scores we obtained were for the bubbles overlapping the expected mutated positions of *gyrA* and *parC*, regardless of the value of k, varying between 21 and 61 (Figure 2.6 shows the scores obtained with $k = 41$). The bubbles mapping to the expected *aac*(6') gene, however, did not present a score above the background noise. It was difficult to judge from these results if this experiment failed to retrieve the *aac*(6') gene insertion

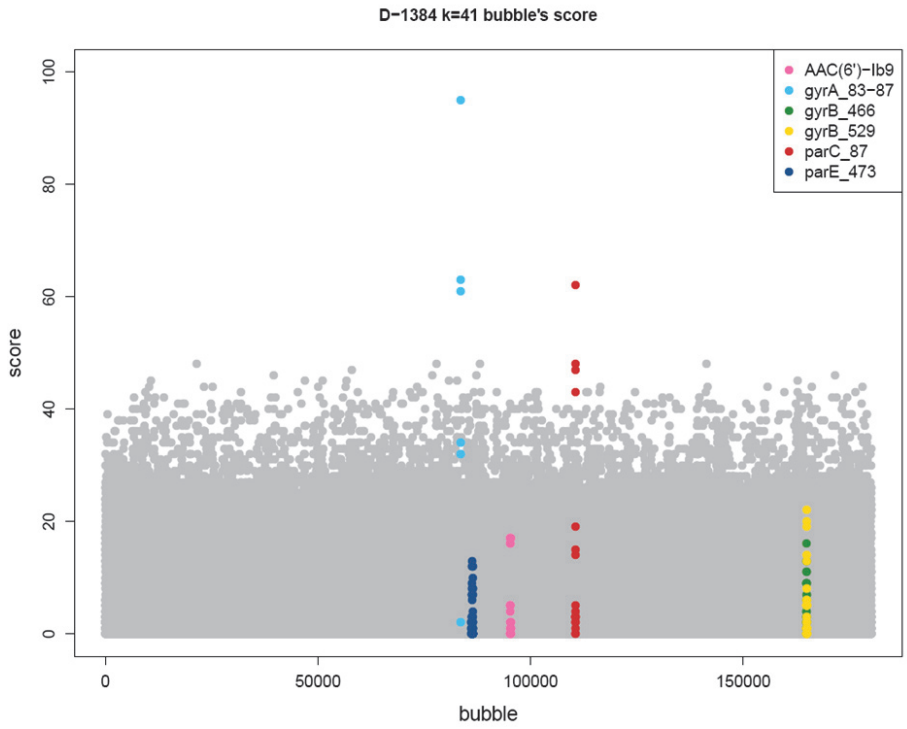


Figure 2.6: **Bubble scores obtained for the simplified genomes.** The 282 genomes were split into two conditions (susceptible/non-susceptible) from their levofloxacin resistance phenotype. The bubble score was computed on all bubbles described by KisSplice, launched on these two sets, with $k = 41$.

because the score was too naive, because of the chosen experiment (*aac*(6') association was tested on the full panel, with categorical MIC while this experiment was conducted on the reduced panel with binary phenotypes), or because the bubbles were not a suitable representation for this type of variant.

The question remained: how to use the pan-cDBG to capture variation in the genomes, including gene insertions, significantly associated to a phenotype ? Such insertions are represented by bubbles which are too complex to be enumerated as a single entity with current bubble-based variant search tools. On the other hand, any complex bubble belongs to a subgraph of the pan-cDBG, and any subgraph can capture potentially phenotype-associated variation. Once decorated, as in Figures 2.3 and 2.5, we can recognise the type of variation represented in a subgraph: a local polymorphism, such as the SNPs in *gyrA* is represented by a subset of blue and red nodes in the decorated cDBG, distinguishing between susceptible and non-susceptible strains (Figure 2.3). By contrast, the integration of foreign DNA in the genome, such as a complete gene acquired by HGT, is represented by several consecutive red nodes forming successive bubbles which represent the gene variability at the studied population level (Figure 2.5).

These preliminary experiments suggested that a good representation of genetic variation should be able to capture decorated subgraphs rather than bubbles. It is however not trivial to build a genotype measure at the subgraph level. While the presence or absence of the unitig sequence is clearly defined for each genome, it does not make sense to ask if a subgraph belongs to a genome or not. We could use operators such as OR (resp. AND) between the unitig presence pattern, however this could rapidly leads to patterns containing only presences (resp. absences). Advances in penalised regressions offer interesting tools and strategies to identify directly the cDBG subgraphs representing

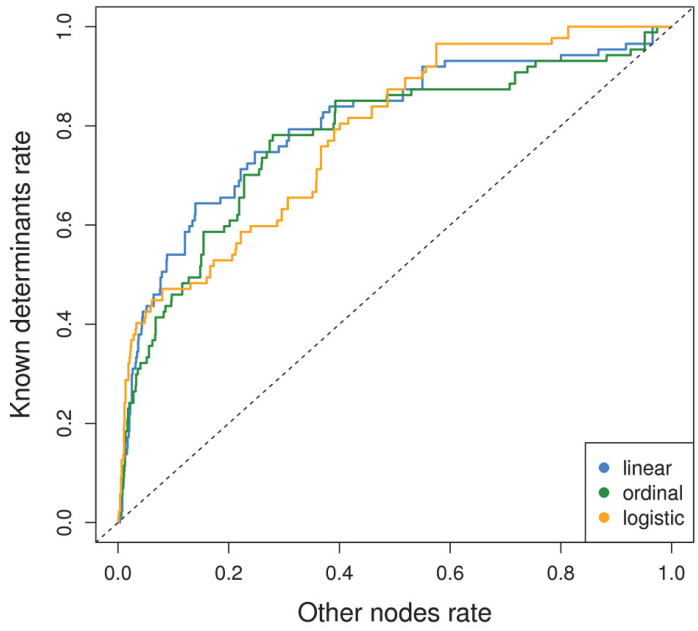


Figure 2.7: **Comparison of three ways to model the amikacin phenotype.** An ordinal, a linear and a logistic models were applied to transformed MIC values, using the presence/absence patterns of the unitigs, and correcting for the population structure.

these regions. The network-constrained regularisation described in 2008 [119] proposed to use the Laplacian matrix of the graph to penalise high differences of estimated effect between adjacent nodes. The path-coding penalty for directed acyclic graphs (DAG) introduced in 2013 [133] proposed a group-lasso-like way to select groups of nodes forming a path, applying constraints on the number of selected paths and on their length. Active-set strategies were applied to RNA-Seq data in 2014 for path discovery in a DAG [19] and allowed, under the assumption of a very sparse solution, to select paths without explicitly enumerating them. Neither of the first two methods select subgraphs explicitly, but nodes or groups of nodes while the third one selects (non-branching) paths. Neither of the last methods are directly applicable to a pan-cDBG as they were developped for DAGs, and, as soon as a k-mer is repeated, a cDBG contain cycles.
An alternative approach is to test unitigs. Unitigs do not represent subgraphs themselves, however their presence or absence can be marker of a gene insertion: In Figure 2.5, while paths describe particular variants, some nodes of the graphs (the red ones) represent regions of the genes that are conserved enough to retain the information of the insertion at the population level. Before trying to build an adapted subgraph-dedicated testing strategy, we thus decided to test the presence/absence patterns of unitigs, and chose to use marginal regressions with multiple testing adjustment. We realised later that the post-processing we built, using the cDBG edge information on the significant unitigs selected from the association test, could be a good proxy to identifying sugraphs directly from a complex model.

2.1.4 Antimicrobial resistance (AMR) phenotype modelling

Our bacterial GWAS framework had, at this point, a variant description (unitig presence/absence patterns among the panel), and a testing strategy (marginal regressions),

which required modelling choices. The AMR raw phenotype in our *P. aeruginosa* panel was a minimum inhibitory concentration (MIC) value. We introduced in Chapter 1 the ordinal model encoding the MIC as ordered categories. We also tested a linear model on the logarithm of the MIC values, and a logistic model on the binarised MIC values. The binarisation was done using the Clinical Laboratory Standards Institute (CLSI) guidelines which determine MIC breakpoints defining susceptible, intermediate and resistant strains. We then gathered the intermediate and resistant categories to create a non-susceptible category. We added to all univariate models a correction for the population structure using the principal components of the matrix of core SNPs explaining 90% of the variance, as factors with a fixed effect.
We applied these three models to the levofloxacin and amikacin resistance phenotypes for the *P. aeruginosa* reduced panel: we tested the hypothesis of a null effect of the unitig pattern on the phenotype and produced p-values using a likelihood ratio procedure, then transformed them using the Benjamini-Hochberg method to guarantee an FDR control in the situation of multiple testing. We finally built ROC curves to evaluate the ability of each model to retrieved the genuine variants described in Supplementary Table S1.3.
All methods showed a similar enrichment of lower p-values for known determinants of amikacin resistance (Figure 2.7): the three methods performed equally to retrieve 70% of the known determinants. However the logistic model presented worse performances to retrieve the remaining known determinants. We hypothesised this was caused by a lack of numerical convergence for some combinations of X and W leading to poorly conditioned problems. Moreover, the logistic and ordinal regressions required a higher computational time than the linear regression. From this comparison, we were confident in using a linear model.
We then wanted to benchmark several population structure correction methods. This comparison, described in the manuscript, integrated a linear mixed model implemented in the R package `bugwas`. Indeed, this package is dedicated to adjustment for the strong population effects observed in bacterial population [61], and also estimates effects at the lineage level. However, this method is restricted to binary phenotypes, even though it applies a linear model. In order to make a fair comparison, we chose to also apply a linear model on binarised phenotypes for other correction methods, as described in the pre-print.

The following manuscript describes the cDBG-based method built from this preliminary work: the variant matrix is built from a pan-cDBG generated by KisSplice, and the presence/absence unitig patterns are used in multiple linear models of a binarised phenotype. We discuss the choice of the population structure correction, and introduce a post-processing procedure using the information on ingoing and outgoing edges of the significant unitigs in the pan-cDBG.

2.2 Pre-print released on bioRχiv (2017)

"Representing genetic determinants in bacterial GWAS with compacted De Bruijn graphs" doi:113563

Abstract

Motivation: Antimicrobial resistance has become a major worldwide public health concern, calling for a better characterization of existing and novel resistance mechanisms. GWAS methods applied to bacterial genomes have shown encouraging results for new genetic marker discovery. Most existing approaches either look at SNPs obtained by sequence alignment or consider sets of kmers, whose presence in the genome is associated with the phenotype of interest. While the former approach can only be performed when genomes are similar enough for an alignment to make sense, the latter can lead to redundant descriptions and to results which are hard to interpret.
Results: We propose an alignment-free GWAS method detecting haplotypes of variable length associated to resistance, using compacted De Bruijn graphs. Our representation is flexible enough to deal with very plastic genomes subject to gene transfers while drastically reducing the number of features to explore compared to kmers, without loss of information. It accomodates polymorphisms in core genes, accessory genes and noncoding regions. Using our representation in a GWAS leads to the selection of a small number of entities which are easier to visualize and interpret than fixed-length kmers. We illustrate the benefit of our approach by describing known as well as potential novel determinants of antimicrobial resistance in *P. aeruginosa*, a pathogenic bacteria with a highly plastic genome.
Availability and implementation: The code and data used in the experiments will be made available upon acceptance of this manuscript.

Contact: `magali.dancette@biomerieux.com`

Introduction

Antimicrobial resistance has become a major worldwide public health concern, as illustrated by the increase of hospital-acquired infections on which both empirical and targeted treatments fail because of multi-resistant bacterial strains [146]. This worrisome situation calls for a better comprehension of the genetic bases of resistance mechanisms. Genome-wide association studies (GWAS) aim at linking genetic determinants to phenotypes, and seem appropriate for this purpose. Indeed over the past four years, bacterial GWAS have shown encouraging results for genetic marker discovery thanks to the increase in rich panels of bacterial genomes and phenotypic data [3, 43, 61, 64, 183].
GWAS rely on a particular definition of genetic variants, such as the presence in the genome of SNPs against a reference genome, of genes in a predefined list or of fixed-length kmers. Each genome in the panel is encoded as a vector with one entry per genetic variant – indicating, *e.g.*, whether the genome contains the variant – and all variants are tested for association with the phenotype of interest. The objective of this paper is to describe a novel representation of genetic variation for bacterial GWAS, and to discuss its advantages over existing ones.
Most existing bacterial association studies use approaches developed for human GWAS to encode genome variation: they align all genomes in the panel against a reference genome, identify SNPs and represent each strain by a presence/absence vector with one entry

per SNP [3, 43, 64]. However a suitable reference is not always available, in particular for species with extensive genome plasticity and a large accessory genome. The accessory genome is the part of the genome not found in all strains of the same species, and is largely composed of genetic material acquired by horizontal gene transfer. For highly plastic species – including pathogenic and antibiotic resistant species such as *P. aeruginosa*–, it can represent more than a quarter of the complete genome, leading to manifold genomes which vary by their size and content [109]. Aligning such genomes against a reference makes little sense and alternative representations of genetic variation are required.
To account for the variation in gene content, some studies also use as candidates the presence or absence of genes represented in the studied panel [61]. However, genetic determinants linked to transcriptional or translational regulation may be located in non-coding regions, and thus are missed by approaches relying on this representation, whose quality also depends a lot on the quality of the available annotation.
Finally to get around these issues, other studies have represented genomes as vectors of presence or frequency of kmers, *i.e.*, of length k sequences in the genome [61, 183]. Contrarily to SNP- or gene-based approaches, kmers are able to describe genome diversity without requiring an alignment against a reference genome or prior annotation. A major issue of this approach however is that the number of distinct kmers contained in a set of genomes increases with k, easily reaching tens of millions of candidates and leading to very high memory requirements, time loads and complexity in feature interpretation. At the same time it is clear that the information carried by kmers is highly redundant as each single locus is represented by several overlapping kmers, suggesting that they are not the optimal resolution to describe genome variation. Thus, the best way to encode genomic variation in bacterial GWAS remains an open question [167, 175].
Our proposed representation is based on compacted De Bruijn graphs [52] (DBG), which are widely used for *de novo* genome assembly [163, 223] and variant calling [93, 115]. All fixed-length kmers corresponding to the same long sequence in a set of genomes are represented as a single longer word associated with a node in the graph. The nodes of the compacted DBG therefore provide a lossless, data dependent compression of the fixed-length kmers, leading to a resolution adapted to the local variability of the genomes. We show in this paper how using these nodes to define genetic variants for bacterial GWAS indeed leads to selecting a few entities which are easier to interpret and make more sense biologically than fixed-length kmers. We also show how DBGs themselves facilitate the analysis of a set of candidate variants found to be significantly associated with microbial antibiotic resistance. We illustrate these advantages using a panel of *P. aeruginosa* strains with multiple phenotypic resistances to antimicrobial drugs.

Methods

We here introduce our proposed definition of genomic variants, showing how it generalizes two standard alternatives based on presence/absence of:

- SNPs obtained by alignment against a reference genome,
- Fixed-length kmers.

We then detail how we use it in a GWAS context and how we assess its performance.

Encoding genome variation using compacted De Bruijn graphs

DBGs are directed graphs representing overlaps between a set of strings. More specifically, the DBG nodes are all unique kmers extracted from the sequences and an edge is drawn

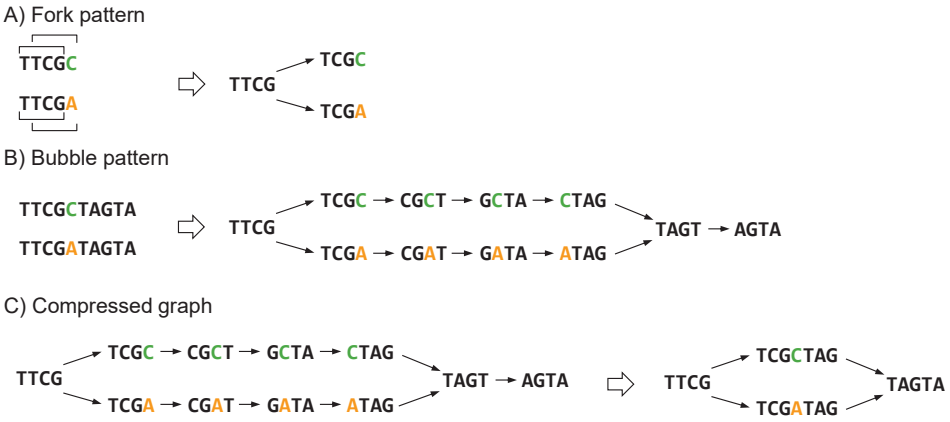


Figure 2.8: **DBG construction.** For this example, k=4. A) the 4-mer "TTCG" present in both sequences overlaps two other 4-mers ("TCGC" and "TCGA") but these two 4-mers differ by their 4th base and we obtain a fork pattern. B) both branches of the fork join on the shared 4-mer "TAGT", and this creates a bubble pattern representing here the SNP C to A. C) linear paths of the graph are compacted ; the remaining graph contains fewer nodes representing longer kmers (unitigs): two 4-mers and two 7-mers instead of eleven 4-mers before compaction. Compacted nodes have variable length.

between any two nodes if the (k-1)-length suffix of one equals the (k-1)-prefix of the other. When considering a set of similar sequences, a single DBG built over all these sequences displays a particular topology, providing information on any variation among sequences in the set. A SNP for example leads to kmers which are constant across genomes, followed by kmers differing by one letter, followed by more constant kmers. When building the DBG, if a kmer overlaps two other kmers but these two kmers differ by their kth base we obtain a fork pattern in the graph (Figure 2.8A). When both branches of the fork join again into one shared kmer, we obtain a bubble pattern with two branches of equal length, representing the SNP (Figure 2.8B). Insertions of large sequences in some of the genomes lead to bubbles with one branch longer than the other, and can therefore be represented in the same framework. This makes DBGs a tool of choice to describe genomic variants [115]. Interestingly, these graphs can be compacted by first using a unique node to store a kmer sequence and its reverse complement, and then merging linear paths, *i.e.*, sequences of nodes not linked to more than two other nodes. This compression is done without loss of information, because it only affects redundant descriptors, *i.e.*, kmers whose presence/absence pattern is identical across genomes [34, 44, 221]. Thus, the nodes of the compacted DBG can be thought of as haplotypes of variable length in different regions of the genomes, including coding and noncoding regions as well as core and accessory genome (Figure 2.8C). In the remainder, we denote by unitig the variable-length kmer associated with a node in the compacted graph.
Rather than representing genomes by presence/absence patterns of SNPs, full genes or fixed-length kmers, we propose to use presence/absence patterns of these unitigs. We discuss in Section *Unitigs, SNPs and fixed-length kmers* how they generalize in an adaptive fashion existing representations based on presence/absence patterns of fixed length kmers or of SNPs defined by alignments against a reference genome.

Unitigs, SNPs and fixed-length kmers

When dealing with a clonal panel of very similar genomes, genomic variants in prokaryote genomes are classically defined as the presence/absence of SNPs identified by alignment of

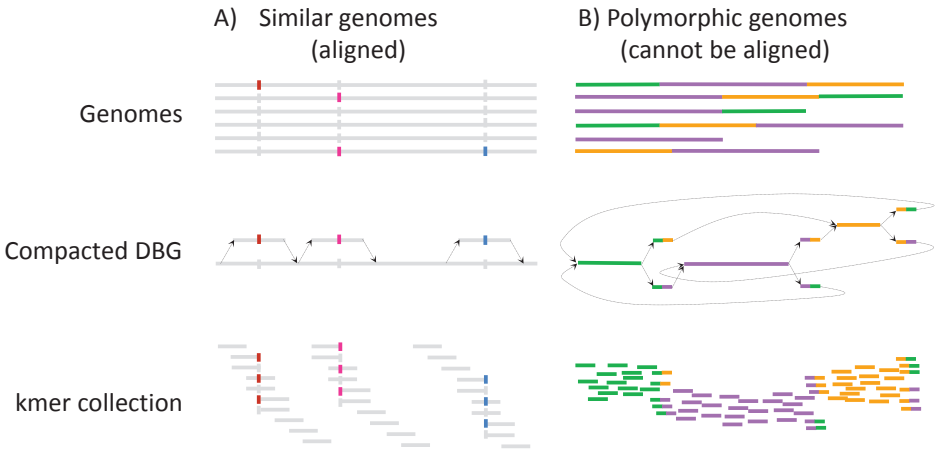


Figure 2.9: **Alignment to a reference (when possible), DBG and kmers obtained for similar (A) and very polymorphic sequences (B).** In the first case, the 3 loci represented as polymorphic in the alignment lead to 3 bubble patterns in the DBG, and numerous redundant kmers. In the second case, genomes are so polymorphic that an alignment is not possible. The DBG summarizes well the common regions and the links between them, while the collection of unique kmers still contains redundancy.

each genome against a reference. For highly plastic genomes on the other hand, alignment against a single reference genome is unsuitable and genomic variation is often encoded as the presence/absence of fixed-length kmers in the genomes. The presence/absence of unitigs of a DBG built over the genomes of several individuals provides a flexible representation thereof which interpolates between these two alternatives in a data adaptive fashion.

At one extreme in the case of a clonal panel with only SNPs as genetic variants (Figure 2.9A), the DBG is a path with a few bubble patterns – assuming genomes do not contain repeated regions longer than k. This graph is isomorph to a reference genome with SNPs. On the other hand, the collection of fixed-length kmers belonging to these genomes is very redundant: all kmers containing the same SNP at different positions have the same presence/absence across strains by construction. Those containing no SNP – most of them – do not represent any polymorphism: they are present in all genomes in the panel and their presence/absence representation would be 1 identically across strains.

As variability across individual genomes increases and alignment of the genomes becomes ill-defined (Figure 2.9B), the DBG drifts away from a path to accommodate local variation beyond isolated SNPs. Fixed-length kmers are also able to represent this variation but still contain a lot of redundancy: all kmers with a given color arise from the same larger colored segment (or junction between segments). They correspond to the same unitig, and their presence/absence across strains is the same. By contrast, the DBG exploits the fact that some regions can be more or less polymorphic across genomes to compact redundant kmers into single longer non-redundant unitigs: their presence/absence across strains is different – unless the corresponding regions are present in the exact same set of genomes because of linkage disequilibrium (LD). In the extreme case where genomes in the panel have so little in common that no compaction is possible in the DBG, the unitig representation reduces to the fixed-length kmer representation.

In this sense, unitigs always represent the best of both worlds between a SNP-based representation of genetic variation and one based on a set of unstructured fixed-length kmers. It results in a locally optimal resolution: regions of the genome which are conserved

across individuals are represented as single long words while regions which are too variable are fractioned into shorter structured kmers.
In addition to removing redundancy compared to fixed-length kmers, DBGs maintain an information regarding how kmers follow each others in the panel, and can be used to interpret those whose presence in the genome is associated with resistance by visualizing the proportion of resistant strains in which they are present. We use these facts in Section *Making sense of the selected patterns using the compacted DBG* to interpret our results.

Choice of k

Each choice of a fixed-length k leads to a different DBG (Supplementary Figure S2.4), and there is no general rule as to how to choose k. Small values of k produce very connected sets of non-specific kmers which fail to represent the specificities of the data. In particular, any region larger than k which is repeated in two different parts of the genome creates a cycle in the DBG. On the contrary, large values of k can fail to create 2 different nodes for 2 different SNPs separated by less than k bases. In this case, the 2 SNPs will be considered as a unique variant. We tested a few values of k and judged by the general aspect of the DBG obtained on our panel and by the GWAS performance, as detailed in Supplementary Figures S2.4 and S2.5. We fix k to 31 for the rest of this study, as this value leads to both an exploitable topology for the DBG built on the gyrA gene, and good performances on GWAS. We found our results to be robust to small variations of k. We discuss the effect of k in more detail in Section *Extracting fixed-length kmers and unitigs from complete genomes.*

Testing procedure

We build our test using a linear model relating resistance phenotypes to a candidate genetic determinant and population structure. Let n be the number of observed samples (*i.e.*, strains with available genome and phenotype). When testing any particular haplotype (presence/absence of a unitig or a fixed-length kmer in the genome) for association with the resistance phenotype, we use the following model:

$$Y_i = X_i\beta + W_i^\top \alpha + \varepsilon_i, \quad i = 1, \ldots, n, \tag{2.2}$$

with $\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, $\sigma^2 > 0$. For any sample i, Y_i is a binarized antibiotic susceptibility status: 0 for susceptible strains and 1 for non-susceptible (resistant and intermediary) strains, X_i is 1 when the sample has the minor version of the haplotype, 0 otherwise. We discuss the set of tested candidates X in Section *Genome-wide variant matrix building.* β is the effect of the haplotype on the phenotype, $W_i \in \mathbb{R}^l$ is a factor representing the population structure, $\alpha \in \mathbb{R}^l$ is the effect of this population structure on the phenotype. We choose to use a linear model rather than a logistic one even though our outcomes Y_i are binary: we tried a logistic model in preliminary experiment, but obtained worse detection performances. Many combinations of X and W factors indeed led to poorly conditioned optimization problems and poor numerical solutions. The logistic model also led to much longer computation for the test.
Our objective is to detect haplotypes whose presence in the genome is associated with antimicrobial resistance. Formally for each haplotype X, we test $H_0 : \beta = 0$ versus $H_1 : \beta \neq 0$ in model (2.2).
It is well known from the human GWAS literature that spurious associations can be detected if the effect of the population structure is not taken into account [11, 207, 226]. For example, assume a clade contains only resistant individuals because a mutation acquired

by a common ancestor of this clade confers resistance. Then all other mutations which are acquired later in evolution and are more present in the clade will also be found to be associated with the resistance phenotype. Population structure can be very strong within bacterial strains [61, 63]. We estimate this structure from the whole design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, where p is the number of unitigs or kmers (as discussed in Section *Genome-wide variant matrix building*, $\mathbf{X}$ typically has several identical columns). We evaluated with three models on both simulated and real data: (i) no correction, (ii) fixed effect α and (iii) random effect α. Denoting $\mathbf{X} = U\Lambda V^\top$ the singular value decomposition (SVD) of $\mathbf{X}$, we use $W = U_q$ (the matrix formed by the first q columns of U) in the fixed effect model and $W = U\Lambda$ in the random effect model. For the first two models, we compute p-values for H_0 using a likelihood ratio test. For the random effect model, we use the bugwas implementation of [61] to test H_0, providing a pre-computed population structure W. Note that bugwas also offers to detect "lineage effect", namely columns of the population matrix W which are associated with resistance, as a mean to avoid throwing away candidates whose association is explained away by the population structure: some of them could actually be causal, and bugwas would return the whole lineage along with correlated candidates as a lower resolution entity.

Genome-wide variant matrix building

One goal of this paper is to illustrate the advantage of testing unitigs rather than fixed-length kmers for association with antimicrobial resistance. To do so, we need to represent both fixed-length kmers and unitigs as 2-levels factors coding for the presence/absence of kmers/unitigs in model Eq. (2.2). More precisely, we consider both as generalized haplotypes with two alleles: presence or absence of the kmer or unitig in the genome. We then express each haplotype as a binary vector $X \in \{0,1\}^n$, with $X_i = 1$ if sample i has the minor allele (the less frequent one across the dataset), 0 otherwise. Consistently with [61], we refer to such a binary vector as a pattern in the remainder of this paper, to emphasize the difference with the actual kmer or unitig they represent. Different kmers or unitigs can indeed be represented by the same binary vector because their presence/absence pattern across the genomes is the same. We only perform one test for each unique pattern (presence/absence binary vector), but retain the link between each pattern and the corresponding kmers and unitigs for later interpretation.

Both fixed-length kmers and unitigs lead to the same set of distinct patterns (represented by vectors in $\{0,1\}^n$) across the genomes. Indeed, every unitig represents (at least) one fixed-length kmer, and conversely every fixed-length kmer is represented by one (single) unitig. As a consequence, the set of patterns tested for association with microbial resistance is identical for the two representations, which further illustrates the fact that using unitigs does not remove information compared to fixed-length kmers.

Every pattern we test often corresponds to a large number of fixed-length kmers. Many of them can come from a single longer sequence of DNA which is either entirely present or absent in each genome of the panel: in this case, they all map to the same unitig. This redundancy is a nuisance because it amounts to artificially fractioning a single pattern into several pieces only because we are not working at the right resolution. For example, the SNP on Figure 2.8 can be represented by one long kmer (unitig) whose only variation across all genomes is at the position of this SNP. Likewise, a unitig can correspond to a gene which is present in some of the genomes but not all of them. In both cases, breaking the unitig into several shorter fixed-length kmers does not bring any additional information and makes the results harder to interpret.

Each pattern in turn typically corresponds to a much lower number of unitigs than kmers.

By construction, two unitigs related to the same pattern cannot correspond to overlapping words – they would have been compacted as a single longer unitig otherwise. They are only redundant in the sense that a genome contains one of the unitigs if and only if it contains the other. This redundancy can be dealt with by inspecting the DBG, as we discuss in Section *Making sense of the selected patterns using the compacted DBG*. The reasons can range from nearby nodes being separated by a rare variant, to two separate genomic regions being in LD.
We build a single compacted DBG from 282 *P. aeruginosa* genome assemblies (see Section *Dataset*) using the kissplice software, version 2.3.1 [180]. A specific aspect of our approach is that we build our compacted DBG from assembled genomes (more precisely, from contigs) rather than from primary sequence reads. This allows us to avoid dealing with sequencing errors, which are present in reads but are mostly eliminated during the assembly process. We choose kissplice settings in order to have no filter on the kmer frequencies or occurrences (-c 0 -C 0.001) and build one DBG per tested kmer length: k=13, 15, 17, 19, 21, 31, 41, 51 and 61 pbs. All resulting fixed-length kmers and DBG unitig sequences are then mapped without mismatch to the original genome assemblies using Bowtie 2 [114] in order to determine the presence or absence of each kmer and unitig in each genome, as this information is not provided by kissplice.

Dataset

We use a panel of 282 strains of *P. aeruginosa* species, a ubiquitous bacterial species responsible of various infections, highly adaptable thanks to its ability to exchange genetic material. The species accessory genome is particularly important, in terms of size and diversity, and carries a large part of the genetic determinants already described to confer resistance to antimicrobial drugs [98]. This strain panel was gathered from two collections including mostly clinical strains: the bioMérieux collection (n=219) [200] and the Pirnay collection (n=63) [164]. Genomes were sequenced on Illumina HiSeq 2500, assembled using a modified version of the IDBA_UD assembler [161], and annotated for the identification of core and accessory genes [200]. Both sequencing and assembly are available on NCBI with accession number PRJNA297679.
Antibiotic resistance phenotypes were obtained by broth dilution assays complemented with VITEK2 testing (bioMérieux, Marcy-l'Étoile, France), for several drugs commonly used in *P. aeruginosa* infections, including amikacin (280/282 strains) and levofloxacin (117/282 strains) [200]. A minimal inhibitory concentration (MIC) value was thus available for all the characterized strain/antibiotic couples. Clinical and Laboratory Standards Institute (CLSI) guidelines were applied on the resistance data to determine susceptibility or non-susceptibility. The reader is referred to [200] and [98] for more information on all strains and their analysis.

Evaluation

We evaluate two complementary aspects of our unitigs. First, we verify that when used in GWAS, they lead to the detection of true genetic determinants on both simulated and real data, under different population structures. Then we assess how insightful the representation is and what type of event underlies each tested pattern.

Ability to detect variants associated with resistance

Quantifying how well a detection method works is difficult, as not all genetic determinants of antimicrobial resistance are known. If a method calls an association between resistance

and a particular variant which was never described as causal it may be a false positive but it may also be because the method discovered a new unreported mechanism. We therefore choose to evaluate how well our test detects true determinants on three complementary indicators.

First, we simulate resistance phenotypes based on our real genomes, arbitrarily fixing which patterns X built in Section *Genome-wide variant matrix building* have a non-zero effect on the phenotype Y. Let $\widetilde{\mathbf{X}}$ be an $n \times q$ matrix whose columns are the unique patterns (in contrast with $\mathbf{X} \in \mathbb{R}^{n \times p}$ whose columns correspond to typically non-unique kmers or unitigs), β is an $\mathbb{R}^q$ vector of the corresponding effects. We sample the phenotype Y_i of each sample i from a multivariate logistic model:

$$Y_i \sim \mathcal{B}(\pi_i), \quad \pi_i = \frac{1}{1 + e^{-\widetilde{\mathbf{X}}_i \beta - W_i \alpha}}. \tag{2.3}$$

Using this set of positive and negatives, we can plot a Receiver Operating Characteristic (ROC) curve for each of the three methods introduced in Section *Testing procedure*. The simulation is multivariate, accounting for the fact that resistance can stem from a combination of causes rather than a single one whereas we are using univariate model Eq. (2.2) for our test. Since we use a logistic model which is the generalized linear model (GLM) of choice to handle binary outcomes, as opposed to the linear model which we use for convenience in our test, it also takes into account the potential misspecification between the model underlying our procedure and the actual distribution of the data. On the other hand, the conclusions we draw from this simulation are contingent upon the capacity of the logistic model Eq. (2.3) to represent the relationship between haplotype and phenotype.

Using the true phenotype data for both amikacin and levofloxacin resistance, we also evaluate a metric based on libraries of known genetic determinants of resistance [97] (mentionned thereafter as reported causal variants) which we use as our positive set. In this case we do not need the assumptions made in the simulation, but we lose the exact knowledge of which haplotypes are negative, *i.e.*, have no effect on the phenotype: some selected patterns may not be linked to any know genetic determinant of resistance just because there are still unreported. Instead of ROC curves, we therefore resort to plotting the true positive rate (TPR) – using identified and hence known positives – as a function of the number of positives called by the method – the false positive rate corresponding to this number being unknown. Assigning each selected pattern (which can represent several mutations or presence of accessory genes) to a true or false status requires a mapping step and some type of approximation: we choose to identify a pattern as a true determinant if it corresponds to at least one kmer/unitig which maps to a known genetic determinant from a resistance gene sequence database [97].

Finally using the true phenotype data for amikacin resistance, we plot the proportion of reported causal variants recovered as a function of the number of kmers or unitigs called positive. We restrict ourselves to the million kmers (resp. unitigs) with the lowest p-values. While the first two metrics focus on unique patterns and do not distinguish between kmer and unitig encoding (both leading to the same set of patterns), this third metric allows us to compare the number of kmers and unitigs that need to be inspected to identify a given proportion of all reported causal variants. This number can be different as each presence/absence pattern corresponds to different numbers of kmers and unitigs.

Making sense of the selected patterns using the compacted DBG

The analysis we describe in Section *Ability to detect variants associated with resistance* is necessary because we need to verify that our test actually discriminates between patterns

corresponding to causal variants and those not corresponding to any causal variant. It is however not sufficient to ensure that our procedure is suited to identifying unreported genetic determinants of antibiotics resistance: to be able to perform this analysis, we had to define which patterns were true determinants using annotated SNPs and genes known to be linked to resistance. In addition to being approximate, this definition cannot be used to go beyond recovering existing determinants.

In order to perform this task, we must be able to interpret the selected patterns. Assuming a pattern is found to be associated with resistance in our test, its interpretation in a fixed-length kmer paradigm can be cumbersome: it typically requires to map all kmers corresponding to the pattern to all genomes – as there is no single reference genome in this context – and to make sense of these mappings. For example, one may find that several of these kmers map to similar regions or annotated genes in all genomes. The task can be heavy as each pattern is typically associated to a large number of redundant fixed-length kmers.

Annotation of our unitigs is easier for three reasons. First, the number of unitig sequences to be mapped is much lower than the number of fixed-length kmers, as illustrated on Figure 2.10. Second, unitigs are longer than kmers, making them more likely to map to a unique region in the genome. Finally, the DBG itself and its colored version [93] can help us understand which type of event is associated with a unitig. The colored DBGs we use rely on node sizes to represent allele frequencies, *i.e.*, the proportion of genomes containing the sequence. They also rely on node colors to represent the proportion of resistant strains containing the corresponding unitig, countinuously interpolating between a red node for unitigs found in resistant strains only and a blue node for those found in susceptile strains only.

Concretely, we select a few patterns with lowest p-values from the GWAS results. We then retrieve all unitigs corresponding to these low p-value patterns – some unitigs can share their presence/absence profiles because of LD, and thus are duplicated. We build the subgraph of our colored DBG induced by these top unitig plus all their neighboring unitigs for a given neighboring size s. We refer to this subgraph as the s-neighboring DBG. This representation offers several advantages:

- It can be done regardless of the association of the pattern with the resistance phenotype and whether or not any annotation is available for the studied genomes. Its topology reflects the nature of the variant: bubbles for example correspond to SNPs while paths represent gene insertion.

- Node colors visually help understand which unitigs or more complex subgraphs are associated with resistance. This allows us to identify bubbles (*e.g.* SNPs or indel) whose branches differentiate phenotype status, and can still be done when no genome annotation is available, using only the strain phenotypes.

- Top unitigs which are close to each others in the genomes will be gathered into connected components of the induced subgraph. These components may represent well-defined genomic regions such as genes or mobile genetic elements – not all connected components will correspond to genomic regions however: some may result from repeated regions in the genome. On the other hand, unitigs mapping to different connected components – distant neighborhoods – carry information on LD, *i.e.*, separate haplotypes which happen to be present in the same set of samples, possibly because of the population structure.

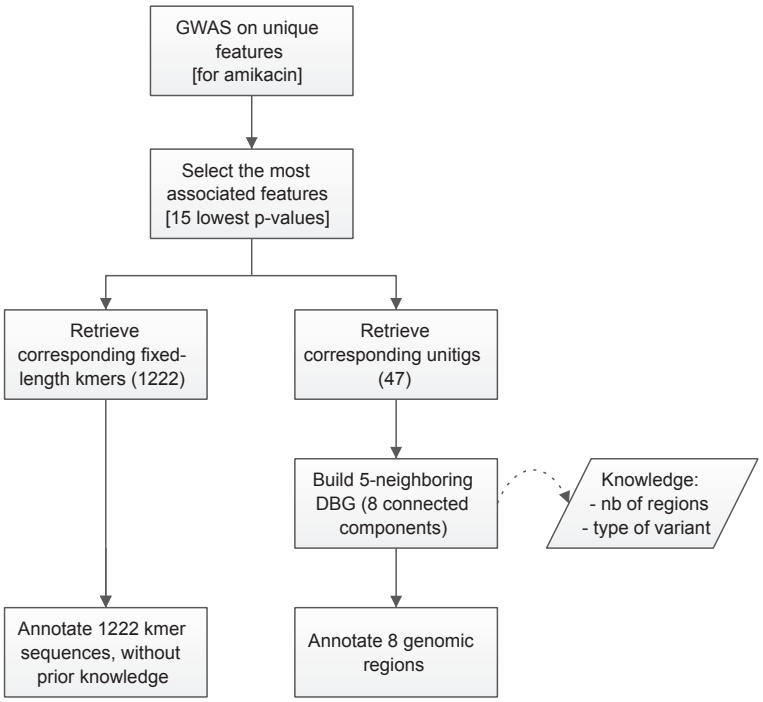


Figure 2.10: **Flowchart of post-processing.** The flowchart is illustrated with the results obtained for amikacin resistance: settings are given between brackets while resulting numbers are given between parenthesis. The annotation burden is lighter when using the DBG unitigs than fixed-length kmers. Indeed in the case of amikacin resistance, the number of kmer sequences to map against all genome exceeds 1000, while using the DBG unitigs we map no more than 47 unitig sequences and can also rely on the identified 8 genomic regions for a complementary interpretation.

Results

We describe the results obtained in our experiments on simulated and real antibiotic resistance phenotypes. We study both the ability of the unitigs to detect causal variants when used in GWAS and the interpretability of the detected objects.

Extracting fixed-length kmers and unitigs from complete genomes

The length k of the kmers used to build the DBG determines how the DBG represents our set of genomes and its ability to provide some level of compression. Small values (below 20) generate words of low complexity which are highly repeated in the genomes, creating numerous loops in the DBG. Consequently, the graph is hardly compacted, as it is very connected and contains few linear parts. For k=15, we only count twice more kmers than unitigs (34 M versus 15 M). As k increases, the number of kmers increases but they become more specific and less repeated within genomes, leading to better levels of graph compaction. For k=41, we obtain 62.5 M kmers and 2.2 M unitigs. More generally, panel A of Figure 2.11 shows that as k increases, the number of kmers increases whereas the number of unitigs remains stable.
Simultaneously, panel B of Figure 2.11 shows that increasing k leads to unitigs of increasingly variable size – larger or equal to k by construction. For k=41, the median length of unitigs is 54 and the longest unitig is 163017 bp long. This illustrates both the redundancy of the fixed-length kmer representation and the capacity of unitigs to produce descriptors whose resolution is adapted to the local variation observed across the genomes.
Panel C of Figure 2.11 represents for each k the percentage of kmers or unitigs which we filter out from our GWAS because their minor allele frequency (MAF) is too low (dark grey). Furthermore as discussed in Section *Genome-wide variant matrix building* several kmers or unitigs can have the same presence/absence pattern on a given set of genomes, so we also represent the proportion of kmers or unitigs which are filtered out from our GWAS because they correspond to duplicated kmers or unitigs (light grey). As expected, this proportion is much larger for fixed-length kmers than for unitigs: a large fraction of fixed-length kmers associated with a single pattern are summarized as a single unitig. This is consistent with the observation that the number of fixed-length kmers is much larger than the number of unitigs but that both representations ultimately lead to the same number of unique patterns.

Phenotype simulation study

We generate synthetic data with two scenarios under model Eq. (2.3) to illustrate the capacity of our test to detect patterns associated with resistance and the importance of adjusting for population structure. This will also help interpret results on real data in Section *Application on real data.*
We use the design matrix $\mathbf{X}$ built from our panel of *P. aeruginosa* genomes. We compute its singular value decomposition $\mathbf{X} = U\Lambda V^\top$ and set $W = U\Lambda$.
Our first scenario is intended to illustrate the case where there is a population effect on the observed resistance (some clades are enriched or depleted in resistant samples) which is not explained by the set of patterns in the tested design X. In practice, this could be a non-genetic (*e.g.* environmental or batch) effect. More importantly, this could happen if some genetic determinants are not included in the model used for testing. This is likely to be the case when we use model Eq. (2.2) which is univariate, *i.e.*, which only considers one pattern at a time. For example, it could be the case that one mutation A causing resistance was acquired by the ancestor of a clade and transmitted to its descendants: the

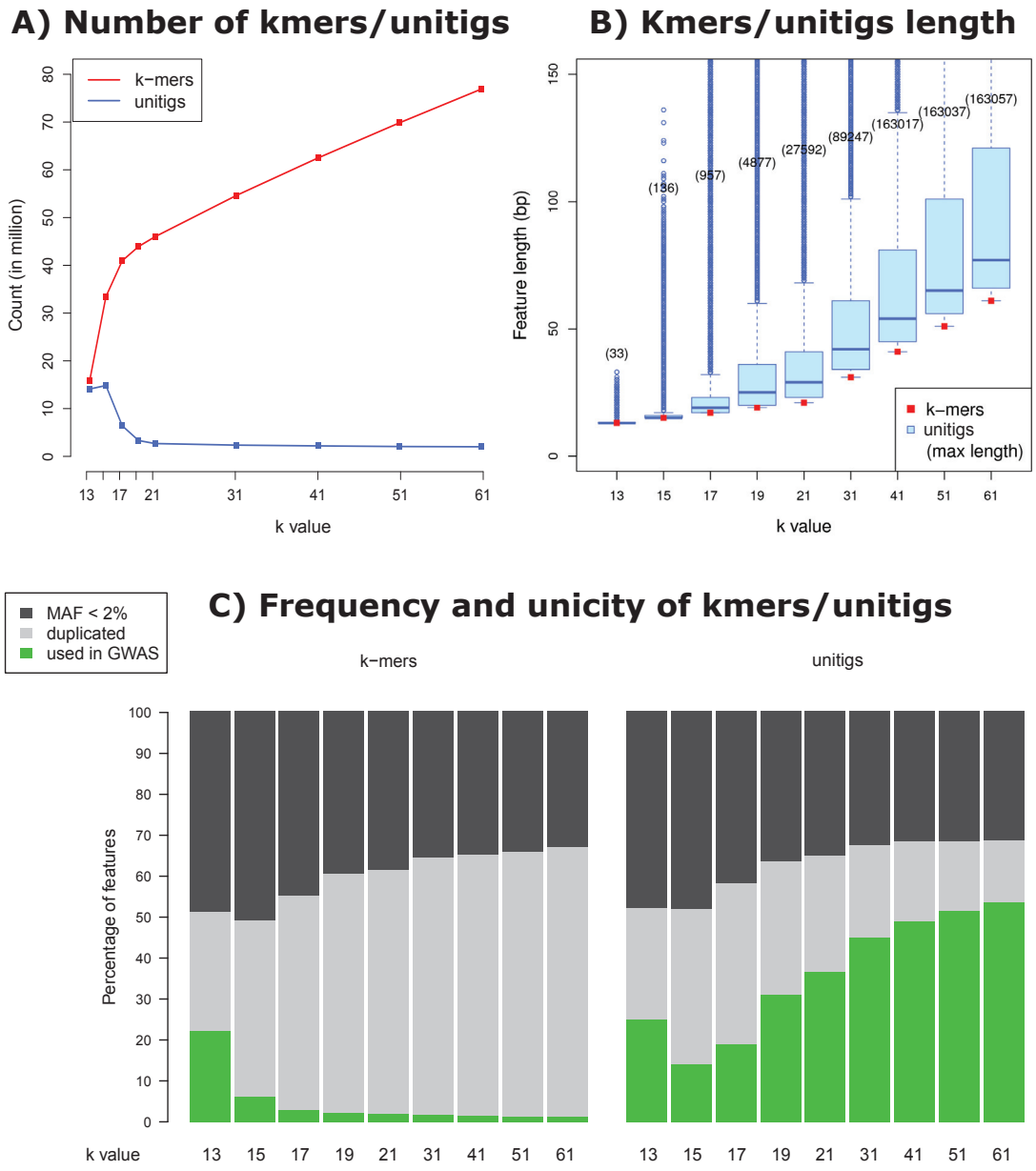


Figure 2.11: **Preprocessing.** Panel A shows the number of fixed-length kmers (red) and unitigs (blue) in the data as a function of k. Panel B shows the corresponding distribution of variable length kmers associated with each unitig. Panel C shows which proportion of kmers and unitigs correspond to unique presence/absence patterns in the data.

clade would then be enriched in resistant individuals. If a second mutation B not related to resistance is acquired by a close descendant of the common ancestor and transmitted, many samples from the clade will also have mutation B. A univariate test of association of B with resistance will not account for A. If the test does not account for population structure either, it may assign a smaller p-value to B than to other mutations with an actual causal effect, *e.g.* because these mutations involve fewer individuals, which leads to a lower power to detect true determinants.

To simulate this scenario, we arbitrarily assign two columns of W (the second and the sixth) to have non-zero effects α, so $l = 2$. By construction, the first columns of W represent a large fraction of the variation across strains. A non-zero effect α in the GLM Eq. (2.3) used to simulate resistance phenotypes therefore makes resistance associated with the population structure. We then select 10 distinct patterns from $\widetilde{\mathbf{X}}$ as true determinants (*i.e.*, coordinates $j \in 1, \ldots, r$ associated to non-zero effects β_j). To do so, we compute the largest dot product of each pattern with the first six columns of W (two of which have non-zero effects α), and choose our true determinants among those whose largest dot product is below the fifth percentile of dot products calculated across all patterns. This allows us to simulate the case where true determinants are independent from the population structure (their effect is not inflated by the $W\alpha$ term). The odd ratios e^{β_j} are fixed to 6 for these patterns. We also randomly select 290 patterns from $\widetilde{\mathbf{X}}$ as non-determinants, *i.e.*, with a $\beta_j = 0$ effect in the model, so $r = 300$ in our simulation. The population structure can lead to spurious discoveries, as we do not control the dot product between columns of W and these patterns with zero effect. Finally in order to control the amplitude of the population effect, we normalize $W\alpha$ to 6 times the median value of the $|\widetilde{\mathbf{X}}^{\mathbf{j}}\beta_j|$ across non-zero β_j, where $\widetilde{\mathbf{X}}^{\mathbf{j}}$ denotes the j-th columns of $\widetilde{\mathbf{X}}$.

We then apply the three versions of our univariate test described in Section *Testing procedure* to each of the patterns. For the fixed effect correction, we use the first 10 columns of W, and for the random effect correction we provide the entire W matrix to bugwas. We perform 100 repetitions of this simulation, and plot a Receiver Operating Characteristic (ROC) curve after pooling the results (Supplementary Figure S2.6). As expected, the test which does not account for the population structure has very low power to detect patterns associated with the phenotype: by construction, some patterns with zero actual effect have large dot products with $W\alpha$ which inflates the estimate of their effect and leads to false discoveries. Taking the population structure into account in the model improves the power by limiting this inflation.

Our second scenario is meant to illustrate the case where there is little population effect observed on the phenotype except for that caused by the association of modeled causal patterns X with W, *i.e.*, outside of $\widetilde{\mathbf{X}}\beta$ in Eq. (2.3). In other words, we assume that all the imbalance in proportion of resistant samples across clades is explained by patterns in the design $\widetilde{\mathbf{X}}$. This can happen if most of the true causal patterns are not too related to the population structure, *e.g.* because they appeared by homoplasy on several unrelated individuals and there is no imbalance. In this case, correcting for the population structure can decrease the estimated effect of causal patterns which do have some association with this structure, *i.e.*, which were acquired by ancestors. To simulate this scenario, we use the same setting as before but we select the 10 true determinants among those that have a large dot product with W rather than a small one, and set all α effects to zero. We apply the same three tests as in the previous scenario over 100 data generations and plot a ROC curve. This time, we observe the opposite effect as in the previous scenario: correcting for the population structure decreases the power to detect true determinants. Assuming there is a population effect when there is no such effect in reality leads to artificially deflating

the estimated effects of patterns which are associated with the population structure.

Application on real data

We then turn to results obtained from real actual amikacin and levofloxacin resistance measured on this panel.

True positive rate vs number of positive predicted

Supplementary Figure S2.7A is produced by bugwas, and shows the p-value of the test for association of each column of W with the phenotype [61]. In the case of amikacin, two columns are found to have a significant effect at level 0.01, whereas all columns have p-values larger than 0.01 in the case of levofloxacin. Accordingly, Supplementary Figure S2.7B shows that correcting for population structure increases the proportion of known genetic determinants of resistance to amikacin recovered for every number of predicted positives, but decreases this proportion in the case of levofloxacin.
The results on the amikacin resistance phenotype are consistent with our first simulation, where the population structure had a non-zero effect α on resistance: the estimated effect $\hat{\beta}$ of true determinants which are not associated with the population structure (low dot product between X and $W\alpha$) is unaffected by the presence of a population effect while the $\hat{\beta}$ of some patterns confounded with $W\alpha$ but with zero actual effect β are inflated. Consequently, the true determinants are not ranked among the first patterns, leading to decreased performances on Supplementary Figure S2.7. Correcting for the population structure limits this inflation of $\hat{\beta}$ for negative patterns associated with the population structure.
Conversely, assuming there is indeed no unmodeled effect of the population structure on levofloxacin resistance, corrected models may just underestimate the effect of true determinants whose presence is associated with the population structure, as in our second simulation. For example, if a causal SNP is shared by a clade which is consequently enriched in resistant samples and all the other SNPs shared by this clade also are causal, correcting for the population structure only decreases the estimated effect of the true determinant, leading to decreased performances on Supplementary Figure S2.7.
The random effect approach of bugwas is a good choice on both simulated (Supplementary Figure S2.6 and real data (Supplementary Figure S2.7B) regardless of the effect of the population structure on the phenotype: it outperforms both the uncorrected and the fixed effect approaches in the presence of a population effect and is only moderately affected by the absence of such effect.

True positive rate vs number of explored features

The analysis of Sections *Phenotype simulation study* and *True positive rate vs number of positive predicted* establishes that representing genomes by their unitig content in GWAS allows to discriminate between (reported) causal variants and other variants (including non-causal and unreported causal variants in our experiment on real resistance phenotypes). However necessary, this result does not illustrate an advantage of unitigs compared to fixed-length kmers as both lead to the same set of presence/absence patterns and the analyses of both Sections only involve these patterns.
By contrast, Figure 2.12 shows the TPR for detecting reported causal variants for amikacin resistance as a function of the number of kmers and unitigs called positive – from 1 to 10^6. In other words, this metric indicates which proportion of reported causal variants is recovered after inspecting a given number of elements. The unitigs perform much better

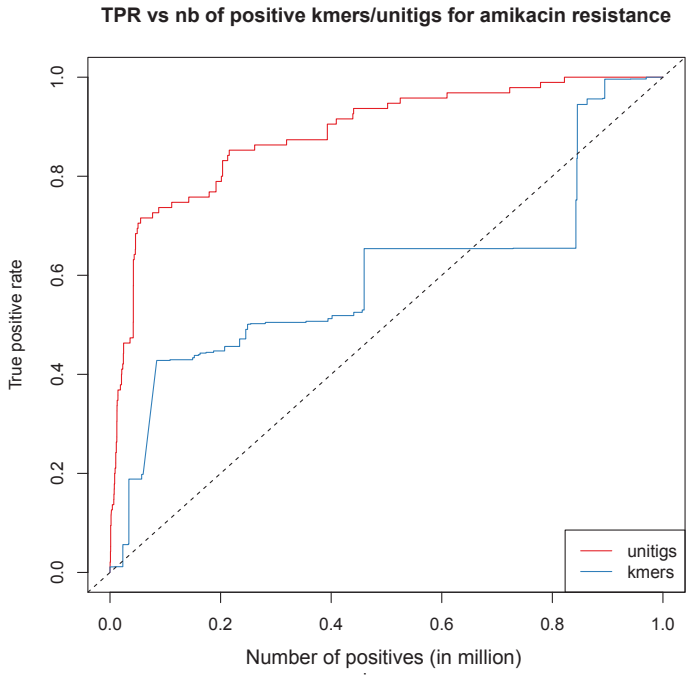


Figure 2.12: **Proportion of true positive versus number of predicted positive kmers and unitigs** for the first 10^6 positive calls using bugwas on the amikacin resistance phenotype.

than the kmers in this metric because every false positive pattern typically leads to a very large number of false positive kmers, and a lower number of false positive unitigs. This illustrates the fact that manipulating kmers is more cumbersome than unitigs as it is necessary to inspect, map and annotate more kmers than unitigs to recover the same number of causal variants.

Analysis of the selected haplotypes

We build the 5-neighboring DBGs from the 15 patterns with lowest bugwas p-values, for both amikacin and levofloxacin resistance, as described in Section *Making sense of the selected patterns using the compacted DBG*. Using the uncorrected or fixed effect approaches leads to very similar lists of 15 patterns.
The top 15 patterns correspond to 47 unitigs for amikacin (resp. 22 for levofloxacin) or 1222 (resp. 262) kmers. The 5-neighboring DBG induced by these unitigs has 8 (resp. 6) connected components whose unitigs consistently map to a small number of annotated events (Supplementary Figures S2.8 and S2.9).
The annotation of the 8 components found for amikacin highlights the importance of the accessory genome in resistance. Indeed, all top patterns map within or near mobile elements: more than half the connected components represent coding or non-coding neighborhood of transposase or integrase. By contrast, half of the 6 components found in the levofloxacin experiment represent SNPs in core gene, recognizable by paths of node with a high prevalence in all strains (violet nodes), and forks that split between a red (resistant phenotype) and blue (sensitive phenotype) path. This matches the current knowledge about levofloxacin resistance mechanism, mainly based on target alteration.
As discussed in Section *Making sense of the selected patterns using the compacted DBG*, the few connected components induced by the top 15 patterns are much easier to interpret than the corresponding large sets of fixed-length kmers. We select 6 of these connected

components (Supplementary Figure S2.8g, a, h and 6b, c, f) and extend their neighborhood up to distance 20 rather than 5 to illustrate the large variety of variants which are selected by our procedure.

SNP in an accessory gene (amikacin)

Figure 2.13A contains a quasi linear structure which evokes a polymorphic gene. The purple color of the structure suggests that the gene is more present in resistant than in sensitive samples, but that the differential of presence is not very important – the nodes would be red otherwise. In the middle of this structure (green box on the figure), the path forks into one blue and one red node, which suggests we have identified a SNP whose presence is associated with amikacin resistance. Note that we are able to make this interpretation regardless of any gene annotation, just by analyzing the topology of the graph component enriched by strain resistance information. Mapping the unitig sequences of this component onto our annotation reveals that the subgraph corresponds to the AAC accessory gene, whose presence is indeed known to be involved in *P. aeruginosa* resistance to amikacin. However, the selected event here is not the presence of the gene but the particular SNP within this gene.

SNP in a core gene (levofloxacin)

Components D to F of Figure 2.13 describe SNPs in core genes. Like in the previous AAC SNP example, each of these subgraphs is a linear structure in which most nodes are present in the same proportion of resistant and sensitive individuals. The linear structure contains a fork which separates resistant (red) and sensitive (blue) samples. Mapping the unitigs on sample genomes reveals that the first two components represent the well-known gyrA (D) and parC (E) quinolone resistance-determining region (QRDR). The third subgraph corresponds to a gene which is not present in our resistance database: the hybrid sensor histidine kinase/response regulator (HS histidine kinase/RR). This gene may be found associated with resistance to levofloxacin because it is in LD with a causal region, or may be itself causal.

Whole plasmid (amikacin)

Figure 2.13B shows a connected component with mostly red nodes assembled in a linear structure suggesting that this entire structure, as opposed to a point mutation, is involved in the detected event. This is in clear contrast with Figure 2.13A, where most of the linear structure is purple with a localized fork involving one red and one blue node. The unitigs of this subgraph corresponding to the top 15 patterns map to the pHS87b plasmid, which was recently described as being involved in resistance [20]. Our representation extracts the whole plasmid, with both its coding and noncoding regions which makes it easier to understand that the selected patterns correspond to an integration of this plasmid.

Noncoding region (amikacin)

The unitigs of the component represented in Figure 2.13C map to a noncoding region in the *P. aeruginosa* genomes. Interestingly, this region contains a path of unitigs strongly associated with resistance (colored in red). Not all of these unitigs belong to the top 15, but the DBG view highlights this long linear structure. This haplotype is not compacted as a single unitig because it is not either present or absent in each genome: some only contain parts of this haplotype.

Alternative approaches

Our approach is able to select and detect any kind of event where current methods could be limited to some regions or patterns. SNPs called against a reference genome are of limited interest in the context of *P. aeruginosa* because of the size of the species accessory genome; causal variants in the accessory genome not represented by the chosen reference would not be detected at all. Gene presence/absence and SNPs called in the pangenome would miss all events in noncoding regions, by construction. Even assuming that only coding

regions are causal, the noncoding region may have a strong association with resistance because of LD, and be among the top patterns in our test whereas the coding region is not because of noise, finite sample or model misspecification. Methods targeting only coding regions would miss the marker in this case. Finally, the gene presence/absence approach would miss the SNP that we identify in the AAC accessory gene. It could have detected the presence of the full gene as being associated with resistance to amikacin, but with less power: only one mutated version of the gene is involved in resistance. Fixed-length kmer approaches are able to target any region of all genomes. However in the case of an event defined by the presence of a complete plasmid such as pHS87b, a fixed length kmer representation would lead to identifying disconnected regions. Identifying the whole plasmid rather than sets of disconnected hits makes it easier to understand which mechanism underlies the selected patterns.

Discussion

We have introduced unitigs as a new and efficient mean to represent candidate genetic determinants in GWAS. Unitigs correspond to variable length kmers: genomic regions which are constant across samples map to single long kmers while more polymorphic regions are supported by several shorter kmers, leading to higher resolution. This representation generalizes both SNPs obtained by alignments against reference genomes and fixed-length kmers. Compared to the former, it is more flexible and can deal with highly plastic genomes. Compared to the latter, it is less redundant and leads to a drastic reduction in the number of candidate entities that need to be tackled without loss of information, leading to easier computation and interpretation of the result. Furthermore, extracting neighboring De Bruijn subgraphs provides additional insight as to what type of genomic event underlies a unitig which is detected as being associated with a phenotype of interest. Experiments on *P. aeruginosa* illustrate that our representation is able to capture very different genomic features ranging from SNPs to large gene insertions.
We conjecture that using unitigs rather than fixed-length kmers could also yield better estimates of the population structure. Typical estimators of this structure are based on representations of the genomes by their haplotypes rather than their unique patterns to avoid down-weighting haplotypes which map to the same presence/absence profile. While duplicated unitigs only represent biological duplicates, *i.e.*, regions in perfect LD, duplicates within kmers also account for neighbor sequence overlaps and can lead to arbitrary inflation of the weight of single long haplotypes. Validating our conjecture that DBG nodes provide better population structure estimates than kmers and lead in turn to more power for detecting genetic determinants requires simulation of synthetic genomes from a given phylogeny and will be the subject of future work.
Finally an important improvement would be to generalize our representation to paths or more general subgraphs of the DBG, *i.e.*, to larger haplotypes defined by conjunctions of those represented in unitigs. This could help filter out minor variations in the genome which are unrelated to resistance but prevent long haplotypes to be merged into a single node. The De Bruijn neigbhoring subgraphs we selected in our experiments suggest that this configuration happens frequently in practice.

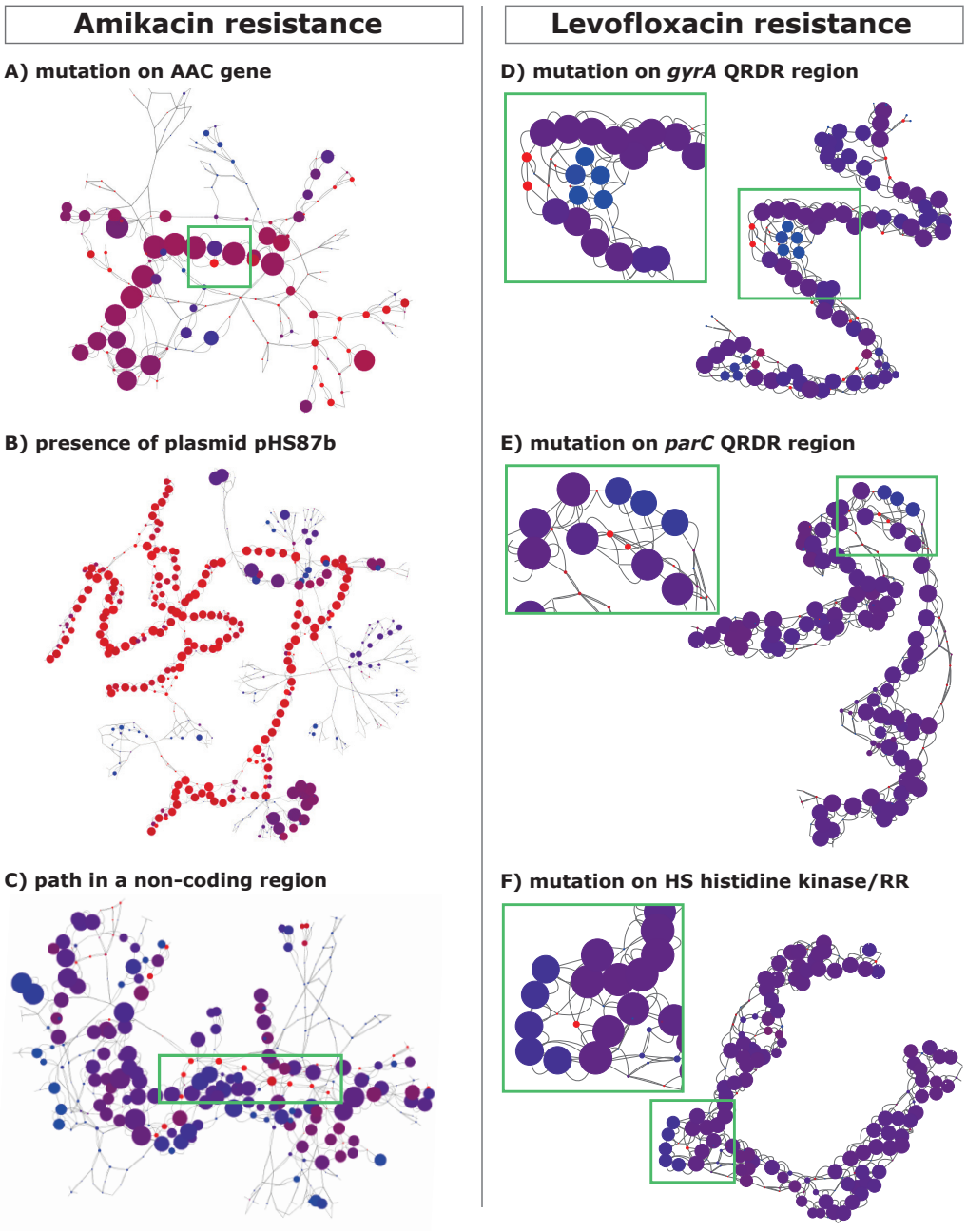


Figure 2.13: **Neighboring De Bruijn subgraphs** Subgraphs of the De Bruijn graph obtained by retaining nodes separated by less than 5 edges from a node corresponding to a pattern whose p-value for association with resistance is among the 15 smallest. Node size represent the frequency of the corresponding haplotype, node color the proportion of resistant/sensitive ratio of samples containing this haplotype, from red for resistant only to blue for sensitive only. The left panel shows the result for amikacin resistance, the right panel for levofloxacin resistance.

2.3 Concluding remarks

In this Chapter, we focused our attention in pan-cDBGs and how they could be of interest to improve bacterial GWAS. We studied the graph ability to represent variations and collapse repetitions, and found how their graphical representation could be useful to highlight phenotype-associated regions, by adding decorations (node colour and size). We tested the use of bubbles to capture differential presence of pairs of paths, however we found this representation not suitable enough to describe the polymorphic insertions of MGE. Because unitigs represent possibly conserved fragments of the inserted MGE sequence rather than a particular insertion of a specific variant – as bubbles do – they can carry the information of the insertion at the population level. We evaluated several strategies for modelling the relationship between the unitig presence/absence in the genomes and the phenotype, and different methods of population structure adjustment in the model. As a result, we selected the the R package `bugwas` [61] which implements a linear mixed model.
From these considerations we designed a bacterial GWAS strategy and introduced a post-processing producing a visualisation of the significant unitigs based on decorated graphs. We showed how this representation can help in the interpretation of the unitig sequences, proposing so a valuable extension to k-mer-based GWAS.
The manuscript describing this method was submitted in 2017 to the *Intelligent Systems for Molecular Biology* (ISMB) conference, but was not accepted. One of the major concerns expressed by the reviewers was that no real benchmark was presented: no comparison with previously developed approaches, including SNP/gene based GWAS, or demonstration of the memory usage and running time of the unitig approach. The scope of the study, only one *P. aeruginosa* panel of limited size, was reproached as it would have be more interesting to see how the method performed across different species, and how it scaled on larger panels . The claimed gain in interpretation did not convinced all the reviewers. Doubts were expressed about the added value of the subgraph representation (Appendix S2.1: Comments from ISMB referees).
In parallel however, the positive feedbacks obtained when presenting the method to (micro)biologists, and their wish to test it, encouraged us to improve the current implementation. Indeed it required some crucial improvements. First, the variant matrix production, based on KisSplice and Bowtie2, was computationally not efficient. Second, the complete pipeline was not fully integrated and we needed to launch several independent scripts to get the final results. Last but not least, visualising the neighbour subgraphs was not straightforward: it required to have the Cytoscape software installed plus some basic knowledges of its usage. It required also complementary analyses to get the subgraphs sorted by significance, or to get some gene annotation.
The development of this integrated tool is presented in the following Chapter. The new tool ended to be so different from the prototype presented in this Chapter, that it justified a new Chapter – and a distinct manuscript submission. However, as both implementations are based on the same method, the content of both Chapters naturally overlaps.

Chapter 3

DBGWAS software for cDBG-based GWAS

3.1 Preamble

Convinced by the potential of cDBG for bacterial GWAS, we were committed to make this approach accessible to the scientific community. The subgraph representation in particular aroused a lot of interest around us.
This motivated the development of a new integrated and efficient tool in collaboration with the LBBE/Erable Inria team. The solution was built mostly in C++, the language of GATB [57], a library for low memory and efficient DBG creation from NGS data. The new tool also used cystoscape.js JavaScript library [67] to produce interactive graphical results without requiring any software installation. The tool usage was also simplified: only one command is called, which requires only one input file. Leandro Lima achieved these developments.
We designed the output together, in particular we chose to present first a summary page with all subgraphs sorted by the minimal q-value found among each subgraph and offering a high level annotation; plus additional pages to navigate within each subgraph, with detailed metadata and annotations at the unitig level. We named this tool DBGWAS, a mix of DBG and GWAS.

This tool was presented at talks at the *Intelligent Systems for Molecular Biology* (ISMB 2017) conference in Prague, at the *Statistical Methods for Post Genomic Data* (SMPGD 2018) conference in Montpellier, and at the *European Congress of Clinical Microbiology and Infectious Diseases* (ECCMID 2018) in Madrid.
The manuscript hereafter, accepted in 2018 for publication in PLOS Genetics, describes DBGWAS, the choices made in its design and implementation (Method section), and explains largely how to read and interpret DBGWAS output results. It includes a benchmark of DBGWAS to other resitome-based and kmer-based GWAS methods. It was written for potential microbiologist users and submitted in a generalist journal to reach this audience. In this context, we included new bacterial panels, most of them comprising around one thousand genomes. A total of 28 GWAS were run, for 3 species (*M. tuberculosis*, *S. aureus* and *P. aeruginosa*) and 28 phenotypes (9 to 10 per species).

3.2 Manuscript accepted for publication in PLOS Genetics (2018)

"A fast and agnostic method for bacterial genome-wide association studies: bridging the gap between k-mers and genetic events"

Magali Jaillard☯, Leandro Lima☯, Maud Tournoud, Pierre Mahé, Alex van Belkum, Vincent Lacroix ,Laurent Jacob

☯These authors contributed equally to this work.

Abstract

Motivation: Genome-wide association study (GWAS) methods applied to bacterial genomes have shown promising results for genetic marker discovery or detailed assessment of marker effect. Recently, alignment-free methods based on k-mer composition have proven their ability to explore the accessory genome. However, they lead to redundant descriptions and results which are sometimes hard to interpret.
Methods: Here we introduce DBGWAS, an extended k-mer-based GWAS method producing interpretable genetic variants associated with distinct phenotypes. Relying on compacted De Bruijn graphs (cDBG), our method gathers cDBG nodes, identified by the association model, into subgraphs defined from their neighbourhood in the initial cDBG.
Results: DBGWAS is fast, alignment-free and only requires a set of contigs and phenotypes. It produces enriched subgraphs representing local polymorphisms as well as mobile genetic elements (MGE) and offers a graphical framework to interpret GWAS results. We validated our method using antibiotic resistance phenotypes for three bacterial species. DBGWAS recovered known resistance determinants such as mutations in core genes in *Mycobacterium tuberculosis* and genes acquired by horizontal transfer in *Staphylococcus aureus* and *Pseudomonas aeruginosa* – along with their MGE context. It also enabled us to formulate new hypotheses involving genetic variants not yet described in the antibiotic resistance literature.
Conclusion: Our method is computationally efficient and is able to retrieve phenotype-associated genetic variants such as local polymorphisms and MGEs without relying on prior annotation or reference genomes. Experiments took one hour and a half on average, and produced a compact set of meaningful subgraphs, thereby facilitating the interpretation of the results.
Availability: Open-source tool available at `https://gitlab.com/leoisl/dbgwas`.

Author summary

Genome-wide association studies (GWAS) help explore the genetic bases of phenotype variation in a population. Our objective is to make GWAS amenable to bacterial genomes. These genomes can be too different to be aligned against a reference, even within a single species, making the description of their genetic variation challenging. We test the association between the phenotype and the presence in the genomes of DNA subsequences of length k – the so-called k-mers. These k-mers provide a versatile descriptor, allowing to capture genetic variants ranging from local polymorphisms to insertions of large mobile genetic elements. Unfortunately, they are also redundant and difficult to interpret. We

rely on the compacted De Bruijn graph (cDBG), which represents the overlaps between k-mers. A single cDBG is built across all genomes, automatically removing the redundancy among consecutive k-mers, and allowing for a visualisation of the genomic context of the significant ones. We provide a computationally efficient and user-friendly implementation, enabling non-bioinformaticians to carry out GWAS on thousands of isolates in a few hours. This approach was effective in catching the dynamics of mobile genetic elements in *Staphylococcus aureus* and *Pseudomonas aeruginosa* genomes, and retrieved known local polymorphisms in *Mycobacterium tuberculosis* genomes.

Introduction

The aim of Genome-Wide Association Studies (GWAS) is to identify associations between genetic variants and a phenotype observed in a population. They have recently emerged as an important tool in the study of bacteria, given the availability of large panels of bacterial genomes combined with phenotypic data [3, 43, 61, 64, 98, 117, 183].
GWAS rely on a representation of the genomic variation as numerical factors. The most common approaches are based on single nucleotide polymorphisms (SNPs), defined by aligning all genomes of the studied panel against a reference genome [3, 43, 64] or against a pangenome built from all the genes identified by annotating the genomes [156], and on gene presence/absence, using a pre-defined collection of genes [61, 98]. The use of a reference genome becomes unsuitable when working on bacterial species with a large accessory genome – the part of the genome which is not present in all strains. On the other hand, methods focusing on genes are unable to cover variants in noncoding regions, including those related to transcriptional and translational regulation [24, 222]. Moreover, some poorly studied species still lack a representative annotation [83].
To circumvent these issues and make bacterial genomes amenable to GWAS, recent studies have relied on k-mers: all nucleotide substrings of length k found in the genomes [61, 117, 183]. The presence of k-mers in genomes can account for diverse genetic events such as the acquisition of SNPs, (long) insertions/deletions and recombinations. Unlike SNP- or gene-based approaches, k-mer analyses do not require a reference genome or any assumption on the nature of the causal variants and can even be performed without assembling the genome sequences [115].
While k-mers can reflect any genomic variation in a panel, they do not themselves represent biological entities. Translating the result of a k-mer-based GWAS into meaningful genetic variants typically requires mapping a large and redundant set of short sequences [61, 117, 174, 183]. Recent studies have suggested reassembling the significantly associated k-mers to reduce redundancy and retrieve longer marker sequences [117, 174]. Nonetheless, k-mer representation often loses in interpretability what it gains in flexibility, and the best way to encode the genomic variation in bacterial GWAS is not yet clearly defined [167, 175].
Our approach, coined DBGWAS, for *De Bruijn Graph GWAS*, bridges the gap between, on the one hand, SNP- and gene-based representations lacking the right level of flexibility to cover complete genomic variation, and, on the other hand, k-mer-based representations which are flexible but not readily interpretable. We rely on De Bruijn graphs [52] (DBGs), which are widely used for *de novo* genome assembly [163, 223] and variant calling [93, 115]. These graphs connect overlapping k-mers (here DNA fragments), yielding a compact summary of all variations across a set of genomes. Fig 3.1 illustrates the construction of such a graph for a simple example, where the only variation among the aligned genomes is a point mutation. DBGs also accommodate more complex disparities including rearrangements and insertions/deletions (S3.1 Suppl).

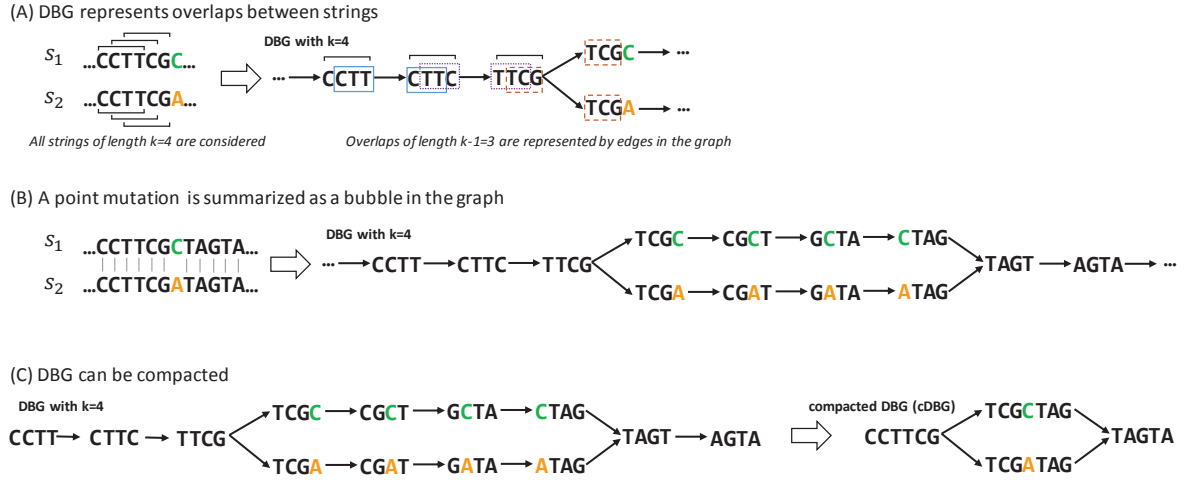


Figure 3.1: **Compacted DBG construction over a set of sequences differing by a single point mutation.** In this example two sequences s_1 and s_2 of length 12 differ by a single letter. (A) All k-mers ($k = 4$) present in these sequences are listed. A link is drawn between two k-mers when the $k - 1 = 3$ last nucleotides of the first k-mer equal the 3 first nucleotides of the second k-mer. (B) The bubble pattern represents the SNP C to A; each branch of the bubble represents an allele. (C) Linear paths of the graph are compacted; the compacted DBG of the example only contains four nodes (unitigs) and represents the same variation as the original DBG, which contained 13 nodes (k-mers).

DBGWAS relies on the ability of compacted DBGs (cDBGs) to eliminate local redundancy, reflect genomic variations, and characterise the genomic environment of a k-mer at the population level. More precisely, we build a single cDBG from all the genomes included in the association study (in practice, up to thousands). The graph nodes – called unitigs – represent, by construction, sequences of variable length and are at the right level of resolution for the set of genomes considered, taking into account adaptively the genomic variation. The unitigs are individually tested for association with the phenotype, while controlling for population structure. The unitigs found to be phenotype-associated are then localised in the cDBG. Subgraphs induced by their genomic environment are extracted. They often provide a direct interpretation in terms of genetic events which results from the integration of three types of information: 1) the *topology* of the subgraph, reflecting the nature of the genetic variant, 2) the *metadata* represented by node size and colour, allowing us to identify which unitigs in the subgraph are associated to a particular phenotype status, and 3) an optional sequence *annotation* helping to detect unitig mapping to – or near – a known gene.
We benchmarked our novel method using several antibiotic resistance phenotypes within three bacterial species of various degrees of genome plasticity: *Mycobacterium tuberculosis*, *Staphylococcus aureus* and *Pseudomonas aeruginosa*. The subgraphs built from significant unitigs described SNPs or insertions/deletions in both core and accessory regions, and were consistent with results obtained with a resistome-based association study. In addition, novel genotype-to-phenotype associations were also suggested.

Results

We developed DBGWAS, available at `https://gitlab.com/leoisl/dbgwas`, and validated it on panels for several bacterial species for which genome sequences and antibiotic resistance phenotypes were available. DBGWAS comprises three main steps: it first builds

a variant matrix, where each variant is a pattern of presence/absence of unitigs in each genome. Each variant is then tested for association with the phenotype using a linear mixed model, adjusting for the population structure. Finally, it uses the cDBG neighbourhood of significantly associated unitigs as a proxy for their genomic environment. DBGWAS outputs a set of such subgraphs ordered by $\min_q$, which is the smallest q-value observed over unitigs in each subgraph. The top subgraphs therefore represent the genomic environment of the unitigs most significantly associated with the tested phenotype. Fig 3.2 summarises the main steps of the process. A detailed description of the pipeline is presented in the Methods section.

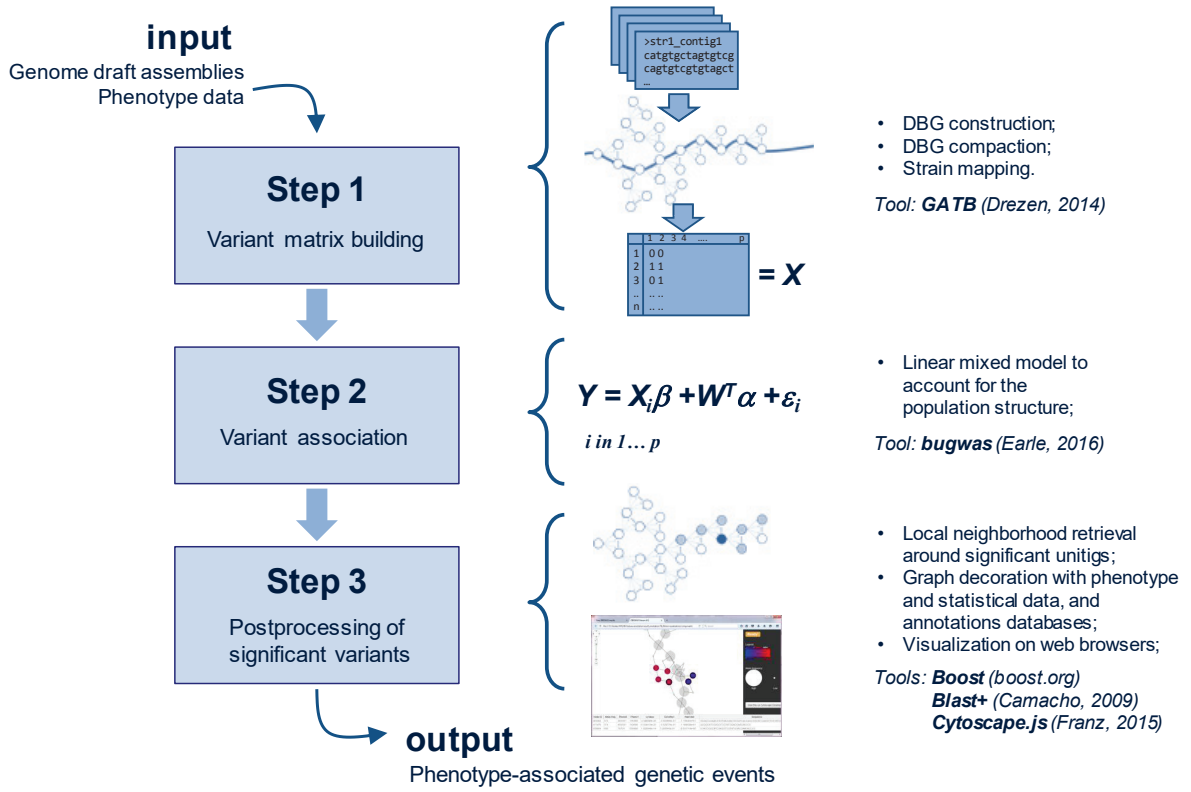


Figure 3.2: **DBGWAS pipeline.** DBGWAS takes as input draft assemblies and phenotype data for a panel of bacterial strains. A variant matrix X is built in *step* 1 using cDBG nodes (called unitigs). Variants are tested in *step* 2 using a linear mixed model taking into account the population structure. Significant variants are post-processed in *step* 3 to provide an interactive interface assisting their interpretation.

Here we rely on a few experiments to illustrate how the subgraphs output by DBGWAS can be read as genetic events. We then benchmark DBGWAS against two other k-mer-based approaches and one resistome-based approach. DBGWAS recovers known variants, while suggesting novel candidates out of the range of the resistome-based approach. We also find it to be more computationally efficient and to provide more interpretable outputs than the other k-mer-based methods.

A synthetic description of the discussed subgraphs is provided in Table 3.1, while a description of the top subgraphs obtained for all tested antibiotics is provided in S3.2 Suppl, S3.3 Suppl, and S3.4 Suppl. The subgraphs themselves are available online[1].

[1]http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/experiments/#DBGWAS_all_results

Table 3.1: **Resistance determinants identified by DBGWAS for *S. aureus* (SA), *M. tuberculosis* (TB) and *P. aeruginosa* (PA) panels.**

Panel	Phenotype	Rank	Sign. unitigs	min_q	Est. effect	Annotation	Type	Knowledge on markers
SA	Methicillin	1	71/565	7.68×10^{-188}	0.949	*mecA* + 7000 bp of SCC*mec*	MGE	Pos
		2	99/735	3.39×10^{-72}	0.865	6000 bp of SCC*mec*	MGE	$r^2 = 0.96$
		3	11/190	2.14×10^{-61}	0.813	2000 bp of SCC*mec*	MGE	$r^2 = 0.94$
		4	13/117	2.29×10^{-37}	0.957	1500 bp of SCC*mec*	MGE	$r^2 = 0.93$
	Ciprofloxacin	1	7/57	8.67×10^{-104}	-0.893	*parC* QRDR	LPG	Pos
		2	7/31	2.21×10^{-76}	0.955	*gyrA* QRDR	LPG	Pos
	Erythromycin	1	110/510	2.69×10^{-100}	0.823	*ermC* + circular plasmid	MGE	Pos
	Fusidic acid	1	7/50	2.75×10^{-136}	-0.910	*fusA*	LPG	Pos
		2	214/882	7.94×10^{-49}	0.924	*fusC* + SCC*fusC* cassette	MGE	Pos
		3	22/260	5.35×10^{-43}	0.924	1,500 bp of SCCfusC	MGE	$r^2 = 0.98$
		3	1/72	5.35×10^{-43}	0.924	200 bp of SCC*fusC*	MGE	$r^2 = 0.98$
		5	5/64	2.02×10^{-22}	-0.888	*purN*	LPG	$r^2 = 2 \times 10^{-3}$
	Trimethoprim	1	7/54	8.38×10^{-24}	0.969	*folA*	LPG	Pos
		2	3/41	9.30×10^{-18}	-0.966	btw. hyp. prot. & VOC prot.	LPN	$r^2 = 0.19$
		3	11/70	9.30×10^{-18}	-0.966	*ybaK*	LPG	$r^2 = 0.44$
		4	2/30	6.82×10^{-10}	-0.632	*mqo1*	LPG	$r^2 = 0.29$
	Gentamicin	1	173/1193	1.30×10^{-205}	0.873	*aac(6')* gene within a plasmid	MGE	Pos
		2	127/367	9.02×10^{-75}	0.751	seq. of plasmid carrying *aac(6')*	MGE	$r^2 = 0.38$
		3	2/23	9.01×10^{-53}	0.634	seq. of plasmid carrying *aac(6')*	MGE	$r^2 = 0.40$
		4	1/29	1.04×10^{-40}	0.579	seq. of plasmid carrying *aac(6')*	MGE	$r^2 = 0.48$
		5	2/56	1.49×10^{-33}	-0.831	*odhB*	LPG	$r^2 = 8 \times 10^{-5}$
TB	Rifampicin	1	36/115	4.84×10^{-70}	-0.577	*rpoB* RRDR	LPG	Pos
		2	6/37	4.35×10^{-20}	-0.355	*katG*	LPG	CR
		3	5/41	4.02×10^{-8}	-0.224	*embB* M306V	LPG	Pos
	Streptomycin	1	5/30	3.70×10^{-31}	0.544	*rpsL* (30S ribos.protein S12)	LPG	Pos
		2	6/37	1.06×10^{-28}	-0.428	*katG*	LPG	CR
		3	25/113	2.87×10^{-16}	-0.339	*rpoB* RRDR	LPG	CR
		4	6/45	1.40×10^{-9}	-0.271	*embB* M306V	LPG	CR
		5	8/31	2.86×10^{-9}	-0.535	*rrs*, 16S rRNA C517T	LPG	Pos
		6	13/69	9.18×10^{-5}	-0.216	*gyrA* QRDR	LPG	CR
		7	2/20	1.20×10^{-3}	0.739	*espG1*	LPG	$r^2 = 3 \times 10^{-3}$
	Ofloxacin	1	31/85	9.66×10^{-144}	-0.888	*gyrA* QRDR	LPG	Pos
		2	9/68	1.59×10^{-4}	0.507	*ubiA* (Rv3806c)	LPG	CR
		3	3/32	3.86×10^{-2}	-0.746	Rv3909	LPG	$r^2 = 9 \times 10^{-3}$
	Ethionamide	1	9/39	7.86×10^{-11}	-0.462	*fabG1* promoter	LPN	Pos
		2	15/47	5.16×10^{-10}	-0.406	*gyrA* QRDR	LPG	CR
		3	4/26	5.55×10^{-4}	0.319	*rrs*, 16S rRNA A1401G	LPG	CR
	XDR	1	6/68	3.66×10^{-39}	0.905	*rpoB* I1187T (out. RRDR)	LPG	Ukn
		1	3/27	3.66×10^{-39}	0.905	Rv2000	LPG	$r^2 = 1$
		3	3/24	9.58×10^{-36}	0.883	*espA* promoter	LPN	$r^2 = 0.98$
PA	Amikacin	1	4/83	5.86×10^{-9}	0.621	SNP in *aac(6')*	LPG	Pos
		2	3/82	1.37×10^{-6}	0.662	DEAD/DEAH box helicase	LPG	$r^2 = 0.55$
		3	38/315	2.21×10^{-6}	0.523	plasmid mapping on pHS87b	MGE	$r^2 = 0.17$
	Levofloxacin	1	5/27	7.21×10^{-29}	-0.884	*gyrA* QRDR	LPG	Pos
		2	5/29	5.68×10^{-6}	-0.737	*parC* QRDR	LPG	Pos
		3	5/38	1.87×10^{-2}	0.688	Histidine kinase/response regulator	LPG	$r^2 = 0.17$

For each antibiotic, we report subgraphs with their rank, number of significant unitigs over all unitigs in the subgraph (Sign. unitigs), q-value of the unitig with the lowest q-value ($\min_q$), the corresponding estimated effect ($\hat{\beta}$ coefficient of the linear mixed model) and annotation of the subgraph. The type of event represented by the subgraph is colour-coded as: yellow for MGE, light blue for local polymorphism in gene (LPG), and dark blue for local polymorphism in noncoding region (LPN). Known resistance markers are indicated in dark green (Pos), determinants whose presence was described to be caused by co-resistance in orange (CR), unknown variants arriving at the first rank in grey (Ukn). For other subgraphs, an r^2 value relative to the first subgraph is provided as an estimation of linkage disequilibrium with the first subgraph. It was computed between the most significant patterns of the first and the considered subgraphs.

Coloured bubbles highlight local polymorphism in core genes, accessory genes and noncoding regions

For *P. aeruginosa* levofloxacin resistance, the subgraph obtained with the lowest $\min_q$ highlighted a polymorphic region in a core gene (Fig 3.3A). Indeed, it showed a linear structure containing a complex bubble, with a fork separating susceptible (blue) and resistant (red) strains. The annotation revealed that all unitigs in this subgraph mapped to the quinolone resistance-determining region (QRDR) of the *gyrA* gene. *gyrA* codes for a subunit of the DNA gyrase targeted by quinolone antibiotics such as levofloxacin and its alteration is therefore a prevalent and efficient mechanism of resistance [90, 130]. In all our experiments related to quinolone resistance, DBGWAS identified QRDR mutations in either *gyrA* or *parC*, which codes for another well-known quinolone target: *P. aeruginosa* levofloxacin (first subgraph, *gyrA*: $\min_q = 7.21 \times 10^{-29}$ and second, *parC*: 5.68×10^{-06}), *S. aureus* ciprofloxacin (first, *parC*: $\min_q = 8.67 \times 10^{-104}$ and second, *gyrA*: 2.21×10^{-76}), and ofloxacin resistance in *M. tuberculosis*, whose genome does not contain the *parC* gene [165] (first, *gyrA*: $\min_q = 9.66 \times 10^{-144}$).
For *P. aeruginosa* amikacin resistance, the top subgraph ($\min_q = 5.86 \times 10^{-9}$) highlighted a SNP in an accessory gene (Fig 3.3B). As in Fig 3.3A, it contained a fork separating a blue and a red node. However, other remaining nodes were not grey: they represented an accessory sequence because they were not present in all the strains. Most of these nodes were pale-red, showing that the accessory sequence was more frequent in resistant samples. The annotation revealed that this subgraph corresponded to *aac(6')*, a gene coding for an aminoglycoside 6-acetyltransferase, an enzyme capable of inactivating aminoglycosides, such as amikacin, by acetylation [112]. Most unitigs in this gene had a low association with resistance, except for the ones describing this particular SNP. Mapping the sequence of these unitigs on the UniProt database [197] revealed an amino-acid change at L83S, right in the enzyme binding site. This SNP was previously shown to be responsible for substrate specificity alteration in a strain of *Pseudomonas fluorescens* [113]. It appears to increase the amikacin acetylation ability of *aac(6')*, making its association to amikacin resistance more significant than the gene presence itself.
Finally, for *M. tuberculosis* ethionamide resistance, the top subgraph ($\min_q = 7.86 \times 10^{-11}$, Fig 3.3C) represented a polymorphic region in a core gene promoter. The subgraph was mostly grey and linear with a localised blue and red fork. The most reliable annotation for this subgraph was *fabG1* (also known as *mabA*), a core gene previously shown to be involved in ethionamide and isoniazid resistance [65, 116]. None of the significantly associated unitigs mapped to the *fabG1* gene, but their close neighbours did (highlighted in Fig 3.3C by black circles), suggesting that the detected variant was located in the promoter region of the gene. This was confirmed by mapping the significant unitig sequences using the Tuberculosis Mutation database of the *mubii* resource [66].

Long single-coloured paths denote mobile genetic element insertions

For *S. aureus* resistance to methicillin, the top subgraph ($\min_q = 7.68 \times 10^{-188}$), shown in Fig 3.3D, revealed a gene cassette insertion. It contained a long path of red nodes, and a branching region including another red node path. The first path mapped to the *mecA* gene, extensively described in this context and known to be carried by the Staphylococcal Cassette Chromosome *mec* (SCC*mec*) [79, 94, 130]. The other part of the subgraph represented a >5,000 bp fragment of the cassette. It was less linear because it summarised several types of the cassette differing by their structure and gene content [94]. The next subgraphs represented other regions of the same cassette. Interestingly, retaining a greater number of unitigs to build the subgraphs leads to merging these individual subgraphs,

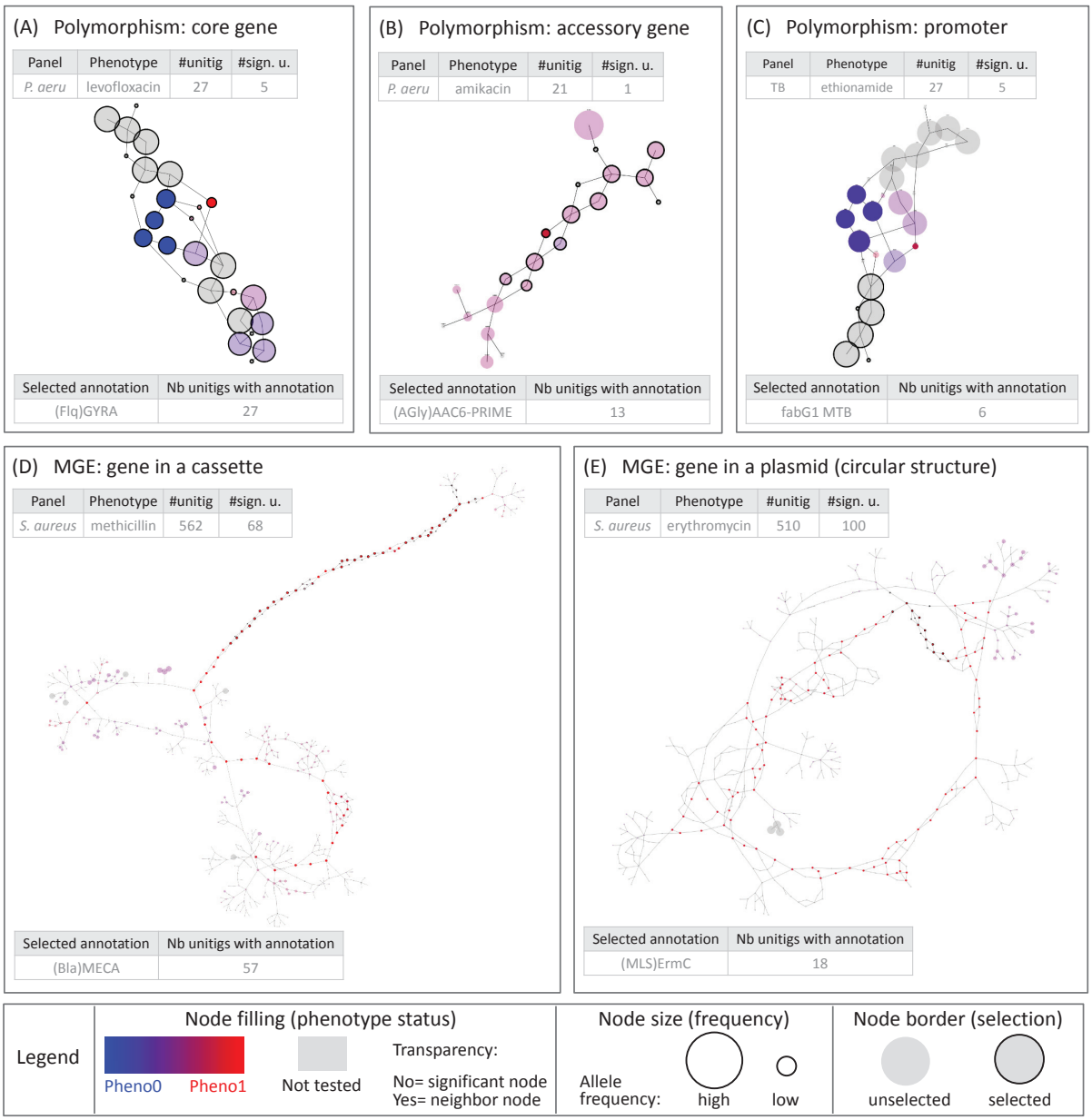


Figure 3.3: **Different types of genetic events identified by DBGWAS.** Each subgraph represents a distinct genetic event. Colours are continuously interpolated between blue for susceptible unitigs and red for resistant ones. Untested unitigs, present in $> 99\%$ or $< 1\%$ of the strains, are shown in grey. Nodes found to be not significative are shown with a transparency degree. The node size relates to its allele frequency: the larger the node, the higher the allele frequency. Circled black nodes map to annotated genes. The two tables in each panel provide information on the sugraph nodes. As an example, the subgraph in panel (A) is composed of 27 unitigs, 5 of which were significantly associated with resistance. All unitigs of this subgraph mapped to the *gyrA* gene. The subgraphs presented in the four other panels correspond to the top subgraphs (with lowest $\min_q$) obtained for different panels/phenotypes. All subgraphs are snapshots taken from DBGWAS interactive visualisation and are available online.

representing related genomic regions, into a single one. This can be done by increasing the Significant Features Filter (SFF) parameter value, which defines the unitigs used to build the subgraphs. By default, the unitigs corresponding to the 100 lowest q-values are retained ($SFF = 100$). Increasing the SFF value to 150 (150th q-value $= 1.60 \times 10^{-27}$) allowed us to reconstruct the entire SCC*mec* cassette, as shown in S3.6 Suppl.

For *S. aureus* erythromycin resistance, a unique subgraph was generated ($\min_q = 2.69 \times 10^{-100}$). As shown in Fig 3.3E, the subgraph described the circular structure of a 2,500 bp-long plasmid known to carry the causal *ermC* gene together with a replication and maintenance protein in strong linkage disequilibrium with *ermC* [79, 206].
For *P. aeruginosa* amikacin resistance, the third subgraph ($\min_q = 2.21 \times 10^{-6}$) represented a 10,000 bp plasmid acquisition. Using the NCBI nucleotide database [16], most of the unitigs in this subgraph mapped to the predicted prophage regions of an integrative and conjugative plasmid, whose structure corresponds to a plasmid, pHS87b, recently described in the amikacin resistant *P. aeruginosa* HS87 strain [20]. S3.7 Suppl and S3.8 Suppl provide more examples of MGEs recovered by DBGWAS, and the Interpretation of significant unitigs (step 3) subsection of the Methods section discusses SFF default value and tuning.

DBGWAS reports expected variants without prior knowledge

Although resistance determinants are not perfectly or exhaustively known for all species, some resistance mechanisms are well described. This is the case of *gyrA* and *parC* alteration in fluoroquinolone resistance in *P. aeruginosa* [90], and of the alteration of two streptomycin targets: the ribosomal protein S12 (coded by *rpsL*) and the 16S rRNA (coded by *rrs*) in *M. tuberculosis* [158]. Here we verify the ability of bacterial GWAS methods to recover these known mechanisms. We compared DBGWAS results to those obtained by applying the same association model to a collection of known resistance genes and SNPs [51, 98] (see the Resistome-based association studies subsection of the Methods section), and to two other recent k-mer-based methods: pyseer [117, 118], and HAWK [174].
For *P. aeruginosa* levofloxacin resistance (Table 3.2), both DBGWAS and pyseer identified the two expected known causal determinants reported by the prior resistome-based study: *gyrA* and *parC*, while HAWK only reported *gyrA*. pyseer reported 224 k-mers, all mapping to *gyrA* and *parC*, while the other methods reported less than 10 features (subgraphs or reassembled k-mers), among which were several unknown, potentially new candidate markers.

Table 3.2: **Resistance determinants found by the four methods for *P. aeruginosa* levofloxacin resistance.**

Legend	resistome-based	DBGWAS	pyseer	HAWK
Time (mem)	37m (7.2 GB)	21m (3.2 GB)	24h22m (14.5 GB)	39m (4.2 GB)
Nb reported	2 variants	5 subgraphs	224 k-mers	8 reassembled k-mers
Known positive	*gyrA* (2.11×10^{-22}) *parC* (1.83×10^{-5})	*gyrA* (7.21×10^{-29}) *parC* (5.68×10^{-6})	*gyrA* (1.97×10^{-17}) *parC* (5.68×10^{-9})	*gyrA* (2.82×10^{-14})
Unknown		HK/RR (1.87×10^{-2}) tnp *topA*		tnp (1.66×10^{-14}) NC near tnp

This table presents the annotation of the features identified by the tested methods with default parameters. The total number of reported features, as well as the execution time and memory load (in Gigabytes) are given in the header. For k-mer-based methods, annotations were retrieved by mapping unitig/k-mer sequences to the resistance and Uniprot databases (see Interpretation of significant unitigs (step 3) subsection of the Methods section), and completed when needed by Blast on NCBI Nucleotide database. Green cells correspond to resistance determinants already described in the literature. Grey cells represent unknown determinants. Within each category, annotations are ordered by increasing minimum p/q-values. p/q-values are reported only for the most significant annotations. For each method, the annotation with the lowest p/q-values is underlined. 'NC' means noncoding region and 'tnp' transposase.

For *M. tuberculosis* streptomycin resistance (Table 3.3), the four methods reported the two expected known causal determinants *rpsL* and *rrs*. However, while the resistome-based study and DBGWAS methods ranked the causal *rpsL* determinant first, pyseer and HAWK reported their lowest p/q-values for the false positive *katG* determinant. *katG* and other false positives caused by co-resistance were among the top-ranked features for all methods and this is a well described phenomenon in *M. tuberculosis* species [158, 195].

Table 3.3: **Resistance determinants found by the four methods for *M. tuberculosis* streptomycin resistance.**

Legend	resistome-based	DBGWAS	pyseer	HAWK
Time (mem)	1h31m (2.1 GB)	42m (4.3 GB)	14h14m (102.4 GB)	3h01m (3.7 GB)
Nb reported	28 variants	24 subgraphs	85,011 k-mers	2,038 reassembled k-mers
Known positive	*rpsL* (1.96×10^{-33}) *rrs* (5.40×10^{-8})	*rpsL* (3.70×10^{-31}) *rrs* (2.86×10^{-9})	*rpsL* (4.85×10^{-55}) *rrs* (1.63×10^{-14})	*rpsL* (5.72×10^{-47}) *rrs* (3.45×10^{-20})
Determinant described for other antibiotics	*katG* (2.61×10^{-30}) *rpoB* *gidB* *gyrA* *embB* *fabG1* promoter *pncA* *rpoC* *inhA*	*katG* (1.06×10^{-28}) *rpoB* *embB* *gyrA* *gidB* *rpoC* *fabG1* promoter ***ubiA***	*katG* (2.12×10^{-71}) *rpoB* *embB* ***ubiA*** *pncA* *fabG1* promoter *gyrA* *gidB* ***ethA*** ***embA*** ***embC***	*katG* (1.44×10^{-57}) *embB* ***kasA*** ***embC*** *gyrA* ***iniA*** ***embA*** ***embR*** *gidB* ***tsnR*** *rpoB* *pncA* ***ethA***
Unknown (top list)		*espG1* (1.20×10^{-3}) *rpsN* NC near tnp/PPE *rnj* Rv2672 *espA* promoter Rv2456c promoter *whiB6* ...	NC near tnp/PE (1.13×10^{-19}) Rv0270 Rv2665 Rv2743c Rv2522c NC near tnp/PPE *guaA* *kdpD* ...	NC near tnp/PPE (2.93×10^{-57}) tnp Rv2825c/Rv2828c 13E12 repeat family protein PPE CRISPR repeats, down *Cas* genes *mmpL14* *esxM* ...

This table presents the annotation of the features identified by the tested methods with default parameters. The total number of reported features, as well as the execution time and memory load (in Gigabytes) are given in the header. For k-mer-based methods, annotations were retrieved by mapping unitig/k-mer sequences to the resistance and Uniprot databases (see Interpretation of significant unitigs (step 3) subsection of the Methods section), and completed when needed by Blast on NCBI Nucleotide database. Green cells correspond to resistance determinants already described in the literature, orange cells to resistance determinants described for association with other antibiotics. The annotations not found by the resistome-based strategy are written in bold. Grey cells represent unknown determinants. Within each category, annotations are ordered by increasing minimum p/q-values. p/q-values are reported only for the most significant annotations. For each method, the annotation with the lowest p/q-values is underlined. 'NC' means noncoding region, 'tnp' transposase, 'PE' stands for PE-family protein and 'PPE' for PPE-family protein.

Additional results for all antibiotics can be found in S3.10 Suppl and S3.11 Suppl for resistome-based association studies, and in S3.2 Suppl and S3.4 Suppl for DBGWAS.

DBGWAS provides novel hypotheses

In addition to resistance markers, all three k-mer-based approaches reported several unknown variants, not described in the context of resistance. Among them, in the context of streptomycin resistance, a noncoding region between a transposase and a PPE-family pro-

tein was reported by the three methods but, as expected, not by the resistome-based approach, as only resistance genes were included in this analysis. More generally, knowledge-based approaches such as SNP-, gene- or resistome-based GWAS can be limited in the context of new marker discovery, since any causal variant absent from the chosen reference would remain untested. Besides being time-consuming, preparing such a list of genetic variants can be problematic for bacterial species without extensive annotation or reference availability. Here we describe associations identified by DBGWAS and which were never described in the antibiotic resistance literature.

In our *P. aeruginosa* panel, the second subgraph obtained for amikacin resistance ($\min_q = 1.37 \times 10^{-6}$) gathered unitigs mapping to the 3' region of a DEAD/DEAH box helicase, known to be involved in stress tolerance in *P. aeruginosa* [91]. The unitig with the lowest q-value was present in 13 of 47 resistant strains and in only 1 of 233 susceptible strains and represented a C-C haplotype summarising two mutated positions: 2097 and 2103. This annotation was not an artefact of the population structure, properly taken into account by the linear mixed model. Indeed the 13 resistant strains corresponded to distinct clones belonging to two phylogroups, one of them containing the susceptible strain. In *P. aeruginosa* levofloxacin resistance, the third subgraph ($\min_q = 1.87 \times 10^{-2}$) represented a L650M amino-acid change in a hybrid sensor histidine kinase/response regulator. Such two-components regulatory systems play important roles in the adaptation of organisms to their environment, for instance in the regulation of biofilm formation in *P. aeruginosa* [5], and as such may play a role in antibiotic resistance.

In *S. aureus*, polymorphisms within genes not known to be related to resistance were identified for several antibiotics: *purN* ($\min_q = 2.02 \times 10^{-22}$) for fusidic acid, *odhB* ($\min_q = 1.49 \times 10^{-33}$) for gentamicin, *ybaK* and *mqo1* ($\min_q = 9.30 \times 10^{-18}$, resp. 6.82×10^{-10}) for trimethoprim. None of these genes have been associated with antibiotic resistance before, to the best of our knowledge.

In *M. tuberculosis*, polymorphisms in two genes encoding proteins involved in *cell wall and cell processes*, *espG1* and *espA*, were found associated with streptomycin (seventh subgraph, $\min_q = 9.43 \times 10^{-4}$) and XDR phenotype (third subgraph, $\min_q = 9.58 \times 10^{-36}$), respectively. Again, these genes have never been reported in association with antibiotic resistance before.

Although experimental validation would be required to tell whether these hypotheses are false positive (e.g., in linkage with causal variants) or actual resistance mechanisms not yet documented, DBGWAS is a valuable tool to screen for novel candidate markers. Moreover it provides a first level of variant description (SNPs in gene or promoter, MGE, etc) which can directly drive the biological validation.

DBGWAS facilitates the interpretation of k-mer-based GWAS

Other k-mer-based approaches are as agnostic as DBGWAS and were also able to provide novel hypotheses, but interpreting their output can prove more challenging than a SNP/gene-based GWAS. In the *M. tuberculosis* streptomycin resistance experiment for example, they reported several thousands of features, while DBGWAS reported only 24 annotated subgraphs without missing any expected determinant (see Table 3.3). The thousands of k-mers generated by HAWK and pyseer are of course also amenable to interpretation: to build our Table 3.3, we mapped these k-mers to references and extracted annotated variants which showed at least one hit. However, doing so required additional efforts and a working knowledge of the most appropriate annotated references. In addition, k-mers which do not map to the chosen reference cannot be interpreted. By contrast, DBGWAS always returns a subgraph containing these k-mers. Even when no annotation

exists, the topology and colours of the subgraphs may hint towards the nature of the causal variant.

In addition to providing context for significant k-mers and guiding their interpretation as SNPs or MGEs, DBGWAS clustering of close variants into a subgraph can describe hypervariable regions as single entities, and highlight highly associated haplotypes. As an example, the top subgraph for rifampicin resistance ($\min_q = 4.84 \times 10^{-70}$) contained 36 significant unitigs, distinguishing between susceptible (blue) and resistant (red) strains. Instead of a single point mutation, this subgraph represented a polymorphic region known as the rifampicin resistance-determining region (RRDR) of the *rpoB* gene. The unitig with the lowest q-value covered several mutant positions, defining a particular haplotype strongly associated with rifampicin resistance. Where DBGWAS reported in this case only one subgraph, pyseer, for instance, reported 470 k-mers with the *rpoB* annotation, and the resistome-based association study reported in this case 4 distinct SNPs in *rpoB* (S3.10 Suppl). In another user-submitted example, DBGWAS identified mosaic alleles of three *pbp* genes involved in beta-lactam resistance of *Streptococcus pneumoniae*. Like in the RRDR example, it returned five subgraphs corresponding to the three genes – three subgraphs were annotated *pbp2x* and represented three distinct polymorphic regions of the gene. Each subgraph summarised the polymorphism of the gene, as opposed to one separate feature for each SNP.

Admittedly, some subgraphs output by DBGWAS are not readily interpretable: they are neither coloured bubbles highlighting SNPs, nor long single-coloured paths denoting MGE insertions. This was the case of several subgraphs produced for *P. aeruginosa* amikacin resistance, and presented in S3.9 Suppl. Genetic variants inserted in variable regions, for example, lead to subgraphs with a high average degree, or to very large subgraphs. The fourth subgraph for instance ($\min_q = 2.21 \times 10^{-6}$) contains a path of three red (positively-associated) nodes lying in a noncoding region between variable accessory genes. Consequently, their neighbour unitigs branch to various other unitigs, making the structure complex and hard to interpret. Complex subgraphs also arise when several associated variants have overlapping neighbourhoods (as defined in the Graph neighbourhoods subsection in the Methods section, and tuned with the *nh* parameter) in at least one strain. This is the case for the subgraph with the smallest $\min_q$ which aggregates *aac*(6′) acetyltransferase and the CML efflux pump.

The interpretation of such subgraphs is not straightforward. We often found it helpful to tune the *nh* and SFF parameters to break large subgraphs into a set of smaller ones, as discussed in the Methods section. For the *aac*(6′) subgraph, where nearby variants are aggregated into a large subgraph, reducing the SFF value to 15 provided a much smaller and easier-to-interpret subgraph focusing on the *aac*(6′) mutation (Fig 3.3B). Otherwise, we recommend to focus on the topology of the most significant unitigs and their close neighbours.

DBGWAS is fast, memory-efficient, and scales to very large panels

To assess the scalability of DBGWAS to large datasets, we retrieved 5,000 genomes from *M. tuberculosis*, 9,000 genomes from *S. aureus* and 2,500 genomes from *P. aeruginosa*, as described in the Large panels subsection of the Methods section. We present in S3.12 Suppl the runtime and memory usage performances for these panels. All 180 runs took less than 5 days and 250 GB of RAM on 8 cores. Both the computational time and memory usage increase log-linearly with the panel size. Moreover, at equal panel size, DBGWAS performance also depends on the genome complexity, requiring less computational resource for more clonal genomes such as *M. tuberculosis*.

We also compared the computational performance of DBGWAS with pyseer and HAWK. The benchmark was performed on 13 datasets, including one large dataset of 2,500 genomes for each of the 3 species (see the Datasets subsection in the Methods section for details). Detailed results are presented in S3.13 Suppl. DBGWAS was the fastest tool in 11 out of 13 experiments, always taking less than 2 hours. HAWK ran in less than 10 hours in 12 out of 13 experiments, and was a little faster than DBGWAS on two of the large-scale datasets. pyseer took from 13 to 53 hours on 9 experiments, and failed on the 4 others: one exceeded the disk space limit of 1TB, three exceeded the runtime limit of five days. HAWK was more parsimonious in memory usage than DBGWAS on the large scale panels. This can be explained by the fact that the 0.8.3-beta version of HAWK which we are using does not take into account the population structure, and as such does not have to compute an $n \times n$ covariance matrix, providing it a large gain in memory usage – and, to a lesser extent, runtime – for large panels. On the other hand, disregarding the population structure could also lead to spurious discoveries. HAWK v0.9.8-beta offers an adjustment but failed to recover the known true positives, which is why we chose to present the results of the 0.8.3-beta version. DBGWAS and HAWK typically used one order of magnitude less memory than pyseer. The most memory-consuming step for pyseer was the k-mer counting step relying on fsm-lite.

Discussion

In this article we introduce an efficient method for bacterial GWAS. Our method is agnostic: it considers all regions of the genomes and is able to identify potentially new causal variants as different as SNPs in noncoding regions and MGE insertions/deletions. It performs as well as the current SNP- and gene-based gold standard approaches for retrieving known determinants, from genome pre-assemblies and without relying on annotations or reference genomes.

DBGWAS exploits the genetic environment of the significant k-mers through their neighbourhood in the cDBG, providing a valuable interpretation framework. Because it uses only contig sequences as input, it allows GWAS on bacterial species for which the genomes are still poorly annotated or lack a suitable reference genome. DBGWAS makes bacterial GWAS possible in two hours using a single-core computer (see S3.14 Suppl), outperforming other state-of-the-art k-mer-based approaches.

Underlying our method, graph-based genome sequence representations such as DBGs, extend the notion of the reference genome to cases where a single sequence stops being an appropriate approximation [136, 159]. As demonstrated in this paper, they pave the way to GWAS on highly plastic bacterial genomes and could also be useful for microbiomes [10] or human tumours [174].

DBGWAS currently relies on the Benjamini-Hochberg procedure to control the FDR and offers no advance exploiting the dependence among presence/absence patterns. An important improvement would be to control the false discovery rate at the subgraph level instead of the unitig level. DBGWAS could be extended to different statistical tasks by adapting its underlying association model, to allow for continuous phenotypes or identify epistatic effects, for instance. The interpretability of the extracted subgraphs could also be improved by training a machine learning model to predict which types of event they represent [96]. This automated labelling could guide users in their interpretation and allow them to search for specific events, such as SNPs in core genes or rearrangements.

Several recent studies describe *in silico* models for defining a genomic antibiogram and hopes are high that such technologies will complement the classic phenotypic methods [59]. Several studies have already demonstrated that in some cases, genomic antibiograms can

be at least as good as phenotypic ones [29, 79, 105, 148]. Contrary to our approach, these studies require extensive resistance marker databases. DBGWAS will surely contribute to the extension of such databases or to the development of agnostic genomic antibiograms.
In conclusion, we demonstrate for three medically important bacterial species that resistance markers can be detected rapidly with relative ease, using simple computer equipment. Our integrated software and visualisation tools offer an intuitive variant representation, hence will provide future users with an enhanced insight into genotype to phenotype correlations, in all domains of microbiology, beyond that of antibiotic resistance. This will include complex traits such as biofilm formation, epidemicity and virulence.

Methods

Encoding genomic variation with compacted DBGs

DBGs are directed graphs that efficiently represent all the information contained in a set of sequences. Nodes represent all the unique k-mers (genome sequence substrings of length k) extracted from the input sequences. Edges represent $(k-1)$-exact-overlaps between k-mers: an edge connects a node n_1 to a node n_2 if and only if the $(k-1)$-length-suffix of n_1 equals the $(k-1)$-length-prefix of n_2 (Fig 3.1A).
These graphs can be compacted into cDBGs by merging linear paths (sequences of nodes not linked to more than two other nodes) into a single node referred to as a *unitig* [34, 44, 221] (Fig 3.1C). Compaction yields a graph with locally optimal resolution: regions of the genome which are conserved across individuals are represented by long unitigs, while regions which are highly variable are fractioned into shorter unitigs (S3.1 Suppl).

Representing strains by their unitig content (step 1)

cDBG construction

We build a single DBG from all genomes given as input using the GATB C++ library [57]. We start from contigs rather than reads and, consequently, we do not need to filter out low abundance k-mers, allowing for the exploration of any variation present in the set of input genomes. We then compact the DBG using a graph traversal algorithm, which identifies all linear paths in the DBG – each forming a unitig in the cDBG. During this step, we also associate each k-mer index to its corresponding unitig index in the cDBG.
There is no general rule for choosing the ideal k-mer length as it depends on many factors, including the assembly quality, complexity of the input genomes, or presence of repeats. High values of k lead to haplotypes containing multiple SNPs instead of distinct single SNPs, if these SNPs are separated by less than k bases. As k increases, the k-mer-defined haplotypes also become more specific to a genome sub-population, leading to a loss of power to detect genotype to phenotype associations. Low values of k, on the other hand, produce highly connected sets of non-specific k-mers. In particular, any repeated region with at least k bases may create a cycle in the DBG (Fig 3.4). We use $k = 31$ by default, as it produced the best performance to retrieve known markers of *P. aeruginosa* resistance to amikacin and levofloxacin (Fig 3.5). We found DBGWAS results to be robust to small variations of k between 21 and 41. Similar graph structures were generated whatever the tested value of k for the clonal *M. tuberculosis* species (S3.15 Suppl). More variability was observed for *P. aeruginosa* resistance to amikacin, which involves more complex resistance mechanisms (S3.16 Suppl).

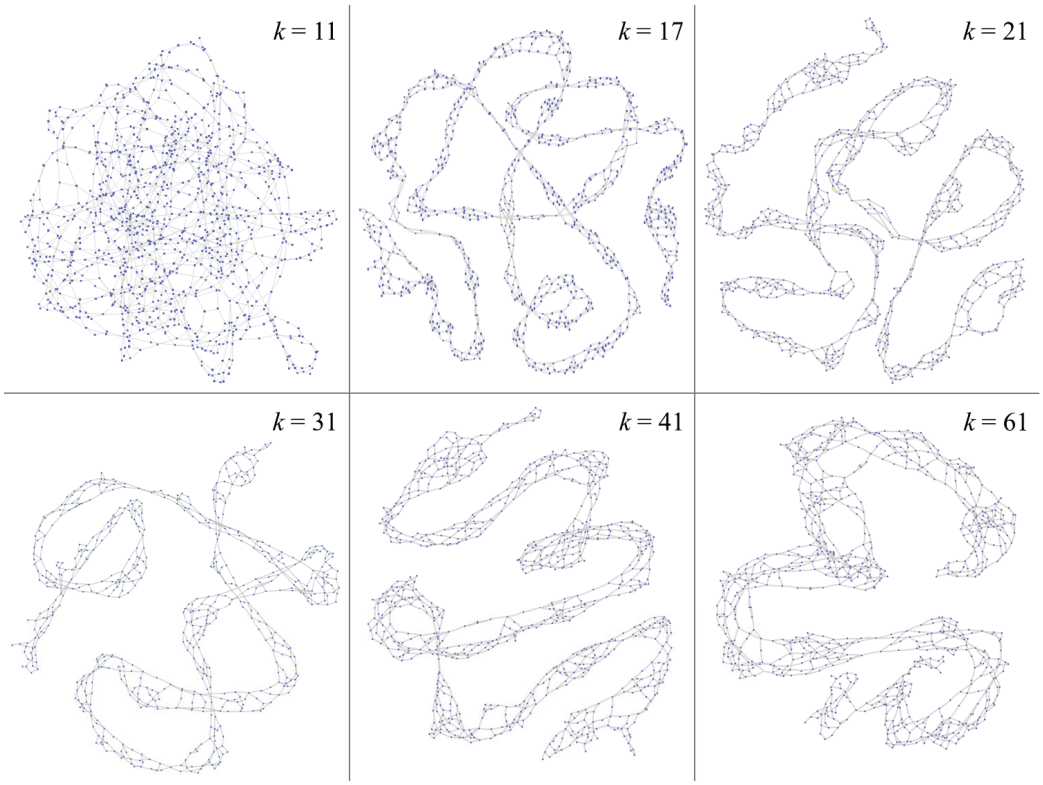


Figure 3.4: **Effect of k on the graph topology.** A cDBG was built from the *P. aeruginosa gyrA* gene sequences from several strains. When k is small, k-mers are highly repeated, which generate numerous loops. As k increases, k-mer sequences become more specific and the graph gets more linear. For large values of k, few k-mers are shared by all the strains, and the linear path thickens into parallel paths belonging to variable strain populations.

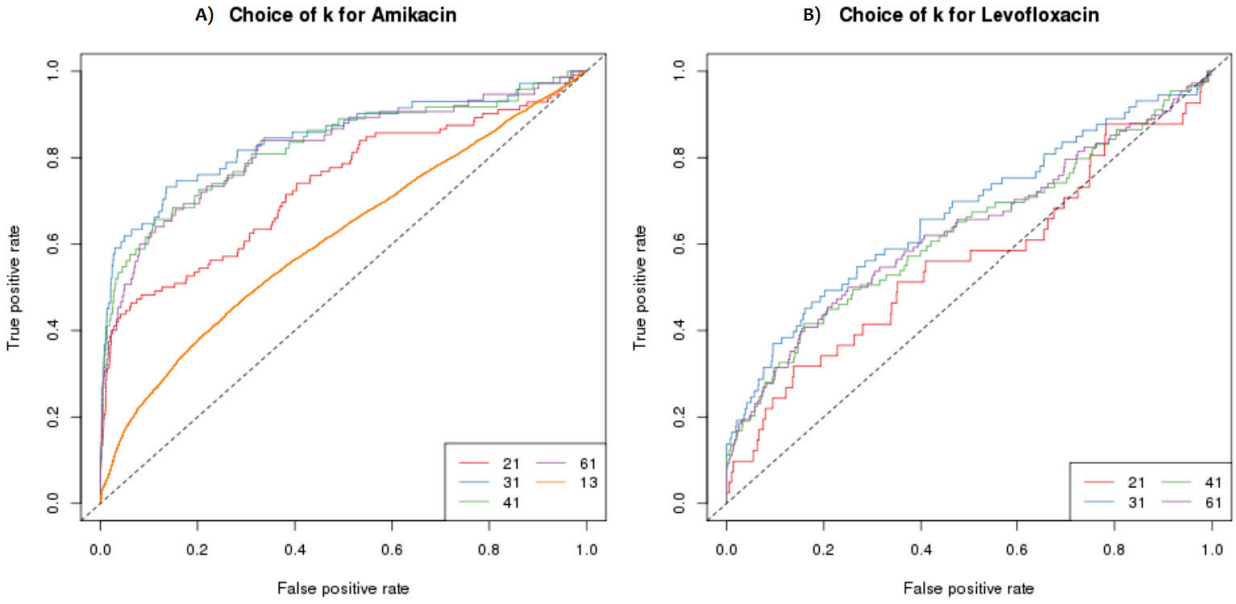


Figure 3.5: **Choice of k.** True positive *versus* false positive curves for several values of k for both amikacin and levofloxacin resistance phenotypes. True positives are unitigs mapping to genuine variants described in resistance databases for the studied drugs [98]. In both cases, the value of k leading to the best AUC is $k = 31$.

Unitig presence across genomes

Each genome is represented by a vector of presence/absence of each unitig in the cDBG. To do so, we query the unitig associated to each k-mer in a given genome. This procedure is efficient because it relies on constant time operations. Firstly, we use GATB's Minimal Perfect Hash Function (MPHF) [123] to retrieve the index of a given k-mer, and then we use the previously computed association between k-mer and unitig indices to know which unitigs the given genome contains. Since these two operations take constant time, producing this vector representation for a genome takes linear time on the size of the genome. It is important to note that the GATB's MPHF can be successfully applied here because we always use the same list of k-mers, *i.e.*, after building the DBG, the set of k-mers is fixed and not updated, and because we always query k-mers that are guaranteed to be in the DBG (since we do not filter out any k-mer).
The unitig description on all the input genomes is stored into a matrix U:

$$U_{i,j} = \begin{cases} 1, \text{ if the } j\text{-th unitig is present in the } i\text{-th input genome;} \\ 0, \text{ otherwise.} \end{cases}$$

We then transform the matrix U into Z, which represents the minor allele description, in terms of presence [61]: Z is identical to U except for columns with a mean larger than 0.5, which are complemented: $Z_j = 1 - U_j$ for these columns.
We then restrict Z to its set of unique columns. If several unitigs have the same minor allele presence pattern, then they will be represented by a single column. Keeping duplicates would lead to performing the same statistical test several times. Finally, we filter out columns whose average is below 0.01 – the user can specify this threshold using the `-maf` option. We denote the de-duplicated, filtered matrix of patterns by X.
Importantly, both k-mers and unitigs lead to the same set of distinct patterns across the genomes. Indeed, every unitig represents (at least) one k-mer, and conversely every k-mer is represented by one (single) unitig. When de-duplicated, the two representations therefore lead to the same set of patterns to be tested for association with the phenotype.

Testing unitigs for association with the phenotype (step 2)

Human GWAS literature extensively discusses how testing procedures can result in spurious associations if the effect of the population structure is not taken into account [11, 207, 226]. Population structures can be strong in bacteria because of their clonality [48, 61, 63, 117]. An additional performance analysis comparing several models for population structure, on both simulated and real data, showed that correcting for population structure using LMMs is often preferable to using a fixed effect correction or not correcting at all (Appendix S3.1: Evaluation of association models).
We thus rely on the bugwas method [61], which uses the linear mixed model (LMM) implemented in the GEMMA library [225], to test for association with phenotypes while correcting for the population structure. This method also offers the possibility to test for lineage effects, by calculating p-values for association between the columns of the matrix representing the population structure, and the phenotype [61]. DBGWAS optionally provides bugwas lineage effect plots when the user specifies a phylogenetic tree using the `-newick` option. An example of the generated figures is available online[2].
Formally, the LMM represents the distribution of the binarized phenotype Y_i, given the j-th minor allele pattern X_{ij} and the population structure represented by a set of factors $W \in \mathbb{R}^{n \leqslant p}$, by:

[2] http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/full_dataset_visualization

$$Y_i = X_{ij}\beta + W_i^T\alpha + \varepsilon_{ij}, \quad j = 1, \ldots, p. \tag{3.1}$$

β is the fixed effect of the tested candidate on the phenotype, $\alpha \sim \mathcal{N}(0, \sigma_a^2)$, $\sigma_a^2 > 0$ is the random effect of the population structure, and $\varepsilon_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$ are the residuals with variance $\sigma^2 > 0$. W is estimated from the Z matrix, which includes duplicate columns representing both core and accessory genome. More precisely, denoting $Z = U\Lambda V^\top$ the singular value decomposition of Z, we use $W = U\Lambda$.
We test $H_0 : \beta = 0$ *versus* $H_1 : \beta \neq 0$ in Eq 3.1 for each pattern using a likelihood ratio procedure producing p-values and maximum likelihood estimates $\hat{\beta}$. To tackle the situation of multiple testing caused by the high number of tested patterns, we compute q-values, which are the Benjamini-Hochberg transformed p-values controlling for false discovery rate (FDR) [14].

Interpretation of significant unitigs (step 3)

The LMM is used to identify de-duplicated minor allele presence patterns significantly associated with the phenotype at a chosen FDR level. While the testing step is done at the pattern level, the interpretation of the selected features is done at the unitig level. As a result of the de-duplication procedure, a given pattern may correspond to several distinct unitigs. To faithfully interpret the results, all the unitigs corresponding to the significant patterns are retrieved and are assigned the q-value of their pattern. We now show how the initial cDBG can be used in the interpretation step.

Significance threshold

The interpretation step focuses on the unitigs with the lowest q-values. These unitigs are indeed used to build the resulting annotated subgraphs. The unitig selection can be either based on the FDR (q-value thresold) or on a number of presence/absence patterns ordered by increasing q-values. Practically, this is done in DBGWAS using a Significant Features Filter (SFF). For a selection based on a FDR threshold, the SFF value is set between 0 and 1, while any integer value > 1 defines the number of patterns to consider. In our experiments, we choose not to apply a fixed FDR threshold, even though DBGWAS offers this option. Different datasets lead to different q-values, even by several orders of magnitude, and a single FDR threshold would lead to selecting a large number of unitigs generating more than 1,000 subgraphs on some of them (e.g. *S. aureus* ciprofloxacin) as shown in S3.17 Suppl. Instead, we retain the 100 patterns with lowest q-values. Although arbitrary, this choice is tractable for all datasets and provides satisfactory results in our experiments. It does not provide and explicit control of the FDR: only the q-value provides an estimation of the proportion of false discoveries incurred when considering patterns below this value. Checking the q-values of the selected unitigs is therefore essential to assess their significance. If the default SFF=100 is not satisfactory, it is also possible to re-run the third step only, with a more suitable SFF value.

Graph neighbourhoods

We define the neighbourhood of each significant unitig u (defined by the SFF) as the set of unitigs whose shortest path to u has at most $ne = 5$ edges. Users can modify the ne value using the `-nh` option. The objects returned by DBGWAS are the connected components of the graph induced by the neighbourhoods of all significant unitigs in the cDBG. As illustrated in Fig 3.6, nearby significant unitigs might belong to the same

connected component, so this process groups unitigs which are likely to be located closely in the genomes. We refer to the connected components as *subgraphs* in the Results section.

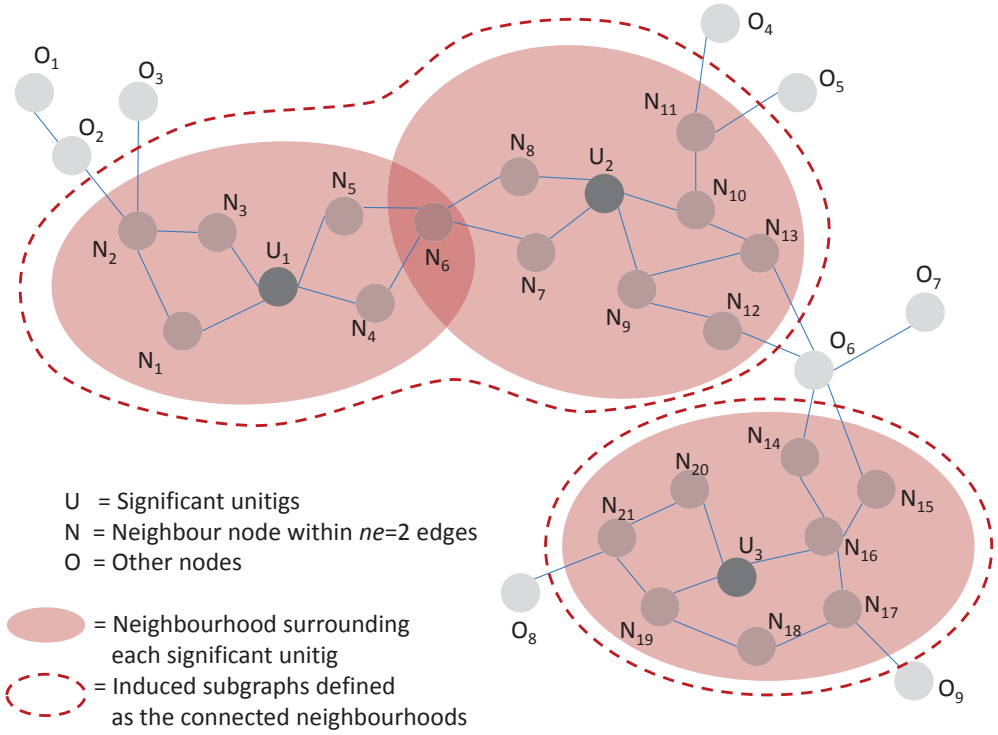


Figure 3.6: **Subgraphs induced by the neighbourhood of significantly associated unitigs.** In this example, a neighbourhood of size $ne = 2$ was used: any unitig distant up to 2 edges from a significant unitig is retrieved to define its neighbourhood. Neighbourhoods are merged if they share at least one node, e.g. the neighbourhoods of U_1 and U_2 are merged because they share N_6, and will be represented in a single subgraph.

The SFF value can be tuned to optimise the number and size of the output subgraphs. It has no impact on subgraphs describing SNPs in core sequences (S3.5 Suppl). On the other hand, when significant unitigs map to different regions of a single MGE, such as a plasmid, several subgraphs are generated but can be gathered into a single subgraph by increasing the SFF threshold (S3.7 Suppl). When significant unitigs map to several distinct mobile regions, which can be found in different contexts (transposon, integron, etc.) at the population level, the resulting subgraph can become very large and highly branching: decreasing the SFF threshold allows to select the few most significant unitigs, generating a subgraph focusing on the most relevant region (S3.9 Suppl). Reducing the graph complexity can also be done by decreasing the ne value, using the `-nh` option.

Representing metadata with coloured DBGs

The subgraphs are enriched with metadata to make their interpretation easier. We use the node size to represent allele frequencies, *i.e.*, the proportion of genomes containing the unitig sequence. We describe the effect β of each unitig as estimated by the LMM using colours, in the spirit of the coloured DBGs [93]. Colours are continuously interpolated between red for unitigs with a strong positive effect and blue for those with a strong negative effect.

Annotating the subgraphs

DBGWAS can optionally integrate an automated annotation step using the Blast suite [39] (version 2.6.0+) on local user-defined protein (`-pt-db` option) or nucleic acid (`-nt-db` option) sequence databases. We annotate the subgraphs of interest by blasting each unitig sequence to the available databases. Users can then easily retrieve the annotations which are the most supported by the nodes in the subgraph, or with the lowest E-value. Importantly, DBGWAS works with any nucleotide or protein Fasta files as annotation databases straight away. However, users can customize the annotation databases by changing the Fasta sequences headers to make DBGWAS results more interpretable. A common example is compacting the annotation in the summary page by using abbreviations or gene class names, and expanding them to full names in the subgraph page. Other custom fields can also be included in the annotation table by adding specific tags to the headers. A detailed explanation on how to customize annotation databases for DBGWAS can be found online[3]. We also provide on the DBGWAS website a resistance determinant database built by merging the ResFinder, MEGARes, and ARG-ANNOT databases [81, 111, 220], and a subset of UniProt restricted to bacterial proteins [197]. Subgraphs discussed in the Results section were annotated using these databases.

Interactive visualisation

DBGWAS produces an interactive view of the enriched and annotated subgraphs, allowing the user to explore the graph topology together with information on each node: allele and phenotype frequencies, q-value, estimated effect, and annotation. The view is built using HTML, CSS, and several Javascript libraries, the main one being Cytoscape.js [67]. Results can be shared and visualised in a web browser. As a large number of components can be produced in one run of DBGWAS, we provide a summary page allowing users to preview and filter the subgraphs. Filtering can be based upon the minimum q-value of all unitigs in the component ($\min_q$), or based on the annotations. A complete description of the DBGWAS interactive interface is available online[4].

Re-running from *step 2* or *step 3*

It is possible to re-run a part of the analysis if a first run with the default values was unsatisfactory. The `-skip1` option allows to re-run from the second step, for instance to compute the lineage effects (adding the `-newick` option). It is also possible to re-run only the third step by using the `-skip2` option, for instance when the default SFF and nh values generated highly connected graphs, or if the annotation was incomplete.

Datasets

We used in our experiments genome sequences from three bacterial species with various degrees of genome plasticity, from more clonal to more plastic: *M. tuberculosis*, *S. aureus*, and *P. aeruginosa*. We also built large datasets with random phenotypes for these 3 species, and used them only for time performance and memory usage assessment. All panels are summarised in Table 3.4.

[3] https://gitlab.com/leoisl/dbgwas/wikis/Customizing-annotation-databases
[4] https://gitlab.com/leoisl/dbgwas/wikis/DBGWAS-web-based-interactive-visualization

Table 3.4: **Microbial panels.**

Species	Genome plasticity	Range of genome length	Panel name	Source	Phenotype	Number of available genomes
M. tuberculosis	very low	4.4 Mbp	TB	[51]	rifampicin	1,197
					isoniazid	1,287
					ethambutol	1,041
					streptomycin	1,166
					kanamycin	671
					ofloxacin	696
					ethionamide	420
					MDR	1,211
					XDR	689
			Large TB	[83]	random	5,000
S. aureus	low	2.7-3.1 Mbp	SA	[79]	methicillin	501
					ciprofloxacin	991
					erythromycin	991
					penicillin	991
					tetracycline	991
					fusidic acid	991
					trimethoprim	323
					gentamicin	991
					rifampin	991
					mupirocin	490
					vancomycin	501
			Large SA	[83]	random	9,000
P. aeruginosa	high	5.8-7.6 Mbp	PA	[200]	amikacin	280
					levofloxacin	117
					meropenem	280
					piperacillin	280
					colistin	164
					polymyxin B	117
					chloramphenicol	103
					cefepime	280
					fosfomycin	113
			Large PA	[83]	random	2,500

We selected 3 bacterial species with distinct levels of genome plasticity, and with antibiotic resistance phenotypes available for several drugs. For each species, we also created large datasets by computing random phenotypes for all available genome assemblies from NCBI RefSeq.

TB panel

M. tuberculosis (TB) is a human pathogen causing 1.7 million deaths each year [213]. This species is known for its apparent absence of horizontal gene transfer (HGT) and, accordingly, most of the reported resistance determinants are chromosomal mutations [82] in core genes or gene promoters. Intergenic regions are also described to be instrumental in multidrug-resistance (MDR) and extensively drug-resistant (XDR) phenotypes [222]. We use the PATRIC AMR phenotype data, as well as genome assemblies from their

resource [51, 204]. We thus gather a total of 1302 genomes after filtering based on genome length. Phenotype data include isoniazid, rifampicin, streptomycin, ethambutol, ofloxacin, kanamycin and ethionamide resistance status. Except for the last three drugs, phenotype data are available for more than a thousand genomes. We reconstruct MDR and XDR phenotypes based on the WHO definition [213]. XDR phenotype could only be defined for 689/1302 strains as it required data for at least 4 drugs. Information on how phenotype data and genome assemblies were obtained is available on the PATRIC website.

SA panel

S. aureus is a human pathogen causing life-threatening infections. It is subject to HGT and many plasmids, mobile elements, and phage sequences have been described in its genome. However, this does not affect the species' genome size, which is always close to 3 Mbp [147]. Most antibiotic resistance mechanisms are well determined by known variants, as shown in a previous study [79]. This study obtained an overall sensitivity of 97% for predicting 12 phenotypes from rules based on antibiotic marker mapping. We use this study panel of 992 strains obtained by merging their derivation and validation sets.

PA panel

P. aeruginosa is a ubiquitous bacterial species responsible for various types of infections. It is highly adaptable thanks to its ability to exchange genetic material within and between species[126]. The species accessory genome is particularly important both in terms of size and diversity, and carries more than half of the genetic determinants already described to confer resistance to antimicrobial drugs [98, 109, 200]. We use a panel of 282 strains, gathered from two collections which mostly include clinical strains: the bioMérieux collection [200] (n=219) and the Pirnay collection [164] (n=63). Genome assemblies and categorical phenotypes for 9 antibiotics are available [98]. Binarised phenotypes of amikacin resistance are available on the DBGWAS project page as an example for users.

Phenotype binarisation

Most available phenotypes are categorical, with S, I and R levels, respectively, for susceptible, intermediary, and resistant. We binarise them by assigning a zero value to susceptible strains (S) and one to others (I and R).

Large panels

We built large panels for the three species, in order to analyse the computational performance at a comprehensive scale. To do so, we gathered all genome assemblies of *M. tuberculosis* (5,504), *S. aureus* (9,331), and *P. aeruginosa* (2,802) available on the NCBI RefSeq bacterial genome repository [83], and removed poor quality genomes. For each panel, we generated random binary phenotypes. For a detailed time and memory assessment, we built several sub-panels from these three large panels at size points of 100, 250, 500, 1,000, 2,500, 5,000 and 9,000 genomes. To build these sub-panels, we sampled genomes uniformly from the panels. To take into account the variability among subsamplings, each sub-panel was randomly built 10 times.

Resistome-based association studies

We benchmarked DBGWAS against a targeted approach to ensure its ability to retrieve all expected resistance determinants. We thus performed association studies under the same model, using as input a collection of known causal resistance SNPs and genes, defining the resistome.
In this validation study, we used bugwas with the same phenotypes and population structure matrix W, so the resistome-based analyses and DBGWAS only differ by their input variant matrix (unitigs *versus* SNPs or genes presence/absence).
For *P. aeruginosa* resistome, we use a variant matrix previously described [98], which includes presence/absence of known resistance gene variants, as well as the SNPs called against these reference gene variants. For *M. tuberculosis* resistome, we built the variant matrix using the same approach as for *P. aeruginosa* [98]: we called the SNPs from a list of 32 known resistance genes and promoters [47, 82, 158]. The time and memory usage required for the complete analysis (from the mapping of the resistance genes and positions on the genome assemblies to the association study) are provided in Tables 3.2 and 3.3.
We sort the annotated features by q-values. S3.10 Suppl and S3.11 Suppl summarise all top variants using their q-value ranks, while Tables 3.2 and 3.3 report the annotations of all variants with a q-value < 0.05 for *P. aeruginosa* levofloxacin and *M. tuberculosis* streptomycin resistance, respectively.

k-mer-based GWAS

pyseer

We installed pyseer [117, 118] commit ID `d17602500a4530b0e68a679ed675fdb12942f56f` (9 commits ahead of pyseer v1.1.1). pyseer pipeline is composed of four steps: 1) k-mer counting; 2) population structure estimation; 3) running pyseer; 4) downstream analysis. To use the correct parameters, we followed the pyseer tutorial[5]. For k-mer counting, we used fsm-lite[6], filtering out all k-mers with a minor allele frequency smaller than 1%. For population structure estimation, we used Mash v2.0 [153]. To run pyseer, we used 8 cores and a LRT p-value threshold of 0.05. Downstream analysis involved getting the k-mers which exceeded the significance threshold (which can be found using the `scripts/count_patterns.py` script), sorting them by LRT p-value, blasting them against the two databases presented in the Interpretation of significant unitigs (step 3) subsection, and keeping the best hit for each k-mer. For reproducibility purposes, the scripts we used to run pyseer can be found online[7].

HAWK

We firstly ran HAWK [174] v0.9.8-beta, as it allows correcting for population structure. Unfortunately, it was unable to find the known causal variants reported for *P. aeruginosa* levofloxacin and *M. tuberculosis* streptomycin resistances by other methods (see Tables 3.2 and 3.3). We therefore kept in our benchmarks an earlier version, HAWK v0.8.3-beta, which presented better qualitative performance for these two evaluated panels. HAWK pipeline is composed of five steps: 1) k-mer counting with a modified version of jellyfish [135]; 2) running HAWK; 3) assembling significant k-mers with ABYSS [95]; 4) getting statistics on the assembled sequences; 5) downstream analysis. The first four steps were

[5] https://pyseer.readthedocs.io/en/master/tutorial.html
[6] https://github.com/nvalimak/fsm-lite
[7] https://gitlab.com/leoisl/DBGWAS_support/tree/master/scripts/pySEER

performed as described in HAWK's github page. However, in the first step, we had to remove the lower-count cutoff in `jellyfish dump` (parameter `-L`), since we are working with contigs and not reads. The last step was performed similarly as the one described for pyseer. For reproducibility purposes, the scripts we used to run HAWK v0.8.3-beta can be found online[8].

[8] https://gitlab.com/leoisl/DBGWAS_support/tree/master/scripts/HAWK_0_8_3_beta

3.3 Concluding remarks

In this Chapter, we presented the DBGWAS software and argued that it is an efficient method for bacterial GWAS. Indeed DBGWAS is a reference-free k-mer-based method: it only requires genome assemblies and phenotypes to identify relevant determinants in any region of the core and accessory genome, including noncoding regulatory regions. Compared to other k-mer-based methods, it was shown to be efficient in terms of time and memory footprint, and improved greatly the k-mer-based result interpretation by clustering significant features which are close in the cDBG into single entities. These entities often represent a particular genomic region such as a polymorphic site in a gene or the insertion of an MGE. We provide an enriched interactive representation of these graphs allowing an easier classification of the variant type: SNPs are identifiable as bubbles with blue (susceptible) and red (resistant) paths, and MGE are represented by linear topologies with mostly red (insertion in resistant strains) or blue (deletion in resistant strains) nodes. The manuscript received encouraging feedbacks when published on bioRχiv[9], and PLOS Genetics peer reviewers were also positive about the use of cDBG to improve k-mer-based GWAS (Appendix S3.2: Decision of PLOS Genetics referees) even though they raised important points from the first submitted version of the manuscript (Appendix S3.3: Response to PLOS Genetics referees).

The interest of researchers for applying DBGWAS for their particular issues stimulated new ideas of functionality developments: working on raw reads instead of contigs, counting k-mers instead of using their presence/absence, computing a score at the subgraph level, adding a measure of the correlation between patterns within and between subgraphs to provide an information on potential linkage disequilibrium, mapping back the unitigs on their initial genomes or to a phylogenetic tree, etc. I hope these developments will be realised in the future.

We began to work on a recurrent demand for more flexibility in the choice of the analysis computing during the second step (analysis of the variant matrix). We implemented a modified analysis step in a prototype, to allow for continuous or ordinal phenotypes. We tested the ordinal regression on unitigs using the same panel and strategy as presented in Chapter 1. The results, presented in Supplementary Table S3.5, showed that DBGWAS was able to find new candidates outside the resistome. No improvement was observed for the phenotypes of limited quality (chloramphenicol, fosfomycin, piperacillin), as discussed in Chapter 1.

Finally, while most subgraphs are straightforward to interpret, a part of them remain hard to interpret without further investigations. This is why we focused our efforts on providing automatic tools to help in this interpretation, as presented in the following Chapter.

[9] https://biorxiv.altmetric.com/details/36093427/twitter

Chapter 4

Predicting DBGWAS graph labels

Predicting DBGWAS subgraph labels is a next logical step after DBGWAS implementation. This was motivated by two lessons learned from our experience of visualising DBGWAS resulting subgraphs.
First, we developed an expertise over time to recognize rapidly a local polymorphism in core or accessory gene, or in a gene promoter, or a gene insertion possibly with its MGE context. However we realised that this interpretation was not obvious to new users.
Second, we regularly came across graphs for which we were unable to conclude on such labels without further investigations. They were too complex or not so typical, as illustrated in the examples of Fig. 4.1.
We also encountered graphs for which a too rapid interpretation based only on the graphical representation led to labelling mistakes. In particular, the insertion of a conserved gene at a conserved position can be mistaken for a SNP. Indeed, as the unitig length does not appear on our graphical representation, a gene of hundreds of conserved base pairs is represented by a few unitigs (Supplementary Fig. S4.1).
We thus felt a need to help DBGWAS users by automatically providing labels informing on the type of variants identified in the output graphs.
Even though this work is very specific to DBGWAS graphs, we think that the proposed strategy could be applied to other tools providing (compacted) De Bruijn graphs to explore genetic variations, such as MetaCherchant [151] or KisSplice [180].

4.1 Introduction

Graph-based representation of genetic variations within a population of genomes is currently a field of intensive research [136, 159], when the alignment-based representation paradigm reaches some of its limits [175]. Numerous tools have already been developed using graph-based representations, for applications to human genetics: transcriptomic analysis with kisSplice [180], GWAS post-processing with HAWK [173], SNP calling with DiscoSNP++ [198], genotyping of variants with coloured DBG [93] or variation graphs [74], and in microbial genetics: prediction of antibiotic resistance phenotype with Mykrobe [29], antibiotic genes in metagenomes with MetaCherchant [151], bacterial GWAS with DBGWAS [99].
Graphs generated from DBGWAS, such as those presented in Fig. 3.3, can be quite straightforward to understand for a trained user, however new users would need a guide to read them correctly. Moreover, some graphs are clearly difficult to apprehend if they do not follow a linear structure, but instead present a complex, highly branching network structure, and if, in addition, they integrate thousands of unitigs mapping to a large number of protein annotations.

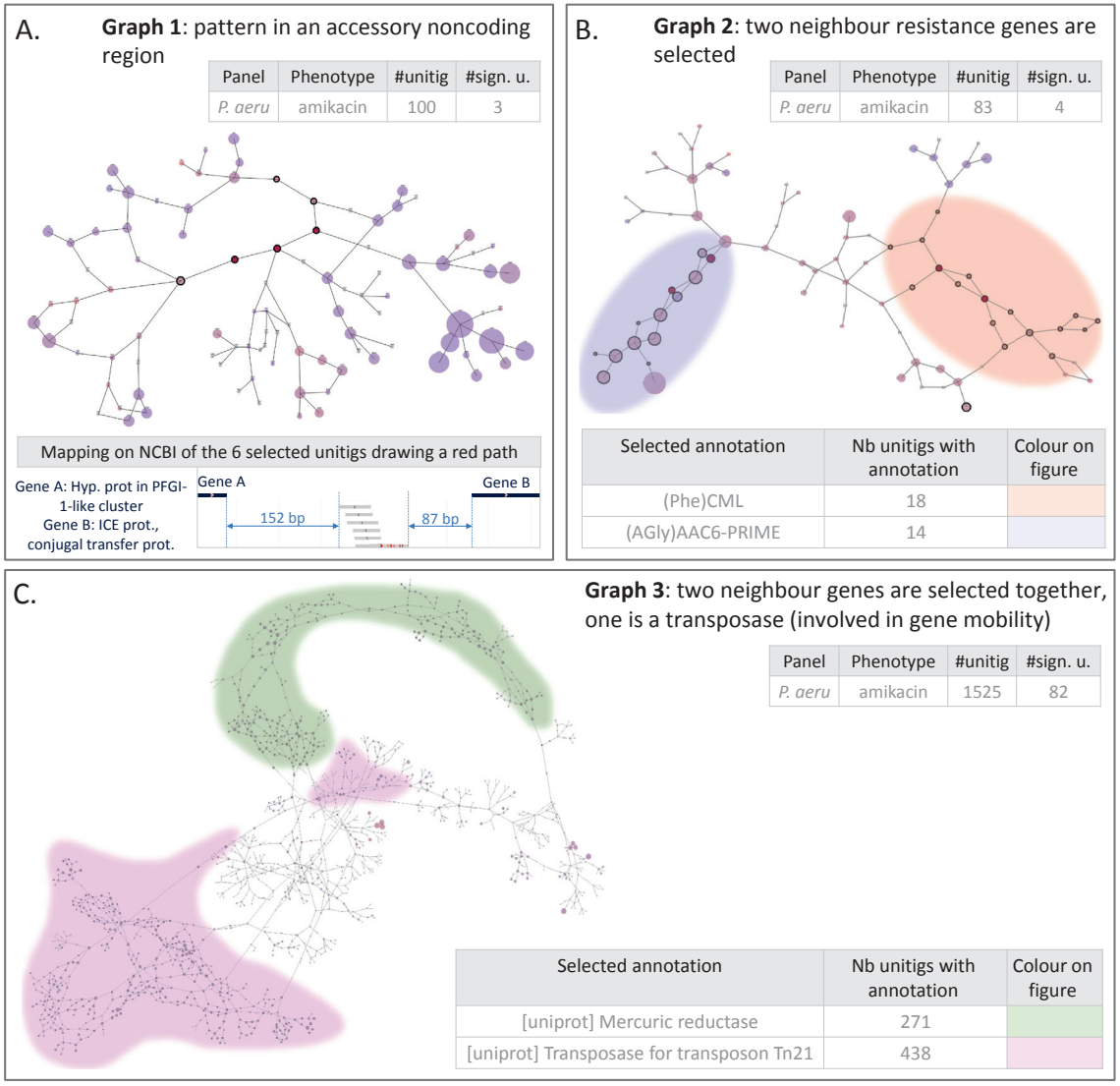


Figure 4.1: **Examples of complex graphs.** Panel A presents a graph with few nodes, however not obviously a SNP or a gene insertion: its annotation reveals a particular sequence found in a noncoding region between two accessory genes. DBGWAS annotation functionality allows to better understand the complex graph presented in Panel B: unitigs belonging to 2 resistance genes were found significant, however, as they were not distant enough, they were gathered in a single graph. That is also the case for the graph in Panel 3, however, the high polymorphism of the genes, and the fact that these genes are found in several copies make the number of nodes in the graph very large, and its topology hard to apprehend.

Mapping the most significant unitig sequences of such graphs to the NCBI nucleotide database [16] allowed us to find some clues on why these graphs were complex. We identified that such graphs with a low number of unitigs (Fig. 4.1A) often integrate a path of red (positively-associated) nodes lying in a non-coding region between polymorphic accessory genes (here, integrative and conjugative elements). They seem to be markers of some mobile genetic elements, however as these paths stand in a noncoding region between polymorphic genes, their neighbour unitigs branch to various other unitigs, making the structure complex and hard to interpret.
When several distinct variants are aggregated because they are close to each other in at least one strain of the studied panel, this also generates complex graphs. Indeed, if their neighbourhoods (as defined with DBGWAS `nh` parameter) overlap, they are gathered in a single graph. This is the case of *aac(6')* acetyltransferase and CML efflux pump in Fig. 4.1B. Moreover, the interpretation of the complete graph can lead to mistakes as the most significant node does not represent the *aac(6')* gene insertion itself, but a particular SNP in the gene, as presented in Table 3.1.
Graphs become even more complex when they gather genes found in several copies in each genome and are found in variable genomic environments. In particular, when genes involved in gene mobility, such as transposases or integrases, are captured in a graph, the number of unitigs explodes, as in Fig. 4.1C. In this example, the transposase for transposon Tn*21* is selected from the association test. This transposon is known to disseminate multiple antibiotic resistance genes, but it also carries a mercury resistance operon containing the *merA* gene [122]. The *merA* gene is also significantly associated with the phenotype and, because of its proximity to the transposase, both appear in a single graph.
More generally, the graphs representing genes found in several copies contain multiple loops and branches, which makes them more complex to interpret.
When several close genes are gathered in a single graph, it can be simplified by breaking some edges and splitting it into several subgraphs. This can be done by modifying the parameters of DBGWAS third step (`nh` and `SFF`). Indeed lowering `nh`, the number of edges defining the significant unitig neighbourhood (Fig. 3.6) can break links between significant unitigs which are distant of $d > (\texttt{nh} \times 2)$ - 1 edges. Lowering the `SFF` Significant Features Filter parameter can also help as it would select less significant unitigs from which to build the neighbour graphs. In this study, we consider strategies based on the `nh` parameter.
We have two aims in this work. First to guide non-expert users of DBGWAS in the graph interpretation. And second, provide advice when encountering a complex graph, such as the use of a better `nh` value. To reach both aims, we need first to build a reliable label prediction for the easy cases. 'Easy' here is defined in opposition to the complex graphs presented above. Practically, we build a dataset of labelled graph from real and simulated data, and use it to train models to differentiate between local polymorphisms (LP) and mobile genetic events (MGE). We build models for each value of `nh` from 1 to 5.

4.2 Methods and algorithms

Accurately discriminating between local (SNP, indel, etc.) and large scale polymorphism (insertion or deletion of a large sequence such as a gene, a plasmid, etc.) is crucial when interpreting DBGWAS results.
We thus define a first supervised classification task: the discrimination between local polymorphism (LP) and mobile genetic elements (MGE). The LP class integrates single or multiple SNPs/indel, in core or accessory genes, while the MGE class comprises insertion or deletion of a large sequence.

We also define a second supervised multiclass classification task to discriminate between the different LP and MGE subclasses: single LP, multiple LP (several SNPs in a single graph), MGE insertion, and MGE deletion.

4.2.1 Datasets

We build a dataset of labelled graphs which is further randomly split into a training (two thirds of the dataset) and a test dataset. The training dataset is used to learn predictive models and select the optimal one, while the test dataset is used to compute the prediction performances. The dataset comprises labelled graphs obtained from the antibio-resistance analyses presented in Chapter 3, and simulated graphs.

Labelling real data

We manually annotate 997 graphs from the graphs labelled during the analyses presented in Chapter 3. We keep only the easy graphs. In Chapter 3, the graphs were generated with DBGWAS default settings: `nh`=5 and `SFF`=100. We re-run the third step of DBGWAS using 5 values of `nh` (from 1 to 5) and 3 values of `SFF` (15, 70 and 100). We are able to transpose the known labels to the new graphs because the new tested values of `nh` and `SFF` lead to less complex graphs, from the same DBGWAS step 2 output: the q-values and annotations related to each unitigs are identical. This allows to identify the new graphs containing the same min_q as the ones obtained with the maximal `nh` and `SFF` values, and validate it using the annotation.
Because this process does not generate independent graphs – several of them describe the same variant – we stratify the graphs among the training and test datasets using the antibiotic names, so the graphs describing variants identified for a particular antibiotic are all grouped in one split (either training or test). Finally the training dataset comprises 695 real graphs, related to 11 antibiotics while the test dataset comprises 302 graphs related to 10 antibiotics.

Simulation of labelled graphs

We extend the dataset with graphs obtained from simulated genomes and phenotypes. To build this synthetic dataset, we define a global strategy to generate controlled LP and MGE graphs using the *P. aeruginosa* dataset described previously [99], composed of $N = 282$ assembled genomes, and for which a collection of core and resistance gene alignments is available. The core genes were previously obtained by annotating the genome assemblies and defining core gene families. For each gene family, the gene sequences were extracted from all genome assemblies and a multiple alignment was computed between them.
We define synthetic core genomes by concatenating a selection of C ordered core genes randomly picked among the 1384 available core genes, as described in Chapter 2 and in Figure S2.1, panels A, B and C. For both LP and MGE simulations, we use a general procedure to insert the phenotype-associated variants and obtain controlled graphs: we sample a vector of mutation pattern, X, from a binomial distribution with a prevalence p (taking values between 0.15 and 0.5). If $X_i = 1$, the i-th genome is mutated ; if $X_i = 0$, it is not. Then we sample a phenotype Y_i of each genome i from a multivariate logistic model, related to X_i under a given odd-ratio (taking values between 4 and 10):

$$Y_i \sim \mathcal{B}(\pi_i), \quad \pi_i = \frac{1}{1 + e^{-\mathbf{X_i}\beta}}, \quad \beta = log(\text{odd-ratio}) \tag{4.1}$$

In order to lower the number of DBGWAS runs, when possible, we generate several

graphs per run. To do so, we simulate M variants, avoiding close proximity between them in the genome to prevent overlaps of significant unitig neighbourhoods. Indeed this would result in collapsing several graphs into a single one. We also use a q-value cutoff for significant pattern selection by setting the `SFF` parameter to 0.05, in order to control a false discovery rate at 5% on average. All graphs generated by such a DBGWAS run are assigned a label, according to the simulation strategy used to generate the graph, as described below. Finally, several values of `nh` (from 1 to 5) are used for the third step of DBGWAS.

The following paragraphs present the specificity of each label, within this simulation framework.

Simulation of genomes with MGE insertion or deletion.

We used two strategies to simulate genomes with MGE presence associated to a phenotype. In both strategies, the MGE sequence is not inserted within the core genome, but included in the genome Fasta file as a distinct sequence.

The first strategy uses a sub-collection of M gene alignments (core and resistance genes) of N sequences, and adds the i-th sequence of the j-th alignment to the i-th genome, according to its X_i value, as detailed in Algorithm 1.

Prepare a core genome of $C = 200$ **genes.**
for i *in* $1..N = 282$, j *in* $1..M = 50$ **do**
  if *type* = *'insertion'* **then**
    if $X_i = 1$ **then** Add $sequence_i$ of $alignment_j$ in genome i;
    if $X_i = 0$ **then** Do nothing;
  else if *type* = *'deletion'* **then**
    if $X_i = 1$ **then** Do nothing;
    if $X_i = 0$ **then** Add $sequence_i$ of $alignment_j$ in genome i;
  end
end

Algorithm 1: MGE simulations based on the *gene* strategy

The second strategy only allows for MGE insertions, however it uses real plasmid sequences to represent a particular MGE context. The purpose is to add more complexity to the MGE graphs by adding a varying genomic environment.

For this strategy, all genomes receive a plasmid, however if $X_i = 1$, the i-th genome receives a plasmid sequence containing a given annotation, and if $X_i = 0$, the i-th genome receives a plasmid sequence which does not contain the annotation, as detailed in Algorithm 2. The plasmid collection is built from the PlasFlow database [108]. The annotation of the plasmids is obtained by determining resistance gene presence within the plasmid collection, by blast search: over the 7604 plasmid sequences of the database, 1848 have at least one hit on a resistance gene database. This corresponds to 349 resistance gene families. However after filtering gene families observed in less than 15 plasmid sequences, only 146 families are retained.

Prepare a core genome of $C = 50$ **genes.**
Sample *Res*, **a resistance gene family among the** 146 **mapped in at least** 15 **plasmids.**
for i *in* $1..N = 282$ **do**
  if $X_i = 1$ **then** Add in genome i a plasmid containing a *Res* gene;
  if $X_i = 0$ **then** Add in genome i a plasmid NOT containing any *Res* gene ;
end

Algorithm 2: MGE simulations based on the plasmid strategy

In this strategy, only one variant associated to the phenotype is integrated – the insertion of a resistance gene carried by a plasmid. Only the first graph – expected to represent

this variant – is kept, so one run of DBGWAS leads to only one 'MGE'-labelled graph. Using both strategies, 6127 graphs were simulated with an 'MGE' label, among them, half of gene-based insertions, a quarter of gene-based deletions and a quarter of plasmid-based insertions. This corresponds to 1225 graphs on average per `nh` value.

Simulation of genomes with LP mutations.

We build synthetic genomes by preparing a sequence s of $C = 150$ core genes, and among them we pick $M = 50$ genes which will receive an LP mutation. In order to simulate LP mutations in both core and accessory genes, we apply an independent gene frequency threshold f taking values between 0.15 and 1 to build a vector of mutated gene presence/absence $G_i \sim \mathcal{B}(f)$: if $G_i = 0$, the M genes are removed from the core gene sequence of the i-th genome.

Several types of LP are then created, to ensure some variability within the LP class. First, the nb_mut parameter allow to distinguish between single and multiple LP in a gene, as presented in Fig.4.2. Using the $dist$ parameter, multiple SNPs/indels can be grouped, and thus will represent a 'hotspot' or can be distant, for 'sequential' mutations. Finally, multi-allelic SNPs are simulated by adjusting an $allele\_fq$ parameter, as described in Algorithm 3.

M **contains the indices of the 50 genes to modify.**
for i *in* $1..N = 282$, j *in* $1..C = 150$ **do**
  1. Define nucleic sequence on which to work
  if $j \in M$ *and* $G_i = 1$ **then**
    2. Pick mutation position
    for $k = 1$ **to** nb_mut **do**
      Pick a position, pos_k, avoiding the gene extremities
      (150 first and last positions);
      if $k > 1$ **then** apply $dist$ between pos_k and pos_{k-1};
    end
    3. Do the mutation
    foreach *position* **do**
      $Allele \sim \mathcal{B}(allele\_fq)$;
      if $X_i = 1$ **then**
        if $Allele = 1$ **then** $Base \leftarrow$ 'A';
        if $Allele = 0$ **then** $Base \leftarrow$ 'C';
      if $X_i = 0$ **then**
        if $Allele = 1$ **then** $Base \leftarrow$ 'G';
        if $Allele = 0$ **then** $Base \leftarrow$ 'T';
      Replace position by $Base$;
    end
  end
end

Algorithm 3: Definition of the LP types

A total of 5868 graphs with an 'LP' label were generated using this strategy, among them 1467 graphs on average of each of the following types: accessory-single, accessory-multiple, core-single and core-multiple. This corresponds to 1175 graphs per `nh` value, on average. As the simulations are based on a finite set of genes, a stratification is also a applied to limit the dependency between the training and test datasets. The simulated training dataset is assigned the 1522 first gene alignments, over 1920. Doing so, the training dataset comprises a total of 9420 simulated graphs and the test dataset 2575 simulated graphs. The test dataset is used to assess final generalisation performances of the prediction model selected from the training dataset analysis.

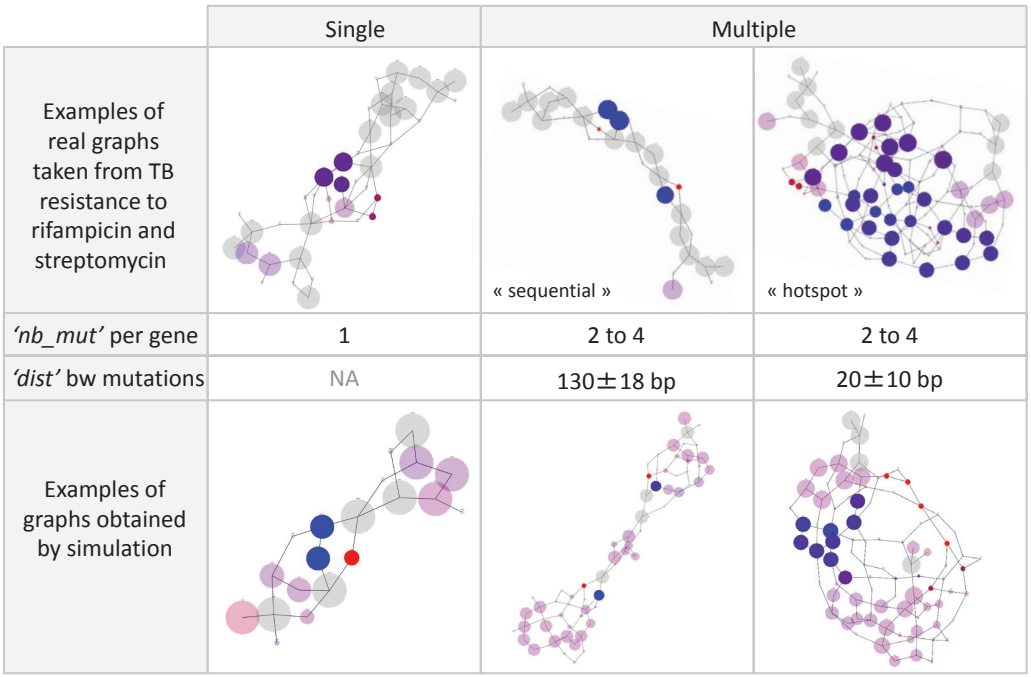


Figure 4.2: **Simulations for the LP class.** .

Training datasets.

Both real (695 graphs) and simulated (9420 graphs) training dataset are merged into a 'all `nh`' training dataset. This dataset is split according to the `nh` value used to build the graphs, in order to train prediction models for each `nh` value. Each of the 5 '`nh`' models is thus trained on 2040 graphs on average.

4.2.2 Features used for the model

Description of the features.

We designed 33 features to describe the graphs and predict their label. These features are obtained from data generated during the first and second steps of a DBGWAS run and are computed during the third step of a customised version of DBGWAS. They summarise the graph size and complexity, its nodes estimated effects – from the association test, as defined in Chapter 3, allele frequencies, and the length of the node's corresponding nucleic sequences. Labels are assigned at the graph level, so when the descriptors are at the node level, a summary of their distribution, such as the means or the quantiles, is computed (Table 4.1).
We intuitively searched to reassemble unitigs with a positive (resp. negative) effect into 'positive' (resp. 'negative') contigs, based on the idea that a local polymorphism would be represented by a 'positive' and a 'negative' contigs of approximatively the same length and with a high level of homology, while a long insertion would present a difference in contig length and a low homology.
Contig-related variables are computed during the third step: an assembly of the node sequences is generated using the seqAn C++ library [176]. In the case of graphs containing several hundreds, or thousands of nodes, such as the graphs generated with the plasmid-based strategy, this step took several hours, and we had to apply a timeout in order to produce all covariates in a reasonable amount of time. As a consequence, only a few of the plasmid-based graphs have values for the contig-related variables.

Table 4.1: Features describing the graphs and used to model their label.

	Feature name	Feature description
Graph size	node_number	ng, Total number of nodes in the graph
	sig_node_number	ns, Number of significant nodes.
	sig_node_ratio	Ratio of significant nodes: ns/ng.
Graph complexity	node_degree*	Distribution of node degrees. The degree indicates the number of edges connected to a node.
	sign_node_degree*	Distribution of degree for significant nodes only
Allele frequency	allele_fq_mean	Mean allele frequency.
	allele_fq_sd	Standard deviation of allele frequencies.
	diff_alleles	Absolute difference of allele frequencies between phenotype 1 and phenotype 0.
Node effect	node_effect*	Distribution of estimated effect for all nodes.
	sign_node_effect*	Distribution of estimated effect for significant nodes only.
	pos_effect_ratio	Ratio of significant nodes with a positive effect (nsp), among significant nodes: nsp/ns.
Sequence length	sum_length	Sum of sequence length of all nodes.
	length_pos_sum	sp, Sum of length of nodes with a positive effect.
	length_neg_sum	sn, Sum of length of nodes with a negative effect.
	length_pos_mean	Mean length among nodes with a positive effect.
	length_neg_mean	Mean length among nodes with a negative effect.
	diff_length	Absolute difference of negative versus positive effect sequence length, normalised by the maximal sum: $\|sp - sn\|/max(sp; sn)$.
	max_contig_pos	cp, Length of the longest contig we were able to reassemble from positive node sequences.
	max_contig_neg	cn, Length of the longest contig we were able to reassemble from negative node sequences.
	diff_contig	Absolute difference of negative versus positive effect contig, normalised by the maximal contig length: $\|cp - cn\|/max(cp; cn)$
	contig_homology	Homology score betwen the longest contig reassembled from positive, and negative effect nodes.

* For these features, 4 variables were computed to represent the feature distribution: 5%, 50% and 95% quantiles, and the standard deviation.

Dealing with NA values.

A total of 8 features can produce NA values. This is the case for $sign\_node\_effect\_sd$ and $sign\_node\_degree\_sd$, when there is only one significant node in the graph, and for $length\_pos\_mean$ and $length\_neg\_mean$, when there is no node with a positive – resp. negative – effect on the phenotype. In these cases, we replace the NA values by 0.

This is also the case for the 4 contig-based features, when no assembly was computed due to timeout, as described previously. In this case, we evaluated two methods to deal with the NA values. The first method, 'repVal' replaces the NA values by 0, as for other variables. The second method 'remCol' removes the four contig-based variables from the

study. Even if we can expect a loss of information by removing variables, this method is worth being tested with regards to the prediction performances, because the computation of the contig-based variable values is time consuming.

Data normalisation.

The training variable matrix x, describing the variable values for all graphs in the training dataset is scaled and centred, so all variables have an average 0 values and a standard deviation of 1.
Once the classifier method is selected and optimised on the scaled training dataset, it is used to predict labels of new scaled datasets. The scaling factors used for the training dataset (mean and standard deviation of each column of x) are applied to scale any new dataset.

4.2.3 Benchmarking prediction models.

The main challenge when building a prediction model is to have it to be able to predict correctly on any new independent dataset. Minimising only the model training loss, can generate a model too specific to the training dataset, which is less generalisable to new data [69]. We refer in this case to optimistic models and overfitting. The overfitting is related to the model complexity: the more complex, the better the fit on the training dataset will be, however to the detriment of prediction generalisation. This issue is also known as the bias-variance trade-off: too simple models lead to a large bias, while too complex models lead to a large variance.
A simple and widely used method to select, among several, a model offering the best bias-variance trade-off, is cross-validation (CV) [69]. The K-fold cross-validation strategy directly estimates the average generalisation error by splitting the dataset into K-folds, learn on $K-1$ and assess the model performance on the K-th fold. Performances obtained across the K runs can be then aggregated. The model with the best estimated performance is selected, and its prediction performance is assessed on a new independent test dataset.
We include in our study six prediction methods each defining a family of models with distinct parameter values. We use a 10-fold cross-validation strategy to optimise the model hyperparameters – determining the model complexity and behaviour – and select the best optimised method among the six. Finally, we estimate the prediction performance of the selected model on the test dataset.
We now present the prediction model families we included in this study, as well as the hyperparameters we optimised.

Penalised regression

Penalised regressions are a family of methods for empirical risk minimisation, which explicitly implements the bias-variance trade-off: the minimisation problem can indeed be expressed by two terms, a loss function describing the fit to the data, and a regularisation function, which provides a cursor for tuning the model complexity (see Eq 4.2).

$$\min_f \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda\Omega(f), \tag{4.2}$$

where f is a model family, $L(y, f(x))$ is the loss function to be minimised, Ω a complexity measure for functions f, and λ a parameter which allows controlling the trade-off between model complexity and learning error.

Here, we use a logistic loss function for the binary classification tasks, as described in Eq 4.3.

$$L(y, f(x)) = \log(1 + \exp^{-yx^\top \beta}), \tag{4.3}$$

with y, the graph label vector, X the matrix describing the feature values for each graph and β the vector of coefficients to estimate. For this task, an 'LP' label corresponds to a value of $+1$ and an 'MGE' label of -1.
For the second task, a multiclass classification problem, a multinomial loss function is used, which is a multiclass generalisation of the logistic function[1]. The label is attributed to the class with the highest probability.
We test three widely used regularisation functions: the $l1$-penalty (lasso) [191], the $l2$-penalty (ridge) [87] and a mixture of the two (elastic-net) [227], which are expressed in Eq 4.4. The two main characteristics of the lasso penalty are the shrinkage of the regression coefficients and sparsity: penalising the $l1$ norm leads to estimators in which many coefficients are exactly 0. This increases the model bias, however decreases the variance of the prediction results. Indeed, thanks to the shrinkage and sparsity, it lowers the complexity of the model and improves the overall prediction accuracy by preventing the model from overfitting the training data. The ridge penalty does not induce sparsity, and thus will not chose one variable among several correlated variable as the lasso does, but will provide close coefficient estimates to variables which are close in Euclidean norm. The elastic-net is a compromise between lasso and ridge penalties: it is sparse, and tends to select groups of correlated variables.

$$\lambda\Omega_\alpha(\beta) = \lambda\left(\frac{1}{2}(1-\alpha)||\beta||_{l2}^2 + \alpha||\beta||_{l1}\right)\begin{cases}\text{lasso} & \text{, if } \alpha = 1,\\ \text{ridge} & \text{, if } \alpha = 0,\\ \text{elastic-net} & \text{otherwise.}\end{cases} \tag{4.4}$$

λ and α are the two hyperparameters to optimise while selecting the best model.
We use the R package `glmnet` [70] to implement the three penalised methods. We optimise the λ and α values as described in paragraph **Model optimisation and selection**.

Support vector machine

The support vector machine (SVM) algorithm was designed for binary classification [186]. It is based on a linear classifier and fits into the penalised risk minimisation framework described in Eq 4.2: it can be expressed with a hinge loss function (Eq 4.5) and a ridge penalty (case $\alpha = 0$ in Eq 4.4).

$$L(y, f(x)) = (1 - y(x^\top \beta + b))_+, \tag{4.5}$$

where y is the graph label vector, X the matrix describing the feature values for each graph, and β and b are the estimated parameters defining the hyperplane and its margin, used to separate the data according to their label y, in the input parameter space. The support vectors are X_i belonging to the margins.
The overfitting is controlled by a cost $C = 1/\lambda$, which defines the tolerance to classification errors. The lower C, the larger the margin, and the higher the tolerance to misclassified data.
A non-linear transformation of the input space $\mathcal{X}$ to a feature space $\mathcal{F}$ can be considered to apply a linear SVM in $\mathcal{F}$. This can be particularly interesting when the data are not linearly separable. This problem can be solved by the use of kernel methods, which allow to compute dot products defining the transformations [201]. A kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ can

[1] http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression

be seen as a measure of similarity between two points x and x'. The radial basis function (RBF) kernel (Eq 4.6) is widely applied to SVM:

$$K(x, x') = \exp(-\gamma||x - x'||^2), \tag{4.6}$$

where γ is the parameter of the kernel. The lower γ is, the slower the similarity decreases with the distance between x and x'. Each data point has non-zero similarity to a larger number of samples, making the model less prone to overfitting.
We use the R package `e1071` [70] to implement two methods: a linear SVM and an RBF-kernelised (radial) SVM. We optimise the C value for the linear SVM and $(C; \gamma)$ value couples for the radial SVM, by applying the strategy described in paragraph **Model optimisation and selection**.

Random forest

Random forests belong to a different family of classification methods. They are based on decision trees, recursively splitting the dataset into binary parts according to a threshold applied to a variable's value. They end with a division of the input space into multiple regions, and a label is then attributed to each region based on the majority class. As single trees do not provide a stable solution, and do not generalise well,the random forest algorithm proposes to compute several independents trees and attributes the most frequent predicted label among all trees. It also includes a bagging (for bootstrap aggregating) strategy: each tree is computed on a bootstrap sample of the training data, and for each new node in the tree, the variable used for splitting the data is chosen among a random sample of all variables, which helps decorrelating the trees. This generates less predictive individual trees, however improves the generalisation capacity of the method [32].
The main hyperparameters to tune for these models are the number of trees (we used the default value of 500 provided by the R package `randomForest` [121] used for the algorithm implementation), the number of randomly selected variables m used in the bootstrap, and the *node_size*, the minimum size of each final region. This is correlated to the number of regions dividing the input space: the lower the *node_size*, the more local (and less generalisable) the model will be.
We optimise both m and the *node_size*, using the strategy described in paragraph **Model optimisation and selection**.

Model optimisation and selection

We use functions from the R package `mlr` [22] to design a hyperparameter tuning strategy: we pick 100 random values within the parameter search space (see Supplementary Table S4.6), and compute performances using a 10-fold cross-validation. We keep the hyperparameter values of the model maximising the accuracy.
For all model tuning, we draw performance curves/maps to check that the tested hyperparameter spaces indeed lead to a maximal performance value, and adjust the tested value boundaries when needed (Supplementary Fig. S4.3).
Finally the method – among the six – with the overall best accuracy is retained and its prediction performances are then computed on the independent test dataset.

4.3 Results

After a brief presentation of the graph simulation results, we present in this section how we chose the best models for the predictions (12 models were selected: one per `nh` value

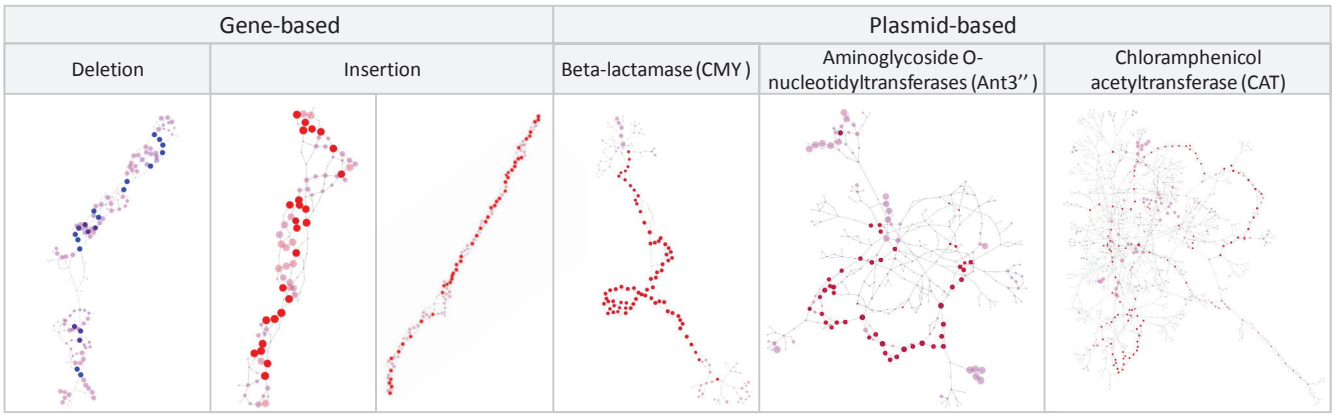


Figure 4.3: **Simulations for the MGE class.** Examples of graphs simulated from the insertion or deletion of simple gene or gene within a plasmid. The gene-based methods produces similar topologies while the diversity of the graphs generated from plasmid can be very important, depending on the gene family inner variability (from low in the case of CMY to very high, for CAT).

plus one gathering all `nh` values, and one per classification task). We then present some applications of these prediction models to the graphs in Table 3.1, and to the complex graphs described in Fig. 4.1

4.3.1 Graph simulations

For this study, around 12000 graphs were simulated and labelled. Several pre-tests were done before launching all the simulations in order to optimise the simulation strategies and parameters.
Globally, the generated graphs showed the expected topology, as presented in Fig. 4.2 and 4.3. For the plasmid-based strategy, the complexity of the graphs depends on the chosen gene family: a conserved gene family produces linear paths while heterogeneous families produce highly branching graphs, as illustrated in Fig. 4.3.
Even though the gene-based strategy generated more regular graph topologies, the pre-tests encouraged us to apply a filter in order to remove graphs comprising less than $2+3\times(\texttt{nh}-1)$ nodes. Indeed, some gene alignments were so conserved that the complete sequences were collapsed into a few unitigs (Supplementay Fig. S4.4). Because we introduced artificially these sequences independently from the others, the generated subgraphs only had 1 to 3 nodes with `nh`=5, which would never happen in practice. Thanks to this filter, we guarantee a minimal graph neighbourhood.
The pre-tests also revealed another limit encountered in the generation of 'sequential' multiple SNPs/indels : depending on the variability of the gene in which the mutations were inserted, the neighbourhood of two significant nodes involved into two distinct mutations does not always overlap, which results into a splitting of the 'LP multiple' graph into several 'LP single' graphs. This has however no impact on the binary LP/MGE labelling.

4.3.2 Exploratory analysis

A principal component analysis (PCA) was computed on the training dataset, for each value of `nh`, as illustrated in Fig. 4.4. We first validated that the simulated graph populations overlapped with graphs obtained from real data. This was the case, even though the 'LP' simulated graph distribution was closer to the 'MGE insertion' population than

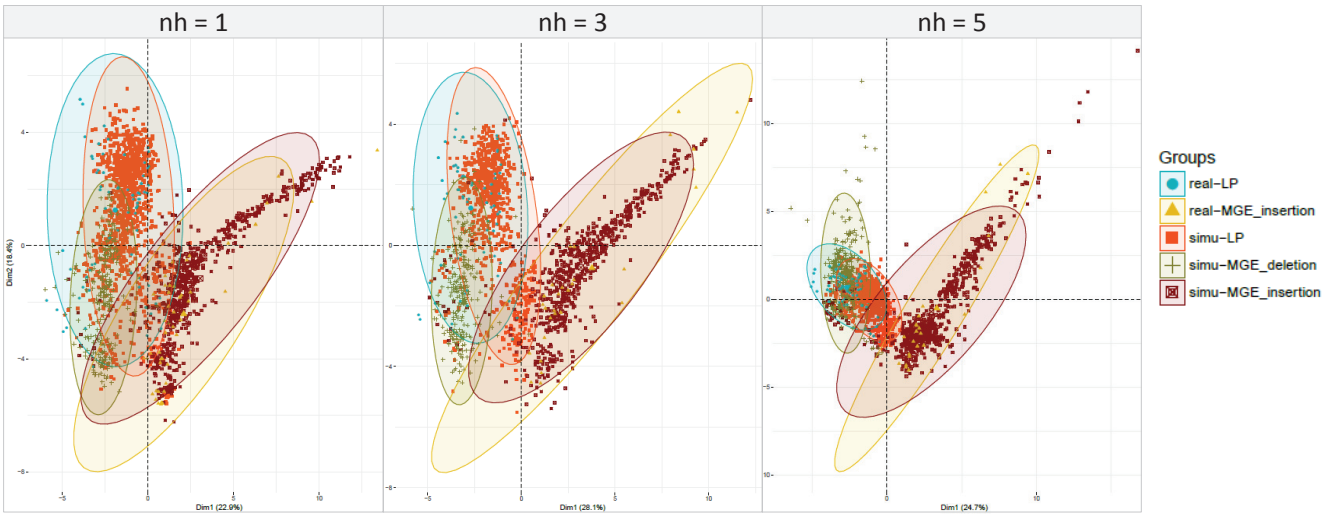


Figure 4.4: **PCA for 3 values of `nh`, using the 'repVal' method to deal with NAs.** In this data projection, 5 colours were used to distinguish between 'LP' labels produced by simulations (red) from those obtained from real data (blue), and between 'MGE' from real data (yellow) and from simulations. Within this last class, MGE insertions were coloured in brown while deletion in green.

the real 'LP', in the two first PC projection space. We also observed that, in this space, the 'MGE-deletion' class overlapped the 'LP' graphs. The 'MGE-deletion' graphs differ from the 'MGE-insertion' mostly by the sign of their effect on the phenotype – apart from that, the same simulation parameters were used. Variables describing the node effects are important in the PC1 composition, and this may explain why both categories of graphs were found distant in this projection.
More precisely, for `nh` = 5, PC1 was mostly composed of variables describing the unitig allele frequency (bottom-left block in the correlation matrix of the covariates, presented in Supplementary Fig. S4.2), the sum of unitig sequence length (top-right block in Supplementary Fig. S4.2), and the median estimated effect, while PC2 was mostly composed of the estimated effect extreme quantiles and standard deviation, and the difference of negative *versus* positive effect sequence length. For `nh` = 1, when only one neighbour node was considered around the significant unitigs, PC1 was mostly composed of variables describing the node number and node length, while PC2 was mostly composed of the ratio of significant nodes, the difference of negative *versus* positive effect, sequence length, and of variables describing the node degree.

4.3.3 Model selection

We based the selection on the prediction accuracy of the model, *i.e.* the overall rate of good predictions. Figs. 4.5 and 4.6 show the mean cross-validation accuracy for the binary LP/MGE (resp. multiclass) classifiers, for each value of `nh`, and for the two methods applied to deal with the NA values: 'repVal' method replaced NAs by '0' and 'remCol' method removed variables with NAs.
The random forest classifier provided the best accuracy in 19 out of 24 models (Figs. 4.5 and 4.6). The radial SVM was a close second while the four other methods presented worse performances. We selected the random forest method for all models (tasks and `nh` values), and optimised it independently for each model.
The influence of the `nh` value on the CV performance was not the same on all methods: the ridge regression performance for instance was particularly affected by the `nh` value

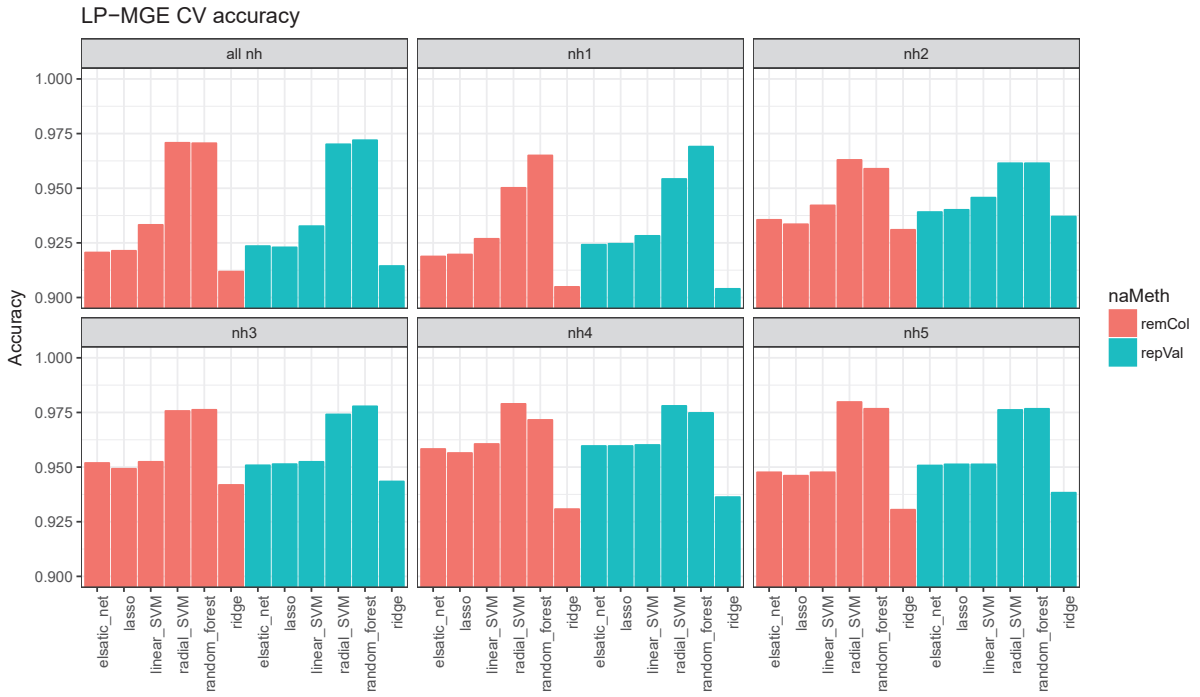


Figure 4.5: **Cross-validation mean accuracy, LP/MGE prediction.** Mean accuracy obtained by 10-fold cross-validation for each model, for all values of `nh`, and for both NA methods: 'repVal' replaced NA values by '0' and 'remCol' removed variables with NA values.

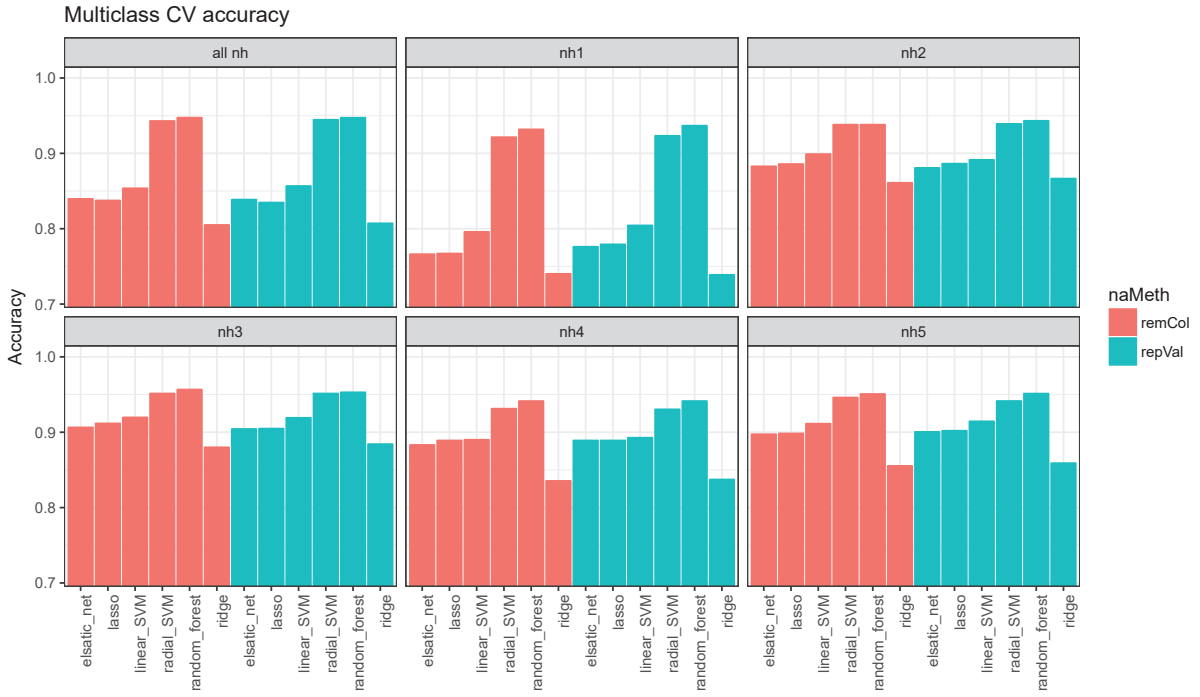


Figure 4.6: **Cross-validation mean accuracy, multiclass prediction.** Mean accuracy obtained by 10-fold cross-validation for each model, for all values of `nh`, and for both NA methods: 'repVal' replaced NA values by '0' and 'remCol' removed variables with NA values.

and obtained accuracy values between 0.74 and 0.88 for the multiclass task while the random forest obtained accuracy values between 0.93 and 0.96 for this task. Regardless of the method and the prediction task, the worst performance was obtained for graphs with a small neighbourhood (`nh`=1 or `nh`=2), which may reflect the importance of the neighbourhood for label prediction. The performances obtained with the random forest models trained across all values of `nh` ('all `nh`' in the Tables) were better than the ones obtained for graphs with `nh`=1 or `nh`=2, but were never the best.

The method used to deal with the NAs had no major impact on the performances: we thus selected the 'remCol' strategy for the rest of the analysis, as computing the contig-related covariates was particularly time-consuming.

Variable importance was computed for all random forest classifiers. It represents the average decrease in node impurity over all trees, when splitting on the variable. The node impurity is measured by the Gini index: $G(i) = 1 - \sum_{k=1}^{K} P_i[k]^2$ where $P_i[k]$ is the proportion of class k in node i.

Table 4.2: Random forest models: top-5 variables sorted by variable importance

		LP-MGE	multiclass
nh=1	1	sign_node_effect_sd	sign_node_effect_Qu0.05
	2	diff_length	sign_node_effect_Qu0.95
	3	sign_node_effect_Qu0.05	length_neg_sum
	4	sign_node_effect_Qu0.95	sign_node_effect_Median
	5	diff_alleles	pos_effect_ratio
nh=2	1	sign_node_effect_sd	sign_node_effect_Qu0.05
	2	diff_length	length_neg_sum
	3	sign_node_effect_Qu0.05	length_neg_mean
	4	length_pos_sum	pos_effect_ratio
	5	pos_effect_ratio	sign_node_effect_Qu0.95
nh=3	1	diff_length	node_effect_Median
	2	sign_node_effect_sd	sign_node_effect_Qu0.95
	3	allele_fq_sd	allele_fq_sd
	4	node_degree_Qu0.95	pos_effect_ratio
	5	sign_node_degree_Qu0.95	sign_node_effect_Qu0.05
nh=4	1	diff_length	node_effect_Median
	2	sign_node_effect_sd	sign_node_effect_Qu0.05
	3	allele_fq_sd	sign_node_effect_Qu0.95
	4	node_effect_Median	length_pos_sum
	5	pos_effect_ratio	diff_length
nh=5	1	diff_length	node_effect_Median
	2	sign_node_effect_sd	length_neg_sum
	3	node_effect_Median	sign_node_effect_Qu0.05
	4	sign_node_effect_Qu0.95	sign_node_effect_Qu0.95
	5	allele_fq_sd	pos_effect_ratio
all `nh`	1	diff_length	sign_node_effect_Qu0.05
	2	sign_node_effect_sd	sign_node_effect_Qu0.95
	3	sign_node_effect_Qu0.95	length_neg_sum
	4	sign_node_degree_Qu0.95	pos_effect_ratio
	5	allele_fq_sd	node_effect_Median

Table 4.2 shows the top-5 variables used in each model, ordered by variable importance. Globally, we observe that variables describing the node effect were always among the top-

5. The high correlation between these variables (as shown in Supplementary Fig. S4.2) explains that several of these variables were found together. Variables describing the sequence length were found in the top-5 of all except one model (the multiclass task, with `nh`=3). In this model, the standard deviation of the allele frequency presented the third variable importance. Variables describing the allele frequency were also present in all except one LP/MGE models. Variables related to the node degree were found important in two models: the models for LP/MGE prediction with `nh`=3 and with all `nh` values together.

4.3.4 Label prediction

Prediction performances on the test datasets

We computed the accuracy of the predictions obtained on the real and simulated test datasets with the selected models. Table 4.3 summarises these performances.

Table 4.3: Accuracy obtained on the test datasets, for both classification tasks

	LP/MGE		Multiclass	
`nh`	real graphs	simulated graphs	real graphs	simulated graphs
1	0.93	0.95	0.73	0.93
2	0.78	0.96	0.68	0.95
3	0.92	0.97	0.70	0.96
4	0.97	0.98	0.79	0.96
5	0.91	0.97	0.70	0.95
all `nh`	0.93	0.97	0.75	0.94

The performance obtained on the simulated test dataset were close to the one obtained in cross-validation. However, the performances obtained on the real test dataset were lower, specially for the multiclass task, for which the best accuracy was 0.79, obtained with `nh`=4. In this case, errors occurred mostly from graphs labelled 'LP single': 58% (14) of these graphs were correctly predicted and 29% (7) were predicted as 'LP multiple', as shown in Table 4.4.

Table 4.4: Contingency table obtained for the multiclass classifier on the real test dataset, for `nh`=4

		predicted			
		LP multiple	LP single	MGE deletion	MGE insertion
truth	LP multiple	10	3	0	0
	LP single	7	14	**2**	**1**
	MGE deletion	0	0	0	0
	MGE insertion	0	0	0	24

The worst performances were obtained for `nh`=2, whatever the task. In this case, errors happened exclusively from the 'LP' to the 'MGE' class for the LP/MGE classifier, as shown in contingency Table 4.5. For the multiclass classifier, errors occurred mainly from graphs labelled 'LP single': only 41% (11) of these graphs were correctly predicted and 26% (7) were predicted as 'MGE deletion', as shown in Table 4.6.
The errors obtained when predicting the simulated test dataset were more balanced between the classes. For instance, for the multiclass task with `nh`=2, there were 4 errors within the 'LP' subclasses and 2 errors within the 'MGE' subclasses, and there were

Table 4.5: Contingency table obtained for the LP/MGE classifier on the real test dataset, for `nh`=2

		predicted	
		LP	MGE
truth	LP	25	**14**
	MGE	0	24

Table 4.6: Contingency table obtained for the multiclass classifier on the real test dataset, for `nh`=2

		predicted			
		LP multiple	LP single	MGE deletion	MGE insertion
truth	LP multiple	8	4	0	0
	LP single	5	11	**7**	**4**
	MGE deletion	0	0	0	0
	MGE insertion	0	0	0	24

9 (resp. 11) errors between the classes: from both 'LP' subcategories to both 'MGE' subcategories (resp. from 'MGE' to 'LP' subcategories), as shown in Table 4.7.

Table 4.7: Contingency table obtained for the multiclass classifier on the simulated test dataset, for `nh`=2

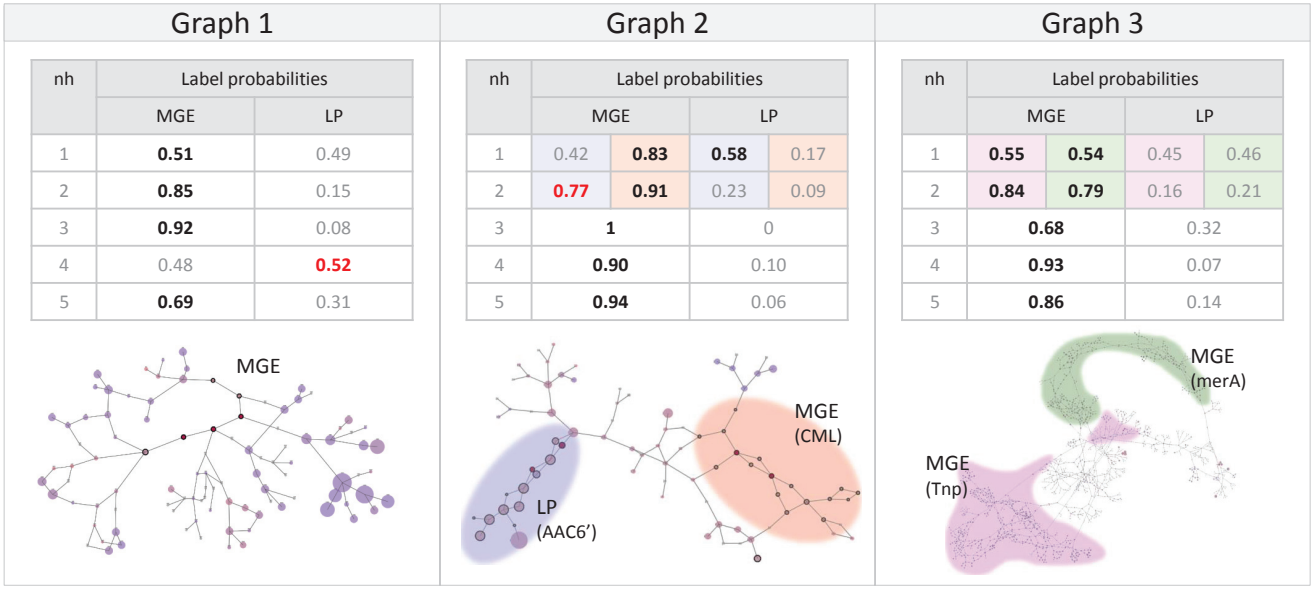
		predicted			
		LP multiple	LP single	MGE deletion	MGE insertion
truth	LP multiple	96	2	**1**	**5**
	LP single	2	118	**1**	**2**
	MGE deletion	**1**	0	45	2
	MGE insertion	**5**	**5**	0	211

Label prediction for graphs from Table 3.1 of Chapter 3

We ran the selected `nh5` models for the LP/MGE and multiclass classifications in order to predict the labels of the graphs presented in Chapter 3, Table 3.1. The results, presented in Table 4.8 showed a misclassification rate in concordance with the accuracy values presented in Table 4.3. Note that only errors between 'LP' and 'MGE' classes were highlighted in red for the multiclass classification. All incorrectly LP/MGE predicted labels obtained a probability below 0.90, except for the amikacin *aac(6')* graph. In this case, as presented in Fig. 4.1B, two variants were represented, and the 'LPG' label provided in Table 3.1 concerned only the region presenting the lowest q-value, and representing a SNP in *aac(6')* gene, while the complete graph represented the insertion of the CML gene in addition. When considering the more detailed labels from Supplementary materials S3.2, S3.3 and S3.4, all graphs labelled either 'polymorphic region in a gene' or 'pattern in a noncoding region' (purple labels) were predicted with a probability < 0.90. These labels were attributed to complex graphs, often presenting a path of red or blue nodes in a branching environment, such as the graph presented in Fig. 4.1A. This illustrates a limitation of our approach focusing only on easy graphs and not able to predict correctly these more complex graphs. While it would be useful to investigate further and describe better the type of variants represented by these graphs for a better prediction of their label, providing the probability associated to the majority class could already be useful to DBGWAS users: when the probability is < 0.90, DBGWAS is unable to predict an LP/MGE label but informs on the graph complexity. In these case, the user could re-run the third step with a lower `SFF` or `nh` value.

Table 4.8: **Labels predicted for graphs from Table 3.1**. Binary and multiclass models selected for nh=5 were used to predict known labels. The three antibiotics included in the train datasets are shown with an asterisk. Prediction errors at the LP/MGE levels are highlighted in red. Probabilities associated to the predicted label are added in parenthesis.

Panel	Phenotype	Rank	Annotation	Label Tab.3.1	Label Suppl. S3.2-S3.4	Pred. LP/MGE (prob)	Pred. subclass (prob)
SA	Methicillin	1	*mecA*+ 7000 bp of SCC*mec*	MGE	MGE with gene annotation	MGE (1)	MGE insertion (.97)
		2	6000 bp of SCC*mec*	MGE	MGE	MGE (1)	MGE insertion (1)
		3	2000 bp of SCC*mec*	MGE	MGE	MGE (1)	MGE insertion (1)
		4	1500 bp of SCC*mec*	MGE	MGE	MGE (.91)	MGE insertion (.90)
	Ciprofloxacin *	1	*parC* QRDR	LPG	hot-spot in a core gene	LP (1)	LP single (.95)
		2	*gyrA* QRDR	LPG	hot-spot in a core gene	LP (1)	LP single (.95)
	Erythromycin	1	*ermC* + circular plasmid	MGE	MGE with gene annotation	MGE (1)	MGE insertion (1)
	Fusidic acid	1	*fusA*	LPG	polymorphic region in a gene	**MGE (.50)**	**MGE deletion (.55)**
		2	*fusC* + SCC*fusC* cassette	MGE	MGE with gene annotation	MGE (1)	MGE insertion (1)
		3	1,500 bp of SCCfusC	MGE	MGE	MGE (1)	MGE insertion (1)
		3	200 bp of SCC*fusC*	MGE	MGE	MGE (.88)	MGE insertion (.76)
		5	*purN*	LPG	polymorphic region in a gene	**MGE (.73)**	**MGE deletion (.68)**
	Trimethoprim	1	*folA*	LPG	SNP in a gene	LP (.98)	LP single (.57)
		2	btw. hyp. prot. & VOC prot.	LPN	pattern in a noncoding region	LP (.74)	**MGE deletion (.53)**
		3	*ybaK*	LPG	polymorphic region in a gene	**MGE (.71)**	**MGE deletion (.89)**
		4	*mqo1*	LPG	polymorphic region in a gene	LP (.76)	LP single (.33)
	Gentamicin *	1	*aac(6')* gene within a plasmid	MGE	MGE with gene annotation	MGE (.99)	MGE insertion (1)
		2	seq. of plasmid carrying *aac(6')*	MGE	MGE	MGE (1)	MGE insertion (1)
		3	seq. of plasmid carrying *aac(6')*	MGE	MGE	MGE (1)	MGE insertion (1)
		4	seq. of plasmid carrying *aac(6')*	MGE	MGE	MGE (.99)	MGE insertion (.99)
		5	*odhB*	LPG	polymorphic region in a gene	LP (.88)	LP single (.90)
TB	Rifampicin	1	*rpoB* RRDR	LPG	hot-spot in a core gene	LP (1)	LP multiple (.98)
		2	*katG*	LPG	SNP in a core gene	LP (1)	LP single (.93)
		3	*embB* M306V	LPG	SNP in a core gene	LP (.93)	LP single (.51)
	Streptomycin	1	*rpsL* (30S ribos.protein S12)	LPG	SNP in a core gene	LP (.94)	LP single (.45)
		2	*katG*	LPG	SNP in a core gene	LP (1)	LP single (.81)
		3	*rpoB* RRDR	LPG	hot-spot in a core gene	LP (1)	LP multiple (.91)
		4	*embB* M306V	LPG	SNP in a core gene	LP (.94)	LP single (.47)
		5	*rrs*, 16S rRNA C517T	LPG	SNP in a core gene	LP (.88)	LP multiple (.42)
		6	*gyrA* QRDR	LPG	hot-spot in a core gene	LP (.97)	LP multiple (.84)
		7	*espG1*	LPG	SNPs in a core gene	LP (.95)	LP single (.73)
	Ofloxacin	1	*gyrA* QRDR	LPG	hot-spot in a core gene	LP (1)	LP multiple (.91)
		2	*ubiA* (Rv3806c)	LPG	SNP in a core gene	LP (1)	LP multiple (.64)
		3	Rv3909	LPG	SNP in a core gene	**MGE (.88)**	**MGE deletion (.60)**
	Ethionamide	1	*fabG1* promoter	LPN	SNP in a core gene promoter	LP (.92)	LP multiple (.76)
		2	*gyrA* QRDR	LPG	hot-spot in a core gene	LP (1)	LP multiple (.81)
		3	*rrs*, 16S rRNA A1401G	LPG	SNP in a core gene	LP (.90)	LP single (.63)
	XDR *	1	*rpoB* I1187T (out. RRDR)	LPG	SNP in a core gene	LP (1)	LP single (.99)
		1	Rv2000	LPG	SNP in a core gene	LP (1)	LP multiple (.80)
		3	*espA* promoter	LPN	SNP in a core gene promoter	LP (1)	LP single (.97)
PA	Amikacin	1	*aac(6')*	LPG	SNP in an accessory gene + accessory gene	**MGE (.94)**	**MGE insertion (.97)**
		2	DEAD/DEAH box helicase	LPG	polymorphic region in a gene	**MGE (.74)**	**MGE insertion (.53)**
		3	plasmid mapping on pHS87b	MGE	MGE	MGE (1)	MGE insertion (1)
	Levofloxacin	1	*gyrA* QRDR	LPG	hot-spot in a core gene	LP (.88)	LP single (.65)
		2	*parC* QRDR	LPG	hot-spot in a core gene	LP (1)	LP single (.91)
		3	Hist. kinase/resp. regulator	LPG	SNP in a core gene	LP (.96)	LP single (.84)



Graph 1		
nh	Label probabilities	
	MGE	LP
1	**0.51**	0.49
2	**0.85**	0.15
3	**0.92**	0.08
4	0.48	**0.52**
5	**0.69**	0.31

Graph 2				
nh	Label probabilities			
	MGE		LP	
1	0.42	**0.83**	**0.58**	0.17
2	**0.77**	**0.91**	0.23	0.09
3	**1**		0	
4	**0.90**		0.10	
5	**0.94**		0.06	

Graph 3				
nh	Label probabilities			
	MGE		LP	
1	**0.55**	**0.54**	0.45	0.46
2	**0.84**	**0.79**	0.16	0.21
3	**0.68**		0.32	
4	**0.93**		0.07	
5	**0.86**		0.14	

Figure 4.7: **Labels predicted to complex graphs, for several values of `nh`.** This figure presents 3 complex graphs generated by DBGWAs, for which predicted labels and associated probabilities are shown, for 5 values of `nh`. Sub-regions in the graphs are colour-coded. Indeed, for low values of `nh`, the graph is split into several graphs describing each sub-region.

Complex graphs and `nh`-based label prediction

We compared the predicted labels obtained with several values of `nh`, for the three complex graphs presented in Fig. 4.1. Indeed, our idea was to test the feasibility of an automated selection of the best `nh` value, based on the idea that an optimal value would allow for the best prediction. In particular, a low `nh` value would split complex graphs representing several regions into several each of them being easier to interpret.
Fig. 4.7 presents the probabilities associated to each label and highlights the predicted labels (highest probability) in bold. For low `nh` values, when the initial graph is split into two graphs representing two distinct regions, the colour-coded cell background indicates to which region the prediction is related.
For instance, Graph 2 represents, at `nh`=5 a complex graph with two regions, one annotated *aac(6')* and one CML. A single graph includes both regions for `nh`>2, however both regions are represented in separated graphs for `nh`≤ 2, and in this case, separate predictions are done for each graph.
The labels predicted with `nh`=1 were all correct, which would provide a valuable information specially to interpret the SNP in the *aac(6')* gene, unfortunately the probabilities associated to the labels were very close to 0.50, and the labels therefore could not be trusted enough to automate such an `nh` value optimisation strategy.

4.4 Discussion and concluding remarks

In this Chapter, we developed a strategy to predict the type of the genetic variants represented in DBGWAS graphs. Our objective was indeed to help in the interpretation of the graphs generated by the tool. We built a dataset composed mostly of simulated graphs. The simulations aimed at producing graphs representing easy cases: local polymorphisms in a gene and long insertion or deletion of a gene, with or without MGE (plasmid) context. We were particularly interested by the `nh` value which defines the size of the neighbourhood

retained around the unitigs significantly associated to the phenotype to build the graphs. Indeed, tuning this value can help to lower the complexity of graphs gathering several polymorphic genes. We built one model per `nh` value, however, the model trained on all graphs (with all `nh` values together) showed equal performance in average: a single model is sufficient to predict the labels whatever the `nh` value used to generate the graph.
The LP/MGE label prediction performances were very good on simulated graphs (accuracy around 0.97), regardless of the `nh` value. The performances on real graphs were also pretty high (around 0.92), nonetheless, the gap in performance between simulated and real data was large when predicting a finer label with the multiclass classifier. The low performances obtained on the real dataset (around 0.72) can be explained by the limited scope of the simulations focusing on 'easy' graphs, together with the low percent of real graphs included in the training dataset. However, obtaining trustworthy labels for real graphs implies manual investigations and is highly time-consuming.
In order to improve the training dataset composition, a particular attention should be given to the real 'LP single' graphs which were wrongly labelled. Better understanding difficulties of classification would help design new simulation strategies to enrich the current set. We could focus on the graphs for which low probabilities (< 0.90) were obtained, in particular graphs labelled as patterns or polymorphic regions (as in Supplementary materials S3.2, S3.3 and S3.4). They might represent a type of variant we did not consider or at least which cannot be summarised under either the general LP or the MGE labels. This dataset could be enriched with other labels representing for instance rearrangements, or mosaic genes.
The prediction performances could also be improved by considering other modelling strategies. In our approach, we used descriptors at the subgraph level, and summarised the unitig-level information. Graph kernel methods were developed to mine graph data, based on walk, subtrees, paths, or cycles in the graphs [27, 75]. They allow a finer exploration of the graph topologies, and could improve the prediction. Deep graph kernels aims at detect sub-structures in graphs [216], which could be useful to detect multiple variants within a single graphs (such as Graph 2 of Fig. 4.1B, which contains a SNPs in *aac(6')* gene and the insertion of the CML gene) and by predicting the labels locally. Hierarchical strategies could also be set in order to take into account the hierarchy between LP/MGE label and their sub-labels predicted in the multiclass task. Other sub-labels could also be considered.
In the meantime, the results presented in this Chapter could lead to short-term improvements of DBGWAS. First, the probability associated to the predicted label provides information which could be valuable for DBGWAS users. If an LP/MGE label can be predicted with high probability (> 0.90), specific additional features could be computed, such as an alignments between 'positive' and 'negative' contigs for LP, which would highlight the position of the mutations in the sequence. A sub-label could also be provided when the LP/MGE label is well predicted. And second, if the prediction performance is a proxy for the interpretability of a graph, this work could suggest considering `nh`=4 as a default value instead of `nh`=5, as the best performances were obtained with this `nh` value, for real and simulated datasets.

Conclusions, discussions and perspectives

In this thesis, we were interested by the application of GWAS to bacterial genomes, for the study of antibiotic resistance. Our main species model was *P. aeruginosa*, which has a large plastic genome. It makes great use of its accessory genome and regulatory pathways, which may involve subtle resistance mechanisms not well described. The prediction of the amikacin resistance status for instance cannot be based only on the presence of the genuine markers such as aminoglycoside-modifying enzymes or MexXY–OprM efflux system [105]. This complexity motivated us to work on the adaptation of GWAS tools to the high plasticity of bacterial genomes, mainly by working on the representation of the genetic variations in these genomes.

We evaluated several methods to model the relationship between the phenotype and the genotype, including different measures of the resistance, and different adjustments for population structure. From these evaluations, we selected the R package `bugwas` [61], based on linear mixed models that offered in our experiments the best compromise for different configurations of population structures in *P. aeruginosa*. We aimed at widening the search space compared to our first RWAS study, in order to screen for variations in the core as well as the accessory genome, without missing noncoding regulatory regions in bacterial genomes such as *P. aeruginosa* which are particularly plastic. We focused our search on k-mers as they offer the necessary flexibility but they are very redundant and do not allow for a direct interpretation of the results. We then searched how to take advantage of a cDBG for variant representation. Bubbles captured simple variations but not polymorphic insertions. The method we developed tests the presence or absence of unitigs: the nodes of a cDBG built from all input genomes. We focused our efforts on post-processing, and proposed a visualisation of the genomic context of the unitigs associated with the phenotype. This synthetic view allows to understand the sequence of interest in its context and informs about its nature: *e.g.* a local polymorphism, or the acquisition of a gene within a plasmid. We published DBGWAS, a turnkey tool implementing efficiently the complete analysis. It only requires a set of genome pre-assemblies and corresponding phenotypes to generate in a few hours (one hour and a half in average) ordered, decorated subgraphs representing sequences associated with the antibiotic resistance. These subgraphs are in most cases good proxy of a particular genomic region such as a polymorphic region in a gene or the insertion of a plasmid carrying a resistance gene. We were able to correctly predict the type of genetic variation represented by the subgraph (local polymorphism or MGE) in 92% of the cases. However the performance was lower to predict subtypes or complex branching subgraphs.

DBGWAS could benefit from this work as a short-term improvement. The classifier could be improved by using finer methods such a graph kernel-based methods, able to take into account the graph topology, locally or globally. In the short-term, DBGWAS could also be extended to offer other association models than the one implemented by `bugwas`, *e.g.* models allowing for continuous or ordinal phenotypes. Another quick win would be to add an LD measure within subgraphs (*.i.e* among their significant unitigs) or between

subgraphs (for instance between the most significant unitigs of each subgraph).
DBGWAS unitig-based strategy could be reconsidered to offer a control of the FDR at the subgraph level. A two-step approach could be envisaged, for instance selecting significant unitigs first, and then testing the association of their induced subgraph with the phenotype. For this second step, methods testing the combined presence of close variants, such as the intervals of genetic heterogeneity [127], could help to report a genotype measure at the subgraph level: a genetic interval is represented by a presence/absence pattern of at least one of the SNP within the interval. It is computed using an OR operator between all SNP presence/absence pattern. This approach would need to be adapted to fit large subgraphs, in which such operation would rapidly lead to patterns containing only presences. Another approach could be a supervised construction of a pattern graph, where any bubble, whose pair of paths is not differentially observed between the phenotypes, would be replaced by a linear path with a 'N' instead of the variable part of its pair of paths. However, this approach, like the previous one, would result in inferring on data already used for a first selection, and would raise post-selective inference issues.
The tool we set up has already been successfully tested to inspect another bacterial phenotype, the natural transformation in *Legionella pneumophila* [60]and will be soon tested on non-model eukaryote species to retrieve sex-related markers. Our tool was presented to pharma who had expressed their need to better understand the genetic impacts of the new molecules candidates to become our future antibiotics. The DBGWAS subgraph representation could also be applied to interpret k-mer-based models built to predict an antibiotic resistance status [58, 193].
Bacterial GWAS will certainly benefit from our DBGWAS tool, however the investigation of antibiotic resistance mechanisms and their genetic basis still requires more general improvements. More efforts can be done to screen the complete genomic variations: first, in our approach, we do not address the copy number variations (CNV), whereas the number of plasmids in a cell, for instance, can have a direct relationship to the cell resistance level. DBGWAS could be adapted to take as input raw reads and thus retrieve this information. Second, DBGWAS does not address co-resistance or cross-resistance: it analyses the phenotypes one by one. Approaches modelling several antibiotic responses could help find co-markers and deal with the observed correlation between phenotypes [226]. Third, we do not address interactions between variants, which can be driven by epistasis [82, 143], nor do we address potential additive effects since we test the variants individually. The genome is not expected to carry alone all information on the variations causing antibiotic resistance: post-genomics could bring missing clues. Bacteria are subject to epigenetics: DNA methylation was reported to affect virulence gene regulation [12, 41]. RNA expression level can be potentially marker of antibiotic resistance [13, 102]. Post-translational mutations may also arise and help the cell adapt and diversify [38, 80]. Bacterial GWAS studying the antibiotic resistance would benefit from advances in multi-omics modelling approaches [28]. Resistance determinants identified by current GWAS methods are markers of *in vitro* resistance. The clinical expression of resistance is not always directly reflected by these markers [196]. Taking into account the host response, for instance by integrating the host genome or transcriptome as covariates, could model host-pathogen interactions and open a path to personalised antibiotic therapy [131]. Finally, strain panels have to be carefully prepared, to assure balanced phenotypes and spread among the species, and need to be regularly updated. Indeed, no GWAS computed from 'old' strains (before 2010) would be able to find the recently appeared *mcr*-1 colistin marker.
The fight against antibiotic resistance and multi-resistance will be long and difficult. Each single step in this fight will count.

Bibliography

[1] 1000 Genomes Project Consortium (2010). A map of human genome variation from population-scale sequencing. *Nature*, **467**(7319), 1061.

[2] Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotugu Akaike*, pages 199–213. Springer.

[3] Alam, M. T., Petit, R. A., Crispell, E. K., Thornton, T. A., Conneely, K. N., Jiang, Y., *et al.* (2014). Dissecting vancomycin-intermediate resistance in *Staphylococcus aureus* using genome-wide association. *Genome biology and evolution*, **6**(5), 1174–1185.

[4] Alekshun, M. N. and Levy, S. B. (2007). Molecular mechanisms of antibacterial multidrug resistance. *Cell*, **128**(6), 1037–1050.

[5] Ali-Ahmad, A., Fadel, F., Sebban-Kreuzer, C., Ba, M., Pélissier, G. D., Bornet, O., *et al.* (2017). Structural and functional insights into the periplasmic detector domain of the GacS histidine kinase controlling biofilm formation in *Pseudomonas aeruginosa*. *Scientific reports*, **7**(1), 11262.

[6] Amsler, K., Santoro, C., Foleno, B., Bush, K., and Flamm, R. (2010). Comparison of broth microdilution, agar dilution, and Etest for susceptibility testing of doripenem against Gram-negative and Gram-positive pathogens. *Journal of clinical microbiology*, **48**(9), 3353–3357.

[7] Andersson, D. I. and Hughes, D. (2011). Persistence of antibiotic resistance in bacterial populations. *FEMS microbiology reviews*, **35**(5), 901–911.

[8] Andrews, J. M. (2001). Determination of minimum inhibitory concentrations. *Journal of antimicrobial Chemotherapy*, **48**(suppl_1), 5–16.

[9] Aubert, D., Poirel, L., Chevalier, J., Leotard, S., Pages, J.-M., and Nordmann, P., *et al.* (2001). Oxacillinase-mediated resistance to cefepime and susceptibility to ceftazidime in *Pseudomonas aeruginosa*. *Antimicrobial agents and chemotherapy*, **45**(6), 1615–1620.

[10] Baaijens, J. A., El Aabidine, A. Z., Rivals, E., and Schönhuth, A. (2017). *De novo* assembly of viral quasispecies using overlap graphs. *Genome research*, **27**(5), 835–848.

[11] Balding, D. J. (2006). A tutorial on statistical methods for population association studies. *Nature reviews genetics*, **7**(10), 781–791.

[12] Banas, J. A., Biswas, S., and Zhu, M. (2011). DNA methylation affects virulence gene expression in *Streptococcus mutans*. *Applied and environmental microbiology*, pages AEM–00543.

[13] Barczak, A. K., Gomez, J. E., Kaufmann, B. B., Hinson, E. R., Cosimi, L., Borowsky, M. L., *et al.* (2012). RNA signatures allow rapid identification of pathogens and antibiotic susceptibilities. *Proceedings of the national academy of sciences*, page 201119540.

[14] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300.

[15] Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188.

[16] Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., *et al.* (2012). Genbank. *Nucleic acids research*, **41**(D1), D36–D42.

[17] Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J., Brown, C. G., *et al.* (2008). Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, **456**(7218), 53.

[18] Berk, R., Brown, L., Buja, A., Zhang, K., Zhao, L., *et al.*, *et al.* (2013). Valid post-selection inference. *The annals of statistics*, **41**(2), 802–837.

[19] Bernard, E., Jacob, L., Mairal, J., and Vert, J.-P. (2014). Efficient RNA isoform identification and quantification from RNA-Seq data with network flows. *Bioinformatics*, **30**(17), 2447–2455.

[20] Bi, D., Xie, Y., Tai, C., Jiang, X., Zhang, J., Harrison, E. M., *et al.* (2016). A site-specific integrative plasmid found in *Pseudomonas aeruginosa* clinical isolate HS87 along with a plasmid carrying an aminoglycoside-resistant gene. *PloS one*, **11**(2), e0148367.

[21] Birol, I., Jackman, S. D., Nielsen, C. B., Qian, J. Q., Varhol, R., Stazyk, G., *et al.* (2009). *De novo* transcriptome assembly with ABySS. *Bioinformatics*, **25**(21), 2872–2877.

[22] Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., *et al.* (2016). `mlr`: machine learning in R. *Journal of machine learning research*, **17**(170), 1–5.

[23] Bishop, C. M. (2006). *Pattern recognition and machine learning*, volume 1. Springer Information Science and Statistics.

[24] Blair, J. M., Webber, M. A., Baylay, A. J., Ogbolu, D. O., and Piddock, L. J. (2015). Molecular mechanisms of antibiotic resistance. *Nature reviews microbiology*, **13**(1), 42–51.

[25] Blanchard, G., Neuvial, P., and Roquain, E. (2017). Post hoc inference via joint family-wise error rate control. *arXiv preprint arXiv:1703.02307*.

[26] Bland, J. M. and Altman, D. G. (1995). Multiple significance tests: the Bonferroni method. *Bmj*, **310**(6973), 170.

[27] Borgwardt, K. M. and Kriegel, H.-P. (2005). Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining*, pages 8–pp. IEEE.

[28] Boulesteix, A.-L., De Bin, R., Jiang, X., and Fuchs, M. (2017). Ipf-lasso: Integrative l_1-penalized regression with penalty factors for prediction based on multi-omics data. *Computational and mathematical methods in medicine*, **2017**.

[29] Bradley, P., Gordon, N. C., Walker, T. M., Dunn, L., Heys, S., Huang, B., *et al.* (2015). Rapid antibiotic-resistance predictions from genome sequence data for *Staphylococcus aureus* and *Mycobacterium tuberculosis*. *Nature communications*, **6**, 10063.

[30] Brandenburg, F. J. (1999). Drawing decorated graphs.

[31] Branton, D., Deamer, D. W., Marziali, A., Bayley, H., Benner, S. A., Butler, T., *et al.* (2010). The potential and challenges of nanopore sequencing. In *Nanoscience and technology: a collection of reviews from Nature journals*, pages 261–268. World Scientific.

[32] Breiman, L. (2001). Random forests. *Machine learning*, **45**(1), 5–32.

[33] Bush, W. S. and Moore, J. H. (2012). Genome-wide association studies. *PLoS computational biology*, **8**(12), e1002822.

[34] Butler, J., MacCallum, I., Kleber, M., Shlyakhter, I. A., Belmonte, M. K., Lander, E. S., *et al.* (2008). ALLPATHS: *de novo* assembly of whole-genome shotgun microreads. *Genome research*, **18**(5), 810–820.

[35] Buzdugan, L., Kalisch, M., Navarro, A., Schunk, D., Fehr, E., and Bühlmann, P., *et al.* (2016). Assessing statistical significance in multivariable genome wide association analysis. *Bioinformatics*, **32**(13), 1990–2000.

[36] Cabot, G., Ocampo-Sosa, A. A., Domínguez, M. A., Gago, J. F., Juan, C., Tubau, F., *et al.* (2012). Genetic markers of widespread extensively drug-resistant (XDR) *Pseudomonas aeruginosa* high-risk clones. *Antimicrobial agents and chemotherapy*, pages AAC–01388.

[37] Cabot, G., López-Causapé, C., Ocampo-Sosa, A. A., Sommer, L. M., Domínguez, M. Á., Zamorano, L., *et al.* (2016). Deciphering the resistome of the widespread *Pseudomonas aeruginosa* sequence type 175 international high-risk clone through whole-genome sequencing. *Antimicrobial agents and chemotherapy*, **60**(12), 7415–7423.

[38] Cain, J. A., Solis, N., and Cordwell, S. J. (2014). Beyond gene expression: the impact of protein post-translational modifications in bacteria. *Journal of proteomics*, **97**, 265–286.

[39] Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., *et al.* (2009). BLAST+: architecture and applications. *BMC bioinformatics*, **10**(1), 421.

[40] Carattoli, A., Zankari, E., García-Fernández, A., Larsen, M. V., Lund, O., Villa, L., *et al.* (2014). *In silico* detection and typing of plasmids using PlasmidFinder and plasmid multilocus sequence typing. *Antimicrobial agents and chemotherapy*, **58**(7), 3895–3903.

[41] Casadesús, J. and Low, D. A. (2013). Programmed heterogeneity: epigenetic mechanisms in bacteria. *Journal of biological chemistry*, **288**(20), 13929–13935.

[42] Chang, C. C., Chow, C. C., Tellier, L. C., Vattikuti, S., Purcell, S. M., and Lee, J. J., *et al.* (2015). Second-generation PLINK: rising to the challenge of larger and richer datasets. *Gigascience*, **4**(1), 7.

[43] Chewapreecha, C., Marttinen, P., Croucher, N. J., Salter, S. J., Harris, S. R., Mather, A. E., *et al.* (2014). Comprehensive identification of single nucleotide polymorphisms associated with beta-lactam resistance within pneumococcal mosaic genes. *PLoS genetics*, **10**(8), e1004547.

[44] Chikhi, R., Limasset, A., and Medvedev, P. (2016). Compacting de Bruijn graphs from sequencing data quickly and in low memory. *Bioinformatics*, **32**(12), i201 – i208.

[45] Cho, H. H., Kwon, K. C., Kim, S., and Koo, S. H. (2014). Correlation between virulence genotype and fluoroquinolone resistance in carbapenem-resistant *Pseudomonas aeruginosa*. *Annals of laboratory medicine*, **34**(4), 286–292.

[46] Chowdhury, P. R., Scott, M., Worden, P., Huntington, P., Hudson, B., Karagiannis, T., *et al.* (2016). Genomic islands 1 and 2 play key roles in the evolution of extensively drug-resistant ST235 isolates of *Pseudomonas aeruginosa*. *Open biology*, **6**(3), 150175.

[47] Coll, F., McNerney, R., Preston, M. D., Guerra-Assunção, J. A., Warry, A., Hill-Cawthorne, G., *et al.* (2015). Rapid determination of anti-tuberculosis drug resistance from whole-genome sequences. *Genome medicine*, **7**(1), 51.

[48] Collins, C. and Didelot, X. (2018). A phylogenetic method to perform genome-wide association studies in microbes that accounts for population structure and recombination. *PLOS Computational Biology*, **14**(2), 1–21.

[49] Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the royal statistical society. Series B (methodological)*, pages 215–242.

[50] Cule, E., Vineis, P., and De Iorio, M. (2011). Significance testing in ridge regression for genetic data. *BMC bioinformatics*, **12**(1), 372.

[51] Davis, J. J., Boisvert, S., Brettin, T., Kenyon, R. W., Mao, C., Olson, R., *et al.* (2016). Antimicrobial resistance prediction in PATRIC and RAST. *Scientific reports*, **6**, 27930.

[52] de Bruijn, N. (1946). A combinatorial problem. *Proceedings of the koninklijke nederlandse akademie van wetenschappen. Series A*, **49**(7), 758.

[53] de Jong, E., van Oers, J. A., Beishuizen, A., Vos, P., Vermeijden, W. J., Haas, L. E., *et al.* (2016). Efficacy and safety of procalcitonin guidance in reducing the duration of antibiotic treatment in critically ill patients: a randomised, controlled, open-label trial. *The Lancet Infectious Diseases*, **16**(7), 819–827.

[54] Dehman, A., Ambroise, C., and Neuvial, P. (2015). Performance of a blockwise approach in variable selection using linkage disequilibrium information. *BMC bioinformatics*, **16**(1), 148.

[55] Devlin, B. and Roeder, K. (1999). Genomic control for association studies. *Biometrics*, **55**(4), 997–1004.

[56] Didelot, X., Bowden, R., Wilson, D. J., Peto, T. E., and Crook, D. W. (2012). Transforming clinical microbiology with bacterial genome sequencing. *Nature reviews genetics*, **13**(9), 601.

[57] Drezen, E., Rizk, G., Chikhi, R., Deltel, C., Lemaitre, C., Peterlongo, P., *et al.* (2014). GATB: genome assembly & analysis tool box. *Bioinformatics*, **30**(20), 2959–2961.

[58] Drouin, A., Giguère, S., Déraspe, M., Laviolette, F., Marchand, M., and Corbeil, J., *et al.* (2015). Greedy biomarker discovery in the genome with applications to antimicrobial resistance. *arXiv doi:1505.06249*.

[59] Dunne Jr, W. M., Jaillard, M., Rochas, O., and Van Belkum, A. (2017). Microbial genomics and antimicrobial susceptibility testing. *Expert review of molecular diagnostics*, **17**(3), 257–269.

[60] Durieux, I., Ginevra, C., Picq, K., Attaiech, L., Juan, P.-A., Jarraud, S., *et al.* (2018). Widespread natural transformation in *Legionella pneumophila* clinical isolates. In *Proceedings of the 5th ESCMID study group for Legionella infections conference*, number I.09, page 22. ESCMID.

[61] Earle, S. G., Wu, C.-H., Charlesworth, J., Stoesser, N., Gordon, N. C., Walker, T. M., *et al.* (2016). Identifying lineage effects when controlling for population structure improves power in bacterial association studies. *Nature microbiology*, page 16041.

[62] Eyre, D. W., De Silva, D., Cole, K., Peters, J., Cole, M. J., Grad, Y. H., *et al.* (2017). WGS to predict antibiotic mics for *Neisseria gonorrhoeae*. *Journal of antimicrobial chemotherapy*, **72**(7), 1937–1947.

[63] Falush, D. and Bowden, R. (2006). Genome-wide association mapping in bacteria? *Trends in microbiology*, **14**(8), 353–355.

[64] Farhat, M. R., Shapiro, B. J., Kieser, K. J., Sultana, R., Jacobson, K. R., Victor, T. C., *et al.* (2013). Genomic analysis identifies targets of convergent positive selection in drug-resistant *Mycobacterium tuberculosis*. *Nature genetics*, **45**(10), 1183–1189.

[65] Farhat, M. R., Sultana, R., Iartchouk, O., Bozeman, S., Galagan, J., Sisk, P., *et al.* (2016). Genetic determinants of drug resistance in *Mycobacterium tuberculosis* and their diagnostic value. *American journal of respiratory and critical care medicine*, **194**(5), 621–630.

[66] Flandrois, J.-P., Lina, G., and Dumitrescu, O. (2014). MUBII-TB-DB: a database of mutations associated with antibiotic resistance in *Mycobacterium tuberculosis*. *BMC bioinformatics*, **15**(1), 107.

[67] Franz, M., Lopes, C. T., Huck, G., Dong, Y., Sumer, O., and Bader, G. D., *et al.* (2015). Cytoscape.js: a graph theory library for visualisation and analysis. *Bioinformatics*, **32**(2), 309–311.

[68] Freschi, L., Jeukens, J., Kukavica-Ibrulj, I., Boyle, B., Dupont, M.-J., Laroche, J., *et al.* (2015). Clinical utilization of genomics data produced by the international *Pseudomonas aeruginosa* consortium. *Frontiers in microbiology*, **6**, 1036.

[69] Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:.

[70] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, **33**, 1–22.

[71] Fuchs, P., Barry, A., and Brown, S. (1997). Susceptibility testing quality control studies with fosfomycin tromethamine. *European journal of clinical microbiology and infectious diseases*, **16**(7), 538–540.

[72] Galardini, M., Biondi, E. G., Bazzicalupo, M., and Mengoni, A. (2011). Contiguator: a bacterial genomes finishing tool for structural insights on draft genomes. *Source code for biology and medicine*, **6**(1), 11.

[73] Gardner, S. N., Slezak, T., and Hall, B. G. (2015). kSNP 3.0: SNP detection and phylogenetic analysis of genomes without genome alignment or reference genome. *Bioinformatics*, **31**(17), 2877–2878.

[74] Garrison, E., Sirén, J., Novak, A. M., Hickey, G., Eizenga, J. M., Dawson, E. T., *et al.* (2018). Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature biotechnology doi: 10.1038/nbt.4227*.

[75] Gärtner, T., Flach, P., and Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In *Learning theory and kernel machines*, pages 129–143. Springer.

[76] Gillings, M. R. (2014). Integrons: past, present, and future. *Microbiology and molecular biology reviews*, **78**(2), 257–277.

[77] Goeman, J. J. (2010). $l1$ penalized estimation in the Cox proportional hazards model. *Biometrical journal*, **52**(1), 70–84.

[78] Goeman, J. J. and Solari, A. (2014). Multiple hypothesis testing in genomics. *Statistics in medicine*, **33**(11), 1946–1978.

[79] Gordon, N., Price, J., Cole, K., Everitt, R., Morgan, M., Finney, J., *et al.* (2014). Prediction of *Staphylococcus aureus* antimicrobial resistance by whole-genome sequencing. *Journal of clinical microbiology*, **52**(4), 1182–1191.

[80] Grangeasse, C., Stülke, J., and Mijakovic, I. (2015). Regulatory potential of post-translational modifications in bacteria. *Frontiers in microbiology*, **6**, 500.

[81] Gupta, S. K., Padmanabhan, B. R., Diene, S. M., Lopez-Rojas, R., Kempf, M., Landraud, L., *et al.* (2014). ARG-ANNOT, a new bioinformatic tool to discover antibiotic resistance genes in bacterial genomes. *Antimicrobial agents and chemotherapy*, **58**(1), 212–220.

[82] Gygli, S. M., Borrell, S., Trauner, A., and Gagneux, S. (2017). Antimicrobial resistance in *Mycobacterium tuberculosis*: mechanistic and evolutionary perspectives. *FEMS microbiology reviews*, **41**(3), 354–373.

[83] Haft, D. H., DiCuccio, M., Badretdin, A., Brover, V., Chetvernin, V., O'Neill, K., *et al.* (2017). RefSeq: an update on prokaryotic genome annotation and curation. *Nucleic acids research*, **46**(D1), D851–D860.

[84] Hernando-Amado, S., Sanz-García, F., Blanco, P., and Martínez, J. L. (2017). Fitness costs associated with the acquisition of antibiotic resistance. *Essays in biochemistry*, **61**(1), 37–48.

[85] Hindler, J. A. and Humphries, R. M. (2013). Colistin MIC variability by method for contemporary clinical isolates of multidrug resistant Gram-negative bacilli. *Journal of clinical microbiology*, pages JCM–03385.

[86] Hocking, R. R. (1976). The analysis and selection of variables in linear regression. *Biometrics*, **32**(1), 1–49.

[87] Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, **12**(1), 55–67.

[88] Hoffman, G. E. (2013). Correcting for population structure and kinship using the linear mixed model: theory and extensions. *PLoS One*, **8**(10), e75707.

[89] Holmes, A. H., Moore, L. S., Sundsfjord, A., Steinbakk, M., Regmi, S., Karkey, A., *et al.* (2016). Understanding the mechanisms and drivers of antimicrobial resistance. *The Lancet*, **387**(10014), 176–187.

[90] Hooper, D. C. and Jacoby, G. A. (2015). Mechanisms of drug resistance: quinolone resistance. *Annals of the New York academy of sciences*, **1354**(1), 12–31.

[91] Illakkiam, D., Shankar, M., Ponraj, P., Rajendhran, J., and Gunasekaran, P. (2014). Genome sequencing of a mung bean plant growth promoting strain of *P. aeruginosa* with biocontrol ability. *International journal of genomics*, **2014**.

[92] International HapMap Consortium (2003). The international HapMap project. *Nature*, **426**(6968), 789.

[93] Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., and McVean, G. (2012). *De novo* assembly and genotyping of variants using colored de Bruijn graphs. *Nature genetics*, **44**(2), 226–232.

[94] IWG-SCC consortium (2009). Classification of staphylococcal cassette chromosome *mec* (SCC*mec*): guidelines for reporting novel SCC*mec* elements. *Antimicrobial agents and chemotherapy*, **53**(12), 4961–4967.

[95] Jackman, S. D., Vandervalk, B. P., Mohamadi, H., Chu, J., Yeo, S., Hammond, S. A., *et al.* (2017). ABySS 2.0: resource-efficient assembly of large genomes using a bloom filter. *Genome research*, **27**(5), 768–777.

[96] Jaillard, M. (2018). Fine mapping of antibiotic resistance determinants. *PhD thesis*, **in preparation**.

[97] Jaillard, M., Schicklin, S., Larue-Triolet, A., and Veyrieras, J.-B. (2013). A comprehensive microbial knowledge base to support the development of *in vitro* diagnostic solutions in infectious diseases. In *I-SEMANTICS*, pages 55–59.

[98] Jaillard, M., van Belkum, A., Cady, K. C., Creely, D., Shortridge, D., Blanc, B., *et al.* (2017). Correlation between phenotypic antibiotic susceptibility and the resistome in *Pseudomonas aeruginosa*. *International journal of antimicrobial agents*, **50**(2), 210–218.

[99] Jaillard, M., Lima, L., Tournoud, M., Mahé, P., van Belkum, A., Lacroix, V., *et al.* (2018). A fast and agnostic method for bacterial genome-wide association studies: bridging the gap between k-mers and genetic events. *bioRxiv*, page 297754.

[100] Kang, H. M., Zaitlen, N. A., Wade, C. M., Kirby, A., Heckerman, D., Daly, M. J., *et al.* (2008). Efficient control of population structure in model organism association mapping. *Genetics*, **178**(3), 1709–1723.

[101] Katoh, K., Misawa, K., Kuma, K.-i., and Miyata, T. (2002). MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research*, **30**(14), 3059–3066.

[102] Khaledi, A., Schniederjans, M., Pohl, S., Rainer, R., Bodenhofer, U., Xia, B., *et al.* (2016). Transcriptome profiling of antimicrobial resistance in *Pseudomonas aeruginosa*. *Antimicrobial agents and chemotherapy*, pages AAC–00075.

[103] Klein, E. Y., Van Boeckel, T. P., Martinez, E. M., Pant, S., Gandra, S., Levin, S. A., *et al.* (2018). Global increase and geographic convergence in antibiotic consumption between 2000 and 2015. *Proceedings of the national academy of sciences*, page 201717295.

[104] Koren, S. and Phillippy, A. M. (2015). One chromosome, one contig: complete microbial genomes from long-read sequencing and assembly. *Current opinion in microbiology*, **23**, 110–120.

[105] Kos, V. N., Déraspe, M., McLaughlin, R. E., Whiteaker, J. D., Roy, P. H., Alm, R. A., *et al.* (2014). The resistome of *Pseudomonas aeruginosa* in relationship to phenotypic susceptibility. *Antimicrobial agents and chemotherapy*, pages AAC–03954.

[106] Kos, V. N., McLaughlin, R. E., and Gardner, H. A. (2016). The elucidation of mechanisms of ceftazidime resistance among clinical isolates of *Pseudomonas aeruginosa* using genomic data. *Antimicrobial agents and chemotherapy*, pages AAC–03113.

[107] Köser, C. U., Holden, M. T., Ellington, M. J., Cartwright, E. J., Brown, N. M., Ogilvy-Stuart, A. L., *et al.* (2012). Rapid whole-genome sequencing for investigation of a neonatal MRSA outbreak. *New England journal of medicine*, **366**(24), 2267–2275.

[108] Krawczyk, P. S., Lipinski, L., and Dziembowski, A. (2018). PlasFlow: predicting plasmid sequences in metagenomic data using genome signatures. *Nucleic acids research*, **46**(6), e35–e35.

[109] Kung, V. L., Ozer, E. A., and Hauser, A. R. (2010). The accessory genome of *Pseudomonas aeruginosa*. *Microbiology and molecular biology reviews*, **74**(4), 621–641.

[110] Laabei, M., Recker, M., Rudkin, J. K., Aldeljawi, M., Gulay, Z., Sloan, T. J., *et al.* (2014). Predicting the virulence of MRSA from its genome sequence. *Genome research*, **24**(5), 839–849.

[111] Lakin, S. M., Dean, C., Noyes, N. R., Dettenwanger, A., Ross, A. S., Doster, E., *et al.* (2017). MEGARes: an antimicrobial resistance database for high throughput sequencing. *Nucleic acids research*, **45**(D1), D574–D580.

[112] Lambert, P. (2002). Mechanisms of antibiotic resistance in *Pseudomonas aeruginosa*. *Journal of the royal society of medicine*, **95**(Suppl 41), 22.

[113] Lambert, T., Ploy, M., and Courvalin, P. (1994). A spontaneous point mutation in the *aac(6')-Ib'* gene results in altered substrate specificity of aminoglycoside 6'-*N*-acetyltransferase of a *Pseudomonas fluorescens* strain. *FEMS microbiology letters*, **115**, 297–304.

[114] Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature methods*, **9**(4), 357–359.

[115] Le Bras, Y., Collin, O., Monjeaud, C., Lacroix, V., Rivals, É., Lemaitre, C., *et al.* (2016). Colib'read on galaxy: a tools suite dedicated to biological information extraction from raw NGS reads. *GigaScience*, **5**(1), 1.

[116] Lee, H., Cho, S., Bang, H., Lee, J., Bai, G., Kim, S., *et al.* (2000). Exclusive mutations related to isoniazid and ethionamide resistance among *Mycobacterium tuberculosis* isolates from Korea. *The international journal of tuberculosis and lung disease*, **4**(5), 441–447.

[117] Lees, J. A., Vehkala, M., Välimäki, N., Harris, S. R., Chewapreecha, C., Croucher, N. J., *et al.* (2016). Sequence element enrichment analysis to determine the genetic basis of bacterial phenotypes. *Nature communications*, **7**, 12797.

[118] Lees, J. A., Galardini, M., Bentley, S. D., Weiser, J. N., and Corander, J. (2018). pyseer: a comprehensive tool for microbial pangenome-wide association studies. *Bioinformatics*, page bty539.

[119] Li, C. and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, **24**(9), 1175–1182.

[120] Li, J., Das, K., Fu, G., Li, R., and Wu, R. (2010). The bayesian lasso for genome-wide association studies. *Bioinformatics*, **27**(4), 516–523.

[121] Liaw, A. and Wiener, M. (2002). Classification and regression by `randomForest`. *R News*, **2**(3), 18–22.

[122] Liebert, C. A., Hall, R. M., and Summers, A. O. (1999). Transposon Tn*21*, flagship of the floating genome. *Microbiology and Molecular Biology Reviews*, **63**(3), 507–522.

[123] Limasset, A., Rizk, G., Chikhi, R., and Peterlongo, P. (2017). Fast and scalable minimal perfect hashing for massive key sets. *arXiv doi:1702.03154*.

[124] Lister, P. D., Wolter, D. J., and Hanson, N. D. (2009). Antibacterial-resistant *Pseudomonas aeruginosa*: clinical impact and complex regulation of chromosomally encoded resistance mechanisms. *Clinical microbiology reviews*, **22**(4), 582–610.

[125] Liu, J., Wang, K., Ma, S., and Huang, J. (2013). Accounting for linkage disequilibrium in genome-wide association studies: a penalized regression method. *Statistics and its interface*, **6**(1), 99.

[126] Liu, Y.-Y., Wang, Y., Walsh, T. R., Yi, L.-X., Zhang, R., Spencer, J., *et al.* (2016). Emergence of plasmid-mediated colistin resistance mechanism MCR-1 in animals and human beings in China: a microbiological and molecular biological study. *The Lancet infectious diseases*, **16**(2), 161–168.

[127] Llinares-López, F., Grimm, D. G., Bodenham, D. A., Gieraths, U., Sugiyama, M., Rowan, B., *et al.* (2015). Genome-wide detection of intervals of genetic heterogeneity associated with complex traits. *Bioinformatics*, **31**(12), i240–i249.

[128] Lockhart, R., Taylor, J., Tibshirani, R. J., and Tibshirani, R. (2014). A significance test for the lasso. *Annals of statistics*, **42**(2), 413.

[129] Loman, N. J., Constantinidou, C., Chan, J. Z., Halachev, M., Sergeant, M., Penn, C. W., *et al.* (2012). High-throughput bacterial genome sequencing: an embarrassment of choice, a world of opportunity. *Nature reviews microbiology*, **10**(9), 599.

[130] Lowy, F. D. (2003). Antimicrobial resistance: the example of *Staphylococcus aureus*. *Journal of clinical investigation*, **111**(9), 1265.

[131] MacPherson, A., Otto, S. P., and Nuismer, S. L. (2018). Keeping pace with the red queen: Identifying the genetic basis of susceptibility to infectious disease. *Genetics*, **208**(2), 779–789.

[132] Mah, T.-F. C. and O'toole, G. A. (2001). Mechanisms of biofilm resistance to antimicrobial agents. *Trends in microbiology*, **9**(1), 34–39.

[133] Mairal, J. and Yu, B. (2013). Supervised feature selection in graphs with path coding penalties and network flows. *The journal of machine learning research*, **14**(1), 2449–2485.

[134] Manolio, T. A., Collins, F. S., Cox, N. J., Goldstein, D. B., Hindorff, L. A., Hunter, D. J., *et al.* (2009). Finding the missing heritability of complex diseases. *Nature*, **461**(7265), 747.

[135] Marçais, G. and Kingsford, C. (2011). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, **27**(6), 764–770.

[136] Marschall, T., Marz, M., Abeel, T., Dijkstra, L., Dutilh, B. E., Ghaffaari, A., *et al.* (2016). Computational pan-genomics: status, promises and challenges. *Briefings in bioinformatics*, page bbw089.

[137] Martínez, J. L. and Rojo, F. (2011). Metabolic regulation of antibiotic resistance. *FEMS microbiology reviews*, **35**(5), 768–789.

[138] Martínez, J. L., Fajardo, A., Garmendia, L., Hernandez, A., Linares, J. F., Martínez-Solano, L., *et al.* (2008). A global view of antibiotic resistance. *FEMS microbiology reviews*, **33**(1), 44–65.

[139] Masuda, N., Sakagawa, E., Ohya, S., Gotoh, N., Tsujimoto, H., and Nishino, T., *et al.* (2000). Substrate specificities of MexAB-OprM, MexCD-OprJ, and MexXY-oprM efflux pumps in *Pseudomonas aeruginosa*. *Antimicrobial agents and chemotherapy*, **44**(12), 3322–3327.

[140] Mazel, D. (2006). Integrons: agents of bacterial evolution. *Nature Reviews Microbiology*, **4**(8), 608.

[141] McArthur, A. G., Waglechner, N., Nizam, F., Yan, A., Azad, M. A., Baylay, A. J., *et al.* (2013). The comprehensive antibiotic resistance database. *Antimicrobial agents and chemotherapy*, pages AAC–00419.

[142] McCullagh, P. (1980). Regression models for ordinal data. *Journal of the royal statistical society. Series B (Methodological)*, pages 109–142.

[143] Mckinney, B. and Pajewski, N. (2012). Six degrees of epistasis: statistical network models for GWAS. *Frontiers in genetics*, **2**, 109.

[144] Meinshausen, N., Meier, L., and Bühlmann, P. (2009). *P*-values for high-dimensional regression. *Journal of the american statistical association*, **104**(488), 1671–1681.

[145] Meyerson, M., Gabriel, S., and Getz, G. (2010). Advances in understanding cancer genomes through second-generation sequencing. *Nature reviews genetics*, **11**(10), 685.

[146] Micek, S. T., Wunderink, R. G., Kollef, M. H., Chen, C., Rello, J., Chastre, J., *et al.* (2015). An international multicenter retrospective study of *Pseudomonas aeruginosa* nosocomial pneumonia: impact of multidrug resistance. *Crit Care*, **19**(219.10), 1186.

[147] Mlynarczyk, A., Mlynarczyk, G., and Jeljaszewicz, J. (1998). The genome of *Staphylococcus aureus*: a review. *Zentralblatt für Bakteriologie*, **287**(4), 277–314.

[148] Moradigaravand, D., Palm, M., Farewell, A., Mustonen, V., Warringer, J., and Parts, L., *et al.* (2018). Precise prediction of antibiotic resistance in *Escherichia coli* from full genome sequences. *bioRxiv*, page 338194.

[149] Murray, J. L., Kwon, T., Marcotte, E. M., and Whiteley, M. (2015). Intrinsic antimicrobial resistance determinants in the superbug *Pseudomonas aeruginosa*. *Mbio*, **6**(6), e01603–15.

[150] Noble, W. S. (2009). How does multiple testing correction work? *Nature biotechnology*, **27**(12), 1135.

[151] Olekhnovich, E. I., Vasilyev, A. T., Ulyantsev, V. I., Kostryukova, E. S., and Tyakht, A. V. (2017). MetaCherchant: analyzing genomic context of antibiotic resistance genes in gut microbiota. *Bioinformatics*, **34**(3), 434–444.

[152] Oliver, A., Mulet, X., López-Causapé, C., and Juan, C. (2015). The increasing threat of *Pseudomonas aeruginosa* high-risk clones. *Drug resistance updates*, **21**, 41–59.

[153] Ondov, B. D., Treangen, T. J., Melsted, P., Mallonee, A. B., Bergman, N. H., Koren, S., *et al.* (2016). Mash: fast genome and metagenome distance estimation using MinHash. *Genome biology*, **17**(1), 132.

[154] O'Neill, J. *et al.* (2016). Tackling drug-resistant infections globally: final report and recommendations. *Review on antimicrobial resistance*.

[155] Overbeek, R., Olson, R., Pusch, G. D., Olsen, G. J., Davis, J. J., Disz, T., *et al.* (2013). The SEED and the rapid annotation of microbial genomes using subsystems technology (RAST). *Nucleic acids research*, **42**(D1), D206–D214.

[156] Page, A. J., Cummins, C. A., Hunt, M., Wong, V. K., Reuter, S., Holden, M. T., *et al.* (2015). Roary: rapid large-scale prokaryote pan genome analysis. *Bioinformatics*, **31**(22), 3691–3693.

[157] Pallen, M. J., Loman, N. J., and Penn, C. W. (2010). High-throughput sequencing and clinical microbiology: progress, opportunities and challenges. *Current opinion in microbiology*, **13**(5), 625–631.

[158] Palomino, J. C. and Martin, A. (2014). Drug resistance mechanisms in *Mycobacterium tuberculosis*. *Antibiotics*, **3**(3), 317–340.

[159] Paten, B., Novak, A. M., Eizenga, J. M., and Garrison, E. (2017). Genome graphs and the evolution of genome inference. *Genome research*, **27**(5), 665–676.

[160] Peng, Y., Leung, H. C., Yiu, S.-M., and Chin, F. Y. (2010). IDBA–a practical iterative de Bruijn graph *de novo* assembler. In *Annual international conference on research in computational molecular biology*, pages 426–440. Springer.

[161] Peng, Y., Leung, H. C., Yiu, S.-M., and Chin, F. Y. (2012). IDBA-UD: a *de novo* assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*, **28**(11), 1420–1428.

[162] Peterlongo, P., Schnel, N., Pisanti, N., Sagot, M.-F., and Lacroix, V. (2010). Identifying SNPs without a reference genome by comparing raw reads. In *International symposium on string processing and information retrieval*, pages 147–158. Springer.

[163] Pevzner, P. A., Tang, H., and Waterman, M. S. (2001). An Eulerian path approach to DNA fragment assembly. *Proceedings of the national academy of sciences*, **98**(17), 9748–9753.

[164] Pirnay, J.-P., Bilocq, F., Pot, B., Cornelis, P., Zizi, M., Van Eldere, J., *et al.* (2009). *Pseudomonas aeruginosa* population structure revisited. *PLoS one*, **4**(11), e7740.

[165] Piton, J., Petrella, S., Delarue, M., André-Leroux, G., Jarlier, V., Aubry, A., *et al.* (2010). Structural insights into the quinolone resistance mechanism of *Mycobacterium tuberculosis* DNA gyrase. *PLoS one*, **5**(8), e12245.

[166] Popescu, A.-A., Harper, A. L., Trick, M., Bancroft, I., and Huber, K. T. (2014). A novel and fast approach for population structure inference using kernel-PCA and optimization (PSIKO). *Genetics*, pages genetics–114.

[167] Power, R. A., Parkhill, J., and de Oliveira, T. (2017). Microbial genome-wide association studies: lessons from human GWAS. *Nature reviews genetics*, **18**(1), 41–50.

[168] Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., and Reich, D., *et al.* (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, **38**(8), 904.

[169] Pritchard, J. K., Stephens, M., Rosenberg, N. A., and Donnelly, P. (2000). Association mapping in structured populations. *The american journal of human genetics*, **67**(1), 170–181.

[170] Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., *et al.* (2007). PLINK: a tool set for whole-genome association and population-based linkage analyses. *The american journal of human genetics*, **81**(3), 559–575.

[171] Qi, L., Li, H., Zhang, C., Liang, B., Li, J., Wang, L., *et al.* (2016). Relationship between antibiotic resistance, biofilm formation, and biofilm-specific resistance in *Acinetobacter baumannii*. *Frontiers in microbiology*, **7**, 483.

[172] Quail, M. A., Smith, M., Coupland, P., Otto, T. D., Harris, S. R., Connor, T. R., *et al.* (2012). A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC genomics*, **13**(1), 341.

[173] Rahman, A., Hallgrímsdóttir, I., Eisen, M. B., and Pachter, L. (2017). Association mapping from sequencing reads using k-mers. *bioRxiv doi: 10.1101/141267*.

[174] Rahman, A., Hallgrímsdóttir, I., Eisen, M., and Pachter, L. (2018). Association mapping from sequencing reads using k-mers. *eLife*, **7**, e32920.

[175] Read, T. D. and Massey, R. C. (2014). Characterizing the genetic basis of bacterial phenotypes using genome-wide association studies: a new direction for bacteriology. *Genome medicine*, **6**(11), 109.

[176] Reinert, K., Dadi, T. H., Ehrhardt, M., Hauswedell, H., Mehringer, S., Rahn, R., *et al.* (2017). The SeqAn C++ template library for efficient sequence analysis: a resource for programmers. *Journal of biotechnology*, **261**, 157–168.

[177] Reller, L. B., Weinstein, M., Jorgensen, J. H., and Ferraro, M. J. (2009). Antimicrobial susceptibility testing: a review of general principles and contemporary practices. *Clinical infectious diseases*, **49**(11), 1749–1755.

[178] Rothberg, J. M., Hinz, W., Rearick, T. M., Schultz, J., Mileski, W., Davey, M., *et al.* (2011). An integrated semiconductor device enabling non-optical genome sequencing. *Nature*, **475**(7356), 348.

[179] Rowe, W., Baker, K. S., Verner-Jeffreys, D., Baker-Austin, C., Ryan, J. J., Maskell, D., *et al.* (2015). Search engine for antimicrobial resistance: a cloud compatible pipeline and web interface for rapidly detecting antimicrobial resistance genes directly from sequence data. *PLoS one*, **10**(7), e0133492.

[180] Sacomoto, G. A., Kielbassa, J., Chikhi, R., Uricaru, R., Antoniou, P., Sagot, M.-F., *et al.* (2012). KISSPLICE: *de novo* calling alternative splicing events from RNA-seq data. *BMC bioinformatics*, **13**(Suppl 6), S5.

[181] Seemann, T. (2014). Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, **30**(14), 2068–2069.

[182] Setakis, E., Stirnadel, H., and Balding, D. J. (2006). Logistic regression protects against population structure in genetic association studies. *Genome research*, **16**(2), 290–296.

[183] Sheppard, S. K., Didelot, X., Meric, G., Torralbo, A., Jolley, K. A., Kelly, D. J., *et al.* (2013). Genome-wide association study identifies vitamin B5 biosynthesis as a host specificity factor in Campylobacter. *Proceedings of the national academy of sciences*, **110**(29), 11923–11927.

[184] Slatkin, M. (2008). Linkage disequilibrium—understanding the evolutionary past and mapping the medical future. *Nature reviews genetics*, **9**(6), 477.

[185] Stanton, J. M. (2001). Galton, Pearson, and the peas: a brief history of linear regression for statistics instructors. *Journal of statistics education*, **9**(3).

[186] Stitson, M., Weston, J., Gammerman, A., Vovk, V., and Vapnik, V. (1996). Theory of support vector machines. *University of London*, **117**(827), 188–191.

[187] Stoesser, N., Batty, E., Eyre, D., Morgan, M., Wyllie, D., Del Ojo Elias, C., *et al.* (2013). Predicting antimicrobial susceptibilities for *Escherichia coli* and *Klebsiella pneumoniae* isolates using whole genomic sequence data. *Journal of antimicrobial chemotherapy*, **68**(10), 2234–2244.

[188] Tasoulis, S., Cheng, L., Välimäki, N., Croucher, N. J., Harris, S. R., Hanage, W. P., *et al.* (2014). Random projection based clustering for population genomics. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 675–682. IEEE.

[189] Thomas, C. M. and Nielsen, K. M. (2005). Mechanisms of, and barriers to, horizontal gene transfer between bacteria. *Nature reviews microbiology*, **3**(9), 711.

[190] Thornton, T. and McPeek, M. S. (2010). ROADTRIPS: case-control association testing with partially or completely unknown population and pedigree structure. *The american journal of human genetics*, **86**(2), 172–184.

[191] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

[192] Toprak, E., Veres, A., Michel, J.-B., Chait, R., Hartl, D. L., and Kishony, R., *et al.* (2012). Evolutionary paths to antibiotic resistance under dynamically sustained drug selection. *Nature genetics*, **44**(1), 101.

[193] Tournoud, M. and Mahé, P. (2018). Predicting bacterial resistance from whole-genome sequences using *k*-mers and stability selection. *BMC Bioinformatics*, **in preparation**.

[194] Tran, F. and Boedicker, J. Q. (2017). Genetic cargo and bacterial species set the rate of vesicle-mediated horizontal gene transfer. *Scientific reports*, **7**(1), 8813.

[195] Traore, H., Fissette, K., Bastian, I., Devleeschouwer, M., and Portaels, F. (2000). Detection of rifampicin resistance in *Mycobacterium tuberculosis* isolates from diverse countries by a commercial line probe assay as an initial indicator of multidrug resistance. *The international journal of tuberculosis and lung disease*, **4**(5), 481–484.

[196] Tuite, N., Reddington, K., Barry, T., Zumla, A., and Enne, V. (2014). Rapid nucleic acid diagnostics for the detection of antimicrobial resistance in Gram-negative bacteria: is it time for a paradigm shift? *Journal of antimicrobial chemotherapy*, **69**(7), 1729–1733.

[197] UniProt consortium (2017). UniProt: the universal protein knowledgebase. *Nucleic acids research*, **45**(D1), D158–D169.

[198] Uricaru, R., Rizk, G., Lacroix, V., Quillery, E., Plantard, O., Chikhi, R., *et al.* (2014). Reference-free detection of isolated SNPs. *Nucleic acids research*, **43**(2), e11–e11.

[199] van Belkum, A. and Dunne, W. M. (2013). Next generation antimicrobial susceptibility testing. *Journal of clinical microbiology*, pages JCM–00313.

[200] van Belkum, A., Soriaga, L. B., LaFave, M. C., Akella, S., Veyrieras, J.-B., Barbu, E. M., *et al.* (2015). Phylogenetic distribution of CRISPR-Cas systems in antibiotic-resistant *Pseudomonas aeruginosa*. *mBio*, **6**(6), e01796–15.

[201] Vert, J.-P., Tsuda, K., and Schölkopf, B. (2004). A primer on kernel methods. *Kernel methods in computational biology*, **47**, 35–70.

[202] Walker, T. M., Kohl, T. A., Omar, S. V., Hedge, J., Elias, C. D. O., Bradley, P., *et al.* (2015). Whole-genome sequencing for prediction of *Mycobacterium tuberculosis* drug susceptibility and resistance: a retrospective cohort study. *The Lancet infectious diseases*, **15**(10), 1193–1202.

[203] Walsh, C. *et al.* (2003). *Antibiotics: actions, origins, resistance.* American Society for Microbiology (ASM).

[204] Wattam, A. R., Davis, J. J., Assaf, R., Boisvert, S., Brettin, T., Bun, C., *et al.* (2016). Improvements to PATRIC, the all-bacterial bioinformatics database and analysis resource center. *Nucleic acids research*, **45**(D1), D535–D542.

[205] Wei, Q., Tarighi, S., Dötsch, A., Häussler, S., Müsken, M., Wright, V. J., *et al.* (2011). Phenotypic and genome-wide analysis of an antibiotic-resistant small colony variant (SCV) of *Pseudomonas aeruginosa*. *PloS one*, **6**(12), e29276.

[206] Westh, H., Hougaard, D., Vuust, J., and Rosdahl, V. (1995). Prevalence of *erm* gene classes in erythromycin-resistant *Staphylococcus aureus* strains isolated between 1959 and 1988. *Antimicrobial agents and chemotherapy*, **39**(2), 369–373.

[207] Widmer, C., Lippert, C., Weissbrod, O., Fusi, N., Kadie, C., Davidson, R., *et al.* (2014). Further improvements to linear mixed models for genome-wide association studies. *Scientific reports*, **4**.

[208] Wiehlmann, L., Wagner, G., Cramer, N., Siebert, B., Gudowius, P., Morales, G., *et al.* (2007). Population structure of *Pseudomonas aeruginosa*. *Proceedings of the national academy of sciences*, **104**(19), 8101–8106.

[209] Wilks, S. S. (1938). The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The annals of mathematical statistics*, **9**(1), 60–62.

[210] Witney, A., Gould, K., Pope, C., Bolt, F., Stoker, N., Cubbon, M., *et al.* (2014). Genome sequencing and characterization of an extensively drug-resistant sequence type 111 serotype O12 hospital outbreak strain of *Pseudomonas aeruginosa*. *Clinical Microbiology and Infection*, **20**(10), O609–O618.

[211] Woerther, P.-L., Angebault, C., Jacquier, H., Clermont, O., El Mniai, A., Moreau, B., *et al.* (2013). Characterization of fecal ESBL-producing *Escherichia coli* in a remote community during a long term period. *Antimicrobial agents and chemotherapy*, pages AAC–00848.

[212] World Health Organization (2014). Antimicrobial resistance: global report on surveillance. *Geneva: WHO Press Release*.

[213] World Health Organization (2017). Global tuberculosis report. *Geneva: WHO Press Release*.

[214] Wright, L. L., Turton, J. F., Hopkins, K. L., Livermore, D. M., and Woodford, N. (2015). Genetic environment of metallo-β-lactamase genes in *Pseudomonas aeruginosa* isolates from the UK. *Journal of antimicrobial chemotherapy*, **70**(12), 3250–3258.

[215] Wu, T. T., Chen, Y. F., Hastie, T., Sobel, E., and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, **25**(6), 714–721.

[216] Yanardag, P. and Vishwanathan, S. (2015). Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1365–1374. ACM.

[217] Yoon, S., Xuan, Z., Makarov, V., Ye, K., and Sebat, J. (2009). Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome research*.

[218] Yu, J., Pressoir, G., Briggs, W. H., Bi, I. V., Yamasaki, M., Doebley, J. F., *et al.* (2006). A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. *Nature genetics*, **38**(2), 203.

[219] Zankari, E., Hasman, H., Kaas, R. S., Seyfarth, A. M., Agersø, Y., Lund, O., *et al.* (2012a). Genotyping using whole-genome sequencing is a realistic alternative to surveillance based on phenotypic antimicrobial susceptibility testing. *Journal of antimicrobial chemotherapy*, **68**(4), 771–777.

[220] Zankari, E., Hasman, H., Cosentino, S., Vestergaard, M., Rasmussen, S., Lund, O., *et al.* (2012b). Identification of acquired antimicrobial resistance genes. *Journal of antimicrobial chemotherapy*, **67**(11), 2640–2644.

[221] Zerbino, D. and Birney, E. (2008). Velvet: algorithms for *de novo* short read assembly using de Bruijn graphs. *Genome research*, pages gr–074492.

[222] Zhang, H., Li, D., Zhao, L., Fleming, J., Lin, N., Wang, T., *et al.* (2013). Genome sequencing of 161 *Mycobacterium tuberculosis* isolates from China identifies genes and intergenic regions associated with drug resistance. *Nature genetics*, **45**(10), 1255–1260.

[223] Zhang, W., Chen, J., Yang, Y., Tang, Y., Shang, J., and Shen, B., *et al.* (2011). A practical comparison of *de novo* genome assembly software tools for next-generation sequencing technologies. *PloS one*, **6**(3), e17915.

[224] Zhao, S., Tyson, G., Chen, Y., Li, C., Mukherjee, S., Young, S., *et al.* (2015). Whole genome sequencing analysis accurately predicts antimicrobial resistance phenotypes in Campylobacter. *Applied and environmental microbiology*, pages AEM–02873.

[225] Zhou, X. and Stephens, M. (2012). Genome-wide efficient mixed-model analysis for association studies. *Nature genetics*, **44**(7), 821–824.

[226] Zhou, X. and Stephens, M. (2014). Efficient multivariate linear mixed-model algorithms for genome-wide association studies. *Nature methods*, **11**(4), 407.

[227] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**(2), 301–320.

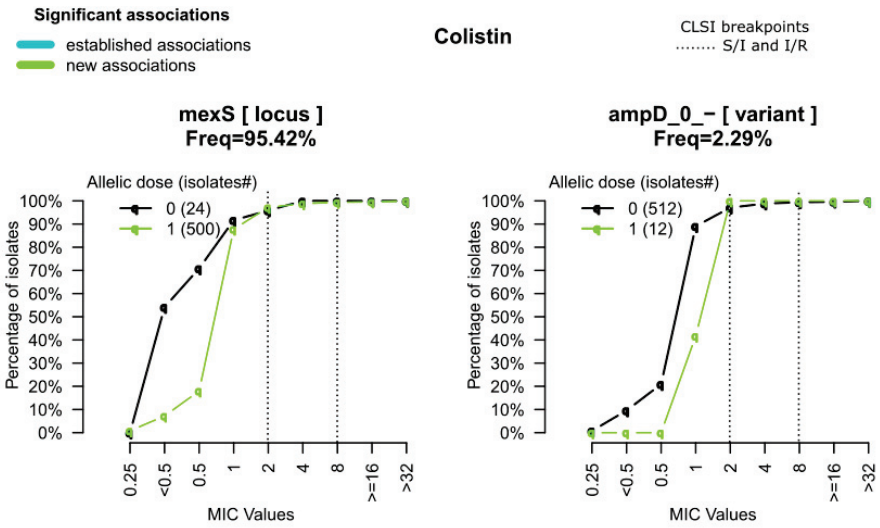
Appendix

Table S1.1: **Phenotypes.** This table gives for each isolate included in the panel, its CLSI status (I, R or S), for each of the drug. Missing values (NA) appear when the isolate was not phenotyped for a given drug. This table provides only a short view of the complete published Supplementary Table.

Isolate id	Amikacin	Meropenem	Levofloxacin	Colistin	Cefepime	Piperacillin	...
VTK_100073	S	S	S	S	S	S	...
VTK_102200	S	S	S	S	I	R	...
VTK_102323	R	I	S	S	S	R	...
VTK_102327	S	S	S	S	S	S	...
VTK_104414	S	S	S	NA	I	R	...
VTK_104415	S	S	S	NA	S	R	...
VTK_104416	S	R	R	NA	R	R	...
VTK_104417	S	S	S	NA	R	R	...
VTK_104430	S	I	S	NA	I	R	...
VTK_104431	S	S	S	NA	S	R	...
VTK_104432	S	S	S	NA	S	R	...
VTK_104477	R	S	S	NA	S	S	...
VTK_104478	R	I	S	NA	R	R	...
VTK_104566	S	S	S	NA	S	S	...
VTK_104571	S	S	S	NA	S	S	...
VTK_104572	S	S	I	NA	S	S	...
VTK_105072	S	S	R	S	I	R	...
VTK_105076	S	S	R	S	I	R	...
VTK_105352	S	S	R	S	S	R	...
VTK_105355	S	S	R	S	S	R	...
VTK_105356	S	S	R	S	I	R	...
VTK_105406	S	S	R	S	I	R	...
VTK_105617	S	R	R	S	R	R	...
...	...	...	...	...	...	...	...

File S1: Integrase sequences (first sequence only)

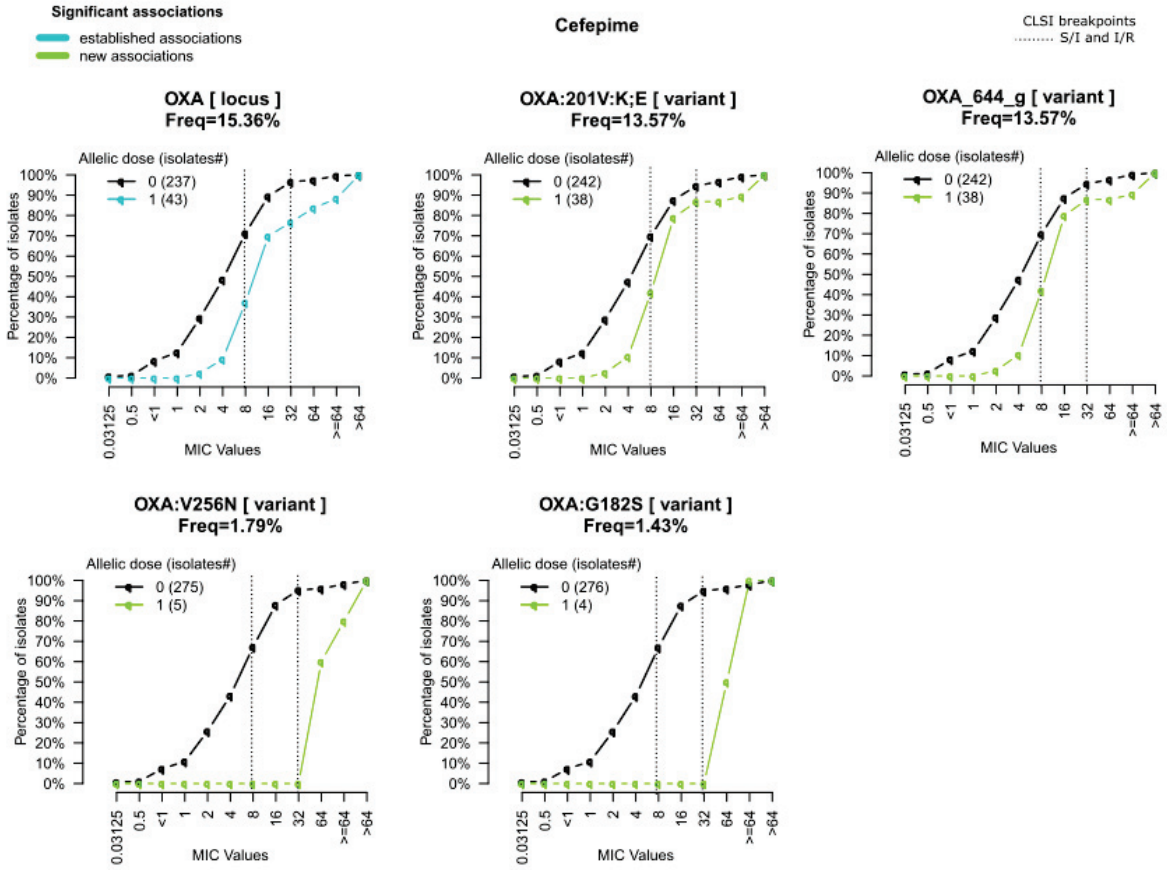
>tr|Q58G63|Q58G63_PSEAI DNA integrase IntI1 (Fragment) OS=Pseudomonas aeruginosa GN=intI1 PE=4 SV=1
MKTATAPLPPLRSVKVLDQLRERIRYLHYSLRTEQAYVNWVRAFIRFHGVRHPATLGSSE
VEAFLSWLANERKVSVSTHRQALAALLFFYGKVLCTDLPWLQEIGRPRPS



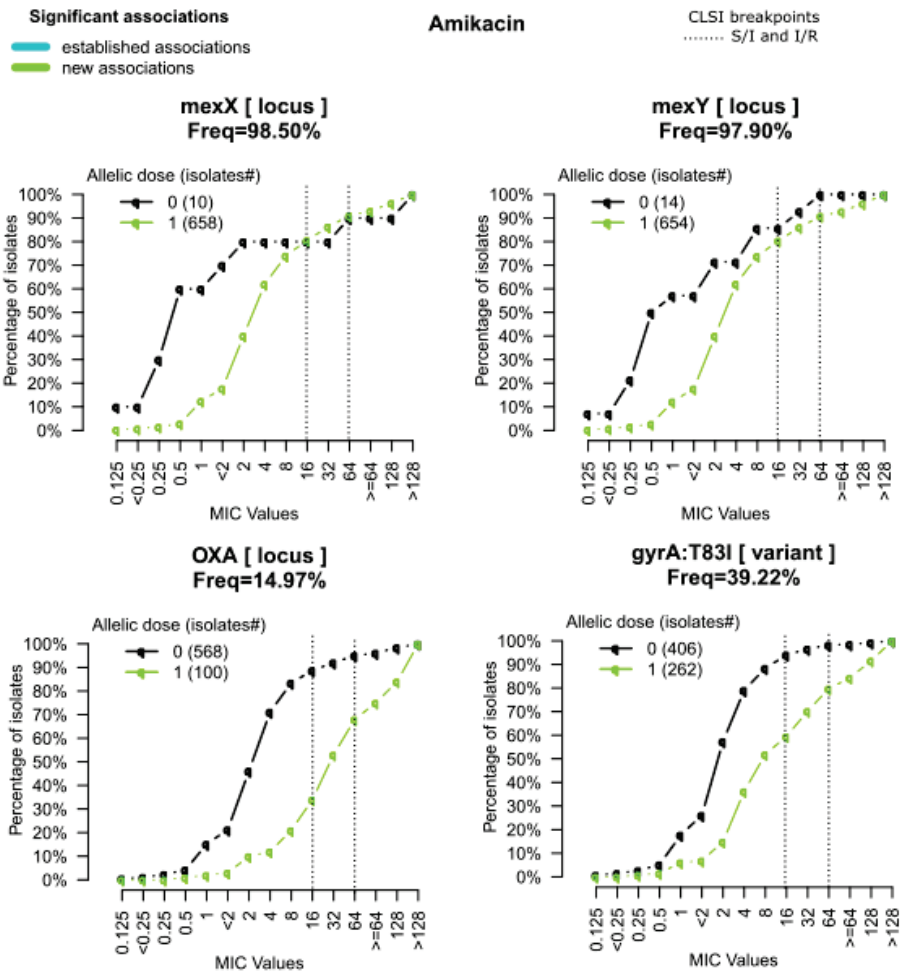
Supplementary Figure S1.1: **Significant associations to colistin MIC variation.** Each graph represents the percent of tested isolates for each value of MIC for a given determinant. The black curve shows this percent among isolate lacking the allele (either absence of gene or absence of mutation), while the colored curve represents the percent of isolates with the allele. The left dashed lines represent CLSI breakpoints between susceptible (S) and intermediate (I) strains while the right dashed lines represent the CLSI breakpoints between I and resistant (R) strains. The only two determinants found significantly associated to colistin MIC are new associations, and the MIC variations stand below the S/I breakpoint.

Table S1.2: **Resistotypes.** This table gives for each isolate included in the panel, the copy number of each AR genetic determinant found in its genome. This table provides only a short view of the complete published Supplementary Table.

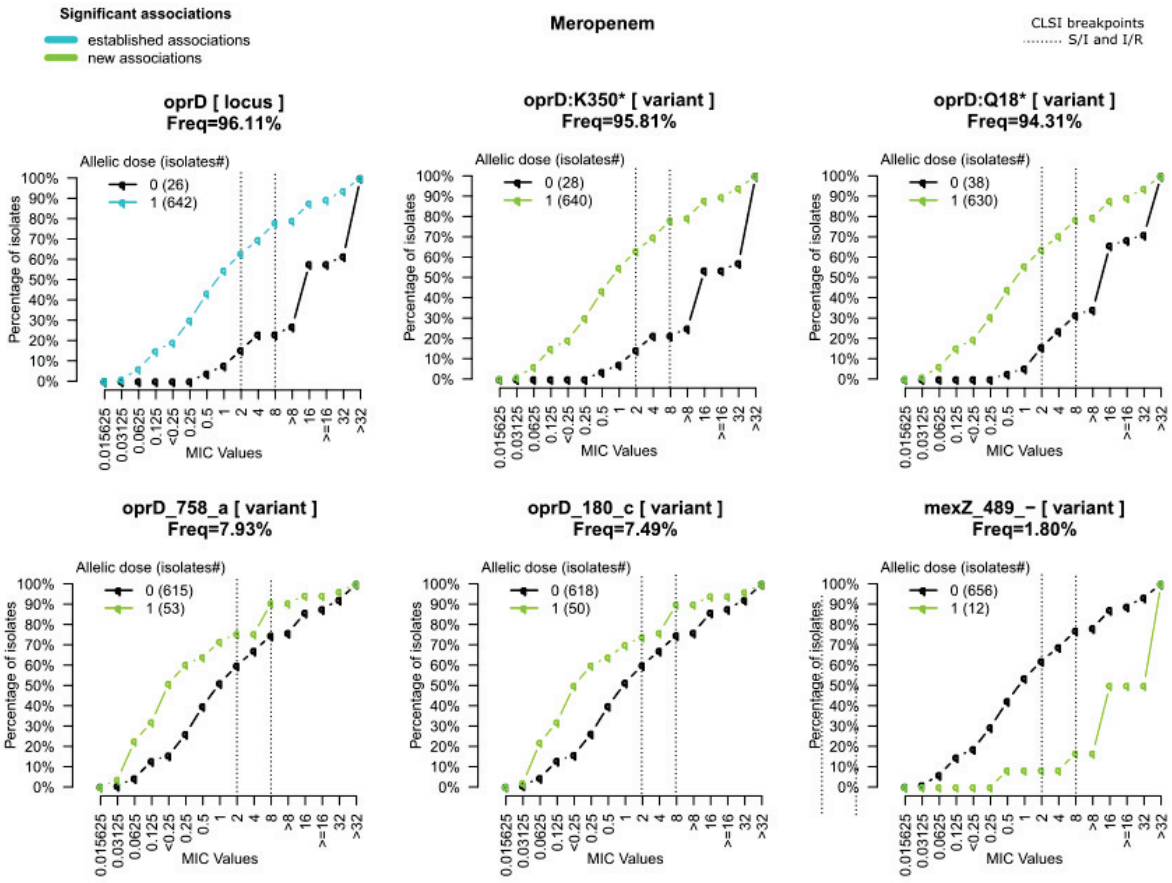
Isolate id	folP	FosA	GES	gyrA	gyrA:T83I	gyrA:D87Y	gyrB:I529V	...
VTK_104572	1	1	0	1	1	0	0	...
VTK_105072	1	1	0	1	1	0	0	...
VTK_105076	1	1	0	1	1	0	0	...
VTK_105352	1	0	0	1	1	0	1	...
VTK_105355	1	1	0	1	1	0	0	...
VTK_105356	1	0	0	1	1	0	1	...
VTK_105406	1	1	0	1	1	0	0	...
VTK_105617	1	1	0	1	0	0	0	...
VTK_106155	1	1	0	1	1	0	0	...
VTK_106156	1	1	0	1	0	0	1	...
VTK_106158	1	0	0	1	1	0	1	...
VTK_106176	1	1	0	1	0	1	0	...
VTK_106177	1	1	0	1	0	1	0	...
VTK_106179	1	1	0	1	0	0	0	...
VTK_106180	1	1	0	1	1	0	0	...
VTK_106181	1	1	0	1	1	0	0	...
VTK_106182	1	1	0	1	1	0	0	...
VTK_106188	1	1	0	1	0	0	0	...
VTK_106189	1	1	0	1	0	0	0	...
VTK_106191	1	1	0	1	1	0	0	...
...	...	...	...	...	...	...	...	...



Supplementary Figure S1.2: **Significant associations to cefepime MIC variation.**Each graph represents the percent of tested isolates for each value of MIC for a given determinant. Black curve show this percent among isolate lacking the allele (either absence of gene or absence of mutation), while the colored curve represent the percent of isolates with the allele. The left dashed lines represent CLSI breakpoints between susceptible (S) and intermediate (I) strains while the right dashed lines represent the CLSI breakpoints between I and resistant (R) strains. The established OXA association was found significant, as well as variants within the OXA gene.



Supplementary Figure S1.3: **Significant new associations to amikacin MIC variation.** Each graph represents the percent of tested isolates for each value of MIC for a given determinant. Black curve show this percent among isolate lacking the allele (either absence of gene or absence of mutation), while the colored curve represent the percent of isolates with the allele. The left dashed lines represent CLSI breakpoints between susceptible (S) and intermediate (I) strains while the right dashed lines represent the CLSI breakpoints between I and resistant (R) strains. This panel only show new associations to amikacin MIC variation, among them gyrA:T83I and OXA seem to be spurious associations. Other associations are not shown.



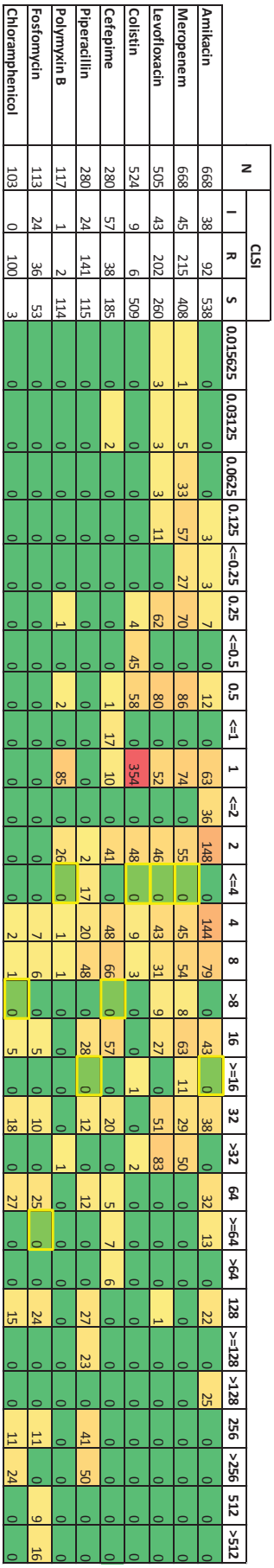
Supplementary Figure S1.4: **Significant associations to meropenem MIC variation.**Each graph represents the percent of tested isolates for each value of MIC for a given determinant. Black curve show this percent among isolate lacking the allele (either absence of gene or absence of mutation), while the colored curve represent the percent of isolates with the allele. The left dashed lines represent CLSI breakpoints between susceptible (S) and intermediate (I) strains while the right dashed lines represent the CLSI breakpoints between I and resistant (R) strains. This panel show only a part of the associations, that are detailed in the article. Other associations are not shown.

Table S1.3: **Association results for established determinants.** This table describe the results of the RWAS for known determinants. The first 3 columns detail the determinant, columns 4 and 5 give the association score, column 6 gives the determinant minor allele frequency, columns 7 and 8 describe the antibiotic, and the last column mentions if the determinant was kept in the model built to find new associations. This table provides only a short view of the complete published Supplementary Table.

variant_id	type	p-value	q-value	frequency	antibiotic	kept as covariate
gyrA:T83I	variant	1.29E-61	2.32E-60	3.94E-01	Levofloxacin	TRUE
parC:S87L	variant	3.52E-40	3.17E-39	2.63E-01	Levofloxacin	TRUE
gyrA:D87N	variant	9.14E-08	5.49E-07	5.21E-02	Levofloxacin	TRUE
AAC(6')-Ib9	locus	2.57E-07	1.15E-06	2.65E-01	Levofloxacin	FALSE
parC:S87W	variant	1.32E-03	4.37E-03	3.42E-02	Levofloxacin	TRUE
parC:E91K	variant	1.46E-03	4.37E-03	1.49E-02	Levofloxacin	TRUE
AAC(6')-Ib-H.	locus	1.20E-02	3.00E-02	4.46E-02	Levofloxacin	FALSE
gyrB:E468D	variant	1.33E-02	3.00E-02	8.93E-03	Levofloxacin	FALSE
AAC(6')-Ib'	locus	2.37E-02	4.75E-02	5.65E-02	Levofloxacin	FALSE
AAC(6')-Ib3	locus	5.17E-02	9.31E-02	3.57E-02	Levofloxacin	FALSE
gyrB:S466Y	variant	8.37E-02	1.37E-01	2.68E-02	Levofloxacin	TRUE
parE:A473V	variant	1.25E-01	1.75E-01	1.79E-02	Levofloxacin	TRUE
gyrB:467/468	variant	1.27E-01	1.75E-01	1.49E-03	Levofloxacin	FALSE
gyrB:S466F	variant	2.14E-01	2.75E-01	4.46E-03	Levofloxacin	TRUE
gyrA:D87Y	variant	6.13E-01	7.36E-01	1.79E-02	Levofloxacin	FALSE
gyrB:I529V_2	variant	7.46E-01	7.90E-01	1.93E-02	Levofloxacin	TRUE
gyrB:I529V_1	variant	7.46E-01	7.90E-01	1.93E-02	Levofloxacin	TRUE
gyrA:T83A	variant	9.86E-01	9.86E-01	7.44E-03	Levofloxacin	FALSE
rmtD	locus	2.31E-09	6.00E-08	1.19E-02	Amikacin	TRUE
APH(3')-VIa	locus	9.82E-09	1.28E-07	2.38E-02	Amikacin	TRUE
AAC(6')-Ib9	locus	3.41E-07	2.95E-06	2.65E-01	Amikacin	TRUE
AAC(6')-Ib3	locus	2.79E-06	1.81E-05	3.57E-02	Amikacin	TRUE
AAC(3)-IIa	locus	3.64E-06	1.89E-05	5.95E-03	Amikacin	TRUE
AAC(6')-29a	locus	4.50E-04	1.95E-03	1.34E-02	Amikacin	TRUE
AAC(6')-Ib-H.	locus	2.82E-03	1.05E-02	4.46E-02	Amikacin	TRUE
AAC(6')-33	locus	1.21E-02	3.92E-02	4.46E-03	Amikacin	TRUE
AAC(6')-29b	locus	1.43E-02	4.13E-02	8.93E-03	Amikacin	TRUE
AAC(3)-IIIb	locus	2.29E-02	5.95E-02	5.95E-03	Amikacin	TRUE
ANT(4')-IIb	locus	9.18E-02	2.17E-01	1.49E-03	Amikacin	FALSE
APH(3')-VIb	locus	1.11E-01	2.40E-01	1.04E-02	Amikacin	TRUE
APH(2")-IIa	locus	1.22E-01	2.40E-01	1.49E-03	Amikacin	FALSE
rmtF	locus	1.29E-01	2.40E-01	1.49E-03	Amikacin	FALSE
APH(3')-Ib	locus	2.00E-01	3.47E-01	8.93E-03	Amikacin	TRUE
AAC(6')-Iai	locus	2.22E-01	3.61E-01	1.49E-03	Amikacin	FALSE
AAC(6')-Ib'	locus	3.27E-01	5.00E-01	5.65E-02	Amikacin	TRUE
AAC(3)-Ic	locus	3.63E-01	5.24E-01	4.46E-03	Amikacin	TRUE
...	...	...	...	...	...	...
mexT	locus	1.14E-01	3.41E-01	9.84E-01	Chloramphenicol	FALSE
nfxB	locus	6.15E-01	6.34E-01	9.79E-01	Chloramphenicol	TRUE
nalC	locus	6.34E-01	6.34E-01	9.97E-01	Chloramphenicol	FALSE
...	...	...	...	...	...	...

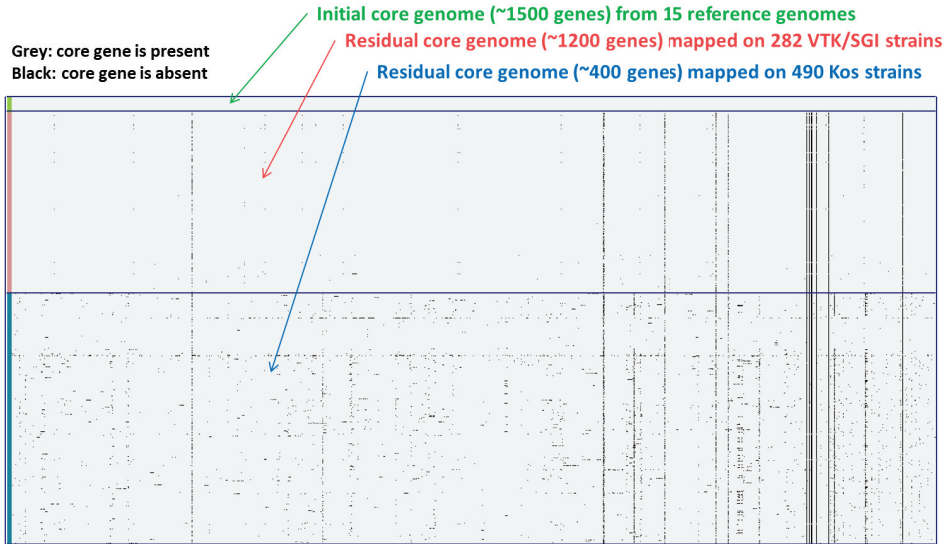
Table S1.4: **Association results for new candidates.** This table describe the results of the RWAS for new candidates. The first 2 columns describe the antibiotic, the following 4 columns detail the determinant, column 7 gives the determinant minor allele frequency, columns 8 and 9 give the association score, and the last column mentions the effect sign (a positive sign represents an increase of the MIC).

antibiotic	locus_name	allele	position (bp)	frequency	p-value	q-value	effect sign
Amikacin	mexX	locus	NA	9.85E-01	5.97E-10	4.70E-06	+
Amikacin	gyrA	c	247	6.06E-01	4.09E-09	1.61E-05	-
Amikacin	mexY	locus	NA	9.79E-01	3.39E-08	8.91E-05	+
Amikacin	OXA	locus	NA	1.50E-01	2.01E-06	3.96E-03	+
Colistin	mexS	locus	NA	9.55E-01	1.23E-05	2.93E-02	+
Colistin	ampD	-	0	2.68E-02	2.38E-05	2.93E-02	+
Cefepime	OXA	a	545	2.23E-02	1.46E-05	3.61E-02	+
Cefepime	OXA	a	622	1.25E-01	2.57E-05	3.61E-02	+
Cefepime	OXA	g	644	1.26E-01	2.57E-05	3.61E-02	+
Cefepime	OXA	a	767	2.38E-02	2.57E-05	3.61E-02	+
Meropenem	gyrA	c	247	6.06E-01	8.02E-13	6.33E-09	-
Meropenem	oprD	c	1266	2.13E-01	2.67E-07	6.04E-04	-
Meropenem	parC	c	259	7.02E-01	2.72E-07	6.04E-04	-
Meropenem	oprD	g	1200	1.98E-01	3.13E-07	6.04E-04	-
Meropenem	oprD	c	1283	2.35E-01	4.59E-07	6.04E-04	-
Meropenem	oprD	g	1278	2.17E-01	4.59E-07	6.04E-04	-
Meropenem	oprD	g	1343	2.17E-01	8.32E-07	9.36E-04	-
Meropenem	oprD	c	1134	2.37E-01	1.28E-06	1.26E-03	-
Meropenem	oprD	a	758	8.04E-02	3.98E-06	3.48E-03	-
Meropenem	oprD	c	180	7.59E-02	7.16E-06	5.51E-03	-
Meropenem	oprD	c	1182	2.44E-01	9.59E-06	5.51E-03	-
Meropenem	oprD	c	641	7.20E-01	9.79E-06	5.51E-03	+
Meropenem	oprD	c	650	2.41E-01	9.79E-06	5.51E-03	-
Meropenem	oprD	c	875	2.40E-01	9.79E-06	5.51E-03	-
Meropenem	oprD	c	712	7.38E-01	1.07E-05	5.62E-03	-
Meropenem	oprD	c	522	7.32E-01	1.39E-05	6.87E-03	+
Meropenem	oprD	a	1082	2.38E-01	2.10E-05	9.52E-03	-
Meropenem	oprD	c	683	7.44E-01	2.30E-05	9.52E-03	-
Meropenem	oprD	a	827	2.17E-01	2.30E-05	9.52E-03	-
Meropenem	oprD	c	386	7.17E-01	2.79E-05	1.08E-02	+
Meropenem	ampO	a	892	8.93E-03	2.88E-05	1.08E-02	-
Meropenem	oprD	a	437	2.26E-01	3.15E-05	1.08E-02	-
Meropenem	oprD	g	446	7.35E-01	3.15E-05	1.08E-02	-
Meropenem	oprD	c	396	2.51E-01	3.46E-05	1.14E-02	-
Meropenem	oprD	c	410	7.26E-01	4.88E-05	1.43E-02	+
Meropenem	oprD	a	1051	9.58E-01	5.80E-05	1.63E-02	-
Meropenem	oprD	c	1336	7.29E-01	6.19E-05	1.68E-02	-
Meropenem	oprD	c	55	9.43E-01	6.45E-05	1.69E-02	-
Meropenem	mexZ	-	489	1.79E-02	7.38E-05	1.84E-02	+
Meropenem	oprD	a	1200	7.13E-01	7.50E-05	1.84E-02	-
Meropenem	oprD	c	1173	7.40E-01	7.69E-05	1.84E-02	-
Meropenem	oprD	c	1309	7.35E-01	1.16E-04	2.69E-02	-
Meropenem	oprD	c	1330	4.52E-01	1.23E-04	2.77E-02	+
Meropenem	mexX	c	1185	9.81E-01	1.45E-04	3.18E-02	-
Meropenem	mexD	c	1752	9.91E-01	1.85E-04	3.94E-02	+

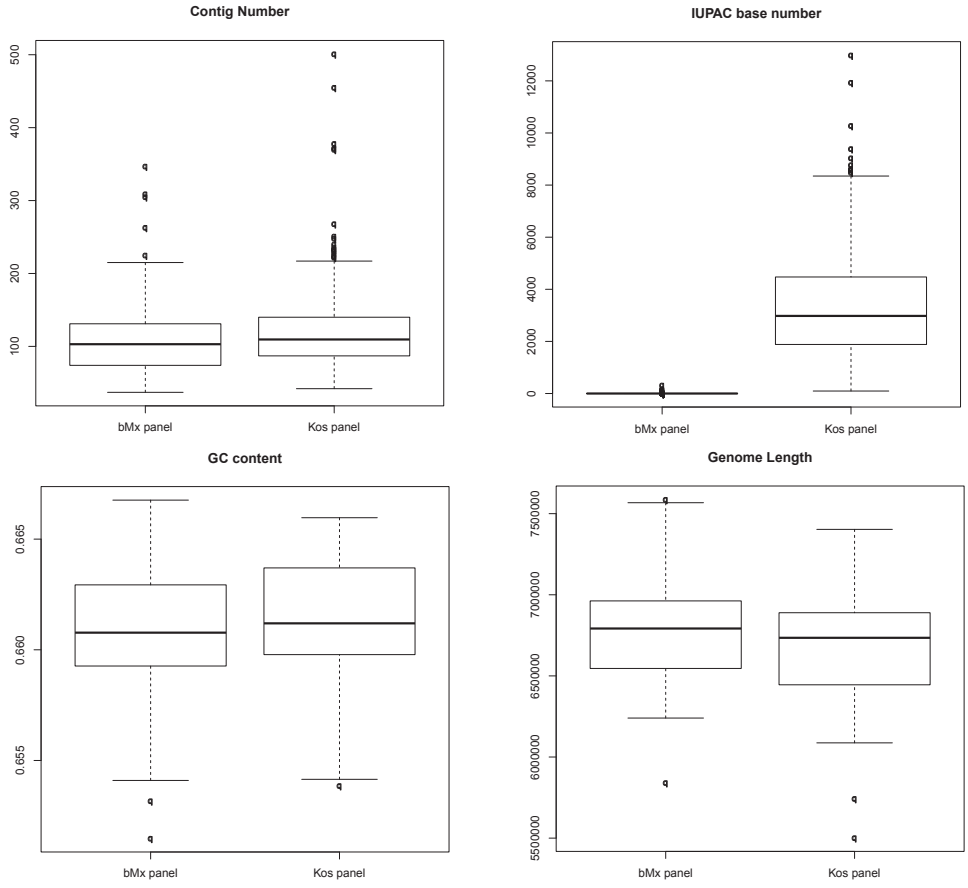


Supplementary Figure S1.5: **MIC distribution.** This table shows for each drug, the number of phenotyped isolates (column "N"), the number of isolates within each CLSI category (columns "I", "R" and "S"), and the number of isolates found for each of the MIC category (from 0.015625 5µg/ml to 512 µg/ml). Cells with a zero count are colored in green while other cells show different shade of orange, darker when the isolate count is higher. Cells with a yellow border show, for each drug, the CLSI breakpoint distinguishing S from I or R strains.

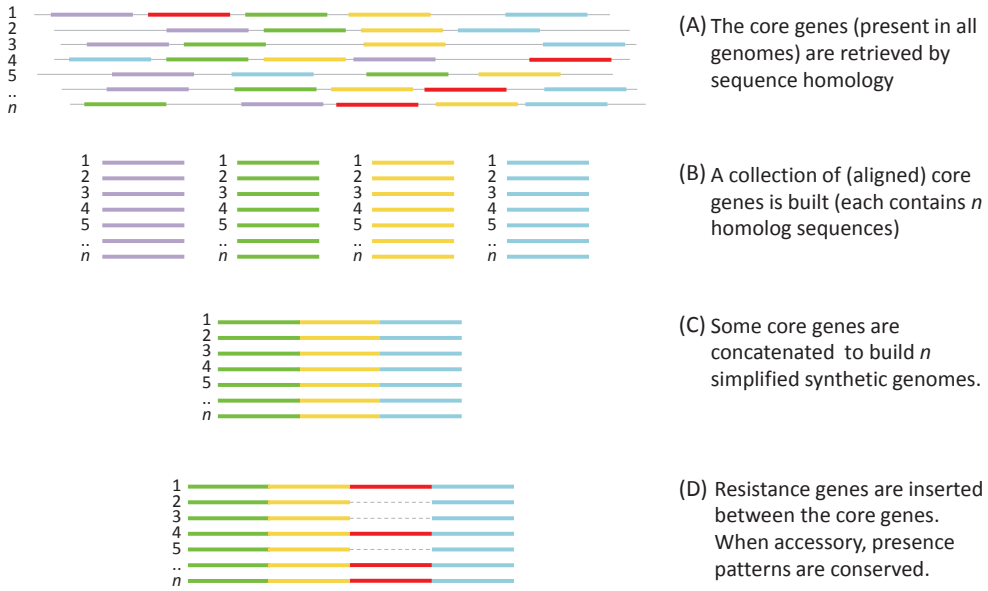
(A) Core genes are defined from 15 annotated reference genomes, and are then searched by homology in the assemblies



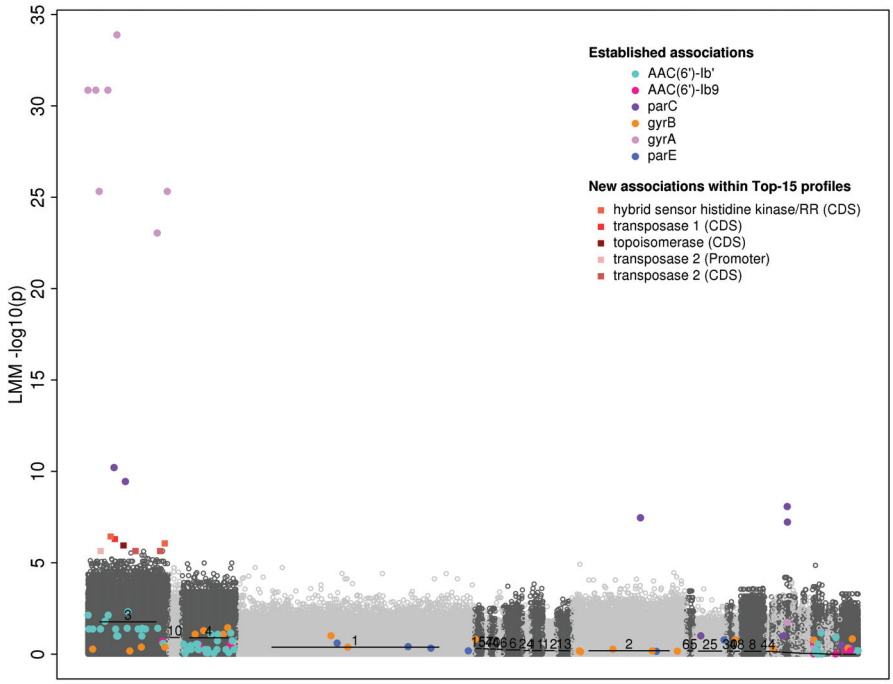
(B) The distribution of several characteristics of the assemblies is compared



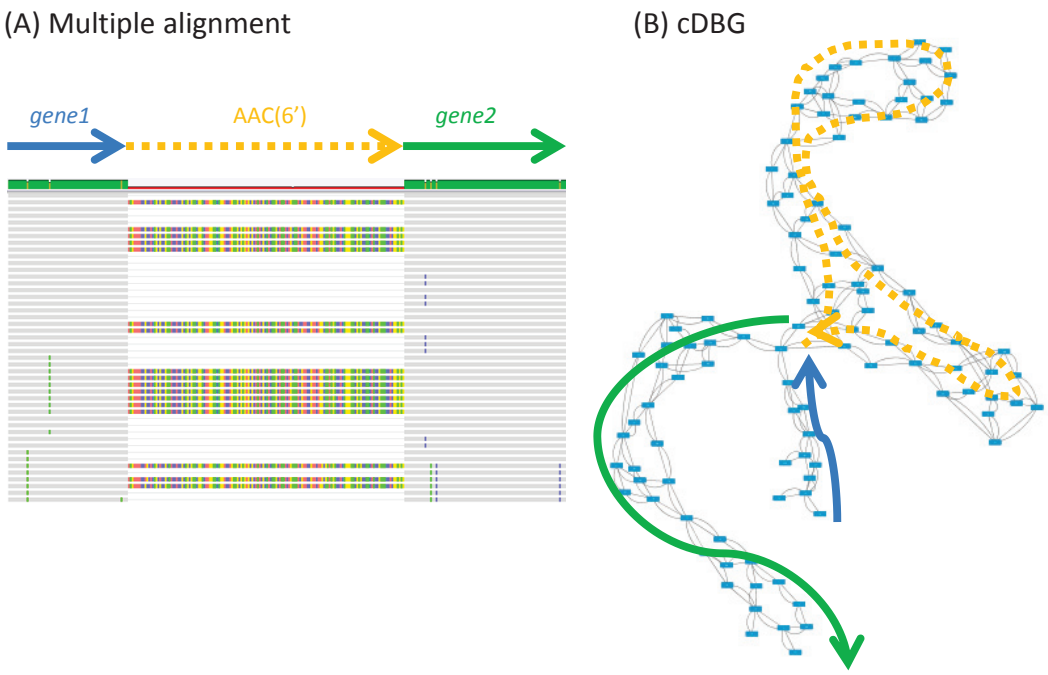
Supplementary Figure S1.6: **Quality of the assemblies.** (A) We observed an important difference between Kos and the other collections (Pirnay and bioMérieux, mentioned as 'VTK/SGI') in terms of retrieving the core genes, by homology (blast with 70% identity and T0% coverage). (B) The distribution of several characteritics was plotted to compare between Kos and the other panels (mentioned as 'bMx'): the IUPAC base (number of non-canonical bases) is significantly higher in the Kos assemblies.



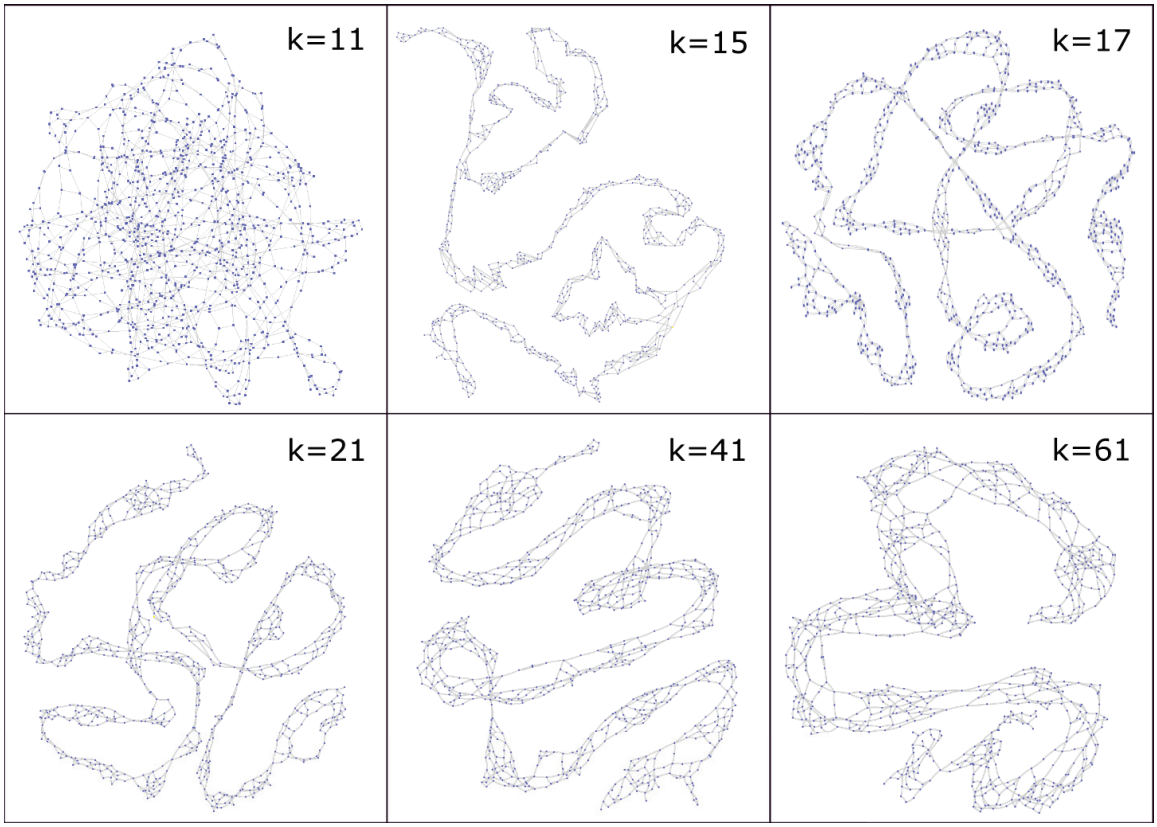
Supplementary Figure S2.1: **Simulation of simplified genomes using a core gene collection.** This figure illustrates how synthetic simplified genomes were build from a set of assemblies and annotations. This process was used to evaluate the bubble approach (steps (A) to (D), in Chapter 2) as well as to build the core genomes used to produce simulated graphs (steps (A) to (C), in Chapter 4).



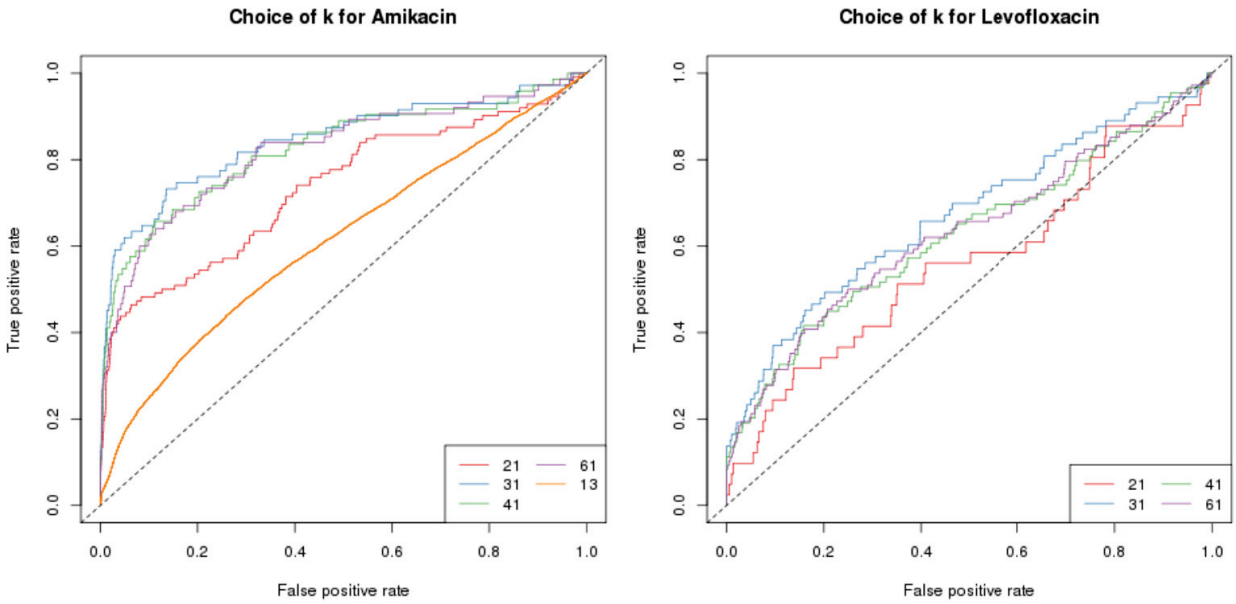
Supplementary Figure S2.2: `bugwas` **applied to the unitig patterns, for the levofloxacin resistance phenotype.** This annotated Manhattan plot provides the $-log_10$ p-values computed by the linear mixed model implemented in `bugwas` on the unitigs of a pan-cDBG built with $k = 31$. Unitigs are ordered by PCs [61], and annotations were added to highlights the genuine levofloxacin variants and the new variant found within the 15 first patterns.



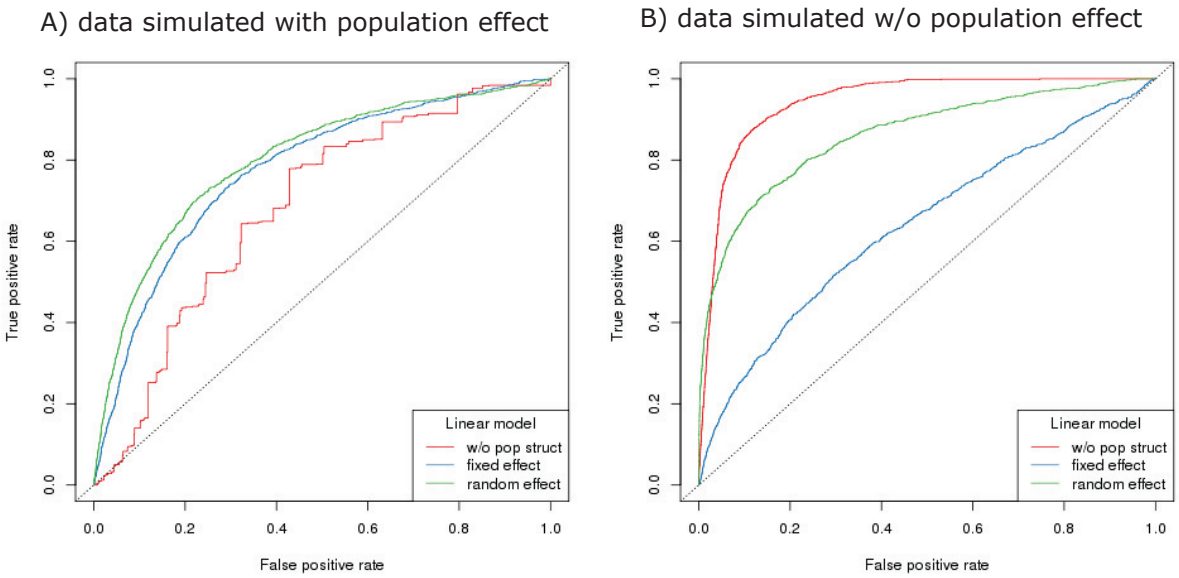
Supplementary Figure S2.3: **Multiple alignment (A) and cDBG (B) obtained for the insertion of the accessory gene variant family AAC(6') between two genes.**



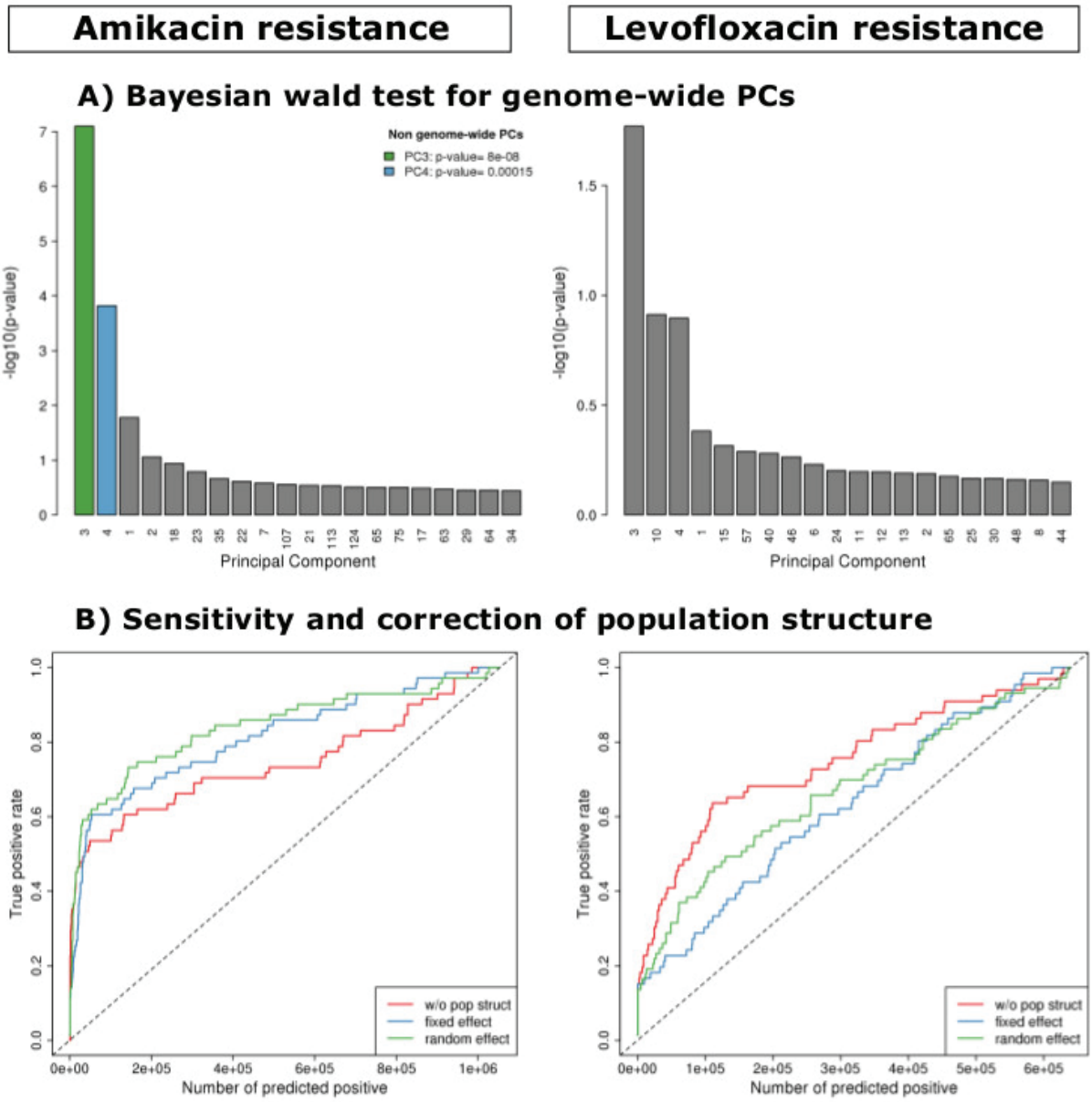
Supplementary Figure S2.4: **Effect of k on the graph topology** A cDBG was built from *P. aeruginosa gyrA* gene sequences. When k is small, k-mers are highly repeated, which generate numerous loops. As k increases, k-mer sequences becomes more specific and the graph gets more linear. For large values of k, few k-mers are shared by all the strains and the linear path thickens into parallel paths belonging to variable strain populations.



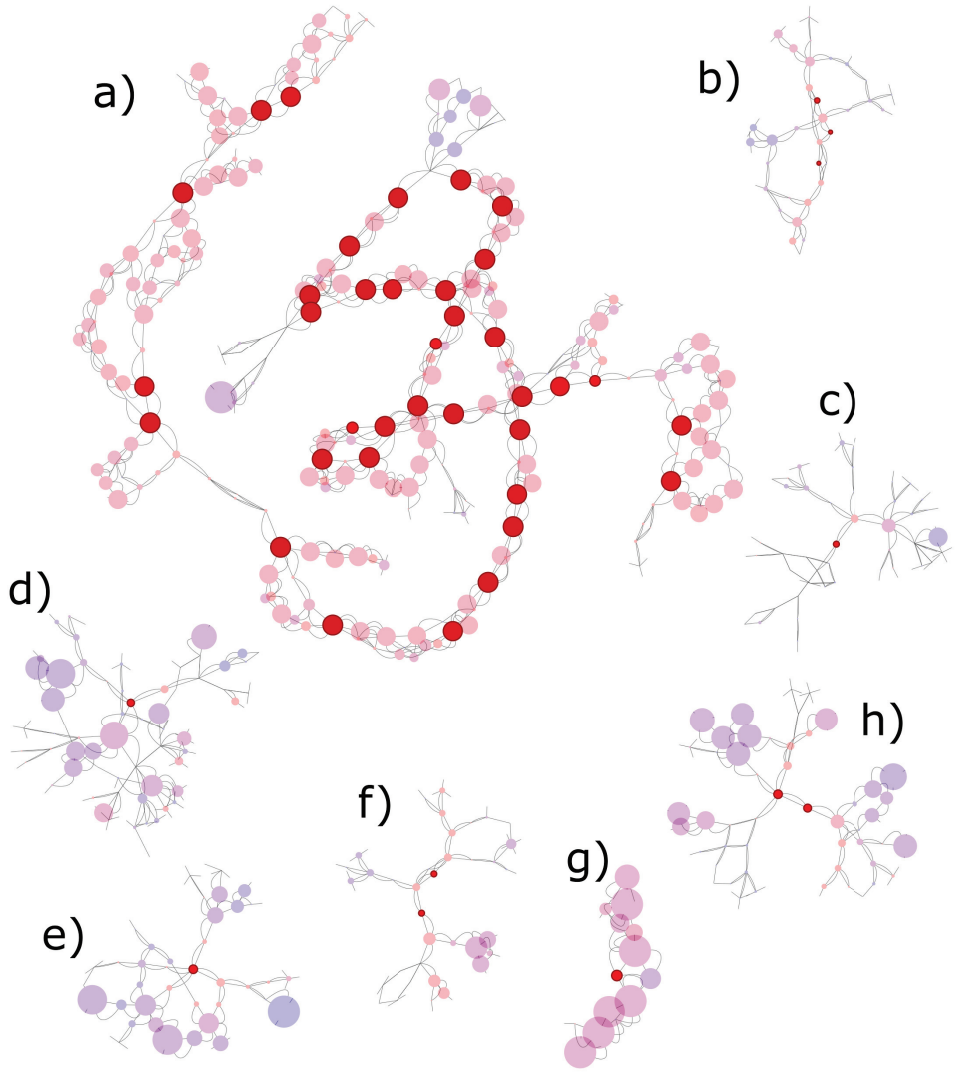
Supplementary Figure S2.5: **Choice of k.** True positive *versus* false positive curves for several values of k for both amikacin and levofloxacin resistance phenotypes. True positives are unitigs mapping to genuine variants described in resistance databases for the studied drugs. In both cases, the value of k leading to the best AUC is $k = 31$.



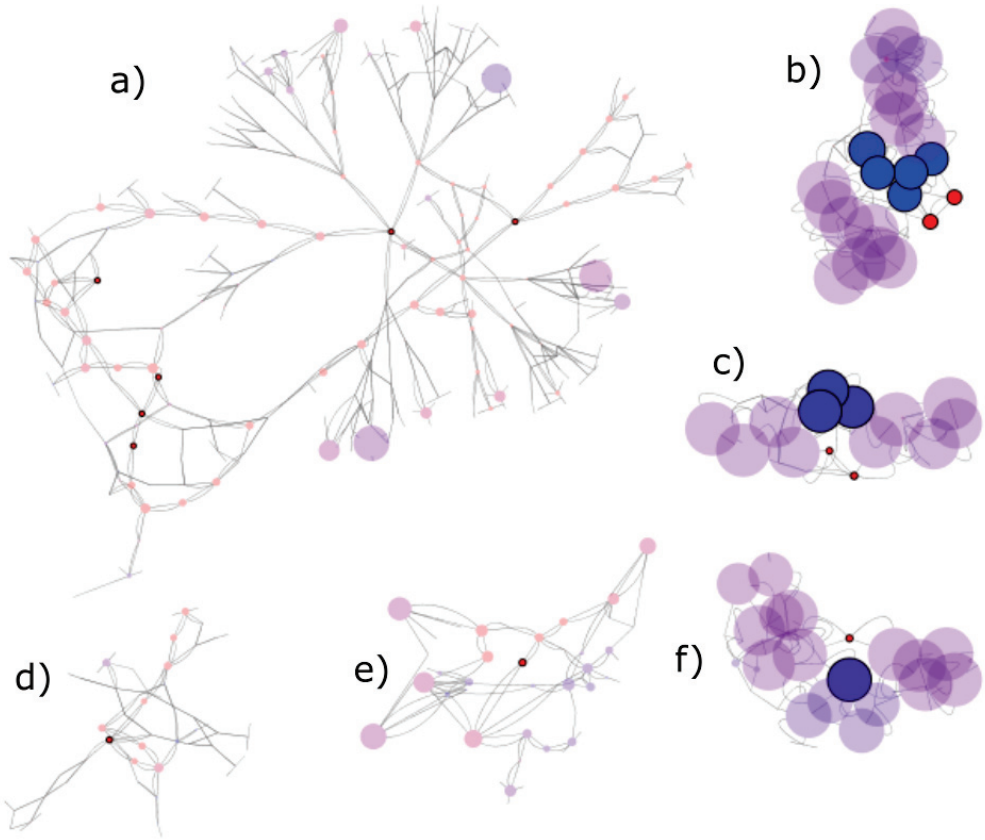
Supplementary Figure S2.6: **GWAS results of simulated data.** Scenario A and B intend to illustrate the model ability to detect true positives when there is or not population effect on the observed resistance. In the
first case, the ROC curve shows that taking the population structure into account in the model improves the power, while in the second case, correcting for the population structure when there is not, decreases the power to detect true determinants. Using a random effect model is however more robust and loses less power than using a
fixed-effect model, and thus can be a good compromise for any situation.



Supplementary Figure S2.7: **Gwas results.** Panel A shows the $-log_{10}$ association p-value of each principal component (representing population structure) with resistance. Panel B shows the proportion of true positive versus the number of predicted positive obtained by sorting candidates by p-value from three testing procedure: not correcting for population structure, or correcting using a fixed or a random effect term. Each panel shows results for both amikacin and levofloxacin resistance.



Supplementary Figure S2.8: **Genomic neighborhood of the top 15 amikacin resistance-associated profiles.** Representation of the 5-neighboring DBG for amikacin GWAS results. Coloration ranges continuously from blue (0% of resistant strain) to red (100% of resistant strain). Node size is proportional to allele frequency. Annotation identified component a) as plasmid pHS87b (extented in Figure 2.13B), b) hypothetical protein, upstream an integrase , c) transposase, d) helicase, e) DEAD/DEAH box helicase, f) non-coding region close to a transposase, g) SNP in AAC gene, and (extented in Figure 2.13A) h) non-coding region close to an integrase (extented in Figure 2.13C).



Supplementary Figure S2.9: **Genomic neighborhood of the top 15 levofloxacin resistance-associated profiles.** Representation of the 5-neighboring DBG for levofloxacin GWAS results. Coloration ranges continuously from blue (0% of resistant strain) to red (100% of resistant strain). Node size is proportional to allele frequency. Annotation identified component a) promoter and 5 region of transposase the multiple loops in this component seem to relate to a repeated region, which is in accordance with the multiple transposase homologs represented in all genomes, b) SNP in *gyrA* (extented in Figure 2.13D), c) SNP in *parC* (extented in Figure 2.13E), d) transposase CDS, e) topoisomerase, and f) SNP in hybrid sensor histidine kinase/response regulator (extented in Figure 2.13F).

Appendix S2.1: Comments from ISMB referees.

Reviewer 1 (-2)

In the paper entitled "Representing Genetic Determinants in Bacterial GWAS with Compacted De Bruijn Graphs", Jaillard et al. consider the problem of performing GWAS on bacterial strains for detecting associations between genetic markers and antimicrobial resistance (bacterial GWAS). The authors suggest a new approach for data preparation / representation prior to performing the GWAS. Essentially, the authors construct a compressed De Bruijn graph (DBG) from the assembled genomes of the strains in the sample, and define a binary indicator variable ("pattern") for each node in the graph ("unitig"), indicating for each sample whether it has this genomic component in the genome or not. Finally, each (unique) pattern can be tested for an association with a phenotype of interest, and the DBG representation can be used for visualization and interpretation. I believe the problems introduced by bacterial GWAS are interesting and important; therefore developing methods for improving the analysis of such data is of interest. However, I have quite a few concerns with this paper.
Major issues:
1. I think the authors overrated most of the advantages they discussed about the unitigs representation. First, Jaillard et al. essentially test for associations between the phenotype of interest and unique patterns. As mentioned by the authors several times in the paper, either using kmers (e.g., as in Earle et al.) or using their suggested unitigs representation provides the same set of unique patters to test. Therefore, the association tests and results are the same for both approaches. Second, the authors suggest that working with unitigs is more efficient for interpretation of the results and consider working with kmers as cumbersome: "Assuming a pattern is found to be associated with resistance in our test, its interpretation in a fixed-length kmer paradigm can be cumbersome: it typically requires to map all kmers corresponding to the pattern to all genomes – as there is no single reference genome in this context – and to make sense of these mappings." However, it sounds like an implementation issue that can be fairly easily solved in the kmers approach using dictionaries: each unique kmer used can map to the genomes and genomic locations from which it was extracted. In addition, a reverse dictionary can be generated to map between each unique pattern to the set of its corresponding kmers (during the construction of the patterns). The authors also suggest that using unitigs provide advantage in the interpretation step. They demonstrate this using a nice set of figures, which make the point that using unitigs indeed provides a good approach for visualization of the results. Yet, the authors did not provide an alternative visualization based on the kmers approach.
2. Core details are missing:
- Did the authors filter out genetic variants prior to analysis based on low minor allele frequency (MAF)? If so, what threshold was used? Fig. 4c suggests that the authors may have filtered out unitigs with MAF $< 2\%$. I suspect this threshold is not stringent enough given the sample size used: consider the levofloxacin phenotype for which values were available for 117 strains. The 2% threshold in this case means that some patterns should have only a very small number of occurrences of the minor allele (e.g., 3). In this case, the probability that these 3 samples have relatively extreme phenotypic values merely by a chance (and therefore yield a significant p-value) is relatively high. Since many patterns are tested, I would expect to see some spurious associations only based on this low MAF threshold. The authors can confirm that using permuted data where no true associations are anticipated. This approach can be also used to determine an appropriate MAF value.
- The authors didn't report p-values for their experiments. If the p-values are deflated

because of relatively small sample size then it raises a sever concern about the validity of the real data results. Moreover, even though the results presented in 3.3.3 seem to be interesting, the authors should present an enrichment test, or otherwise perform some sort of permutation test for verification.
- What are the lengths of the genomes in the real data analysis (the distribution)? How many unique patterns were generated?
3. The authors consider a linear model for the association testing in spite the fact that their outcomes are binary. The authors reason this by reporting worse detection performances for a logistic model in a preliminary experiment. Since a logistic model is more natural and statistically justified, I suspect that lower detection performance of a logistic model implies that it adjusts for the population structure confounder better and therefore it may have a better control for false positives. Of course, it is not possible to assess this a long as no additional details are provided. I would recommend the authors to provide more details here, possibly show an experiment, to justify their use of a linear model over a logistic one.
4. I believe Fig. 5 is somewhat misleading – most of the different kmers that are represented by the same patterns are likely to be from the same genomic region (if that's not true the authors should show that). Since the interpretation of the results in the kmers approach is also eventually made on the genomic locations of the kmers, it makes little sense to consider several false positives for the kmers approach in case of a false positive pattern.
Minor / Technical issues:
- The authors refer to several Supplementary Tables throughout the main text; however, I couldn't find any of those in the supplementary file.
- "A specific aspect of our approach is that we build our compacted DBG from assembled genomes (more precisely, from contigs) rather than from primary sequence reads." – this should be mentioned earlier in the text for clarity.
- There are several content repetitions in the text- for example, the explanation about the potential effect in the presence of population structure. In summary, I feel that a more reasonable approach would be to perform Bacterial GWAS based on unique patterns using the kmers approach (as in Earle et al.). A careful implementation of this approach should be more efficient than constructing DBG. Then, given a set of interesting patterns, the kmers they map to can be used to construct a compressed DBG, as proposed here by Jaillard et al. I believe this approach would be simpler and more efficient and can still benefit from the improved visualization and ease of interpretation provided by DBG approach.

Reviewer 2 (-1)

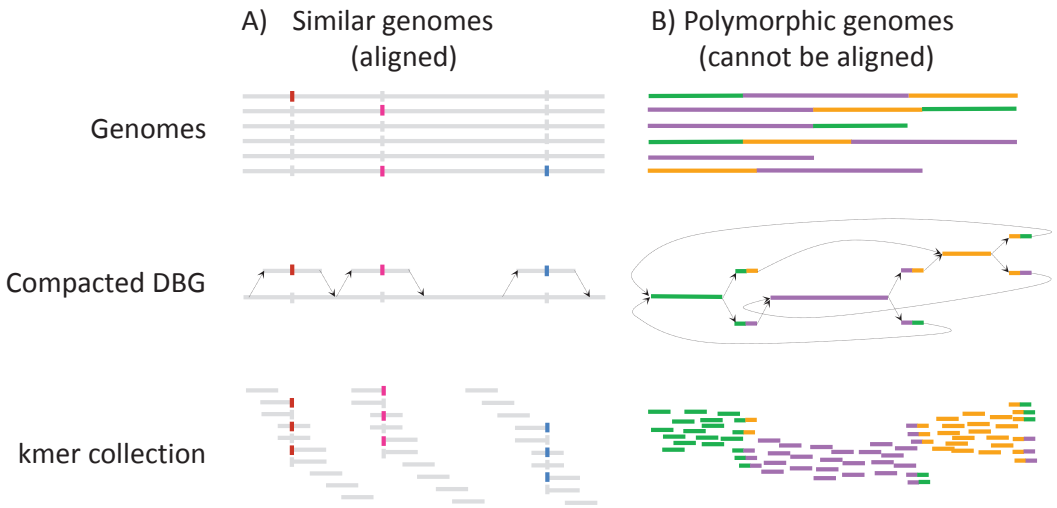
Jaillard *et al* proposed a very interesting idea to represent genetic determinants with compacted de Bruijn graphs. The genetic markers for antimicrobial resistance can be discovered through GWAS approaches. Traditional GWAS approaches aim to discover significant SNPs or k-mers that are corresponding to specific phenotypes. Instead, the authors applied GWAS on "unitigs", which are variable-length k-mers represented by the collapsed nodes in compacted de Bruijn graphs. The usage of unitigs subtly resolves the issue of choosing the appropriate "k" in k-mer based approach and provides a more meaningful interpretation than the SNP approach. The idea of "unitigs" is interesting and presents several advantages over existing approaches; however, I have several concerns as mentioned below:
Main concerns:

1) The authors claimed that they proposed an alignment-free GWAS method. However, in the method section, they mentioned that the unitigs from the newly assembled DBG are aligned against the original genome assemblies using Bowtie2. The claim of "alignment-free" seems to contradict with the proposed method.
2) The major disadvantage of SNP-based approach is the challenge of aligning highly plastic genomes to a single reference genome. The proposed approach built the compacted DBG from assembled genomes of multiple strains. Isn't it true that the SNP-based approach can be simply improved by using a meta-reference constructed from multiple strains as well? A direct comparison between these two ideas would make it easier to evaluate the advantages of unitigs. The SNP-based approach was mentioned multiple times in the introduction, but there wasn't any experiment conducted directly on the SNP-based approach.
3) The method is evaluated on a highly plastic genome, P. aeruginosa, with 282 strains. However, it is still within a single species. It would be more interesting to see how the method performs across different species.
4) In the introduction, the authors stated that general k-mer approaches require more computational resources, such as more memory for storage and more time for loading and interpretation. However, the authors never demonstrated the memory usage and running time of the unitigs approach. It is difficult to judge the performance of this approach.
5) The simulated dataset is generated using a multivariate logistic model, but the proposed solution uses multivariate linear regression. Section 2.2 claimed that logistic model did not perform well on preliminary experiments, but the authors never discussed the reasons for this observation.
Minor concerns:
1) Even if it is a popular jargon, "plastic genome" could be described or explained at least once in the paper.
2) S2.1, "constant kmer" is a confusing term. I think it refers to "shared" kmers, which is being used in the following paragraph.
3) S2.1, "in the remainder" sounds weird. It should be "in the remainder of the paper".
4) S2.1 "adaptive fashion existing representations" -> adaptive fashion "of the" existing representation.
5) The axis fonts and legends are too small to read in Figures 4 and 5.
6) Some of the references are not consistent. For example, both abbreviation and full journal names are seen (Nat. methods and Nature methods).

Reviewer 3 (0)

The authors present an approach based on compacted De Brujin graphs to perform GWAS in bacteria. The approach is very interesting, and might be of use for bacterial GWAS. However, the presentation of the results suffers from several drawbacks.
Major:
1) No comparison with previously developed approaches is presented. Authors only compare results obtained without adjusting for population structure and including population structure as a random or fixed effect.
2) Figure 5 is hard to understand: On the x-axis you have up to 1 Million positive calls (which is quite a high number for bacterial genomes). On the y-axis you plot the true positive rate. However, since true positive rate is TP/P, you should have found more than 800,000 reported causal variants. Or maybe, you computed TPR using a number of Positives different from the values used on the x-axes.

3) It is difficult to understand the performance of the approach, since no data is presented. How many significant associations were observed in the simulations? How many were expected?
4) No detail is given on the identification of reported causal variants. How many reported causal variants were detected? How many unreported?
Minor:
What is the meaning of nodes in Figure 4, panels B and C? And where are the unitigs represented in that figures? Maybe thew word node is used instead of unitigs?
"Furthermore as discussed in Section 2.3 several kmers or unitigs can have the same presence/absence pattern on a given set of genomes, so we also represent the proportion of kmers or unitigs which are filtered out from our GWAS because they correspond to duplicated kmers or unitigs (light grey)"
I don't understand why kmers (unitigs) with the same PA pattern in a set of genomes should be duplicated. I thought that they are simply conserved (or in other words not polymorphic). Am I missing something? Authors should explain this, since readers can have my same difficulties in understanding this statement.
Mentionned → mentioned
to any know genetic → to any known genetic
there are still unreported → these are still unreported



Supplementary Material S3.1: **Alignment to a reference (when possible), cDBG, and k-mers obtained for similar (A) and very polymorphic genomes (B).** In the first case, the 3 loci represented as polymorphic in the alignment lead to 3 bubble patterns in the cDBG, and numerous redundant k-mers. In the second case, genomes are so polymorphic that an alignment is not possible. The cDBG summarizes well the common regions and the links between them, while the collection of unique k-mers still contains redundancy.

Phenotype (#subgraph with min_q <0.05 / #subgraphs)	Subgraph rank	#sign. unitigs over all subgraph unitigs	Unitig with the minimal q-value				Genetic event	Annotation	Comment : green = has been described for the studied phenotype orange = has been described for another phenotype white = not described for resistance	References	SNP rank (if found by the resistome SNP approach) and q-value if not in Table 3
			min_q	Estimated effect	count per phenotype: Pheno0 (susceptible)	count per phenotype: Pheno1 (non-susceptible)					
Rifampicin	1	36/115	4.84E-70	-5.77E-01	518/531	271/666	hot-spot in a core gene	*rpoB* RRDR	hot-spot region described for rifampicin resistance.	Palomino, 2014	1, 4, 5, 8
	2	6/37	4.35E-20	-3.55E-01	455/531	118/666	SNP in a core gene	*katG*	described for isoniazid resistance	Palomino, 2014	2, 3
(6/25)	3	5/41	4.02E-08	-2.24E-01	507/531	258/666	SNP in a core gene	*embB* M306V	this mutation was described for ethambutol resistance, but also for rifampicin resistance	Shi, 2007	6, 7
	4	1/19	1.79E-02	7.27E-01	373/531	445/666	SNP in a core gene	Rv0226c	transmembrane protein		*not found*
	5	4/65	3.90E-02	-6.57E-01	16/531	58/666	hot-spot in a core gene	*gyrA* QRDR	hot-spot region described for quinolones resistance	Palomino, 2014	9
Isoniazid	1	9/51	4.84E-119	-7.21E-01	460/472	124/815	SNP in a core gene	*katG*	described for isoniazid resistance	Palomino, 2014	1, 2
	2	30/113	6.36E-25	-3.96E-01	451/472	271/815	hot-spot in a core gene	*rpoB* RRDR	hot-spot region described for rifampicin resistance.	Palomino, 2014	3, 10
	3	6/44	1.01E-08	-2.44E-01	462/472	364/815	SNP in a core gene	*embB* M306V	described for ethambutol and rifampicin resistance	Shi, 2007	4, 5, 7, 9
(11/31)	4	6/36	1.75E-03	-1.96E-01	460/472	515/815	SNP in a core gene promoter	*mabA(fabG1)-inhA*	described for ethionamide and isoniazid resistance	Lee, 2000	71 (2.40E-14)
	5	1/22	8.39E-03	4.89E-01	459/472	805/815	SNP in a core gene	Rv3812	PE-PGRS family protein PE_PGRS62	*also found for MDR*	*not found*
	6	3/26	1.63E-02	-4.23E-01	470/472	796/815	SNP in a core gene	*inhA*	described for ethionamide and isoniazid resistance	Lee, 2000	101 (7.26E-07)
Ethambutol	1	6/44	2.63E-23	-3.94E-01	618/709	126/332	SNP in a core gene	*embB* M306V	described for ethambutol and rifampicin resistance	Shi, 2007	1
	2	21/87	5.55E-23	-3.84E-01	565/709	64/332	hot-spot in a core gene	*rpoB* RRDR	hot-spot region described for rifampicin resistance.	Palomino, 2014	3, 6
	3	14/61	6.90E-15	-3.67E-01	683/709	200/332	hot-spot in a core gene	*gyrA* QRDR	hot-spot region described for quinolones resistance	Palomino, 2014	7
(23/23)	4	6/37	4.05E-12	-2.78E-01	494/709	50/332	SNP in a core gene	*katG*	described for isoniazid resistance	Palomino, 2014	4, 5
	5	6/36	6.93E-06	-2.51E-01	658/709	201/332	SNP in a core gene promoter	*mabA(fabG1)-inhA*	described for ethionamide and isoniazid resistance	Lee, 2000	12 (7.30E-05)
	6	4/33	8.79E-06	2.72E-01	32/709	112/332	SNP in a core gene	*rrs*, 16S rRNA A1401G	described for kanamycin resistance	Palomino, 2014	8
	7	8/34	3.04E-05	-3.24E-01	697/709	284/332	SNP in a core gene	*embB* Q497R	described for ethambutol resistance	Palomino, 2014	16 (4.91E-04)
	8	2/21	1.43E-04	-2.33E-01	639/709	223/332	SNP in a core gene	*rpsL* (30S ribos.protein S12)	described for streptomycin resistance	Palomino, 2014	9
	9	3/34	2.12E-04	-3.84E-01	706/709	274/332	SNP in a core gene	*ubiA* (Rv3806c)	described for ethambutol resistance	Safi, 2013	*not found*
Streptomycin	1	5/30	3.70E-31	5.44E-01	660/677	310/489	SNP in a core gene	*rpsL* (30S ribos.protein S12)	described for streptomycin resistance	Palomino, 2014	1, 4, 5
	2	6/37	1.06E-28	-4.28E-01	509/677	73/489	SNP in a core gene	*katG*	described for isoniazid resistance	Palomino, 2014	2, 3
	3	25/113	2.87E-16	-3.39E-01	536/677	131/489	hot-spot in a core gene	*rpoB* RRDR	hot-spot region described for rifampicin resistance.	Palomino, 2014	6, 9
(17/17)	4	6/45	1.40E-09	-2.71E-01	582/677	216/489	SNP in a core gene	*embB* M306V	described for ethambutol and rifampicin resistance	Shi, 2007	13 (1.61E-04)
	5	8/31	2.86E-09	-5.35E-01	670/677	450/489	SNP in a core gene	*rrs, 16S rRNA* C517T	described for streptomycin resistance	Palomino, 2014	7, 8, 10
	6	13/69	9.18E-05	-2.16E-01	615/677	325/489	hot-spot in a core gene	*gyrA* QRDR	hot-spot region described for quinolones resistance	Palomino, 2014	12 (1.51E-04)
	7	2/20	9.43E-04	7.46E-01	1/677	19/489	SNP on core gene	*espG1*	Cell wall and cell processes		*not found*
	7	3/20	9.43E-04	7.46E-01	1/677	19/489	SNPs in a core gene	30S ribosomal protein S14			*not found*
Kanamycin	1	4/38	1.24E-77	8.13E-01	8/484	144/187	SNP in a core gene	*rrs, 16S rRNA* A1401G	described for kanamycin resistance	Palomino, 2014	1
	2	2/28	1.72E-09	6.91E-01	0/484	62/187	SNP in a core gene	*rpoB* I1187T (outside the RRDR)	Position not described in the litterature.	*also found for XDR*	3
	2	3/27	1.72E-09	6.91E-01	0/484	62/187	SNP in a core gene	Rv2000	near a putative transposase for insertion sequence element	*also found for XDR*	*not found*
(15/28)	4	15/102	2.26E-09	6.28E-01	1/484	66/187	hot-spot in a core gene	*rpoB* RRDR	hot-spot region described for rifampicin resistance.	Palomino, 2014, *also found for XDR*	2, 6, 7
	5	3/24	1.30E-07	6.45E-01	1/484	62/187	SNP in a core gene promoter	*espA*	cell wall and cell processes	*also found for XDR*	*not found*
	5	13/724	1.30E-07	6.45E-01	1/484	62/187	pattern in a noncoding region	upstream Rv3122 + each side of repeated transposase for IS6110.	IS6110 is an inserted sequence used in the diagnosis of TB and in TB epidemiology. It may be associated due to lineage effect	Millan, 2013, *also found for XDR*	*not found*
	7	2/16	7.18E-07	5.76E-01	0/484	60/187	SNP in a core gene promoter	Rv0571c	Function unknown	*also found for XDR*	*not found*
	8	10/62	3.89E-06	5.93E-01	2/484	62/187	SNP in a core gene	*ubiA* (Rv3806c)	described for ethambutol resistance	Safi, 2013, *also found for XDR*	*not found*
	9	15/85	1.50E-04	3.16E-01	21/484	85/187	hot-spot in a core gene	*gyrA* QRDR	hot-spot region described for quinolones resistance	Palomino, 2014, *also found for XDR*	4
Ofloxacin	1	31/85	9.66E-144	-8.88E-01	447/458	26/238	hot-spot in a core gene	*gyrA* QRDR	hot-spot region described for quinolones resistance	Palomino, 2014 - Hooper, 2015	1, 2
	2	9/68	1.59E-04	5.07E-01	1/458	61/238	SNP in a core gene	*ubiA* (Rv3806c)	described for ethambutol resistance	Safi, 2013	*not found*
(4/47)	3	3/32	3.86E-02	-7.46E-01	456/458	232/238	SNP in a core gene	Rv3909	Function unknown		*not found*
	4	2/31	4.95E-02	-6.17E-01	456/458	232/238	SNP in a core gene	Rv0244c	Probable acyl-CoA dehydrogenase FadE5		*not found*
Ethionamide	1	9/39	7.86E-11	-4.62E-01	220/248	76/172	SNP in a core gene promoter	*mabA(fabG1)-inhA*	described for ethionamide and isoniazid resistance	Lee, 2000	4
	2	15/47	5.16E-10	-4.06E-01	212/248	60/172	hot-spot in a core gene	*gyrA* QRDR	hot-spot region described for quinolones resistance	Palomino, 2014	2
	3	4/26	5.55E-04	3.19E-01	32/248	96/172	SNP in a core gene	*rrs, 16* S rRNA A1401G	described for kanamycin resistance	Palomino, 2014	1
(16/42)	4	10/68	5.95E-04	-2.80E-01	159/248	37/172	hot-spot in a core gene	*rpoB* RRDR	hot-spot region described for rifampicin resistance.	Palomino, 2014	5, 8
	5	3/29	2.39E-03	-3.13E-01	221/248	90/172	SNP in a core gene	*embB* M306V	described for ethambutol and rifampicin resistance	Shi, 2007	7
	6	3/20	1.10E-02	-6.58E-01	239/248	111/172	SNP in a core gene	Rv0103c	Probable cation-transporter P-type ATPase B CtpB		*not found*
	6	10/71	1.10E-02	-7.38E-01	248/248	164/172	SNP in a core gene	*katG*	described for isoniazid resistance	Palomino, 2014	62 (4.36E-06)
MDR	1	31/110	2.45E-62	-5.38E-01	549/580	257/631	hot-spot in a core gene	*rpoB* RRDR	described for rifampicin resistance. Almost all rifampicin-resistant strains are also resistant to isoniazid.	Palomino, 2014 Traore, 2000	
defined by resistance to rifampicin and isoniazid	2	10/74	2.87E-44	-4.97E-01	497/580	86/631	SNP in a core gene	*katG*	Gene mutation in katG occurs more frequently in MDR strains	Palomino, 2014 Hazbon, 2006	
	3	5/41	9.81E-14	-2.88E-01	549/580	230/631	SNPs in a core gene	*embB* M306V	described for ethambutol and rifampicin resistance	Shi, 2007	
	4	1/19	7.63E-03	5.94E-01	569/580	628/631	SNP in a core gene promoter	Rv3333c	putative proline rich protein		
(5/26)	5	1/22	2.91E-02	4.58E-01	563/580	625/631	SNP in a core gene	Rv3812	PE-PGRS family protein PE_PGRS62	*also found for isoniazid*	
XDR	1	6/68	3.66E-39	9.05E-01	1/577	59/112	SNP in a core gene	*rpoB* I1187T (outside the RRDR)	I1187T. In LD with the 5 next sugraphs.	*also found for kanamycin*	
	1	3/27	3.66E-39	9.05E-01	1/577	59/112	SNP in a core gene	Rv2000	near a putative transposase for IS element	*also found for kanamycin*	
	3	3/24	9.58E-36	8.83E-01	2/577	59/112	SNP in a core gene promoter	*espA*	cell wall and cell processes	*also found for kanamycin*	
defined by resistance to rifampicin, isoniazid, kanamycin and ofloxacin	3	16/1118	9.58E-36	8.83E-01	2/577	59/112	pattern in a noncoding region	upstream Rv3122 + each side of repeated transposase for IS6110.	IS6110 is an inserted sequence used in the diagnosis of TB and in TB epidemiology. It may be associated due to lineage effect	Millan, 2013, *also found for kanamycin*	
	5	12/90	2.10E-34	8.21E-01	3/577	60/112	hot-spot in a core gene	*rpoB* RRDR	hotspot - mostly position 516	Palomino, 2014, *also found for kanamycin*	
	6	7/57	1.14E-32	8.58E-01	3/577	59/112	SNP in a core gene	*ubiA* (Rv3806c)	described for ethambutol resistance	Safi, 2013, *also found for kanamycin*	
	7	2/16	1.14E-32	8.44E-01	1/577	57/112	SNP in a core gene promoter	Rv0571c	Function unknown	*also found for kanamycin*	
(26/26)	8	14/80	2.05E-31	6.20E-01	17/577	75/112	hot-spot in a core gene	*gyrA* QRDR	hot-spot region described for quinolones resistance	Palomino, 2014 *also found for kanamycin*	

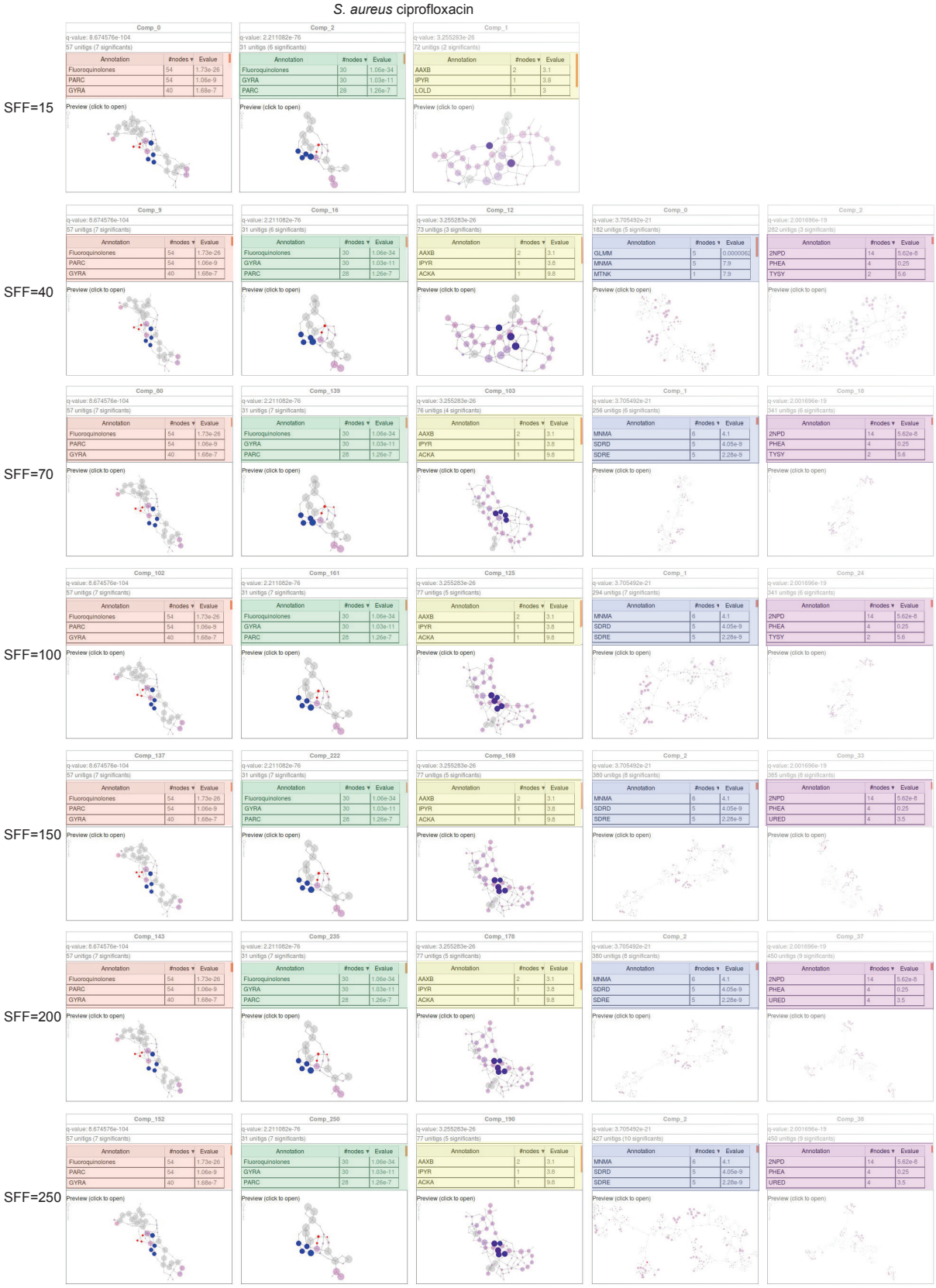
DBGWAS results for *M. tuberculosis* resistance to antibiotics. For each antibiotic, top subgraphs were reported with their rank, the q-value of the unitig with the lowest q-value (min_q), the corresponding estimated effect (estimated β of the linear model) and the number of susceptible (resp. resistant) strains harbouring this unitig (count per phenotype). The type of event represented by the subgraph, its annotation and some comments and references on this annotation were also provided. Comments were coloured if the annotation was previously described in antibiotic resistance literature: in green if this description concerned the tested antibiotic, in orange otherwise.

Phenotype (#subgraph with min_q <0.05 / #subgraphs)	Subgraph rank	#sign. unitigs over all subgraph unitigs	Unitig with the minimal q-value				Genetic event	Annotation	Comment : green = has been described for the studied phenotype orange = has been described for another phenotype white = not described for resistance	References
			min_q	Estimated effect	count per phenotype: Pheno0 (susceptible)	count per phenotype: Pheno1 (non-susceptible)				
Methicillin	1	71/565	7.68E-188	9.49E-01	2/342	158/159	MGE with gene annotation	*mecA* gene + neighborhood covering 7000pbs of the SCC*mec* cassette	mecA is the known causal gene for methicillin resistance	Gordon, 2014 ; Lowy, 2003
(4/4)	2	99/735	3.39E-72	8.65E-01	5/342	157/159	MGE	recombinase genes + neighborhood covering 6000pbs of SCC*mec*	complete SCC*mec* cassette in LD with *mecA* gene	IWG-SCC, 2009
	3	11/190	2.14E-61	8.13E-01	7/342	157/159	MGE	covers 2000pbs of SCC*mec*	complete SCC*mec* cassette in LD with *mecA* gene	IWG-SCC, 2009
	4	13/117	2.29E-37	9.57E-01	5/342	155/159	MGE	covers 1500pbs of SCC*mec*	complete SCC*mec* cassette in LD with *mecA* gene	IWG-SCC, 2009
Ciprofloxacin	1	7/57	8.67E-104	-8.93E-01	726/749	13/242	hot-spot in a core gene	*parC* QRDR	described for quinolones resistance	Gordon, 2014 ; Hooper, 2015
(163/163)	2	7/31	2.21E-76	9.55E-01	4/749	222/242	hot-spot in a core gene	*gyrA* QRDR	described for quinolones resistance	Gordon, 2014 ; Hooper, 2015
	3	5/77	3.26E-26	-5.38E-01	607/749	166/242	pattern in a noncoding region	upstream multidrug efflux MFS transporter *norA*	described for quinolones resistance	Hooper, 2015
	4	7/294	3.71E-21	8.99E-01	1/749	66/242	pattern in a noncoding region	between *glmM* and *fmtB* genes	genes described for methicillin resistance	Komatsuzawa, 2000
	5	6/341	2.00E-19	9.07E-01	0/749	65/242	pattern in a noncoding region	each side of a transposase	transposases required in the replication of transposon (MGE)	
Erythromycin (1/1)	1	110/510	2.69E-100	8.23E-01	8/775	113/216	MGE with gene annotation	*ermC* + replication and maintenance protein, representing a complete circular plasmid	described for erythromycin resistance. All significant unitigs fall in the same sugbraph, suggesting we could increase the SFF value to explore other regions.	Gordon, 2014
Penicillin	1	43/441	3.88E-77	9.34E-01	27/168	818/823	MGE with gene annotation	*blaZ*	*blaZ* inactivates the penicillin (beta-lactamase activity)	Gordon, 2014 ; Lowy, 2003
(14/14)	1	16/188	3.88E-77	9.24E-01	25/168	816/823	MGE with gene annotation	*blaR1*	*blaR1* regulates *blaZ* activity	Lowy, 2003
	3	31/332	2.26E-66	8.92E-01	23/168	812/823	MGE with gene annotation	*blaR1* (3' region) + *bla1* penicillinase repressor	*blaR1* regulates *blaZ* activity	Lowy, 2003
	4	2/76	1.99E-54	9.01E-01	26/168	813/823	MGE with gene annotation	*blaR1* (5' region)	*blaR1* regulates *blaZ* activity	Lowy, 2003
	5	2/52	1.53E-20	9.72E-01	149/168	634/823	pattern in a noncoding region	between an hypothetical protein and *thyA* gene (thymidylate synthase)		
Tetracycline (1/1)	1	142/816	2.62E-263	9.29E-01	3/945	42/46	MGE with gene annotation	*tetK* gene + replication initiation protein, and plasmid related proteins	*tetK* described for tetracycline resistance	Gordon, 2014
Fusidic acid	1	7/50	2.75E-136	-9.10E-01	906/907	67/84	polymorphic region in a gene	*fusA* (elongation factor G), target of fusidic acid	described for fusidic acid resistance	Gordon, 2014
(14/14)	2	214/882	7.94E-49	9.24E-01	0/907	49/84	MGE with gene annotation	SCC*fusC* cassette, including fusidic acid resistance protein (*fusC*) (subgraph covers 12,000 pbs, including ~14 CDS)	described for fusidic acid resistance	Chen, 2010
	3	22/260	5.35E-43	9.24E-01	0/907	48/84	MGE	covers 1,500 pbs of SCC*fusC*	sequence in LD with the causal gene	
	3	1/72	5.35E-43	9.24E-01	0/907	48/84	MGE	covers 200 pbs of SCC*fusC*	sequence in LD with the causal gene	
	5	5/64	2.02E-22	-8.88E-01	874/907	81/84	polymorphic region in a gene	*purN*		
Trimethoprim	1	7/54	8.38E-24	9.69E-01	0/308	5/15	SNP in a gene	*folA* (dihydrofolate reductase, target of the drug)	described for trimethoprim resistance	Gordon, 2014
(51/51)	2	3/41	9.30E-18	-9.66E-01	308/308	11/15	pattern in a noncoding region	between an hypothetical protein and a VOC family protein	gene *fosB* belongs to the VOC family and is described for fosfomycin resistance	Morar, 2010
	3	11/70	9.30E-18	-9.66E-01	308/308	11/15	polymorphic region in a gene	Cys-tRNA(Pro) deacylase (*ybaK*)		
	4	2/30	6.82E-10	-6.32E-01	306/308	11/15	polymorphic region in a gene	*mqo1*		
	5	9 subgraphs are fifth *ex-aequo* with min_q = 3.28E-08								
Gentamicin	1	173/1193	1.30E-205	8.73E-01	2/980	9/11	MGE with gene annotation	*aac(6')* gene within a MGE	described for gentamicin resistance	Gordon, 2014
(23/23)	2	127/367	9.02E-75	7.51E-01	7/980	8/11	MGE	sequence of the plasmid carrying *aac(6')*	sequence in LD with the causal gene	
	3	2/23	9.01E-53	6.34E-01	4/980	7/11	MGE	sequence of the plasmid carrying *aac(6')*	sequence in LD with the causal gene	
	4	1/29	1.04E-40	5.79E-01	5/980	7/11	MGE	sequence of the plasmid carrying *aac(6')*	sequence in LD with the causal gene	
	5	2/56	1.49E-33	-8.31E-01	584/980	5/11	polymorphic region in a gene	*odhB*		
Rifampin (103/103)	1	3/31	5.42E-133	-6.67E-01	968/983	2/8	polymorphic region in a gene	*rpoB*	described for rifampin resistance	Gordon, 2014
(103/103)	2	25 subgraphs are second *ex-aequo* with min_q = 1.52E-29. Note that this panel contains only 7 non-susceptible strains								
Mupirocin	1	1/14	1.13E-48	4.59E-01	7/483	6/7	pattern in the promoter region of an accessory gene	*ermA*	described for erythomycin resistance	Gordon, 2014
(64/64)	1	2/42	1.13E-48	-9.95E-01	431/483	5/7	polymorphic region in a gene	*gcvPA*		
	1	2/46	1.13E-48	9.95E-01	52/483	2/7	SNP in a gene	*fabD*		
	4	22 subgraphs are fourth *ex-aequo* with min_q = 1.52E-29. Note that this panel contains only 7 non-susceptible strains.								
Vancomycin	NA	No non-suscpetible strain								

DBGWAS results for *S. aureus* to antibiotics. For each antibiotic, top subgraphs were reported with their rank, the q-value of the unitig with the lowest q-value (minq), the corresponding estimated effect (estimated β of the linear model) and the number of susceptible (resp. resistant) strains harbouring this unitig (count per phenotype). The type of event represented by the subgraph, its annotation and some comments and references on this annotation were also provided. Comments were coloured if the annotation was previously described in antibiotic resistance literature: in green if this description concerned the tested antibiotic, in orange otherwise.

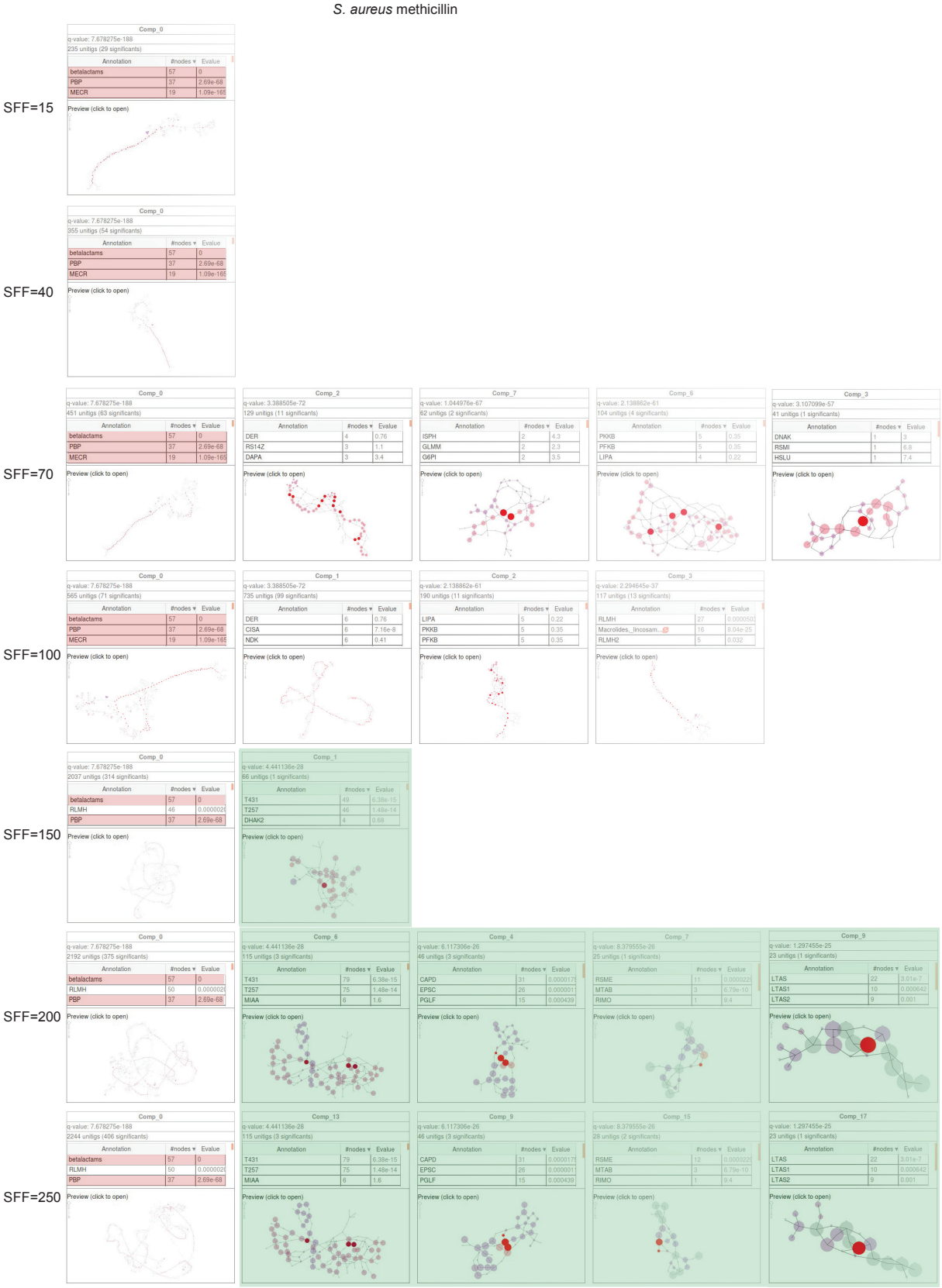
Phenotype (#subgraph with min_q <0.05 / #subgraphs)	DubWaph rank	#siWnBunitiWs over all subWaph unitiWs	UnitiWs ith the minimal q-value				= enetic event	Rnnotation	Gomment : Ween A described for the studied phenotype oranWs A described for another phenotype blue A described for multidruW(not specific) S hite A not described for resistance	. eferences	feature rank (if found by the resistome approach)
			min_q	Estimated effect	count per phenotype: Pheno0 (susceptible)	count per phenotype: Pheno1 (non-susceptible)					
Amikacin (59/59)	1	4/83	5.86E-09	6.21E-01	2/233	15/47	SNP in an accessory gene	Among the 4 significant unitigs, 2 are annotated AAC(6')-Ib3	In current resistance database, the gene presence is described for amynoglycosides. However, this mutation was described in 1999 to increase the affinity of the antibiotic binding site for hydrolysis.	Lambert, 1994	not found
							accesory gene	2 are annotated CML	Phenicol efflux pumps.	George, 2002	170
	2	3/82	1.37E-06	6.62E-01	1/233	13/47	polymorphic region in an accessory gene	DEAD/DEAH box *helicase*			*not found*
	3	38/315	2.21E-06	5.23E-01	8/233	21/47	MGE	plasmid sequences mapping on pHS87b	most unitigs map on prophage regions	Bi, 2016	*not found*
	3	3/100	2.21E-06	6.58E-01	1/233	13/47	pattern in an accessory noncoding region	between an hypothetical protein and a conjugal transfer protein			*not found*
	4	82/1525	3.36E-06	7.12E-01	2/233	10/47	MGE	mercury reductase and transposase			*not found*
Levofloxacin (5/57)	1	5/27	7.21E-29	-8.84E-01	69/70	7/47	hot-spot in a core gene	*gyrA* QRDR	hot-spot region described for quinolones resistance	Hooper, 2015	1
	2	5/29	5.68E-06	-7.37E-01	70/70	25/47	hot-spot in a core gene	*parC* QRDR	hot-spot region described for quinolones resistance	Hooper, 2015	2
	3	5/38	1.87E-02	6.88E-01	1/70	17/47	SNP in a core gene	Histidine Kinase			*not found*
	4	49/781	2.37E-02	7.00E-01	0/70	16/47	MGE	transposase			*not found*
	5	2/33	4.51E-02	6.46E-01	1/70	17/47	MGE with gene annotation	DNA topoisomerase 1 (*topA*)			*not found*
Meropenem (1/71)	1	7/43	3.91E-05	-3.94E-01	144/192	23/88	hot-spot in a core gene	*gyrA* QRDR	hot-spot region described for quinolones resistance	Hooper, 2015	1
Piperacillin (1/62)	1	7/43	2.45E-03	-4.05E-01	100/115	65/165	hot-spot in a core gene	*gyrA* QRDR	hot-spot region described for quinolones resistance	Hooper, 2015	1
Colistin (113/113)	1	121/2058	1.07E-11	9.63E-01	0/156	3/8	MGE	plasmid sequence (annotated either in non-coding region or in putative transmembrane protein)	The overall allele frequencies are very low, showing a lack of power for suitable detection. Note that only one node shows the minq. Other nodes have q-values >= 2.18E-06 (which is the minq of the 19 next subgraphs)		*not found*
	2 --> 21	the 19 following subgraphs shows the same min(q-value)=2.18E-06, among them some unitigs correspond to the MEX family (see below)									
			2.18E-06	6.67E-01	0/156	2/8	MGE with gene annotation	MEX protein family	gene presence described for multidrug efflux pump	Mesaros, 2007	3
Polymyxin B (53/53)	1 --> 13	There are only 3 non-susceptible strains in the panel. The 13 first subgraphs show a min(q-value)=4.43e-47, among them some unitigs correspond to the MEX family (see below)									
		161/1281	4.43E-47	1.00E+00	0/114	2/3	MGE with gene annotation	MEX protein family	gene presence described for multidrug efflux pump	Mesaros, 2007	4
Chloramphenicol (125/125)	1 --> 34	There are only 3 susceptible strains in the panel. The 34 first subgraphs show a min(q-value)=4.47e-09, among them some unitigs correspond to the MEX family (see below)									
		2/35	4.47E-09	6.57E-01	1/3	99/100	MGE with gene annotation	MEX protein family	gene presence described for multidrug efflux pump	Mesaros, 2007	1
Cefepime (0/59)	no association found										one association found: SNP on gyrA
Fosfomycin (0/61)	no association found										no association found

Cg = w RD results for *P. aeruginosa* resistance to antibioticsB For each antibiotic, top subgraphs were reported with their rank, the q-value of the unitig with the lowest q-value (minq), the corresponding estimated effect (estimated β of the linear model) and the number of susceptible (resp. resistant) strains harbouring this unitig (count per phenotype). The type of event represented by the subgraph, its annotation and some comments and references on this annotation were also provided. Comments were coloured if the annotation was previously described in antibiotic resistance literature: in green if this description concerned the tested antibiotic, in orange otherwise. Blue represents multidrug determinants.



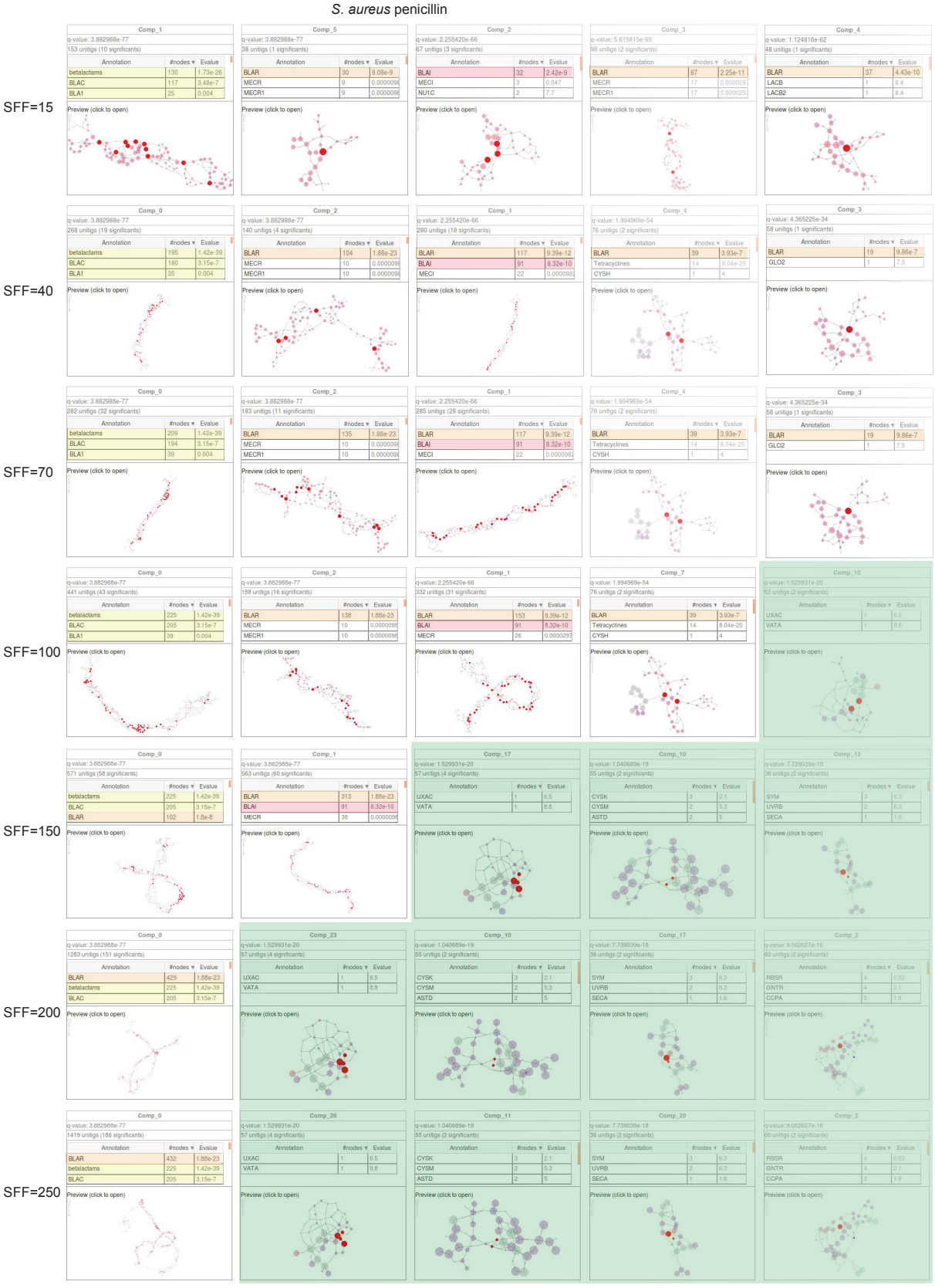
The raw DBGWAS results with the different values for SFF, which are summarised in this figure, are available at http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/experiments/index.html#DBGWAS_all_results_different_SFF

Supplementary Material S3.5: **Effect of SFF on the top subgraphs generated for *S. aureus* ciprofloxacin resistance.** Annotation of the first subgraphs is strictly conserved (red for *parC*, green for *gyrA*, yellow for *norA* promoter region, blue for noncoding between *glmM* and *fmtB* and violet for transposase flanking regions).



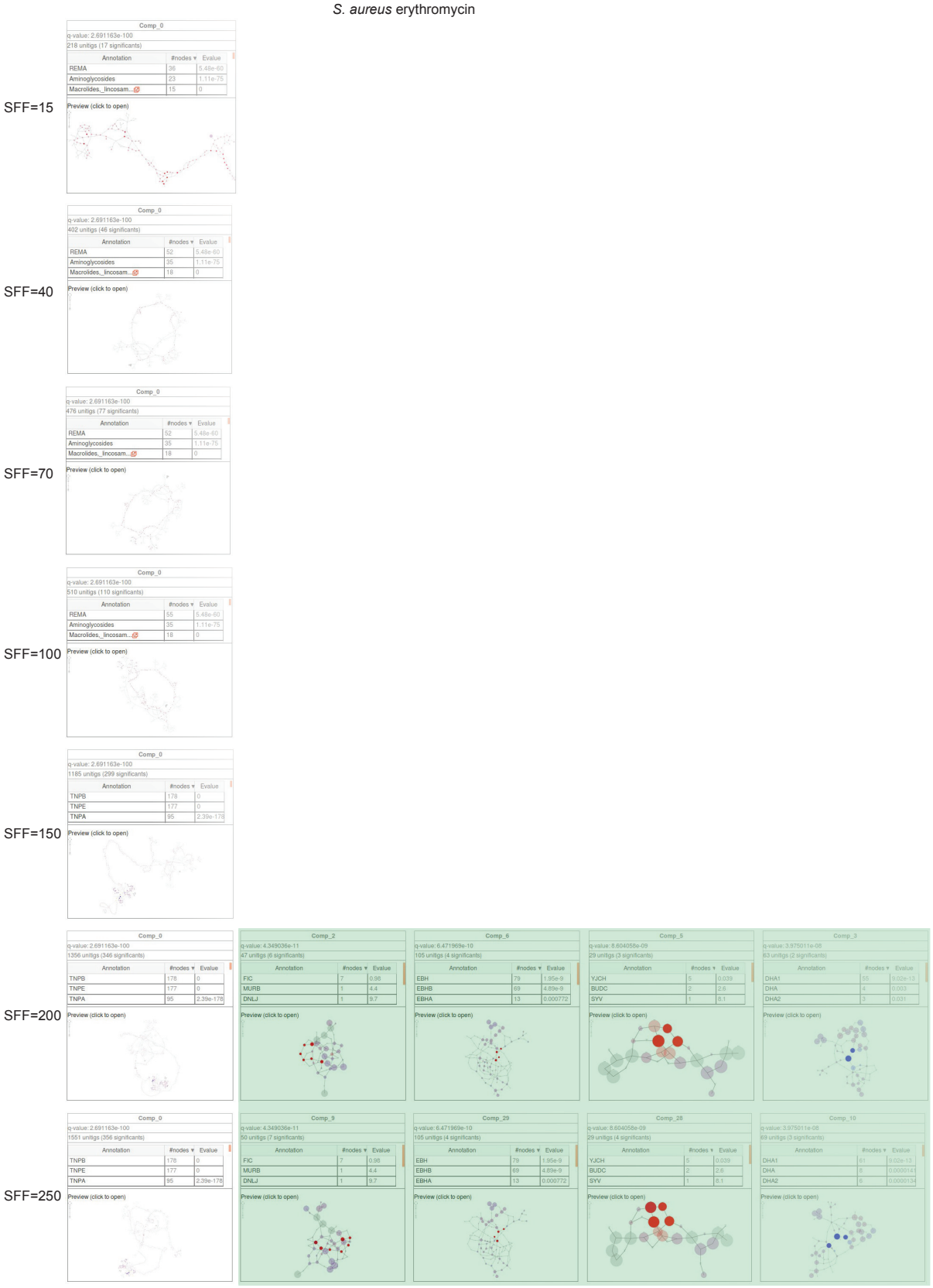
The raw DBGWAS results with the different values for SFF, which are summarised in this figure, are available at http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/experiments/index.html#DBGWAS_all_results_different_SFF

Supplementary Material S3.6: **Effect of SFF on the top subgraphs generated for *S. aureus* methicillin resistance.** Only one subgraph, containing the *mecA* gene (highlighted in red) is generated for lower SFF values. Then several regions of the SCC*mec* cassette appear for $SFF = 70$, and are aggregated into a single subgraph for $SFF \geq 150$. Green subgraphs do not concern the *mecA* MGE.



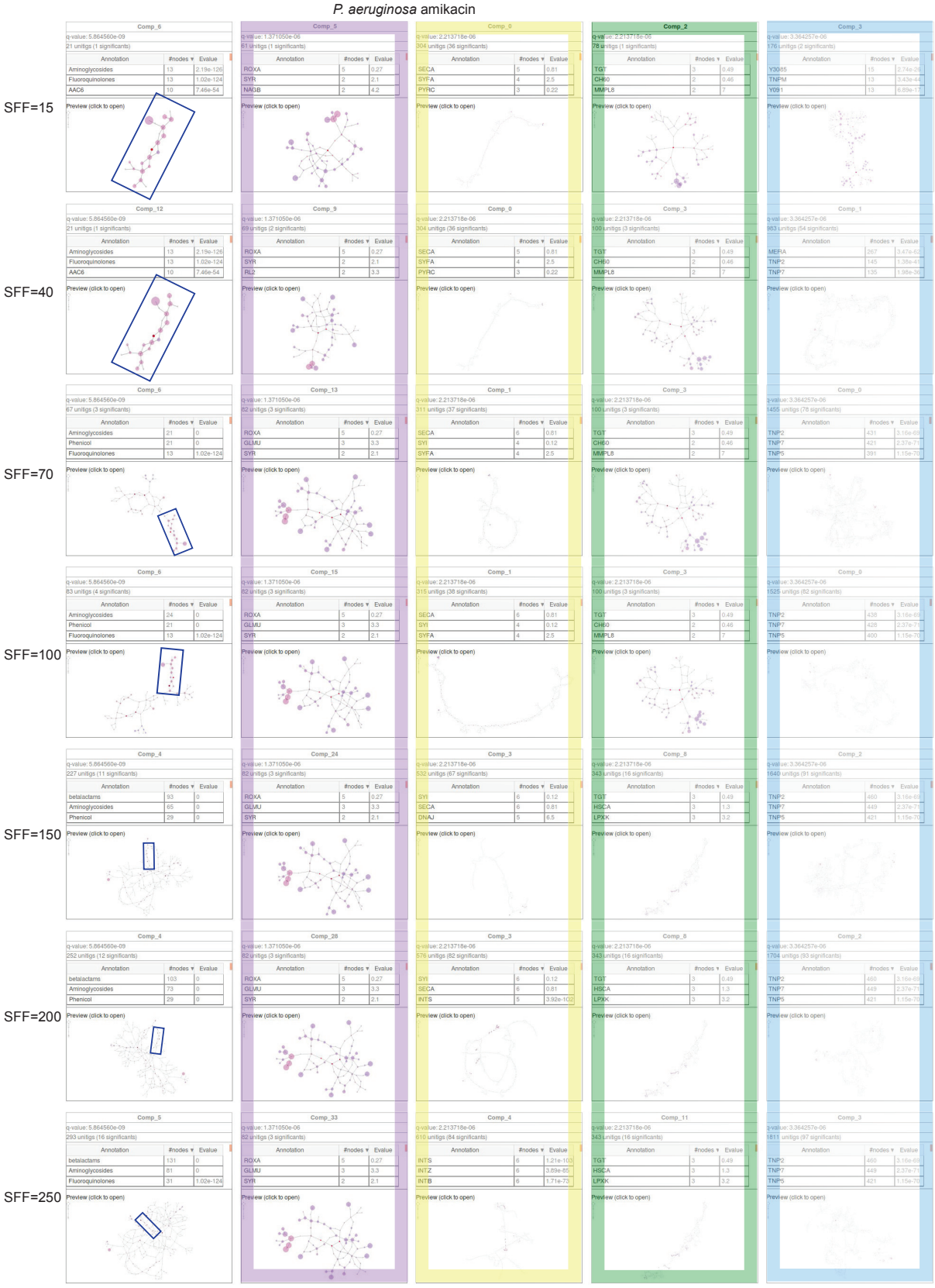
The raw DBGWAS results with the different values for SFF, which are summarised in this figure, are available at http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/experiments/index.html#DBGWAS_all_results_different_SFF

Supplementary Material S3.7: **Effect of SFF on the top subgraphs generated for *S. aureus* penicillin resistance.** Green subgraphs do not concern the *blaZ* MGE. Annotations are ordered by number of nodes carrying it. Yellow, orange and pink highlight *blaZ*, *blaR1* and *blaI*, respectively.



The raw DBGWAS results with the different values for SFF, which are summarised in this figure, are available at http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/experiments/index.html#DBGWAS_all_results_different_SFF

Supplementary Material S3.8: **Effect of SFF on the top subgraphs generated for *S. aureus* erythromycin resistance.** Only one subgraph, describing the *ermC* and its plasmid is outputted when $SFF < 200$. Green subgraphs do not concern the *ermC* MGE.



The raw DBGWAS results with the different values for SFF, which are summarised in this figure, are available at http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/experiments/index.html#DBGWAS_all_results_different_SFF

Supplementary Material S3.9: **Effect of SFF on the top subgraphs generated for *P. aeruginosa* amikacin resistance.** Nodes corresponding to *aac(6')* gene are shown in a blue frame. When the SFF parameter increases, these nodes aggregate to others genes found at least once close to *aac(6')*. The annotation of the following subgraphs are well conserved (same color legend as in S3.16 Fig).

Phenotype #feat with q-val <0.05	Feature rank	Q-value	Estimated effect	Genetic event	Feature name	green = described for the studied phenotype orange = described for another phenotype	Subgraph rank (if found by DBGWAS) and min_q if not in Table S3
Rifampicin	1	1.43E-45	5.04E-01	SNP in a core gene	*rpoB*	hot-spot region described for rifampicin resistance.	1
	2	7.10E-22	-3.55E-01	SNP in a core gene	*katG*	described for isoniazid resistance	2
	3	2.09E-18	-3.36E-01	SNP in a core gene	*katG*	described for isoniazid resistance	2
10	4	2.59E-11	3.31E-01	indel in a core gene	*rpoB*	hot-spot region described for rifampicin resistance.	1
	5	6.28E-08	3.14E-01	indel in a core gene	*rpoB*	hot-spot region described for rifampicin resistance.	1
	6	4.89E-06	2.01E-01	SNP in a core gene	*embB*	described for ethambutol resistance	3
	7	8.26E-06	1.86E-01	SNP in a core gene	*embB*	described for ethambutol resistance	3
	8	3.20E-04	2.39E-01	SNP in a core gene	*rpoB*	hot-spot region described for rifampicin resistance.	1
	9	1.08E-03	-6.57E-01	SNP in a core gene	*gyrA*	described for quinolones resistance	5
	10	2.24E-02	1.40E-01	SNP in a core gene	*rpsL*	described for streptomycin resistance	19 (5.30E-01)
Isoniazid	1	8.41E-274	-7.68E-01	SNP in a core gene	*katG*	described for isoniazid resistance	1
	2	2.14E-248	-7.43E-01	SNP in a core gene	*katG*	described for isoniazid resistance	1
	3	8.96E-61	4.57E-01	SNP in a core gene	*rpoB*	described for rifampicin resistance	2
240	4	3.13E-39	4.32E-01	SNP in a core gene	*embB*	described for ethambutol and rifampicin resistance	3
	5	1.11E-37	4.31E-01	SNP in a core gene	*embB*	described for ethambutol and rifampicin resistance	3
	5	1.11E-37	-4.99E-01	SNP in a core gene	*gyrA*	described for quinolones resistance	*not found*
	7	2.98E-37	-4.91E-01	SNP in a core gene	*embC*	described for ethambutol resistance	*not found*
	8	3.05E-37	-4.94E-01	SNP in a core gene	*kasA*	described for thiolactomycin resistance	*not found*
	9	1.08E-35	-4.78E-01	SNP in a core gene	*embB*	described for ethambutol resistance	*not found*
	10	4.54E-30	4.01E-01	indel in a core gene	*rpoB*	described for rifampicin resistance	2
Ethambutol	1	1.55E-19	3.97E-01	SNP in a core gene	*embB*	described for ethambutol resistance	1
	2	4.01E-19	4.06E-01	SNP in a core gene	*embB*	described for ethambutol resistance	1
	3	8.51E-15	2.96E-01	SNP in a core gene	*rpoB*	described for rifampicin resistance	2
67	4	8.08E-14	-2.78E-01	SNP in a core gene	*katG*	described for isoniazid resistance	4
	5	3.23E-11	-2.59E-01	SNP in a core gene	*katG*	described for isoniazid resistance	4
	6	2.52E-07	2.70E-01	indel in a core gene	*rpoB*	described for rifampicin resistance	2
	7	6.00E-07	3.37E-01	SNP in a core gene	*gyrA*	described for quinolones resistance	3
	8	7.93E-07	2.58E-01	SNP in a core gene	*rrs*	described for kanamycin resistance	6
	9	4.16E-06	2.33E-01	SNP in a core gene	*rpsL*	described for streptomycin resistance	8
	10	1.92E-05	5.36E-01	SNP in a core gene promoter	*eis_promoter*	described for low level resistance to kanamycin	11 (3.32E-03)
Streptomycin	1	1.96E-33	5.44E-01	SNP in a core gene	*rpsL*	described for streptomycin resistance	1
	2	2.61E-30	-4.24E-01	SNP in a core gene	*katG*	described for isoniazid resistance	2
	3	1.63E-26	-4.08E-01	SNP in a core gene	*katG*	described for isoniazid resistance	2
28	4	2.74E-13	5.68E-01	SNP in a core gene	*rpsL*	described for streptomycin resistance	1
	5	3.24E-13	5.75E-01	SNP in a core gene	*rpsL*	described for streptomycin resistance	1
	6	2.59E-08	2.35E-01	SNP in a core gene	*rpoB*	described for rifampicin resistance	3
	7	5.40E-08	5.64E-01	SNP in a core gene	*rrs*	described for streptomycin resistance	5
	8	1.75E-07	5.70E-01	SNP in a core gene	*rrs*	described for streptomycin resistance	5
	9	1.93E-06	2.56E-01	indel in a core gene	*rpoB*	described for rifampicin resistance	3
	10	1.41E-04	4.92E-01	SNP in a core gene	*rrs*	described for streptomycin resistance	5
Kanamycin	1	2.44E-149	8.55E-01	SNP in a core gene	*rrs*	described for kanamycin resistance	1
	2	2.43E-45	7.82E-01	indel in a core gene	*rpoB*	described for rifampicin resistance	4
	3	1.08E-44	8.09E-01	SNP in a core gene	*rpoB*	described for rifampicin resistance	2
105	4	1.83E-43	6.16E-01	SNP in a core gene	*gyrA*	described for quinolones resistance	9
	5	3.12E-31	6.14E-01	SNP in a core gene	*pncA*	described for pyrazinamide resistance	*not found*
	6	1.01E-29	5.79E-01	SNP in a core gene	*rpoB*	described for rifampicin resistance	4
	7	1.03E-28	4.63E-01	indel in a core gene	*rpoB*	described for rifampicin resistance	4
	8	1.98E-25	4.92E-01	indel in a core gene	*pncA*	described for pyrazinamide resistance	*not found*
	9	1.12E-20	4.64E-01	SNP in a core gene promoter	*fabG1_promoter*	described for ethionamide and isoniazid resistance	*not found*
	10	2.75E-18	3.76E-01	SNP in a core gene	*gidB*	described for streptomycin resistance	*not found*
Ofloxacin	1	2.83E-59	7.48E-01	SNP in a core gene	*gyrA*	described for quinolones resistance	1
	2	2.19E-32	6.87E-01	SNP in a core gene	*gyrA*	described for quinolones resistance	1
	3	4.95E-28	7.10E-01	SNP in a core gene	*rpoB*	described for rifampicin resistance	*not found*
99	4	8.36E-27	6.65E-01	indel in a core gene	*rpoB*	described for rifampicin resistance	*not found*
	5	5.57E-22	4.11E-01	SNP in a core gene	*rrs*	described for kanamycin resistance	*not found*
	6	1.70E-19	4.82E-01	SNP in a core gene promoter	*fabG1_promoter*	described for ethionamide and isoniazid resistance	*not found*
	7	1.36E-18	5.20E-01	SNP in a core gene	*pncA*	described for pyrazinamide resistance	*not found*
	8	7.31E-18	6.74E-01	SNP in a core gene	*gyrA*	described for quinolones resistance	1
	9	3.45E-17	4.90E-01	SNP in a core gene	*rpoB*	described for rifampicin resistance	*not found*
	10	5.49E-17	4.02E-01	SNP in a core gene	*gidB*	described for streptomycin resistance	*not found*
Ethionamide	1	1.15E-19	4.73E-01	SNP in a core gene	*rrs*	described for kanamycin resistance	3
	2	1.19E-18	5.54E-01	SNP in a core gene	*gyrA*	described for quinolones resistance	2
	3	1.91E-18	5.35E-01	SNP in a core gene	*gidB*	described for streptomycin resistance	19 (6.32E-02)
105	4	1.10E-16	6.08E-01	SNP in a core gene promoter	*fabG1_promoter*	described for ethionamide and isoniazid resistance	1
	5	2.04E-16	6.04E-01	SNP in a core gene	*rpoB*	described for rifampicin resistance	4
	6	3.78E-16	6.15E-01	SNP in a core gene	*pncA*	described for pyrazinamide resistance	15 (4.84E-02)
	7	1.50E-15	4.43E-01	SNP in a core gene	*embB*	described for ethambutol resistance	5
	8	1.74E-15	6.12E-01	indel in a core gene	*rpoB*	described for rifampicin resistance	4
	9	2.15E-15	5.37E-01	SNP in a core gene	*gidB*	described for streptomycin resistance	19 (6.32E-02)
	9	2.15E-15	5.37E-01	SNP in a core gene	*gidB*	described for streptomycin resistance	19 (6.32E-02)

Resistome-based GWAS results for *M. tuberculosis* resistance to antibiotics.

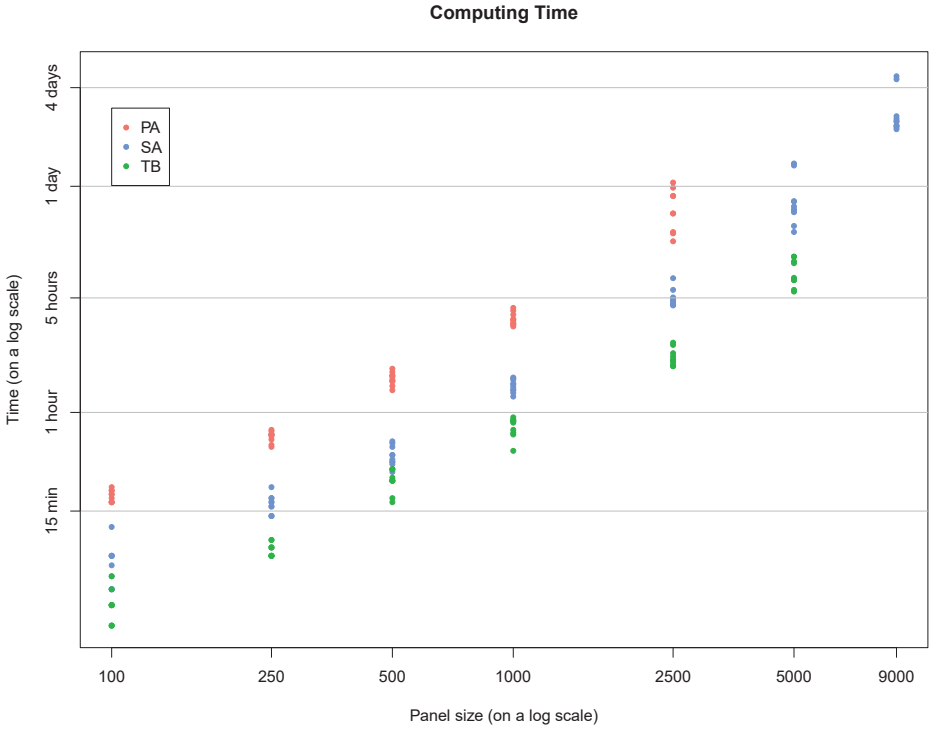
For each antibiotic, the 10 first features most associated to the phenotype were reported, with their rank, q-value, and estimated effect (estimated β of the linear model). The type of targeted variant, with its gene annotation were also provided. Comments were coloured if the annotation was previously described in antibiotic resistance literature: in green if this description concerned the tested antibiotic, in orange otherwise. The last column presents the corresponding subgraphs found by DBGWAs, with their rank and min_q.

Phenotype #feat with q-val <0.05	Feature rank *	Q-value	Estimated effect	Genetic event	Feature name	green = described for the studied phenotype orange = described for another phenotype blue = described for multidrug (not specific)	min_q if found by DBGWAS
Amikacin	1	9.78E-14	4.93E-01	SNP in an accessory gene	OXA	gene presence described for beta-lactams resistance	
	30	1.02E-11	4.29E-01	accessory gene	OXA	described for beta-lactams resistance	2.47E-01
	67	2.06E-08	5.70E-01	SNP in an accessory gene	AAC(6')-Ib3	gene presence described for amynoglycosides resistance (in LD with gene presence)	not found
850	68	2.06E-08	5.70E-01	accessory gene	AAC(6')-Ib3	described for amynoglycosides resistance	6.05E-03
	170	3.81E-07	5.87E-01	SNP in an accessory gene	*cmlA1*	gene presence described for chloramphenicols efflux pumps	1.20E-03
	214	2.01E-06	5.39E-01	accessory gene	*cmlA1*	described for chloramphenicols efflux pumps	3.74E-04
	495	2.61E-04	8.91E-01	accessory gene	TEM	described for beta-lactams resistance	
	508	1.10E-03	-6.19E-01	SNP in an accessory gene	*mexR*	gene presence described for multidrug efflux pump	
	659	1.78E-03	5.56E-01	indel in an accessory gene	*opmE*	gene presence described for multidrug efflux pump	
	664	3.06E-03	-2.06E-01	SNP in an accessory gene	*gyrA*	SNP described for quinolones resistance	
Levofloxacin	1	2.11E-22	-8.54E-01	SNP in a core gene	*gyrA*	SNP described for quinolones resistance	7.21E-29
2	2	1.83E-05	-7.02E-01	SNP in a core gene	*parC*	SNP described for quinolones resistance	5.68E-06
Meropenem	1	5.63E-04	-3.30E-01	SNP in a core gene	*gyrA*	SNP described for quinolones resistance	3.91E-05
	2	2.85E-03	3.81E-01	SNP in an accessory gene	OXA	gene presence described for beta-lactams resistance	*not found*
90	79	4.22E-03	5.97E-01	SNP in an accessory gene	*ef-tu*	gene presence described for elfamycin resistance	
Piperacillin	1	1.33E-04	-3.86E-01	SNP in a core gene	*gyrA*	SNP described for quinolones resistance	2.45E-03
2	2	4.84E-02	-3.32E-01	SNP in a core gene	*parC*	SNP described for quinolones resistance	
Colistin	1	9.25E-04	3.37E-01	SNP in an accessory gene	AAC(6')-Ib'	gene presence described for amynoglycosides resistance	
	2	9.25E-04	3.37E-01	accessory gene	AAC(6')-Ib'	described for amynoglycosides resistance	
	3	1.85E-03	-9.53E-01	SNP in an accessory gene	*mexJ*	gene presence described for multidrug efflux pump	2.18E-06
19	3	1.85E-03	-9.53E-01	SNP in an accessory gene	*mexV*	gene presence described for multidrug efflux pump	2.18E-06
	3	1.85E-03	-9.53E-01	SNP in an accessory gene	*mexW*	gene presence described for multidrug efflux pump	2.18E-06
	3	1.85E-03	-9.53E-01	SNP in an accessory gene	*mvat*	gene presence described for multidrug efflux pump	
	3	1.85E-03	-9.53E-01	SNP in an accessory gene	*opmE*	gene presence described for multidrug efflux pump	
	3	1.85E-03	-9.53E-01	SNP in an accessory gene	*parC*	SNP described for quinolones resistance	
	3	1.85E-03	-9.53E-01	SNP in an accessory gene	*parS*	regulates efflux pump and porin	
	3	1.85E-03	-9.53E-01	SNP in an accessory gene	PDC	gene presence described for beta-lactam resistance	
Polymyxin B	1	3.24E-13	-9.77E-01	SNP in an accessory gene	*mvat*	regulates efflux pump	
	2	4.08E-11	-9.82E-01	SNP in an accessory gene	APH(3')-IIb	gene presence described for amynoglycosides resistance	
	4	1.19E-10	6.58E-01	SNP in an accessory gene	*mexW*	gene presence described for multidrug efflux pump	4.44E-47
18	5	9.32E-07	-9.83E-01	SNP in an accessory gene	*opmE*	gene presence described for multidrug efflux pump	
	5	9.32E-07	-9.83E-01	SNP in an accessory gene	*parC*	SNP described for quinolones resistance	
	5	9.32E-07	-9.83E-01	SNP in an accessory gene	*parS*	regulates efflux pump and porin	
	5	9.32E-07	-9.83E-01	SNP in an accessory gene	*PDC*	gene presence described for beta-lactam resistance	
Chloramphenicol	1	6.09E-11	6.57E-01	SNP in an accessory gene	*mexE*	gene presence described for multidrug efflux pump	4.47E-09
	1	6.09E-11	6.57E-01	SNP in an accessory gene	*mexF*	gene presence described for multidrug efflux pump	4.47E-09
292	2	9.44E-08	9.80E-01	SNP in an accessory gene	*opmE*	gene presence described for multidrug efflux pump	
	2	9.44E-08	9.80E-01	indel in an accessory gene	*oprD*	porin described for beta-lactam resistance	
	2	9.44E-08	9.80E-01	SNP in an accessory gene	*parC*	SNP described for quinolones resistance	
	2	9.44E-08	9.80E-01	SNP in an accessory gene	*parS*	regulates efflux pump and porin	
	2	9.44E-08	9.80E-01	SNP in an accessory gene	*PDC*	gene presence described for beta-lactam resistance	
Cefepime 1	1	2.10E-02	-3.01E-01	SNP in a core gene	*gyrA*	SNP described for quinolones resistance	
Fosfomycin	no association found						

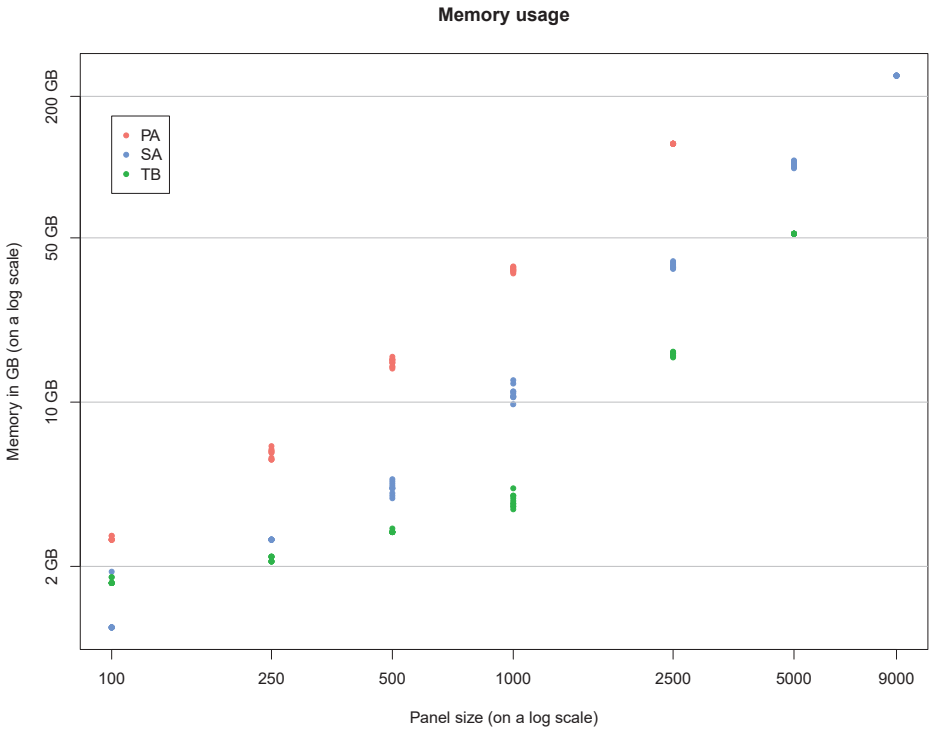
* when several variants are described for a gene, only the first one (with the min q-value) is given in this table

Resistome-based GWAS results for *P. aeruginosa* resistance to antibiotics.

For each antibiotic, the 10 first features most associated to the phenotype were reported, with their rank, q-value, and estimated effect (estimated β of the linear model). The type of targeted variant, with its gene annotation were also provided. Comments were coloured if the annotation was previously described in antibiotic resistance literature: in green if this description concerned the tested antibiotic, in orange otherwise. The last column presents the corresponding subgraphs found by DBGWAs, with their min_q.



(A)



(B)

Supplementary Material S3.12: **Large scale analysis on computational resources usage.** This figure describes how DBGWAS scales in terms of time and memory usage for large datasets, containing up to 9,000 genomes. The large panels used here are described in the Large panels subsection of the Methods section. To understand better DBGWAS performance behaviour, we present performance curves for each panel at size points of 100, 250, 500, 1,000, 2,500, 5,000 and 9,000 genomes. The executions were done in a cluster, instead of a single machine, and used 8 cores each. In order to reduce subsampling and machine heterogeneity problems, each sub-panel was randomly built 10 times and we present the time and memory usage for all these executions. Although these two measures not only depends on the number of input genomes but also on their length and complexity, this figure allows estimations of the computational resources usage on small and large panels with different genome plasticities.

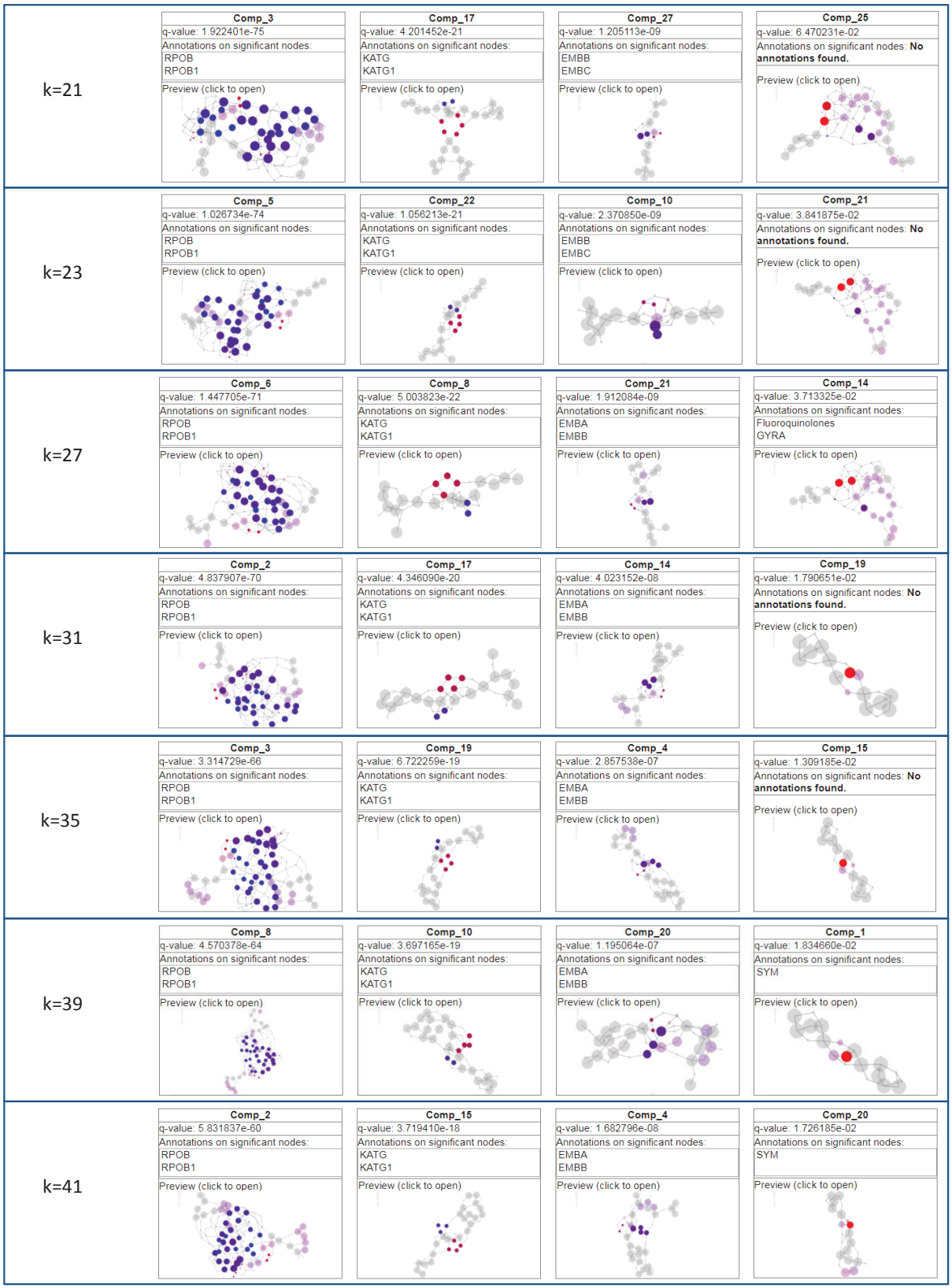
Panel	Phenotype	DBGWAS	fsm-lite	pyseer				HAWK					HAWK		
				Mash v2.0		pyseer							ABYSS	Stats	Total
				DQE		DQE		**Total**	Count k-mers		HAWK				
TB	ethambutol	**37m (3.8)**	6h48m (93.6)	4m (0.2)		6h10m (3.6)		**13h02m (93.6)**	2h11m (1.5)		26m (3.7)		1m (0.3)	5m (0.1)	**2h43m (3.7)**
	streptomycin	**42m (4.3)**	7h47m (102.4)	5m (0.2)		6h22m (4.1)		**14h14m (102.4)**	2h26m (1.5)		25m (3.7)		1m (0.3)	9m (0.1)	**3h01m (3.7)**
	rifampicin	**43m (4.4)**	8h01m (104.7)	5m (0.2)		7h45m (4.1)		**15h51m (104.7)**	2h30m (1.5)		25m (3.7)		1m (0.3)	14m (0.1)	**3h10m (3.7)**
	ethionamide	**12m (3.5)**	2h18m (34.8)	1m (0.1)		1h35m (1.3)		**3h54m (34.8)**	53m (1.5)		17m (3.5)		1m (0.3)	1m (0.1)	**1h12m (3.5)**
Large TB	random	**1h44m (15.7)**	23h31m (215.8)	14m (0.9)		6h13m (11.3)		**29h58m (215.8)**	5h15m (1.5)		48m (4.2)		1m (0.3)	1m (0.1)	**6h05m (4.2)**
SA	ciprofloxacin	**1h16m (11.2)**	6h31m (53.8)	3m (0.2)		Timeout		**Timeout**	1h31m (1.5)		27m (4.0)		1m (0.4)	22h11m (0.1)	**24h10m (4.0)**
	erythromycin	**1h17m (11.2)**	6h32m (53.8)	3m (0.2)		Timeout		**Timeout**	1h31m (1.5)		26m (4.1)		1m (0.3)	7h56m (0.1)	**9h54m (4.1)**
	methicillin	**29m (4.3)**	3h15m (27.1)	1m (0.1)		Timeout		**Timeout**	46m (1.5)		19m (3.8)		1m (0.3)	8h29m (0.1)	**9h35m (3.8)**
Large SA	random	**5h11m (37.4)**	18h35m (149.5)	12m (0.9)		26h53m (6.8)		**45h40m (149.5)**	3h51m (1.5)		41m (5.8)		1m (0.3)	1m (0.1)	**4h34m (5.8)**
PA	meropenem	**49m (8.0)**	6h32m (34.7)	2m (0.1)		33h59m (0.5)		**40h33m (34.7)**	50m (1.5)		21m (4.5)		1m (0.3)	1m (0.1)	**1h13m (4.5)**
	levofloxacin	**21m (3.2)**	3h03m (14.5)	1m (0.1)		21h18m (0.3)		**24h22m (14.5)**	21m (1.5)		16m (4.2)		1m (0.3)	1m (0.1)	**39m (4.2)**
	amikacin	**51m (8.0)**	5h59m (34.7)	2m (0.1)		46h44m (0.5)		**52h45m (34.7)**	50m (1.5)		22m (4.5)		1m (0.3)	1m (0.1)	**1h14m (4.5)**
Large PA	random	**13h21m (125.8)**	DQE	DQE		DQE		**DQE**	7h24m (1.5)		2h05m (8.6)		1m (0.3)	1m (0.1)	**9h31m (8.6)**

Supplementary Material S3.13: **Benchmarking DBGWAS, pyseer and HAWK: comparison of time and maximal memory load.** The total execution time is presented with the maximal memory consumption in parenthesis, in order of GBs. For pyseer and HAWK, the time and memory for each step is also detailed. All tools were ran on a same machine with 8 *Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz* cores, 315 GB of RAM and 1 TB of disk space. Each execution used all the 8 available cores. The datasets are described in the Datasets subsection of the Methods section. However, for the three large panels (Large TB, Large SA, and Large PA), here we just chose a random 2,500-genome sub-panel. Moreover, DBGWAS was ran with the default parameters, without optional steps (lineage effect analysis nor annotation of subgraphs). The parameters for pyseer and HAWK were the ones described in the k-mer-based GWAS subsection of the Methods section. We did not consider the time and memory consumed in the last step for these two tools (downstream analysis). The runs taking more than 5 days to finish were interrupted and are shown as *Timeout*. The runs that exceeded 1 TB of disk space were interrupted and are shown as *DQE* (Disk Quota Exceeded).

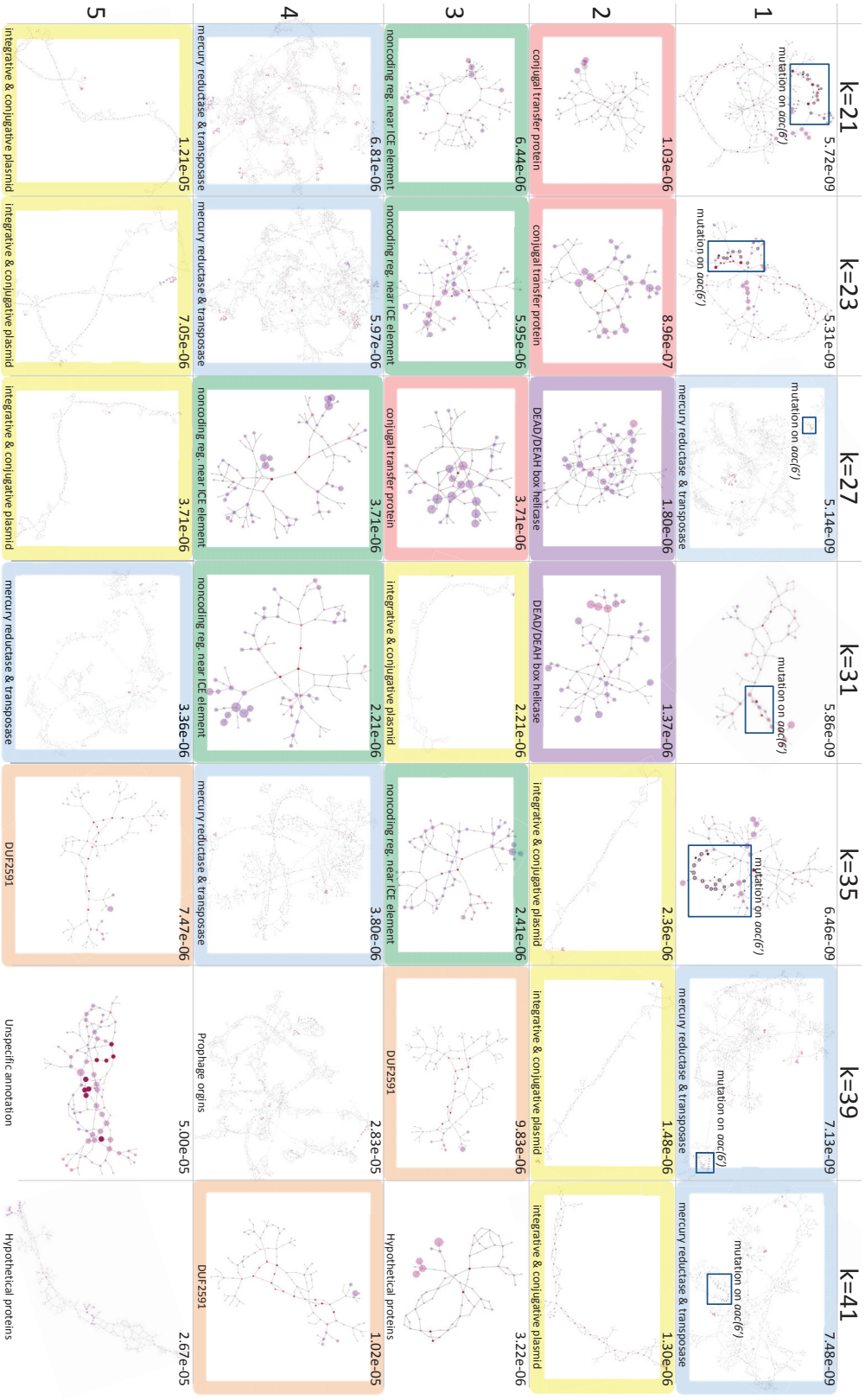
Panel (gen len Mbp)	Phenotype	Panel size	k-mers in M	Unitigs in M	Patterns[1]in M	Runtime on 1 core (mem GB)
TB (4.4)	ethambutol	1041	8.22	0.36	0.05	1h28m (3.5)
	streptomycin	1166	8.32	0.37	0.05	1h40m (4)
	rifampicin	1197	8.29	0.37	0.05	1h42m (4.1)
	ethionamide	420	5.74	0.13	0.02	32m (3.4)
SA (2.7-3.1)	ciprofloxacin	991	23.13	1.12	0.48	2h15m (11)
	erythromycin	991	23.13	1.12	0.48	2h10m (11)
	methicillin	501	17.85	0.81	0.30	51m (4.1)
PA (5.8-7.6)	meropenem	280	54.46	2.36	1.01	1h37m (7.6)
	levofloxacin	117	41.40	1.70	0.64	42m (2.9)
	amikacin	280	54.46	2.36	1.01	1h37m (7.6)

[1]Patterns are the columns of the X matrix, containing the unique unitig minor allele description profiles (see *Methods* section). Association tests are computed on the patterns.

Supplementary Material S3.14: **DBGWAS time and maximal memory load on a single core.** All runs presented in this table were executed with the default parameters, without optional steps (lineage effect analysis nor annotation of subgraphs), on a single *Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz* core. The datasets are described in the Datasets subsection of the Methods section. DBGWAS ran in less than 2,5 hours for all experiments in our benchmark. The maximum memory load (given between parenthesis in the Runtime column) was 11 GB of RAM. The panel size and genome length (given between parenthesis in the Panel column) did not drive alone the running performances; the genome complexity played an important role as well. To view the gain in performance of DBGWAS when running on multiple (8) cores, see S3.13 Suppl.



Supplementary Material S3.15: **Effect of k on the four first subgraphs obtained for TB rifampicin resistance.** With a k value varying between 21 and 41, the first 3 subgraphs always have the same ordering, shape and annotation, as well as comparable q-values, although smaller q-values are observed for lower values of k. The number of significant unitigs per subgraph is also well conserved. The fourth top-rated subgraphs are not always the same: the *gyrA* mutation appears at a lower rank when k is smaller.



Supplementary Material S3.16: **Effect of k on the five first subgraphs obtained for *P. aeruginosa* amikacin resistance.** When k varies, the plasmid (yellow) and the mercury reductase and transposase (blue) remain among the five top-rated subgraphs. However, k has an effect on the aggregation of subgraphs corresponding to different genetic events: the mutation on *aac(6')* gene (blue frame) always appears in the first subgraph but is merged with the large mercury reductase and transposase subgraph for $k = 27, 39$ and 41. The order of the subgraphs also varies with k: up to four ranks for some subgraphs, and others leave the top-5 list.

Panel	Phenotype	number of subgraphs	
		top 100	5% FDR
TB	MDR	28	3
	XDR	26	23
	ethambutol	23	23
	streptomycin	17	24
	rifampicin	25	6
SA	ciprofloxacin	163	1877
	erythromycin	1	444
	methicillin	4	352
PA	meropenem	70	1
	levofloxacin	56	5
	amikacin	59	397

Supplementary Material S3.17: **Number of subgraphs generated using different significance thresholds.** This table shows the number of subgraphs generated when defining the significant unitigs as the ones with the 100 lowest q-values (default $SFF = 100$, 'top 100') or when using a 5% false discovery rate (FDR) threshold ($SFF = 0.05$, '5% FDR'). Different datasets lead to different q-values, even by several orders of magnitude. For instance, a single FDR threshold leads to selecting a large number of unitigs generating several hundreds subgraphs for SA (*S. aureus*) panel.

Appendix S3.1: Evaluation of association models.

We evaluate two models controlling for population structure: (i) using the first principal components as covariates with a fixed effect [168] (implemented in the PLINK software [42]) and (ii) linear mixed models [61, 100, 218], and compare them to (iii) a model not accounting for the population structure.
We evaluate the three models by their ability to detect (a) true positive unitigs simulated under different population structures, and (b) unitigs from real data mapping genuine variants described in the literature.

Evaluated models

Let $\mathbf{Z}$ be the full matrix of unitig minor allele frequency patterns (before de-duplication) and $\mathbf{X}$ be the matrix of unique patterns (after de-duplication) as defined in the *Methods* section of the main manuscript. For each pattern X_{ij}, we test $H_0 : \beta = 0$ versus $H_1 : \beta \neq 0$ in the following linear model, relating binarized antibiotic susceptibility phenotypes to X_{ij} candidate genetic determinant and population structure:

$$Y_i = X_{ij}\beta + W_i^T\alpha + \varepsilon_{ij}, \quad j = 1, \ldots, p \tag{4.7}$$

with $\varepsilon_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, $\sigma^2 > 0$, β the effect of the tested candidate on the phenotype, $W \in \mathbb{R}^l$ a factor representing the population structure, and $\alpha \in \mathbb{R}^l$ the effect of this population structure on the phenotype.
Denoting $\mathbf{Z} = U\Lambda V^\top$ the singular value decomposition (SVD) of $\mathbf{Z}$, we use:

(i) $W = U_q$ (matrix formed by the first q columns of U) and a fixed effect α;

(ii) $W = U\Lambda$ and a random effect $\alpha \sim \mathcal{N}(0, \sigma_a^2)$, $\sigma_a^2 > 0$;

(iii) $\alpha = 0$.

For the first two models, we compute p-values for H_0 using a likelihood ratio test. For the random effect model, we use bugwas [61] to test H_0, providing the pre-computed population structure W as described above.

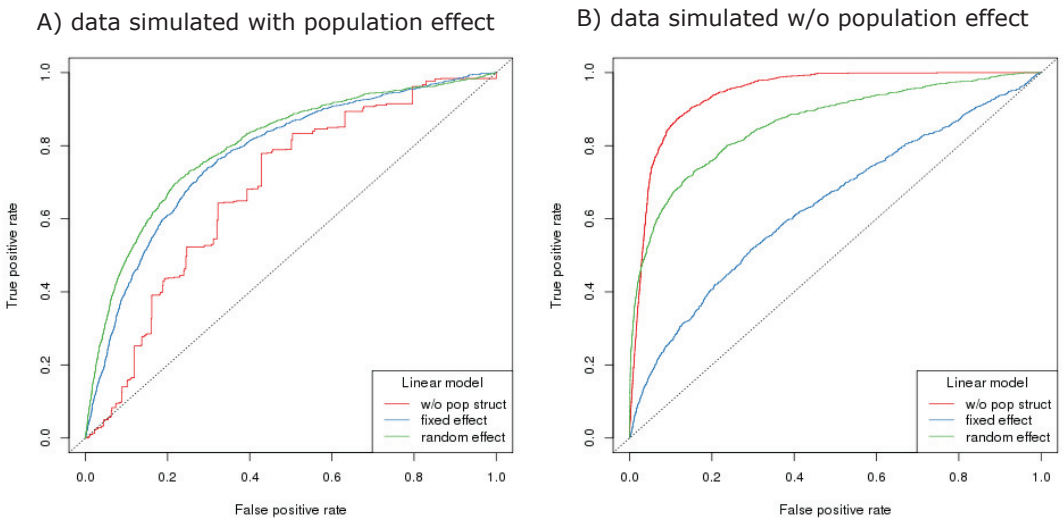
Simulated data

We simulate resistance phenotypes based on the 282 *P. aeruginosa* genomes, arbitrarily fixing which patterns X have a non-zero effect β on the phenotype Y: we sample the phenotype Y_i of each sample i from a multivariate logistic model:

$$Y_i \sim \mathcal{B}(\pi_i), \quad \pi_i = \frac{1}{1 + e^{-\mathbf{X_i}\beta - W_i\alpha}}. \tag{4.8}$$

We generate synthetic data under model Eq 4.8 with two scenarios. The first scenario illustrates a case where there is a population effect on the observed resistance, which is not explained by the set of patterns in the tested design X. The second scenario illustrates the case where there is little population effect observed on the phenotype, except for that caused by the association of modeled causal patterns X with W, *i.e.*, outside of $\mathbf{X}\beta$ in Eq 4.8.
To simulate the first scenario, we arbitrarily assign the 2nd and 6th columns of $W = U\Lambda^{\frac{1}{2}}$ to have non-zero effects α. We then select 10 distinct patterns from $\mathbf{X}$ as true determinants. To do so, we compute the largest dot product of each pattern with the

first six columns of W, and choose our true determinants among those whose largest dot product is below the fifth percentile of dot products calculated across all patterns. This allows us to simulate the case where true determinants are independent from the population structure (their effect is not inflated by the $W\alpha$ term). The odd ratios e^{β_j} are fixed to 6 for these patterns.



Supplementary Material S3.18: **Evaluation of test models on simulated data.** Scenarios (A) and (B) intend to illustrate the model ability to detect true positives in the presence or in the absence of a population effect on the observed resistance. In the first case, the ROC curve shows that taking the population structure into account increases the power, while in the second case, correcting for the population structure when there is not, decreases the power to detect true determinants. Using a random effect model is however more robust and leads to a smaller power loss than using a fixed-effect model.

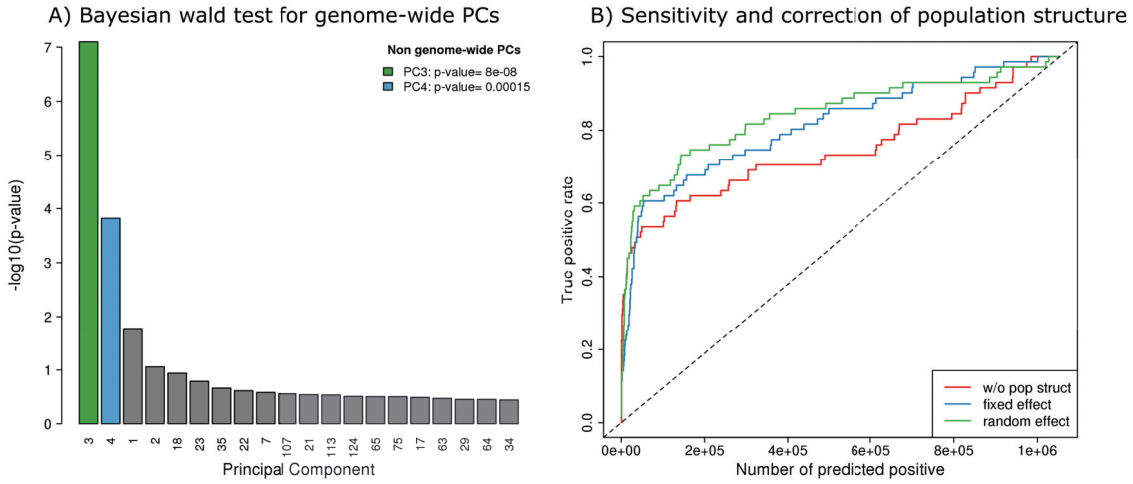
We also randomly select 290 patterns from $\mathbf{X}$ as non-determinants, *i.e.*, with a $\beta_j = 0$ effect in the model, so $p = 300$ in our simulations. The population structure can lead to spurious discoveries, as we do not control the dot product between columns of W and these patterns with zero effect. Finally in order to control the amplitude of the population effect, we normalize $W\alpha$ to 6 times the median value of the $|\mathbf{X^j}\beta_j|$ across non-zero β_j, where $\mathbf{X^j}$ denotes the j-th column of $\mathbf{X}$.

To simulate the second scenario, we use the same settings as before, but we select the 10 true determinants among those that have a large dot product with W, rather than a small one, and set all α effects to zero.

We apply the three versions of our univariate test described in Eq 4.7, with $q = 10$ for model (i), to both scenarios over 100 data generations and plot ROC curves (Fig S3.18 in S1 Appendix). As expected, for the first scenario, the test which does not account for the population structure has very low power to detect patterns associated with the phenotype: by construction, some patterns with zero actual effect have large dot products with $W\alpha$, which inflates the estimate of their effect and leads to false discoveries. Taking the population structure into account in the model improves the power by limiting this inflation. For the second scenario, we observe the opposite effect: correcting for the population structure decreases the power to detect true determinants. Assuming there is a population effect when there is no such effect in reality, leads to artificially deflating the estimated effects of patterns which are associated with the population structure.

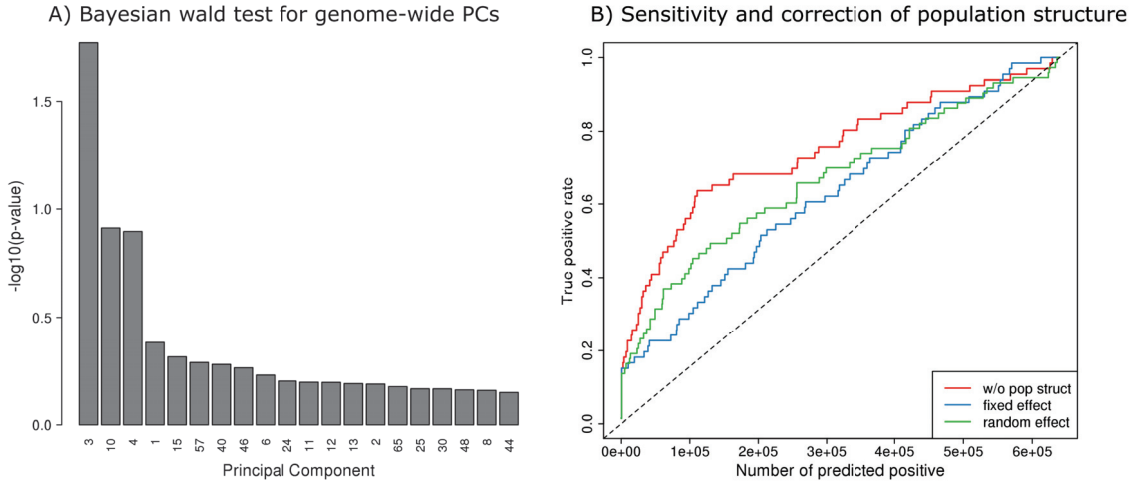
Real data

Using the true phenotype data for both amikacin and levofloxacin resistance in *P. aeruginosa*, we also evaluate a metric based on libraries of known genetic determinants of resistance [98] which we use as our positive set. In this case, we lose the exact knowledge of which unitigs are negative, *i.e.*, have no effect on the phenotype: some selected patterns may not be linked to any know genetic determinant of resistance just because they are still unreported. Instead of ROC curves, we therefore resort to plotting the true positive rate (TPR) – using identified and hence known positives – as a function of the number of positives called by the method – the false positive rate corresponding to this number being unknown. Assigning each selected pattern to a true or false status requires a mapping step: we choose to identify a pattern as a true determinant if it corresponds to at least one unitig which maps to a known genetic determinant from the resistance gene sequence database [98].



Supplementary Material S3.19: **Amikacin resistance and association models.** (A) Two PCs were found significantly associated to the *P. aeruginosa* amikacin phenotype. (B) In this case, the random-effect model performs the best. The model which does not account for the population structure effect performs the worst to retrieve genuine variants of amikacin resistance.

Figs S3.19A and S3.20A in S1 Appendix are produced by bugwas, and show the p-value of the test for association of each column of W with the phenotype. In the case of amikacin, two columns are found to have a significant effect at level 0.01, whereas all columns have p-values larger than 0.01 in the case of levofloxacin. Figs S3.19B and S3.20B in S1 Appendix show that correcting for population structure increases the proportion of known genetic determinants of resistance to amikacin recovered for every number of predicted positives, but decreases this proportion in the case of levofloxacin.
The random effect approach of bugwas is a good choice on both simulated and real data regardless of the effect of the population structure on the phenotype: it outperforms both the uncorrected and the fixed effect approaches in the presence of a population effect, and is only moderately affected by the absence of such effect. We thus implement this model using the bugwas package in DBGWAS.



Supplementary Material S3.20: **Levofloxacin resistance and association models.** (A) No PC was found significantly associated to the P. aeruginosa levofloxacin phenotype. (B) In this case, the model which does not account for the population structure effect performs the best. The linear mixed model still performs better than the fixed effect model to retrieve genuine variants of levofloxacin resistance.

Table S3.5: **Results of the 'ordinal-DBGWAS' on the 'full' *P.aeruginosa* panel.** We ran DBGWAS' first step, and used the matrix of presence/absence patterns to test unitigs associations using the ordinal model of Eq 1.1. We produced the unitig subgraphs with DBGWAS' third step. In this table, for each antibiotic, the genuine variants integrated to the model are provided, as well as a recall of the RWAS new candidates (mutations are shown by a * and gene presence/absence by a +). The last column shows the new candidates found with DBGWAS (each subgraph is represented by [ ]). Whatever the approach, new candidates were found only for few phenotypes, certainly due to the limits of the panel for the other antibiotics. The DBGWAS strategy provided new insights by describing candidates out of the resistome (highlighted in orange).

	Prior	New with RWAS	New with DBGWAS
Amikacin	AAC6+, AAC3*, APH3+, *rmtD*+	OXA+, *mexY*+, *mexX*+, *gyrA**	[biotin metabolism], [*mexX-mexY* operon], [*gyrA*], [hydroxymethylglutaryl-CoA], [acyl-CoA-dehyrogenase], [transglutaminase-like superfamily protein], [RNA polymerase + promoter]
Cefepime	OXA+	OXA*	/
Chloramphenicol	/	/	/
Colistin	/	*mexS*+, *ampD**	[acyl-CoA dehydrogenase, C-terminal domain protein], [cytochrome c-type biogenesis protein CcmI], [hyp. Prot], [*fptB*], [acyl-CoA reductase]
Fosfomycin	/	/	/
Levofloxacin	*gyrA**, *gyrB**, *parC**, AAC6+	/	/
Meropenem	KPC+, VEB+, IMP+, OXA+, *oprD*+	*mexD**, *mexX**, *mexZ**, *oprD**, *ampD**, *gyrA**, *parC**	[*gyrA*], [*parC*], [*oprD*],[*trmU*]
Piperacillin	/	/	/
Polymyxin B	/	/	[noncoding region near MFS transporter], [Ig domain-containing protein], [DNA-binding protein]

Appendix S3.2: Decision of PLOS Genetics referees.

Reviewer 1:

The authors have answered or responded to all my comments from the previous submission. They have clearly put a substantial amount of effort into this revision, both in the manuscript and updating the software. I am convinced that, thanks to the authors' efforts, unitigs will come to replace k-mers as the variant of choice in bacterial GWAS. The final section of the results and tables 2/3 demonstrate this well. Additionally, with the authors' changes I was able to run the software without compilation - I think this will help their software be used more extensively.
I also thank the authors for putting the extra time in to use our updated pyseer method. The only thing I was surprised by was that the disk usage would probably not have been exceeded if the output of fsm-lite was piped through gzip (at least, I have never seen anything over tens of Gb, even with similar numbers of samples). However, I see this may not have been clear in the tutorial the authors followed. Perhaps this could be noted, but it is not necessary for the authors to re-run any of their analysis.
Finally, the full set of output files noted in point 1.10 is available here, if needed:
`https://www.dropbox.com/s/0jg9812y1ywb31g/mass_pen_dbgwas_all.tar.bzip2`
John Lees

Reviewer 2:

I'm extremely satisfied with the changes that have been made to the manuscript, software, and accompanying documentation. The new release of the software is now working on my computing cluster. I'm very happy for the manuscript to be published in its current form.

Appendix S3.3: Response to PLOS Genetics referees.

Response to the reviewers:

We are very grateful to the reviewers for their accurate and valuable comments which helped us to improve the manuscript. We addressed all the reviewers' comments below and we updated the manuscript accordingly.

In particular, as suggested by several reviewers, we worked on the portability of the distributed binary, and used the Holy Build Box (http://phusion.github.io/holy-build-box/) to make the precompiled binaries more portable. It should now work on most glibc-based x86 and x86-64 Linux distribution released since 2007 (e.g.: Debian >= 6, Ubuntu >= 10.04, Red Hat Enterprise Linux >= 5, CentOS >= 5, etc). We also fixed some other software related issues, like correcting the building process, and improving the interaction of DBGWAS with its third-party dependencies.

Furthermore, we drastically modified the benchmark conditions as we upgraded SEER to pyseer, and created dedicated single-species large datasets to assess the computational scalability of the three evaluated tools. For this purpose, we removed the WHO panel and used instead all *P. aeruginosa*, *M. tuberculosis* and *S. aureus* genome assemblies available on the NCBI, and produced simulated phenotypes for them. We also tried to upgrade HAWK from v0.8.3-beta to v0.9.8-beta, since only the last version allows for population structure correction, but unfortunately the results obtained with HAWK v0.9.8-beta were a lot poorer than with the previous version. As such, we decided to keep HAWK v0.8.3-beta in our benchmarks.

Following the reviewers' suggestions, we also modified the text structure of the Results section and integrated parts of the Supplementary materials in the Methods section.

We hope this new version will fulfill the reviewers' expectations.

Sincerely,
M. Jaillard *et al.*

Reviewer 1:

In their manuscript, Jaillard, Lima and colleagues describe a new method for performing genome-wide associations studies in bacteria. Rather than focussing on changes to the underlying association model, they instead propose changing the variants being tested from a k-mer to unitig extracted from the population de Bruijn graph. I think this is a very clever idea, and a natural extension to current bacterial GWAS methods. This is timely, as this community is becoming more comfortable with thinking of and viewing genetic variation through graphs. They test their method on antibiotic resistance in a number of species, now becoming a standard test for bacterial GWAS methods, and successfully find known and potentially new resistance mechanisms.

I thought the paper was very well written, and a lot of work has clearly gone into the analysis (including extensive supplementary material). After reading through the paper, I became very excited to try the software myself, which did not disappoint. I was able to get it running on the provided test dataset with only minor issues. I then tried it on my own set of test data, and got excellent results from the tool.

I fully support the publication of this manuscript in PLOS Genetics. The method itself will be useful to bacterial genomics researchers, and the investigations performed in the paper also contain interesting nuggets of information for those particularly interested in bacterial GWAS. I have some comments listed below that I would like to see the authors consider before publication, though these are mostly minor changes or suggestions.

Finally, I note from the cover letter that the PLOS Genetics editors may have had concerns with 'previous publication' of some of this material on bioRxiv. It is my opinion that this should in no way negatively affect publication of this submission, which is clearly a distinct piece of work.

John Lees

Main comments

> *1.1 The main advantage of this method is its representation of variants, rather than improvements or changes to the underlying model. As such, it would be nice if the examples of biological novelty provided were arrived at clearly through the use of unitigs rather than k-mers. In the final section of the results, could the existing methods (bugwas/SEER/HAWK) make the same discoveries using k-mers? Can the authors make it clear how their use of unitigs as variants lead to potentially novel results, rather than use of bacterial GWAS generally?*

We thank the reviewer for this remark.

First, we acknowledge that the position of the final section of the previously submitted manuscript was misleading. We did not mean that the other GWAS methods would not find novelties. We wanted to highlight that our method was able to find original hypotheses. We hope that the modified order of the sections avoids this ambiguity. We also modified the introduction of this section to avoid any confusion. See in the Results section '**DBGWAS provides novel hypotheses**' (check the red text in lines 182-184).

> "In addition to resistance markers, all three k-mer-based approaches reported several unknown variants, not described in the context of resistance."

Second, we would like to clarify an important element regarding the difference between testing k-mers and unitigs : the set of statistical tests performed by our method is identical to the set obtained by testing presence/absence patterns of all k-mers, after duplicate removal. Indeed the presence/absence pattern of each unitig is the same as the pattern of the k-mers it summarizes so the compaction does not create new patterns, and the set of unique patterns obtained after removing duplicates is the same whether using unitigs or k-mers. We apologize if this point was already clear but we wanted to be sure that there was no confusion about it. To avoid any confusion for the reader, we added a paragraph explicitly explaining this in the Methods section '**Unitig presence across genomes**' (check the red text in lines 404-408).

> "Importantly, both k-mers and unitigs lead to the same set of distinct patterns across the genomes. Indeed, every unitig represents (at least) one k-mer, and conversely every k-mer is represented by one (single) unitig. When de-duplicated, the two representations therefore lead to the same set of patterns to be tested for association with the phenotype."

Finally, as shown in Tables 2 and 3, the novelties found by the 3 k-mer-based methods are not the same. The differences are not a consequence of testing unitigs rather than k-mers, but arise from several elements:

- Even though the 3 methods are based on k-mers, their pre-processings differ and lead to different variant matrix. For example, SEER's informative k-mers are based on several values of k, while DBGWAS and HAWK only use one value. As a side note, as of now we were not able to establish a formal relationship between unitigs obtained by compacting k-mers with a given value of k, and informative k-mers obtained for k in [k_min, k_max].
- DBGWAS, HAWK and SEER use different testing procedures to test a given profile, which will lead to different decisions as to which ones are significant. As pointed out, these differences are not a consequence of the novelty introduced by DBGWAS, as we rely on an existing method (bugwas) for our testing procedure.
- The different downstream analyses may also impact the interpreted results. k-mers which do not map to a reference cannot be interpreted. By contrast, DBGWAS always returns a subgraph containing these k-mers. Even when no annotation exists, the topology and colors of the subgraphs may hint towards the nature of the causal variant. The pHS87b plasmid output by DBGWAS for amikacin resistance for example would only be found by HAWK or SEER if present in the references used in the downstream mapping, while DBGWAS provides a large linear subgraph with all red nodes pointing towards an unknown resistance-associated MGE.

We now highlight this last point, which, as underlined by the reviewer, is the bulk of our contribution: the downstream representation of unitigs in subgraphs, clustering them by neighborhood provides consolidated information which help in the interpretation – even without available annotation. See in the Results section '**DBGWAS facilitates the interpretation of k-mer-based GWAS**' (check the red text in lines 228-236).

> "The thousands of k-mers generated by HAWK and pyseer are of course also amenable to interpretation: to build our Table 3, we mapped these k-mers to references and extracted annotated variants which showed at least one hit. However, doing so required additional efforts and a working knowledge of the most appropriate annotated references. In addition, k-mers which do not map to the chosen reference cannot be interpreted. By contrast, DBGWAS always returns a subgraph containing these k-mers. Even when no annotation exists, the topology and colours of the subgraphs may hint towards the nature of the causal variant."

1.2. The annotation of the output graphs is very important, and it is difficult to interpret the results without it. As far as I can tell, the process for making this annotation seems to require some manual curation. Could more guidance on this be provided?

DBGWAS integrates a first-level automated annotation which can be customized depending on the studied topic. Indeed, by adding -nc-db and/or -pt-db options to the DBGWAS command, providing respectively paths to nucleic and/or proteic Fasta files with annotated headers, a Blast is computed during DBGWAS step 3 on all nodes of all output subgraphs. The annotation is consolidated at the subgraph level and also summarized in the summary page, with information of how many nodes each annotation maps to, the minimal E-value, and any other custom fields specified by the user. It can further be used in the subgraph page to query the nodes carrying a specific annotation and thus retrieve a gene location.

Importantly, DBGWAS works with any nucleotide or protein Fasta files as annotation databases straight away. However, the user can customize the annotation databases by changing the Fasta sequence headers to aid the interpretability of DBGWAS results. Indeed, DBGWAS reads specific tags and parses organized fields which are then used in the resulting summary page or to fill the annotation table in the subgraph page. A common example is compacting the annotation in the summary page by using abbreviations or gene class names, and expanding them to full names in the subgraph page. Other custom fields can also be included in the annotation table by adding specific tags to the headers. We also provide users with 2 annotated Fasta files, one containing known resistance determinants and the other containing all Uniprot bacterial proteic sequences (see https://gitlab.com/leoisl/dbgwas#dbgwas-in-a-nutshell-running-the-tool-in-one-example). It may happen that the Fasta files used for the automated annotation do not contain hits for some nodes and thus these nodes are finally not annotated. In this case, the user can either provide a more complete annotation file (containing for instance non-coding regions), or can do the annotation afterwards by manually blasting the node sequences (Fasta export can be retrieved with a right-click on the subgraph). Note that a DBGWAS run, which had initially been launched without the annotation step, can be re-annotated afterwards using the -skip2 option. In this case, only step 3 will be launched.

We heavily modified the Methods section '**Annotating the subgraphs**' (check the red text in lines 497-505) and added these two sections:
https://gitlab.com/leoisl/dbgwas#customizing-annotation-databases and
https://gitlab.com/leoisl/dbgwas/wikis/Customizing-annotation-databases to the tool's website in order to provide guidance to the user on this aspect.

Finally, to help even further the interpretation of the output graphs, we are currently working on a tool which will offer a prediction for the nature of the subgraphs. We are training predictive models over a large number of both real and simulated subgraphs which we know to describe SNPs or MGEs, and applying the trained models to the output of DBGWAS. We will not be able to add this option to the next release of DBGWAS, but it will be described in Magali Jaillard's PhD manuscript and offered as soon as possible in a future release.

1.3. The supplementary appendices are somewhat piecemeal, and I would suggest cleaning up and rearranging. Some of this material will be of particular interest to readers of PLOS Genetics. Specifically: S1 is probably best online, as part of the

software. S2: the effect of k-mer length is interesting and reassuringly robust. It should be moved to the main text (fig S7 and S8 especially). S3/Figure S15 is likewise an interesting comparison, and I would put it and the associated simulation methods in the main text. S4 doesn't seem to add anything new. S5 is overall either subjective or already well-known, and I would suggest removing.

We agree with the reviewer and thank him for his suggestions. Here is what we have changed:

- we removed the previous **S1 Appendix** from the supplementary materials and now provide it as a Wiki page available online at https://gitlab.com/leoisl/dbgwas/wikis/DBGWAS-web-based-interactive-visualization , and also referenced on the webpage for the tool at https://gitlab.com/leoisl/dbgwas#learning-how-to-use-the-dbgwas-web-based-interactive-visualization ;
- we integrated text from (previous) **S2 Appendix** and removed it from the appendices, and figs S7 and S8 in the Methods section '**cDBG construction**' (check the red text in lines 368-381). We kept only figures S9 and S10 as supplementary materials;
- we chose to keep the previous **S3 Appendix** as the new **S1 Appendix**, since integrating it in the main document would increase significantly the Methods section, which we would rather keep focused on the description of the 3 steps implemented in DBGWAS;
- we integrated information from **S4 Appendix**, which was not in the main document, in the Methods sections '**Significance threshold**' (check the red text in lines 447-452) and '**Graph neighbourhoods**' (check the red text in lines 467-468) and removed it from the appendices;
- we removed **S5 Appendix**.

Thanks to these modifications, there is only one remaining supplementary Appendix.

1.4. The WHO panel, used only for time and memory comparisons, should be removed. I think this cross-species collection is an unrealistic use of bacterial GWAS and leads to an inflation of computational resources required for all the methods tested. It would be more useful either to extrapolate times based on knowledge of the processes required, or test on a larger single-species dataset.

The reviewer is right. We removed the WHO panel and replaced it by 3 single-species large datasets, in order to analyse the computational performance at this scale. To do so, we gathered all genome assemblies of *Mycobacterium tuberculosis* (5,504), *Staphylococcus aureus* (9,331), and *Pseudomonas aeruginosa* (2,802) available on the NCBI RefSeq bacterial genome repository, and removed poor quality genomes. For each panel, we generated random binary phenotypes.

Extrapolating time for DBGWAS cannot be done globally, as the time not only depends on the number of analysed genomes, but also on their length and homology/complexity. In order to assess the scalability of DBGWAS, we built several sub-panels from these three large panels at size points of 100, 250, 500, 1,000, 2,500, 5,000 and 9,000 genomes. To build each such sub-panel, we sampled genomes at random from our large panel. The execution of DBGWAS on all these panels was also done on a cluster, instead of a single machine, and used 8 cores each. In order to account for subsampling and machine heterogeneity problems, each sub-panel was randomly built 10 times. We present in **(new) S9 Figure** the time and memory usage performance curves for all these panels, which allows a better understanding of DBGWAS performance behaviour and estimations of the computational resources usage on small and large panels with different genome plasticities. The longest run was for one of the S. aureus 9,000 genome subpanel, which took less than 5 days and 250 GB of RAM on 8 cores. DBGWAS performance was generally log-linear in the panel size, and is strongly affected by the number of input genomes, as well as their sizes and plasticities. The more clonal a genome population is, the less time and memory are required.

Likewise, when comparing DBGWAS against pyseer and HAWK in (new) S2 Table, we replaced the WHO panel by a random 2500-genome sub-panel for each species (Large TB, Large SA, and Large PA).

We added several sections in the text to describe the modifications we did here. The large panels are introduced in the Methods section '**Datasets**' (check the red text in lines 532-534) and detailed in the new Methods section '**Large panels**' (check the red text in lines 575-584). The study of the scalability of DBGWAS to these very large panels is detailed in the new Results section '**DBGWAS is fast, memory-efficient, and scales to very large panels**' (check the red text in lines 275-284 and S9 Fig). The comparison of the computational efficiency between DBGWAS, pyseer and HAWK was extended and is described in the same subsection (check the red text in lines 285-303).

Minor comments

1.5 Author summary: 'Any possible type of genetic variation'. Do de Bruijn graphs represent copy number variants, large structural changes and inversions well? If not, a short discussion in the introduction about exactly which variants can be found (SNPs, short and large indels?) and which can't would be useful. Also, only common locus variants will work using this approach. This statement is also made at the start of the discussion, and should be modified there too.

We completely updated the **Author summary** and the **Discussion** (check the red text in lines 306-310) sections as suggested by the reviewer, in order to remove any ambiguity.

1.6 Introduction: Another bacterial GWAS method that should be referenced is treeWAS (http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005958).

We now reference treeWAS in the **Methods** (in lines 412-413):

"Population structures can be strong in bacteria because of their clonality [5, 6, 57, 58]."

[58] is the reference to treeWAS.

1.7 Introduction: A useful feature of traditional SNP-based GWAS is the local linkage-disequilibrium, which can be seen on Manhattan plots and increases power to find signals. I think the analogue here is the inclusion of nearby nodes in the graph. A discussion about LD in the context of unitigs would therefore be useful. 'LD' appears in table 1, but I am not sure how it is defined.

We did not specifically work on LD within (nor between) the subgraphs, even though would be an interesting field to explore. In particular, the subgraph concept may help highlight local LD, which is not straightforward in reference-free k-mer-based methods. However, subgraphs are not always the right entity for local LD as connected unitigs can represent distant regions which just happen to share a subsequence.

When we mentioned LD in Table 1, we based our tags on the annotation: we hypothesized for example that subgraphs mapping to the SCC*mec* cassette were in LD with the mecA gene. We now modified this part by adding a quantitative LD information and provide in **Table 1** a measure of the correlation (r^2 value) of each subgraph with the first subgraph. It is computed between the minor allele presence patterns with the min_q of the first subgraph and the considered subgraph.
This measure is a good proxy for LD. As an example, in *S. aureus* resistance to methicillin, subgraphs annotated as parts of the SCC*mec* cassette have an r^2 value > 0.9, and can be considered to be in LD with the causal gene.

1.8 *Results: The datasets used are relatively small for GWAS. Can the authors predict how the tool scales in terms of memory and CPU usage for larger datasets, and give an estimate for 10^3-10^4 samples? This could also be noted in methods lines 305-315.*

We included in the manuscript a new study to evaluate the scalability in terms of time and memory usage of our tool. As mentioned in our answer to major comment 1.4, scalability does not only depend on the number of genomes but also on the nature of the genomes. The new S9 Figure highlights the performance of DBGWAS for 3 different panels, and increasing sizes. Please see the answer to major comment 1.4 for more details.

1.9 *Results: A further potential advantage of using unitigs is the smaller number of patterns, and therefore lower multiple testing burden. For the test datasets, could you add the number of k-mer as well as having the unitig patterns in supplementary table 1, and state this advantage in the text?*

The multiple testing burden is not lowered when using unitigs, as both k-mers and unitigs lead to the same de-duplicated pattern matrix (see also response to major comment 1.1). This point is now explicitly mentioned in Section "**Unitig presence across genomes**" (check red text in lines 404-408), it indeed deserved some clarifications. Moreover, **S1 Table** now provides for each species-antibiotic couple the number of distinct k-mers, distinct unitigs, and distinct de-duplicated patterns (which is the number of association tests performed).

1.10 *Results: I wondered how well the unitig approach would cope with complex regions of the genome, as I was concerned that close SNPs may split up the graph into low frequency paths and lose association power (a known problem with k-mers). [I tested this by using DBGWAS to run an analysis for beta-lactam resistance in S. pneumoniae, which is caused by mosaic alleles of three pbp genes. DBGWAS, with default settings, returned five hits, which using blastn of the sequences of the most significant nodes of the graph against a reference genome could be determined to be* pbp2b*,* pbp2x*,* pbp2x*,* pbp1a*, and* pbp2x *(ordered by q-value). I was very impressed by this result, and it could be a useful example for addressing this concern. If the authors find this example useful at all the output from DBGWAS is here: https://www.dropbox.com/s/equet6znoylzzkk/mass_pen.tar.bz2?dl=0. Resource usage: 7.2 Gb RAM and 2.6 hrs CPU. Dataset details: http://pyseer.readthedocs.io/en/master/tutorial.html]. Perhaps this potential concern could be noted in the text, and how DBGWAS would cope with such highly variable regions. Either this example or a simulated phenotype with adjacent SNPs each contributing some variance explained could be instructive.*

We thank the reviewer for this very valuable example of mosaic genes. We found this example useful and very illustrative, and we created a new section on DBGWAS gitlab page for user examples (see https://gitlab.com/leoisl/dbgwas#user-case-studies). We added the reviewer's results in this section, and used it as an illustrative example in section '**DBGWAS facilitates the interpretation of k-mer-based GWAS**' of our manuscript (check the red text in lines 249-254). We noted however that the subgraphs in the output were not annotated. So, if the reviewer wishes to either send us all DBGWAS output folder or the raw input, we could recompute the third step with an annotation.

1.11 *Results: A few times specific amino acid changes are mentioned (e.g. lines 103, 211), which I think are known causal variants. Is it possible to achieve this single mutation resolution with the unitig approach in any of the datasets? An attempt, or discussion of, such fine-mapping would be interesting. It should also be made clear exactly how/whether these specific mutations relate to the DBGWAS output.*

The amino-acid changes are indeed not obtained directly from DBGWAS. DBGWAS highlights the local polymorphism thanks to the red-*versus*-blue unitigs, and looking at the sequences (length and base pairs) of the red/blue unitig often provides a resolution at the SNP level. However, a mapping on UniProt referenced sequences (with amino-acid positions) was necessary to provide the information presented in lines 103, 211. We added a sentence to explain how we retrieved these positions (check the red text in lines 110-111):

> "Mapping the sequence of these unitigs on the UniProt database revealed an amino-acid change at L83S, right in the enzyme binding site."

1.12 *Results: I am surprised that SEER used so much memory and run-time, and I think this is probably due to quirks in the static binary release. I would appreciate it if our updated version, pyseer (https://goo.gl/EYmS4D), could be used in this comparison instead. Also, could the authors clarify whether the large memory use was from SEER itself, or during k-mer counting? (If cDBG offers a more scalable way to count k-mers, than that's great.)*

We thank the reviewer for pointing out pyseer, the updated version of SEER. We fully replaced SEER by pyseer in this work (including the text and the benchmarks).

When running SEER, we used fsm-lite for k-mer counting, Mash v2.0 for population structure estimation, and then SEER itself. Regarding the runtime, on the M. tuberculosis and P. aeruginosa panels, the k-mer counting procedure is the most expensive step. On the S. aureus and the WHO panels on the other hand, the SEER step was the most expensive one. Regarding memory usage, fsm-lite always used several Gigabytes, while Mash and SEER never exceed 150mb. The large memory use of the pipeline was therefore from fsm-lite in the k-mer counting procedure, as suspected by the reviewer. The raw numbers we logged from SEER's executions can be seen here:

SEER runs	time (hh:mm:ss)				mem (gb)		
Panel	fsmlite	mash	SEER	Total	fsmlite	mash	SEER
Etham	7:37:00	0:08:00	0:45:00	8:30:00	93.6	<=150mb	<=60mb
strepto	8:28:00	0:08:00	1:34:00	10:10:00	102.4	<=150mb	<=60mb
rifam	8:02:00	0:08:00	4:00:00	12:10:00	104.7	<=150mb	<=60mb
ethio	2:48:00	0:03:00	0:42:00	3:33:00	34.8	<=150mb	<=60mb
cipro	5:37:00	0:05:00	N/A	N/A	53.8	<=150mb	N/A
erythro	6:18:00	0:05:00	62:55:00	69:18:00	53.8	<=150mb	<=60mb
methi	3:13:00	0:02:00	N/A	N/A	27.1	<=150mb	N/A
meropenem	5:07:00	0:03:00	1:54:00	7:04:00	34.7	<=150mb	<=60mb
levofloxacin	2:16:00	0:01:00	0:30:00	2:47:00	14.5	<=150mb	<=60mb
amikacin	4:59:00	0:03:00	1:52:00	6:54:00	34.7	<=150mb	<=60mb
WHO	8:43:00	0:06:00	95:56:00	104:45:00	51.5	<=150mb	<=60mb
piperacilin	5:43:00	0:03:00	2:02:00	7:48:00	34.7	<=150mb	<=60mb

We also would like to point out a mistake on our end. We executed fsm-lite as: `fsm-lite -v -l fsm_files.txt -t tmp_idx -s 10 -S 593`, following the SEER tutorial at https://github.com/johnlees/seer/wiki/Tutorial . However, we did not realise that the parameters -s and -S should vary according to the panel size. So, all the SEER runs presented in the first submission of this work had this problem. We fixed this on the pyseer runs following the instructions at https://github.com/johnlees/seer/wiki/Usage#count-your-k-mers . Since we use a 1% minor allele frequency cutoff in DBGWAS and pyseer, we set `-s = 0.01 * PS` and `-S = 0.99 * PS`, where `PS` is the panel size.

In order to: 1) avoid errors on executing other tools again; 2) be transparent on how we ran pyseer and HAWK; 3) provide reproducibility of our results; we decided to make public the scripts we used to run pyseer (available here: https://gitlab.com/leoisl/DBGWAS_support/tree/master/scripts/pySEER) and HAWK (available here: https://gitlab.com/leoisl/DBGWAS_support/tree/master/scripts/HAWK_0_8_3_beta). These links were added to the Methods section '**k-mer-based GWAS**' (check the red text in lines 621 and 637).

To help the readers understand pyseer and HAWK performances, and to solve potential similar readers doubts, we also modified the (new) **S2 Table** by adding how much time and memory each step of pyseer (fsm-lite, Mash v2.0, and pyseer itself) and HAWK (Count k-mers, HAWK itself, ABYSS and Stats) took, besides the total time and maximal memory of both pipelines.

pyseer took longer to run than SEER. We believe that it might be due to: 1) pyseer having more k-mers to analyse, due to our incorrect usage of the fsm-lite parameters in SEER's pipeline; 2) SEER filtering out k-mers by raw p-value at the begin of the process, while we used the --lrt-pvalue parameter when running pyseer, which does lrt-p-value filtering at the very end of the process. Consequently, pyseer has more computational steps to perform.

The main sections that were updated due to replacing SEER by pyseer were:

1) Results section '**DBGWAS reports expected variants without prior knowledge**': SEER and pyseer found the same resistance determinants for *P. aeruginosa* levofloxacin (**new Table 2**). pyseer found more "Determinant described for other antibiotics" for *M. tuberculosis* streptomycin (**new Table 3**). Overall, the qualitative results are similar;
2) Results section '**DBGWAS is fast, memory-efficient, and scales to very large panels**': a breakdown of pyseer's performance is provided in **S2 Table**;
3) Methods section '**k-mer-based GWAS**', where we describe how we installed and ran pyseer, and provide pyseer's scripts for reproducibility reasons.

Finally, we do not believe that cDBGs offer a more scalable way to count k-mers, since in order to build unitigs, algorithms usually need to first find or count the k-mers themselves. In DBGWAS implementation, for example, the first step is to count k-mers using DSK (G. Rizk, D. Lavenier, R. Chikhi. (2013) DSK: k-mer counting with very low memory usage, Bioinformatics, 29(5):652-3), in order to build the DBG and then compress it into a cDBG.

1.13 Results: In the method comparison, the most useful statistics to state for each run and method would be discovery power and false discovery rate rather than the number of signals found. The fairest comparison with k-mer methods, rather than the total number of individual k-mers, would be to map these to a reference and compare the number of genes they overlap (or take 1kb windows), which is usually how the downstream interpretation of these results is performed.

We agree that users could map k-mers output by SEER or HAWK to a reference (when available and annotated). By displaying the number of signals found (k-mers for SEER/HAWK, subgraphs for DBGWAS), we wanted to highlight the main contribution of DBGWAS, which is to aggregate the results in a more interpretable form, compact overlapping k-mers with identical presence/absence profiles in a single unitig, and exploit the colored DBG to understand its genomic context. After DBGWAS is run, the user is left with subgraphs which often correspond directly to SNPs, hotspots containing several SNPs or MGEs - admittedly, subgraphs are sometimes less clear, for example when several events are aggregated together, and we now acknowledge this point in the '**DBGWAS facilitates the interpretation of k-mer-based GWAS**' section (check the red text in lines 255-274). Notably even if no annotation is available, these subgraphs retain their interpretability as -unannotated - SNPs or MGEs.
We modified Section '**DBGWAS facilitates the interpretation of k-mer-based GWAS**' (check the red text in lines 228-236) to make it clear that the k-mers output by SEER and HAWK could also be post-processed (we actually perform this post-processing to build Tables 2 and 3) but we also think it is useful to show how many k-mers they would output because this post-processing is a non trivial, additional step.

Regarding power and FDR, they would require a set of known negative examples to be computed whereas we only have access to a small number of validated positive examples. Consequently, we chose to compare the number of known causal elements detected as significant by every method after post-processing.

1.14 Discussion line 236: What exactly are the 'strong prior assumptions'?

We modified the first paragraph of the **Discussion** (check the red text in lines 308-310):

> "It performs as well as the current SNP- and gene-based gold standard approaches for retrieving known determinants, from genome pre-assemblies and without relying on annotations or reference genomes."

1.15 Discussion line 248: The HAWK paper tests the use of k-mers in human genomes (which also contain complex variation and an accessory genome), so may be worth citing again here.

We cite HAWK paper in the **Discussion** when mentioning work on human tumours (check the red text in line 321):

> "...they pave the way to GWAS on highly plastic bacterial genomes and would also be useful for microbiomes [43] or human tumours[13]." ([13] is HAWK)

1.16 Discussion line 251: I would consider allowing for continuous phenotypes in the initial release – many users will have these. I think the underlying model being fitted is already continuous?

We are currently working on this feature, which still requires some work on the interfaces (currently tailored for binary outputs), and further beta testing to be safe. Unfortunately we will not be able to integrate it in the initial release but plan to add it soon.

1.17 Discussion: Final paragraph. The main advances are the more intuitive variant representation, potentially lower multiple testing burden, and integrated software and visualisation. I would state these in the final paragraph instead of or as well as the general use of GWAS and antibiotic resistance mechanisms, which have been described before.

We modified the last paragraph of the **Discussion** (check the red text in lines 341-343):

> "Our integrated software and visualisation tools offer an intuitive variant representation, hence will provide future users with an enhanced insight into genotype to phenotype correlations, in all domains of microbiology, beyond that of antibiotic resistance."

1.18 Methods lines 320-324: While removing duplicate patterns is necessary to get the correct q-values, this could potentially lose information. It would be better to test all unitigs but use an FDR correction based on the number of patterns. This is partly noted in line 351-353, but I was still unclear whether a pattern can still be related to multiple unitigs in the final step (step 3). Looking at the output files I think this is done, but this needs to clarified in the text.

We modified text at the beginning of the Methods section '**Interpretation of significant unitigs (step 3)**' to make it clear that step 3 is done at the unitig level, without loss of information (check the red text in lines 439-444):

> "The LMM is used to identify de-duplicated minor allele presence patterns significantly associated with the phenotype at a chosen FDR level. While the testing step is done at the pattern level, the interpretation of the selected features is done at the unitig level. As a result of the de-duplication procedure, a given pattern may correspond to several distinct unitigs. To faithfully interpret the results, all the unitigs corresponding to the significant patterns are retrieved and are assigned the q-value of their pattern."

1.19 Methods line 342: It needs to be stated exactly how W is calculated from Z.

We detailed the construction of W from Z in the Methods section '**Testing unitigs for association with the phenotype (step 2)**' (check the red text in lines 431-432):

> "W is estimated from the Z matrix, which includes duplicate columns representing both core and accessory genome. More precisely, denoting $Z = U\Lambda V^\top$ the singular value decomposition of Z, we use $W = U\Lambda^{\frac{1}{2}}$."

1.20 Methods: The potential sensitivity of results to the SFF concerned me at first, but after seeing the various supplementary material and description I better understood its use. The use of the default value on trial data also seemed appropriate. Might I suggest an alternative option to set the number of connected graphs reported in the output, rather than the top 100 unitigs themselves? Users will care mostly about each graph and its annotation, rather than individual unitigs. From the perspective of downstream lab follow-up, I think it is probably sensible to set a maximum number on these graphs/regions.

We thank the reviewer for the idea. We study how to implement it in a future release, however this proposition has the same cons of the current SFF parameter: it may generate subgraphs with high q-values, as q-values are computed at the unitig level and not at the subgraph level. For now, we are working on improving this by applying a double filter, on both top patterns and q-value threshold together.

1.21 Methods line 549: I didn't like the term resistome/RWAS, and found it confusing when used in the results. Are we sure these variants always explain 100% of resistance and/or are all possible resistance conferring variants? The analysis itself is useful, but it would be clearer to describe this as a comparison between unitigs and known/ positive control resistance SNPs and genes under the same association model.

We replaced 'RWAS' by 'resistome-based association study' and better defined in the Methods (new) section '**Resistome-based association studies**' what we mean by 'resistome' (check the red text in lines 585-589):

> "We benchmarked DBGWAS against a targeted approach to ensure its ability to retrieve all expected resistance determinants. We thus performed association studies under the same model, using as input a collection of known causal resistance SNPs and genes, defining the resistome."

1.22 Table 1: Effect sizes would probably be more usefully represented as odds-ratios. Could ORs also be used as an option to colour the DBG (methods line 383), and stated for the result on line 208 (which does not currently incorporate the population structure used in calculation of the associated q-value).

OR is indeed a more interpretable measure of association between a covariate and a binary phenotype. However, a crude OR estimation would not take into account the effect of population structure and we could observe discrepancies between OR estimation and min_q ordering. That is why we chose to present in Table 1 the effect estimated by DBGWAS, i.e the linear effect of a unitig on a binary phenotype adjusted for population structure. Nevertheless, as suggested by the reviewer, in a future version of our tool based on a logistic model, we could add population-structure-adjusted ORs.

1.23 Table 1: How were the annotations in the final column determined? Was it through the use of unitigs, or previous knowledge?

We used the unitig sequences to find their annotation (thanks to the automated annotation included in DBGWAS, completed when needed by manual annotation using blast) (column 'Annotation'), then we used previous knowledge from the literature (see 'Reference' column in **S4-6 Tables**) to define if the annotation was already described as a positive. In the case of SNP, we also check the SNP position to define if it had already been described (using the mubii tool for instance).

1.24 *Figures: I would suggest a re-ordering such that the overview figure (currently 4) is first, and put figure 1 in the supplementary (as the DBG representation is clearly shown with the real data in figures 2 and 5).*

We liked the idea to present (previous) Fig 4 at the beginning of the document, and integrated it to introduce the **Results** section, as **Fig 2**. We chose to keep Fig 1 in the main document, as it helps readers not familiar to k-mer-based methods to understand the link between the DNA sequences and the nodes of the graph.

1.25 *Figure 2: This is the actual output of DBGWAS? Make this clear in the legend and the text, as I think this will appeal to potential users.*

We thank the reviewer for the suggestion. We modified the **(new) Figure 3 legend accordingly** (check the red text in Figure 3's caption):

"All subgraphs are snapshots taken from DBGWAS interactive visualisation and are available online."

1.26 *Figures S2-S6. These are hard to read as figures, as they contain a lot of information. They are however useful for those readers interested in the SFF. Could they also be hosted online somewhere for interactive use?*

These figures are compilations of the raw outputs from several runs of DBGWAS, and cannot be used interactively. However we added the 35 runs used to generate these figures in the page providing all experiment results. They can be accessed here: http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/experiments/index.html#DBGWAS_all_results_different_SFF . We now reference this link in S2-S6 Figs.

Typos:

1.27 *Title: 'A fast and agnostic method'. Agnostic to what? Presumably type/source of variation – this needs to be stated.*

We thank the reviewer for his suggestion but we would rather not change the title for the sake of concision. We hope the information given in the abstract / author summary helps to avoid any confusion but we would be happy to update it if the reviewer is concerned it may not be clear enough.

1.28 *Title: You may consider changing from kmers to k-mers throughout, which is the more common form (useful for pubmed indexing etc).*

We modified as suggested by the reviewer.

1.29 *Abstract: First sentence of conclusion should be reworded. Without prior knowledge of what? Also change 'Our novel method proved its efficiency' -> 'Our computationally efficient method was able to'*

We rephrased this sentence (check the red text in the Abstract):

"Our method is computationally efficient and is able to retrieve phenotype-associated genetic variants such as local polymorphisms and MGEs without relying on prior annotation or reference genomes. Experiments took one hour and a half on average, and produced a compact set of meaningful subgraphs, thereby facilitating the interpretation of the results."

1.30 *Author summary: Sentence starting 'Thanks to the use of cDBG'. I don't understand this sentence.*

We wrote a new version of the author summary, where we removed this sentence.

1.31 *Results line 191: 'blue or red' -> state this in terms of the actual phenotype*

We modified the sentence (check the red text in line 91):

"...with a fork separating susceptible (blue) and resistant (red) strains."

1.32 *Discussion line 259: 'the classic methods' – explain what these are.*

We modified the sentence (check the red text in line 333):

"classic phenotypic methods."

1.33 *Methods line 312: 'size of the genome' – should this be size of the pangenome?*

In this particular case, the correct term is 'size of the genome', since the procedure described produces a unitig-presence/absence vector representation of *one* given genome.

1.34 *Methods line 323: 'below 0.01' – note that the user can specify this (important for larger or smaller datasets, as a count rather than frequency is most appropriate).*

We modified as suggested (check the red text in lines 402-403):

"- the user can specify this threshold using the -maf option."

1.35 *Figure 3: This is a table, not a figure.*

Previous Figure 3 is now splitted into two tables, **new Tables 2 and 3.**

1.36 *Appendix S1: I got a 404 error for all the links*

It is now fixed. Thank you for pointing this. We were able to verify also that some PDF readers have problems correctly opening URLs spanning more than one line (i.e. including a line break). We made sure that all URLs are correct, but if this problem persists, please try clicking on the URL (copy-paste seems to be more problematic). We verified that all links worked with Adobe Acrobat PDF Reader.

1.37 *Tables S3-5: Make sure these are available as supporting data (text file or similar) so they can be used by others, and not just as a PDF.*

All GWAS results in **(new) S3-S7 Tables** are now provided as Excel files.

1.38 *Figure S5 legend. There is an incorrectly rendered '<' which should probably be \textless.*

It's now fixed. Thank you for pointing this.

Notes on software

I installed and ran DBGWAS on the provided test dataset, then on my own test dataset (as noted above).

Here are some notes I made while running through this process which may help improve usability:

1.39 The static binary did not work for me on two different Linux machines due to a version mismatch of glibc (is this because they had Intel rather than AMD architecture?). Would it be possible to make a fully static binary, or cross-compile for Intel too?

This response concerns the problems raised by multiple reviewers regarding the difficulties of installing and/or running DBGWAS, like precompiled binary incompatibilities, problems with the building process, etc. We are grateful for the reviewers for pointing out these problems, as we tested the packages in some of our and colleagues' machines, but we had not experienced any problems, so we were totally unaware of them. The detailed input of the reviewers allowed us to solve several problems in this aspect.

1. Precompiled binary issues

Regarding the precompiled binary, **we were able to reproduce the problem stated by Reviewer 1 (item 1.39)** by running the DBGWAS precompiled binary v0.5.1 on Debian-7.11.0:

```
debian@debian:~/051_test/bin$ ./DBGWAS
./DBGWAS:  /lib/x86_64-linux-gnu/libc.so.6:  version  `GLIBC_2.17'
not found (required by ./DBGWAS)
./DBGWAS:  /lib/x86_64-linux-gnu/libc.so.6:  version  `GLIBC_2.14'
not found (required by ./DBGWAS)
```

DBGWAS also uses GEMMA for the statistical test, but GEMMA binary also failed on Debian-7.11.0:

```
debian@debian:~/051_test/bin/DBGWAS_lib$ ./gemma.0.93b
./gemma.0.93b: /lib/x86_64-linux-gnu/libc.so.6: version
`GLIBC_2.14' not found (required by
/home/debian/051_test/bin/DBGWAS_lib/sharedobjects/libgsl.so.0)
```

We agreed that a good solution to build a portable binary was to use Holy Build Box (http://phusion.github.io/holy-build-box/) following Páll Melsted's blog (https://pmelsted.wordpress.com/2015/10/14/building-binaries-for-bioinformatics/) on how he did it for kallisto (NL Bray, H Pimentel, P Melsted and L Pachter, Near optimal probabilistic RNA-seq quantification, Nature Biotechnology 34, p 525--527 (2016)). By doing so, the new precompiled DBGWAS binary (v0.5.2) is more portable: it should work on pretty much any glibc-based x86 and x86-64 Linux distribution released since 2007 (e.g.: Debian >= 6, Ubuntu >= 10.04, Red Hat Enterprise Linux >= 5, CentOS >= 5, etc).

However, the DBGWAS package not only relies on the DBGWAS binary, but also on 3 binaries from the Blast suite (blastn, blastx, and makeblastdb), the GEMMA binary and the phantomjs binary. We found the Blast suite binaries and the phantomjs binary to be reasonably portable and thus we did not recompile them. The GEMMA binary, however, did not work on some systems, like the Debian-7.11.0. DBGWAS uses bugwas, which requires a modified GEMMA version 0.93 to work. The source code of this modified GEMMA version is available at https://github.com/danny-wilson/gemma0.93b .
We just forked this repository at https://github.com/leoisl/gemma0.93b , and used Holy Build Box (http://phusion.github.io/holy-build-box/) to create a static portable binary of this specific version of GEMMA. This new GEMMA binary is packaged with DBGWAS in v0.5.2.

2. Building process issues

Despite all our efforts, it could be that DBGWAS or the other binaries it uses might not work on a particular machine. To cover the case where the DBGWAS precompiled binary does not work, we improved the compilation process. We fixed two issues with DBGWAS compilation: 1) not recognizing CMAKE arguments (**like -DCMAKE_BUILD_TYPE=Release, as pointed out by Reviewer 2, item 2.13**); 2) wrong CMAKE version (**as pointed out by Reviewer 1, item 1.41**). Since we were able to compile DBGWAS using Holy Build Box, which is built CentOS 5 (release date 12 April 2007) with a simple compiler toolchain (GCC 4.8.2, GNU make, CMake 3.6.3), we believe now that the source code can be compiled on a vast number of linux distributions. To cover the cases where the other packaged binaries do not work, we introduced three new parameters allowing the user to change the Blast, GEMMA and phantomjs executable paths: -GEMMA-path, -Blast-path and -phantomjs-path. As such, the user can compile or download their own versions of these executables and tell DBGWAS to use them through these parameters, instead of the ones packaged. Since phantomjs could be complicated to compile, and it is not essential in DBGWAS (i.e. it only provides the components preview in the summary output page), we introduced the parameter -no-preview allowing DBGWAS to skip this step and thus do not call phantomjs.

3. Interfacing with R and bugwas

Reviewer 1 pointed out in item 1.42 that the main issue he had was interfacing with R and bugwas. Due to this, we now offer a command line option (-Rscript-path) allowing the path to Rscript to be user-specified.

4. Validation of the new package (DBGWAS v0.5.2)

To attest that the new precompiled binary truly works on older linux distributions, we tested DBGWAS v0.5.2 precompiled binary on Debian 7.11.0, Ubuntu 14.04.5 and CentOS 7. However, we conjecture that it works on all distributions supported by Holy Build Box.

We are grateful for the reviewers' feedback, which allowed us to improve the compatibility of DBGWAS precompiled binaries and its building process. We do hope they will now be able to use the software more easily.

1.40 The download is very large and takes a long time to extract – is this because of boost? I think it is possible to just include the boost libraries you are using, I would consider doing this.

Thanks for pointing out the problematic large size of the repository. We just include the Boost library version we need, and also as a compressed file - it is extracted only when building the package. However, we hope the users will download the new portable precompiled binary instead of cloning the repository and compiling DBGWAS. DBGWAS v0.5.2 precompiled binary package is indeed still heavy, (182 Mbs). We could verify that more than half of it, 96 MBs, comes from the packaged sample example. The remaining 86 MBs comes mainly from the statically built binaries, which increase a lot the size of the executable due to the packaging of its dependencies in the executable. We could indeed remove the sample examples from DBGWAS package, reducing the package size from 182 MBs to 86 MBs, but we have a slight preference on providing a heavier, but complete package.

1.41 I managed to compile from source ok, but the minimum version of cmake required is v3.1.0 (not v2.0 as stated) due to one of the dependencies.

We switched to CMAKE v3.6. This is required by Holy Build Box (http://phusion.github.io/holy-build-box/) to build the portable binary (see response of item 1.39), and this CMAKE version also solve this issue. Thanks for pointing this out.

1.42 *The main issue I had was the interfacing with R and bugwas. The Rscript on my path was the wrong version for bugwas, and (I think) due to the way the command is being run from C++ none of the bash tricks I tried to fix it would work. In the end this required me to edit the hard-coded R command in the source code and recompile from scratch, which is something I think many users would avoid. I would offer a command line option allowing the path to 'Rscript' to be user-specified, but also consider allowing the whole R command to be changed without recompiling.*

Please see response of item 1.39

Reviewer #2:

Jaillard et al. present their software tool, DBGWAS, as an alternative to existing bacterial GWAS methods (and possibly GWAS methods more broadly) for identifying genetic variants associated with a phenotype. Their major argument for the utility of their approach is that it strikes a balance between the flexibility of a k-mer based approach for detecting a range of types of genetic variation that may be associated with phenotype, and the interpretability of mapping-based methods. They achieve this by taking 31-mer words from assembled contigs and forming a collapsed de Bruijn graph from them. 'Unitigs' formed in the de Bruijn graph are then attributed to different isolates based on their constituent k-mers. After collapsing redundant presence/absence patterns, these unitig patterns are then tested for association with phenotype using bugwas. A graphical interface aids in visualising and interpreting results.

This is a novel contribution to the suite of tools available for performing GWASs, and I think it will prove popular provided the software is easy to install and enough help is provided for interpreting results. I like the idea of this tool, and look forward to trying it on my own data, however I think the presentation and benchmarking of the software in this paper could be improved upon. Critically, installation of the software needs to be more straightforward, and more help needs to be provided in interpreting graphs that don't follow the straightforward structure presented in Fig 2.

Minor comments:

2.1 *Trees showing the population structure for each test dataset, and labelling phenotype on tips would give us a better idea of the structure of the datasets and differences between them.*

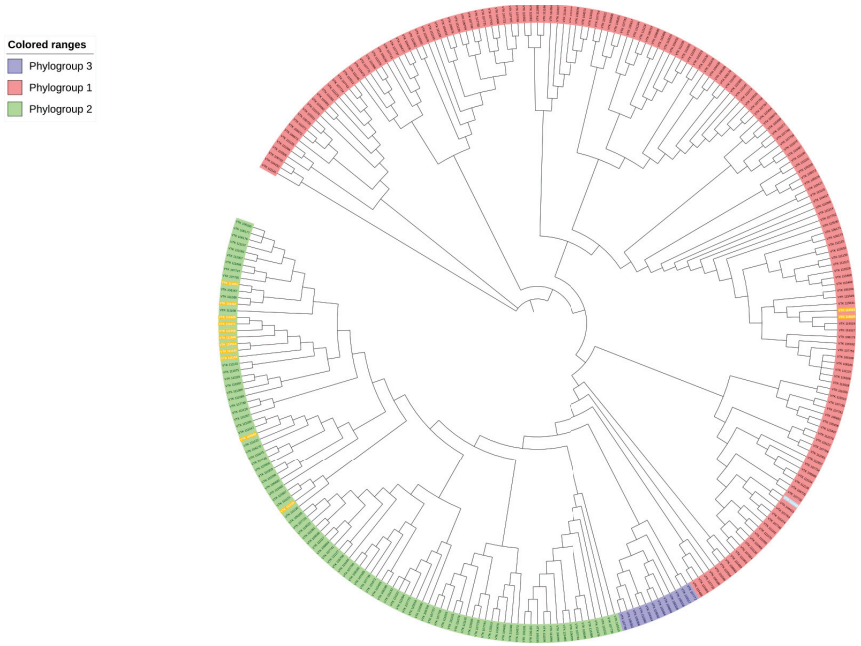
We thank the reviewer for this comment. DBGWAS provides these trees with tips showing the observed and predicted phenotypes when the user specifies the -newick option and provides a newick file (these trees are produced by the bugwas package). We provide an example of these trees at http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/full_dataset_visualization/ (click on the "Show figures on lineage effect" button). We added a paragraph in the Methods section '**Testing unitigs for association with the phenotype (step 2)**' to make this clear (check the red text in lines 421-424):

> "DBGWAS optionally provides bugwas lineage effect plots when the user specifies a phylogenetic tree using the -newick option. An example of the generated figures is available at http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/full_dataset_visualization/ ."

2.2 *Listing counts of isolates carrying hits not previously associated with phenotype is not sufficient to indicate a significant difference, since most isolates carrying a unitig could be all from one clone. Allowing the mapping of the unitigs to a tree would allow better assessment of how convincing a hit is.*

The reviewer is right, counts of isolates are not corrected for the population structure and are not sufficient to indicate a significant difference. Our testing procedure relies on bugwas to take into account the population structure, so the resulting beta coefficient and associated q-values are the reliable measures of significance. The node colors in subgraphs output by our step 3 are derived from the estimated beta coefficient from the LMM, not from a crude statistic based on isolated counts. Both in DBGWAS and our results section, counts are only provided as an additional information. We verified that the counts reported for the second subgraph of amikacin resistance for *P. aeruginosa* (13/47 resistants, 1/233 sensitives) were not triggered by a single clone: as shown on the figure below, the 13 resistant strains carrying the significant unitig (highlighted in orange) belong to two different phylogroups. 11 of them are reasonably spread out in Phylogroup 2 while 2 others belong to Phylogroup 1, which also contains the sensitive strains carrying the unitig (highlighted in blue). We now clarify this point in the **Results** section (check the red text in lines 199-202):

> "This annotation was not an artefact of the population structure, properly taken into account by the linear mixed model. Indeed the 13 resistant strains corresponded to distinct clones belonging to two phylogroups, one of them containing the susceptible strain."



As suggested in comment 2.1, the -newick option can also be used to execute the bugwas analysis of lineage effect and output a tree representation of the strains and p-values per PC.
Remapping significant unitigs from a subgraph to a tree is also a nice idea which we would like to implement in a future version of DBGWAS. We thank the reviewer for the suggestion.

2.3 I'd like to see file S1 on the webpage for the tool, because the manual isn't quite as extensive as it could be.

We thank the reviewer for the suggestion. We removed the previous **S1 Appendix** from the supplementary materials and now provide it as a Wiki page available online at https://gitlab.com/leoisl/dbgwas/wikis/DBGWAS-web-based-interactive-visualization , and also referenced on the webpage for the tool at https://gitlab.com/leoisl/dbgwas#learning-how-to-use-the-dbgwas-web-based-interactive-visualization ;

2.4 I believe "S2 Section" needs to be changed to "S2 Appendix" in the main text. I found S2 Appendix very helpful, thank you for including it. I did notice that in Figure S8, k=31 was not included for visualisation, which seemed odd since it was the final choice for k-mer length.

Thank you for the suggestions. We integrated text from (previous) **S2 Appendix** (and removed it), and Figs S7 and S8 (n**ow Figs 4 and 5**) in the Methods section '**cDBG construction**' (check the red text in lines 368-381). We kept only figures S9 and S10 as supplementary materials. In the new Fig 4 (previous S8 Fig), we replaced the graph with k=15 by the one with k=31.

2.5 Number of samples in each dataset should be added to Table 2.

As suggested, we added the number of available genomes in the **new Table 4** (previous Table 2).

2.6 Where reference [27] was cited, I'd like to see several studies comparing the accuracy of genomic and phenotypic antibiograms, since this is what is stated in the text.

As suggested, we added three other references in the **Discussion** section (check the red text in lines 334-335):

"Several studies have already demonstrated that in some cases, genomic antibiograms can be at least as good as phenotypic ones [30, 46-48]."

These references are:

30. Gordon N, Price J, Cole K, Everitt R, Morgan M, Finney J, et al. Prediction of Staphylococcus aureus antimicrobial resistance by whole-genome sequencing. Journal of clinical microbiology. 2014;52(4):1182-1191.

46. Kos VN, Deraspe M, McLaughlin RE, Whiteaker JD, Roy PH, Alm RA, et al. The resistome of Pseudomonas aeruginosa in relationship to phenotypic susceptibility. Antimicrobial agents and chemotherapy. 2014; p. AAC-03954.

47. Bradley P, Gordon NC, Walker TM, Dunn L, Heys S, Huang B, et al. Rapid antibiotic-resistance predictions from genome sequence data for Staphylococcus aureus and Mycobacterium tuberculosis . Nature communications. 2015;6:10063.

48. Moradigaravand D, Palm M, Farewell A, Mustonen V, Warringer J, Parts L. Precise prediction of antibiotic resistance in Escherichia coli from full genome sequences. bioRxiv. 2018; p. 338194.

Choice of benchmarking datasets:

2.7 The dataset composed of genomes of different species seemed like an especially unusual choice, as I haven't seen any studies do this before. For a k-mer based approach in particular this seems incredibly unlikely to identify any shared resistance mechanisms. I'm unsure what the purpose of this inclusion of this dataset was, but I think this part of the analysis should be removed.

Thank you for this comment. We removed the unrealistic WHO panel and replaced it by 3 single-species large datasets with random phenotypes, in order to analyse the computational performance at a comprehensive scale. More details are given in **item 1.4** as it was also a concern of Reviewer 1.

Choice of software comparisons

2.8 I was a bit unsure about the comparisons with other methods. In my mind, since this tool strikes a nice balance between k-mer and SNP/gene-based approaches, I'd like to see a comparison with both types or approach, in terms of results and runtime/memory requirements.

DBGWAS is doing the same set of association tests as one would with plain k-mers. Our contribution compared to testing plain k-mers lies in the ability of DBGWAS to aggregate the results in a more interpretable form, compacting overlapping k-mers with identical presence/absence profiles in a single unitig, and exploiting the colored DBG to understand its genomic context. Comparing with a k-mer approach in terms of results therefore amounts to comparing different ways to interpret the set of significant k-mers. Plain k-mers would need to be aligned against annotated references, leading to a lot of manual inspection. We thought that a relevant baseline would be HAWK and pyseer, which are other ways to work with k-mers.
Comparing with plain k-mers in terms of runtime and memory, amounts to quantifying the overhead caused by our step 3, which we now show in **new Table S2**.

Regarding SNP/gene-based approaches, we chose to compare the results of DBGWAS with those obtained using a resistome approach: for each species and each drug, we built a design matrix describing the presence of known causal genes and the SNPs within these genes in the genome of each sample, and applied the same statistical test used by DBGWAS to this matrix. Our rationale was to check that DBGWAS did not miss causal variants that could be recovered by SNP/gene-based approaches (regardless of its ability to discover new elements). We now provide an estimation of the runtime and memory usage required to build the resistome matrix for the examples presented in **Tables 2 and 3**.

2.9 For gene-based analyses its common to use gene presence/absence analysis based on a pangenome created from assemblies (e.g. the output from Roary or similar). This pangenome-based analysis isn't reference dependent, so the critique in the text that

standard gene-based approaches are limited based on the choice of reference is inaccurate.

Thank you for this remark. We modified the text in the **Introduction** (check the red text in lines 8-15):

> "The most common approaches are based on single nucleotide polymorphisms (SNPs), defined by aligning all genomes of the studied panel against a reference genome [1, 3, 4] against a pangenome built from all the genes identified by annotating the genomes [8], and on gene presence/absence, using a pre-defined collection of genes [5, 7]. The use of a reference genome becomes unsuitable when working on bacterial species with a large accessory genome - the part of the genome which is not present in all strains. On the other hand, methods focusing on genes are unable to cover variants in noncoding regions, including those related to transcriptional and translational regulation [9, 10]. Moreover, some poorly studied species still lack a representative annotation [11]."

Note: [8] refers to roary

2.10 The manuscript cited for HAWK is the BioRxiv version, which is quite different to the version recently accepted by eLife. Notably, the version of the software reported in BioRxiv did not allow correction for population structure. Also, the current version only includes two principal components when correcting for population structure by default, which tends to be inadequate for bacterial GWASs. In their benchmarking with real bacterial sequence data they use 10 PCs, which is more appropriate, but it's unclear from the methods whether this procedure was followed in the benchmarking in this paper. If including results from a comparison with HAWK, using the version that corrects for population structure is essential.

We thank the reviewer for pointing out the new paper and version for HAWK. We agree that in order to add HAWK to a bacterial GWAS benchmark, correcting for population structure is essential.

We ran the new version of HAWK, HAWK v0.9.8-beta, in the two panels we used to compare resistome-based association studies, DBGWAS and pyseer qualitatively: *P. aeruginosa levofloxacin* resistance (**new Table 2**) and *M. tuberculosis* streptomycin resistance (**new Table 3**). Unfortunately, HAWK found no significant k-mers for either of these two panels. By investigating HAWK's code and discussing with its author (our interaction is available here: https://github.com/atifrahman/HAWK/issues/5), we realised that HAWK v0.9.8-beta was using very strict k-mer filters. More precisely, we removed the following filters: https://github.com/atifrahman/HAWK/blob/master/hawk.cpp#L503-L510 , https://github.com/atifrahman/HAWK/blob/master/convertToFasta.cpp#L47-L50 , and https://github.com/atifrahman/HAWK/blob/master/convertToFasta.cpp#L71-L74 in the hope of retrieving some significant k-mers, to be able to assemble them and then check the causal variants they represent. These modifications did allow us to retrieve 2 contigs for *P. aeruginosa* levofloxacin resistance and 7 contigs for *M. tuberculosis* streptomycin resistance. Unfortunately, the downstream analysis of these contigs revealed that they did not correspond to the expected known positive causal variants identified by the resistome-based association study. On the other hand, the previous version of HAWK, HAWK v0.8.3-beta, which does not correct for population structure, was able to find 8 contigs for *P. aeruginosa* levofloxacin resistance and 2,038 contigs for *M. tuberculosis* streptomycin resistance. For the first panel, it found one of the two known positive causal variants, and for the second panel, it found the two. Thus, although HAWK v0.9.8-beta is theoretically the correct version of the tool to run, since it is more appropriate for bacterial GWAS, corrects for population structure, and is more recent than HAWK v0.8.3-beta, our qualitative comparison showed that the results obtained with HAWK v0.8.3-beta are far superior than with HAWK v0.9.8-beta. Moreover, the results we obtained with HAWK v0.9.8-beta required modifications of filters that we believed were the source of problems, which might not be the correct way to fish back the significant k-mers. A deeper analysis of why the k-mers are being filtered out in these datasets would be needed, but we thought that further debugging of HAWK v0.9.8-beta's code to make it correctly execute on our datasets was out of the scope of this work.

We briefly explained this decision in the Methods section "**k-mer-based GWAS**" (check the red text in lines 624-629):

> "We firstly ran HAWK [13] v0.9.8-beta, as it allows correcting for population structure. Unfortunately, it was unable to find the known causal variants reported for P. aeruginosa levofloxacin and M. tuberculosis streptomycin resistances by other methods (see Tables 2 and 3). We therefore kept in our benchmarks an earlier version, HAWK v0.8.3-beta, which presented better qualitative performance for these two evaluated panels."

2.11 A more appropriate comparison would be SEER and a SNP/gene based tool (perhaps bugwas since it's the tool underlying DBGWAS) as these are the methods most potential users will be using currently. I think it would be valuable to demonstrate that DBGWAS runs in a comparable time scale and produces more/clearer true positives than a GWAS of SNPs and genes. I think collapsing many gene hits into a single mobile element in the results would be very appealing to people, as would highlighting multiple SNPs in the same gene within a single graph.

We thank the reviewer for this suggestion. Our Methods section **'Resistome-based association studies'** features SNPs and genes tested using bugwas, and following the reviewer's suggestion we added a specific subsection entitled '**DBGWAS facilitates the interpretation of k-mer-based GWAS**' in the Results section, in order to better highlight that DBGWAS indeed collapses multiple SNPs in a single subgraph (check the red text in lines 223-274).

<u>*Choice of benchmarking conditions:*</u>

2.12 While performance of these methods on a single core is interesting, this seems like an unrealistic test scenario, since methods would be likely to be run in a parallel computing environment (or even using multiple cores on a laptop). Does this method remain competitive when the tools are given access to multiple cores? SEER and HAWK can be run in parallel to drastically reduce run time and it's not clear if DBGWAS can.

We now present two benchmarks where all methods use multiple (8) cores.

The first one is a new benchmark to assess the scalability of DBGWAS. We built several panels from three single-species large datasets with random phenotypes. We present in **(new) S9 Figure** the time and memory usage performance curves, which provides a better understanding of DBGWAS performance behaviour and estimations of the computational resources usage on small and large panels with different genome plasticities. For more details, please see our response to **item 1.4**.

The second one is an update of the previous benchmark comparing DBGWAS, pyseer and HAWK, presented in **(new) S2 Table**. We updated SEER to pyseer, added three large panels, detailed the computational resource usage of all steps of pyseer and HAWK, and ran all tools on 8 cores in this benchmark. We also kept DBGWAS performance on 1 core in **(new) S1 Table**, allowing the reader to check the gain in performance when working on multiple cores, by comparing these results to the ones in **(new) S2 Table**.

Note that the time and memory usage reported on 1 core differ from those reported in the previously submitted version because we ran all our benchmarks of **(new) S1 Table** and **(new) S2 Table** on a single unique machine, with 8 Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz cores, 315 GB of RAM, and 1 TB of disk space.

Installation and use:

> *2.13 I tried running the precompiled version of the software on our computing cluster, but it was not the right kind of machine, so set about compiling it from scratch. The Readme file attached to the software omits the -DCMAKE_BUILD_TYPE=Release flag in the cmake command, causing compilation to take an extremely long time (we had to kill the process eventually). Once I applied the instructions from the website, installation proceeded a small way then failed, and I haven't been able to decipher why yet based on the log file produced. I tried on a laptop and got another, different error (error: invalid argument '-std=c++11' not allowed with 'C/ObjC') which I haven't yet been able to resolve using suggestions I've found online. I asked around about other people who've tried the tool and found one person who'd succeeded in installing and running it after a lot of work and having to change the source code several times to get it to run. Ease of installation of the software needs to be improved if users are to adopt this method. Getting more reports from users attempting to install and the errors they receive will likely improve this though, so I don't think this should necessarily be a barrier to publication.*

We thank the reviewer for their efforts in installing DBGWAS in spite of these errors. The difficulties encountered during the installation and execution of the tool were also a concern of Reviewer 1. We answered all Reviewers' queries on installation and running issues (including this one) in the response of **item 1.39.** There, we provide a full update of what we have done to make the execution of the statically compiled binary as straightforward as possible, as well as the compilation of the source code.

> *2.14 I did have a look at the example results in the online browser tool. I noticed many of the hits formed compacted de Bruijn graphs which were less straightforward to interpret than the examples in the paper. I wasn't sure how the unitigs corresponded to the genome/plasmids. In these cases I would have needed to extract the sequences and map them to understand what the results meant. The collapsing of k-mers would be a benefit in this case, but I think the interpretability of these graphs relative to a k-mer analysis has been overblown a bit. In particular, if the authors could explain the graphs which don't form a linear or circular structure, but rather a complex network of unitigs that all appear to be quite interconnected, some of which are accompanied by a large number of protein annotations, this would be helpful. A more extensive set of examples and their interpretation laid out in a manual or a wiki on the GitLab page could address this.*

The interpretation of the less straightforward graphs is also one of our concerns. A work initiated on these more connected subgraphs allowed us to find some clues on why these graphs are so complex. We identified that such graphs with a low number of unitigs (http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/experiments/Paeru_Amikacin_visualisations/components/comp_3.html) often integrate a path of red (positively-associated) nodes lying in a non-coding region between variable accessory genes (here, ICE elements). They seem to be markers of some mobile genetic elements, however as these paths stand in a noncoding region between variable genes, their neighbor unitigs branch to various other unitigs, making the structure complex and hard to interpret.
For similar reasons, when genes involved in gene mobility, such as transposases or intergrases, are captured in a subgraph, the number of unitigs in the subgraph explodes (http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/experiments/Paeru_Amikacin_visualisations/components/comp_0.html). Indeed, these genes often exist in several copies in each genome and are found in variable genomic environments. When represented at a bacterial population scale, this generate loops and multiple branches, making such subgraphs very complex to interpret.
Complex graphs also arise when several variants are aggregated because they are close to each other in at least one strain of the studied panel, and their neighborhood (as defined with DBGWAS -nh parameter) overlap (this is the case of AAC6' acetyltransferase and CML efflux pump in http://pbil.univ-lyon1.fr/datasets/DBGWAS_support/experiments/Paeru_Amikacin_visualisations/components/comp_6.html). In this case, the graph can be simplified by breaking some edges and splitting the subgraph into several subgraphs. This can be done by modifying the parameters of DBGWAS third step (-nh and -SFF). For now, this step indeed requires trial and error to fine-tune the choice of -SFF and -nh values. We recommend to re-run the third step with lower -nh and/or -SFF values (new Methods section **'Re-running from step 2 or step 3'** - check the red text in lines 523-528), and also in Methods section '**Graph neighborhoods**'.
We now acknowledge the fact that some subgraphs have no straightforward interpretation using the examples above in the '**DBGWAS facilitates the interpretation of k-mer-based GWAS**' section (check the red text in lines 255-274).

The automated detection of SNP, MGE or of more complex graph - and the automation of a -nh/-SFF parameter tuning is however not trivial, it is a project in itself. We produced a beta version of a prediction tool able tell apart SNPs from MGEs. To build this tool, we trained predictive models over real and simulated data, producing subgraphs which we knew corresponded to SNPs or MGEs. We will not be able to include it in the next release of DBGWAS, but it will be described in Magali Jaillard's PhD manuscript and added to a future release as soon as possible.

> *Final comments:*
> *I really like the idea of this tool, and if I manage to get it installed I will use it and get a better feel for the interpretation of these graphs, however in its current form I'm not able to get it running. I do hope this tool is published, however a better comparison of the tool to other methods is required.*

Reviewer 3:

Overall I enjoyed the paper as it is a neat idea for tackling the gap between kmer and SNP approaches. It is quite a dense manuscript in terms of content – the authors have clearly done a lot of impressive work- and their method is clearly successfully working. I have a few comments however:

> *3.1 The paper is fairly statistics heavy, and I think it would benefit from a longer more drawn out explanation of the approach in the introduction. I found it a bit challenging to understand on the first read.*

Thank you for highlighting this difficulty. We moved a brief explanation of the pipeline, together with the pipeline overview figure (**new Fig 2**), to the beginning of the **Results** section, which we hope will help the readers (check the red text in lines 64-75):

> "We developed DBGWAS, available at https://gitlab.com/leoisl/dbgwas, and validated it on panels for several bacterial species for which genome sequences and antibiotic resistance phenotypes were available. DBGWAS comprises three main steps: it first builds a variant matrix, where each variant is a pattern of presence/absence of unitigs in each genome. Each variant is then tested for association with the phenotype using a linear mixed model, adjusting for the population structure. Finally, it uses the cDBG neighbourhood of significantly associated unitigs as a proxy for their genomic environment. DBGWAS outputs a set of such subgraphs ordered by minq, which is the smallest q-value observed over unitigs in each subgraph. The top subgraphs therefore represent the genomic environment of the unitigs most significantly associated with the tested phenotype. Fig 2 summarises the main steps of the process. A detailed description of the pipeline is presented in the Methods section."

> *3.2 I think there needs to be more discussion of multiple testing and how significance is defined. In my mind the problem of large numbers of correlated kmers is one of the big problems with the kmer approach, and so any sense of whether the approach described here can make advances would be appreciated. As it stands however I feel the definition of 'significant' associations is arbitrary to the point of being hard to judge how well the results would stand up in a less extreme phenotype than antibiotic resistance (where the hits are very clear)*

We acknowledge that the definition of the significance was not enough explicit in the document, and added an additional sentence for the readers, in Method section '**Significance threshold**' (check the red text in lines 447-452):

> "The interpretation step focuses on the unitigs with the lowest q-values. These unitigs are indeed used to build the resulting annotated subgraphs. The unitig selection can be either based on the FDR (q-value thresold) or on a number of presence/absence patterns ordered by increasing q-values. Practically, this is done in DBGWAS using a Significant Features Filter (SFF). For a selection based on a FDR threshold, the SFF value is set between 0 and 1, while any integer value > 1 defines the number of patterns to consider."

To answer your remark about multiple testing, we take the test multiplicity into account by returning q-values rather than uncorrected p-values. Admittedly, unitig presence/absence profiles can be highly correlated. The correlation is typically positive: even if two k-mers are mutually exclusive, their presence profiles are the same since all profiles are converted to their minor allele version before they are tested. Under such positive dependency, the Benjamini-Hochberg procedure that we use still controls the FDR as the q-value it provides is an upper bound on the true FDR. The major problem in this case is that the variance of the FDP around the FDR increases with the level of dependency, and the FDR itself becomes less informative. Unfortunately, to the best of our knowledge there is no better alternative. We discussed this point with multiple testing experts and would like to find better alternatives in the future but we are not making any advance to this problem in this paper.

The use of BH method was made more clear in the Methods, '**Testing unitigs for association with the phenotype (step 2)**' (check the red text in lines 434-437):
"To tackle the situation of multiple testing caused by the high number of tested patterns, we compute q-values, which are the Benjamini-Hochberg transformed p-values controlling for false discovery rate (FDR) [60]."
and we discuss about possible improvements the **Discussion** (check the red text in lines 322-325):

> "DBGWAS currently relies on the Benjamini-Hochberg procedure to control the FDR and offers no advance exploiting the dependence among presence/absence patterns. An important improvement would be to control the false discovery rate at the subgraph level instead of the unitig level."

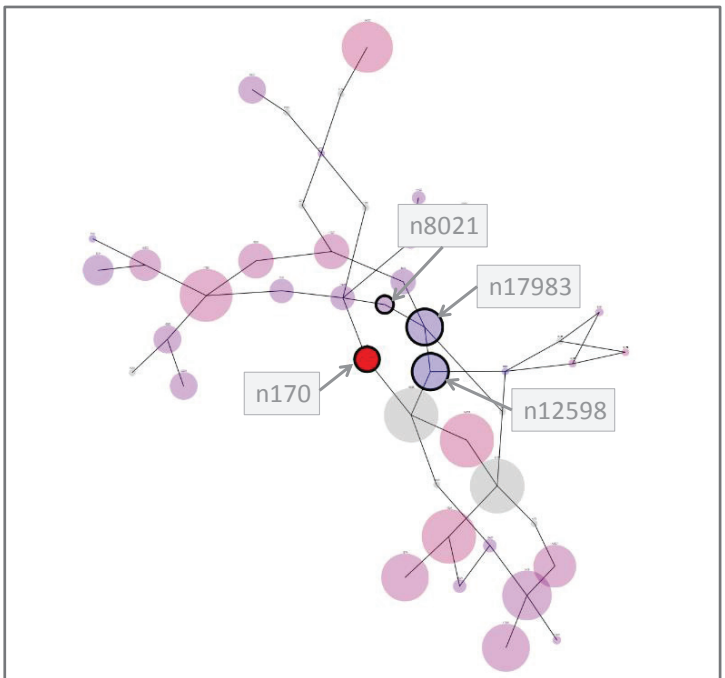
> *3.3 On a similar note the novel associations are very hard to judge as there is no way to tell if they are truly causal (a pro of the method) or just random noise (a con). I think some work should be done here to at least try to dig into this further – even if purely just splitting samples into replication and discovery samples and looking at how well novel associations in the discovery sample replicate (rather than wet lab work).*

We agree that not being able to tell apart truly causal and spurious associations among novel associations is frustrating. We believe however that solving this problem would necessarily require wet lab experiments. Unfortunately, splitting samples into replication and discovery would only say something about the robustness of the statistical association between variants and phenotypes, not their being causal. A SNP which is strongly associated with resistance by LD for example could be validated by this approach, whereas a truly causal SNP with weaker association could fail.

In the absence of wet lab experiments, we chose to (i) assess how well DBGWAS would recover known determinants and (ii) describe novel associations and, when possible, how likely they were to be truly causal. Some of them (e.g. L650M amino-acid change for *P. aeruginosa* resistance to levofloxacin) evoke possible resistance mechanisms. Others (see **(new) Table 1**) are likely to be caused by a selection bias in the design of the panels (orange 'CR' (for 'co-resistance') in the last column) or LD (high r^2 index in the last column). Finally, we also assessed the ability of DBGWAS to tell apart truly causal and spurious associations using simulated data, where we knew the true causal elements. The results were previously presented in a separate bioRxiv pre-print (https://www.biorxiv.org/content/early/2017/03/03/113563) and are now included as an appendix to this manuscript (**new S1 Appendix**).

> *3.4 The results are compared to SEER, but I believe pyseer is probably a better comparison for speed.*

We thank the reviewer for pointing out pyseer, the updated version of SEER. We fully replaced SEER by pyseer in this work (including the text and the benchmarks). As this was also a concern of Reviewer 1, we provide more details in **item 1.12** on what changed in the manuscript and in the benchmarks by replacing SEER by pyseer.

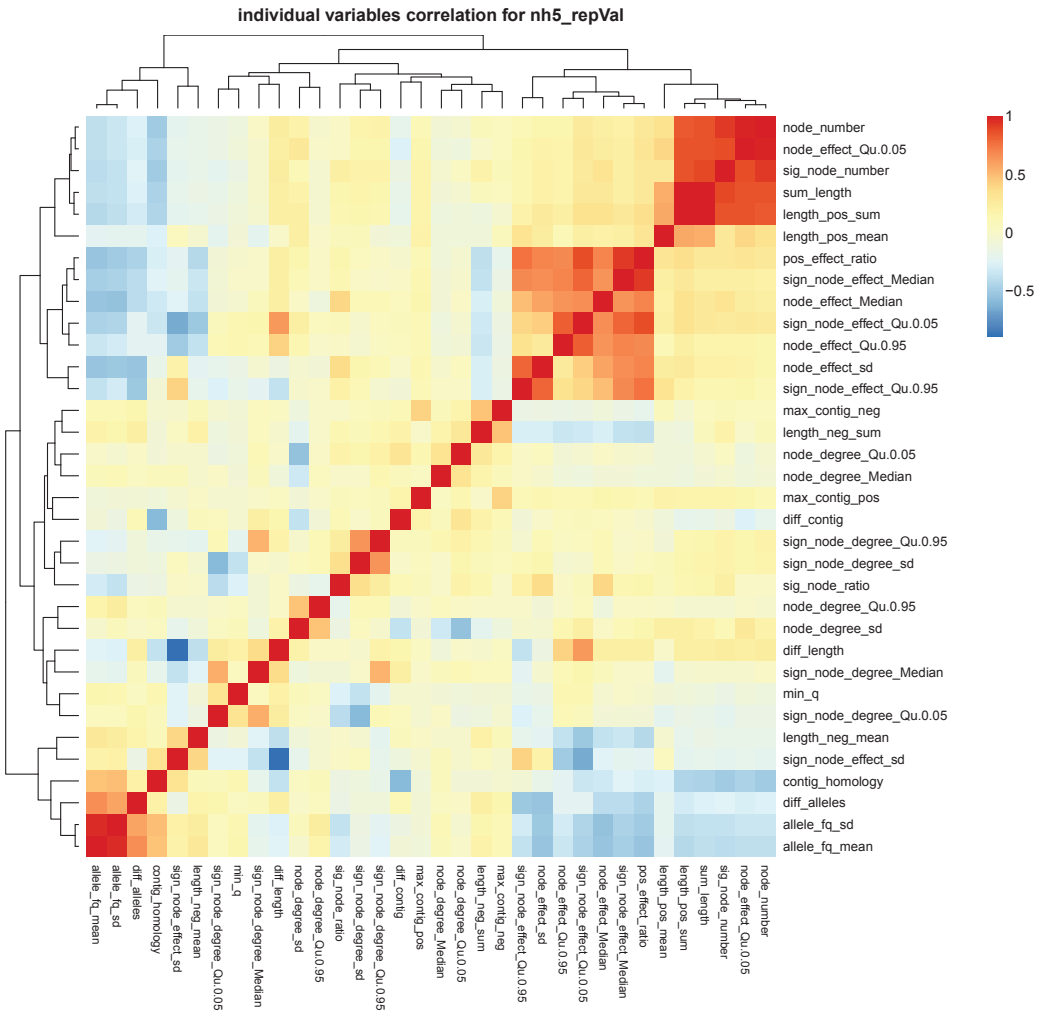


Node ID	Allele feq	Pheno0	Pheno1	Seq Length
n170	106	10/163	96/119	663
n12598	174	151/163	23/119	36
n17983	172	149/163	23/119	36
n8021	63	53/163	10/119	45

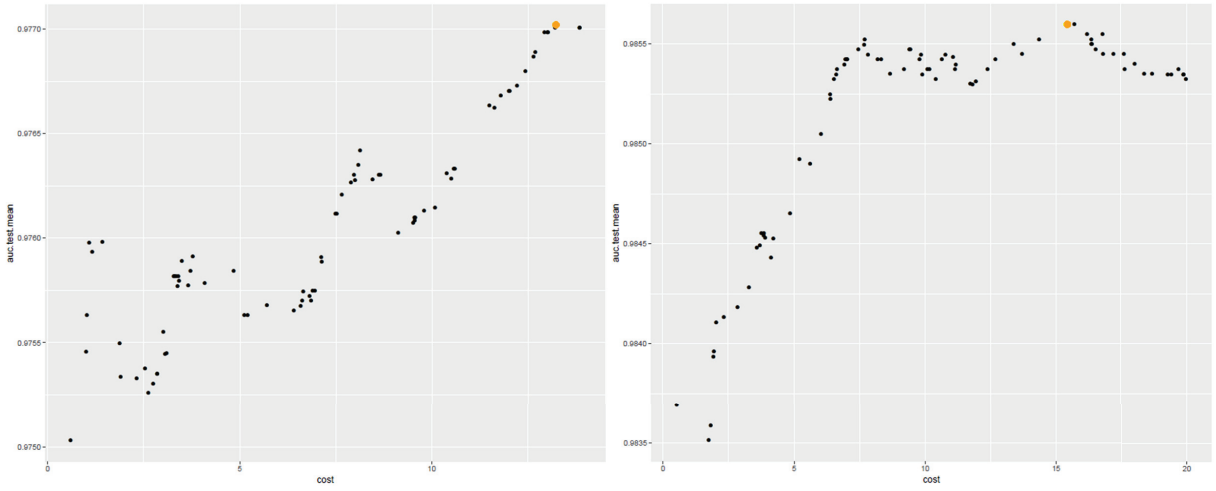
Supplementary Figure S4.1: **Example of a gene insertion which might be (wrongly) taken for a SNP.** This graph, obtained by simulation, shows indeed a single red node in a branch which is opposed to a branch with three blue nodes. However when looking at the sequence length, the red node represents the complete 663 pb of AAC(6')-Ib4 aminoglycoside acetyltransferase gene, while the three blue nodes only represent 41 pb once the two overlapped sequence of length $k-1$ pb are removed.

Table S4.6: Hyperparameter search space

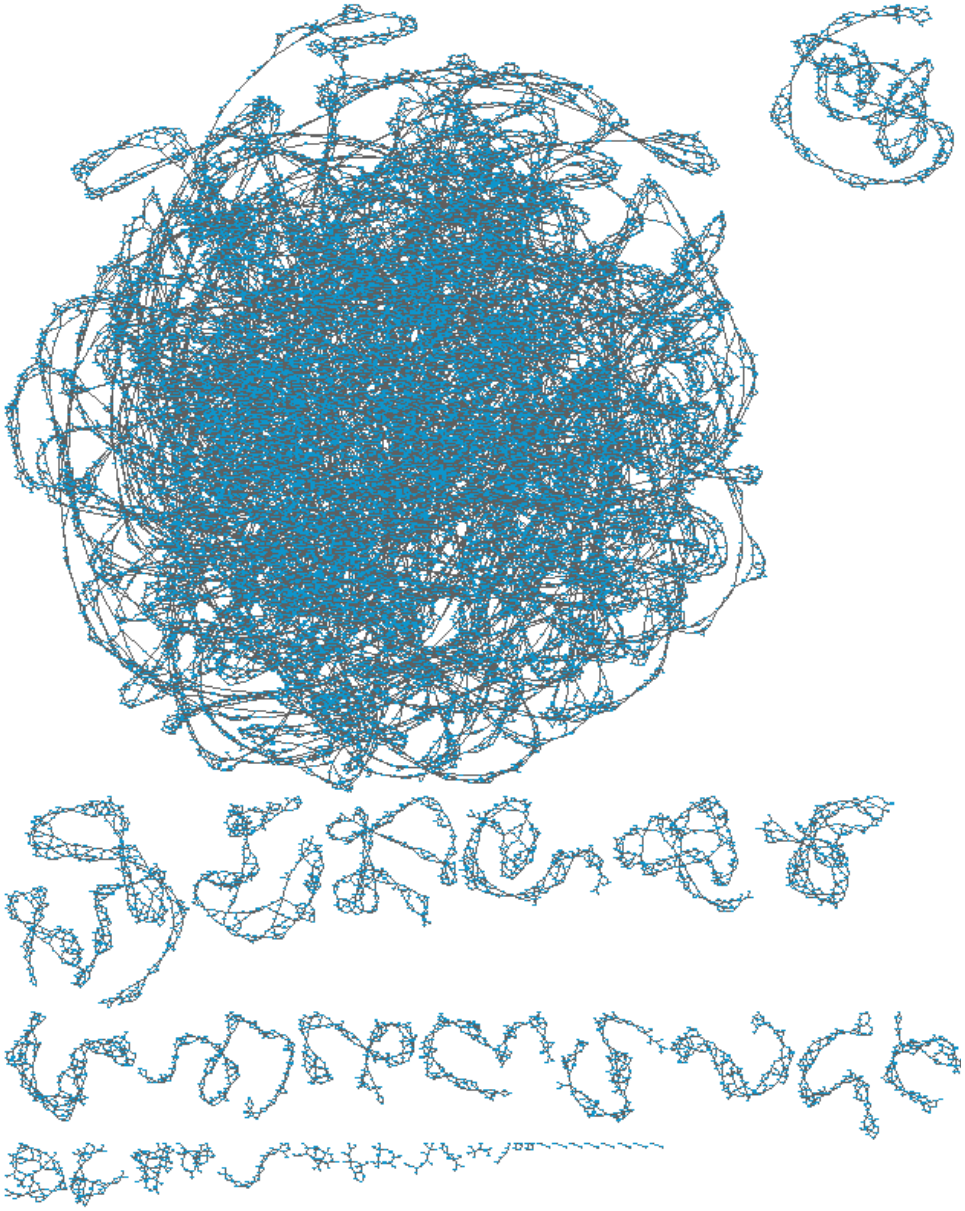
	Hyperparameter	Tested values
lasso	λ	$[1 \times 10^{-4}; 0.4]$
ridge	λ	$[1 \times 10^{-2}; 370]$
elastic-net	λ	$[1 \times 10^{-4}; 0.4]$
	α	$]0; 1[$
linear SVM	C	$[0.5; 40]$
radial SVM	C	$[0.5; 40]$
	γ	$[0; 0.3]$
random forest	m	$[2; 20]$
	node size	$[1; 25]$



Supplementary Figure S4.2: **Correlation matrix of the covariates.** This matrix represents the correlation between the covariates used to build the prediction models. This matrix was obtained for `nh` = 5, with the 'repVal' NA method.



Supplementary Figure S4.3: **Example of model hyperparameters tuning.** Panels A and B present the figures generated to validate the hyperparameter search spaces: if the selected value reaches a border of the space (Panel A), we increase it until it falls in a maxima (Panel B).



Supplementary Figure S4.4: **cDBG of a genome simulated with the gene-base strategy.** The big connected component represents the core-genome while all other connected components represent the inserted genes, among them really small graphs, with only 1 to 3 nodes.