

Résistance aux antibiotiques

Prédiction par apprentissage statistique

Vadim BERTRAND, Cheikh-Darou BEYE

9 janvier 2023

Sommaire

1	L'antibiorésistance	2
2	Exploration des données	2
3	Approche pour la prédiction	3
3.1	Pré-traitements	3
3.2	Réduction de la dimensionnalité	4
3.2.1	ACP (à Noyau)	4
3.2.2	Stability selection	5
3.2.3	Tests multiples	6
3.3	Classifieurs considérés	6
3.4	Mise en oeuvre	6
4	Résultats	6
5	Perspectives	6

1 L'antibiorésistance

Les antibiotiques sont développés pour contrer les infections dues aux bactéries. Certaines bactéries peuvent acquérir une résistance à des antibiotiques, via l'obtention de nouveaux gènes ou par la mutation de gènes existants. L'antibiorésistance représente un grand risque pour la santé publique, il est donc important de la limiter. Cela passe notamment par une meilleure compréhension des mécanismes de résistances comme l'identification de gènes résistants ou des bactéries résistantes. Cette résistance de certaines bactéries à des antibiotiques peut être traitée comme une tâche de classification en apprentissage statistique.

Dans cette étude nous aurons à notre disposition un jeu de données constitué de 3 matrices de régresseurs et une matrice réponse pour 414 bactéries :

- X_gpa , codant la présence ou l'absence de 16005 gènes,
- X_nsp , codant la présence ou l'absence de 72236 mutations génétiques,
- X_genexp , représentant l'expression génétique de 6026 gènes ;
- Y , codant la résistance ou la sensibilité à 5 antibiotiques : la Ceftazidime, la Ciprofloxacine, la Colistine, le Méropénème et la Tobramycine.

Notre objectif est de prédire la résistance des bactéries aux antibiotiques à partir des régresseurs et d'identifier quelles matrices de régresseurs sont les plus intéressantes pour cette tâche, selon l'antibiotique considéré.

Dans un premier temps, nous procéderons à une courte exploration des données. Puis, nous détaillerons notre démarche : pré-traitements utilisés sur les données, proposition d'approches de réduction de dimension, classifieurs considérés et mise en œuvre via la librairie *scikit-learn*. Enfin, nous présenterons les résultats obtenus et nous proposerons quelques pistes d'amélioration.

2 Exploration des données

Avant de nous lancer dans la prédiction de la résistance aux antibiotiques, nous avons souhaité nous pencher sur les données que nous manipulons.

Naturellement nous avons commencé par observer les types de données que nous manipulons et l'éventuelle présence de données manquantes. Sur les 4 matrices à notre disposition, 3 contiennent des données binaires (Y , X_gpa , X_nsp) tandis que X_genexp contient des données quantitatives.

Comme le montre la table 1, les matrices de régresseurs ne contiennent pas de données manquantes, mais certaines informations de résistance aux antibiotiques sont manquantes, notamment pour la Ceftazidim avec 20% de données absentes. Etant donné que la taille du jeu de données est réduite, que nous procéderons par la suite à une validation croisée et que nous ne disposerons donc pas d'un jeu de test, nous avons choisi de ne pas imputer les données manquantes afin d'éviter de fausser la généralisation des résultats. Par conséquent, les bactéries dont la résistance à un antibiotique est manquante ne seront pas utilisées lors de l'évaluation des classifieurs sur l'antibiotique correspondant.

Nous pouvons également observer sur la table 1 que les variables réponses ne sont pas toujours équilibrées : 2 fois plus de bactéries résistantes à la Méropénem, et à l'inverse 2 à 3 fois plus de bactéries susceptibles à la Tobramycin et la Colistin. De même, les gènes ou les mutations sont bien plus souvent absentes que présentes.

Table 1: Résumé des variables réponses et des régresseurs.

	Résistance					Présence		
	Tobramycin	Ceftazidim	Ciprofloxacine	Meropenem	Colistin	<i>gpa</i>	<i>snps</i>	<i>genexp</i>
# NA	8	80	56	60	0	0	0	0
# VRAI	130	165	199	244	85	3581	8218	NaN
# FAUX	276	169	159	110	329	12424	64018	NaN

Pour aller un peu plus loin, nous avons représenté nos données regroupées par clustering hiérarchique avec des cartes de chaleur afin de faire apparaître des structures. La figure 1 correspondant à la carte de chaleur ainsi obtenue pour la matrice X_gpa permet par exemple de supposer que cette matrice porte de l'information intéressante pour prédire la résistance à la Tobramycin et la Ciprofloxacine, mais probablement moins pour la Colistin. Pour celle-ci, nous avons observé par le même biais que la matrice X_genexp sera sûrement indispensable.

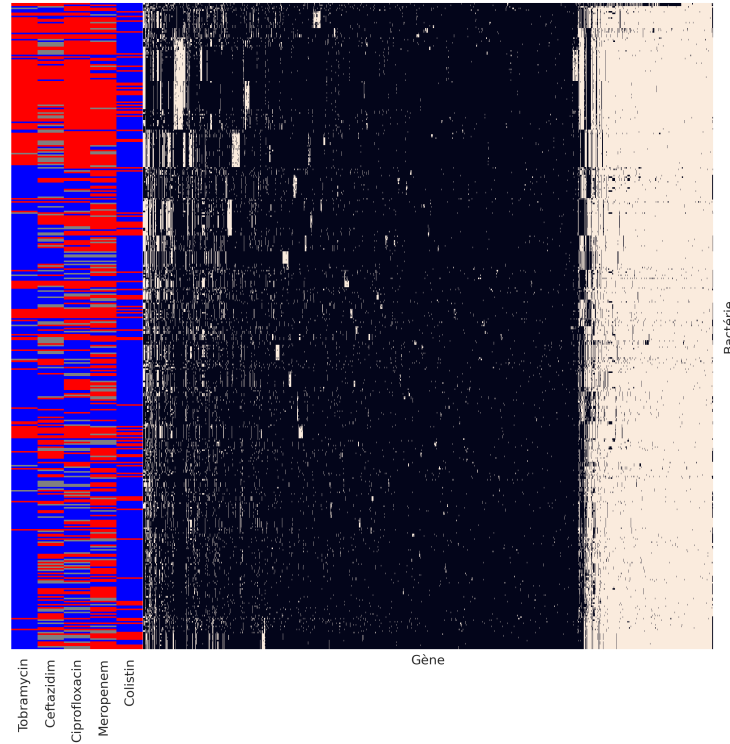


Figure 1: Carte de chaleur du clustering hiérarchique sur les lignes et les colonnes de la matrice X_gpa . Les couleurs à gauche des lignes permettent de déterminer si la bactérie est sensible (bleu) ou résistante (rouge) à l'antibiotique, le gris correspond aux données manquantes.

3 Approche pour la prédiction

3.1 Pré-traitements

Comme expliqué dans la section §2, nous avons fait le choix de supprimer les données manquantes. Cette suppression est faite de manière “intelligente” en ce sens où les bactéries dont la résistance est

absente sont éliminées uniquement pour les antibiotiques concernés et demeurent disponible pour les autres antibiotiques.

Nous nous sommes ensuite contentés de centrer/réduire les expressions génétiques de la matrice X_{genexp} grâce au transformateur **StandardScaler** de *scikit-learn*.

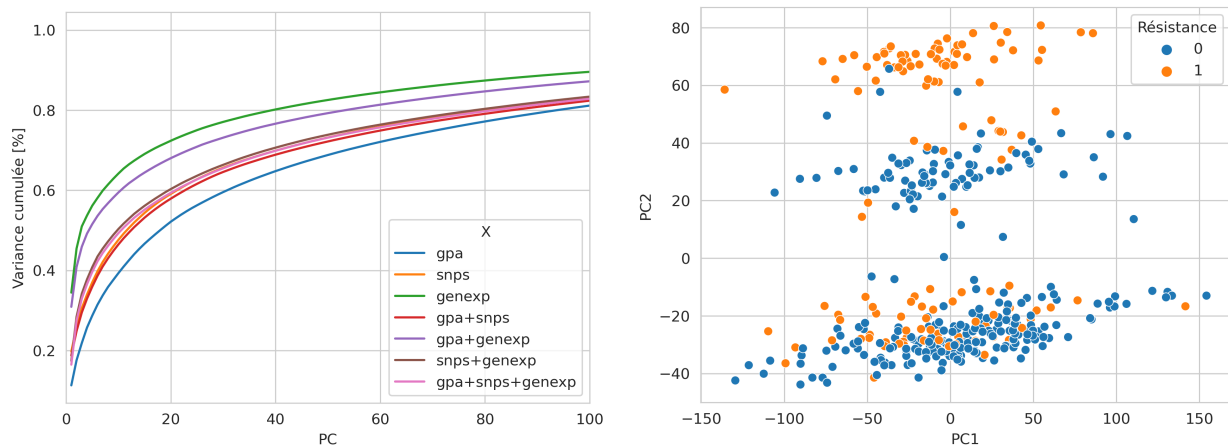
S'est ensuite posée la question de la mutualisation des informations contenues dans les 3 matrices de classifieurs. En traçant les cartes de chaleur de ces matrices, nous avons aperçu que toutes ne sont pas nécessairement pertinentes pour tous les antibiotiques. C'est pourquoi nous avons fait le choix de ne pas systématiquement agréger les matrices, mais de considérer les 7 arrangements possibles : 1 seule matrice (3), 2 matrices (3) et les 3 matrices (1).

3.2 Réduction de la dimensionnalité

Nous avons vu que nos matrices de régresseurs contiennent beaucoup de covariables, jusqu'à 94267 lorsque nous les concaténons entres-elles, relativement aux nombres d'observations dont nous disposons. Il est donc impératif de réduire cette dimensionnalité. Cela peut se faire en amont de la tâche de classification, ou alors de manière intégrée en incluant une pénalisation sur les poids du modèle associés aux regresseurs. Nous avons considéré les deux approches et nous détaillerons dans cette partie les 3 méthodes que nous avons employées en amont.

3.2.1 ACP (à Noyau)

L'Analyse en Composantes Principales (ACP) est une approche bien connue permettant de représenter les observations dans un sous-espace vectoriel dont les composantes sont décorrélées. Les composantes sont ordonnées en ordre décroissant de variance expliquée, de sorte qu'il est simple d'utiliser le nombre de composantes conservées pour représenter les données comme un hyper-paramètre permettant de plus ou moins réduire la dimensionnalité tout en maximisant le pourcentage de variance expliquée pour un nombre de composantes fixé.



(a) Variance cumulée exprimée en pourcentage. (b) Nuage de points selon les 2 premières dimensions.

Figure 2: Représentations de l'ACP avec un noyau linéaire.

Nous avons appliqué l'ACP à notre jeu de données, en considérant tous les arrangements possibles d'utilisation des matrices de descripteurs et comme nous pouvons le voir sur la figure 2, les résultats sont mitigés. D'un côté, la figure 2a ne montre pas de point d'inflexion franc dans la courbe de variance cumulée, ce qui suggère qu'il n'existe pas de point au delà duquel les composantes

sont peu importantes. En revanche, nous pouvons voir sur la figure 2b que la projection dans les deux premières dimensions de l’ACP (en ayant agrégé l’ensemble des matrices de régresseurs) de nos données permet de partiellement séparer les bactéries sensibles et résistantes, ce qui est encourageant en vue de l’étape de classification.

Par ailleurs, grâce au *kernel-trick* il est possible de mettre en place une variante de l’ACP permettant une réduction de dimensions non linéaire en utilisant divers noyaux. Via *scikit-learn*, il suffit alors d’utiliser la classe **KernelPCA**.

3.2.2 Stability selection

La *stability selection* est une approche permettant de corriger l’instabilité de la sélection de variables de l’estimateur Lasso en présence de covariables corrélées. Pour ce faire, des estimateurs pénalisés avec la norme $L1$ sont ajustés sur de multiples ré-échantillonnages du jeu de données. Chaque estimateur produit un chemin de régularisation le long duquel les poids associés aux covariables les moins importantes pour la tâche de régression ou de classification sont progressivement annulés quand la pénalisation augmente. À partir de ces chemins de régularisation, l’estimateur de *stability selection* construit un chemin de stabilité représentant la fréquence de sélection des covariables parmi l’ensemble des estimateurs Lasso. Au moyen d’un seuil de stabilité, il est donc possible de sélectionner les variables alors considérées comme *stables*. Cette procédure est généralement utilisée comme un estimateur plutôt que comme une méthode de réduction de dimensions, mais nous avons souhaité essayer de l’utiliser comme telle.

La *stability selection* n’est pas proposée par *scikit-learn* (uniquement le *bootstrap Lasso*), nous avons donc implémenté notre propre estimateur héritant des classes de bases **TransformerMixin** et **BaseEstimator** afin de pouvoir l’utiliser ensuite dans un **Pipeline**. La figure 3 illustre le chemin de régularité obtenu en utilisant l’ensemble des matrices de régresseurs pour construire la matrice de covariables et avec un seuil de régularité de 0.6. Un tel seuil conduit à conserver environ 1250 covariables sur plus de 96000 au départ.

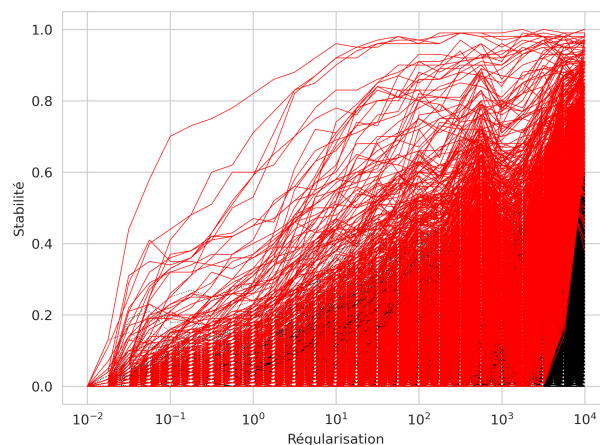


Figure 3: Illustration du chemin de stabilité obtenu par *stability selection*.

L’un des inconvénients principal de la *stability selection* est le temps d’ajustement très important en raison du nombre de ré-échantillonnages à effectuer (100 dans notre cas) et de la longueur du chemin de régularisation (fixé à 25). Afin de palier ce problème, nous avons effectué les sélections de variables en amont pour chacun des arrangements et des antibiotiques plutôt que de répéter les sélections pendant l’ajustement des hyper-paramètres. Cela a pour effet de rendre les scores de val-

idation croisée obtenus pendant la procédure de recherche des hyper-paramètres non généralisables, car la sélection de variables est faite sur l’ensemble du jeu de données et pas sur les sous-ensembles de validation croisée. Une fois le modèle le plus performant déterminé, nous avons donc répété l’évaluation par validation-croisée sans fixer au préalable les variables stables pour obtenir des scores extrapolables à de nouvelles données.

3.2.3 Tests multiples

Une autre manière de réaliser la réduction de dimensionnalité pourrait être de considérer que de nombreuses variables sont inutiles pour la prédiction, car leurs distributions entre les bactéries susceptibles ou résistantes est la même.

Nous pouvons donc procéder à une nouvelle sélection de variables s’appuyant cette fois-ci sur deux tests statistiques : le Z-test pour les matrices X_gpa et X_snps et le T-test pour la matrice X_genexp . Nous avons utilisé la librairie *statsmodels* qui propose l’implémentation de ces tests. En raison du grand nombre de tests réalisés de manière indépendante, il convient d’ajuster les *p*valeurs. Nous avons choisi d’utiliser la procédure de Benjamini-Hochberg permettant de contrôler le FDR (*False Discovery Rate*), plutôt que des procédures contrôlant le FWER (*Family-Wise Error Rate*) afin de découvrir plus de variables statistiquement différentes selon les deux modalités de résistance. Nous nous sommes tourné pour cela vers la librairie *MultiPy* qui permet de choisir parmi de nombreuses méthodes d’ajustement.

Comme le montre la table 2, cette approche permet de réduire très fortement le nombre de covariables utilisées par le classifieur, et le niveau α peut-être employé comme un hyper-paramètre permettant de jouer sur le nombre de variables rejetées.

Table 2: Nombre de covariables sélectionnées pour différents niveaux α .

	# régresseurs	Niveau α		
	total	0.1	0.05	0.01
gpa	16005	575	374	157
snps	72236	5258	3053	1233
genexp	6026	96	77	56

Le désavantage que nous voyons à cette approche par tests multiples est que les variables sélectionnées sont potentiellement corrélées.

3.3 Classifieurs considérés

3.4 Mise en oeuvre

4 Résultats

5 Perspectives