

Apprentissage Statistique en Génomique

Dell'Aiera Clément, Prévosteau Clément

Résumé

Dans ce document, nous présentons une application des méthodes apprises au cours donné par Jean-Philippe Vert intitulé *Machine Learning for Computational Statistics*. Il s'agit d'extraire un profil génétique à partir des niveaux d'expression de 4654 gènes récoltés sur 184 individus qui ont ou n'ont pas rechuté après une chimiothérapie, profil dont le but est de savoir, étant donné les niveaux d'expression de ces gènes chez un patient, s'il risque de rechuter ou non, en vue d'adapter le traitement *a priori*.

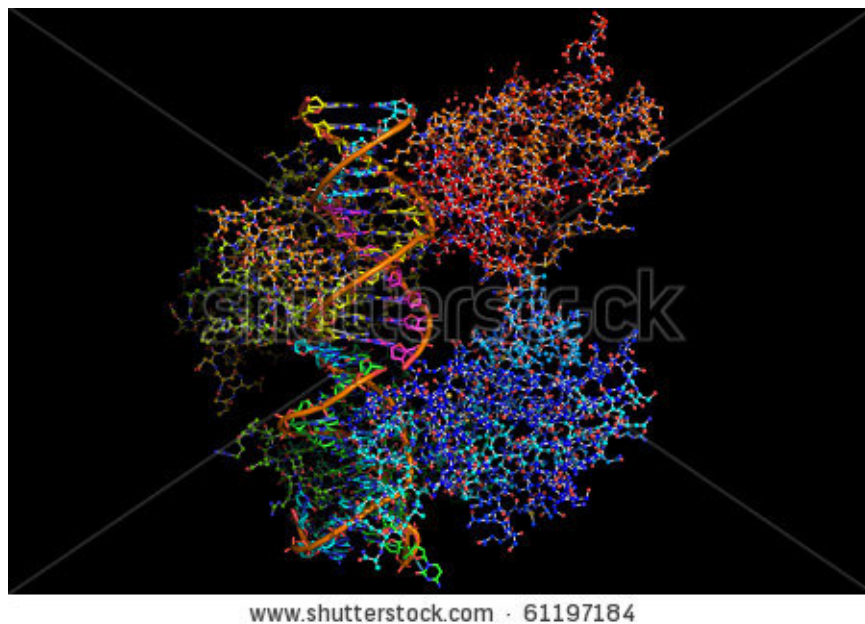


FIGURE 1 – A ball and stick model of molecules of protein p53 binding to a strand of DNA. The majority of human cancers involve mutations which make this protein inactive, from http://fr.123rf.com/photo_7823791_.html

Table des matières

1	Support Vector Machine	3
1.1	Commentaires sur le code	4

1 Support Vector Machine

Nous avons entraîné des SVM sur la base d'apprentissage des 184 patients initiaux, en faisant varier différents paramètres qu'offrait le package *kernelab* sous *R*. Par exemple, nous avons testé les noyaux : linéaires, gaussiens, Anova, TanH, Bessel et Laplace. A chaque fois, la marge a été déterminée par *cross-validation* sur les 184 patients, en divisant l'échantillon en 5 classes.

Voici les marges optimales obtenues sur les 3 noyaux les plus efficaces, à savoir linéaires, gaussiens et Anova :

Noyau	Marge
Linéaire	0.0009765625
Gaussien	0.125
Anova	

TABLE 1 – Marges optimales des SVM obtenues par *cross-validation*

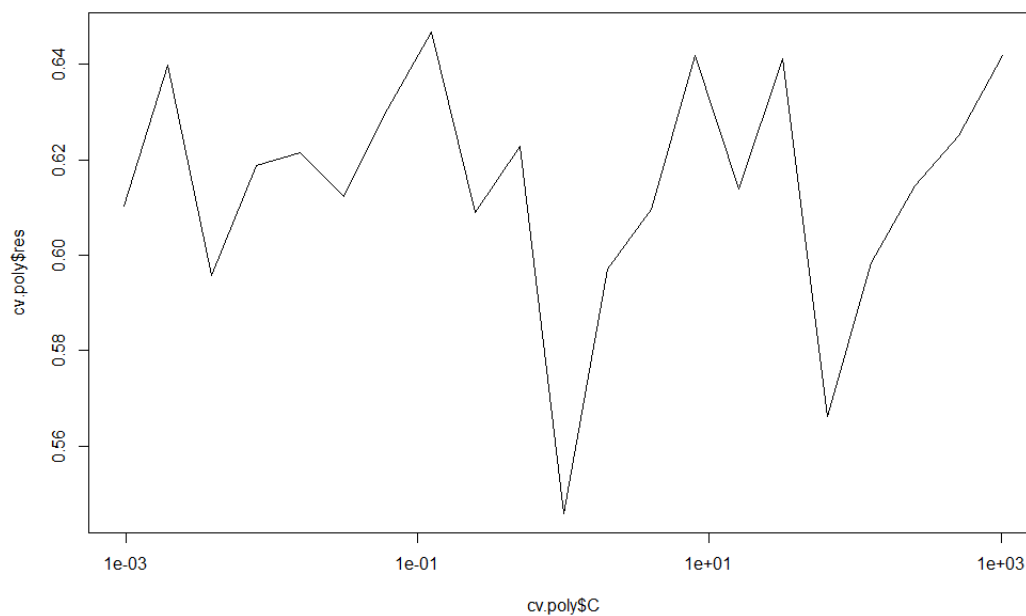


FIGURE 2 – Courbe de l'AUC pour un noyau polynomial de degré 2

L'échantillonnage en 5 sous-classes étant aléatoire, la marge C obtenue varie à chaque itération. Afin d'obtenir un résultat plus robuste, nous avons itéré la

cross-validation 100 fois, puis fais la moyenne des marges optimales obtenues. Sur 20 itérations de cross-validation, nous avons obtenu une marge optimale moyenne de :

$$CC = 87.45796$$

Voici le vecteur prédit à partir des 92 individus du dossier *xtest.txt* :

- | | | | | | | | | | | | | | | | | | | | | | | | |
|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1. [1] | -1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | 1 |
| 2. [26] | -1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 |
| 3. [51] | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 |
| 4. [76] | -1 | -1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |

1.1 Commentaires sur le code

Nous avons principalement utilisé le package *kernelab*. Le code du projet est entièrement disponible à la page :

<https://github.com/cdellaie/MachineLearningforComputationalStat>