Analyses de survie sur données transcriptomiques

Delphine Bonnetier

Rapport de stage

Master Mathématique spécialité : Statistiques et Applications

Analyses de survie sur données transcriptomiques

Stage réalisé à TRANSGENE S.A Illkirch-Graffenstaden
Sous la direction de Bérangère BASTIEN

Mémoire présenté par **Delphine BONNETIER**

FEVRIER - AOUT 2010

Remerciements

Je remercie monsieur Philippe Archinard, directeur général de la société Transgene pour m'avoir accueilli au sein de son établissement.

Je remercie Bérangère Bastien, biostatisticienne et tutrice de stage, qui a su me laisser la liberté nécessaire à l'accomplissement de mon travail tout en y gardant un œil critique et avisé. Je lui adresse mes plus vifs remerciements pour son encadrement, sa disponibilité et ses précieux conseils qui m'ont permis de mettre en pratique les connaissances acquises lors de ma formation, cultivant ainsi mon intérêt pour le domaine de la biostatistique, ainsi que pour la patience et la confiance qu'elle m'a accordées tout au long de ce stage.

J'associe à mes remerciement toute l'équipe « Biomarqueurs » composée de : Philippe Ancian (responsable du département « Immunologie Moléculaire » et de l'équipe « Biomarqueurs ») Bérangère Bastien, Benoit Grellier (bioinformaticien), Christophe Zemmour (biostatisticien) qui m'ont bien intégrée dans leur équipe et prodigué des conseils judicieux.

Enfin, un grand merci au reste du personnel de Transgene pour leur enthousiasme communicatif me permettant ainsi d'évoluer dans un climat de bonne humeur.

SOMMAIRE

Chapitre 1: Introduction ____ *4*

Chapitre 2: Données transcriptomiques ____ *6*

2.1 La molécule d'ADN ____ **6**

2.2 La puce à ADN ____ **7**

Chapitre 3: Analyse de survie et modèle de Cox ____ *11*

3.1 L'analyse de survie ____ **11**

3.2 L'estimation de la survie : méthode de Kaplan-Meier ____ **12**

3.3 Risque instantané de décès ____ **14**

3.4 Le modèle de Cox ____ **15**

3.4.1 Représentation du modèle de Cox et fonctions liées ____ 15
3.4.2 Méthode du maximum de la vraisemblance ____ 16
3.4.3 Hypothèses du modèle de Cox ____ 17

Chapitre 4: Réduction du nombre de variables ____ *20*

4.1 Description des différentes méthodes ____ **20**

4.2 Les méthodes les plus pertinentes ____ **23**

4.3 Sélection du nombre de composantes ____ **25**

4.3.1 Regression PLS ____ 25
4.3.2 Méthode LARS ____ 26

Chapitre 5: Capacité prédictive du modèle ____ *27*

5.1 Les courbes ROC ____ **27**

5.2 L'aire sous la courbe (AUC) ____ **28**

Chapitre 6: Application aux données cliniques ____ *30*

6.1 Travail sous R: ____ **30**

6.1.1 Présentation de R ____ 30
6.1.2 Descriptif des fonctions utilisées ____ 31

6.2 Jeux de données publiques ____ **32**

6.2.1 Analyse de survie : ____ 32
6.2.2 Analyse transcriptomique ____ 33
6.2.3 Résultats: ____ 35

Chapitre 7: Conclusion ____ *38*

Bibliographie ____ *39*

Chapitre 1: Introduction

Basée à Illkirch-Graffenstaden, Transgene est une société biopharmaceutique qui conçoit et développe des vaccins thérapeutiques et des produits de biothérapie pour le traitement des cancers et des maladies infectieuses chroniques telles que l'hépatite B ou l'hépatite C.

Cette société a choisi depuis quelques années de consacrer une partie de sa recherche à l'étude des biomarqueurs. L'objectif de telles études est de découvrir des indicateurs biologiques capables de révéler l'état physiologique ou biologique d'un individu pour pouvoir à terme fournir à chaque patient une thérapie adaptée à son profil. Parmi la grande diversité de biomarqueurs étudiables (cellules, enzymes, molécules, paramètres physiologiques...), le transcriptome (ensemble des ARN messagers d'une cellule) est un de ceux qui suscite de nos jours le plus d'intérêt car les méthodes expérimentales sont très robustes et permettent la compréhension globale des mécanismes biologiques.

C'est dans cette optique que Transgene a décidé de mettre en place en 2006 une plateforme de puces à ADN de type Affymetrix. Son but est d'étudier l'ensemble des ARN messagers des gènes, c'est à dire le transcriptome. La technologie Affymetrix permet de visualiser simultanément le niveau d'expression de la quasi-totalité des gènes d'un tissu d'un individu à un temps donné. Chez l'homme, on considère qu'il existe environ 39 000 gènes qui s'expriment de manière différentielle dans plus de 200 types cellulaires.

Les données ainsi recueillies peuvent alors faire l'objet d'un grand nombre d'analyses statistiques (analyse différentielle, régression linéaire, classification,...), mais l'abondance de l'information tend à les rendre relativement complexes. Ceci est notamment vrai dans le cas de l'analyse de survie qui est un cas particulier de la régression linéaire multiple dont une des principales hypothèses est que le nombre d'individus est supérieur au nombre de variables par individu. Or, dans les études s'appuyant sur les puces à ADN, la situation est très largement inversée puisque les quelques dizaines de milliers de variables que représentent les valeurs d'expression des gènes ne sont d'ordinaire disponibles que pour quelques dizaines de patients. Il faut dans ce cas avoir recours à des techniques d'apprentissage supervisé plus récentes et plus adaptées aux jeux de données de très grande dimension.

D'un point de vue transcriptomique, l'évolution des cancers est plus certainement liée à l'association de nombreux gènes interagissant entre eux qu'à une origine monogénique.
Les méthodes pouvant prendre en compte l'ensemble des gènes d'intérêt pour définir des profils d'expression prédictifs de la survie vont permettre d'adapter au mieux les traitements au profil du patient et le développement de nouveaux outils diagnostiques.

Prédire la probabilité de survie d'individus selon leur profil transcriptomique et l'impact éventuel d'un traitement pourra permettre d'adapter au mieux le traitement aux patients.

Le but de ce stage est donc de trouver une liste de gènes (éventuellement corrélés entre eux) en vue de prédire la survie de patients atteints de cancer du poumon et ayant reçu le traitement (vaccin thérapeutique) ou non.

Durant ce stage, que j'ai effectué au sein du département « immunologie moléculaire », mon travail a tout d'abord consisté à explorer et sélectionner les différentes méthodes de réduction de variables (dans le cas de l'analyse de survie) déjà existantes de manière à pouvoir les comparer, l'objectif du stage étant d'être capable de construire à partir d'un jeu de données le meilleur modèle de survie possible et ainsi trouver une liste de gènes permettant de prédire la survie des patients.
Dans un deuxième temps la réduction de la dimension de l'espace des gènes ainsi que la construction du modèle permettant de prédire la survie ont été effectuées.
Dans un troisième temps, il s'agissait d'appliquer le modèle construit aux données cliniques et d'étudier l'impact éventuel d'un vaccin thérapeutique anti-cancer du poumon.

Ce rapport présente premièrement le fonctionnement des puces à ADN qui représentent le support du jeu de données, puis la présentation de l'analyse de survie et du modèle de Cox.
Les principales méthodes de réduction de la dimension de l'espace des gènes sont ensuite décrites. Enfin, le choix de la méthodologie retenue à l'issue de l'étude est expliqué et son application sur des données cliniques est décrite.

Chapitre 2: Données transcriptomiques

2.1 La molécule d'ADN

L'acide désoxyribonucléique ou ADN est une grosse molécule qui est le principal support physique des gènes. Il se présente sous la forme d'un long filament et constitue la partie la plus importante des chromosomes présents dans le noyau de nos cellules.

L'ADN est constitué d'éléments appelés nucléotides, qui sont formés par l'association d'un "sucre" (le désoxyribose), d'un groupement phosphate (l'acide phosphorique) et d'une base azotée à savoir l'adénine (A), la cytosine (C), la guanine (G) ou la thymine (T).
L'ADN est formé de deux brins enroulés en double hélice, qui sont chacun formés d'une succession de nucléotides. Ces deux brins sont complémentaires et présentent la particularité de s'unir deux à deux suivant la règle de complémentarité : l'adénine appariée avec la thymine et la cytosine appariée avec la guanine (voir figure 1).

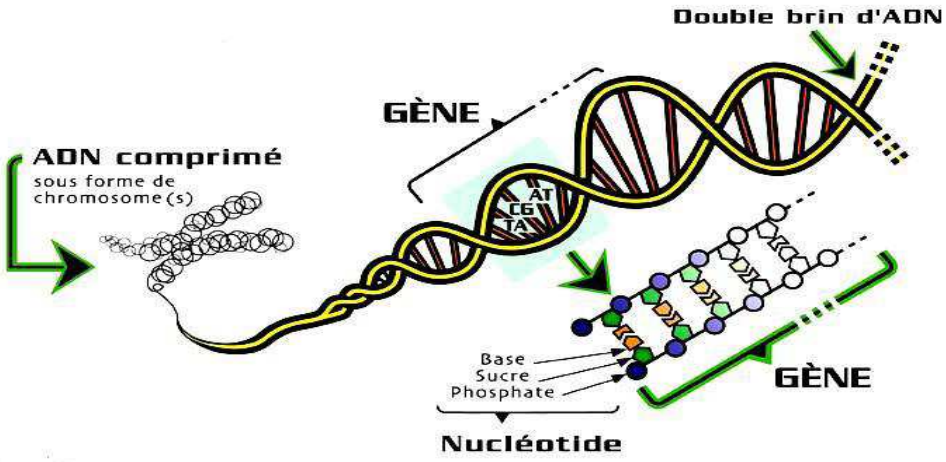

Figure 1 : Structure d'une molécule d'ADN

L'ARN (ou acide ribonucléique) est le messager de l'information génétique codée par l'ADN. Transcrit à partir de l'ADN, il est le support de la traduction de l'information génétique en protéine. Il existe différents types de molécules d'ARN qui assurent chacune une fonction particulière dans la traduction de l'information génétique. L'ARN messager (ARNm), formé à partir de l'ADN, transporte l'information génétique recueillie du noyau vers le cytoplasme. L'ARN messager va ensuite se placer dans le ribosome (unité d'assemblage des protéines) pour y élaborer directement la protéine.
[http://lesmessagersdutemps.com/ADNARN.html].
Une molécule d'ARNm correspond à un gène et chaque gène code une protéine.

Le transcriptome est l'ensemble des ARN messagers issus de l'expression d'une partie du génome d'un tissu ou d'un type de cellule. La caractérisation et la quantification du transcriptome dans un tissu donné et dans des conditions données permettent d'identifier les gènes actifs, de déterminer les mécanismes de régulation de l'expression des gènes et de définir les réseaux d'expression des gènes. Une des techniques utilisée pour mesurer simultanément le niveau d'expression d'un grand nombre de types différents d'ARN messager est celle de la puce à ADN (Figure 2).

Figure 2 : Deux puces à ADN de type Affymetrix

2.2 La puce à ADN

Les données transcriptomiques, ou données d'expression des gènes, sont des données issues de technologies variées telles que la technologie des biopuces ou puce à ADN. Celles-ci sont utilisées pour identifier et quantifier la sur- ou sous- expression d'un ensemble de gènes, en mesurant le niveau d'expression d'un grand nombre d'ARN messagers différents mais relatif au même gène, dans une situation biologique donnée.
La technologie des puces à ADN est devenue un outil puissant pour l'analyse génétique. Le principe de la biopuce repose sur l'hybridation qui consiste en un appariement, par complémentarité des bases, (A, C, G, T) de deux séquences d'ADN, dont l'une, connue, constitue la sonde (brin monocaténaire synthétique) et l'autre représente la séquence cible (figure 3).

Figure 3 : Principe d'une puce : une séquence d'ADN va servir de sonde (probe) pour capturer son brin complémentaire dans un mélange d'acides nucléiques

Transgene utilise des puces à oligonucléotides (fragments de gènes synthétiques) commercialisées par la société Affymetrix. Une puce, carré d'environ 1 cm^2, comporte quelques centaines à plusieurs dizaines de milliers d'unités d'hybridation. Chacune est constituée d'un oligonucléotide correspondant à des sondes de séquences connues.

Les sondes sont des oligonucléotides synthétisés in situ par une technique de photolithographie.

Chaque élément de la puce est un carré de 11 μm × 11 μm contenant plus de 10^7 copies d'un oligonucléotide donné (figure 3). Il est possible de synthétiser jusqu'à 1 300 000 oligonucléotides, correspondant à 38 500 gènes (il faut plusieurs oligonucléotides pour un seul gène dans sa totalité) sur une même puce auxquels s'ajoutent 7000 sondes qui servent de contrôles (positifs ou négatifs).

Les ARN messagers sont extraits de l'échantillon biologique (cellule, sang, tissu...). Après rétro transcription, les ARNm sont amplifiés et marqués à l'aide d'un nucléotide modifié (biotinylé) avant d'être hybridés sur la puce.
Le rôle de chaque sonde est de reconnaître et de fixer sa séquence complémentaire dans le mélange "cible" (ADN des cellules prélevées) appliqué à la surface de la biopuce.

Après révélation du signal par ajout de composés fluorescents, l'acquisition des images est réalisée avec des scanners à laser de haute précision, adaptés aux marqueurs utilisés. Les images sont ensuite traitées par des logiciels d'analyse d'images qui permettent de quantifier l'intensité des signaux lumineux, mais aussi de relier chaque sonde (probeset) à l'annotation qui lui correspond (nom du gène, numéro de l'ADNc utilisé, séquence de l'oligonucléotide, etc.).

Dans le cas des puces Affymetrix, chaque sonde est associée à une sonde identique à deux bases près (sonde PM pour PerfectMatch et sonde MM pour MissMatch). L'hybridation non spécifique peut être évaluée par un algorithme utilisant les ratios des signaux PM/MM.

L'intensité du signal de fluorescence pour chaque couple gène/sonde est proportionnelle à l'intensité d'hybridation, donc à l'expression du gène ciblé. L'intensité de la couleur est proportionnelle à la force du signal, le noir symbolisant l'absence de signal.

Pour détecter un gène donné, la technologie Affymetrix utilise jusqu'à 40 oligonucléotides choisis dans des régions de ce gène qui présentent le moins de similitudes avec des régions d'autres gènes. De ces régions, 11 à 20 oligonucléotides sont choisis comme "PerfectMatch" (parfaitement complémentaires à l'ARN messager cible de ce gène), et 11 à 20 oligonucléotides sont choisis comme "MisMatch" (détection de bruits de fond, variabilité dans l'intensité aussi appelé « background ») (figure 4).

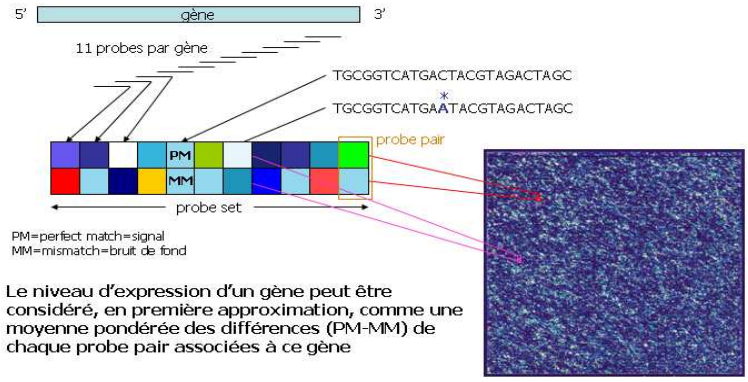

Figure 4 : Zoom d'une puce à ADN

A ce stade, les résultats sont représentés sous la forme d'un tableau comportant autant de lignes que de sondes et autant de colonnes que d'échantillons à analyser. Deux paramètres sont associés à chaque sonde : la valeur d'intensité ou de ratio (souvent sous forme logarithmique (log-ratio), représentant l'expression ou le différentiel d'expression) et la valeur de la p-value donnant la significativité statistique du signal par rapport au bruit de fond.

La significativité est estimée à l'aide d'un test non-paramétrique : le test de Wilcoxon unilatéral qui se base sur le score de discrimination R_i donné par :

$$R_i = \frac{PM_i - MM_i}{PM_i + MM_i}$$

Soient $\tau = 0.015, \alpha_1 = 0.05, \alpha_2 = 0.065$, on a :
H_0 : médiane (R_i - τ)= 0 et H_1 : médiane (R_i - τ)> 0

Un probeset est considéré comme présent si $p < 0.05$, marginal si $0.05 < p < 0.065$ et absent si $p > 0.065$.

De nombreux contrôles sont réalisés à ce stade : évaluation des contrôles positifs et négatifs, homogénéité des pixels pour un même spot, élimination des spots masqués par des taches, homogénéité et intensité du bruit de fond, corrélations des sondes ou des puces en duplicata...

Avant d'être analysés, les résultats sont normalisés. La normalisation permet de gommer les différences entre les diverses puces liées aux variations de quantité d'ARNm de départ, aux biais de marquage ou d'hybridation, aux variations de bruit de fond, etc.

De nombreuses méthodes permettent de normaliser ces données, mais aucune n'est optimale, et leur choix dépend essentiellement du type de puce utilisé et du biais attendu ou observé. Ces différentes méthodes (MASS, RMA, PLIER…) ne seront pas exposées dans ce rapport, la méthode RMA semble la plus reconnue et la plus utilisée aujourd'hui.

Les listes de gènes sont ensuite filtrées afin d'éliminer les résultats les moins fiables. Cette procédure permet généralement de soustraire tous les gènes absents dans la majorité des échantillons ou ayant une faible variation, dont l'intensité est trop proche du bruit de fond ou encore pour lesquels on suspecte une hybridation non spécifique.

Ces critères de tri sont le plus souvent fixés empiriquement et peuvent varier en fonction du nombre de gènes désirés pour l'analyse. Des critères trop sévères limiteront le nombre de faux positifs mais augmenteront le nombre de faux négatifs.

La normalisation, le filtrage ou toute autre transformation doivent cependant être utilisés avec prudence car ils peuvent introduire un biais et avoir un effet non négligeable sur l'interprétation des résultats. Les méthodes de normalisation et d'analyse restent d'ailleurs encore un domaine de recherche actif et sont à l'origine de controverses pour les scientifiques spécialisés dans ce domaine. Une fois normalisées, filtrées et transformées, les valeurs sont regroupées sous forme de matrice avec une ligne par couple gène/sonde (probeset) et une colonne par échantillon. Les données sont ainsi prêtes à servir de base à des processus de réduction de dimension et d'analyse de survie.

Chapitre 3: Analyse de survie et modèle de Cox

3.1 L'analyse de survie

L'analyse de survie est un terme générique qui désigne toute analyse de la survenue au cours du temps d'un événement « en tout ou rien », comme par exemple le décès, et ceci en présence de données censurées (données qui ne sont pas complètement observées au cours de l'étude, par exemple les patients qui ne sont pas morts durant l'étude si l'évènement étudié est le décès, ou encore les personnes qui ont été « perdues de vue » durant l'étude).

Ce type d'analyse est largement utilisé dans les études cliniques. Il permet la description de la survie (le temps s'écoulant entre le début du traitement et la survenue du décès) d'un groupe de patients mais aussi la comparaison de la survie de deux ou plusieurs groupes de patients afin d'étudier les facteurs pronostiques, c'est-à-dire les facteurs susceptibles d'expliquer la survenue du décès (ou d'un autre événement) au cours du temps.

Les méthodes d'analyse de survie permettent :

- d'obtenir une courbe de survie (description graphique des taux de survenue de l'événement étudié dans un ou plusieurs groupes de traitement)
- de déterminer la probabilité de survenue de l'événement étudié après un certain délai
- de comparer la probabilité de survenue de l'événement étudié entre différents groupes de traitement
- de mesurer l'influence d'une variable explicative sur la probabilité de survenue de l'événement étudié
- de stratifier a posteriori, sur la variable explicative, les groupes à comparer et de calculer un nouveau de degré de signification.

Rappelons que, bien que le terme de survie soit le terme consacré au décès pour des raisons historiques (terme utilisé d'abord en cancérologie où la survie des patients est un des critères d'efficacité du traitement), les méthodes d'analyse de survie ne s'appliquent pas seulement à l'étude des décès mais peuvent s'appliquer également à l'étude de tout évènement « unique » susceptible d'apparaître au cours d'un essai : 1ère apparition d'un évènement indésirable, 1er épisode de rechute, 1ère normalisation d'un critère,…

Dans le cadre du sujet du stage, l'évènement étudié sera le décès.

3.2 L'estimation de la survie : méthode de Kaplan-Meier

La courbe de survie est la représentation la plus employée pour décrire la dynamique de survenue des décès au cours du temps. C'est une courbe qui représente le taux de survie **S** (la probabilité de survivre au moins jusqu'au temps t) en fonction du temps.

L'estimation des courbes de survie fait appel principalement à la technique de Kaplan-Meier, méthode non-paramétrique (c'est-à-dire n'utilisant pas un modèle dans lequel la distribution des durées de survie est une fonction du temps) dans laquelle on calcule une estimation de la probabilité de survie à chaque survenue d'un épisode de l'évènement étudié (par exemple le décès).

Dans la méthode de Kaplan-Meier, le temps de participation observé est découpé en intervalles de temps, qui débutent à l'instant t_i où survient un décès et se terminent juste avant le décès suivant, et la survie est estimée sur chaque intervalle de temps, ce qui va donner à la courbe un aspect en « marche d'escalier ».

La méthode du calcul du taux de survie repose sur le principe des probabilités conditionnelles :
Soit S_1 … S_t les probabilités de survie aux temps (en années) 1…t, $S_{2|1}$ est la probabilité de vivre 2 ans (la 2ème année) pour les sujets ayant vécu 1an, et on a :

P(vivre 1 et 2 ans) := $S_2 = S_1 \times S_{2|1}$ et à un temps t, la probabilité (cumulée) de survie est le produit des probabilités de survies conditionnelles calculées pas à pas :

$$S_n = S_1 \times S_{2|1} \times S_{3|2} \times \dots \times S_{t|t-1}.$$

La méthode de Kaplan-Meier calcule une estimation de la probabilité de survie : pour tout intervalle $[t_i ; t_{i+1}]$, la probabilité de survivre au temps t_{i+1} sachant que l'on était vivant à t_i $S(t_i+1|t_i)$ est estimée par :

$1-\frac{d_i}{n_i}$ avec d_i le nombre de décès dans l'intervalle $[t_i, t_{i+1}[$, n_i le nombre d'individus exposés au risque de décéder juste avant t_i. : $n_i = n_{i-1} - d_{i-1} - c_{i-1}$ avec c_i le nombre de censurés dans l'intervalle $[t_i, t_{i+1}[$.

L'estimation de la probabilité de survivre juste après la date t_i, appelée « Estimateur de Kaplan-Meyer », correspond au produit des probabilités conditionnelles de survie et est donné par la formule :

$$\hat{S}(t) = \prod_{t_i \le t} (1 - \frac{d_i}{n_i})$$

L'intervalle de confiance associé à la fonction de survie S(t), sous l'hypothèse que S(t) est distribué normalement, est donné par la formule :

$$IC = \hat{S}(t) \pm Z_{\frac{\alpha}{2}}.(\text{var}\,\hat{S}(t))^{\frac{1}{2}}$$

avec α le risque de 5% et $Z_{\alpha/2}$ la valeur critique à lire dans la table de la loi normale.

Ainsi, dans l'exemple suivant (table I), entre t_0=0 et t_1=0,03, il n'y a pas d'individu censuré et 1 décès a été observé donc $S(t_1|t_0)$=0.998 et $S(t_0)$=0.998.

Table I : Estimation de la probabilité de survie selon la méthode de Kaplan-Meier [1]

t_i	n_i	d_i	c_i	$S(t_{i+1}\|t_i)$	$S(t_i)$
0.00	539	1	0	0.998	0.998
0.03	538	2	2	0.996	0.994
0.19	534	1	0	0.998	0.993
0.23	533	1	1	0.998	0.991
0.26	531	3	1	0.994	0.985
0.29	527	1	0	0.998	0.983
0.32	526	1	0	0.998	0.981
0.39	525	1	0	0.998	0.980
0.46	524	3	0	0.994	0.974
0.49	521	1	0	0.998	0.972
0.52	520	2	0	0.996	0.968
0.55	518	2	3	0.996	0.965
0.92	513	2	0	0.996	0.961
0.95…	511…	2…	0…	0.996…	0.957…

La courbe de Kaplan-Meier suivante (figure 5) représente en fonction du temps la proportion des sujets initialement inclus dans l'essai clinique de notre exemple et toujours vivant au temps t (les traits verticaux sur la courbe représentent les individus censurés).

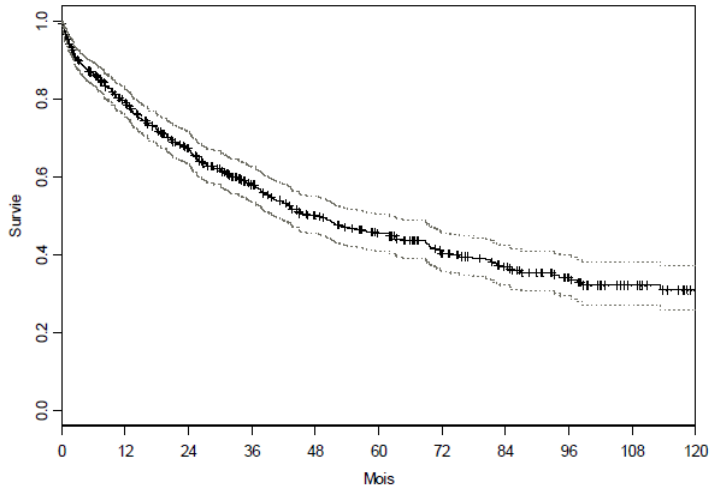

Figure 5 : estimation de survie selon la méthode de Kaplan-Meier

3.3 Risque instantané de décès

Le risque instantané de décès au temps t noté *h(t)* représente la probabilité (par unité de temps), pour une personne vivante à une date donnée, de décéder dans l'instant qui suit, c'est la probabilité d'apparition de l'événement dans un intervalle de temps [t, t+dt] sachant que l'événement ne s'est pas réalisé avant l'instant t. L'intervalle *dt* peut être 1 jour, 1 mois, 1 an et dépend de la gravité associée à l'événement étudié, ainsi qu'au risque de survenir qu'on lui attribue.

Le risque est définit par la formule : $$h(t) = \lim_{dt \to 0} \frac{P(T(t+dt|T > t)}{dt}$$

C'est une probabilité conditionnelle par unité de temps, le numérateur correspondant à la probabilité de décéder juste après le temps t, sachant que le sujet est en vie au temps t.

On montre que $h(t) = \frac{-S'(t)}{S(t)}$, avec *S* la probabilité de survivre au moins jusqu'au temps t, ainsi $S(t) = \exp(-\int h(t)dt)$.

Si le risque instantané de décès est constant au cours du temps (*h(t)=c*), alors *S(t) = exp(-ct)*, c'est à dire que la fonction de survie suit une distribution exponentielle.

La fonction de risque cumulé H est reliée à la fonction de survie *S* et la fonction de risque instantané *h* par :

$$H(t) = -\ln(S(t)) \qquad \textbf{et} \qquad H(t) = \int_0^t h(x)dx$$

3.4 Le modèle de Cox

Le modèle de régression à risques proportionnels proposé par Cox en 1972 pour étudier la relation entre le temps d'apparition d'un événement (par exemple le décès) et un ensemble de variables explicatives (par exemple les gènes) en présence de censure (s'il n'y a pas de censure, le modèle de régression logistique peut aussi être utilisé), a eu un impact considérable dans l'analyse des données de survie, tant du point de vue théorique que pratique, et est rapidement devenu le modèle le plus utilisé. [2,30]

Il suppose cependant (comme tout modèle de régression linéaire multiple) qu'il y ait plus d'observations que de variables, des données complètes et des variables non fortement corrélées entre elles. Ces hypothèses sont souvent impossibles à satisfaire dans la pratique. En oncologie par exemple, la recherche de descripteurs biologiques liés à la durée de survie suppose de prendre en compte l'expression de milliers de gènes pour généralement seulement quelques dizaines d'individus.

La régression semi-paramétrique (estimation de l'influence des facteurs exogènes sans hypothèse concernant la distribution de base) de Cox est la méthode de référence pour l'analyse des données longitudinales issues d'enquêtes de cohortes ou d'essais cliniques.

Au même titre que les autres méthodes de régression, l'expression de la régression de Cox permet de réaliser des prévisions sur la survie d'un patient donné en connaissant ses caractéristiques. On utilisera, d'une part, l'estimation des paramètres $ß_i$ et, d'autre part, la valeur des covariables X_i mesurées ou recueillies chez ce patient.

3.4.1 Représentation du modèle de Cox et fonctions liées

Le modèle de Cox est un modèle multivarié spécialement utilisé pour analyser les données censurées. Il permet d'exprimer la relation entre le risque instantané de survenue de l'événement étudié (ou encore appelé « taux d'incidence instantané ») h(t) et des facteurs de risque exprimés sous la forme de variables explicatives quantitatives X_1,..., X_p (avec X_k le vecteur de longueur n, le nombre d'individus) par la formule suivante :

$$h(t|X) = h_0(t).\exp(\beta_1 X_1 + \ldots + \beta_p X_p)$$

où les $ß_k$ sont les coefficients de la régression de Cox et le terme $h_0(t)$ est le risque instantané de sujets ne présentant aucun facteur de risque, encore appelé « risque instantané de base ».
($h_o(t)$ est là pour faire en sorte de respecter une des hypothèses du modèle de Cox : l'hypothèse des risques proportionnels).
La taille de l'effet du traitement est souvent mesurée à l'aide du rapport des risques instantanés (HR Hazard Ratio en anglais) qui est le rapport entre le risque instantané dans chacun des deux groupes de patients que l'on compare.

Soit T une variable aléatoire de durée de vie, la fonction *f(t)* représente la densité à valeurs dans R^+ et $F(t) = P(T < t) = \int_0^t f(x)dx$ la fonction de répartition qui mesure la probabilité de décéder au plus tard en t. La fonction de survie ou taux de survie *S*, déjà présentée dans le paragraphe 3.2, est la probabilité (en fonction de t) de survivre au moins un certain temps t à compter d'un instant de référence (le début de l'étude par exemple).
Cette fonction s'écrit :

$$S(t) = P(T \geq t) = 1 - F(t) = \int_t^{\infty} f(x)dx$$

La fonction de risque instantané est reliée à la fonction de survie par la relation :

$$h(t) = \frac{f(t)}{S(t)} = \frac{-d(\ln(S(t))}{dt} \qquad \text{ou encore} \qquad S(t) = \exp(-\int_0^t h(x)dx)$$

La fonction de survie est l'élément majeur de l'étude de phénomènes de survenue d'événements, sa représentation graphique s'appelle : courbe de survie.

Enfin, l'espérance de survie si on a vécu jusqu'à t est donnée par : $E(t) = \frac{\int_t^{\infty} S(x)dx}{S(t)}$.

3.4.2 Méthode du maximum de la vraisemblance

Pour estimer les paramètres du modèle de Cox la méthode la plus utilisée est celle qui consiste à maximiser la vraisemblance des paramètres du modèle.

A l'instant t_i, il y a R_i patients encore à risque, la probabilité de décès en t_i de chaque sujet j est donnée par la formule :

$$\lambda_0(t_i)\exp(\beta X_i)\Delta_t$$

La probabilité que ce soit le sujet i qui décède est donnée par la formule :

$$v_i(\beta) = \frac{h_0(t_i)\exp(\beta X_i)\Delta_t}{\sum_{j \in R_i} h_0(t_i)\exp(\beta X_j)\Delta_t} = \frac{\exp(\beta X_i)}{\sum_{j \in R_i}\exp(\beta Z_j)}$$

La vraisemblance d'une valeur pour un paramètre donné, est la probabilité d'obtenir une valeur telle que celle observée, elle est donnée par la formule :

$$V(\beta) = \prod_{i=1}^{n} v_i(\beta) = \prod_{i=1}^{n} \frac{\exp(\beta X_i)}{\sum_{j \in R_i}\exp(\beta X_j)}$$

La vraisemblance partielle ne dépend pas du risque instantané de base h_0, de plus les données censurées ne participent pas à son calcul. La log-vraisemblance est donnée par la formule :

$$L(\beta) = \sum_{i=1}^{n} (\beta X_i - \log(\sum_{j \in R_i} \exp(\beta X_j)))$$

L'estimateur du maximum de vraisemblance est l'estimateur qui associe aux observations la valeur pour laquelle la probabilité de l'observation est la plus forte dans le modèle, c'est la valeur $\hat{\beta}$ de β qui rend maximum *L(β)* c'est-à-dire telle que : $\frac{\partial L(\hat{\beta})}{\partial \beta} = 0$

[http://cybertim.timone.univ-mrs.fr/enseignement/doc-enseignement/statistiques/EISIS-SurvieCox-RG/docpeda_fichier]

3.4.3 Hypothèses du modèle de Cox

Avant d'effectuer un modèle de Cox il convient de vérifier les deux hypothèses requises pour l'utiliser et l'interpréter à bon escient. Ces deux hypothèses sont :

- La log-linéarité des variables : le logarithme de *h(t)* est une fonction linéaire des variables explicatives X_i. (les graphiques de la figure 6 montrent la vérification de cette hypothèse sur le jeu de données publiques présenté dans le paragraphe 6.2)

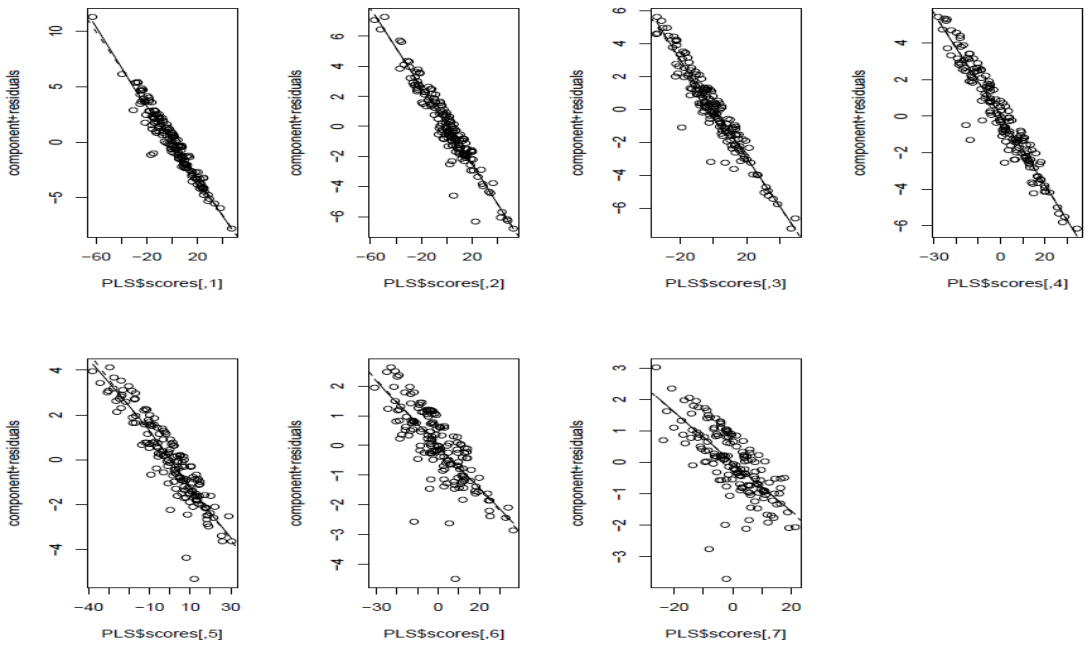

Figure 6 : évaluation graphique de l'hypothèse de la log-linéarité des variables

- La proportionnalité des risques : le ratio des fonctions de risque instantané pour deux patients de deux groupes à comparer est constant au cours du temps, il ne dépend que des variables explicatives, autrement dit les coefficients de régression sont

indépendants du temps. (les graphiques de la figure 7 montrent la vérification de cette hypothèse sur le jeu de données publiques présenté dans le paragraphe 6.2)

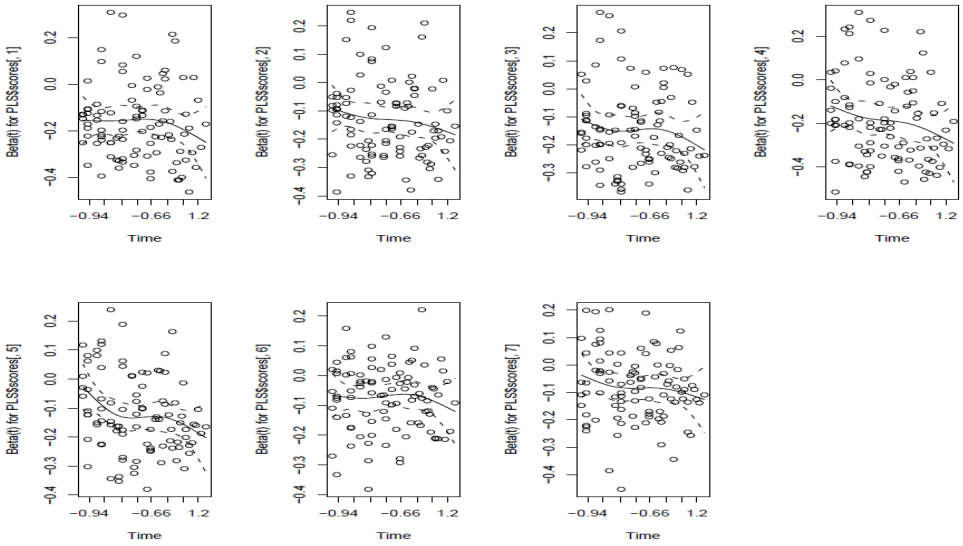

Figure 7 : évaluation graphique de l'hypothèse des risques proportionnels (les coefficients β_j sont indépendants du temps)

La proportionnalité des risques au cours du temps est difficile à voir sur le graphique de la courbe de survie (taux de survie S(t) en fonction du temps). L'hypothèse est mieux explorée en représentant la fonction y = log(-log(S(t))). (le graphique de la figure 8)

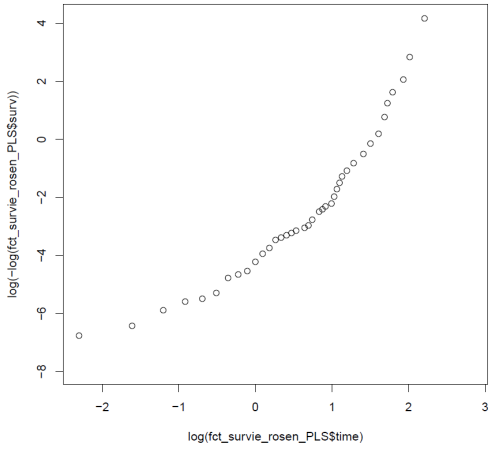

Figure 8 : évaluation graphique de l'hypothèse des risques proportionnels (graphique de log(-log(S(t))) en fonction de log(t))

Plus simplement, on considère souvent que, dès que les courbes de survie obtenues dans les différents groupes ne se coupent pas, l'hypothèse des risques proportionnels est « acceptable ».

- La nature de la censure : le modèle de Cox suppose que la censure est non-informative, c'est-à-dire que la censure est indépendante du risque de décéder.

Chapitre 4: Réduction du nombre de variables

4.1 Description des différentes méthodes

L'analyse des profils d'expression des gènes est de plus en plus utilisée (en oncologie entre autre) pour découvrir de nouveaux marqueurs biologiques et de nouvelles cibles thérapeutiques. Elle suppose la prise en compte de l'expression de milliers de gènes en regard de seulement quelques dizaines d'individus.
Le caractère hautement multi dimensionnel de ces données rend l'application des approches classiques difficile, les modèles de Cox classiques sont mis en défaut face à de telles quantités de données. La solution est d'opérer d'abord une réduction de l'espace des gènes, puis de construire un modèle de Cox avec les variables sélectionnées.
Ici sont présentées différentes méthodes, trouvées dans la littérature, pour réduire la dimension de l'espace des gènes.

L'Analyse des données en cluster :

Le but de cette méthode est de réduire le nombre de variables en les regroupant en fonction de leur profil d'expression génique, c'est une méthode qui permet de classer des groupes en fonction de leur contenu. L'inconvénient majeur est que la relation entre le temps de survie et les variables explicatives originales est « dérangée » par un lien entre l'expression des gènes et les « étiquettes » des clusters. De plus la procédure n'utilise pas efficacement l'information prédictive disponible sur l'expression des gènes [3 – 5].

Les arbres de décision, forêts aléatoires et les méthodes de classification :

Ces méthodes utilisent l'expression de groupes de gènes (qui sont trouvés par regroupement hiérarchique / cluster) en tant que prédicteurs de survie. Le premier inconvénient est que ces prédicteurs sont utilisés dans un modèle à risque proportionnel de Cox comprenant des interactions d'ordre 1, or dans le cadre du sujet il peut y avoir des interactions au niveau de la construction du modèle. Le deuxième inconvénient est que ces méthodes nécessitent un grand nombre de sujets pour découvrir des interactions avec succès. De plus le modèle est sensible à la méthode de clustering utilisée, et des groupes hétérogènes peuvent présenter une moyenne d'expression fortement corrélée avec la variable réponse [6, 8, 14, 15].

La méthode de sélection univariée :

Dans cette méthode, un modèle de régression de Cox est généré pour chaque gène et on se base sur la p-value obtenue en utilisant un test de Wald nous aurions également pu utiliser le test du score ou du rapport de vraisemblance qui vérifie que le coefficient β est nul.

$$W = z^2 = \left(\frac{\beta}{\sqrt{\sigma_\beta}} \right)^2 \sim \chi_1^2$$

L'inconvénient est que la sélection univariée de gènes ne tient pas compte de la corrélation entre les gènes, donc de nombreux gènes fortement corrélés peuvent être sélectionnés. Par conséquent, beaucoup de gènes qui sont significatifs de façon univariée ont des p-valeurs non significatives dans le modèle à risques proportionnels de Cox multivarié [7 – 9].

Le modèle de Cox avec pénalité quadratique (type L2) / régression ridge :

La régression ridge (Hoerl et Kennard - 1970) est une variante de la régression linéaire multiple. Elle permet de contourner l'obstacle de la colinéarité entre les variables explicatives en renonçant à la méthode des moindres carrés pour estimer les paramètres du modèle et en modifiant la matrice X'X pour rendre à son déterminant une valeur acceptable (différent de 0) car lorsque X'X est non inversible la précision de l'estimateur est mauvaise. L'idée est de rendre la matrice X'X inversible en lui ajoutant dans la définition de l'estimateur une matrice symétrique positive appelé « facteur de biais ». Les cœfficients obtenus sont les « coefficients ridge » :

$$\beta_{ridge} = (X'X + \lambda I)^{-1} X'Y \text{ avec } 0 \leq \lambda \leq 1$$

Cependant, cette méthode ne possède pas de procédure de sélections de variables afin de réduire la dimension de l'espace des covariables [10 – 13].

La Régression en Composantes Principales (PCR):

La Régression en Composantes Principales (encore appelée Analyse en Composante Principales Supervisée) est similaire à l'Analyse en Composante Principale classique [16], en effet elle défini des nouvelles variables appelées « composantes principales » qui sont des combinaisons linéaires des variables d'origine. Cette méthode gère le problème de la grande dimension en n'utilisant que les composantes qui ont la plus forte corrélation estimée avec la réponse Y pour l'ACP. Il y a cependant des inconvénients à la méthode PCR : les 1ères composantes principales ne sont pas forcement celles qui expliqueront le mieux Y, de plus, la méthode demande souvent plus de composantes que la PLS pour donner une bonne prédiction [10, 17, 18]. Une explication plus détaillée de cette méthode est disponible en annexe A.

La méthode LARS (modèle de Cox avec pénalité de type L1):

Cette approche consiste à pénaliser la méthode des moindres carrés par ajout d'une contrainte de type L1 sur la vraisemblance partielle du modèle (donc sur l'estimation des coefficients).
La méthode LARS utilise la connexion entre les méthodes LAR et LASSO pour rendre utilisable dans le cas de données de très grande dimension la méthode de sélection de variables LASSO développée et adaptée dans le cadre du modèle de Cox.
L'algorithme du LARS commence avec tous les coefficients de régression β nuls, et consiste à sélectionner les variables explicatives pertinentes en les choisissant les moins corrélées possibles entre elles et le plus corrélées possible avec la variable réponse Y [19 - 24]. Une explication plus approfondie de cette méthode est disponible en annexe B.

La régression des moindres carrés partiels (PLSR) :

La régression linéaire multiple (RLM) constitue le plus simple des modèles linéaires qui cherche à expliquer une ou plusieurs variables réponse à l'aide de p variables explicatives numériques (les covariables), en considérant n individus.
Cependant cette méthode possède quelques défauts comme par exemple l'incapacité à prendre en compte les données manquantes, ce qui conduit souvent au rejet de beaucoup d'observations incomplètes et pourtant contenant de l'information utile.
Un autre défaut de la RLM est l'indétermination des estimateurs lorsque le nombre d'observations est inférieur au nombre de covariables. En effet, dans cette situation la matrice X des covariables n'est pas de rang plein (c'est-à-dire son rang est inférieur à p) et la matrice **XX'** n'est pas inversible (det**(XX')** $\approx$ **0**) donc l'Estimateur des Moindres Carrés Ordinaire donné par $\mathbf{(XX')^{-1}X'Y}$ est mis en défaut.
Le dernier inconvénient de la RLM est la grande sensibilité de la méthode à la colinéarité entre les covariables (multicolinearité), ce qui entraine une variance infinie de l'estimateur. En effet, on sait que l'erreur quadratique moyenne de l'estimateur vaut :

$$MSE = \sigma^2 trace((X'X)^{-1}) = \sigma^2 \sum_{j=1}^{p} \frac{1}{\lambda_j(X'X)}$$ avec λ_j(X'X) la j$^{\text{ième}}$ valeur propre de la matrice X'X, or la multicolinéarité implique que $\lambda_j(X'X) \approx 0$ et donc la MSE devient très grande et les coordonnées de l'estimateur ne sont plus interprétables, donc le modèle de la multicolinéarité influe sur la variance de l'estimateur.

La régression PLS (Partial Least Squares) est une technique permettant de contourner tous ces obstacles. L'idée de cette méthode est, tout comme celle de l'ACP, est de réduire le nombre de covariables en définissant de nouvelles variables (appelées « variables latentes ») qui sont des combinaisons linéaires des variables d'origine. Cette méthode va créer une nouvelle matrice (appelée « matrice des scores ») à partir de la matrice d'origine, dont les k colonnes seront les composantes (variables latentes), avec k < p. Une régression de la variable réponse sur les composantes trouvées est effectuée à chaque étape de la méthode [25 - 30].
La régression PLS est la méthode la plus couramment utilisée dans la réduction de la dimension de l'espace des covariables dans un modèle de régression, elle est expliquée plus en détails en annexe C.

4.2 Les méthodes les plus pertinentes

D'après les recherches faites dans la littérature, les deux méthodes qui pourraient être appropriées pour réaliser la réduction de la dimension de l'espace des covariables dans le cadre du sujet du stage seraient :

- La méthode LARS
- La régression PLS

Comparées à d'autres méthodes de régression pour des données colinéaires, ces méthodes ont l'avantage (tout comme la Régression en Composantes Principales) d'utiliser l'information qui est dans la variable réponse.

Un autre stratégie de réduction de dimension peut être utilsée, il s'agit de transformer les données afin de réduire la dimension du jeu de données. Ainsi, la déviance résiduelle pourra être utilisée à la place du temps de survie comme variable réponse et/ou la matrice X des covariables pourra être remplacée par la matrice « kernel » K (définie à partir du noyau gaussien).

Les notions de déviance résiduelle et de noyau gaussien sont décrites ci-dessous.

Résidus de la déviance :

L'étude des durées de survie peut être abordée d'une autre façon que la façon standard : au lieu de considérer Y le temps de survie comme variable réponse, on représente l'expérience par une variable binaire appelée « processus ponctuel associé » N(t) qui vaut 0 tant que l'évènement (le décès) n'a pas lieu et 1 après, c'est-à-dire $N(t) = 1\{Y \le t\}$ avec $t \ge 0$ et 1 une fonction indicatrice.
L'intensité λ(t) du processus à l'instant t est donné par : $\lambda(t) = I(t)h(t)$ où I(t)=1{t<T} $I(t) = 1\{t \le T\}$ est l'indicateur de présence du sujet avant l'instant t.
L'intensité cumulée du processus ponctuel N est donnée par la formule suivante :
$\Lambda(t) = \int_0^t \lambda(u)du$.
La différence entre le processus ponctuel N et l'intensité cumulée Λ est une martingale M.

Une fois le modèle estimé, afin de vérifier sa qualité d'ajustement on considère la martingale résiduelle associée au sujet i : $M_i(t) = N_i(t) - \Lambda(t) = N_i(t) - \int_0^t I_i(s)\exp(\beta' X_i(s))dH_0(s)$

(avec H_0 la fonction de risque cumulée de base), pour chaque individu i on compare au temps t_i le nombre de morts sur l'intervalle [0 ; t_i] sachant que $T \geq t$ (excès de morts). On a une estimation de M en remplaçant β et *Ho* par leurs estimateurs respectifs.

Pour un modèle de Cox sans covariables dépendant du temps, T_i représentant la durée d'observation du sujet i et D_i le statut final, l'estimation de M se réduit à la forme simple :

$$\hat{M}_i = D_i - \hat{H}_0(T_i)\exp(\hat{\beta}' Z_i)$$

Une variante aux résidus de martingales est donnée par les résidus de la déviance, c'est une renormalisation des résidus de martingales de manière à corriger leur asymétrie (en effet la valeur maximale de M est 1 tandis que sa valeur minimale est $-\infty$).

Le résidu de la déviance du sujet i noté d_i est défini comme :

$$d_i = sign(\hat{M}_i).(-2(\hat{M}_i + D_i \log(D_i - \hat{M}_I)))^{1/2}$$

Les résidus de la déviance sont utilisés à la place du temps de survie comme variable réponse dans un modèle dans le but de réduire considérablement le coût en temps de calcul des algorithmes et d'augmenter l'efficacité de la performance de ces derniers. Cette alternative est très intéressante car particulièrement simple à mettre en œuvre [27,31].

Noyau gaussien:

L'utilisation d'un modèle à noyau (comme les K-plus-proches voisins ou encore la Fenêtre de Parzen) est une astuce qui permet de transformer beaucoup d'algorithmes « linéaires » (par exemple en régression) en algorithme « non-linéaire » tout en gardant les mêmes propriétés d'optimisation. Dans les algorithmes « linéaires », on a une notion de produit scalaire qu'on va remplacer par un nouveau produit scalaire qui sera « non-linéaire » et qui correspondra à une fonction à noyau $K(x, y)$
[http://www.iro.umontreal.ca/~pift6266/A06/cours/kernels.pdf, 25]

La fonction à noyau la plus simple est le noyau gaussien donné par la formule : $K(x, y) = \exp(-\frac{\|x - y\|^2}{2\sigma^2})$ (dans le cas de données centrées réduites σ^2 =1)

Cette fonction appliquée sur la matrice $X_{n,p}$ des données d'origine va permettre de construire une nouvelle matrice $K_{n,n}$ (appelée « kernel ») définie positive qui va permettre d'utiliser une méthode performante appelée « kernel PLS ».

Le chapitre 6 de ce rapport comparera, en utilisant un jeu de données publiques (trouvé sur internet dans des articles intéressants), la performance prédictive de ces 2 méthodes.

4.3 Sélection du nombre de composantes

4.3.1 Régression PLS

Au niveau de la régression PLS, le nombre de composantes t est généralement déterminé par validation croisée : on calcule à l'aide du modèle à t composantes les prédictions Y_i^t et $Y_{i,-i}^t$ de Y_i, calculées respectivement en utilisant toutes les observations puis sans utiliser l'observation i.

On calcule ensuite pour la composante t les critères RSS^t (Residual Sum of Square) et $PRESS^t$ (PRedicted Error Sum of Square) définis par les formules suivantes :

$$RSS^t = \sum_{i=1}^{n} (Y_i - Y_i^t)^2 \text{ (avec } RSS^0 = n-1) \quad \text{et} \quad PRESS^t = \sum_{i=1}^{n} (Y_i - Y_{i,-i}^t)^2$$

Une composante T^t est considérée comme significative et on peut alors décider de la retenir lorsque :

$$(PRESS^t)^{1/2} \leq 0.95.(RSS^{t-1})^{1/2} \Leftrightarrow Q_t^2 \geq 0.0975 \text{ (voir définition de } Q_t^2 \text{ ci-dessous)}$$

Dans leur article, Gauchy et Chagnon [32] présentent différentes méthodes de sélection de variables dans le cas de la régression PLS, ils comparent ces différentes méthodes sur 5 jeux de données publiques.
Une des méthodes qu'ils proposent dans l'article et qui donne (selon eux) les meilleurs résultats est la méthode BQ (Backward-Q^2_{cum}). Cette méthode de sélection descendante élimine à chaque étape la variable dont le coefficient de régression est le plus faible en valeur absolue. On définit le pouvoir prédictif de la $t^{ième}$ composante par la formule :

$$Q_t^2 = 1 - \frac{PRESS_t}{RSS_{t-1}}$$

Le critère de sélection du meilleur modèle est basé sur l'indicateur de bonne prédiction du modèle Q^2_{cum} donné par la formule suivante :

$$Q_{cum}^2 = 1 - \prod_{t=1}^{k} \frac{PRESS_t}{RSS_{t-1}}$$

Où k est le nombre de variables latentes calculées par validation croisée dans le modèle de régression PLS.

C’est un indice compris entre 0 et 1, il est calculé à chaque étape et le modèle correspondant à la valeur de Q^2_{cum} la plus proche de 1 est sélectionné.

4.3.2 Méthode LARS

Afin d’évaluer la pertinence du modèle Lars-Cox et la qualité de la régression, on peut repérer les variables explicatives les plus pertinentes selon le critère du C_p de Mallows qui se doit d’être le plus petit possible. Cet indice est donné par la formule suivante :

$$C_p = \frac{SSE_p}{\sigma^2} - N + 2P$$

avec $SSE_p = \sum_{i=1}^{N} (Y_i - Y_{pi})^2$ l’erreur de prédiction du modèle avec P variables, N le nombre d’observations et Y_{pi} est la valeur prédite par le modèle avec P variables pour la $i^{ième}$ observation.

Pour confirmer le nombre de variables explicatives pertinentes selon le C_p de Mallows on trace la courbe du C_p en fonction de p le nombre de variables sélectionnées.

Chapitre 5: Capacité prédictive du modèle

5.1 Les courbes ROC

Quand la variable réponse est binaire, la qualité de prédiction d'un modèle est généralement évaluée par le taux de bien classés (on évalue par rapport à un seuil de discrimination (qui varie) la valeur diagnostique d'un signe dans la maladie et on classe les individus selon la réponse du test : vivants ou morts).

La qualité de prédiction du modèle est caractérisée par 2 critères complémentaires que sont la sensibilité et la spécificité.
La sensibilité est la probabilité que le test soit positif si la maladie est présente ou encore la proportion de tests positifs parmi la population malade, elle se mesure chez les malades seulement. Elle s'accompagne toujours d'une mesure qu'est la spécificité, c'est la probabilité d'obtenir un test négatif chez les non-malades ou encore la proportion de tests négatifs parmi la population non-malade, elle se mesure chez les non-malades seulement. On a tout intérêt à ce que la sensibilité et la spécificité soient les plus grandes possible.

Le ROC (Receiver Operating Characteristic) est utilisé comme une mesure de la performance du classifieur binaire quand le seuil de discrimination varie. Graphiquement, on représente souvent la mesure ROC sous la forme d'une courbe qui donne le taux de classification correcte dans un groupe (la sensibilité ou taux de faux positifs) en fonction du taux de classifications incorrectes pour ce même groupe (le complément à 1 de la spécificité ou taux de faux positifs).
[http://en.wikipedia.org/wiki/Receiver_operating_characteristic,http://gim.unmc.edu/dxtests/ROC1.htm,http://www.graphpad.com/help/prism5/prism5help.html?sensitivity_and_specificity.htm]

La courbe ROC (figure 6) est un outil graphique permettant de représenter la capacité d'un test à discriminer entre les populations de deux groupes, à un temps donné.

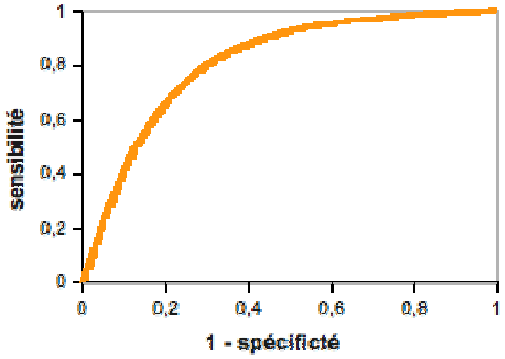

Figure 6 : une courbe ROC

Extension aux données de survie:

Dans le cas où l'on considère des données de survie, celles-ci peuvent être vues sous la forme d'une variable binaire dépendant du temps : $N_i(t) = 1(T_i < t)$ avec T_i la durée de survie de l'individu i et 1 une fonction indicatrice.

Dans le cadre de l'analyse de survie, pour évaluer la capacité prédictive du modèle on utilise des définitions de la spécificité et de la sensibilité dépendant du temps bien particulières, ensuite des courbes ROC (et aires sous la courbe ROC) dépendantes du temps peuvent être construites et interprétées.

Soit M_i la valeur d'un marqueur prédictif de la survie, qui peut être égale a la partie explicative $x_i'\beta$ d'un modèle de régression. M_i peut par exemple être issu de la phase de calibration d'un modèle Cox-PLS.

Soit $dN_i(t) = N_i(t) - N_i(t^-)$.

La qualité du modèle est caractérisée par la sensibilité définie par :

$$P(M_i > c \mid T_i = t) \text{ ou encore } P(M_i > c \mid dN_i(t) = 1)$$

et la spécificité définie par :

$$P(M_i < c \mid T_i > t) \text{ ou encore } P(M_i < c \mid N_i(t) = 0), \text{ avec c un seuil.}$$

La sensibilité mesure l'espérance de la proportion d'individus avec $M_i > c$ parmi les sujets qui décèdent à l'instant t, et la spécificité mesure l'espérance de la proportion de sujets avec $M_i < c$ parmi les individus qui sont a risque à l'instant t et survivent au delà de t. [25, 33]

5.2 L'aire sous la courbe (AUC)

Si on a besoin d'un index simple et quantitatif de la performance d'un modèle, l'AUC est un bon indicateur, il correspond à l'aire sous la courbe de ROC. Ainsi, un graphique simple représentant l'aire sous la courbe ROC en fonction du temps permet d'obtenir une vue d'ensemble de la qualité prédictive du modèle au cours du temps (Figure 7). Elle est connue pour représenter une mesure de la concordance entre la réalité et ce que prédit le modèle sur le statut du patient (vivant ou mort).

L'AUC indique la probabilité pour que la fonction score (le score de Cox dans le cas du modèle de Cox) classe les patients dans le bon groupe, elle montre la précision du modèle.

Dans le cadre les courbes ROC dépendantes du temps, cette probabilité est égale à :

$$P[M_j > M_k \,|\, T_j < T_k] = 2\int_t P[M_j > M_k \,|\, \{T_j = t\} \cap \{t < T_k\}].P[\{T_j = t\} \cap \{t < T_k\}]dt$$

Lorsqu'elle est égale à 1, le test est parfait et peut identifier tous les malades sans faux-positifs (modèle qui prédit bien).

Lorsqu'elle est égale à 0.5 le test est sans valeur et détecte autant de vrais-positifs que de faux-positifs (le modèle ne prédit pas bien).

[http://gim.unmc.edu/dxtests/ROC3.htm,http://en.wikipedia.org/wiki/Receiver_operating_characteristic,25,32]

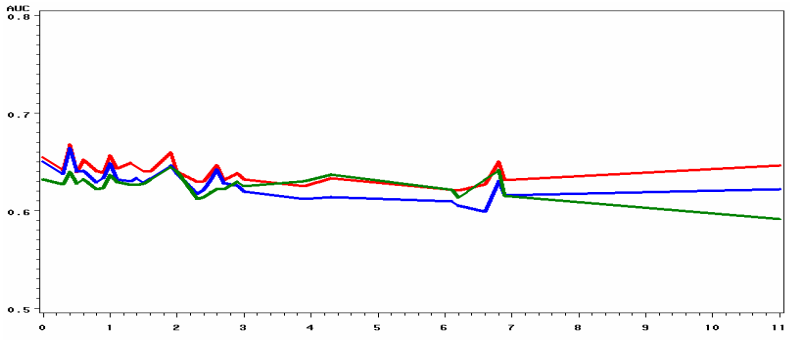

Figure 7 : représentation graphique de l'aire sous la courbe ROC

Chapitre 6: Application aux données cliniques

6.1 Travail sous R:

6.1.1 Présentation de R

Ce logiciel est un environnement de manipulation d'analyse statistique et de représentation graphique des données, qui possède son propre langage de programmation. Il fournit des procédures usuelles et possède des facilités graphiques performantes pour explorer les données. Si les fonctions de base ne suffisent pas, de nombreux modules (packages) additionnels permettent d'étendre ces dernières dans différents domaines.

Nommé par la lettre R en référence à ses deux auteurs Ross Ihaka et Robert Gentelman (auxquels sont venus depuis s'ajouter de nombreux chercheurs), son nom est aussi un clin d'œil au langage S. En effet, c'est au départ un clone de l'outil d'analyse statistique S+ (logiciel payant commercialisé par MathSoft et developpé par Statistical Sciences autour du langage S conçu par les laboratoires Bell) qui a petit à petit acquis son autonomie (il existe depuis une vingtaine d'années) et est devenu une référence dans le monde de la statistique de part son caractère libre qui en fait un outil très dynamique.
Les codes sources et modules d'application de ce langage sont disponibles sur le site du Comprehensive R Archive Network (CRAN) et peuvent être recopiés et diffusés gratuitement. Les instructions sont saisies dans une console et exécutées au fur et à mesure de leur introduction dans la console (mode interactif).

Dans un premier temps développé pour les systèmes d'exploitation libres et gratuits à savoir UNIX et Linux, R est très vite devenu disponible gratuitement (suivant les termes des Licences Publiques Générales, GLP) pour les systèmes d'exploitation Windows et Mac OS X. Son noyau est implémenté essentiellement en langage C et FORTRAN, ses versions sont distribuées sous la forme de codes sources binaires à compiler (UNIX et Linux) ou d'exécutables précompilés (Windows). Les fichiers d'installation sont disponibles à partir du site web du CRAN, ce site répertorie également une importante source de documentation pour l'installation et l'utilisation du logiciel sur chaque système d'exploitation.

R est un langage de programmation interactif et orienté objet ce qui signifie que les variables, les données, les fonctions, les résultats sont stockés dans la mémoire de l'ordinateur sous forme d'objets qui ont chacun un nom.

C′est également un langage interprété c'est-à-dire non compilé. Les commandes entrées au clavier sont directement exécutées et contrairement à la plupart des langages informatiques comme C, FORTRAN ou encore JAVA, la construction d'un programme complet n'est pas nécessaire.

Cette propriété permet d'évaluer rapidement la qualité des algorithmes et de les débuguer. Cependant, l'exécution d'un tel programme peut être plus couteuse en temps machine qu'un programme équivalent compilé.

Enfin, la dernière particularité de ce logiciel est d'être, comme Matlab ou Scilab, un langage évolué basé sur le calcul matriciel et la manipulation simple d'objets complexes (listes, dataframe, etc.). Sa simplicité d'utilisation permet de programmer rapidement des algorithmes évolués. Initialement dédié à la statistique, ce langage est maintenant suffisamment puissant (et précis) pour le calcul scientifique et l'ingénierie mathématique (domaine de prédilection de Matlab). Il constitue aujourd'hui un langage de programmation intégré d'analyse statistique.
[http://www.r-project.org/,http://fr.wikipedia.org/wiki/R_(logiciel),34]

6.1.2 Descriptif des fonctions utilisées

Dans ce paragraphe, sont présentées les fonctions qui ont été choisies pour mettre en application sous R les méthodes décrites dans la section 4.2. Les descriptions détaillées de leur utilisation sont disponibles sur le site web du CRAN.

Package lars :

- La fonction **lars** a été utilisée pour mettre en place la méthode LARS.

Package pls :

- La fonction **plsr** a été utilisée pour mettre en place la méthode PLS
- La fonction **predict** a été utilisée pour obtenir la prédiction de la régression PLS

Package survival :

- La fonction **surv** a été utilisée pour créer un objet de survie à partir de la variable réponse Y.
- La fonction **survfit** a été utilisée pour créer des courbes de survie (Kaplan-Meier), elle utilise comme argument d'entrée un objet de survie.
- La fonction **resid** a été utilisée pour calculer les résidus de la déviance et les résidus de martingale.
- La fonction **coxph** a été utilisée pour construire un modèle de Cox

- La fonction **cox.zph** a été utilisée pour vérifier l'hypothèse des hasards proportionnels de Cox

- La fonction **predict** a été utilisée pour obtenir la prédiction du modèle de Cox (construit avec le « training set ») et ainsi construire les courbes AUC avec le « test set »

Package survivalROC :

- La fonction **survivalROC** a été utilisée pour élaborer les courbes de ROC et les grahiques AUC

6.2 Jeux de données publiques

Ce paragraphe du rapport a été écrit pour présenter le jeu de données publiques (dit « d'entraînement ») utilisé pour tester et comparer les différentes méthodes retenues dans le chapitre 4.

Ce jeu traitant de données sur le type le plus commun de lymphome (via des lymphochip, un type de biopuces différent du type affymetrix) est composé de 7399 variables (ce qui correspond à 4128 gènes) et un nombre total de 240 individus. Ce nombre total d'individus est divisé en 2 groupes, l'un de 160 patients qui va former le « training set », l'ensemble des observations utilisé dans le but de construire le modèle, et l'autre de 80 patients qui va former le « test set » , l'ensemble des observations utilisé dans le but de tester la qualité du modèle.

6.2.1 Analyse de survie :

La figure 8 représente l'estimation de la courbe de survie pour le modèle nul (c'est-à-dire sans aucune variable). La médiane de survie (la durée de suivi pour laquelle la moitié des individus sont décédés) est de 2,9 ans. A la fin de l'étude il reste 1 individu en vie (table II)

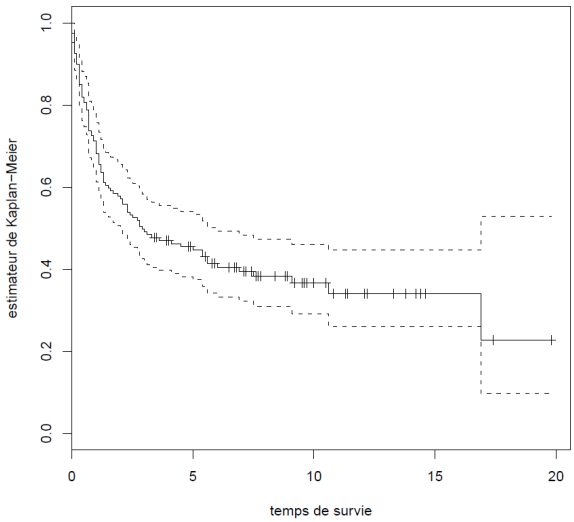

Figure 8 : courbe de survie (méthode de Kaplan-Meier)

Table II : analyse de survie du jeu de données

Nombre total d'individus	160
Médiane de survie (Intervalle de Confiance)	2,9 (2,1 – 6,0)
1er quartile (Intervalle de confiance)	0,7
3ème quartile (Intervalle de Confiance)	10,6
Nombre d'individus vivants à 5 ans	57 (35,6 %)
Nombre d'individus vivants à 10 ans	14 (8,7 %)
Nombres d'individus vivants à 15 ans	4 (2,5 %)
Nombres d'individus vivants à la fin de l'étude	1 (0,6 %)

6.2.2 Analyse transcriptomique

6.2.2.1 Analyse univariée

L'analyse univariéee est une méthode permettant de connaître le pouvoir prédictif de chaque gène (covariable) individuellement. Celle-ci pourrait être utilisée pour sélectionner (sur la base de la p-value associée au test de Wald sur le cœfficient de régression) les gènes les plus significatifs c'est-à-dire ceux qui ont le plus d'impact sur la variable de survie, 971 gènes significatifs ont été trouvés pour le jeu de données publiques étudié.

Le graphique de la figure 9 représente les p-values de toutes les variables, les plus significatives étant en haut (graphique du « -log »). Le trait rouge représente le seuil de significativité de 5 % et le trait vert représente celui de 0.01 %.

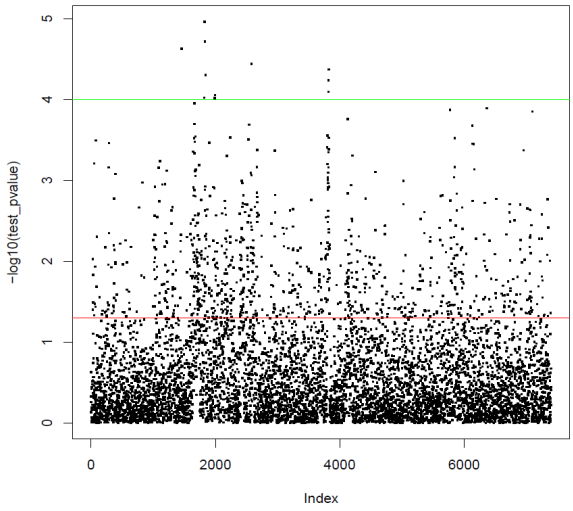

Figure 9 : représentation graphique de la p-value en fonction du nombre de covariables

Le tableau III présente le nombre de gènes qui sont significatifs à différents seuils, on voit que 11 gènes ont une p-value inférieure à 0.01 %, ils sont représentés par les points situés au-dessus du trait vert sur le graphique de la figure 9.

Table III : tableau des gènes les plus significatifs

p-value	**Nombre de gènes**
< 0.05	971
< 0.01	351
< 0.001	64
< 0.0001	11

Comme il l'a été dit dans le paragraphe 4.1 de ce rapport, la méthode de sélection univariée n'est pas pertinente car peut sélectionner des gènes qui sont significatifs de façon univariée mais qui sont fortement corrélés et qui ne seront pas désignés comme significatifs par le modèle de Cox. Ceci a été vérifié dans le jeu de données étudié : parmi les 11 gènes qui sont les plus significatifs (p-value<0.0001) de façon univariée, seul un a été désigné comme significatif par le modèle de Cox.

6.2.2.2 Comparaison des méthodes prédictives qui semblent les plus pertinentes

Il a été décidé de comparer (en utilisant l'AUC) les performances prédictives de 4 méthodes (basées sur les méthodes retenues au chapitre 4) de réduction de dimension. Ces 4 méthodes sont :

- La régression PLS standard (PLS)
- la méthode de Lars-Lasso utilisant la déviance résiduelle à la place du temps de survie comme variable réponse (LARSDR)
- La régression PLS utilisant la déviance résiduelle à la place du temps de survie comme variable réponse (PLSDR)
- La régression PLS utilisant la déviance résiduelle à la place du temps de survie comme variable réponse et utilisant la matrice « kernel » K (définie à partir du noyau gaussien) à la place de la matrice X des covariables (KPLSDR)

6.2.3 Résultats:

6.2.3.1 Nombre de variables séléctionnées

Au niveau de la méthode LARSDR, la figure 10 montre que plus on a de variables, plus l'indice du Cp de Mallows est petit, et donc meilleur est le modèle. On constate sur le graphique de la figure 10 que le nombre optimal de variable à sélectionner pour notre modèle peut être de 40, 70 ou 80 (Cp le plus petit).
Pour des raisons de temps le nombre de composantes optimal n'a pas été utilisé. Le nombre de composantes sélectionnées pour tester la méthode LARSDR sur le jeu de données publiques a été de 6 car c'est le nombre de composantes utilisées par P.Bastien dans son article, dont je parle dans le paragraphe 6.2.3.4, dans le but de comparer les quatre méthodes présentées dans le paragraphe 6.2.2.2.

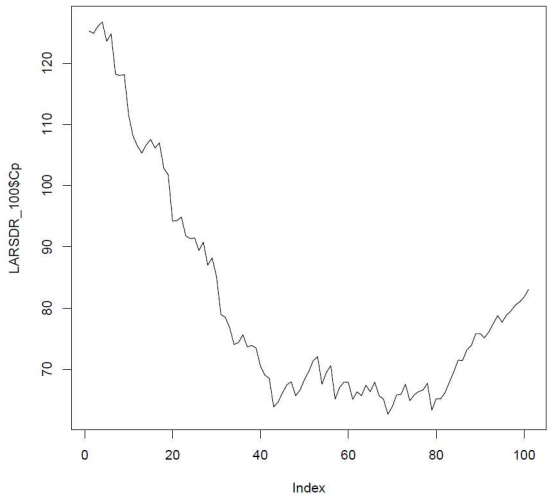

Figure 10 : graphique du C_p de Mallow en fonction du nombre de composantes.

Remarque : Les 4 premières variables sélectionnées par la méthode LARSDR font partis des 11 gènes les plus significatifs trouvés par sélection univariée dans le paragraphe 6.2.2.1.
De plus, toutes les variables sélectionnées par la méthode LARSDR sont significatives au seuil de 5%.

Au niveau des méthodes utilisant la régression PLS, selon le critère du Q^2 aucune composante valide n'a été retenue : dès la première composante le Q^2 était inférieur à 0.0975.

6.2.3.2 Courbes de survie

La figure 11 permet de voir que les modèles élaborés avec les différentes méthodes correspondent bien aux données utilisées pour le « training set » pour le jeu de données publiques étudié.

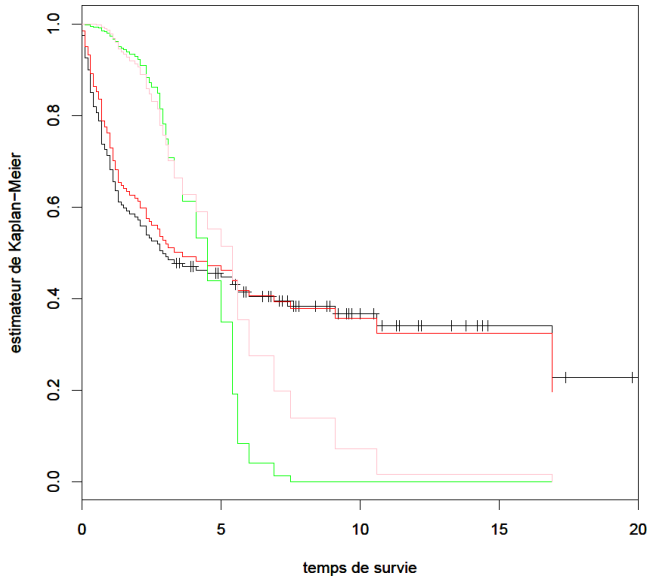

Figure 11 : courbes de survie de Kaplan-Meier (____ Cox-PLSDR , ____ Cox-PLS , ____ Cox-LARSDR, ____ modèle nul)

6.2.3.3 Courbes des AUC

La figure 12 montre, en représentant l'aire sous la courbe ROC, que la méthode de réduction de dimension PLSDR appliquée au modèle de Cox donne un bon pouvoir prédictif au modèle pour le jeu de données publiques étudié, en effet l'aire sous la courbe ROC pour cette méthode est celle qui est la plus élevée (environ 0.7, courbe rose de la figure 12). Ces courbes ont été tracées en utilisant les données du « test set ».

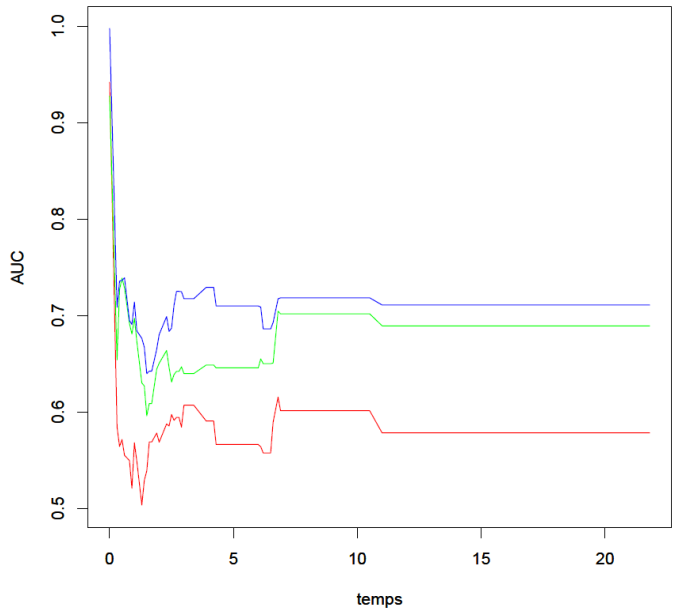

Figure 12 : AUC en fonction du temps (___ Cox-PLSDR, ___ Cox-PLS, ___ Cox-LARSDR)

6.2.3.4 Conclusion de l'analyse du jeu de données publiques

L'étude de ce jeu de données publiques a révélé (graphique 12) que la méthode la plus pertinente à appliquer au modèle de Cox dans le but de bien prédire la survie dans le cas de ce jeu de données est la méthode PLSDR.

Philippe Bastien, chercheur chez l'Oréal avait effectué la même comparaison sur le même jeu de données dans un article [27], écrit dans sa thèse intitulée « Régression PLS et données censurées ». L'auteur avait obtenu les mêmes résultats dans le sens ou la méthode LARSDR (avec un AUC compris entre 0.55 et 0.6 environ) était dans son cas la moins prédictive. Selon lui, la méthode la plus pertinente était la méthode KPLSDR (avec un AUC d'environ 0.7). Par manque de temps cette méthode n'a pu être finalisée au cours de ce stage et le sera par la suite.

Le jeu de données fourni par Transgene sera exploité et analysé en lui appliquant les quatre méthodes présentées dans le paragraphe 6.2.2.2. En effet, il semble que la qualité prédictive d'un modèle soit fortement dépendante du jeu de données utilisé.

Chapitre 7: Conclusion

Le travail effectué à Transgene :

Au cours de ce stage, un travail de bibliographie a permis de trouver différentes méthodes pertinentes de réduction de la dimensionnalité : la régression PLS et la méthode LARS.
Quatre méthodes (LARSDR, PLS, PLSDR et KPLSDR) ont été testées sur un jeu de données publiques, et une s'est révélée la plus pertinente : la méthode PLSDR.
L'analyse de ces méthodes a été rendue complexe par la recherche de différents critères les qualifiant (le noyau gaussien, la déviance résiduelle, le critère du C_p de Mallows, l'indicateur de bonne prédiction du modèle Q^2 , etc)

Les résultats d'une méthode appliquée sur un jeu de données dépendant de la structure du jeu, les 4 méthodes testées sur le jeu de données publiques dans le paragraphe 6.2 seront appliquées sur le jeu de données de Transgene. La comparaison mettra en avant une méthode pertinente qui sera utilisée pour fournir une liste de gènes permettant de prédire la survie des patients relativement au traitement.

Les acquis du stage :

Ce stage a été pour moi une expérience très enrichissante car, j'ai pu d'une part, étudier deux thématiques différentes des statistiques qui sont la réduction de la dimensionnalité et l'analyse de survie, et d'autre part découvrir une application biologique de ces dernières, comprennent l'utilisation des biopuces pour découvrir, via leur profil transcriptomique, l'influence d'un traitement sur des patients.

Bibliographie

[1] **Emmanuel Villar.** *Apport des méthodes récentes de modélisation de survie dans le contexte spécifique des patients dialysés (thèse).* 2007

[2] **Philippe Bastien**. *PLS-Cox model: application to gene expression.* 2004

[3] **A.A Alizadeh, M.B Eisen, R.E Davis, C Ma, I.S Lossos, A Rosenwald, J.C Boldrick, H Sabet, T Tran, X Yu, J.I Powell, L Yang, G.E Marti, T Moore, J Hudson, L Lu, D.B Lewis, R Tibshirani, G Sherlock, W.C Chan, T.C Greiner, D.D Weisenburger, J.O Armitage, R Warnke, R Levy, W Wilson, M.R Grever, J.C Byrd, D Botstein, P.O Brown, L.M Staudt.** *Distinct types of diffuse large B-Cell lymphoma identified by gene expression.* 2000

[4] **P Royston, D.G Altman, W Sauerbrei.** *Dichotomizing continous predictors in multiple regression: A bad idea*. 2006

[5] **Peter J Park.** *Gene expression data and survivals analysis.* Methods of Microarray data analysis IV, p 21-31, 2005

[6] **T Hastie, R Tibshirani, D Botstein, P Brown**. *Supervised harvesting of expression trees.* 2001

[7] **T.K Jenssen, W.P Kuo, T Stokke, E Hovig.** *Associations between gene expressions in beast cancer and patient survival*. 2002

[8] **A.L Boulesteix, .** *WilcoxCV : an efficient R package for variable selection in cross validation*. 2006

[9] **A Statnikov, C.F Aliferis, I Tsamardinos, D Hardin, S Levy.** *A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis*. 2005

[10] **Severine Vancolen.** *La regression PLS*. 2005

[11] **Y Pawitan, J bjohle, S Wedren, K Humphreys, L Skoog, F Huang, L Amler, P Shaw, P Hall, J Bergh.** *Gene expression profiling for prognosis using Cox regression.* 2004

[12] **T Hastie, R Tibshirani.** *Efficient quadratic regularization for expression arrays* 2004

[13] **H.C Van Houwelingen, T Bruinsma, A.A.M Hart, L.J Van't Veer, L.F.A Wessels.** *Cross-validated Cox regression on microarray gene expression data.* 2006

[14] **T Horthon, A Benner, B Lausen, M Radespiel-Tröger.** *Bagging survival trees.* 2004

[15] **T Horthon, P Bühlmann, S Dudoit, A Molinaro, M Van der Laan**. *Survival ensembles.*2006

[16] **C Duby, S Robin.** *Analyse en Composantes Principales (Institut National Agronomique Paris – Grignon).* 2006

[17] **Eric Bair, Trevor Hastie, Debashis Paul and Robert Tibshirani.** *Prediction by supervised principal components.* 2005

[18] **Eric Bair and Robert Tibshirani.** *Semi-supervised methods to predict patient survival from gene expression data* . 2004

[19] M.Y **Park, T Hastie.** *L_1 regularisation path algorithm for genralized linear models.* 2006

[20] **Julien Chiquet.** *Méthodes Lasso pour la sélection de variables en regression linéaire.* 2009

[21] **H Zou, T Hastie.** *Regularization and variable selection via the elastic net.* 2005

[22] **Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani**. *Least Angle Regression*. 2003

[23] **Jiang Gui, Hongzhe Li.** *Penalized Cox Regression Analysis in the High-Dimensional and Low-sample Size Settings, with Applications to microarray gene expression data.* 2004

[24] **Mark R. Segal.** *Microarray gene expression data with linked survival phenotypes: diffuse large-B-cell lymphoma revisited.* 2005

[25] **Philippe Bastien.** *Régression PLS et données censurées (thèse).* 2008

[26] **Danh.V.Nguyen, David. M.Rocke**. *Partial Least Squares proportional hazard regression for application to DNA microarray survival data.* 2002

[27] **Philippe Bastien.** *Deviance residuals based PLS regression for censored data in high dimensional setting.* 2007

[28] **P Bastien, E Vinzi, M tenenhaus.** *PLS generalized linear regression*. 2005

[29] **Bjorn-Helge Mevik, Ron Wehrens.** *The pls Package: Principal Component and Partial Least Squares Regression in R.*2007

[30] **A.L Boulesteix, K Stimmer.** *Partial Least Squares: a versatile tool for the analysis of high-dimensional genomic data*. 2007

[31] **Catherine Huber.** *Modèles pour des durées de survie*

[32] **Jean-Pierre Gauchi, Pierre Chagnon.** *Comparison of selection methods of explanatory variables in PLS regression with application to manufacturing process data.* 2001

[33] **Patrick Heagerty and Yingye Zheng .** *Survival Model Predictive Accuracy and ROC Curves.* 2003

[34] **Denis Puthier.** *Introduction au logiciel d'analyse statistique R.* 2009

Annexe A

La régression en composante principale (PCR)

La PCR (Principal Component Regression) est une technique quantitative de décomposition spectrale étroitement liée à la régression PLS (il s'agit de construire un petit nombre de variables dites « variables latentes » ou « composantes principales » qui sont des combinaisons linéaires des variables d'origine), cependant, dans la PCR le calcul de la matrice W des poids (matrice telle que T=XW comme dans la PLS) est fait d'une manière légèrement différente.
En effet, au lieu d'utiliser l'information de la variable réponse Y en même temps que le processus de décomposition comme dans la PLS, la PCR décompose d'abord la matrice des covariables X en un ensemble de vecteurs propres et de scores, puis les régressent contre Y dans une étape séparée.

Pour réaliser une PCR, on fait une ACP (Analyse en Composantes Principales) [18] d'ordre k du tableau X (dont les colonnes doivent être centrées pour avoir une moyenne nulle) qui va donner un tableau T_{n*k} , le tableau des composantes principales tel que les nouvelles variables (les colonnes du tableau) sont centrées , orthogonales entre elles et ne sont pas corrélées.
Ensuite on effectue la régression

$$\hat{Y} = PY + \varepsilon \text{ avec } P = T(T'T)^{-1}T'$$

et on retrouve bien un modèle linéaire en les variables $X_1 \ldots X_p$.

Les composantes principales (colonnes de la matrice T) sont caractérisées par leur faible variance (ou dispersion) : c'est le critère du minimum de la variance, on a : $\text{var}(T^j) = \lambda_j(X'X)$, où λ_j est la j$^{\text{ième}}$ valeur propre de la matrice X'X.

T^j , la j-ième composante principale est donnée par : $T^j = XW^j$
avec W la matrice des vecteurs propres de la matrice X'X (vecteurs propres rangés par ordre décroissant des valeurs propres associées) : $X'XW^J = \lambda_j W^j$
C'est-à-dire que T^j est une combinaison linéaire des colonnes de X.

L'inconvénient majeur de la PCR est que les 1ères composantes principales ne sont pas forcement celles qui expliqueront le mieux la réponse Y. En effet, la PCR orthogonalise d'abord la matrice X'X régresse ensuite sur la variable réponse Y, qui n'est pas utilisée pour sélectionner les composantes principales à conserver.

Annexe B

La méthode LARS

Depuis une dizaine d'années, afin de permettre la prise en compte d'information provenant de très nombreux descripteurs, et de nombreux développements sont apparus autour du modèle de Cox dans des approches régularisées.

La régularisation peut s'exprimer comme une réduction de la dimensionnalité comme c'est le cas avec la régression PLS, il en résulte alors des coefficients de régression biaisés mais à plus faible variance. La régularisation peut aussi s'exprimer par maximisation de la log vraisemblance partielle avec des contraintes de type L1 sur les coefficients : c'est le principe de la méthode du LASSO (Least Absolute Shrinkage and Selection Operator), une technique utilisée pour l'estimation et la sélection de modèles en régression linéaire. Elle a la particularité de produire des solutions favorisant l'interprétation en exhibant un petit nombre de variables explicatives.

C'est une version pénalisée du problème des moindres carrés ordinaires, elle estime un modèle de régression linéaire multiple comme la solution du problème suivant :

$$\hat{\beta} = \arg\min(\sum_{i=1}^{n}(Y_i - X_i\beta)^2).$$

La méthode LASSO impose une borne supérieure pour la somme des valeurs absolues des coefficients de régression et certains de ces coefficients sont rétrécis vers zéro :

$$\|\beta\|_1 = \sum_{j=1}^{p}|\beta_j| \le t \text{ avec } \beta = (\beta_1 ... \beta_p)' \text{ et t le paramètre de réglage}$$

Une formulation équivalente de LASSO est donnée par la solution du problème des moindres carrés pénalisés :

$$\hat{\beta} = \arg\min(\sum_{i=1}^{n}(Y_i - X_i\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

il y a équivalence en ce sens que, pour, il existe un $t \ge 0$ tel que les deux problèmes ont la même solution, et vice versa.

La méthode LASSO permet d'obtenir un modèle robuste mais son inconvénient est qu'elle ne donne pas une estimation précise des coefficients du modèle lorsque le nombre de variables est très grand devant le nombre d'individus.

L'utilisation d'une procédure appelée LAR (Least Angle Regression) en association avec la régression LASSO va donner la méthode LARS qui est une méthode très efficace en temps de calcul et qui permet par simple modification d'étendre la méthode LASSO aux problèmes de très grande dimension.

Description de l'algorithme du LARS :

La méthode va calculer $\hat{\mu} = X\hat{\beta}$ le vecteur de prédiction. L'algorithme commence avec tous les coefficients de régression β_j nuls : $\hat{\mu}^0 = 0$ il s'agit alors de trouver le prédicteur X_j le plus corrélé avec les résidus (Y étant donné que les coefficients du modèle sont tous nuls à la 1ère étape).
On calcule $\hat{c} = c(\hat{\mu}) = X'.(Y - \hat{\mu})$ ($\hat{c} = X'Y$ dans le cas de la 1ère étape)
$\hat{j} = \arg\max|\hat{c}_j|$ est l'indice de la covariable la plus corrélée avec Y.
Ensuite, LARS augmente la valeur de $\hat{\mu}$ dans la direction de X_j (jusqu'à trouver un autre prédicteur X_k) :

$$\hat{\mu}^1 = \hat{\mu}^0 + \hat{\gamma}^1 X_1$$

.
Regardons l'algorithme d'une manière plus générale pour voir la définition du pas de descente γ (élément principal de la méthode) : supposons que l'on vient d'estimer $\hat{\mu}_A$ on calcule donc

$$\hat{c} = X'(Y - \hat{\mu}_A)$$

A représente l'ensemble des indices correspondant aux variables qui possèdent la meilleure corrélation avec les résidus, elles sont appelées « variables actives » :

$$\hat{C} = \max\left\{|\hat{c}_j|\right\} \text{ et } A = \left\{j : |c_j| = \hat{C}\right\}$$

Définissons des éléments utiles pour calculer le pas de descente γ :

la matrice $X_A = (...s_j X_j ...)_{j \in A}$ où s_j est le signe de $\hat{c}_i$ pour $j \in A$ (s_j est égal à ± 1), $G_A = X'_A X_A$ et $A_A = (1'_A G^{-1}{}_A 1_A)^{-1/2}$ avec 1_A le vecteur de coordonnées 1 et de taille le cardinal de A
$u_A = X_A w_A$ où $w_A = A_A G^{-1}{}_A 1_A$ est la direction de descente et u_A est tel que $X'_A u_A = A_A 1_A$ et $\|u_A\|^2 = 1$, $a \equiv X'u_A$.

La suite de l'algorithme du LARS calcule :

$\hat{\mu}_{A+} = \hat{\mu}_A + \gamma u_A$ avec $\hat{\gamma} = \min_{j \notin A}\left\{\frac{\hat{C} - \hat{c}_j}{A_A - a_j}, \frac{\hat{C} + \hat{c}_j}{A_A + a_j}\right\}$ qui conserve l'équicorrelation.

La méthode possède aussi une règle qui permet de sortir de la sélection une variable qui est entré précédemment.
Ainsi, supposons que l'on vient de finir une étape de l'algorithme du LARS : on a l'ensemble *A*, le vecteur de précision $\hat{\mu}_A$, le pas de descente $\hat{\gamma}$ et la direction de descente w_A.

On définit $\hat{d}$ comme $s_j w_{A_j}$ pour $j \in A$ et 0 sinon. En augmentant μ $(\mu(\gamma) = \hat{\mu}_A + \gamma u_A)$ on a

$$\mu(\gamma) = X\beta(\gamma) \text{ où } \beta_j(\gamma) = \hat{\beta}_j + \gamma \hat{d}_j \text{ pour } j \in A$$

$$\beta_j(\gamma) \text{ changera de signe lorsque } \gamma_j = \frac{-\hat{\beta}_j}{\hat{d}_j}$$

le premier de ces changements se produisant à $\tilde{\gamma} = \min\{\gamma_j\}$ pour $\gamma_i > 0$ ($\tilde{\gamma} = \infty$ lorsqu'aucun γ_j n'est strictement positif) ce qui correspond à la covariable $x_{\tilde{j}}$. Si $\tilde{\gamma} < \hat{\gamma}$, $\beta_j(\gamma)$ ne peut pas être une solution de la méthode Lasso pour $\gamma > \tilde{\gamma}$ puisque la condition $s_j = sign(\hat{\beta}_j) = sign(\hat{c}_j)$ n'est plus vérifiée : $\beta_j(\gamma)$ a changé de signe tandis que $c_j(\gamma)$ non.

Si $\tilde{\gamma} < \hat{\gamma}$ on arrête l'étape en cours, $\gamma = \tilde{\gamma}$ on supprime $\tilde{j}$ de l'ensemble A et on calcule le vecteur de précision μ : $\hat{\mu}_{A+} = \hat{\mu}_A + \tilde{\gamma} u_A$ et ainsi $A = A - \{\tilde{j}\}$

Annexe C

La régression des moindres carrés partielle (PLSR)

Cette méthode itérative consiste à remplacer la matrice initiale $X_{n,p}$ (avec n le nombre d'individus, p le nombre de variables, et n < p) par une nouvelle matrice T (dérivée de X) comprenant le même nombre de lignes n, mais avec un nombre de colonnes (les composantes) k très inférieur à p.

T=XW avec W la matrice dite « des poids », toute la difficulté de la méthode est de déterminer W pour pouvoir calculer la matrice des scores T.

Les facteurs de la régression PLS (c'est-à-dire les colonnes de la matrice T, dites « variables latentes ») sont des vecteurs propres de la matrice Y'XX'Y (avec la matrice X qui est centrée réduite, pour ne pas mettre d'intercept dans le modèle), ce sont des combinaisons linéaires des variables d'origine, et elles sont orthogonales les unes avec les autres

On construit la 1ère composante : $T_1 = \sum_{k=1}^{p} w_k^1 X_k$

Avec $w_k^1 = \frac{\text{cov}(X_k, Y)}{(\sum_{l=1}^{p} \text{cov}(X_l, Y)^2)^{1/2}}$

Puis on fait une régression simple de la variable réponse Y sur T_1 : $Y = c_1 T_1 + \varepsilon_Y^1$

Si le pouvoir explicatif de cette régression est satisfaisant on s'arrête là, mais si le pouvoir explicatif est trop faible on réitère la procédure et on construit alors la composante suivante T_2.

Afin de calculer T_2 on commence par calculer les résidus $\varepsilon_X^{1,k}$ des régressions linéaires simples des variables explicatives X_k (pour k=1…p) sur T_1 et on construit T_2 comme combinaison linéaire de ces résidus : $T_2 = \sum_{k=1}^{p} w_k^2 . \varepsilon_X^{1,k}$

Avec $w_k^2 = \frac{\text{cov}(\varepsilon_X^{1,k}, \varepsilon_1)}{(\sum_{j=1}^{p} \text{cov}(\varepsilon_X^{1,l}, \varepsilon_1)^2)^{1/2}}$

On effectue ensuite une régression de Y sur T_1 et T_2 : $Y = c_1 T_1 + c_2 T_2 + \varepsilon_Y^2$

(c_1 est le même coefficient que dans la régression de Y sur T_1 en raison de l'orthogonalité des composantes T_1 et T_2).

Tant que le pouvoir explicatif de la régression linéaire n'est pas suffisant on peut poursuivre en utilisant de la même manière les résidus ε_Y^2 et $\varepsilon_X^{2,k}$ des régressions respectives de Y et de $X_1 \dots X_p$ sur T_1 et T_2.

Ainsi à la t^ième^ itération, on obtient la t^ième^ composante : $T_t = \sum_{k=1}^{p} w_k^t . \varepsilon_X^{t-1,k}$

Ou encore sous forme matricielle : $T_t = \varepsilon_X^{t-1} . W^t$

Avec $W^t = \frac{(\varepsilon_X^{t-1})' \varepsilon_Y^{t-1}}{\|(\varepsilon_X^{t-1})' \varepsilon_Y^{t-1}\|}$ où $((\varepsilon_X^{t-1})' \varepsilon_Y^{t-1})_k = Cov(\varepsilon_X^{t-1,k}, \varepsilon_Y^{t-1})$

On effectue ensuite une régression de Y sur $T_1 \dots T_t$: $Y = c_1 T_1 + \cdots + c_t T_t + \varepsilon_Y^t$

Tant que le pouvoir explicatif de la régression linéaire n'est pas suffisant on poursuit en effectuant la régression linéaire de chaque variable X_k (k = 1…p) sur T_m (m = 1…t) :

$$X_k = a_k^1 T_1 + \cdots + a_k^t T_t + \varepsilon_X^{t,k}$$

On réitère la procédure jusqu'à juger que le pouvoir explicatif de la régression obtenu est suffisant. L'idée de la PLS est de donner le plus d'information possible sur Y dans les 1ères composantes principales construites.

Annexe D

Programme

```
############# Importation du jeu de données

staudt.x <-
matrix(scan("/data/home/deb/Data_DeB/Rosenwald/staudt.x"),ncol=240,byrow=T)
# données 240 patients (sans names)

staudt.tim <- scan("/data/home/deb/Data_DeB/Rosenwald/staudt.tim")   #
données avec les 240 patients, sans names

staudt.status <- scan("/data/home/deb/Data_DeB/Rosenwald/staudt.status")
# données avec les 240 patients, sans names

staudt.train.indices <-
scan("/data/home/deb/Data_DeB/Rosenwald/staudt.train.indices")   # training
set



# modification des données car on veut faire les essais avec le training
set (160 variables), on réduit le nombre de patients à  160 et on attribue
des names

colnames(staudt.x) <- 1:240

staudt.x.bis <- staudt.x[,c(staudt.train.indices)]

rownames(staudt.x.bis) <- rownames(genes_rosen)

names(staudt.tim) <- 1:240

staudt.tim.bis <- staudt.tim[c(staudt.train.indices)]

names(staudt.status) <- 1:240

staudt.status.bis <- staudt.status[c(staudt.train.indices)]


# modification des données pour avoir le test set (80 variables)

staudt.x.bis2 <- staudt.x[,-c(staudt.train.indices)]

rownames(staudt.x.bis2) <- rownames(genes_rosen)

staudt.tim.bis2 <- staudt.tim[-c(staudt.train.indices)]

staudt.status.bis2 <- staudt.status[-c(staudt.train.indices)]
```

```
# on sauvegarde le jeu de données pour le "training set" (création d'une
liste, 160 individus)

rosen1 <-
c(list(x=scale(t(staudt.x.bis))),list(y=scale(staudt.tim.bis)),list(z=staud
t.status.bis))   # c'est une liste

save("rosen1",file="Data_Rosen1_DeB.RData")

# on sauvegarde le jeu de données pour le "test set" (création d'une liste,
80 individus)

rosen2 <-
c(list(t=scale(t(staudt.x.bis2))),list(u=scale(staudt.tim.bis2)),list(v=sca
le(staudt.status.bis2)))

save("rosen2",file="Data_Rosen1_DeB.RData")



# on considère maintenant la matrice des gènes rosen1$x (données centrées
réduites), le temps de survie rosen1$y (données centrées réduites) et le
status rosen1$z

# toutes les données sont importées, on va pouvoir (sur le training set de
160 variables) comparer les 4 méthodes (article de P.Bastien) :


# CALCUL DES RESIDUS DE LA DEVIANCE / la déviance résiduelle (pour les 160
patients)

survie_rosen <- Surv(staudt.tim.bis,staudt.status.bis)   # création de
l'objet de survie / données non centrées-réduites

fct_survie_rosen <- survfit(survie_rosen~1)   #  fonction de survie pour le
modele nul

modele_nul_rosen <- coxph(survie_rosen~1)  # modèle nul

deviance_residuals_rosen <- resid(modele_nul_rosen,type="deviance")  # (les
fonctions resid et residuals donnent la même chose) calcul des résidus de
la déviance pour le modèle nul (vecteur de taille 160, car c'est pour
chaque individu)...



survie_rosen_centre <- Surv(rosen1$y,rosen1$z) ## calcul de l'objet de
survie pour le temps centré réduit

fct_survie_rosen_centre <- survfit(survie_rosen_centre~1)



######## méthode de lars-lasso (avec temps de survie, avec déviance
résiduelle) LARSDR  6 1ers gènes  (les données doivent être centrées et
réduites)
```

```
LARSDR <- lars(rosen1$x
,scale(deviance_residuals_rosen),max.steps=6,use.Gram=FALSE)  # ...on va
prendre cela a la place du temps de survie dans l'appel du lars

LARS <- lars(rosen1$x ,rosen1$y,max.steps=6,use.Gram=FALSE)  # temps de
survie Ã  la place de la déviance residuelle

matrice_larsdr<-rosen1$x[,names(unlist(LARSDR$actions))]

COX_LARSDR <- coxph(survie_rosen ~ matrice_larsdr)



######## Méthode PLS standard (pour model cox-pls)   7 1eres composantes


PLS_240 <-
plsr(staudt.tim~t(staudt.x),ncomp=7,method=pls.options()$plsralg) # avec
240 patients

PLS <- plsr(staudt.tim.bis~t(staudt.x.bis),ncomp=7,data=rosen1) # avec 160
patients

COX_PLS <- coxph(survie_rosen ~ PLS$scores,data=rosen1)



######## Méthode PLS avec déviance résiduelle à la place du temps de survie
PLSDR  7 1eres composantes

PLSDR <- plsr(deviance_residuals_rosen ~
t(staudt.x.bis),ncomp=7,method=pls.options()$plsralg, data=rosen1)

COX_PLSDR <- coxph(survie_rosen ~ PLSDR$scores ,data=rosen1)



########################  COURBE DE KAPLAN MEIER (modèle nul) pour
illustrer le rapport de stage

pdf("km.pdf")

fct_survie_rosen <- survfit(survie_rosen~1)   #  fonction de survie pour le
modèle nul

plot(fct_survie_rosen,conf.int=TRUE,xlab="temps de survie",ylab="estimateur
de Kaplan-Meier")

dev.off()



########################  VERIFICATION DES HYPOTHESES DU MODELE DE COX

#### HYPOTHESE des hasards proportionnels

hp.COX_LARSDR <- cox.zph(COX_LARSDR, global=TRUE)
print(hp.COX_LARSDR)
par(mfrow=c(2,3))
plot(hp.COX_LARSDR, var=1)
```

```
plot(hp.COX_LARSDR, var=2)
plot(hp.COX_LARSDR, var=3)
plot(hp.COX_LARSDR, var=4)
plot(hp.COX_LARSDR, var=5)
plot(hp.COX_LARSDR, var=6)

# on regarde si le graphe de log(-log(S(t))) en fonction de log(t) est bien
linéaire:

fct_survie_rosen_LARSDR <- survfit(COX_LARSDR)

plot(log(fct_survie_rosen_LARSDR$time),log(-
log(fct_survie_rosen_LARSDR$surv)))


hp.COX_PLS <- cox.zph(COX_PLS, global=TRUE)
print(hp.COX_PLS)
par(mfrow=c(2,4))
plot(hp.COX_PLS, var=1)
plot(hp.COX_PLS, var=2)
plot(hp.COX_PLS, var=3)
plot(hp.COX_PLS, var=4)
plot(hp.COX_PLS, var=5)
plot(hp.COX_PLS, var=6)
plot(hp.COX_PLS, var=7)

fct_survie_rosen_PLS <- survfit(COX_PLS)

plot(log(fct_survie_rosen_PLS$time),log(-log(fct_survie_rosen_PLS$surv)))


hp.COX_PLSDR <- cox.zph(COX_PLSDR, global=TRUE)
print(hp.COX_PLSDR)
par(mfrow=c(2,4))
plot(hp.COX_PLSDR, var=1)
plot(hp.COX_PLSDR, var=2)
plot(hp.COX_PLSDR, var=3)
plot(hp.COX_PLSDR, var=4)
plot(hp.COX_PLSDR, var=5)
plot(hp.COX_PLSDR, var=6)
plot(hp.COX_PLSDR, var=7)

fct_survie_rosen_PLSDR <- survfit(COX_PLSDR)

plot(log(fct_survie_rosen_PLSDR$time),log(-
log(fct_survie_rosen_PLSDR$surv)))


#### HYPOTHESE de la log-linéarité (on trace les résidus de martingale / la
martingale résiduelle en fonction de chaque covariable et on regarde si
c'est linéaire)


par(mfrow=c(2,3))

res_martingale.COX_LARSDR <- resid(COX_LARSDR, type="martingale")   #
résidus de la martingale du modèle, vecteur de longueur le nombre de
patients
```

```
X_LARSDR <-
cbind(rosen1$x[,1825],rosen1$x[,2579],rosen1$x[,1456],rosen1$x[,1994],rosen
1$x[,6134],rosen1$x[,7098])

b_LARSDR <- coef(COX_LARSDR)[c(1,2,3,4,5,6)]

for (j in 1:6) {
plot(X_LARSDR[,j],b_LARSDR[j]*X_LARSDR[,j] + res_martingale.COX_LARSDR,
xlab=c("rosen1$x[,1825]","rosen1$x[,2579]","rosen1$x[,1456]","rosen1$x[,199
4]","rosen1$x[,6134]","rosen1$x[,7098]")[j],ylab="component+residuals")

abline(lm(b_LARSDR[j]*X_LARSDR[,j] + res_martingale.COX_LARSDR ~
X_LARSDR[,j]), lty=2)

lines(lowess(X_LARSDR[,j],b_LARSDR[j]*X_LARSDR[,j] +
res_martingale.COX_LARSDR, iter=0))
                }

dev.new()


par(mfrow=c(2,4))

res_martingale.COX_PLS <- resid(COX_PLS, type="martingale")

X_PLS <-
cbind(PLS$scores[,1],PLS$scores[,2],PLS$scores[,3],PLS$scores[,4],PLS$score
s[,5],PLS$scores[,6],PLS$scores[,7])

b_PLS <- coef(COX_PLS)[c(1,2,3,4,5,6,7)]

for (j in 1:7) {
plot(X_PLS[,j],b_PLS[j]*X_PLS[,j] + res_martingale.COX_PLS,
xlab=c("PLS$scores[,1]","PLS$scores[,2]","PLS$scores[,3]","PLS$scores[,4]",
"PLS$scores[,5]","PLS$scores[,6]","PLS$scores[,7]")[j],ylab="component+resi
duals")

abline(lm(b_PLS[j]*X_PLS[,j] + res_martingale.COX_PLS~X_PLS[,j]), lty=2)

lines(lowess(X_PLS[,j],b_PLS[j]*X_PLS[,j] + res_martingale.COX_PLS,
iter=0))
                }

dev.new()


par(mfrow=c(2,4))

res_martingale.COX_PLSDR <- resid(COX_PLSDR, type="martingale")

X_PLSDR <-
cbind(PLSDR$scores[,1],PLSDR$scores[,2],PLSDR$scores[,3],PLSDR$scores[,4],P
LSDR$scores[,5],PLSDR$scores[,6],PLSDR$scores[,7])

b_PLSDR <- coef(COX_PLSDR)[c(1,2,3,4,5,6,7)]

for (j in 1:7) {
plot(X_PLSDR[,j],b_PLSDR[j]*X_PLSDR[,j] + res_martingale.COX_PLSDR,
xlab=c("PLSDR$scores[,1]","PLSDR$scores[,2]","PLSDR$scores[,3]","PLSDR$scor
```

```
es[,4]","PLSDR$scores[,5]","PLSDR$scores[,6]","PLSDR$scores[,7]")[j],ylab="
component+residuals")

abline(lm(b_PLSDR[j]*X_PLSDR[,j] + res_martingale.COX_PLSDR~X_PLSDR[,j]),
lty=2)

lines(lowess(X_PLSDR[,j],b_PLSDR[j]*X_PLSDR[,j] + res_martingale.COX_PLSDR,
iter=0))
                }


############################## Courbe de survie (training set) pour chacun
des modèles/méthodes (KM COMPARAISON)

pdf("courbes_km.pdf")
plot(fct_survie_rosen,conf.int=F,xlab="temps de survie",ylab="estimateur de
Kaplan-Meier",col="black",xlim=c(0,20))
par(new=TRUE)
plot(fct_survie_rosen_LARSDR,conf.int=F,xlab="temps de
survie",ylab="estimateur de Kaplan-Meier",col="red",xlim=c(0,20))
par(new=TRUE)
plot(fct_survie_rosen_PLS,conf.int=F,xlab="temps de
survie",ylab="estimateur de Kaplan-Meier",col="green",xlim=c(0,20))
par(new=TRUE)
plot(fct_survie_rosen_PLSDR,conf.int=F,xlab="temps de
survie",ylab="estimateur de Kaplan-Meier",col="pink",xlim=c(0,20))
dev.off()

########### programme p-value, graphes p-value et coefficients (SELECTION
UNIVARIEE) ##############################

pvalue_cox<-function(var)
{
      sortie<-coxph(Surv(rosen1$y,rosen1$z)~var)
      return(summary(sortie)$coefficients[,5])
}

test_pvalue<-apply(rosen1$x,2,pvalue_cox)

length(test_pvalue)

significant <-  test_pvalue[test_pvalue<0.05]

length(significant)

plot(test_pvalue,pch=".",cex=2)


plot(significant,pch=".",cex=2)


pdf("pvalue.pdf")

plot(-log10(test_pvalue),pch=".",cex=2)  # graphe -log pour faire un "zoom"
de la p-value et voir + clairement les différences

abline(h=-log10(0.05),col="red")
abline(h=-log10(0.0001),col="green")

dev.off()
```

```
coef_cox<-function(var)
{
sortie<-coxph(Surv(rosen1$y,rosen1$z)~var)
return(summary(sortie)$coefficients[,1])
}

test_coef<-apply(rosen1$x,2,coef_cox)

plot(test_coef,pch=".",cex=2)


################################## AUC se fait sur le test set

### LARSDR-AUC

matrice_larsdr <-rosen2$t[,names(unlist(LARSDR$actions))]

predict.test.set.larsdr <-
predict(COX_LARSDR,newdata=as.data.frame(matrice_larsdr),type="lp")

roc_auc_larsdr_test_set <-function(x)
{
survivalROC(as.integer(staudt.tim.bis2),as.integer(staudt.status.bis2),pred
ict.test.set.larsdr,entry=NULL, predict.time=x,span=0.05)$AUC
}

tps_larsdr <-sort(unique(as.vector(staudt.tim.bis2)))

vect_auc_larsdr <- sapply(tps_larsdr,roc_auc_larsdr_test_set)


### PLS-AUC


mat_pls <- t(staudt.x.bis)
PLSb <- plsr(staudt.tim.bis~mat_pls,ncomp=7)
mat_pls <- PLSb$scores
COX_PLSb <- coxph(survie_rosen ~ mat_pls)
mat_pls <- t(staudt.x.bis2)
pred_pls <- predict(PLSb,mat_pls)
pred_pls <- matrix(pred_pls,80,7)

mat_pls <- pred_pls
colnames(mat_pls) <- colnames(PLSb$scores)
row.names(mat_pls) <- row.names(t(staudt.x.bis2))
predict.test.set.pls <-
(predict(COX_PLSb,newdata=as.data.frame(mat_pls),type="lp"))

roc_auc_pls_test_set <-function(x)
{
survivalROC(as.integer(staudt.tim.bis2),as.integer(staudt.status.bis2),pred
ict.test.set.pls,entry=NULL, predict.time=x,span=0.05)$AUC
}

tps_pls <-sort(unique(as.vector(staudt.tim.bis2)))

vect_auc_pls <- sapply(tps_pls,roc_auc_pls_test_set)
```

```
### PLSDR-AUC

mat_plsdr <- t(staudt.x.bis)
PLSDRb <- plsr(deviance_residuals_rosen~mat_plsdr,ncomp=7)
mat_plsdr <- PLSDRb$scores
COX_PLSDRb <- coxph(survie_rosen ~ mat_plsdr)
mat_plsdr <- t(staudt.x.bis2)
pred_plsdr <- predict(PLSDRb,mat_plsdr)
pred_plsdr <- matrix(pred_plsdr,80,7)

mat_plsdr <- pred_plsdr
colnames(mat_plsdr) <- colnames(PLSDRb$scores)
row.names(mat_plsdr) <- row.names(t(staudt.x.bis2))
predict.test.set.plsdr <-
(predict(COX_PLSDRb,newdata=as.data.frame(mat_plsdr),type="lp"))

roc_auc_plsdr_test_set <-function(x)
{
survivalROC(as.integer(staudt.tim.bis2),as.integer(staudt.status.bis2),pred
ict.test.set.plsdr,entry=NULL, predict.time=x,span=0.05)$AUC
}

tps_plsdr <-sort(unique(as.vector(staudt.tim.bis2)))

vect_auc_plsdr <- sapply(tps_plsdr,roc_auc_plsdr_test_set)


pdf("courbes_auc.pdf")
plot(tps_larsdr,vect_auc_larsdr, type="l",ylim=c(0.5,1),col="red",
xlab="temps",ylab="AUC")
par(new=TRUE)
plot(tps_pls,vect_auc_pls, type="l",ylim=c(0.5,1),col="green",
xlab="temps",ylab="AUC")
par(new=TRUE)
plot(tps_plsdr,vect_auc_plsdr, type="l",ylim=c(0.5,1),col="pink",
xlab="temps",ylab="AUC")

dev.off()


################################################################################
###### travail sur la méthode KPLSDR  TRAVAIL EN COURS ####################
################################################################################

################## Calcul de la matrice "kernel" pour la méthode KPLSDR :


fct_kernel = function(vect1,vect2)
{
coeff_k <- exp((-sum((vect1-vect2)^2))/2)
return(coeff_k)
}


K <- matrix(1,240,240)

for (i in 1:239)
{
```

```
      for (j in (i+1):240)
      {
      K[i,j] <- fct_kernel(staudt.x[i,],staudt.x[j,])
      K[j,i] <- K[i,j]
      }
}


colnames(K2) <- 1:240
rownames(K2) <- 1:240
K2_train <- K2[,c(staudt.train.indices)]
K2_train <-t(K2_train)

K2_set <- K2[,-c(staudt.train.indices)]
K2_set <- t(K2_set)


# on sauvegarde le jeu de données pour le "training set" (création d'une
liste, 160 individus)

rosen3 <-
c(list(a=t(K2_train)),list(b=scale(staudt.tim.bis)),list(c=staudt.status.bi
s))

save("rosen3",file="Data_Rosen3_DeB.RData")

# on sauvegarde le jeu de données pour le "test set" (création d'une liste,
80 individus)

rosen4 <-
c(list(d=t(K2_set)),list(e=scale(staudt.tim.bis2)),list(f=staudt.status.bis
2))

save("rosen4",file="Data_Rosen4_DeB.RData")


### on réalise la PLS et on construit le modèle de Cox avec le training set

KPLSDR <- plsr(scale(deviance_residuals_rosen) ~
rosen3$a,ncomp=10,method=pls.options()$plsralg,)

COX_KPLSDR <- coxph(survie_rosen ~ KPLSDR$scores[,1] + KPLSDR$scores[,2] +
KPLSDR$scores[,3] + KPLSDR$scores[,4] + KPLSDR$scores[,5] +
KPLSDR$scores[,6] + KPLSDR$scores[,7] + KPLSDR$scores[,8] +
KPLSDR$scores[,9] + KPLSDR$scores[,10],)

############################################################

hp.COX_KPLSDR <- cox.zph(COX_KPLSDR, global=TRUE)
print(hp.COX_KPLSDR)
par(mfrow=c(2,5))
plot(hp.COX_PLS, var=1)
plot(hp.COX_PLS, var=2)
plot(hp.COX_PLS, var=3)
plot(hp.COX_PLS, var=4)
plot(hp.COX_PLS, var=5)
plot(hp.COX_PLS, var=6)
plot(hp.COX_PLS, var=7)
plot(hp.COX_PLS, var=8)
plot(hp.COX_PLS, var=9)
```

```
plot(hp.COX_PLS, var=10)

fct_survie_rosen_KPLSDR <- survfit(COX_KPLSDR)

plot(log(fct_survie_rosen_KPLSDR$time),log(-
log(fct_survie_rosen_KPLSDR$surv)))


##########################################################################
##

par(mfrow=c(2,5))

res_martingale.COX_KPLSDR <- resid(COX_KPLSDR, type="martingale")

X_KPLSDR <-
cbind(KPLSDR$scores[,1],KPLSDR$scores[,2],KPLSDR$scores[,3],KPLSDR$scores[,
4],KPLSDR$scores[,5],KPLSDR$scores[,6],KPLSDR$scores[,7],KPLSDR$scores[,8],
KPLSDR$scores[,9],KPLSDR$scores[,10])

b_KPLSDR <- coef(COX_KPLSDR)[c(1,2,3,4,5,6,7,8,9,10)]

for (j in 1:10) {
plot(X_KPLSDR[,j],b_KPLSDR[j]*X_KPLSDR[,j] + res_martingale.COX_KPLSDR,
xlab=c("KPLSDR$scores[,1]","KPLSDR$scores[,2]","KPLSDR$scores[,3]","KPLSDR$
scores[,4]","KPLSDR$scores[,5]","KPLSDR$scores[,6]","KPLSDR$scores[,7]","KP
LSDR$scores[,8]","KPLSDR$scores[,9]","KPLSDR$scores[,10]")[j],ylab="compone
nt+residuals")

abline(lm(b_KPLSDR[j]*X_KPLSDR[,j] +
res_martingale.COX_KPLSDR~X_KPLSDR[,j]), lty=2)

lines(lowess(X_KPLSDR[,j],b_KPLSDR[j]*X_KPLSDR[,j] +
res_martingale.COX_KPLSDR, iter=0))
                }

#####################################################################

### KPLSDR-AUC

mat_kplsdr <- rosen3$a
KPLSDRb <- plsr(scale(deviance_residuals_rosen)~mat_kplsdr,ncomp=10)
mat_kplsdr <- KPLSDRb$scores
COX_KPLSDRb <- coxph(survie_rosen ~ mat_kplsdr)
mat_kplsdr <- rosen4$d
pred_kplsdr <- predict(KPLSDRb,mat_kplsdr)
pred_kplsdr <- matrix(pred_kplsdr,80,10)

mat_kplsdr <- pred_kplsdr
colnames(mat_kplsdr) <- colnames(KPLSDRb$scores)
row.names(mat_kplsdr) <- row.names(rosen4$d)
predict.test.set.kplsdr <-
(predict(COX_KPLSDRb,newdata=as.data.frame(mat_kplsdr),type="lp"))


roc_auc_kplsdr_test_set <-function(x)
{
survivalROC(as.integer(staudt.tim.bis2),as.integer(staudt.status.bis2),pred
ict.test.set.kplsdr,entry=NULL, predict.time=x,span=0.05)$AUC
}
```

```
tps_kplsdr <-sort(unique(as.vector(staudt.tim.bis2)))

vect_auc_kplsdr <- sapply(tps_kplsdr,roc_auc_kplsdr_test_set)

################################################################################
################################

par(new=TRUE)
plot(fct_survie_rosen_KPLSDR,conf.int=F,xlab="temps de
survie",ylab="estimateur de Kaplan-Meier",col="blue",xlim=c(0,20))
```