Université Joseph Fourier (Grenoble I)

École doctorale « Mathématiques, Informatique, Sciences et Technologies de l'Information »

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ JOSEPH FOURIER

Spécialité : mathématiques appliquées

préparée au Laboratoire de Modélisation et Calcul

présentée et soutenue publiquement par

Julie Peyre

le 20 septembre 2005

Analyse statistique des données issues des biopuces à ADN

JURY

M Serge DEGERINE	Président
Mme Florence D'ALCHÉ-BUC	Rapporteur
Mme Irène GIJBELS	Rapporteur
M Badih GHATTAS	Examinateur
Mme Sophie LAMBERT-LACROIX	Examinatrice
M Anestis ANTONIADIS	Directeur de thèse
Mme Marie DUTREIX	Co-directrice de thèse

Remerciements

Je tiens tout d'abord à remercier mes deux directeurs de thèse : Anestis Antoniadis qui a toujours su trouver mot gentil et conseil avisé et Marie Dutreix, on ne pouvait rêver meilleur guide pour s'aventurer dans le monde de la biologie.

Je remercie Irène Gijbels et Florence d'Alché-Buc qui m'ont fait l'honneur et le plaisir d'accepter d'être les rapporteurs de cette thèse.

Merci aussi à Serge Dégerine, président du jury, qui m'a fait part de ses remarques judicieuses, et à Badih Ghattas pour avoir bien voulu faire partie de mon jury.

Sophie, travailler avec toi a été vraiment enrichissant. Et si ça n'a pas toujours été facile et qu'on en a vu de toutes les couleurs, tu as toujours eu un bel optimisme et un amour si évident de la recherche que ces moments passés avec toi à se creuser les méninges ou à déboguer des programmes restent parmi les meilleurs moments de mon travail de thèse.

Merci également à Isabelle Geahel, sans qui cette thèse n'aurait jamais commencé, et à Jacques Berruyer qui a accompagné nos premiers pas dans le domaine.

Un petit clin d'œil aussi à Géraldine et Nathalie toujours patientes pour m'expliquer les phénomènes biologiques. Merci à elles et à tous leurs collègues de l'Institut Curie qui m'ont accueillie à bras ouverts pendant mon DEA, même si je me suis parfois sentie comme une extra-terrestre en étant statisticienne au milieu de biologistes.

Ce mémoire a vu le jour après des années de travail, années pendant lesquelles j'ai vu les maîtres de conférence ou professeurs de mes études devenir des collègues fort sympatiques. Merci à toute l'équipe SMS du LMC où je me suis sentie d'emblée bien accueillie et merci en particulier à Olivier Gaudoin, sans doute pas étranger à mon choix de faire des statistiques.

Merci aussi à l'équipe administrative du labo, Juana, Cathy et en particulier Claudine dont l'efficacité me laisse admirative. Merci aussi à l'équipe informatique, à qui j'ai souvent dû avoir recours pour gérer mes petits problèmes de machine.

Pendant ces années de thèse, j'ai particulièrement apprécié d'avoir l'opportunité d'assurer des enseignements dans le cadre d'un monitorat. Merci donc à Gérard Cognet et à l'ensemble du CIES de Grenoble de m'avoir donné cette chance, ainsi qu'à tous mes étudiants pour ces moments passés avec eux. Lors de cette première expérience de l'enseignement, j'ai côtoyé des enseignants qui avaient le goût du travail bien fait, en particulier Alain Sombardier, Gérard Desquinabo, Anne Mittaine et Christine Kazantsev.

La vie au labo n'aurait pas été la même sans la joyeuse équipe de thésards qui m'entourait, Laurent en particulier qui partageait mon bureau mais qui a aussi su parfois m'en éloigner pour m'initier aux joies de l'escalade ou de la via ferrata, Antoine et Anne, avec qui on passe volontiers une heure comme une journée, Olivier l'homme qui pratiquait 3 milliards de sports, Aude M. toujours prête à partir d'un franc éclat de rire, Laurent, Etienne et Anne avec qui nous nous sommes bien soutenus dans les derniers moments (vivement que tu finisses Laurent, qu'on aille fêter ça). Il y a aussi François, Guillaume et Cécile, toujours partants pour des soirées "jeux et gastronomie", Sophie A. et Jocelyn, qui m'ont vue plus souvent sur les fesses que debout lors d'une mémorable journée surf, Yann un colocataire de bureau très agréable, Erwan avec qui les conversations ont parfois été des plus animées. Claire C., Basile, Aude R., Alex Y., Ouadia, Vincent, Irène, Promin, Elimane, Morgan, Clément, Marc, Alex B., Carine, Claire T., Robin, Olivier : les années passées à vos côtés ont été l'occasion de conversations K'Fet pour le moins divertissantes.

Merci à Caroline et Delphine, deux amies de longue date.

Merci aussi à mes amis de l'ENSIMAG qui, après les supers moments passés ensemble, ont su rester présents au fil des ans : Sébastien, Marion, Ben, Jérôme et Sylvie, Benoît, Julien.

Une petite pensée aussi pour Nicolas, fidèle partenaire de squash toujours gentil et attentionné (enfin tant que Liv Tyler n'est pas dans les parages).

Je m'en voudrais d'oublier mes copines de prépa. Hania et Stéphanie, nos retrouvailles, hélas trop rares, sont toujours pour moi de grands moments de joie. Céline et Tanit, on en a passé des heures au téléphone ces dernières années, vous avez toujours été là pour moi, à toute heure du jour et de la nuit, je ne vous en remercierai jamais assez.

Sophie, avec toi j'ai osé m'aventurer sur les chemins de randonnée comme dans la poudreuse, j'ai aussi beaucoup aimé nos longues discussions notamment lors de soirées travaux manuels. Ludovic, avec toi j'ai mangé trop de rosette, mais j'ai appris à dire « esta buena la banana » (alors 1ère, 2ème ou 3ème année ?).

Et puis, il y a les "cocottes", Hélène et Véro, celles qui m'ont soutenue au quotidien, qui ont vécu en direct les hauts et les bas de la thèse. Leur présence inébranlable et leur amitié inconditionnelle ont embelli les bons moments et adouci les autres... Et les cocottes seraient-elles les cocottes sans leurs adorables "cocos" ? Certainement pas, alors merci à Mathieu et Nicolas, toujours prêts à faire une (bonne) blague.

Je tiens également à remercier ma famille car... "bon sang ne saurait mentir !". Merci donc à mes parents et à ma soeur. Merci aussi aux "coucous", Natacha et Sylvie, plus encore que des cousines, des amies proches.

Enfin, et surtout, merci à Pierre, mon p'tit cœur...

Table des matières

Introduction au contexte biologique : les biopuces à ADN

La recherche en génétique progresse très vite, le séquençage du génôme est maintenant fini pour de nombreuses espèces comme les levures ou les souris... et même le génôme humain n'aurait plus de secrets. Cependant, la connaissance des séquences codantes de tous les gènes d'un organisme nécessite le développement de nouvelles biotechnologies pour étudier tous les gènes. Une étude des gènes un par un n'est pas concevable à cause de leur grand nombre : environ 6000 gènes pour les levures *Saccharomyces cerevisiae*. La technique des biopuces est l'un des outils les plus puissants qui a émergé à la suite du séquençage du génôme.

La technique des biopuces à ADN

Quelle information cherche-t-on à obtenir ?

Pour comprendre comment marchent les biopuces à ADN, il est nécessaire de donner quelques explications biologiques sur la structure de l'ADN et sur les différents acteurs de l'activité de la cellule.

Les protéines constituent la main d'œuvre de la cellule ; elles sont responsables de la structure cellulaire, ce sont elles qui produisent l'énergie et les biomolécules importantes qui participent aux constituants de la cellule (les chromosomes, les membranes et les voies métaboliques). Les protéines sont responsables de l'ensemble du métabolisme de l'ADN : synthèse, réplication et aussi réparation en cas de lésion. Certaines de ces protéines qui réparent l'ADN sont spécifiquement produites une fois que l'ADN a été endommagé. Mais quel est le mécanisme à l'origine de la production des protéines ? Ici, il est nécessaire de revenir au dogme central de la biologie moléculaire.

Il y a deux étapes principales qui conduisent de l'ADN à la protéine : la transcription et la traduction. L'ADN est constitué de deux brins d'acides nucléiques complémentaires qui s'emboîtent tout en s'enroulant l'un autour de l'autre pour former une double hélice (Watson et Crick, 1963). L'intermédiaire entre l'ADN et les protéines s'appelle l'ARN. Des fragments d'ADN sont transcrits en ARN, une molécule simple brin, c'est la transcription. On peut distinguer plusieurs classes d'ARN. La classe la plus diversifiée est celle des ARN messagers (notés ARNm) ;

ce sont les ARNm qui servent à la fabrication des protéines. La séquence portée par l'ADN correspondant à un ARN messager codant pour une protéine est appelée un gène. Il y a environ 6 000 gènes chez la levure et 30 000 chez l'homme. L'étape de construction des protéines à partir de l'ARNm s'appelle la traduction. Ces deux étapes, production d'ARNm (transcription) ou de protéines (traduction) sont régulées par d'autres protéines, selon l'état de la cellule. Ainsi, pour une cellule, la population d'ARNm ou de protéines est caractéristique de son état à un moment donné. Par conséquent, pour avoir une meilleure idée de ce qui se passe dans une cellule, on peut essayer de mesurer l'abondance de tous les ARNm cellulaires ou de toutes les protéines de la cellule selon l'étape de régulation à laquelle on s'intéresse.

Ce travail s'applique à l'étude de la régulation de la transcription, en analysant la population des ARNm de la cellule. La technique des biopuces à ADN récemment développée rend possible une mesure rapide de la quantité relative de chaque espèce d'ARNm dans la population totale des ARN de la cellule. Quand un ARNm est beaucoup produit, cela signifie que le gène correspondant est beaucoup transcrit, on dit que ce gène a un haut niveau d'expression ; les biopuces permettent d'analyser les variations de l'expression des gènes.

Quel est le principe de fonctionnement d'une biopuce ?

Maintenant, essayons de donner une idée de la technique des biopuces à ADN (voir aussi figure 0.1). La technique est fondée sur la structure de l'ADN. Chaque brin d'ADN est composé d'une chaîne de désoxyriboses sur laquelle sont attachés les nucléotides (A, T, C, G) qui codent l'information ; c'est un langage à quatre lettres. Ces nucléotides peuvent être regroupés en deux paires de bases complémentaires par des liaisons hydrogène : A est toujours reliée à T et C à G. Cela signifie par exemple qu'un T est toujours en face d'un A sur l'autre brin. Le principe de la transcription de l'ARN repose sur cette complémentarité des bases : l'ARNm est une sorte de copie inverse du brin d'ADN transcrit, mais avec la base T remplacée par la base U. On extrait l'ARN du pool de cellules qu'on veut étudier. On utilise alors une technique qui permet de créer un brin d'ADN qui est une photocopie inverse fluorescente de cet ARN : c'est la rétro-transcription, c'est à dire une opération inverse à la transcription. Elle permet de reproduire le brin d'ADN qui avait été transcrit (appelé ADN complémentaire et noté ADNc).

La biopuce est une lame de verre où tous les gènes (en simple brin) d'une espèce sont spottés à des endroits précis qui forment une grille. Une biopuce est très petite : environ la taille d'une lame de microscope . Quand on dépose le brin d'ADNc marqué sur la biopuce, par complémentarité des bases, il s'hybride avec le brin d'ADN spotté correspondant. On appelle cette phase l'hybridation. Toute la technique des biopuces repose sur la spécificité de cette hybridation. L'ADN simple brin spotté sur la biopuce est appelé la cible, l'ADNc fluorescent obtenu à partir de l'ARN extrait des cellules est appelée la sonde.

Si on suppose que la rétro-transcrition a complètement réussi, l'intensité fluorescente observée sur chaque spot est proportionnelle à la quantité de l'ARNm corres-

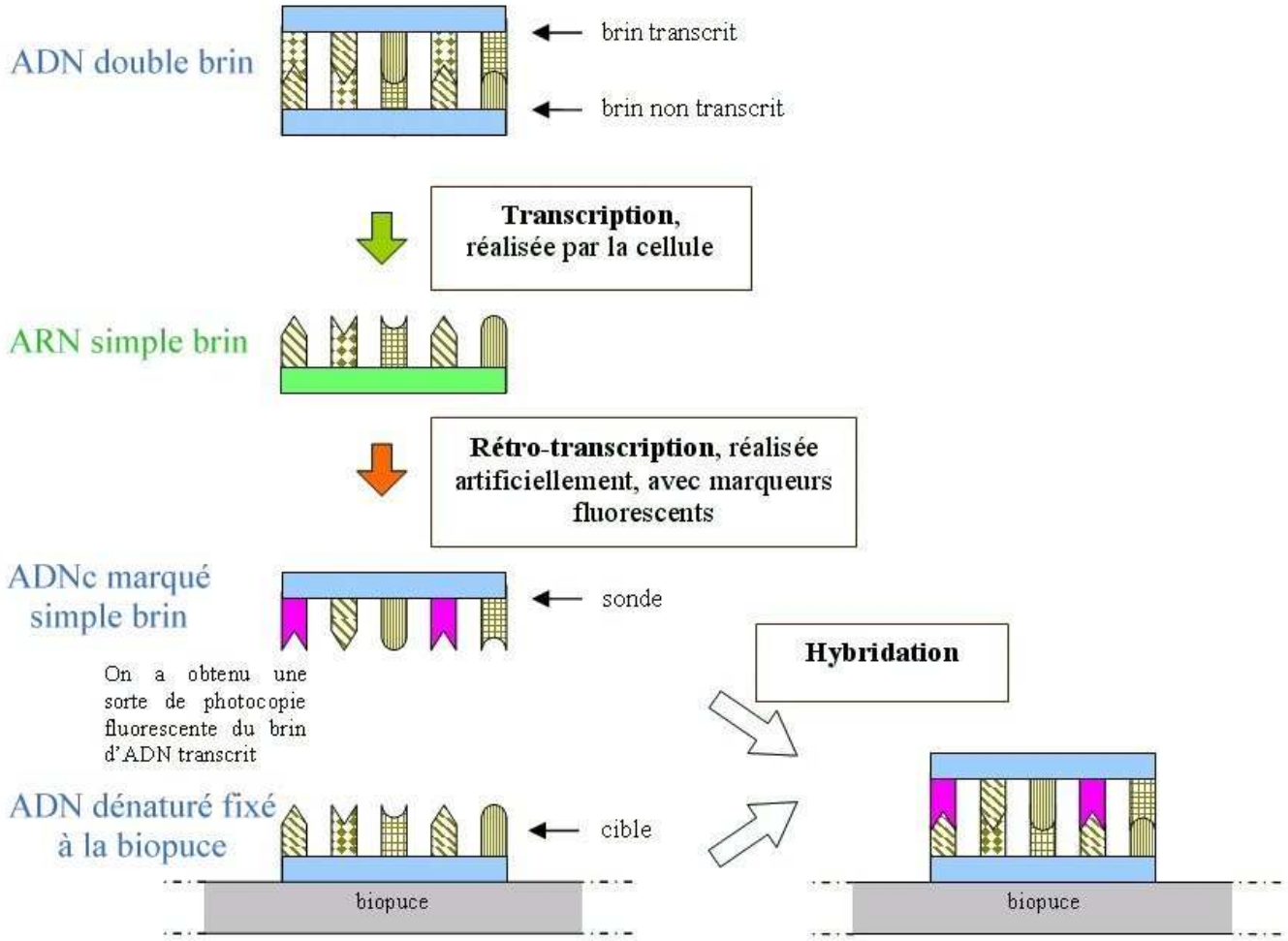

Figure 0.1 – Principe des biopuces à ADNc

pondant qui était présent dans la cellule. Pour mesurer cette intensité, on utilise un scanner et un logiciel d'analyse d'image.

Nous ne nous sommes intéressés qu'à des expériences à double marquage : nous disposions de deux marqueurs fluorescents différents, un pour marquer l'ADNc correspondant à des cellules appelées "contrôle" ou "référence", l'autre pour marquer l'ADNc correspondant aux cellules que l'on souhaite étudier (comparativement à la référence), c'est l'échantillon "test". Après l'hybridation simultanée des deux sondes, on scanne la biopuce dans chacune des deux fluorescences pour mesurer le niveau d'expression des gènes dans chaque échantillon. On fait ainsi une comparaison directe entre les niveaux d'expression des gènes dans deux échantillons différents en calculant le ratio entre les deux signaux mesurés pour chaque spot.

Jeux de données considérés

Tout au long de ce mémoire, on utilisera différents jeux de données. Parmi ces données de biopuces :

- des données publiques, accessibles sur internet, et largement manipulées par les statisticiens qui s'intéressent aux biopuces. Il s'agira des jeux de données Eset, Colon et Leukemia, que nous décrirons au moment où ils interviendront.

- des données de l'Institut Curie. Ces jeux de données concernent tous la levure *Saccharomyces cerevisiae* à laquelle on fait subir des traitements variés. Certains jeux correspondent à des cinétiques étudiées sur différentes souches de cette levure, après une irradiation importante par rayonnement γ. Ce jeu, composé de 26 lames, est appelé "donnee". Il sera globalement considéré comme un échantillon d'individus soumis à de fortes doses d'irradiation, pour cela il faudra toutefois retirer les deux expériences qui correspondent au temps zéro avant irradiation. Un deuxième jeu de données, nommé "faibles doses" et composé de 10 lames, correspond à un traitement par exposition continue à de très faibles débits de dose. Le jeu de données "non irradiés" (12 expériences) avait été réalisé dans l'objectif de mesurer des fluctuations de l'échantillon, c'est-à-dire en utilisant des échantillons identiques - du point de vue biologique - en Cy3 et en Cy5, on considèrera tout simplement qu'il s'agit d'individus n'ayant subi aucun traitement. Le jeu de données "formaldéhyde" (7 expériences) correspond à des levures exposées à un formaldéhyde, agent génotoxique, et ce à différentes doses.
  Il est à noter que tous ces jeux de données de l'Institut Curie correspondent à des expériences à double-marquage. Pour toutes ces expériences, on utilise en Cy3, échantillon contrôle, un mélange d'échantillons non traités, qui constitue une sorte de référence. C'est donc l'échantillon en Cy5, l'échantillon test, qui constitue l'échantillon d'intérêt. Lorsque l'on compare les résultats obtenus pour différentes conditions expérimentales, ce sont donc les expressions d'échantillon test par rapport à un même échantillon que l'on compare, ce qui a un sens.

Des données à grande variabilité : nécessité de traitements mathématiques

Grâce aux biopuces, il est désormais possible de suivre de près l'expression de milliers de gènes simultanément. Le défi est maintenant de donner du sens à de si grands jeux de données. De plus, les résultats obtenus ont une grande variabilité. Une normalisation des données est nécessaire pour identifier et supprimer les sources systématiques de variation. D'abord, voyons quelles sont ces sources de variation.

Variations inhérentes à la technologie des biopuces

Il existe plusieurs sources de variabilité dans la technologie des biopuces : les fluctuations biologiques et les incertitudes liées à la technologie.

Des fluctuations dans les échantillons biologiques

La biologie est, par nature, soumise à de nombreuses variations, il y a de nombreux paramètres expérimentaux et on ne peut jamais affirmer que deux échantillons sont rigoureusement identiques. Des études de fluctuation menées à l'Institut Curie à Orsay ont montré que les résultats bruts pouvaient différer lorsqu'on effectuait deux fois la même expériences avec le même matériel génétique *a priori*.

Incertitudes liées à la technologie

La technique de fabrication des biopuces à ADN donne généralement de bons résultats mais on peut souvent observer :

- des variations dans l'aiguille d'impression : les aiguilles transportent alors des quantités d'ADN cible différentes.
- des fluctuations aléatoires dans le volume cible même quand on utilise la même aiguille, par exemple à cause de l'évaporation pendant l'impression.
- des différences dans la fixation de la cible à la surface de la lame.
- et peut-être aussi des problèmes de vieillissement des lames.

Paramètres expérimentaux et hybridation

Les fluctuations liées aux paramètres expérimentaux pour préparer l'hybridation ou pendant l'hybridation sont nombreux :

- la rétro-transcription pour fabriquer de l'ADNc
- marquage par les fluorochromes : l'efficacité du marquage varie souvent d'un fluorochrome à l'autre.
- Paramètres expérimentaux d'hybridation : l'efficacité de la réaction d'hybridation est influencée par de nombreux paramètres expérimentaux comme la température, le temps, la répartition de la sonde sur la lame, et la quantité totale de molécules sonde utilisées pour l'hybridation.

Scan et analyse d'image

Après l'hybridation, la biopuce est scannée. Il y a plusieurs types de scanners avec des caractéristiques différentes et plusieurs paramètres ajustables. On aimerait que les résultats soit aussi indépendants de ces paramètres que possibles. Il existe plusieurs types de bruit qui peuvent affecter le signal final rendu par le scanner. On peut les partager en deux grandes catégories : les bruits liés à la source et les bruits liés à la détection. Les bruits liés à la source incluent les phénomènes qui peuvent changer les mesures prises par le scanner, indépendamment du scanner utilisé, comme de la poussière sur les biopuces, ou le traitement qui a été fait sur les lames de verre par exemple. Les bruits liées à la détection sont ceux qui sont générés directement par le scanner.

L'analyse d'image est un aspect important des expériences de biopuces, elle peut avoir beaucoup d'influence sur l'identification des gènes différentiellement exprimés. Dans une expérience, après l'hybridation, l'image de la biopuce est analysée pour mesurer les intensités lumineuses dans chacune des fluorescences pour chaque spot sur la lame. Ces intensités fluorescentes correspondent au niveau d'hybridation des deux échantillons à la séquence d'ADN imprimé sur la lame. Les intensités de fluorescence sont stockées comme des images 16 bits que l'on peut considérer comme les données brutes.

Un certain nombre de logiciels d'analyse d'image, commerciaux ou pas, sont disponibles. Il est aisé de comprendre qu'on ne peut accepter que les résultats soit radicalement différents selon le logiciel utilisé (en supposant qu'il est assez bon quand même) !

Il y a trois tâches principales dans l'analyse des images de biopuces :

- Repérage : dans cette étape, il s'agit d'assigner des coordonnées à chacun des spots.
- Segmentation : cette étape permet de sélectionner les pixels qui feront partie du signal et ceux qui seront considérés comme faisant partie du bruit de fond.
- Extraction des intensités : cela signifie calculer, pour chaque spot sur la biopuce, les intensités dans les deux fluorescences pour le signal et pour le bruit de fond.

Dans le cadre de ce travail, on ne disposait pas de jeux de données obtenus à partir de différents scanners ou logiciels. Le logiciel utilisé à l'Institut Curie était le système d'acquisition de données GenePix. GenePix^{TM} 4000A de chez Axon Instruments, Inc., est un système complet qui intègre le scan des biopuces et le logiciel d'analyse.

Problèmes statistiques liés aux données de biopuces

La forte variabilité inhérente à cette technologie montre bien la nécessité de traitements statistiques avant de pouvoir en tirer une quelconque interprétation biologique. Les problèmes statistiques impliqués dans les données de biopuces sont très nombreux ; on peut les classer en six catégories principales :

- analyse d'image pour extraire l'information, *i.e.* les valeurs des intensités, à partir de l'image obtenue par scan.

- choix des données utilisées : il s'agit ici de savoir, parmi la quantité d'informations fournies, quelles sont celles qu'on va utiliser pour caractériser chaque gène (valeur médiane ou moyenne, correction ou pas du bruit de fond ...).
- normalisation d'une biopuce qui devra aussi dépendre du type d'expérience réalisée, ce qui changera les hypothèses que l'on peut raisonnablement faire (peu de gènes exprimés, beaucoup de gènes exprimés...)
- détection des gènes différentiellement exprimés, c'est-à-dire parvenir à décider statistiquement quels sont les gènes qui présentent des différences d'expression significatives, entre deux fluorescences ou entre deux séries d'expériences.
- classification supervisée ou pas des données.
- étude de cinétiques pour essayer par exemple de distinguer les groupes de gènes qui évoluent de façon similaire au cours du temps, et chercher éventuellement des réseaux de régulation.

Les points précédemment cités ne donnent qu'un petit aperçu de toutes les questions statistiques induites par les données de biopuce. Dans le cadre de cette thèse, nous nous sommes intéressés plus particulièrement à trois de ces points.

Dans un premier chapitre, nous étudions les méthodes de normalisation. L'objectif de la normalisation est d'éliminer les variations parasites entre les deux échantillons utilisés dans l'expérience pour ne conserver que les variations expliquées par un phénomène biologique. Dans ce chapitre nous n'étudierons que les données de l'Institut Curie dans la mesure où les autres données considérées ne fournissent pas suffisamment d'éléments pour effectuer une normalisation satisfaisante. La normalisation sera donc développée dans le cadre des expériences à double-marquage. L'hypothèse biologique fondamentale qui sous-tend toute la méthode est de supposer que les différences entre "référence" et "test" ne devraient concerner qu'une minorité de gènes. Nous présentons plusieurs méthodes de normalisations existant dans la littérature pour lesquelles nous proposons des améliorations. Pour être en mesure de choisir une méthode, nous mettons au point une méthode de simulation de biopuces.

Dans un deuxième chapitre, nous nous intéressons à la détection des gènes différentiellement exprimés entre deux séries d'expérience quand on a plusieurs répétitions d'expériences. On suppose que l'on a n_1 biopuces correspondant à une condition expérimentale A donnée et n_2 pour une condition expérimentale B donnée. On veut identifier les gènes différentiellement exprimés de façon significative entre les deux traitements. Plusieurs approches ont été envisagées comme la sélection de modèles par procédure FDR (False Discovery Rate), une méthode de seuillage bayésien ou encore des méthodes de sélection de modèles pénalisées.

Enfin, dans un troisième chapitre, nous considérons les problèmes de classification supervisée pour les données de biopuces. Dans ce type de données, on souffre du « fléau de la dimension », c'est-à-dire que l'on dispose d'un petit nombre d'individus devant le grand nombre de covariables (le nombre de gènes). D'un point de vue

statistique, ce grand nombre de prédicteurs comparé à un petit nombre d'individus rend les problèmes statistiques difficiles. Pour éviter cela, l'idée consiste à utiliser des méthodes de réduction de dimension. Nous avons développé une méthode semi-paramétrique de réduction de dimension, basée sur la maximisation d'un critère de vraisemblance locale dans les modèles linéaires généralisés. L'étape de réduction de dimension est alors suivie d'une étape de régression par polynômes locaux pour effectuer la classification supervisée des individus considérés. Pour l'instant nous nous sommes limités au cadre des modèles en indice simple, c'est-à-dire que la projection est faite sur un espace de dimension 1.

Chapitre 1

Comparaison des méthodes de normalisation et simulation de données issues des biopuces à ADN

Introduction : pourquoi normaliser ?

Comme nous l'avons vu dans la partie précédente, la technique des biopuces est soumise à de nombreuses variations expérimentales qui rendent impossible l'exploitation directe des résultats. Dans les expériences à double marquage (échantillon contrôle en Cy3, échantillon test en Cy5), l'une des sources principales de variation intervient au moment de l'incorporation des fluorochromes Cy3 et Cy5. On sait que l'efficacité de cette incorporation n'est pas la même pour les deux marqueurs. Pour ne garder que les variations entre les deux échantillons dues aux différences de traitement qu'ils ont subi, il faut trouver un moyen de normaliser les données pour éliminer ces différences qui existent entre les deux fluorescences.

Plusieurs méthodes de normalisation des données de biopuces ont été proposées dans la littérature, nous nous proposons de les étudier en détail afin de sélectionner la mieux adaptée. Cette partie, dont le contenu est très important pour les biologistes analysant les données issues des biopuces est principalement descriptive et ne contient que peu de développements mathématiques. Notre apport principal est la mise au point d'une méthode de simulation pour aboutir au choix le plus adéquat de la méthode de normalisation.

Nous allons donc dans un premier temps donner une description détaillée de ces méthodes de normalisation. Par la suite, nous allons mettre au point une méthode de simulation de données issues de biopuces à ADN pour enfin, dans une troisième partie, appliquer les normalisations à ces données simulées et être en mesure de faire un choix que nous ne remettrons plus en question dans la suite de ce travail.

Normalisation des données

1.1.1 Notations et généralités sur la normalisation

Le principe de la normalisation repose sur l'hypothèse fondamentale que la plupart des gènes ont le même niveau d'expression dans les deux échantillons, ce qui signifie que peu de gènes sont différentiellement exprimés. Cette hypothèse pourrait ne pas être valable dans tous les types d'expériences mais nous considérons qu'elle est vérifiée pour nos expériences dans la mesure où les deux échantillons utilisés dans les deux fluorescences sont très similaires du point de vue biologique.

Définissons tout d'abord quelques notations qui nous seront utiles tout au long de ce document. On note p le nombre de spots présents sur la biopuce. On notera $G = (G_i)_{i=1...p}$ les intensités mesurées en Cy3 (échantillon contrôle « Green » par convention) pour les p spots et $R = (R_i)_{i=1...p}$ les intensités mesurées en Cy5 (échantillon test « Red » par convention). R et G peuvent éventuellement ne pas être des données brutes obtenues après analyse de l'image de la biopuce, elles peuvent être les intensités obtenues après correction du bruit de fond (*cf* section 1.1.2).

On choisit souvent de faire une transformation $\log_2$ des données. Cette transformation présente le double avantage de dilater les nombreuses petites valeurs obtenues et de resserrer les quelques valeurs très importantes. Cela permet aussi une meilleure visualisation des données et laisse apparaître une structure dans le nuage de points que nous étions bien incapables de déceler auparavant, c'est une transformation d'échelle. Ainsi, sur la figure 1.1 on a représenté le nuage de points (G, R) sans et avec transformation.

Par la suite, on va s'intéresser à la comparaison des expressions dans les deux fluorescences. Un moyen de la faire est de considérer les ratios de $\frac{R}{G}$. Une représentation qui nous permettrait de voir directement ces ratios serait donc la bienvenue. Le choix que nous allons faire (et qui est courant dans le domaine) est de visualiser les données sous la forme d'un nuage représentant les $\log_2$-ratios notés M en fonction de la $\log_2$-intensité moyenne totale sur le spot notée A.

C'est-à-dire qu'on calcule :

$$A \;=\; log_2(\sqrt{R \times G}) \;=\; \frac{log_2(R) + log_2(G)}{2}$$

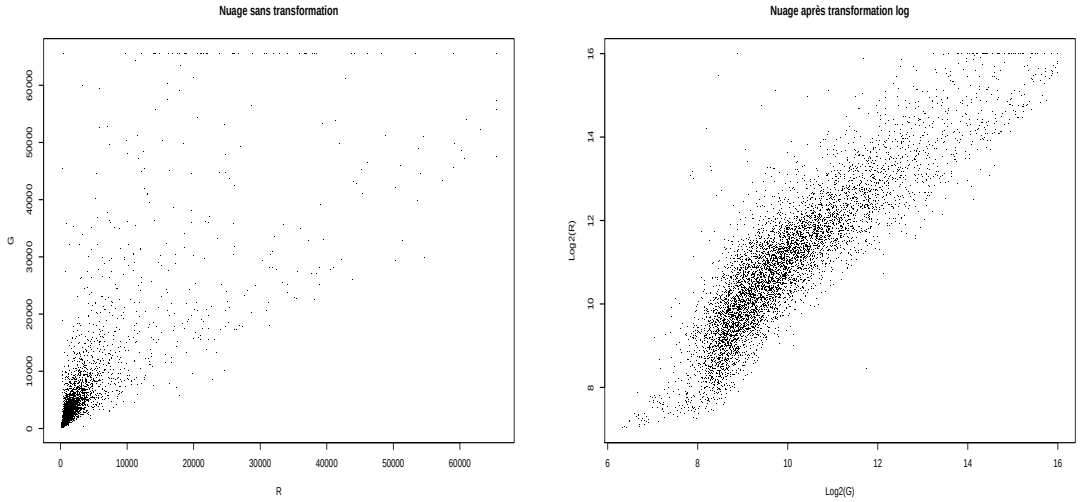

Figure 1.1 – Nuage de points pour une biopuce (donnee12), avant et après la transformation log. Sur cette figure, on peut voir 2 nuages de points correspondant au même jeu de données, à gauche sans aucune transformation, à droite avec un passage au logarithme de base 2.

$$M \;=\; log_2(\frac{R}{G}) \;=\; log_2(R) - log_2(G)$$

On peut noter que le passage du nuage en échelle logarithmique à ce mouveau nuage de point correspond simplement à une rotation de 45^o avec un léger changement d'échelle (*cf* figure 1.2).

L'hypothèse de normalisation implique que les distributions devraient être à peu près identiques en $Cy3$ et en $Cy5$ et que la distribution des rapports R/G (resp. des $\log_2$-ratios M) devrait être centrée autour de 1 (resp. 0) pour les gènes non différentiellement exprimés, c'est-à-dire la majorité des gènes. Or quand on regarde les jeux de données qu'on obtient après expérience, on constate que ce n'est pas le cas. C'est vers ce résultat que va devoir tendre la normalisation. Mais d'abord se pose le problème de savoir quelles données on va normaliser.

1.1.2 Correction du bruit de fond

1.1.2.1 Définition du bruit de fond

Après l'hybridation, une biopuce est scannée pour pouvoir générer des fichiers où les résultats de l'hybridation sont traduits numériquement. On obtient en sortie une grande quantité d'information pour chacune des deux fluorescences. En particulier,

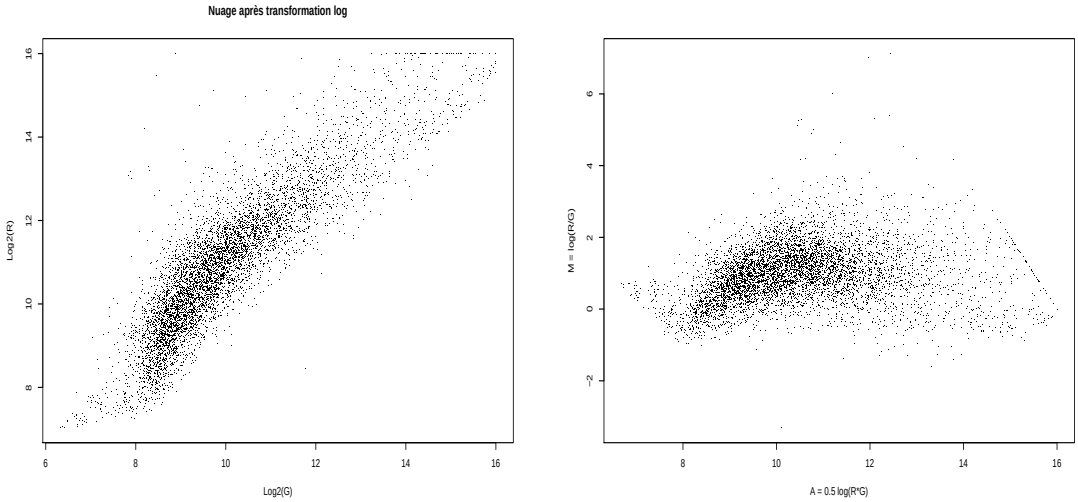

Figure 1.2 – Nuage M vs. A (à droite) pour une biopuce (donnee12) en comparaison avec une simple transformation logarithmique (à gauche)

on a pour chaque gène et pour chaque fluorescence : la moyenne des intensités de tous les pixels sur la zone correspondant au gène (notée X_f), la médiane de ces intensités (Y_f), l'écart-type de ces intensités et le nombre de pixels dans la zone considérée. Nous avons aussi des estimations identiques pour le bruit de fond (X_b et Y_b).La zone sur laquelle on évalue le bruit de fond dépend du logiciel utilisé : il peut s'agir par exemple de tous les pixels dans un voisinage du spot ou de tous les pixels dans un voisinage du spot qui sont au moins à une distance de quelques pixels du spot.

La zone de bruit de fond considérée par le logiciel d'analyses d'image utilisé à l'Institut Curie, GenePix, est donnée par la figure 1.3.

1.1.2.2 Correction du bruit de fond, approche classique

Le bruit de fond apporte une information supplémentaire dont il serait dommage de ne pas tenir compte, mais comment le prendre en considération ? Il est assez répandu dans le domaine des biopuces de retrancher le bruit de fond mesuré pour un gène au signal mesuré pour ce même gène afin d'obtenir une version du signal corrigée par le bruit de fond. Cependant, cette méthode pose des problèmes, notamment lorsque l'intensité lumineuse est faible, du même ordre de grandeur que le bruit de fond. Cela pourra amener à des estimateurs du log_2-ratio très bruités voire même non définis quand l'intensité du bruit de fond devient supérieure à l'intensité du signal. Pourtant, ces mesures contiennent une information valable et utile.

Dans une expérience de biopuces, on s'intéresse à l'intensité lumineuse de chaque

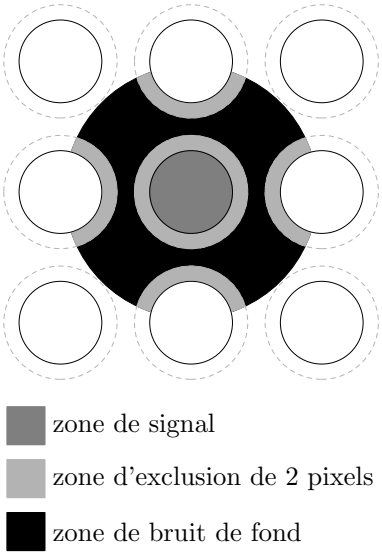

Figure 1.3 – Zone de définition du bruit de fond

spot pour estimer la quantité de matériel génétique qui s'est fixé sur la biopuce. Il y a une hypothèse cachée derrière cela : on suppose que la quantité de brins d'ADN qui se fixent à un spot donné et l'intensité lumineuse de ce spot sont linéairement dépendantes. Dans la zone de bruit de fond, l'ADN peut se fixer à une partie non traitée de la biopuce. Dans la zone de signal, l'ADN peut s'hybrider avec l'ADN complémentaire cible ou se fixer directement sur le verre. On suppose que ces deux effets sont additifs. On formalise en supposant que l'intensité de chaque pixel dans la zone de bruit de fond est une variable aléatoire de moyenne μ_b et que l'intensité de chaque pixel de la zone de signal est une variable aléatoire de moyenne $\mu_f = \mu_t + \mu_b$ avec $\mu_t \geq 0$. μ_t est la valeur moyenne de l'intensité due à l'hybridation spécifique sur la cible (t comme « target » en anglais). On cherche à estimer cette moyenne μ_t. Notons Z_{if} (resp. Z_{ib}) l'intensité lumineuse du pixel i dans le spot (resp. dans la zone de bruit de fond). Si on suppose que la quantité d'ADN qui se fixe sur une zone donnée de la biopuce est indépendante de celle qui se fixe dans une autre zone, alors Z_{if} (resp. Z_{ib}) devrait être proportionnelle à une variable suivant une loi de Poisson. En particulier, αZ_{if} (resp. αZ_{ib}) suivra une loi de Poisson de moyenne μ_f/α (resp. μ_b/α), le coefficient de proportionnalité α pouvant être différent selon la fluorescence considérée (Cy3 ou Cy5).

Retrancher le bruit de fond revient à prendre comme estimateur de μ_t,

$$\hat{\mu_t} = X_f - X_b \text{ ou bien } \hat{\mu_t} = Y_f - Y_b$$

Ces estimateurs posent problème quand μ_f est proche de μ_b car alors μ_t est proche de zéro voire même négatif. Cela s'avère gênant dans la suite des analyses, quand on s'intéresse au ratio μ_{tr}/μ_{tg} (r et g étant les indices pour désigner l'expression en Cy5

et en Cy3 respectivement), car alors les gènes pour lesquels μ_{tr} ou μ_{tg} sont négatifs doivent être ignorés. C'est pour éviter ce désagrément qu'une approche bayésienne a été proposée par Kooperberg *et al.* [28].

1.1.2.3 Approche bayésienne (Kooperberg *et al.*)

L'idée est la suivante : si la vraie intensité du bruit de fond est égale à μ_b, on observe une intensité du bruit de fond X_b qui peut être plus petite ou plus grande, suivant approximativement une loi normale de moyenne μ_b. Si on ne dispose pas de plus d'information, X_b est le meilleur estimateur de μ_b. De la même façon, si la vraie valeur de l'intensité du signal est $\mu_f = \mu_t + \mu_b$, on observera une intensité du signal X_f qui peut être inférieure ou supérieure, suivant approximativement une loi normale de moyenne μ_f, avec juste cette information, X_f serait le meilleur estimateur de μ_f. Mais en fait on dispose d'une information supplémentaire, on sait que $\mu_t \geq 0$ et donc que $\mu_f \geq \mu_b$. Par conséquent, si on observe des valeurs telles que $X_b > X_f$, c'est-à-dire où le bruit de fond est plus important que le signal, on devra au moins avoir $\mu_b < X_b$ et $\mu_f > X_f$ pour obtenir un μ_t positif. Détaillons maintenant les calculs qui permettent d'aboutir à une meilleure estimation de μ_t.

Distribution *a posteriori* de μ_t

Comme X_b et X_f sont des moyennes sur un assez grand nombre de pixels, grâce au théorème central limite, on a : $X_b \sim \mathcal{N}(\mu_b, \sigma_b^2)$ et $X_f \sim \mathcal{N}(\mu_t + \mu_b, \sigma_f^2)$. On suppose que X_b et X_f sont indépendantes conditionnellement à μ_b, μ_t, σ_b et σ_f. On note p_{μ_t} et p_{μ_b} les densités *a priori* des intensités respectives de la cible et du bruit de fond. On suppose que ces distributions *a priori* de μ_t et μ_b sont indépendantes entre elles et indépendantes de σ_b et σ_f.

Grâce au théorème de Bayes, on a :

$$p(\mu_t, \mu_b | \sigma_f, \sigma_b, X_f, X_b) = \frac{p(X_f, X_b | \mu_t \mu_b, \sigma_f, \sigma_b)\ p(\mu_t, \mu_b | \sigma_f, \sigma_b)}{p(X_f, X_b | \sigma_f, \sigma_b)}$$

Ce qui donne :

$$p(\mu_t | \sigma_f, \sigma_b, X_f, X_b) \quad = \quad \frac{\phi\left(\frac{X_f - \mu_t - X_b}{\sigma_d}\right)\ \Phi\left(\frac{\left(X_f - {}_m u_t\right)\sigma_b^2 + X_b \sigma_f^2}{\sigma_f \sigma_b \sigma_d}\right)}{\sigma_d \int_0^\infty \Phi\left(\frac{X_f - v}{\sigma_f}\right)\ \phi\left(\frac{X_b - v}{\sigma_b}\right) dv} \tag{1.1}$$

si $\mu_t \geq 0$ et 0 sinon.

Dans (1.1), $\phi(\cdot)$ désigne la densité de la loi normale centré réduite et $\Phi(\cdot)$ la fonction de répartition correspondante, $\Phi(x) = \int_{-\infty}^{x} \phi(x) dx$.

Revenons aux hypothèses qui ont été faites. Il est assez courant en pratique d'avoir sur un spot des pixels qui ont des niveaux d'hybridation assez différents, sans doute parce que la cible n'est pas uniformément répartie sur le spot. Une façon de contourner ce problème est d'utiliser les intensités médianes Y_f et Y_b au

lieu des intensités moyennes X_f et X_b. En effet la médiane est plus robuste aux valeurs extrêmes. Comme la quantité d'ADN qui s'hybride sur chaque pixel est importante, la distribution de Poisson suivie par αZ_{if} (et αZ_{ib}) est bien approximée par une loi normale. Cela implique que, tout comme X_f et X_b, Y_f et Y_b suivent approximativement une loi normale de moyennes respectives μ_f et μ_b. Et alors, l'équation (1.1) reste approximativement valide si on remplace les moyennes X_f et X_b par les médianes Y_f et Y_b.

Ainsi, pour chacune des deux fluorescences, l'équation (1.1), avec des Y à la place des X donne une méthode pour calculer la loi *a posteriori* de μ_t si on connaît des estimateurs de σ_f et σ_b. Ici, on ne peut pas considérer que les niveaux d'intensité de chacun des pixels sont indépendants au sein d'une même zone (signal ou bruit de fond). On se contentera de supposer que les intensités des différents pixels ont la même distribution et que la corrélation entre l'intensité de deux spots dépend seulement de la distance entre ces deux spots.
Il en découle :

$$\sigma_f = a\frac{S_f}{\sqrt{n_f}} \tag{1.2}$$

$$\sigma_b = a\frac{S_b}{\sqrt{n_b}} \tag{1.2}$$

pour une constante a, en ignorant les effets de bord (Ripley [37]).

La correction par le bruit de fond suppose implicitement que l'intensité du bruit de fond est localement constante. Si on suppose que μ_b est (approximativement) constante d'un spot à l'autre, l'écart-type empirique de Y_b pour ce spot et ses spots voisins est un autre estimateur possible de σ_b pour ce spot. On estime a séparément pour chaque fluorescence en faisant la régression de $\frac{S_b}{\sqrt{n_b}}$ par σ_b pour tous les spots de la biopuce. Cet estimateur $\hat{a}$ est alors combiné aux écart-types du signal (1.2) et du bruit de fond (1.2) fournis par l'analyseur d'image pour obtenir des estimateurs de σ_f et de σ_b.

Cette approche bayésienne pour la correction du bruit de fond suppose que si l'intensité du bruit de fond Y_b est supérieure à celle du signal Y_f, c'est que la moyenne μ_t est très faible, et on observe en fait un événement dû au hasard. En particulier, on suppose que :

$$Y_b - Y_f \sim \mathcal{N}(\mu_t, \sigma_f^2 + \sigma_b^2)$$

Comme $\mu_t \geq 0$, on a :

$$q = \Phi\left(\frac{Y_f - Y_b}{\sqrt{\sigma_f^2 + \sigma_b^2}}\right)$$

qui donne une sorte de niveau de signification pour l'hypothèse $\mu_t \geq 0$ et donc permet de décider si le modèle est raisonnable de ce point de vue.

1.1.2.4 Comparaison des deux approches sur des jeux de données

Nous allons appliquer les deux méthodes de correction du bruit de fond sur des jeux de données réels afin de pouvoir les comparer. Sur la figure 1.4, on peut voir trois nuages de points ; le premier est obtenu sans retrancher le bruit de fond, le deuxième en retranchant le bruit de fond de manière classique, c'est-à-dire en faisant une simple soustraction et enfin, le troisième nuage de points est obtenu est utilisant une méthode bayésienne de correction par le bruit de fond. On remarque qu'une approche bayésienne permet de considérer beaucoup plus de points et donc de gènes que l'approche classique. De plus, quand on regarde où sont situés les points correspondants dans le nuage, on se rend compte que le fait de conserver ces points aura certainement une importance dans la suite des analyses. En effet, il y en a un certain nombre qui présentent, certes avant normalisation, un $log - ratio$ d'expression relativement important. Le problème, c'est que cette correction du bruit de fond semble ajouter du bruit dans les données quelle que soit la méthode de correction utilisée. C'est pourquoi il semble délicat d'effectuer une correction du bruit de fond, et certains auteurs préfèrent ne pas corriger. Dans la suite, pour présenter les différentes méthodes de normalisation, on considèrera des données sur lesquelles aucune correction de bruit de fond n'aura été faite. On reviendra plus tard à des considérations sur le bruit de fond dans la partie simulation de données.

1.1.3 Différents types de normalisation

Maintenant, nous allons présenter les différents types de normalisation qui ont été envisagés au cours de notre étude.

1.1.3.1 Normalisation par la médiane

C'est la normalisation qui a beaucoup été utilisée aux débuts des biopuces et qui est encore souvent proposée par les logiciels d'analyse d'image de biopuces. Cette méthode suppose l'existence d'un coefficient de proportionnalité c entre le rouge et le vert qui ne dépend pas du gène. Il s'agit donc en fait d'une correction linéaire.

Modèle :

$$(R)_i \ = \ c \ \times (G)_i \qquad où \ i = 1 \ldots p$$

ce qui équivaut à :

$$(log_2 R)_i \ = \ K \ + (log_2 G)_i \qquad où \ i = 1 \ldots p$$

ou encore :

$$M_i = K \qquad où \ i = 1 \ldots p$$

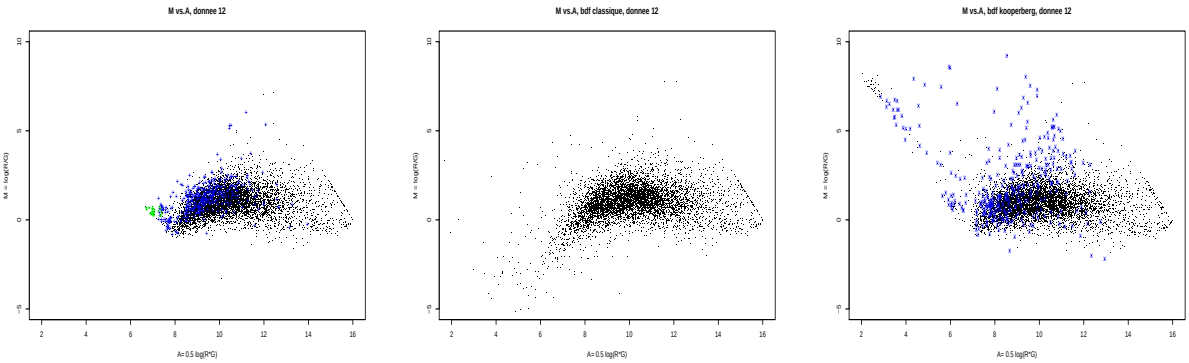

Figure 1.4 – Nuage M vs. A pour une biopuce (donnee12), sans retirer le bruit de fond, et en le retirant de 2 façons différentes. Sur cette figure, on peut voir 3 nuages de points correspondant au même jeu de données, avec des traitements du bruit de fond différents. De gauche à droite : sans correction du bruit de fond, avec correction par l'approche classique et avec correction par l'approche bayésienne. Les points en bleu représentent les gènes enlevés du calcul dans la démarche classique et pas dans la démarche bayésienne ; les points en vert représentent les gènes pour lesquels on ne peut calculer la valeur pour aucune des deux méthodes. Il n'y a aucun gène pour lequel on puisse calculer une valeur corrigée par le bruit de fond par la méthode classique et pas par la méthode bayésienne.

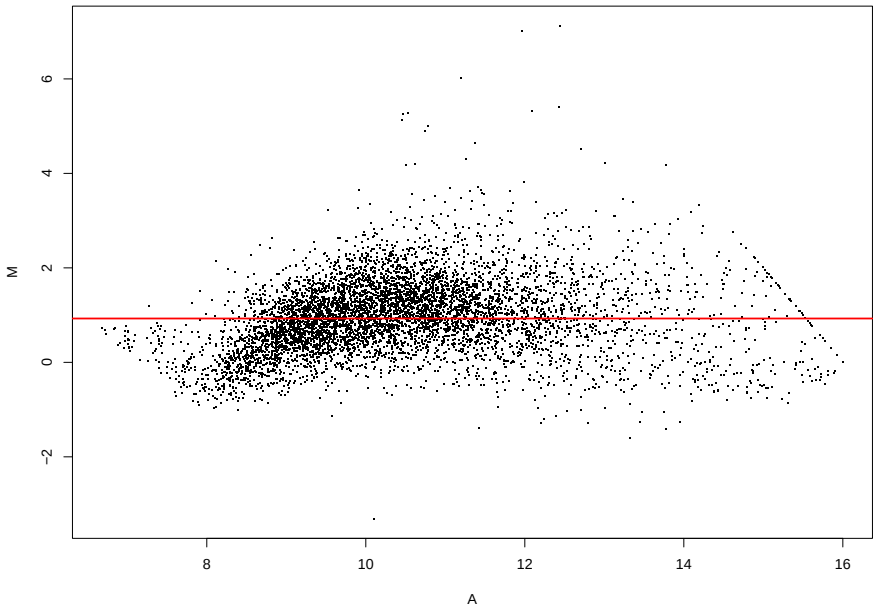

Figure 1.5 – Représentation (A,M) avant toute normalisation des données. La droite tracée est la fonction de normalisation choisie quand on effectue une normalisation par la médiane. On peut voir que cette fonction ne tient pas du tout compte de la forme du nuage. La fonction de normalisation devrait dépendre de A.

Ainsi pour la majorité des gènes on suppose qu'on devrait avoir un $\log_2$-ratio constant égal à $K = log_2(c)$. Pour estimer ce K, on prend la médiane des valeurs des $\log_2$-ratios, moins sensible que la moyenne aux valeurs aberrantes. On retranche ensuite cette valeur à l'ensemble des $\log_2$-ratios pour centrer l'ensemble des mesures sur un $\log_2$-ratio de zéro.

On voit très bien sur la figure 1.5 qu'avec ce genre de normalisation on ne va pas changer la courbure du nuage de points. Après normalisation, le nuage M vs. A aura exactement la même forme mais aura été décalé vers le zéro. Ainsi quand on voit la forme de virgule qu'a le nuage de points, on sait qu'après la normalisation, on aura toujours des faibles $\log_2$-ratios pour les faibles et hautes intensités totales (grandes valeurs de A) et des $\log_2$-ratios plus importants pour les valeurs intermédiaires de l'intensité totale. Cela montre que la normalisation par la médiane n'est sans doute pas adaptée car on ne veut pas que l'expression différentielle soit dépendante de l'intensité des spots. Pour remédier à ce problème, l'idée c'est de chercher une fonction de normalisation qui dépend de la $\log_2$-intensité totale, c'est l'esprit de la normalisation lowess.

Remarque : en corrigeant par le bruit de fond selon l'approche bayésienne proposée par Kooperberg, on aurait la figure 1.6

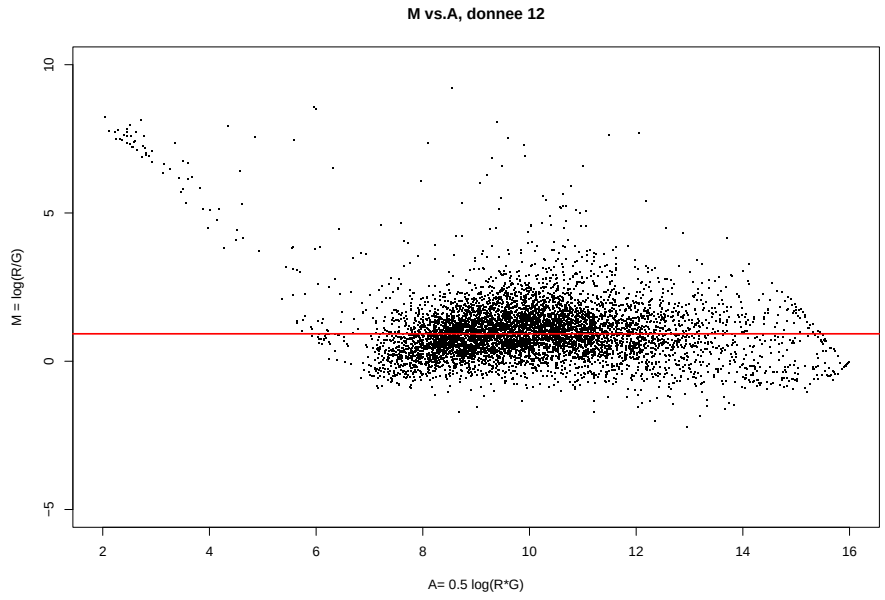

Figure 1.6 – Représentation (A,M) avant toute normalisation des données, en ayant corrigé par le bruit de fond. La droite tracée est la fonction de normalisation choisie quand on effectue une normalisation par la médiane. On peut voir que cette fonction ne tient pas du tout compte de la forme du nuage. La fonction de normalisation devrait dépendre de A.

1.1.3.2 Normalisation lowess et améliorations

Normalisation lowess

L'idée de départ de cette méthode c'est de faire une normalisation qui ne soit pas la même pour tous les gènes. Cela revient à dire :

$$(R)_i \ = \ c_i \ \times (G)_i \qquad où\ i = 1 \ldots p$$

ce qui équivaut à :

$$(log_2 R)_i \ = \ K_i \ + (log_2 G)_i \qquad où\ i = 1 \ldots p$$

ou encore :

$$M_i = K_i \qquad où\ i = 1 \ldots p$$

Mais un tel modèle ne peut être retenu car on a autant de coefficients à estimer que de mesures avec un problème d'identifiabilité important. Une possibilité est de chercher une fonction p de G_i et R_i pour avoir :

$$(log_2 R)_i \ = \ p(G_i \ , \ R_i) \ + (log_2 G)_i \qquad où\ i = 1 \ldots p$$

soit :

$$M_i \ = \ p(G_i \ , \ R_i)$$

Le problème reste de définir cette fonction p. Les graphiques des $\log_2$-ratios M en fonction de de la $\log_2$-intensité totale A nous ont montré que la fonction de normalisation devrait dépendre de A.
C'est pourquoi l'équipe de Terry Speed [49] propose d'estimer une fonction lisse ρ telle que :

$$M_i \ = \ \rho(A_i)$$

C'est pour estimer cette fonction ρ qu'on utilise une approximation *lowess*. Qu'est-ce qu'une approximation lowess ? Lowess veut dire « **LO**cally **WE**ighted **S**catterplot **S**moothing ». Il s'agit d'une technique d'analyse des données qui permet de produire un ensemble lisse de valeurs à partir d'un diagramme de dispersion avec une relation bruitée entre les deux variables. Le lowess combine la simplicité de la régression linéaire par moindres carrés et la flexibilité de la régression non-paramétrique. Il s'agit en fait d'une régression polynomiale locale. Le lowess ajuste des modèles simples à des sous-ensembles localisés du jeu de données pour construire une fonction qui décrit la part déterministe de variation dans les données point par

point. On définit un paramètre de lissage f qui donne la fraction (entre 0 et 1) du jeu de données qui doit être couverte par la fenêtre de lissage. Plus f est grand, plus l'ajustement est lisse. Les polynômes locaux ajustés à chaque sous-ensemble des données sont presque toujours du premier ou du deuxième degré, c'est-à-dire qu'on fait des ajustements localement linéaires ou localement quadratiques. Utiliser un polynôme de degré zéro donnerait une moyenne mobile pondérée. S'il peut être adapté pour certaines situations, un modèle aussi simple risque de ne pas toujours approximer la fonction de façon satisfaisante. Des polynômes de degré plus élevés devraient marcher en théorie mais ils conduisent à des modèles qui ne sont pas vraiment dans l'esprit du lowess. Le lowess est fondé sur l'idée que toute fonction peut être correctement approchée dans un petit voisinage par un polynôme d'ordre faible et que des modèles simples peuvent être ajustés aux données facilement. Des polynômes de degré élevé auraient tendance à trop s'ajuster aux données dans chaque sous-ensemble, et seraient donc instables numériquement, rendant les calculs précis difficiles. Ils introduiraient alors une variabilité indésirable dans le processus de normalisation.

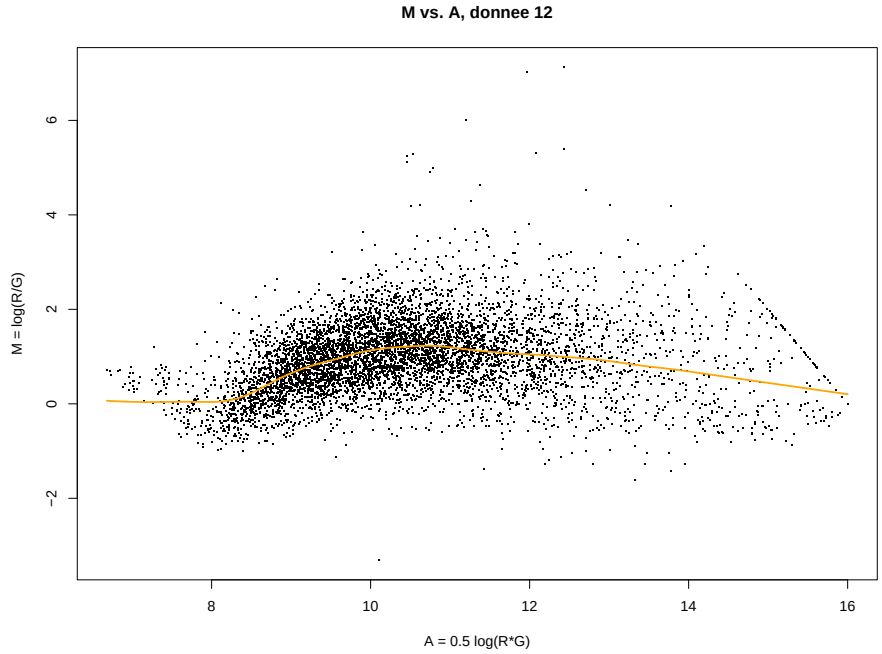

Figure 1.7 – Représentation (A,M) avant toute normalisation des données. La courbe tracée est la fonction de normalisation choisie quand on effectue une normalisation par lowess. On peut voir que cette fonction suit bien la forme du nuage.

Le lowess étant local, il s'adapte bien à la courbure du nuage de points (cf figure 1.7). La normalisation lowess va permettre d'estimer ρ et donc, après normalisation,

produire un nuage plat centré sur zéro. On voit *a priori* que la normalisation lowess sera préférable à une normalisation par la médiane.

Remarque : figure 1.8 obtenue en corrigeant le bruit de fond par approche bayésienne

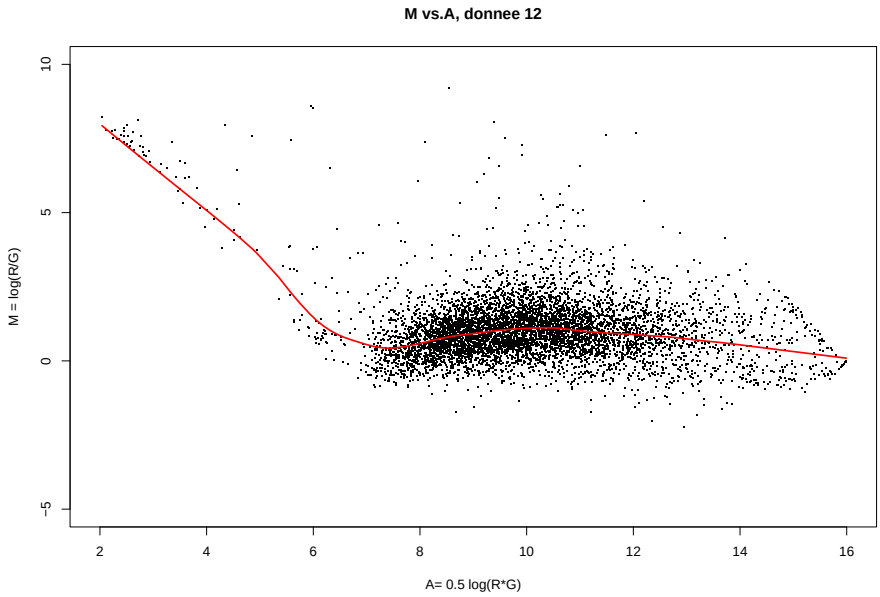

Figure 1.8 – Représentation (A,M) avant toute normalisation des données, en ayant corrigé par le bruit de fond. La courbe tracée est la fonction de normalisation choisie quand on effectue une normalisation par lowess. On peut voir que cette fonction suit bien la forme du nuage.

Des améliorations pour la normalisation lowess

Il existe plusieurs façons, compatibles entre elles, d'améliorer cette méthode de normalisation.

- Le problème des points saturants

La première amélioration qu'on peut apporter c'est de ne pas prendre en compte les spots saturants dans le calcul de la fonction lowess. En effet, les logiciels d'analyse d'image ne peuvent pas mesurer les valeurs d'intensité lumineuse supérieures à 65 535 ($= 2^{16} - 1$). Le logiciel attribue cette valeur d'intensité à tous les pixels qui ont une intensité plus grande. Le logiciel donne aussi en sortie le pourcentage de pixels saturés. On peut raisonnablement penser qu'au-delà d'un certain pourcentage de pixels saturés (on prendra un seuil de 20%), l'estimation de la médiane de

la fluorescence du spot est biaisée. Par conséquent, ces points qu'on appelle saturants sont moins fiables que les autres points, on ne peut donc pas les utiliser de la même façon. Ces points dévient la fonction de normalisation dans la mesure où leur représentation graphique est limitée et ne correspond pas à leur vraie fluorescence (cf figure 1.9). Pour qu'ils ne faussent pas les calculs, on ne les considère donc pas lors du calcul de la fonction lowess de normalisation. Par contre, on leur applique aussi la normalisation car ces points sont intéressants, on veut y extraire la valeur des $\log_2$ ratios normalisés.

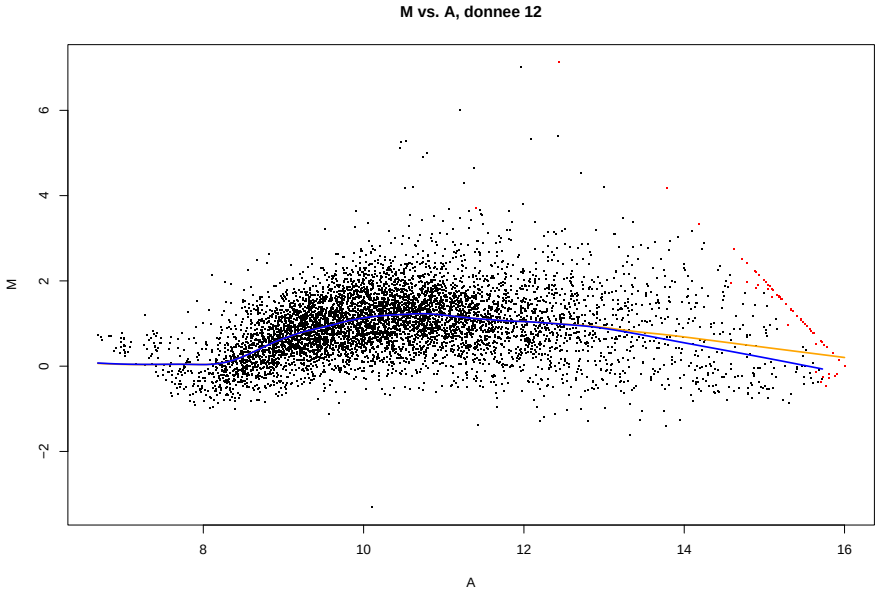

Figure 1.9 – Comparaison des fonctions de normalisation en considérant les points saturants (courbe orange) et sans les considérer (courbe bleue). Les points saturants sont représentés en rouge. On voit que la déviation entre les deux fonctions est significative.

- La normalisation par bloc

Une autre façon d'améliorer la normalisation lowess est de calculer une fonction de normalisation par bloc. Sur une biopuce, chaque bloc est imprimé en utilisant la même tête d'impression. On peut alors se demander si la tête d'impression n'entraîne pas des variations systématiques dans nos expériences. Pour vérifier s'il y a des différences entre les blocs, on normalise nos données en calculant une fonction de normalisation lowess pour chaque bloc, comme suggéré par l'équipe de Terry Speed [49]. Les résultats sont assez hétérogènes d'une expérience à l'autre. Sur la figure 1.10, on voit quelques résultats pour trois jeux de données qui se révèlent particulièrement significatifs. Pour la plupart des jeux de données, les douze courbes de normalisation calculées sont assez proches les unes des autres. Par contre, pour certaines lames,

quelques courbes (une ou deux) ressortent de l'ensemble des courbes calculées. Ceci montre qu'il peut exister un effet de bloc. On va vérifier de façon statistique qu'il existe bien une différence significative entre les fonctions de normalisation estimées sur les différents blocs.

On avait le modèle

$$M_i \;=\; \rho(A_i)$$

On va maintenant écrire le modèle obtenu quand on considère une normalisation par bloc :

$$M_{ki} \;=\; \rho_k(A_{ki})$$

où k est le numéro du bloc, et A_{ki} et M_{ki} désignent respectivement la $\log_2$-intensité totale moyenne et le $\log_2$-ratio pour le $i^{ème}$ gène du $k^{ème}$ bloc.

La question qui se pose ici est de savoir si les fonctions ρ_k sont statistiquement différentes. Pour faire ce test, on va utiliser des méthodes d'analyse de variance factorielle (FANOVA) qui fournissent une extension des modèles ANOVA aux données fonctionnelles, permettant toujours une interprétation simple. Comme les données fonctionnelles sont présentes dans nombre d'applications, les modèles FANOVA ont vu leur popularité croître et la littérature est prolixe à ce sujet. On donnera notamment comme référence [36] et [43]. Si la littérature s'est beaucoup intéressée à la façon d'ajuster les modèles FANOVA et d'estimer leurs composantes, peu d'articles se sont intéressés au développement de procédures de test dans les modèles FANOVA, ce qui est pourtant le problème d'intérêt ici, puisqu'il s'agit de tester si les fonctions de normalisation sont différentes d'un bloc à l'autre. Ce problème est abordé de façon assez exhaustive par Abramovich *et al.* dans [1]. En suivant cette article, on écrit :

$$M_{ki} = \rho_k(A_{ki})dA_{ki} + \varepsilon dW_k(A_{ki}), k = 1 \dots K$$

où K désigne le nombre total de blocs sur la biopuce, ε une constante positive, ρ_k est la fonction réponse inconnue que l'on estime par un lowess et W_k sont des processus de Wiener standard en unidimensionnel. Les fonctions m_k admettent la décomposition unique suivante :

$$\rho_k(A) = m_0 + \mu(A) + \alpha_k + \gamma_k(A), \quad k = 1 \dots K$$

où m_0 est une fonction constante, $\mu(A)$ est soit la fonction nulle soit une fonction non-constante de A (le principal effet de A), α_k est soit zéro soit une fonction non-constante de k (le principal effet du bloc k) et $\gamma_k(A)$ est soit zéro soit une fonction non nulle qui ne peut pas être décomposée en somme d'une fonction de i et d'une fonction de A (la composante d'interaction).

Ainsi, ici, l'effet de bloc est décomposé en deux parties : une partie constante α_k et une partie qui est aussi fonction de A, $\gamma_k(A)$; $\gamma_k(A)$ représente l'effet centré du groupe k.

Dire qu'il n'y a aucun effet de bloc c'est tester les hypothèses $H_0 : \alpha_k = 0,\ k = 1 \dots K$ et $H_0 : \gamma_k(A) = 0,\ k = 1 \dots K$, et les rejeter toutes deux. Une méthode pour

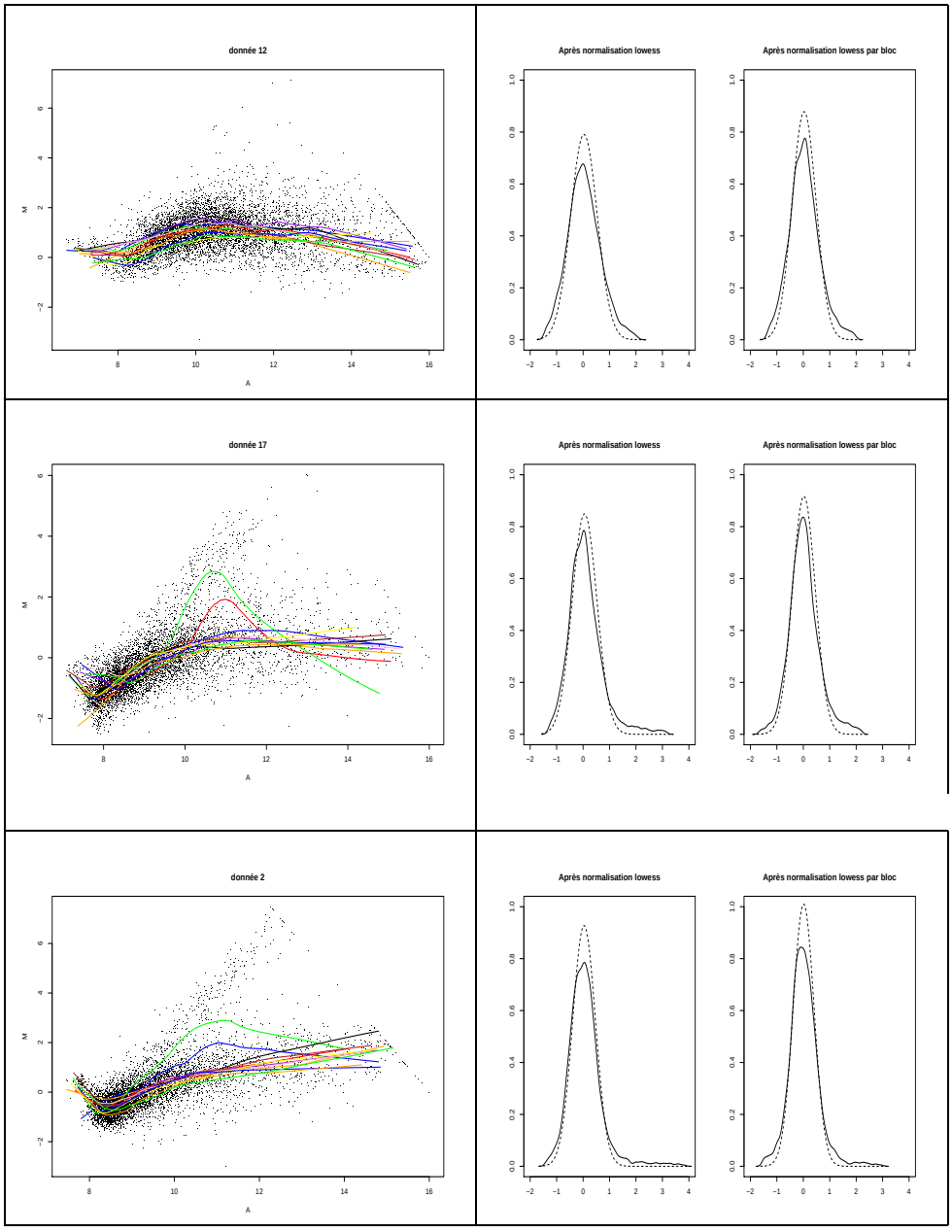

Figure 1.10 – Normalisation par bloc pour trois jeux de données (donnée 12, 17 et 2). **Colonne de gauche** : M vs. A avant normalisation. Les douze lignes colorées sont les courbes lowess pour chacun des blocs. Pour donnée 12 : aucune courbe ne ressort particulièrement. Pour donnée 17 : 2 courbes ressortent (bloc 3 en vert et bloc 4 en rouge). Pour donnée 2 : 2 courbes ressortent (bloc 1 en bleu et bloc 3 en vert). **Colonne de droite** : *en trait plein :* l'estimation lissée de la densité des $\log_2$-ratios après normalisation ; *en pointillés :* la distribution gaussienne correspondante (de même moyenne et écart-type) qu'on aimerait donc approcher ; *à gauche :* après normalisation lowess ; *à droite :* après normalisation lowess par bloc. Dans les deux jeux de données qui présentent des différences entre les courbes de normalisation des blocs, les queues des distributions sont réduites avec la normalisation par bloc. Pour donnée 12, on ne voit pas vraiment de différence.

estimer ces composantes au moyen d'ondelettes et pour effectuer le test est proposé dans [1], il ne sera pas présenté ici, on donnera seulement les résultats en appliquant la méthode proposée à différents jeux de données.

Pour être en mesure d'appliquer la méthode, il faut avoir une estimation des fonctions ρ_k pour les mêmes valeurs de A quelque soit le bloc. Pour cela, on estime une courbe lowess pour le nuage de points correspondant à chaque bloc de la biopuce. Mais on obtient alors une courbe lowess définie sur un ensemble qui peut différer légèrement d'un bloc à l'autre. Pour avoir des estimations sur le même ensemble de définition, on estime ces fonctions lowess de chaque bloc sur une grille de points qui parcourt l'ensemble de l'intersection des différents ensembles de définition de ces fonctions, en prenant un nombre de valeurs égal à une puissance de deux (en pratique, on a pris : $2^{12} = 4096$ valeurs différentes pour A). On peut alors commencer l'estimation des différentes composantes du modèle FANOVA. Avant tout, il faut disposer d'une estimation du paramètre ε. ε représente la variabilité des données autour de la courbe de normalisation lowess pour le bloc. On peut l'estimer en calculant la moyenne des résidus au carré après une normalisation par bloc.

Avant de procéder aux tests, on retrouve graphiquement avec l'estimation des γ_k qu'il semble exister des différences entre les blocs. Ainsi, sur les figures 1.11, 1.12 et 1.13, on représente les fonctions $\gamma_k(A)$ trouvées pour chacun des K blocs pour les 3 jeux de données déjà présentés sur la figure 1.10.

Par la suite, on effectue les tests des hypothèses $H_0 : \alpha_k = 0,\ k = 1 \dots K$ et $H_0 : \gamma_k(A) = 0,\ k = 1 \dots K$. Les résultats sont résumés dans le tableau 1.1

		donnee12	donnee17	donnee2
ε		0.4396	0.4792	0.5297
α_k	Stat test	10000.7687	4229.8811	17765.0909
	Seuil	9.488		
	Conclusion	H_0 rejetée		
$\gamma_k(A)$	Stat test	135.0342	254.4963	266.5154
	Seuil	1.0475	0.9418	0.7822
	Conclusion	H_0 rejetée		

Table 1.1 – Tableau des résultats obtenus pour les tests sur 3 jeux de données avec un seuil $\alpha = 0.05$

On déduit de ce tableau que les courbes de normalisation calculées sur les différents blocs sont significativement différentes, même si cela n'était pas évident graphiquement pour certains jeux de données comme "donnée 12". Ainsi, pour éviter que certains blocs ne faussent l'ensemble des résultats (élimination de l'effet de bloc), il paraît judicieux, dans le cadre d'une normalisation lowess, d'effectuer un calcul de courbe de normalisation différente pour chaque bloc.

Il est aussi intéressant de préciser que si on teste l'hypothèse $\mu(A) = 0$ contre $\mu(A) \neq 0$, on accepte alors $\mu(A) \neq 0$, ce qui confirme qu'une normalisation tenant

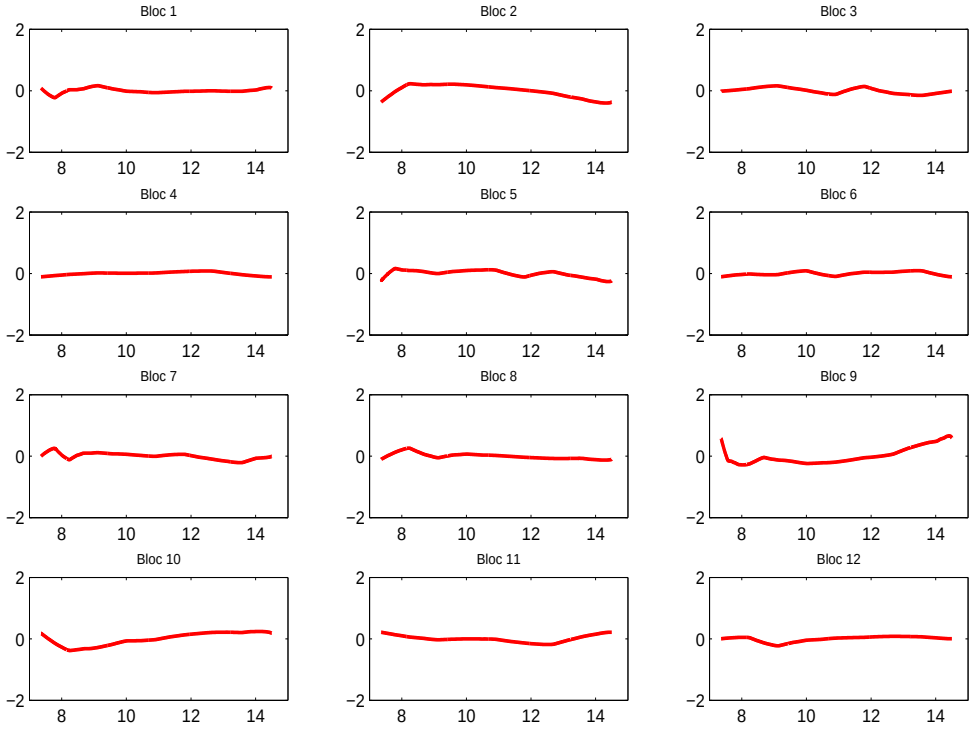

Figure 1.11 – Estimation des fonctions $\gamma_k(A)$ pour chaque bloc pour donnée 12. Ici aucune courbe ne ressort particulièrement.

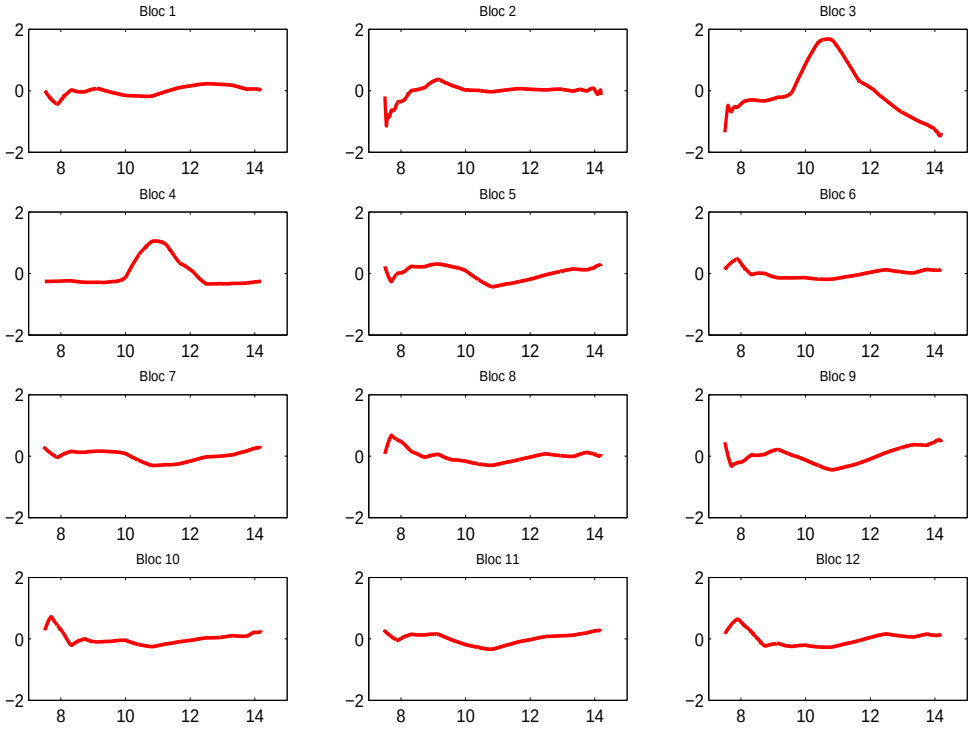

Figure 1.12 – Estimation des fonctions $\gamma_k(A)$ pour chaque bloc pour donnée 17. Ici, on voit nettement ressortir à nouveau les courbes des blocs 3 et 4.

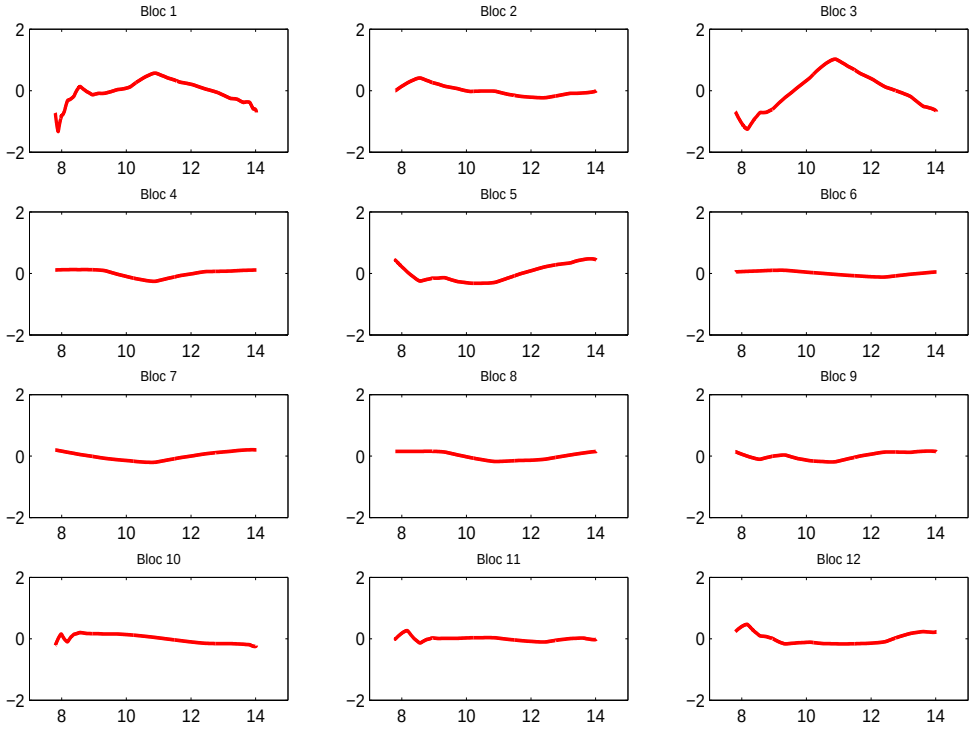

Figure 1.13 – Estimation des fonctions $\gamma_k(A)$ pour chaque bloc pour donnée 2. Ici, on voit à nouveau ressortir la courbe du bloc 1 et légèrement celle du bloc 3.

compte de l'intensité moyenne totale sera judicieuse.

- L'ajustement d'échelle

Après avoir effectué une normalisation lowess par bloc, on a, pour chaque bloc, des $\log_2$-ratios normalisés centrés en zéro. Mais cela ne garantit pas que la dispersion des $\log_2$-ratios est la même pour chaque bloc, c'est pourquoi il est alors nécessaire de procéder à un ajustement d'échelle. Après normalisation, on aimerait approcher la loi des observations par une distribution gaussienne, et nos résultats ont montré que les $\log_2$-ratios normalisés en étaient assez proches. Nous allons donc supposer que, lorsqu'il n'y a ni sur-expression, ni sous-expression (donc pour la majorité des spots), les $\log_2$-ratios suivent une loi gaussienne de moyenne zéro et de vraie variance inconnue σ^2. On suppose maintenant que chaque bloc suit une loi gaussienne centrée en zéro et de variance ${a_k}^2\sigma^2$ où ${a_k}^2$ est un facteur d'échelle pour le $k^{ème}$ bloc. Pour faire un ajustement d'échelle entre les blocs, on doit d'abord estimer a_k. Pour cela, on utilise l'estimateur de maximum de vraisemblance. La théorie de l'analyse de variance nous donne la contrainte :

$$\sum_{k=1}^{K} log_2(a_k)^2 = 0 \; avec \; K \; le \; nombre \; total \; de \; blocs$$

ce qui signifie aussi :

$$\prod_{k=1}^{K} {a_k}^2 = 1 \; avec \; K \; le \; nombre \; total \; de \; blocs$$

En calculant l'estimateur de maximum de vraisemblance $\widehat{a_k}$ pour chaque a_k, on obtient :

$$\widehat{a_k^2} = \frac{\sum_{i=1}^{p_k} M_{ki}^2}{\sqrt[K]{\prod_{j=1}^{K} \sum_{i=1}^{p_j} M_{ji}^2}}$$

où M_{ki} désigne le $i^{ème}$ $\log_2$-ratio du $k^{ème}$ bloc, avec p_k le nombre de gènes dans le $k^{ème}$ bloc (dans nos expériences, p_k = constante).
Mais on préfère utiliser une alternative robuste de cet estimateur :

$$\widehat{a_k} = \frac{MAD_k}{\sqrt[K]{\prod_{j=1}^{K} MAD_j}}$$

où MAD_k est l'écart absolu médian (Median Absolute Deviation) défini par :

$$MAD_k = median_i\{|M_{ki} - median_j(M_{ki})|\}$$

MAD_k est un estimateur robuste de l'écart-type de M_{ki}. Une fois que l'on a les estimateurs $\widehat{a_k}$, on peut faire l'ajustement d'échelle :

$$M_{ki}^{'} = \frac{1}{\widehat{a_k}} M_{ki}$$

Sur la figure 1.14, on voit les résultats de cet ajustement d'échelle pour quelques jeux de données. La visualisation de l'estimation lisse de la densité montre que dans quelques cas, l'ajustement d'échelle permet de réduire considérablement les queues de distribution. Il permet d'avoir une dispersion beaucoup plus homogène d'un bloc à l'autre. C'est encore une amélioration de la normalisation lowess dans la mesure où cela permet de ne pas donner de prépondérance à certains blocs.

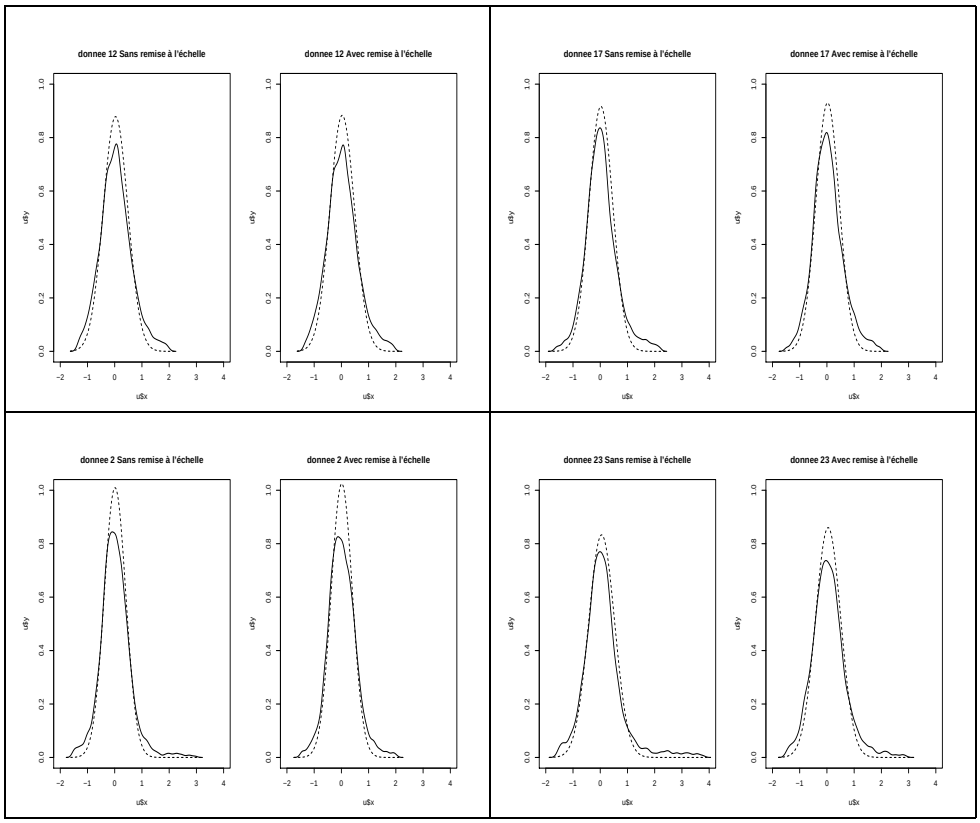

Figure 1.14 – Estimation lissée de la densité après ajustement d'échelle pour quatre jeux de données. Pour chaque graphique, *en trait plein :* l'estimation lissée de la densité des $\log_2$-ratios après les calculs de normalisation ; *en pointillés :* la distribution gaussienne correspondante (qu'on souhaiterait approcher) ; *à gauche :* après normalisation lowess par bloc ; *à droite :* après normalisation lowess par bloc avec ajustement d'échelle. On peut voir une réelle amélioration de la queue de distribution pour donnée 2 et donnée 23. Il n'y a pas de changement important pour donnée 12 et donnée 17.

1.1.3.3 Normalisation par Normal Score (Yi Lin, Samuel T. Nadler, Alan D. Attie, Brian S. Yandell)

Cette normalisation résulte d'une approche assez différente. Au lieu d'essayer de centrer les $\log_2$-ratios, elle propose de faire directement une transformation sur $(R_i)_{i=1..p}$ et $(G_i)_{i=1..p}$. En fait, on fait une transformation qui donne une distribution quasiment normale centrée réduite des données pour chacune des fluorescences. Cela revient plus ou moins à dire qu'on doit avoir globalement la même distribution dans les deux couleurs puisque la grande majorité des gènes ne sont pas différentiellement exprimés. On fait donc une transformation « normal score » c'est-à-dire qu'on prend :

$$NS(R_i) = NormalScore(R_i) = \Phi^{-1}\left(\frac{\widetilde{R_i}}{p+1}\right)$$

où Φ^{-1} est la fonction réciproque de Φ, fonction de répartition d'une loi normale centrée réduite et $\widetilde{R_i}$ désigne la statistique de rang associé à R_i. Cette statistique correspond au rang de R_i dans l'échantillon ordonné dans le sens des valeurs croissantes.

Habituellement, on calcule la statistique de rang avec la formule

$$\widetilde{R_i}^1 = 1 + \sum_{j=1}^{p} \mathbb{1}_{\{R_i<R_j\}},$$

ce qui revient à dire que $\widetilde{R_i}$ est égale à 1 plus le nombre d'observations strictement inférieures à R_i.

Cette définition peut poser des problèmes en cas d'ex æquo. En effet, considérons l'exemple :

R_i	20	40	20	10	20
$\widetilde{R_i}$	2	5	2	1	2

On voit que telle qu'elle est donnée, la définition de la statistique de rang nous conduit à donner la plus petite valeur possible pour les ex æquo à savoir 2 alors qu'on aurait pu donner indifféremment la valeur 2, 3 ou 4. En cas de nombreux ex æquo cette politique de classement peut déséquilibrer cette statistique de rang. Ainsi, on aurait ici préféré donné la valeur moyenne possible, à savoir 3. Pour cela, on introduit une deuxième définition de la statistique de rang qui ne différera de la première que pour les cas d'ex æquo

$$\widetilde{R_i} = 1 + \sum_{j=1}^{p} \mathbb{1}_{\{R_i<R_j\}} + \frac{1}{2}\left[\sum_{j=1}^{p} \mathbb{1}_{\{R_i=R_j\}} - 1\right],$$

Pour l'exemple vu précédemment, on aura maintenant :

R_i	20	40	20	10	20
$\widetilde{R_i}$	3	5	3	1	3

On peut aussi noter au passage, qu'en cas d'un nombre pair d'ex æquo, on peut obtenir des rangs non entiers.

Cette histoire d'ex æquo n'est pas très importante, il s'agit d'avantage d'un souci d'adaptation de la méthode aux données de biopuces. En effet, si on observe des réalisations de variables aléatoires X_i, $i \in \{1, \ldots, p\}$ indépendantes et identiquement distribuées de loi continue, alors la probabilité d'avoir un cas d'ex æquo est nulle. Dans le cas des données de biopuces, comme il s'agit de données mesurées par un scanner, on peut observer des valeurs identiques notamment pour les valeurs les plus hautes et les plus basses à cause des limites de détection du scanner.

On rappelle ici la définition de la fonction Φ ; si X est une variable aléatoire de loi normale centrée réduite, on a :

$$\forall x \in \mathbb{R},\ \Phi(x) = \mathbb{P}(X \leq x)$$

La fonction Φ^{-1} est ce qu'on appelle la fonction quantile associé à la loi normale centrée réduite, on a

$$\forall \alpha \in [0,1], \Phi^{-1}(\alpha) = x_\alpha \Longleftrightarrow \Phi(x_\alpha) \ = \ \mathbb{P}(X \leq x_\alpha) = \alpha$$

Remarque. Si X est un vecteur quelconque de dimension p et de fonction de répartition F, et Y le vecteur obtenu en appliquant la méthode des « normal scores » à X ($Y_i \ = \ NS(X_i)$, $i = 1 \ldots p$) alors Y est un vecteur gaussien centré réduit.
PREUVE — En effet, on a alors :

$$Y_i = NS(X_i) = \Phi^{-1}\left(\frac{\widetilde{X_i}}{p+1}\right)$$

Or, ici, on est dans le cas d'une variable réelle, on peut donc considérer que l'on a $\widetilde{X_i} = 1 + \sum_{j=1}^{p} \mathbb{1}_{\{X_i < X_j\}} = \sum_{j=1}^{p} \mathbb{1}_{\{X_i \leq X_j\}}$.

Notons x_i l'observation de la variable X_i et $\tilde{x_i}$ celle de la statistique de rang associée.

On définit en général la fonction de répartition empirique de la variable aléatoire X en x, F_p, par

$$F_p(x) = \frac{1}{p} \sum_{j=1}^{p} \mathbb{1}_{\{x_j \leq x\}}$$

On a

$$\frac{\tilde{x_i}}{p+1} = \frac{\sum_{j=1}^{p} \mathbb{1}_{\{x_j \leq x_i\}}}{p+1} \xrightarrow[p \to \infty]{} F_p(x_i)$$

De plus, le théorème de Kolmogorov nous donne la convergence presque sûre de la fonction de répartition empirique vers la vraie fonction de répartition de X quand $p \to \infty$:

$$\forall x \in \mathbb{R},\ \ F_p(x) \xrightarrow[p \to \infty]{p.s} F(x)$$

D'ailleurs, on a même la convergence uniforme presque sûrement grâce au théorème de Glivenko-Cantelli, *i.e.* :

$$\forall x \in \mathbb{R},\ \sup |F_p(x) - F(x)| \xrightarrow[p\to\infty]{p.s} 0$$

On a donc aussi

$$\forall i \in 1 \dots p,\ \frac{\tilde{x_i}}{p+1} \xrightarrow[p\to\infty]{} F(x_i)$$

Déterminons maintenant la fonction de répartition de Y :

$$\mathbb{P}(Y_i \leq t) = \mathbb{P}\left(\frac{\widetilde{X_i}}{p+1} \leq \Phi(t)\right) \xrightarrow[p\to\infty]{} \mathbb{P}(F_p(X_i) \leq \Phi(t))$$

Puisque la convergence presque sûre implique aussi la convergence en loi, on a

$$\mathbb{P}(F_p(X_i) \leq \Phi(t)) \xrightarrow[p\to\infty]{} \mathbb{P}(F(X_i) \leq \Phi(t))$$

De plus :

$$\begin{aligned} \mathbb{P}(F(X_i) \leq \Phi(t)) &= \mathbb{P}(X_i \leq F^{-1}(\Phi(t))) \\ &= F(F^{-1}(\Phi(t))) \\ &= \Phi(t) \end{aligned}$$

Par conséquent :

$$\mathbb{P}(Y_i \leq t) \xrightarrow[p\to\infty]{} \Phi(t)$$

D'où :

$$Y_i \sim N(0,1)\ \textit{quand}\ p \to \infty$$

. ⊗

Sur la figure 1.15, on peut visualiser les différentes étapes de la transformation normal scores sur un exemple. Quand les données de départ sont à peu près log-normales alors cette transformation normal scores est très proche d'une transformation logarithmique. Au lieu de considérer le nuage, M vs. A comme on le faisait avec la transformation log, on va considérer le nuage Msco vs Asco où :

$$Asco = NS(R_i) + NS(G_i)$$

$$Msco = NS(R_i) - NS(G_i)$$

Sur la figure 1.16, on voit un exemple de nuage de points obtenu. Le résultat semble assez concluant, le nuage de points perd sa forme incurvée.

Cette méthode présente l'avantage d'être simple tout en donnant des résultats sous une forme intéressante. De plus, il est assez répandu de travailler sur des intensités de fluorescence auxquelles on a retiré le bruit de fond, et cela peut mener à des

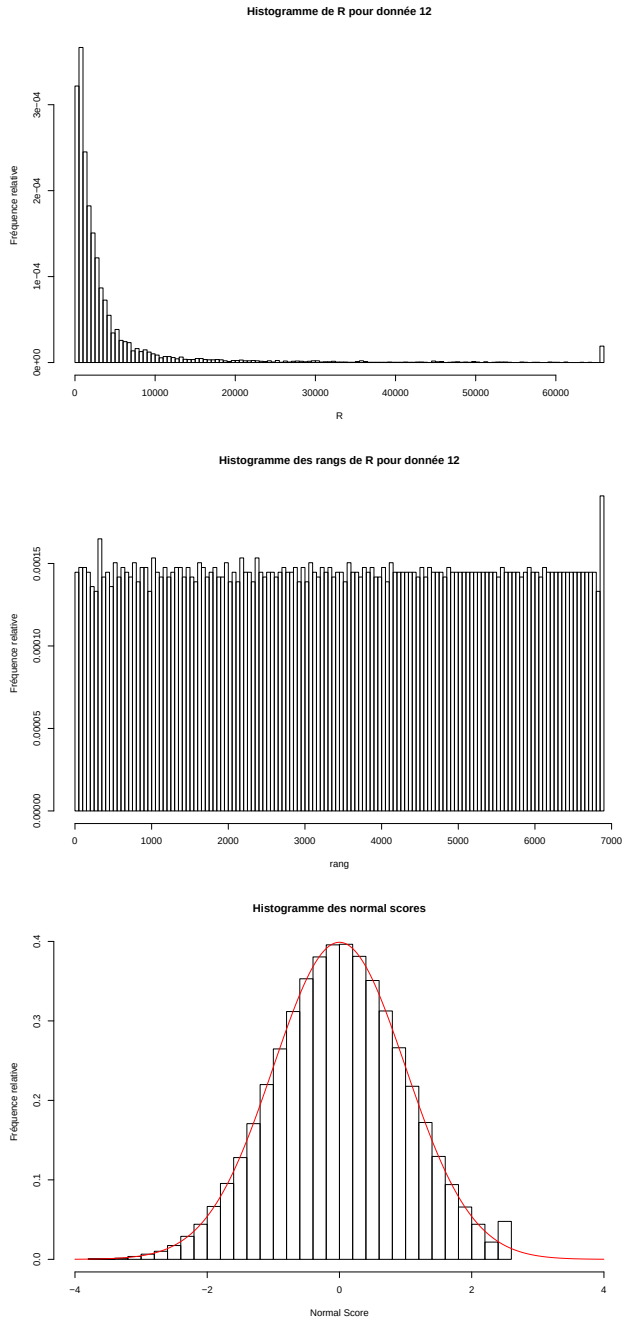

Figure 1.15 – Détail des étapes de la transformation “normal scores” sur un exemple, le canal rouge de la biopuce donnée12, vecteur de données noté R. *Première figure :* histogramme de la distribution de R avant toute transformation. On remarque un pic pour la dernière barre, qui correspond aux grandes valeurs de R pour lesquelles le scanner est arrivé à saturation. *Deuxième figure :* histogramme des rangs calculés à partir de R. *Troisième figure :* distribution du résultat de la transformation “normal scores”. La courbe représente la densité gaussienne centrée réduite. A part un petit pic pour les grandes valeurs dû à des problèmes de saturation, on remarque que la distribution du résultat est quasiment normale centrée réduite.

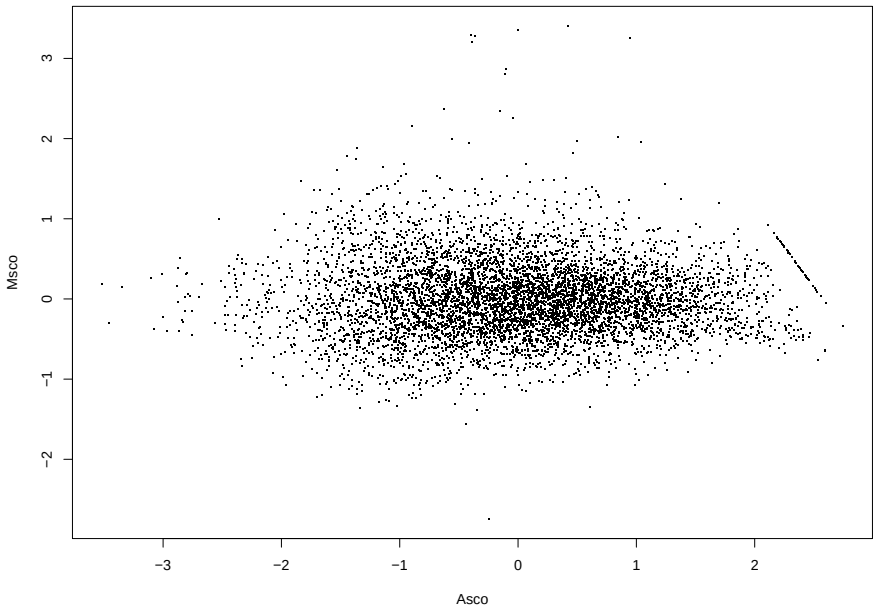

Figure 1.16 – Représentation du nuage de points pour donnée12 après normalisation par la méthode des scores.

valeurs négatives, ce qui ne constitue pas du tout un problème pour cette technique, dans la mesure où seul compte l'ordre dans lequel sont rangées les données.

On peut aussi essayer d'adapter les idées d'amélioration proposées pour la normalisation lowess à la normalisation par les scores. En ce qui concerne les points saturants, il n'est pas utile ici d'en tenir compte de façon spécifique ; en effet, dans la normalisation lowess, ces points déviait la courbe de normalisation, mais pour la méthode des scores, on tient juste compte de l'ordre dans lequel sont rangées les valeurs observées. Ainsi les points saturants ne gênent en rien la normalisation.

On a mis en évidence lors de la normalisation lowess des différences parfois importantes entre les blocs, ces différences sont intrinsèques au jeu de données. Par conséquent on doit aussi en tenir compte ici, et faire une normalisation normal scores par bloc avec remise à l'échelle entre les blocs. En effet, de façon empirique, on observe des différences entre les blocs pour la distribution de M_{sco}.

1.1.3.4 Une autre étape de normalisation : lissage des résidus ou standardisation

Il existe une dernière étape de normalisation : le lissage des résidus. Quand on observe les nuages de points après normalisation on se rend compte que la dispersion des données est souvent fonction de l'intensité totale moyenne. Ceci signifie que si par la suite on ne s'intéresse qu'aux ratios les plus petits ou les plus grands, on risque de n'obtenir que des points situés dans la zone d'intensité totale moyenne pour laquelle les données sont les plus dispersées. Or on veut pouvoir détecter les gènes différentiellement exprimés indépendamment de l'intensité totale moyenne. Pour cela, il faut égaliser la dispersion des données en fonction de cette intensité totale. C'est ce qu'on appelle le lissage des résidus. On calcule les carrés des log-ratios (ou équivalent avec normal score) normalisés c'est-à-dire les $M_{ij}^{'\,2}$. Ces carrés nous donnent une estimation de la variance dans la mesure où les données sont centrées. On peut alors tracer le nuage de points qui donne les résidus en fonction de l'intensité moyenne totale. On calcule par lowess une fonction de lissage $\rho_{residus}$ de ces résidus. On est alors en mesure de corriger nos log-ratios normalisés pour avoir une dispersion qui ne dépend plus de l'intensité totale moyenne. Un exemple de résultat de lissage de résidu est donné figure 1.17

$$M_{i\ apres\ lissage\ residus} = \frac{M_{i\ avant\ lissage\ residus}}{\sqrt{\rho_{residus}(A_i)}} \quad , \quad i = 1..p.$$

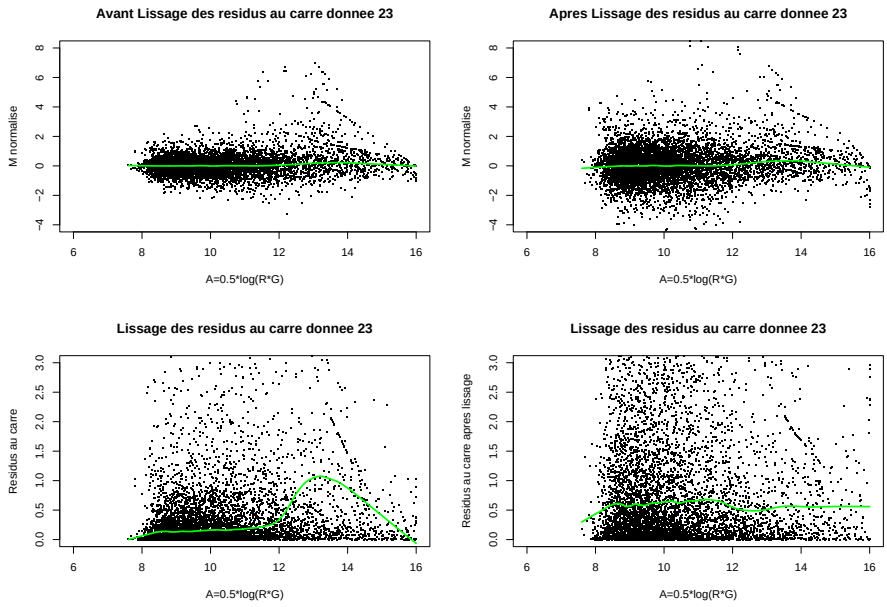

Figure 1.17 – Effet du lissage des résidus - donnée 23. En haut : nuage (A,M) avant et après lissage et tracé du lissage lowess du nuage (les données ont été préalablement normalisées avec une méthode lowess par bloc avec remise à l'échelle. En bas : nuage des résidus au carré (carré des $\log_2$-ratios normalisés) et tracé du lissage lowess associé. A droite : avant lissage des résidus. A gauche : après lissage des résidus. On voit que la variance des $\log_2$-ratios normalisés dépend de l'intensité A (en bas à gauche) et qu'après lissage des résidus cela a été corrigé (en bas à droite)

Conclusion

Nous avons donc présenté trois grands types de normalisation possibles et les différentes variations que l'on pouvait effectuer sur chacune d'elle. La normalisation est une étape décisive de l'analyse des données issues de biopuces à ADN et le choix d'une méthode ne doit pas être fait à la légère. Mais comment choisir une méthode ? C'est ce que nous allons voir dans la partie suivante.

Simulation de données issues des biopuces à ADN

Introduction

Afin de pouvoir comparer les différentes procédures de normalisation et de détection des gènes différentiellement exprimés, il est indispensable de pouvoir simuler des données afin de valider les différentes techniques. Si on sait au départ quels gènes on doit détecter, il sera plus facile de mesurer l'erreur commise dans le traitement des données. Dans cette partie, on va proposer un modèle de simulation de données issues de biopuces à partir de jeux de données existants. On va ensuite utiliser ce modèle sur les données dont nous disposons, mais avec des données ayant subi des traitements différents. Dans un premier temps, on utilisera des données non corrigées par le bruit de fond pour estimer les paramètres de notre modèle et simuler des données. Dans un deuxième temps, on estimera les paramètres du modèle à partir des jeux de données où la méthode bayésienne de correction par le bruit de fond a été utilisée (*cf* 1.1.2). Tout au long, de la présentation du modèle utilisé, les illustrations se réfèreront au cas des données non corrigées par le bruit de fond. Les jeux de données obtenus par simulation nous permettront ensuite de choisir le traitement du bruit de fond et la méthode de normalisation les plus adaptés en connaissance de cause.

1.2.1 Modélisation des formes des nuages

Pour trouver un modèle de simulation de données issues de biopuces, il faut commencer par observer les jeux de données dont nous disposons. On regarde les graphes du log-ratio M en fonction de la log-intensité moyenne A (cf figure 1.18). Sur cette figure, on représente aussi le lissage lowess du nuage de points. On peut constater que les formes générales des nuages et par conséquent des lissages lowess sont bien distinctes.

Il est important de tenir compte de cette diversité. L'idée de départ pour construire un modèle de simulation des données issues de biopuces, c'est donc d'arriver à générer une série de formes de nuages différentes en identifiant la loi des courbes de forme pour les nuages réels. Pour cela, on a essayé d'utiliser un modèle paramétrique, c'est-à-dire qu'on essaie d'approcher les courbes de forme de nuage (obtenues par

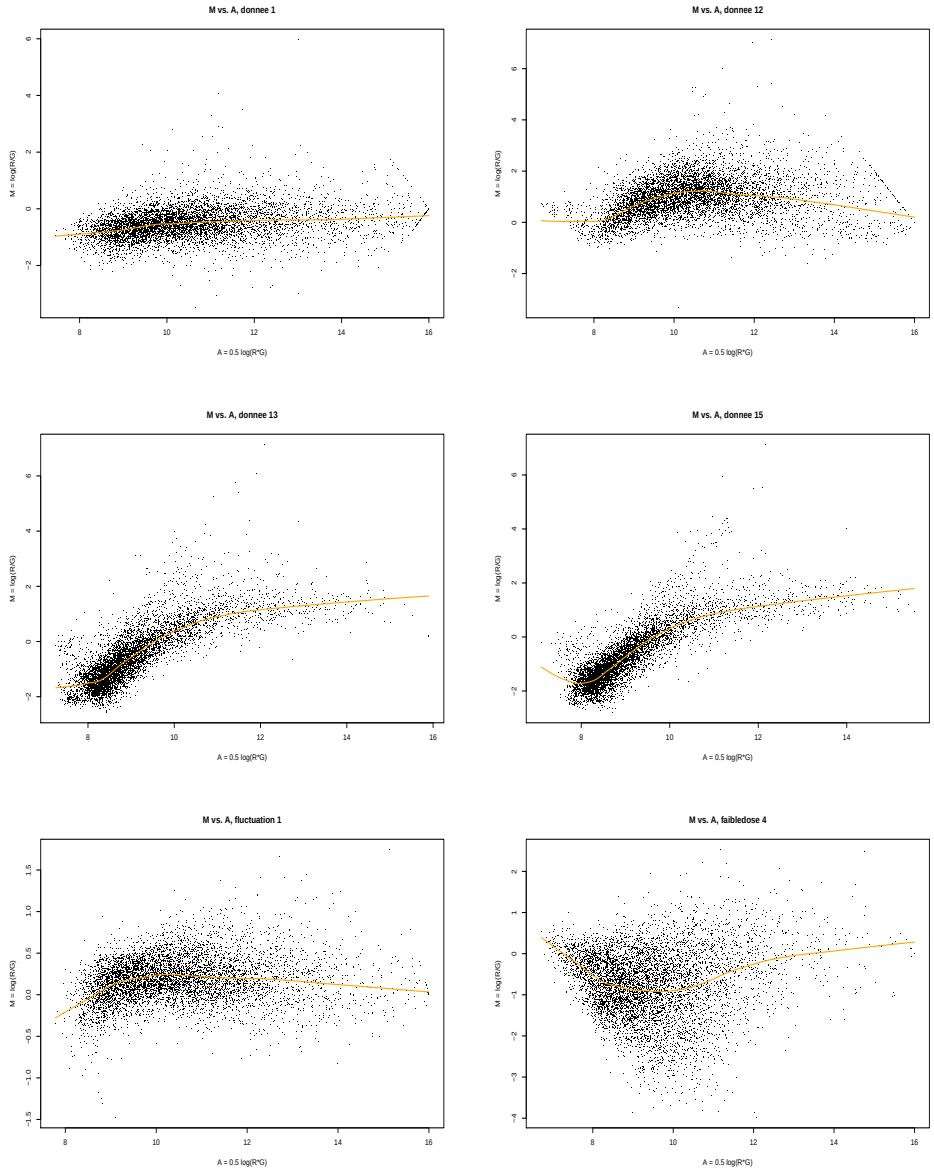

Figure 1.18 – Des formes variées pour les jeux de données réels

lowess) par une fonction appartenant à une famille paramétrée de fonctions. Il faut donc définir cette famille de fonctions, de façon à pouvoir retrouver la diversité de formes des données réelles.

1.2.1.1 Choix d'une fonction

Une famille de fonctions a été choisie pour ses capacités à bien approcher les différentes formes de nuage rencontrées. Il s'agit d'une fonction souvent appelée « fonction baignoire », très utilisée en fiabilité, et qui admet pour expression :

$$\rho(x, \boldsymbol{\theta}) = (x - x_0)^{\alpha} e^{-\beta(x - x_1)} - k$$

où $\boldsymbol{\theta} = (x_0,\ \alpha,\ \beta,\ x1,\ k)$ représente le vecteur des cinq paramètres de la fonction.

La procédure va être la suivante : pour chaque jeu de données réel, on va estimer les 5 coefficients de façon à approcher au mieux le lowess associé. On obtiendra alors une série d'observations de points dans $\mathbb{R}^5$. Au départ, pour un jeu de données, on dispose donc d'un nombre p de points de $\mathbb{R}^2$ qui représentent les couples (a_i, m_i) positionnés sur la courbe lowess calculée pour le nuage de points initial. C'est à partir de ces p points qu'on va chercher la valeur des paramètres (c'est-à-dire un point de $\mathbb{R}^5$) de la courbe « baignoire » qui va donner l'approximation la plus satisfaisante. Pour estimer chacun de ces points de $\mathbb{R}^5$, on utilise une méthode de moindres carrés non linéaires.

On définit notre modèle de régression non linéaire par :

$$M_i = \rho(A_i, \boldsymbol{\theta}) + \varepsilon_i,$$

où on supposera que les ε_i sont des variables aléatoires indépendantes et identiquement distribuées de loi de moyenne nulle et de variance σ^2 inconnue.

On cherche un estimateur $\hat{\boldsymbol{\theta}}$ du vecteur des paramètres θ, on choisit de prendre un estimateur des moindres carrés. Un estimateur des moindres carrés de $\boldsymbol{\theta}$, quand il existe, est un vecteur aléatoire $\hat{\boldsymbol{\theta}}$ dont la réalisation minimise la fonction de perte suivante

$$\mathcal{L}_2(\theta) = \sum_{i=1}^{p} (m_i - \rho(A_i, \boldsymbol{\theta}))^2$$

Cette définition suggère que cet estimateur des moindres carrés n'existe pas nécessairement. Toutefois, dans la pratique, l'espace Θ auquel apppartient $\boldsymbol{\theta}$ peut souvent être supposé compact et dans ce cas, il existe au moins une valeur de $\boldsymbol{\theta}$ pour lequel ce minimum est atteint. Ici, il est tout à fait possible de restreindre l'espace des paramètres et ainsi de supposer Θ compact.

Ici, on a donc une fonction de perte de la forme :

$$\mathcal{L}_2(\boldsymbol{\theta}) = \sum_{i=1}^{p} \left[m_i - (a_i - x_0)^{\alpha} e^{-\beta(a_i - x_1)} + k \right]^2$$

Pour déterminer $\hat{\theta}$, on peut utiliser une méthode de Gauss-Newton ou une méthode de Newton-Raphson. Ici, on a utilisé le package « nls » déjà existant en

R qui permet de déterminer les estimateurs des moindres carrés non linéaires des paramètres d'un modèle non linéaire.

On va maintenant donner quelques exemples de cette approximation, sur les jeux de données réels qu'on avait utilisés figure 1.18 pour illustrer les différentes formes de nuages de points (cf figure 1.19). On voit sur cette figure que l'approximation est assez bonne, comme on pouvait s'y attendre c'est sur les bords qu'on arrive le moins bien à approcher la courbe. L'erreur résiduelle standard demeure assez faible, de l'ordre de 10^{-2}.

1.2.1.2 Etude de la loi conjointe de ces points de $\mathbb{R}^5$

Maintenant qu'on a estimé les paramètres pour approcher la forme des nuages, on va s'attacher à l'estimation de la loi de densité de ces coefficients dans $\mathbb{R}^5$. Notons X_0, A, B, X_1 et K les variables aléatoires dont x_0, α, β, x_1 et k sont des réalisations. Ce que l'on cherche à estimer, c'est la densité conjointe $f_{X_0,A,B,X_1,K}(x_0, \alpha, \beta, x_1, k)$ pour tout $(x_0, \alpha, \beta, x_1, k) \in \mathbb{R}^5$. Les variables X_0, A, B, X_1 et K ne sont pas indépendantes ce qui complique l'estimation de leur loi conjointe. En effet, cela signifie que l'on doit estimer leur loi conjointe dans $\mathbb{R}^5$ à partir d'un petit nombre d'observations (de l'ordre de la cinquantaine). Si les paramètres étaient indépendants, il suffirait d'estimer la loi marginale de chacun des paramètres pour connaître leur loi conjointe, égale au produit des marginales. En effet, si on note pour une variable aléatoire X, $f_X(x)$ sa densité marginale en x, on a la propriété suivante :

$$X_0,\ A,\ B,\ X_1 \text{ et } K \text{ sont indépendantes ssi}$$

$$\forall (x_0, \alpha, \beta, x_1, k) \in \mathbb{R}^5, f_{X_0,A,B,X_1,K}(x_0, \alpha, \beta, x_1, k) = f_{X_0}(x_0) f_A(\alpha) f_B(\beta) f_{X_1}(x_1) f_K(k)$$

C'est pour se ramener à des paramètres indépendants que l'on passe au préalable par une Analyse en Composantes Indépendantes (ACI) décrite notamment par Amato *et al.* dans [5]. Ainsi, on se ramène à l'estimation de cinq densités dans $\mathbb{R}$ au lieu de l'estimation d'une densité dans $\mathbb{R}^5$ avec le même nombre d'observations dans tous les cas. Le problème est donc grandement simplifié. En effet, une fois estimées les densités marginales indépendantes dans l'espace de projection de l'ACI, on pourra simuler de nouveaux points selon ces lois estimées toujours dans l'espace de projection, pour repasser ensuite, via la transformation inverse de l'ACI dans l'espace des coefficients.

L'ACI est une méthode statistique qui permet de transformer linéairement un vecteur aléatoire multidimensionnel X en un vecteur aléatoire Y dont les composantes sont aussi stochastiquement indépendantes l'une de l'autre que possible.

L'objectif de cette méthode est de trouver une matrice P telle que $Y = PX$, avec des composantes Y_i le moins dépendantes possibles. L'ACI donne des résultats significatifs quand la loi de probabilité de X est loin d'être gaussienne. Dans une ACI, on appelle matrice de mélange la matrice G qui est l'inverse (ou pseudo-inverse) de la matrice P. Contrairement à l'Analyse en Composantes Principales (ACP), les vecteurs de base g_i de G ne sont généralement pas mutuellement orthogonaux.

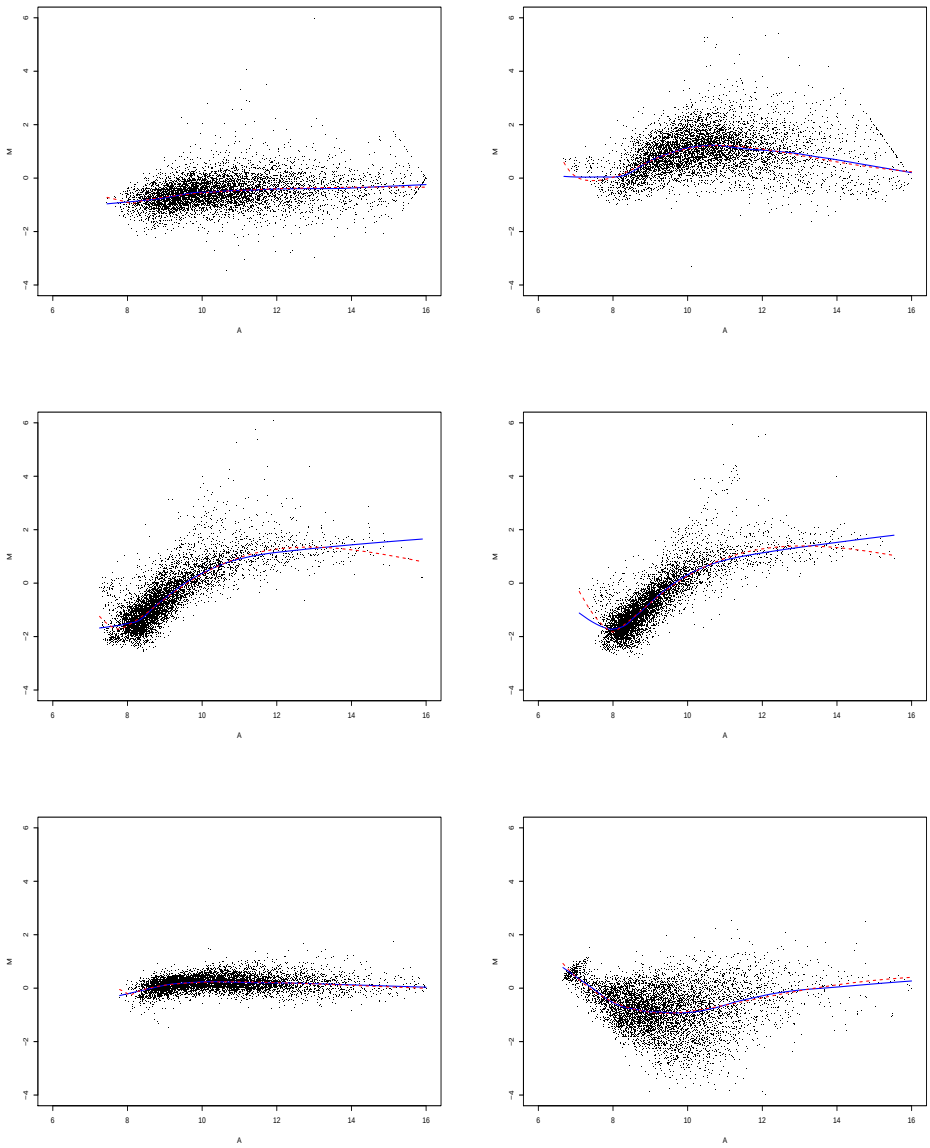

Figure 1.19 – Approche paramétrique de la courbe lowess Pour différents nuages de points, on calcule la fonction lowess qui donne la forme du nuage (trait plein). On approche ensuite ce lowess par une fonction paramétrique de type « baignoire »(trait pointillé). On voit que les courbes paramétriques s'adaptent bien aux différentes formes de nuage de points.

L'ACI repose sur le concept d'information mutuelle. On ne détaille pas ici la théorie de l'ACI. Disons simplement qu'on trouve la transformation linéaire pour l'ACI en minimisant l'entropie mutuelle entre le vecteur résultant de la transformation et le produit de ses marginales.

1.2.1.3 Simulation de nouvelles courbes

Une fois que l'ACI est faite, on peut estimer la densité des 5 composantes indépendantes. A partir de ces densités, on peut générer des points de $\mathbb{R}^5$ suivant les lois des 5 composantes indépendantes. A partir d'un point ainsi généré et de la matrice de mélange G obtenue en faisant l'ACI, on peut alors retrouver le point de $\mathbb{R}^5$ correspondant dans l'espace des coefficients. Ce point permet alors de définir une fonction « baignoire » qui donne la forme d'un nuage.

Une fois cette courbe obtenue, il faut encore lui faire subir une tranformation. En effet, cette courbe est une courbe qui donne le log-ratio M moyen en fonction de la log-intensité totale moyenne A par l'intermédiaire d'une fonction que nous appelerons f_{baig}. Or pour générer un nuage de points, on va générer les intensités lumineuses en Cy3 (R) et en Cy5 (V). Il faut donc avoir une courbe de forme de nuage dans cet espace là. Pour cela, on utilise la relation bijective que l'on a entre les deux couples de variables : (A, M) et (G, R).

On a vu que :

$$\begin{aligned} A &= \frac{log_2(R) + log_2(G)}{2} \\ M &= log_2(R) - log_2(G) \end{aligned}$$

Ce qui donne aussi :

$$\begin{aligned} log_2(G) &= A - \frac{1}{2}M \\ log_2(R) &= A + \frac{1}{2}M \end{aligned}$$

Par conséquent, si on appelle μ_R et μ_G les vecteurs des expressions moyennes des gènes non différentiellement exprimés en rouge et vert, et μ_A et μ_M ceux de A et M. On a la relation :

$$\mu_M = f_{baig}(\mu_A)$$

Ce qui donne entre $log_2(\mu_R)$ et $log_2(\mu_G)$:

$$log_2(\mu_G) - log_2(\mu_R) + f_{baig}\left(\frac{log_2(\mu_G) + log_2(\mu_R)}{2}\right) = 0 \qquad (1.3)$$

Par conséquent, une fois la fonction f_{baig} générée et un des deux vecteurs μ_G ou μ_R fixé, par exemple μ_G, on est en mesure de calculer l'autre vecteur μ_R et on aura la forme du nuage. Pour calculer ce deuxième vecteur, il faut trouver une racine de l'équation 1.3, cela peut parfois poser des problèmes mais en réduisant l'espace de recherche de racine ($log(\mu_R) \in [6, 16]$), on y arrive dans la majorité des cas.

1.2.2 Modèle de simulation de données

Dans la section 1.2.1, on a vu comment on pouvait modéliser la forme des nuages de façon à être en mesure de générer de nouvelles courbes de forme de nuage. Maintenant il faut modéliser l'aléa autour de ces courbes de forme pour passer d'une fonction à un nuage de points.

1.2.2.1 Présentation du modèle

Pour générer des données issues de biopuces, on a utilisé un modèle théorique, avec des termes d'erreur additive et multiplicative, proposé par Rocke et Durbin [38]. Un terme d'erreur additive important permet de modéliser de grandes variations pour les spots de faible intensité. Un terme d'erreur multiplicative important permet de modéliser de grandes variations pour les spots de grande intensité.

Le modèle s'écrit, pour le gène i :

$$\begin{aligned}(y_C)_i &= \alpha_C + \mu_{C,i} e^{\eta_{S,i}+\eta_{C,i}} + \varepsilon_{S,i} + \varepsilon_{C,i} \\ (y_T)_i &= \alpha_T + \mu_{T,i} e^{\eta_{S,i}+\eta_{T,i}} + \varepsilon_{S,i} + \varepsilon_{T,i}\end{aligned} \tag{1.4}$$

où l'indice S désigne une composante de la variance due au spot qui est commune aux deux échantillons sur la lame, et les indices C et T désignent les composantes de la variance spécifiques à l'échantillon C contrôle et à l'échantillon T test. $(y_E)_i$ est la mesure d'intensité et $(\mu_E)_i$ est le niveau d'expression pour l'échantillon E ($E \in \{C, T\}$) et pour le gène i, et α_E est l'intensité moyenne des gènes non exprimés pour l'échantillon E.

ε est le terme d'erreur additive et η le terme d'erreur multiplicative.

Pour un indice $E \in \{S, C, T\}$ et pour le gène i, on a :

$$\begin{aligned}\eta_{E,i} &\sim N(0, \sigma^2_{\eta_E}) \\ \varepsilon_{E,i} &\sim N(0, \sigma^2_{\varepsilon_E})\end{aligned}$$

1.2.2.2 Estimation des paramètres du modèle

Pour utiliser un tel modèle, il faut être en mesure d'en estimer les paramètres.

Estimation du bruit de fond et des $\sigma^2_{\varepsilon_E}$

Ce qu'on entend ici par bruit de fond c'est l'intensité moyenne des gènes non différentiellement exprimés, c'est-à-dire, dans le modèle proposé, les α_E ($E \in \{C, T\}$).

Quand le niveau d'expression est faible pour les deux échantillons, le modèle devient approximativement, pour le gène i :

$$\begin{aligned}(y_C)_i &\approx \alpha_C + \varepsilon_{S,i} + \varepsilon_{C,i} \\ (y_T)_i &\approx \alpha_T + \varepsilon_{S,i} + \varepsilon_{T,i}\end{aligned}$$

D'où, pour les moyennes et variances de y_C et y_T :

$$\begin{aligned} \overline{y_C} &\approx \alpha_C \\ \overline{y_T} &\approx \alpha_T \\ Var(y_C) &\approx \sigma^2_{\varepsilon_S} + \sigma^2_{\varepsilon_C} \\ Var(y_T) &\approx \sigma^2_{\varepsilon_S} + \sigma^2_{\varepsilon_T} \\ Var(y_C - y_T) &\approx \sigma^2_{\varepsilon_C} + \sigma^2_{\varepsilon_T} \end{aligned} \tag{1.5}$$

Ces équations permettent d'obtenir des estimateurs pour les paramètres de la composante additive de la variance. Il faut d'abord estimer $\overline{y_C}$, $\overline{y_T}$, $Var(y_C)$, $Var(y_Y)$ et $Var(y_C - y_T)$ sur des données de faible intensité pour pouvoir en déduire $\sigma^2_{\varepsilon_S}$, $\sigma^2_{\varepsilon_T}$ et $\sigma^2_{\varepsilon_C}$grâce au système d'équations :

$$\begin{aligned} \sigma^2_{\varepsilon_S} &\approx \frac{1}{2}\{Var(y_C) + Var(y_T) - Var(y_C - y_T)\} \\ \sigma^2_{\varepsilon_C} &\approx \frac{1}{2}\{Var(y_C) - Var(y_T) + Var(y_C - y_T)\} \\ \sigma^2_{\varepsilon_T} &\approx \frac{1}{2}\{Var(y_T) - Var(y_C) + Var(y_C - y_T)\} \end{aligned}$$

Le problème c'est que ce système d'équation peut conduire à des valeurs négatives pour un σ^2. Si cela se produit, on estimera que le σ^2 correspondant est nul.
Quand on n'a pas de répétitions d'expérience, l'algorithme 1.1 est proposé dans [38] pour estimer quel est l'ensemble des gènes faiblement exprimés. On pourra alors estimer les paramètres de moyenne et de variance des ε à partir de ce groupe de gènes peu exprimés.

A la fin de l'algorithme, on devrait avoir au moins 95% des gènes non exprimés dans l'ensemble considéré. Selon la distribution des vrais niveaux d'expression, cette estimation pourrait surestimer la moyenne et aussi légèrement la variance, parce qu'en pratique, il est impossible de faire la distinction entre un gène non exprimé et un gène avec un niveau d'expression tellement faible qu'il est en dessous des limites de détection. On peut aussi utiliser une variante de cet algorithme destinée à en réduire le biais. Il suffit de faire tourner l'algorithme avec la médiane et le $MAD/0.6745$ où MAD désigne l'écart médian absolu à la médiane. C'est d'ailleurs cette variante que nous avons choisi d'utiliser pour estimer nos paramètres.

L'algorithme proposé avait été conçu pour les expériences à un seul marquage. Dans le cas où on a des expériences à double marquage, il faut adapter un petit peu l'algorithme. Deux approches sont *a priori* possibles. On peut d'abord envisager de faire tourner l'algorithme séparément pour chaque fluorescence, mais le systeme d'équations 1.5 est valable lorsque l'on considère les gènes peu exprimés simultanément dans les deux fluorescences. C'est pourquoi il semble plus raisonnable d'adapter l'algorithme de façon à chercher un groupe de gènes qui soient peu exprimés dans les deux fluorescences. Cela donne l'algorithme 1.2.

Algorithme 1.1. **Trouver les gènes faiblement exprimés pour une fluorescence donnée.**

— pour pouvoir estimer les paramètres du modèle

1 - On initialise l'algorithme avec un petit ensemble B de gènes d'intensité faible, on prend par exemple les 10% de gènes avec la plus faible mesure d'intensité.
On calcule la moyenne $\overline{x_B}$ et la variance σ_B^2 sur cet ensemble.

2 - On définit alors un nouvel ensemble constitué des gènes dont l'intensité est comprise dans l'intervalle : $[\overline{x_B} - 2\sigma_B, \overline{x_B} + 2\sigma_B]$.
On recalcule $\overline{x_B}$ et σ_B^2 sur ce nouvel ensemble.

3- On répète l'étape précédente jusqu'à ce que l'ensemble des gènes considéré reste inchangé.

Algorithme 1.2. **Trouver les gènes faiblement exprimés pour expériences à deux fluorescences**

— pour pouvoir estimer les paramètres du modèle en double marquage

1 - On initialise l'algorithme avec un petit ensemble de gènes d'intensité faible, on prend par exemple les 10% de gènes avec la plus faible mesure d'intensité.
On calcule la moyenne $\overline{x}_{CB}$ et la variance σ_{CB}^2 de l'échantillon contrôle sur cet ensemble.
On calcule la moyenne $\overline{x}_{TB}$ et la variance σ_{TB}^2 de l'échantillon test sur cet ensemble.

2 - On définit alors un nouvel ensemble B constitué des gènes dont l'intensité pour l'échantillon contrôle est comprise dans l'intervalle $[\overline{x}_{CB} - 2\sigma_{CB}, \overline{x}_{CB} + 2\sigma_{CB}]$ et dont l'intensité pour l'échantillon test est comprise dans l'intervalle $[\overline{x}_{TB} - 2\sigma_{TB}, \overline{x}_{TB} + 2\sigma_{TB}]$
On recalcule les moyennes et variances sur ce nouvel ensemble.

3- On répète l'étape précédente jusqu'à ce que l'ensemble des gènes considéré reste inchangé.

On a en sortie de l'algorithme un groupe estimé de gènes peu exprimés dans les deux fluorescences qui permettent d'obtenir des estimations de $\overline{y_C}$, $\overline{y_T}$, $Var(y_C)$, $Var(y_T)$ et $Var(y_C - y_T)$ pour estimer ensuite les paramètres de notre modèle concernant le bruit de fond et l'erreur additive.

Estimation des $\sigma^2_{\eta_i}$

Ce qu'on veut estimer ici, ce sont les paramètres propres à la composante multiplicative de la variance. Cette erreur multiplicative est particulièrement importante quand les gènes sont fortement exprimés. Quand le niveau d'expression est important dans les deux échantillons, le modèle devient approximativement, pour le gène i :

$$\begin{aligned} ln(y_C)_i &\approx ln(\mu_{C,i} + \alpha_C) + \eta_{S,i} + \eta_{C,i} \\ ln(y_T)_i &\approx ln(\mu_{T,i} + \alpha_T) + \eta_{S,i} + \eta_{T,i} \end{aligned}$$

D'où :

$$\begin{aligned} Var(ln(y_C)) &\approx \sigma^2_{\eta_S} + \sigma^2_{\eta_C} \\ Var(ln(y_T)) &\approx \sigma^2_{\eta_S} + \sigma^2_{\eta_T} \\ Var(ln(y_C) - ln(y_T)) &\approx \sigma^2_{\eta_C} + \sigma^2_{\eta_T} \end{aligned} \tag{1.6}$$

Grâce à ce système, on pourra déduire σ_{η_T}, σ_{η_C} et σ_{η_S} de l'estimation de $Var(ln(y_C))$, $Var(ln(y_T)$ et $Var(ln(y_C) - ln(y_T))$:

$$\begin{aligned} \sigma^2_{\eta_S} &\approx \frac{1}{2}\{Var(ln(y_C)) + Var(ln(y_T)) - Var(ln(y_C) - ln(y_T))\} \\ \sigma^2_{\eta_C} &\approx \frac{1}{2}\{Var(ln(y_C)) - Var(ln(y_T)) + Var(ln(y_C) - ln(y_T))\} \\ \sigma^2_{\eta_T} &\approx \frac{1}{2}\{Var(ln(y_T)) - Var(ln(y_C)) + Var(ln(y_C) - ln(y_T))\} \end{aligned}$$

Il nous reste donc à estimer $Var(ln(y_C))$, $Var(ln(y_T))$ et $Var(ln(y_C) - ln(y_T))$ pour avoir une estimation de tous les paramètres du modèle. On utilise une procédure similaire à celle décrite dans le paragraphe précédent mais pour les log-intensités.

1.2.3 Démarche de simulation et détermination des niveaux d'expression

On a vu dans les deux sections précédentes comment générer une courbe de forme d'un nuage de points et comment modéliser l'aléa autour de cette courbe de forme. On va maintenant pouvoir procéder à la simulation d'un nuage de points à proprement parler. Cependant, dans le jeu de données simulé, il faut aussi intégrer des gènes différentiellement exprimés. Pour cela, on va calculer, à partir de la forme moyenne du nuage, qui correspond en fait à l'expression moyenne pour les gènes non différentiellement exprimés, deux autres courbes de forme qui correspondront

respectivement aux courbes de forme des nuages consitués respectivement des gènes sur-exprimés et des gènes sous-exprimés.

En fait, pour simuler un jeu de données issu de biopuces avec p points, on va simuler non pas un mais trois nuages de p points selon le modèle décrit dans le paragraphe précédent. Il y aura un nuage de gènes non différentiellement exprimés, un nuage de gènes induits et un nuage de gènes réprimés. Et ces trois nuages seront mélangés dans certaines proportions pour donner le jeu de données simulé final.

1.2.3.1 Les trois courbes de forme

Supposons qu'on a généré une fonction baignoire pour définir la forme du nuage que l'on va simuler et notons cette fonction f_{sim}. On dispose aussi des valeurs des paramètres estimés pour le modèle décrit par 1.4 (que nous garderons identiques pour les trois nuages à simuler). Notons μ_G et μ_R les expressions moyennes des gènes. Définir des formes de nuage différentes selon les niveaux d'expression des gènes (non expression différentielle, sur-expression ou sous-expression) revient à modéliser la relation entre les vecteurs μ_G et μ_R de façon différente en fonction du niveau d'expression. On notera $\mu_{G\,nexp}$ et $\mu_{R\,nexp}$ les niveaux d'expression moyens des gènes non différentiellement exprimés, $\mu_{G\,ind}$ et $\mu_{R\,ind}$ ceux des gènes sur-exprimés et $\mu_{G\,repr}$ et $\mu_{R\,repr}$ ceux des gènes sous-exprimés.

La première étape va être de simuler les vecteurs $\mu_{G\,nexp}$ et $\mu_{R\,nexp}$ qui sont d'ailleurs liés par la relation (*cf* équation 1.3) :

$$log_2(\mu_{R\,nexp}) - log_2(\mu_{G\,nexp}) \quad = \quad f_{sim}\left(\frac{log_2(\mu_{G\,nexp}) + log_2(\mu_{R\,nexp})}{2}\right) \quad (1.7)$$

On suppose comme il a été dit dans la section 1.2.1.3 que si l'on dispose de $\mu_{G\,nexp}$, on est alors en mesure de calculer $\mu_{R\,nexp}$. Il suffit donc de générer un vecteur $\mu_{G\,nexp}$. Pour cela, on utilise les jeux de données réels dont on dispose. On trace l'histogramme de la distribution en vert de ces données réelles, dans cet histogramme on compte le nombre de points p_{c_k} par classe c_k, et on génère à partir de cela un nouveau vecteur $\mu_{G\,nexp}$ en espaçant de façon régulière p_{c_k} points dans chaque classe c_k.

Ainsi on dipsose d'un premier couple de vecteurs $\mu_{R\,nexp}$ et $\mu_{G\,nexp}$ qui vont correspondre aux niveaux moyens d'expression pour les gènes non différentiellement exprimés. On génère alors un premier nuage de points selon le modèle décrit par les équations 1.4.

Il faut maintenant générer les vecteurs des niveaux d'expression pour les gènes induits et pour les gènes réprimés. On va calculer ces vecteurs à partir de $\mu_{R\,nexp}$ et $\mu_{G\,nexp}$. On définit tout d'abord A_{nexp} le vecteur d'intensité totale moyenne et M_{nexp} le vecteur des $\log_2$-ratios d'expression du nuage des gènes non différentiellement exprimé par

$$A_{nexp} = \frac{1}{2}\left(log_2(\mu_{G\,nexp}) + log_2(\mu_{R\,nexp})\right)$$

$$M_{nexp} = log_2(\mu_{R\,nexp}) - log_2(\mu_{G\,nexp})$$

On estime la variance de M_{nexp} comme fonction de l'intensité totale moyenne A_{nexp}. Pour cela, on centre les $(M_{nexp})_i$ en utilisant une correction lowess (de la même façon qu'une normalisation lowess) et on utilise un lissage de ces résidus au carré pour estimer la variance $\sigma^2(A_{nexp})$ du nuage en fonction de A_{nexp}.

On pose alors

$$\begin{aligned}
\mu_{G\,ind} &= \mu_{G\,nexp} \times 2^{-c_{ind} \times \sigma(A_{nexp})} \\
\mu_{R\,ind} &= \mu_{R\,nexp} \times 2^{c_{ind} \times \sigma(A_{nexp})} \\
\mu_{G\,repr} &= \mu_{G\,nexp} \times 2^{c_{repr} \times \sigma(A_{nexp})} \\
\mu_{R\,repr} &= \mu_{R\,nexp} \times 2^{-c_{repr} \times \sigma(A_{nexp})}
\end{aligned}$$

où c_{ind} et c_{repr} sont des paramètres positifs choisis arbitrairement. Dans nos applications, on prendra $c_{ind} = c_{repr}$ mais on peut envisager de raffiner le modèle et de prendre des valeurs différentes. La valeur choisie doit réaliser un bon compromis pour ne pas avoir des nuages selon le niveau d'expression complètement confondus mais sans non plus les rendre trop distincts.

Pourquoi cette façon de calculer les expressions moyennes pour les gènes différentiellement exprimés ? Regardons les relations que cela donne entre $mu_{G\,ind}$ et $mu_{G\,ind}$ d'une part et entre $mu_{G\,repr}$ et $mu_{G\,repr}$ d'autre part. En utilisant l'équation 1.7, on obtient :

$$\begin{aligned}
log_2(\mu_{R\,ind}) - log_2(\mu_{G\,ind}) &= f_{sim}\left(\frac{log_2(\mu_{G\,ind}) + log_2(\mu_{R\,ind})}{2}\right) + 2c_{ind}\sigma(A_{nexp}) \\
log_2(\mu_{R\,repr}) - log_2(\mu_{G\,repr}) &= f_{sim}\left(\frac{log_2(\mu_{G\,repr}) + log_2(\mu_{R\,repr})}{2}\right) - 2c_{repr}\sigma(A_{nexp})
\end{aligned}$$

Cela revient à utiliser une courbe de forme de nuage pour les gènes induits (resp. réprimés) qui se situe "au dessus" (resp. "en dessous") de la courbe de forme du nuage des gènes non exprimés, avec une différence entre les deux courbes d'autant plus importante que le nuage des gènes non différentiellement exprimés est dispersé. On peut voir un exemple de résultat à ce stade sur la figure 1.20.

Une fois que l'on a les vecteurs $\mu_{G\,ind}$, $\mu_{R\,ind}$, $\mu_{G\,repr}$ et $\mu_{R\,repr}$, on est en mesure de simuler les deux nuages de points des gènes différentiellement exprimés. On dispose donc maintenant de trois nuages de p points, un pour chaque catégorie d'expression des gènes. Il faut maintenant mélanger ces nuages pour n'en avoir plus qu'un.

1.2.3.2 Mélange des trois nuages

Pour construire le nuage final, nous allons donc procéder au mélange des trois nuages générés précédemment. On introduit ici un nouveau paramètre π_{cont} de « contamination ». Ce paramètre correspond à la probabilité pour un gène d'être différentiellement exprimé (induit ou réprimé). Prendre $\pi_{cont} = 5\%$ semble raisonnable. On génère un vecteur de taille p selon une loi de Bernoulli de paramètre π_{cont}. On construit ainsi un vecteur de taille p qui pour chaque expérience vaut 0 en cas

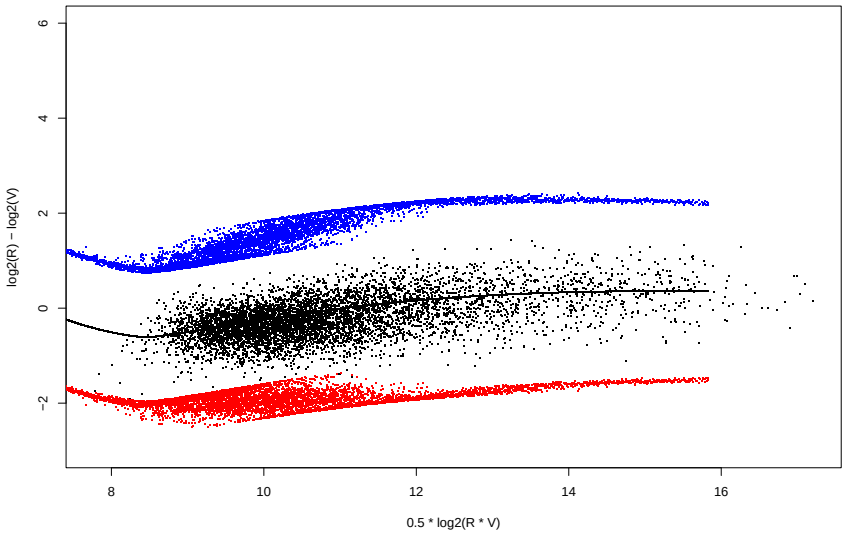

Figure 1.20 – Un étape de la simulation. Points noirs : on a un nuage de p gènes non différentiellement exprimés ; courbe noire : la courbe des expressions moyennes pour les gènes non différentiellement exprimés. En bleu : les expressions moyennes pour les gènes induits. En rouge : les expressions moyennes pour les gènes réprimés.

d'échec et 1 en cas de réussite. Pour notre application, cela revient à dire que nous obtiendrons un 0 pour les gènes non différentiellement exprimés et un 1 pour les autres. L'avantage de cette méthode, c'est que cela évite d'avoir un nombre fixe de gènes différentiellement exprimés. Ce nombre devient avec cette modélisation une variable aléatoire P_{nexp} qui suit une loi $\mathcal{B}(p\ ;\ \pi_{cont})$ et qui a donc une espérance égale à $p \times \pi cont$.

On a donc créé un vecteur qui nous donne un index : gène non différentiellement exprimé, gène différentiellement exprimé. Pour décider si un gène différentiellement exprimé est induit ou réprimé, on procède de façon identique mais en générant cette fois un vecteur index de taille $P_{exp} = p - P_{nexp}$ selon une loi de Bernoulli de paramètre $0,5$, c'est-à-dire qu'on estime qu'on a une chance sur deux pour qu'un gène différentiellement exprimé soit induit ou réprimé. On a donc enfin obtenu un nuage de données simulées, accompagné d'un vecteur qui nous permet de garder l'identité des différents gènes : « non différentiellement exprimé », « induit » ou « réprimé ».

1.2.4 Résultats de simulation

En adoptant le modèle présenté dans les sections précédentes, on est donc désormais en mesure de simuler des données de biopuces, dans lesquelles on sait quels sont les gènes induits ou réprimés. On va simuler des jeux de données selon deux modèles différents. Un modèle construit à partir des jeux de données sans correction du bruit de fond et un jeu de données construit à partir des jeux de données avec correction du bruit de fond.

1.2.4.1 Données non corrigées par le bruit de fond

La dernière étape pour rendre un jeu de données simulé plus réaliste c'est de générer aussi des saturants, qui sont toujours présents dans les données brutes, pour cela c'est très simple : on seuille toutes les fluorescences du jeu simulé supérieures à 65 535 au seuil de saturation, c'est-à-dire 65 535. On montre ici (figure 1.21) quelques exemples de nuages simulés obtenus, les formes sont assez diverses et les nuages ressemblent beaucoup aux vrais jeux de données, notons qu'ici, nous avons pris $c_{ind} = c_{repr} = 10$ (coefficients intervenant pour le calcul des moyennes des différentiellement exprimés). On notera toutefois certaines différences entre les nuages réels et les nuages simulés, notamment au niveau de la dispersion globale des données. On observe une variabilité plus homogène entre les différents nuages simulés. Cela tient au fait que tous les nuages sont simulés avec les mêmes valeurs des paramètres α_C, α_T, $\sigma^2_{\varepsilon_E}$, $\sigma^2_{\eta_E}$, $E \in \{C, T, S\}$, qui sont calculées à partir des jeux de données réels. Pour cela quand on estime le bruit sur chaque fluorescence pour l'ensemble des jeux de données réels, on prend comme bruit estimé d'ensemble la médiane des bruits estimés sur les différents jeux de données. Pour approfondir ces travaux, on pourrait envisager d'estimer la loi des ces estimateurs sur l'ensemble des jeux de données réels

pour générer ensuite des valeurs aléatoires des paramètres du modèle selon cette loi estimée pour chaque nouvelle simulation d'un jeu de données.

1.2.4.2 Données corrigées par le bruit de fond

Nous allons maintenant présenter les résultats obtenus en adaptant le modèle de simulation aux données obtenues avec correction du bruit de fond selon la méthode de Kooperberg.

Dans un premier temps, nous allons considérer les différentes étapes de la construction du modèle, pour vérifier que le modèle utilisé est bien adapté à ces données. Observons d'abord sur la figure 1.22, la forme des nuages obtenus après correction du bruit de fond. Nous allons maintenant calculer la fonction "baignoire" correspondant à ces jeux de données et vérifier sur la figure 1.23, que cette fonction convient bien pour approcher aussi les nuages corrigés par le bruit de fond. Si la fonction baignoire semble un tout petit peu moins adaptée au cas des nuages corrigés par le bruit de fond que sur les données brutes, elle donne toutefois des résultats tout à fait satisfaisants, et peut donc être aussi utilisée dans ce cas-là.

Après avoir estimé les paramètres du modèle décrit par 1.4, on peut simuler des jeux de données. Des exemples de données simulées sont fournis sur la figure 1.24, ici $c_{ind} = c_{repr} = 6$, rapport qui semblait plus adapté pour ce modèle.

Ici il n'y a pas de raison vraiment valable pour effectuer, comme pour le modèle sans correction du bruit de fond, un seuillage des données à 65 535. En effet, cette valeur correspond à une limite du scanner qui se retrouve sur les données brutes mais, après correction du bruit de fond, on a pu augmenter ou réduire les valeurs. Même si sur certains jeux de données, on observe des traces de saturants (lignes diagonales sur le côté droit du nuage de points - figure 1.22 nuage en haut à droite), ceci est beaucoup moins vrai qu'avec les données brutes. On choisit donc de ne pas seuiller les plus grandes valeurs.

Ici, on a l'impression que les jeux de données simulés ressemblent moins aux jeux de données réels. On peut déjà faire la même remarque que pour les données sans correction du bruit de fond : entre les différentes données simulées, on observe moins de variations de la dispersion du nuage qu'entre les données réelles. Un autre problème, qui était était déjà présent, dans une moindre mesure, pour l'autre modèle de simulation est qu'on n'observe presqu'aucun gène ayant une $\log_2$-intensité totale inférieure à 5 (resp. 8 pour le modèle sans correction du bruit de fond) sur les simulations, alors que c'était presque systématiquement le cas pour les données réelles.

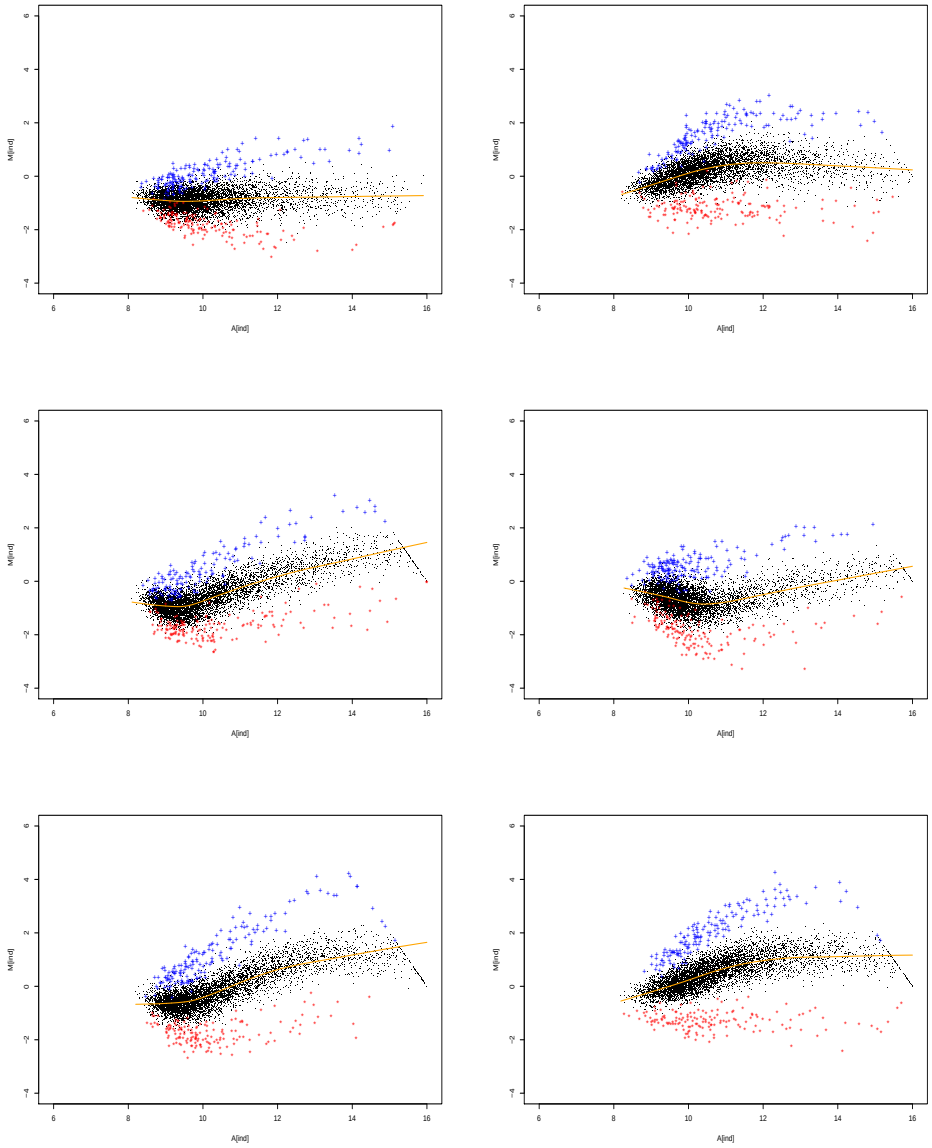

Figure 1.21 – Quelques exemples de jeux de données simulés, sans correction du bruit de fond. Les points noirs représentent les gènes non différentiellement exprimés, les croix bleues les gènes induits et les étoiles rouges les gènes réprimés.

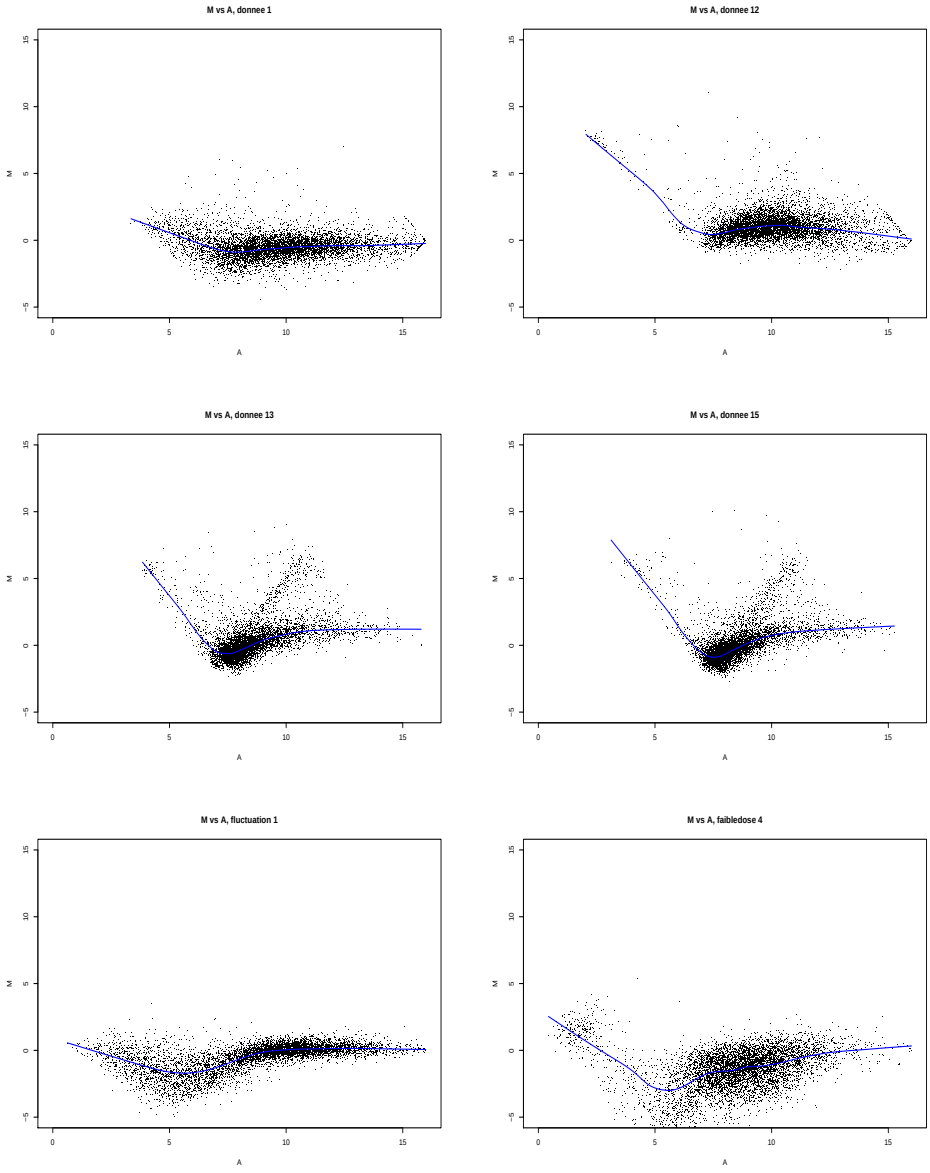

Figure 1.22 – Formes de nuages avec correction du bruit de fond

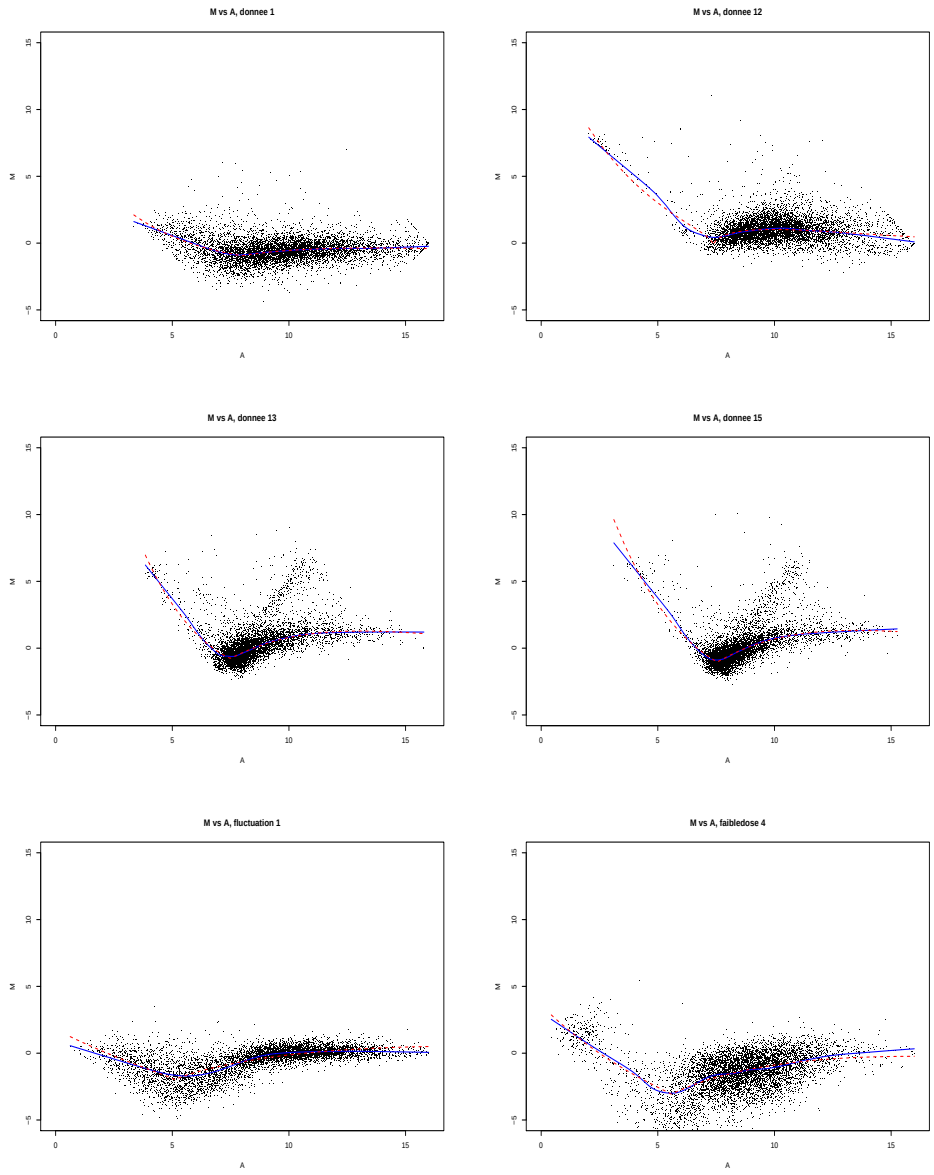

Figure 1.23 – Approche paramétrique de la courbe lowess, données corrigées par le bruit de fond. Pour différents nuages de points, on calcule la fonction lowess qui donne la forme du nuage (trait plein). On approche ensuite ce lowess par une fonction paramétrique de type « baignoire »(trait pointillé). On voit que les courbes paramétriques s'adaptent relativement bien aux différentes formes de nuage de points.

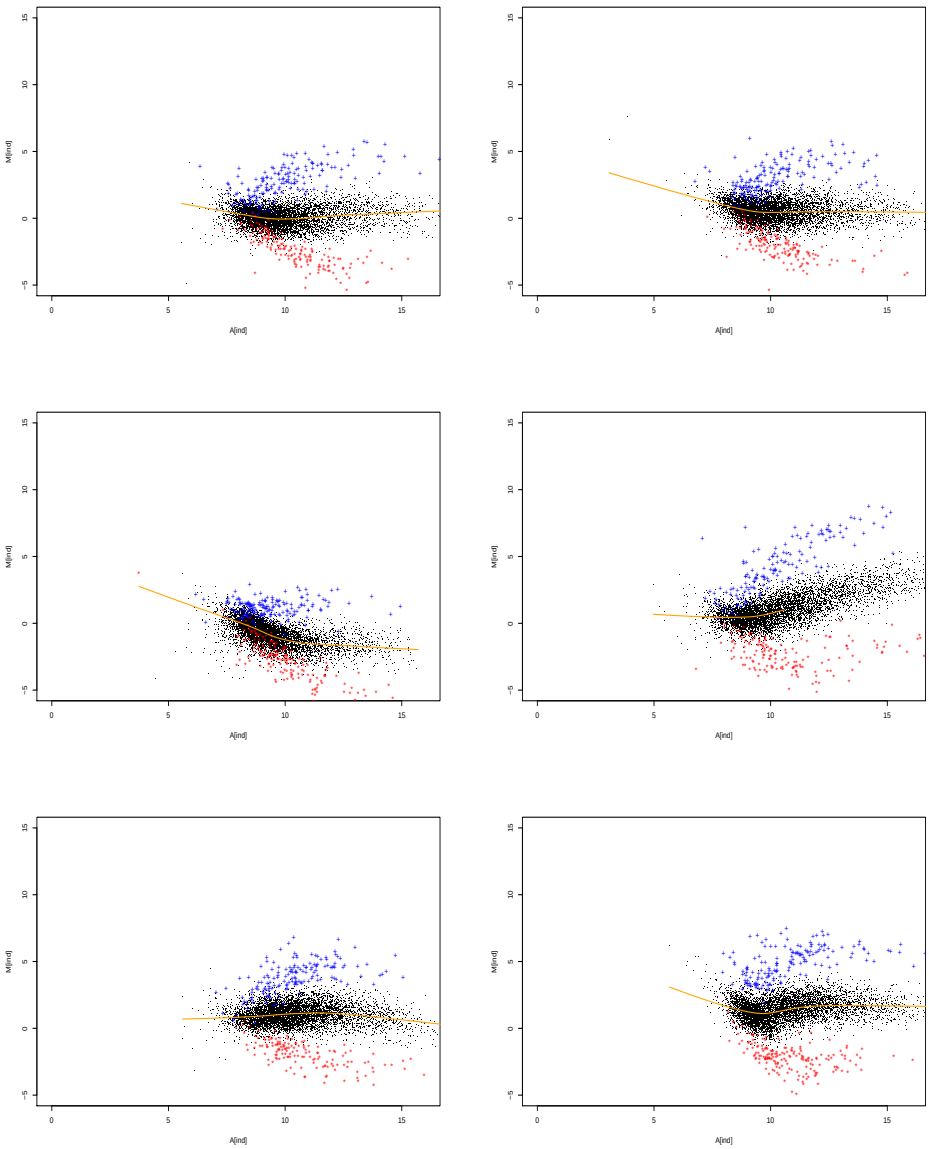

Figure 1.24 – Quelques exemples de jeux de données simulés dans le modèle avec correction bayésienne du bruit de fond. Les points noirs représentent les gènes non différentiellement exprimés, les croix bleues les gènes induits et les étoiles rouges les gènes réprimés.

Conclusion

On a donc mis au point une méthode de simulation de données issues de biopuces, fondées sur un modèle pour lequel on a estimé les paramètres grâce aux vrais jeux de données. Certes cette méthode n'est pas parfaite et comporte encore des imprécisions, mais c'est une première approche qui semble assez réaliste de la simulation de données en matière de biopuces. On va maintenant pouvoir essayer les différentes techniques de normalisation sur ces données simulées, sur des jeux de données simulés à partir du modèle sans correction du bruit de fond et sur des jeux de données simulés à partir du modèle avec correction bayésienne du bruit de fond.

Choix d'une méthode de normalisation

Pour être en mesure de choisir une méthode de normalisation des données issues des biopuces à ADN, on va faire des essais sur des données issues de simulation. Cela nous permettra d'estimer la qualité de la normalisation. Sur la figure 1.25, on peut comparer visuellement les différentes méthodes de normalisation sur des données simulées. On voit que la normalisation par la médiane ne change rien à la forme du nuage et ne permettra pas une bonne détection des gènes différentiellement exprimés. Entre la normalisation lowess (avec lissage des résidus) et les « normal scores » (avec lissage des résidus), il est plus difficile de définir visuellement quelle normalisation est préférable. La normalisation par les scores semblent répartir les gènes différentiellement exprimés plus homogènement selon l'intensité totale moyenne.

Pour être capable de comparer les différentes méthodes de normalisation, on va essayer de les qualifier numériquement. Pour cela, on commence par trier les valeurs absolues des $\log_2$-ratios (ou équivalent pour les scores) obtenus après normalisation.Puisque les jeux de données sont simulés, on connaît p_{de} le nombre réel de gènes différentiellement exprimés et on sait quels sont ces p_{de} gènes différentiellement exprimés. On aura d'ailleurs

$$p_{de} = p_{de,rep} + p_{de,indu}$$

où $p_{de,rep}$ (resp. $p_{de,indu}$) désigne le nombre réel de gènes réprimés (resp. induits). On notera $p^i_{de} (= p^i_{de,rep} + p^i_{de,indu})$ le nombre de gènes différentiellement exprimés pour le jeu de données simulé numéro i. Ce nombre est en moyenne de 324 dans la mesure où, en moyenne, 5% des gènes sont différentiellement exprimés et où on prend $p = 6472$. On va alors calculer un taux de détection des gènes différentiellement exprimés pour chacune des méthodes de normalisation envisagées. Pour le jeu de données simulé numéro i, on sélectionne les p^i_{de} gènes qui ont le plus fort ratio d'expression en valeur absolue après normalisation.

On dispose ainsi pour chaque jeu de données de la liste des p^i_{de} gènes qui sont effectivement différentiellement exprimés et de la liste des p^i_{de} gènes qui seraient détectés pour la méthode de normalisation k ($k \in$\{aucune, mediane, lowess, normal scores \}), nous noterons respectivement ces listes A^i et B^i_k.

Ainsi on définit τ^i_k, le taux de détection des gènes différentiellement exprimés

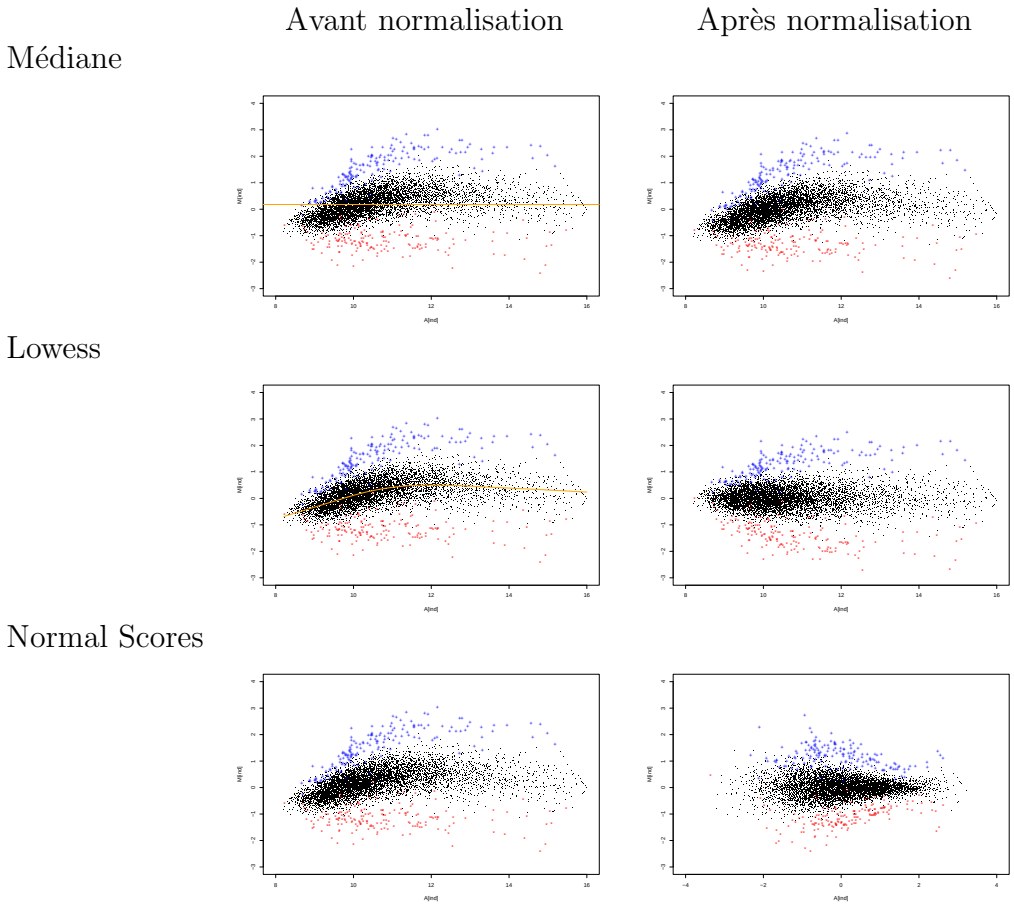

Figure 1.25 – Les différents types de normalisation sur un jeu de données simulées, modèle sans correction du bruit de fond. A gauche, le nuage avant normalisation avec le cas échéant le tracé de la courbe de normalisation ; à droite, le résultat après normalisation, avant lissage des résidus. Les points noirs représentent les gènes non différentiellement exprimés, les croix bleues les gènes induits et les étoiles rouges les gènes réprimés.

pour le jeu de données numéro i avec la méthode de normalisation k par :

$$\tau_k^i = \frac{Card(A^i \cap B_k^i)}{Card(A^i)} = \frac{Card(A^i \cap B_k^i)}{p_{de}^i}$$

De façon analogue, on définit $\tau_{k,rep}^i$ (resp. $\tau_{k,indu}^i$) qui correspond aux taux de détection uniquement pour les gènes réprimés (resp. induits). C'est-à-dire qu'on ne prend pas cette fois les valeurs absolues des $\log_2$-ratio.

Si on a simulé m jeu de données, on définit τ_k, le taux moyen de détection pour la méthode de normalisation k par :

$$\tau_k = \sum_{i=1}^{m} \tau_k^i$$

On note alors $\tau_{k,rep}$ (resp. $\tau_{k,indu}$) le taux moyen de détection des gènes induits (resp. réprimés). La normalisation sera d'autant meilleure que le taux de détection sera grand.

En pratique, on fait le calcul de ces taux sur un ensemble de $m = 200$ simulations, en fait on simule 20 courbes de formes de nuages différentes, et pour chacune de ces courbes on simule 10 jeux de données. Les résultats obtenus pour les taux moyen de détection sont donnés table 1.2 pour le modèle sans correction du bruit de fond et table 1.3 pour le modèle avec correction du bruit de fond.

Méthode de normalisation k	τ_k	$\tau_{k,rep}$	$\tau_{k,indu}$
Aucune normalisation	0.421	0.611	0.536
Normalisation par la médiane			
- avec lissage	0.632	0.637	0.635
- sans lissage	0.525	0.611	0.536
Normalisation lowess			
- avec lissage	0.716	0.718	0.714
- sans lissage	0.688	0.691	0.685
Normalisation par les scores			
- avec lissage	0.702	0.695	0.707
- sans lissage	0.572	0.556	0.587

Table 1.2 – Taux de détection des gènes différentiellement exprimés selon le type de normalisation utilisé : médiane, lowess ou scores, dans les 3 cas avec ou sans standardisation. Modèle sans correction du bruit de fond.

Méthode de normalisation k	τ_k	$\tau_{k,rep}$	$\tau_{k,indu}$
Aucune normalisation	0.461	0.654	0.533
Normalisation par la médiane			
- avec lissage	0.659	0.666	0.660
- sans lissage	0.584	0.654	0.533
Normalisation lowess			
- avec lissage	0.707	0.705	0.709
- sans lissage	0.696	0.697	0.698
Normalisation par les scores			
- avec lissage	0.689	0.680	0.700
- sans lissage	0.600	0.586	0.614

Table 1.3 – Taux de détection des gènes différentiellement exprimés selon le type de normalisation utilisé : médiane, lowess ou scores, dans les 3 cas avec ou sans standardisation. Modèle avec correction du bruit de fond.

► **Choix d'une méthode de normalisation**.

Comme on s'y attendait, la normalisation par la médiane est la moins adéquate. Quand on considère le taux de détection τ_k obtenu à partir des valeurs absolues elle apporte toutefois une plus-value, grâce au centrage des données qu'elle implique.

Entre normalisation par les scores et normalisation par lowess, on peut hésiter car les résultats obtenus sont tout à fait comparables, d'autant plus qu'ici on a fait un choix arbitraire des paramères c_{ind} et c_{repr} qui fixent l'écart entre la moyenne du nuage des gènes induits (resp. réprimés) et la moyenne du nuage des gènes non exprimés. Par exemple ici, pour le modèle sans correction du bruit de fond, on a considéré $c_{ind} = c_{repr} = 10$. En prenant par exemple 12 au lieu de 10, on observe $\tau_k = 81,3\%$ pour la méthode des scores contre $\tau_k = 81,0\%$ pour la méthode lowess, dans les deux cas avec lissage des résidus.

Ce lissage des résidus améliore beaucoup les résultats de détection sur les simulations. Si la méthode lowess y est finalement assez peu sensible, les méthodes "scores" et "médiane" y gagnent beaucoup en puissance de détection.

La normalisation par les scores semble plus universelle et peut-être plus robuste. Elle présente d'autres avantages : elle délivre des résultats qui ont toujours la même distribution, ce qui peut permettre de comparer directement les résultats de deux expériences de biopuces sans avoir à passer éventuellement par une normalisation des données entre les lames. C'est une méthode simple mais qui peut être ressentie par les biologistes comme une "boîte noire" qui modifie les données sans qu'on comprenne très bien son fonctionnement.

En effet, la normalisation par lowess est beaucoup plus "visuelle", on voit bien la fonction de normalisation calculée et puis on comprend le sens biologique de cette normalisation qui tient compte de l'intensité totale d'un spot pour corriger le ratio

d'expression. De plus, la méthode lowess est actuellement celle qui est la plus utilisée dans la littérature pour analyser les données de biopuces. Il semble donc plus justifié de choisir cette normalisation avant de passer aux étapes suivantes de l'analyse des données et c'est ce que nous avons fait.

▶ **Choix d'une méthode de traitement du bruit de fond**.

Cette question est plus difficile à trancher, en effet, on choisit des paramètres (c_{ind}, c_{repr}) différents entre les deux modèles, de fait il est donc difficile de comparer ici les résultats de normalisation. Dans les simulations proposées ici - c'est-à-dire $c_{ind} = c_{repr} = 6$ pour le modèle avec correction du bruit de fond et $c_{ind} = c_{repr} = 10$ pour l'autre modèle - on a une meilleure détection des différentiellement exprimés sans normalisation pour le modèle avec correction du bruit de fond. Cependant, les résultats après normalisation par lowess ou par les scores, avec lissage des résidus, sont légèrement meilleurs pour l'autre modèle. Cela peut certes vouloir dire que notre modélisation s'adapte moins bien aux données corrigées par le bruit de fond, mais cela peut aussi indiquer que la correction du bruit de fond va entraver la détection des gènes différentiellement exprimés après normalisation.

La correction du bruit de fond n'apporte donc pas d'amélioration vraiment visible dans le traitement des données, et semble ajouter du bruit dans les données. On avait déjà fait cette remarque à propos de la figure 1.4, mais cela se voit aussi sur les figures 1.18 et 1.22, où sont représentés les mêmes jeux de données, sans correction du bruit de fond dans la première et avec correction du bruit de fond dans la seconde. De plus sur la figure, 1.22, sur les deux nuages situés sur la ligne du milieu, il semble qu'on puisse distinguer des structures, ce qui est assez étonnant. Dans la mesure où on manque ici d'arguments valables pour choisir d'effectuer une correction bayésienne du bruit de fond, on va préférer utiliser des données brutes, sans correction du bruit de fond.

Dans toute la suite, on considèrera donc des données où aucune correction du bruit de fond n'a été faite, et où la normalisation a été faite par méthode lowess, avec lissage des résidus.

Conclusion du premier chapitre

Dans ce chapitre, nous avons donc étudié les méthodes de normalisation, proposé une méthode de simulation et comparé les différentes normalisations sur des données simulées. Les méthodes de simulation de données sont rares dans le domaine des biopuces. Le modèle qui a été utilisé peut sans aucun doute être amélioré, notamment en faisant varier davantage de paramètres d'une simulation à une autre, mais il apporte déjà quelques réponses sur le traitement des données. Ce qui ressort de l'ensemble du chapitre, c'est que la normalisation lowess, par bloc, avec remise à l'échelle entre les blocs et avec lissage des résidus, est une approche de normalisation sensée dans le cadre des biopuces à ADN. En ce qui concerne le traitement du bruit de fond, le problème n'a pas été vraiment tranché. Cependant vu la difficulté à se mettre d'accord sur une approche en particulier (certains veulent le retrancher, d'autres l'ajouter), et même si le traitement bayésien proposé par Kooperberg *et al.* semble être la méthode la mieux adaptée, on choisira de conserver des données non corrigées par le bruit de fond, en partie parce que nous estimons que ce traitement introduit un bruit non souhaitable dans les données.

Une fois l'étape de prétraitement effectuée, nous allons passer à une autre étape du processus d'analyse des données issues des biopuces à ADN, à savoir, la détection des gènes différentiellement exprimés.

Chapitre 2

Procédures de détection des gènes exprimés dans les expériences de biopuces

Introduction et motivation

L'objectif de cette partie est d'identifier les gènes différentiellement exprimés, c'est-à-dire les gènes dont l'expression est associée à une réponse ou une covariable d'intérêt.

Cette question peut être abordée comme un problème de test d'hypothèses multiples : il s'agit d'effectuer le test simultané que, pour chaque gène, sous l'hypothèse nulle, il n'y a pas de relation entre le niveau d'expression et la réponse ou les covariables.

Comme pour tout test, il y a deux étapes essentielles : le choix de la statistique de test puis le calcul d'un seuil à partir duquel on rejettera l'hypothèse nulle. Si choisir une statistique de test adaptée est essentiel, le choix d'un seuil est sans doute l'étape la plus délicate. Fixer ce seuil c'est choisir une règle de décision en fonction d'un critère donné.

Dans la première section de ce chapitre, nous allons nous intéresser aux différents taux d'erreur que l'on peut vouloir contrôler dans un test d'hypothèses multiples, et étudier différentes statistiques de test que l'on peut envisager d'utiliser. Dans les sections suivantes, nous considèrerons plusieurs règles de décision pour le test, basées sur des critères assez variés : sélection de modèles, pénalisée ou non, mais aussi une règle de seuillage bayésien. On verra également qu'on peut effectuer une décomposition en ondelettes pour décorréler les données avant d'appliquer ces méthodes de détection des gènes différentiellement exprimés. Nous comparerons ces méthodes sur des données simulées, et sur un jeu de données où les gènes différentiellement exprimés sont censés être connus. Nous mettrons en évidence les avantages et les inconvénients de chaque méthode, pour pouvoir conclure quant à l'approche à choisir. Nous confronterons alors ces méthodes sur un autre jeu de données réel où on n'a pas de connaissance *a priori* sur les gènes différentiellement exprimés.

Les tests d'hypothèses multiples et les différents types d'erreurs

2.1.1 Notations

On suppose que l'on dispose de n biopuces. On a n_1 biopuces réalisées à partir de cellules ayant été soumises à une condition expérimentale A donnée (par exemple : absence de traitement, irradiation, exposition à un produit toxique, cellules cancéreuses...) et n_2 $(= n - n_1)$ biopuces réalisées à partir de cellules ayant été soumises à une condition expérimentale B donnée.

On souhaite identifier les gènes qui exhibent une différence statistique significative entre ces deux conditions expérimentales. Par exemple, on voudra repérer les gènes caractéristiques d'un cancer par rapport à des cellules saines, ou bien les gènes caractéristiques dans leur expression d'une irradiation des cellules par rapport à des cellules non irradiées.

Pour chacune des n biopuces, pour chacun des p gènes, on suppose que l'on dispose du $\log_2$ratio normalisé des intensités en rouge et vert que l'on note $X_{ij}^{(k)}$ où $i \in \{1 \dots p\}$ désigne le $i^{ème}$ gène, $k \in \{1\ ,\ 2\}$ désigne le traitement subi, j désigne la $j^{ème}$ expérience ($j \in \{1 \dots n_k\}$).

Pour chaque gène i, on souhaite tester l'hypothèse H_{0i} : « le gène i n'est pas différentiellement exprimé entre les populations 1 et 2 » contre H_{1i} : « le gène i est différentiellement exprimé entre les populations 1 et 2 ». Ici, se pose donc un problème de **Test d'Hypothèses Multiples**. Pour chaque gène i, on effectue un test statistique pour obtenir une p-valeur. Mais que désigne une p-valeur ? La p-valeur associée à l'observation d'une statistique de test est le seuil à partir duquel on rejetterait l'hypothèse nulle compte tenu de cette observation. La p-valeur c'est aussi la probabilité d'observer une valeur au moins aussi extrême que celle observée pour la statistique de test Z si H_0 est « vraie ».

Pour le gène i, dans le cas bilatéral, $\pi_i = \mathbb{P}_{H_0}(Rejet\ de\ H_{0i}) = \mathbb{P}_{H_0}(|Z| \geq |Z_i|)$ avec Z la variable aléatoire correspondant à la statistique de test et Z_i la statistique de test observée pour le gène i.

Si la p-valeur est grande, c'est une indication que les données observées sont plausibles sous H_0 : H_0 n'est pas rejetée. Si la p-valeur est petite, c'est une indication que les données observées ne sont pas plausibles sous H_0 : H_0 est rejetée. Notons

que rejeter H_{0i}, c'est conclure que le gène i est différentiellement exprimé. Dans un test de décision, on fixe un seuil d'erreur α que l'on juge acceptable. Si l'hypothèse H_0 est rejetée, c'est alors avec une probabilité de se tromper inférieure à α. Ainsi, si on calcule la p-valeur, on rejettera H_0 si et seulement si la p-valeur est inférieure à α.

2.1.2 Test d'hypothèses multiples

2.1.2.1 Les taux d'erreur classiques

Notons R le nombre d'hypothèses rejetées et V le nombre d'hypothèses rejetées à tort. R est une variable observable alors que V est non observable. On résume classiquement (cf Benjamini et Hochberg [8]) la situation dans le tableau ci-dessous (2.1).

Conclusion du test / Réalité	Pas de rejet de H_0	Rejet de H_0	
H_0 vraie	U	V	p_0
H_0 fausse	T	S	p_1
	$p-R$	R	p

Table 2.1 – La situation dans un test d'hypothèses multiples.

On a donc p hypothèses, une pour chaque gène, avec p connu, et on note respectivement p_0 et $p_1 = p - p_0$ les nombres d'hypothèses nulles respectivement vraies et fausses. Ces nombres sont des paramètres inconnus. De même que V, S, T et U sont des variables aléatoires non observables. En général, on cherche à minimiser le nombre V de faux positifs ou erreurs de type I et le nombre T de faux négatifs ou erreurs de type II. L'approche standard dans le cas univarié consiste à se fixer un seuil de taux d'erreur de type I acceptable, α (par exemple, $\alpha = 5\%$) et de chercher des tests qui minimisent le taux d'erreur de type II c'est-à-dire aussi qui maximisent la puissance (puissance = 1 - taux d'erreur de type II), au sein de la classe de tests avec un taux d'erreur de type I de α.

Considérons les différents taux d'erreur de type I. Quand on teste une seule hypothèse, disons H_0, la probabilité d'erreur de type I, c'est-à-dire de rejeter l'hypothèse nulle alors qu'elle est vraie est généralement contrôlée à un seuil α. Si on note Z la statistique de test correspondante, cela peut être réalisé en choisissant une valeur critique c_α telle que $\mathbb{P}(|Z_1| \geq c_\alpha | H_0) \leq \alpha$ et en rejettant H_0 quand $|Z_1| \geq c_\alpha$. Plusieurs généralisations au cas des tests mutiples sont possibles, Hochber et Tamhane en proposent plusieurs dans leur livre [25] sur les "Procédures de Comparaison

Multiples". Décrivons maintenant les différents taux d'erreur que l'ont peut envisager dans le cadre de test d'hypothèses multiples, ces taux d'erreur sont notamment décrits par Shaffer dans [41] et Dudoit *et al.* dans [16].

Le PCER (Per Comparison Error Rate)

Le PCER est défini comme le rapport de l'espérance du nombre d'erreurs de type I sur le nombre total d'hypothèses, c'est-à-dire :

$$PCER = E(\frac{V}{p}) \leqslant \alpha$$

Pour contrôler le PCER au taux α, il suffit par exemple de faire pour chaque hypothèse (*ie* chaque gène) un test au seuil α. Cette façon de procéder ne tient pas compte de la multiplicité des données ; en effet, les taux d'erreur de l'ensemble peuvent être importants. Ainsi si on fait pour chaque gène un test au seuil α (classiquement $\alpha = 5\%$), on risque d'avoir une erreur d'ensemble trop importante. C'est-à-dire que si on a $p = 6000$ et $\alpha = 5\%$, on peut très bien avoir 300 faux-positifs. Un tel taux d'erreur n'est pas acceptable.

le FWER (Family-Wise Error Rate)

Pour rendre le test plus conservatif, une idée consiste à faire pour chaque gène un test au seuil $\frac{\alpha}{p}$. Dans cette optique, on contrôle alors un taux appelé FWER (Family Wise Error Rate) au seuil α :

$$FWER = P(V \geqslant 1) \leqslant \alpha$$

Cela veut dire en fait qu'on garantit qu'en moyenne la probabilité d'avoir au moins un faux positif est inférieure à α. Le problème, c'est qu'avec $p = 6000$ et $\alpha = 5\%$, on doit donc faire pour chaque gène un test au seuil $\frac{\alpha}{p} \approx 0.0008\%$! Avec un tel seuil, on ne détectera probablement la sur-expression (sous-expression) d'aucun gène ! Alors, on aura certes peu de chances de se tromper mais si on ne détecte aucun gène, cela n'a pas grand intérêt, la puissance du test sera très mauvaise.

Le PFER (Per-Family Error Rate)

C'est l'espérance du nombre d'erreurs de type I

$$PFER = E(V)$$

Le FDR (False Discovery Rate)

Le FDR a été proposé par Yoav Benjamini et Yosef Hochberg [8]. Au lieu de considérer le nombre d'hypothèses nulles rejetées à tort dans l'absolu ou par rapport

au nombre total d'hypothèses testées p, on considère le nombre d'hypothèses nulles rejetées à tort V par rapport à R, le nombre total d'hypothèses nulles rejetées. On s'intéresse ainsi à la variable $Q = \frac{V}{R}$.

$$Q = \begin{cases} \frac{V}{R} & si\ R \neq 0 \\ 0 & si\ R = 0 \end{cases}$$

Q représente en fait le taux de faux-positifs, c'est-à-dire dans notre problème la proportion de gènes détectés à tort. L'idéal serait de pouvoir contrôler cette variable Q mais c'est impossible. En effet, si toutes les hypothèses nulles sont vraies (p gènes non différentiellement exprimés), alors toutes les hypothèses rejetées le seront à tort et $q = \frac{v}{r} = 1$ ne peut pas être contrôlé. On définit alors le FDR par l'espérance de Q :

$$FDR = E(Q) = E(\frac{V}{R})$$

Il reste maintenant à déterminer quel taux d'erreur sera le plus intéressant dans le contexte qui nous intéresse.

2.1.2.2 Contrôles et comparaison des taux d'erreur

Contrôle de l'erreur

On distingue généralement deux types de contrôle d'un taux d'erreur : le contrôle « fort » et le contrôle « faible ». On parle de contrôle fort si l'erreur de type I est contrôlée quelle que soit la combinaison des hypothèses vraies et des hypothèses fausses c'est-à-dire quelle que soit la valeur de p_0. En revanche, on parle de contrôle faible si on se contente de contrôler l'erreur de type I quand toutes les hypothèses nulles sont vraies, c'est-à-dire quand $p_0 = p$. De façon générale, si on n'a pas d'autres garanties, un contrôle faible n'est pas satisfaisant. Dans le cadre des biopuces, où il y a peu de chances de n'avoir aucun gène différentiellement exprimé (on ne cherche pas à tester des groupes similaires), un contrôle « fort » semble très important, c'est le seul type de contrôle auquel nous nous intéresserons.

Il semble aussi important d'ajouter que l'hypothèse usuelle qui est faite dans le cadre de test multiples est de dire que les tests sont indépendants ce qui n'est pas le cas en pratique pour les données de biopuces. Cependant, nous aurons du mal à nous affranchir de cette hypothèse dans la mesure où le problème n'est pas encore résolu dans le cas de tests dépendants.

Comparaison des taux d'erreur de type I

De manière générale, pour une procédure de test multiple, on a :

$$PCER \leqslant FWER \leqslant PFER$$

Ainsi, pour un seuil α donné, l'ordre s'inverse pour le nombre de rejet R : les procédures qui contrôlent le PFER sont généralement plus conservatives que celles

qui contrôlent le FWER ou le PCER, et les procédures qui contrôlent le FWER sont plus conservatives que celles qui contrôlent le PCER.
On a vu dans la définition de ces taux que le FWER (et par transitivité le PFER) conduisait à des tests trop conservatifs dans le contexte des biopuces alors que le PCER ne semblait pas un taux d'erreur suffisant à contrôler. Intéressons-nous maintenant au FDR.
Tout d'abord, il y a deux petites propriétés intéressantes concernant le FDR :

(a) si toutes les hypothèses nulles sont vraies, le FDR est équivalent au $FWER$

(b) sinon $FDR \leqslant FWER$

Ainsi toute procédure qui contrôle le $FWER$ contrôlera donc aussi le FDR. Donc si on cherche juste à contrôler le FDR, on pourra obtenir des procédures moins contraignantes (c'est-à-dire plus puissantes).
La propriété décisive est la suivante :

$$PCER \leqslant FDR \leqslant FWER$$

Le FDR permettra donc de trouver un bon compromis entre le PCER, qui peut conduire à un trop grand nombre de faux positifs et le FWER, qui minimise le nombre d'erreurs mais qui conduira à une puissance trop faible. C'est ce taux d'erreur, le FDR, qu'on cherchera à contrôler par la suite.

2.1.3 Importance du choix des statistiques de test

Une étape importante de la construction d'un test consiste en le choix et le calcul de la statistique de test que l'on va utiliser. Il faut essayer de choisir la statistique de test la plus représentative de l'hypothèse qu'on cherche à tester. Rappelons tout d'abord que, pour chaque gène i, on souhaite tester l'hypothèse H_{0i} : « le gène i n'est pas différentiellement exprimé entre les populations 1 et 2 » contre l'hypothèse H_{1i} : « le gène i est différentiellement exprimé entre les populations 1 et 2 ».
Soient μ_i^1 et μ_i^2 les vraies valeur de l'expression moyenne du gène i dans les populations 1 et 2 respectivement.

On considère qu'on s'intéresse à un test de comparaison des moyennes.

$$H_{0i} \ : \mu_i = \mu_i^1 - \mu_i^2 = 0 \ , \quad i = 1, \ldots, p$$

contre

$$H_{1i} \ : \mu_i = \mu_i^1 - \mu_i^2 \neq 0 \ , \quad i = 1, \ldots, p$$

La première idée intuitive est d'estimer μ_i^1 et μ_i^2 respectivement par $\overline{\mu}_{in}^1$ et $\overline{\mu}_{in}^2$, où $\overline{\mu}_{in}^k$ désigne un estimateur de la moyenne calculé à partir des données pour la population $k = 1, 2$. La loi des grands nombres permet en particulier de justifier de

prendre comme estimateur la moyenne empirique :

$$\overline{\mu}_{in}^{k} = \frac{1}{n_k} \sum_{j=1}^{n_k} X_{ij}^{k}$$

On peut alors choisir tout simplement comme statistique de test Z la différence des estimations des moyennes, comme suggéré par exemple par Pollard et van der Laan dans [34] et [33], c'est-à dire-qu'on prend :

$$Z_i = \overline{\mu}_{in}^{1} - \overline{\mu}_{in}^{2}$$

Si H_{0i} est vraie, on aura une statistique Z_i centrée avec des valeurs "proches" de zéro en valeur absolue.

Souvent, dans le calcul d'une statistique de test, on préfère utiliser des statistiques standardisées, pour connaître de façon explicite la loi de la statistique de test sous hypothèse nulle. Le t-test mesure si les moyennes de deux groupes sont différentes l'une de l'autre. Il en existe plusieurs versions selon les hypothèses que l'on peut formuler sur les échantillons à comparer. Le t-test le plus connu et le plus standard est le t-test de Student. La statistique de ce test est calculée de la façon suivante :

$$Z_i = \frac{\overline{\mu}_{in}^{1} - \overline{\mu}_{in}^{2}}{\sqrt{\frac{s_i^2}{n_1} + \frac{s_i^2}{n_2}}} \quad i \in \{1 \dots p\}$$

avec

$$s_i^2 = \frac{(n_1 - 1)(s_{in}^{1})^2 + (n_2 - 1)(s_{in}^{2})^2}{n_1 + n_2 - 2} \tag{2.1}$$

où s_{in}^{k} représente l'estimateur non biaisé de la variance au sein du groupe k pour le gène i.

$$(s_{in}^{k})^2 = \frac{1}{n_k - 1} \sum_{j=1}^{n_k} (X_{ij}^{k} - \overline{\mu}_{in}^{k})^2 \quad k \in \{1, 2\}$$

Cependant, cette version du t-test nécéssite une hypothèse préalable d'égalité des variances, qui peut être vérifiée avec un test de Fisher-Snedecor (test de F) pour tester le rapport des variances.

Quand l'hypothèse d'égalité des variances n'est pas vérifiée, on peut utiliser l'approximation de Welch du t-test, pour lequel on calcule la statistique de test

$$Z_i = \frac{\overline{\mu}_{in}^{1} - \overline{\mu}_{in}^{2}}{\sqrt{\frac{(s_{in}^{1})^2}{n_1} + \frac{(s_{in}^{2})^2}{n_2}}} \quad i \in \{1 \dots p\}$$

Notons que dans le cas particulier où $n_1 = n_2$, les deux statistiques de t-test proposées ici coïncident.

Dans les deux cas, la statistique de test sera positive si la première moyenne est plus grande que la deuxième, négative si elle est plus petite. Plus sa valeur sera

éloignée de zéro, plus il sera vraisemblable de conclure que le gène correspondant est différentiellement exprimé. Cependant, il faut être capable de quantifier cette notion d'"éloigné de zéro" afin de pouvoir prendre une décision. Pour cela, il faudra connaître la loi de la statistique de test sous l'hypothèse nulle, ou à défaut, être en mesure de l'approcher.

Une autre statistique de test a été proposée par Turkheimer *et al.* [45] dans le cadre des analyses d'images médicales RMN. On reste sur l'idée intuitive d'une statistique de test proportionnelle à la différence entre les moyennes et réduite par un facteur d'échelle. On considère :

$$Z_i = \frac{\hat{\mu}_i^1 - \hat{\mu}_i^2}{s(X_{i\cdot}^1, X_{i\cdot}^2)}$$

où $s(X_{i\cdot}^1, X_{i\cdot}^2)$ est un estimateur d'échelle et $\hat{\mu}_i^k$ un estimateur approprié de la moyenne μ_i^k.

L'originalité de la démarche proposée réside justement dans le choix de ce $\hat{\mu}_i^k$ ($k \in \{1, 2\}$). Classiquement on peut choisir la moyenne empirique de l'échantillon, c'est elle qui a été définie plus haut sous le nom $\overline{\mu}_{in}^k$ ou la médiane empirique de l'échantillon que nous noterons $\tilde{\mu}_{in}^k$. La médiane présente l'avantage d'être plus robuste en cas de valeurs extrêmes, c'est-à-dire quand on a des distributions à queues plus lourdes qu'une distribution normale, ce qui est souvent le cas pour les données d'expression génétique. En effet, contrairement à la moyenne empirique, la médiane est un estimateur robuste de la tendance centrale d'une distribution. Elle n'est pas affectée par des valeurs très petites ou très grandes qui peuvent par contre biaiser le calcul de la moyenne empirique.

Moyenne empirique et médiane empirique sont toutes deux des caractéristiques de tendance centrale. Choisir la moyenne, c'est choisir de minimiser l'écart quadratique, choisir la médiane, c'est minimiser l'écart absolu moyen.

$$\overline{\mu}_{in}^k = arg\min_x EQM(x) = arg\min_x \sum_{j=1}^{n_k} \left(X_{ij}^k - x\right)^2$$

$$\tilde{\mu}_{in}^k = arg\min_x EAM(x) = arg\min_x \sum_{j=1}^{n_k} \left|X_{ij}^k - x\right|$$

Plutôt que de faire un choix entre médiane et moyenne, Turkheimer *et al.* utilisent une approche assez intuitive initialement développée par Samuel-Cahn en 1994 [39] : une combinaison linéaire de ces deux estimateurs.

$$(\hat{\mu}_i^k)_\lambda = \lambda \overline{\mu}_{in}^k + (1 - \lambda)\tilde{\mu}_{in}^k \qquad \lambda \geq 0$$

où λ est un facteur de pondération.

On appelera cet estimateur $(LCMM(X_i^k))_\lambda$, c'est-à-dire, en anglais, la Combinaison Linéaire de la Moyenne et de la Médiane. Cet estimateur présente de nombreux avantages. En faisant varier la valeur de λ, il peut tirer profit de la bonne

efficacité de la moyenne empirique quand la distribution est proche d'une distribution normale et cependant bénéficier de la robustesse de la médiane quand les queues de distribution sont plus lourdes et/ou qu'il y a des valeurs extrêmes.

Le facteur de pondération λ peut prendre n'importe quelle valeur positive. Dans le cas d'une distribution avec des queues plus légères qu'une gaussienne, la valeur optimale de λ sera supérieure à 1 ; par exemple, le mélange optimal entre la moyenne et la médiane pour estimer l'espérance d'une distribution uniforme a un facteur de pondération égal à 3/2.

Quand on s'attend à avoir des données dont la loi a des queues au moins aussi lourdes qu'une loi normale, les valeurs optimales de λ sont situées dans l'intervalle $0 \leq \lambda \leq 1$.

Pour calculer la statistique de test, nous avons aussi besoin d'un estimateur d'échelle $s(X_{i\cdot}^1, X_{i\cdot}^2)$. Il existe une large variété d'estimateurs robustes de l'écart-type ; en raison de sa robustesse dans le cadre des échantillons de petite taille, on peut choisir un estimateur nommé l'écart-type empirique agrégé proposé par D'Agostino et Cureton en 1973 [12]. La statistique de test Z a alors la forme finale :

$$(Z_i)_\lambda = \frac{(LCMM(X_{i\cdot}^1))_\lambda - (LCMM(X_{i\cdot}^2))_\lambda}{s(X_{i\cdot}^1, X_{i\cdot}^2)}$$

avec

$$s(X_{i\cdot}^1, X_{i\cdot}^2) = \sqrt{\frac{\sum_{j=1}^{n_1}(X_{ij}^1 - \overline{\mu}_{in}^1)^2 + \sum_{j=1}^{n_2}(X_{ij}^2 - \overline{\mu}_{in}^2)^2}{n_1 + n_2 - 2}}$$

ce qui est en fait égal au s_i défini pour la statistique de t-test standard (voir équation 2.1).

La dernière étape pour être en mesure de calculer la statistique de test est de déterminer la valeur optimale de λ en fonction des données. On utilise une technique appelée MVA (Variance Minimum Adaptative) (Turkheimer *et al.* [46] [44]) qui appartient à la classe des techniques d'estimation adaptatives par « bootstrap » originellement introduites par Léger et Romano en 1990 dans [30]. Il a été montré que la technique MVA avait de bonnes propriétés pour des échantillons petits pour une utilisation avec divers estimateurs robustes.

On fixe une grille de λ (dans les programmes, 10 valeurs régulièrement espacées entre 0 et 3) et pour chaque valeur possible du paramètre on réalise un certain nombre de permutations (100 dans nos applications) des données observées, cela signifie que l'on tire aléatoirement avec remise n_1 expériences dans le premier échantillon pour constituer la population 1 et n_2 expériences dans le deuxième échantillon pour constituer la population 2. Pour ces données, on calcule les Z_i. Pour chaque λ, on estime la variance de $(Z_i)_\lambda$ sur l'ensemble des permutations, c'est-à-dire sur un certain nombre de tirages aléatoires des populations effectués. On choisit alors, en utilisant un lissage de la variance en fonction de λ, le λ pour lequel la variance est minimum (l'existence de ce minimum est montrée par Léger

et Romano dans [30] si on a une distribution symétrique). On doit déterminer un λ différent pour chaque gène (voir algorithme 2.1).

Algorithme 2.1. Détermination du λ optimal

— *algorithme de détermination de λ optimal pour un gène donné*
0 - On fixe une grille *lambda.grid* régulièrement espacée entre 0 et 3.
1 - Pour *lambda* dans *lambda.grid*
Pour $i = 1 \ldots nb\_permutation$
Tirer la ième permutation des échantillons
Calculer la statistique de test Z_λ correspondante et la mémoriser
Calculer la variance de Z_λ sur l'ensemble des permutations et la mémoriser.
2 - λ optimal = λ qui minimise une fonction de lissage de la variance de Z_λ en fonction de λ.

Nous disposons donc de plusieurs statistiques de test possibles, et nous les essaierons toutes par la suite afin de pouvoir les comparer, mais pour être en mesure de déterminer les p-valeurs correspondantes, nous avons besoin de connaître la loi de la statistique de test sous hypothèse nulle, c'est-à-dire lorsque le gène n'est pas différentiellement exprimé.

Soit π_i la p-valeur pour le $i_{ème}$ gène, on a par définition :

$$\begin{aligned} \pi_i &= \mathbb{P}_{H_0}(Rejet\ de\ H_{0i}) \\ &= \mathbb{P}_{H_0}(|Z| \geq |Z_i|) \end{aligned}$$

Notons F_0 la distribution de la statistique de test sous l'hypothèse nulle. Si la loi est symétrique ($F_0(x) = 1 - F_0(-x)$) alors on a :

$$\pi_i = 2(1 - F_0(|Z_i|))$$

Etant donné que ces statistiques de test sont appliquées à des données préalablement normalisées et standardisées, en pratique on peut s'attendre à avoir effectivement une loi symétrique. Dans le cas de la statistique de t-test de Student, si dans chaque classe les données sont distribuées selon une loi normale homoscédastique alors la distribution nulle est une loi de Student à $n_1 + n_2 - 2 = n - 2$ degrés de liberté. Pour le t-test de Welch, où on suppose en fait que les données de chaque classe sont distribuées selon une loi normale de moyenne nulle mais de variances différentes (modèle hétéroscédastique), sous l'hypothèse nulle, la statistique de test suit aussi une loi de Student. Le nombre de degrés de liberté *df* de cette loi n'est pas connu

exactement mais peut être estimé par l'approximation de Welch-Satterthwaite, à savoir

$$df = \frac{\left[\frac{(s^1_{in})^2}{n_1} + \frac{(s^2_{in})^2}{n_2}\right]^2}{\frac{((s^1_{in})^2/n_1)^2}{n_1-1} + \frac{((s^2_{in})^2/n_2)^2}{n_2-1}}.$$

Quand les variances sont égales, le calcul se réduit à $df = n_1 + n_2 - 2$ si les deux groupes ont le même nombre d'observation ($n_1 = n_2$), par contre, si ces nombres diffèrent, on obtient une valeur inférieure, ce qui rend le t-test avec la correction de Welch plus conservatif. Ceci est d'autant plus vrai que le déséquilibre entre les effectifs des deux classes est grand. Ici, on préfèrera donc supposer l'égalité des variances et appliquer un t-test standard, qui semble préférable dans le cas de petits échantillons.

Précisons aussi que si le nombre de degrés de liberté est suffisamment grand ($df > 30$) alors la loi de Student peut être approchée de façon satisfaisante par une loi normale centrée réduite. Le nombre de degrés de liberté et grand quand les tailles d'échantillons augmentent, on connaît alors la loi sous l'hypothèse nulle sans avoir besoin de faire l'hypothèse d'échantillons gaussiens.

Souvent, dans le calcul d'une statistique de test, on préfère utiliser des statistiques standardisées, pour connaître de façon explicite la loi de la statistique de test sous hypothèse nulle. Pollard et Van der Laan ([34] et [33]) contestent ce choix dans le cadre des données de biopuces, car souvent, pour ce genre de données, les lois sous l'hypothèse nulle sont assez éloignées d'une loi normale centrée réduite, et ce même pour des tailles d'échantillons relativement importantes. Il n'est alors pas nécessaire d'utiliser des statistiques de test standardisées. Quand on ne connaît pas la loi de la statistique de test sous l'hypothèse nulle, on peut également déterminer la p-valeur par une méthode de rééchantillonnage (expliquée par Westfall et Young dans [47]), c'est d'ailleurs ce que proposent Pollard et van der Laan. Si l'hypothèse nulle est vraie, alors le gène n'est pas différentiellement exprimé entre les deux conditions expérimentales, et le résultat ne devrait alors pas changer si on « mélange » les expériences. En pratique, cela signifie que, pour calculer la p-valeur, on réaffecte aléatoirement les résultats d'expérience aux conditions expérimentales A ou B, et on calcule la statistique de test avec ce nouveau jeu créé artificiellement. En répétant ceci plusieurs fois, de manière indépendante, on obtient pour chaque gène une valeur qui serait celle obtenue sous l'hypothèse nulle. En effet, puisque les échantillons ont été mélangés, il ne devrait pas y avoir de différence entre les deux conditions expérimentales dans le pseudo-jeu de données. On utilise alors la définition de la p-valeur en comparant la statistique de test calculée à partir du « vrai » jeu de données à celles calculées sur les pseudo-jeux de données générés. La p-valeur calculée par rééchantillonnage sera alors égale à la proportion des pseudo-jeux de données obtenus par rééchantillonnage qui donnent une statistique de test au moins aussi extrême que la statistique de test calculée sur le vrai jeu de données. Il est à noter, que dans le cas de la statistique de Turkheimer, on a le paramètre λ à fixer pour chaque gène. Pour calculer la statistique sous hypothèse nulle par rééchantillonnage, on garde comme paramètre λ celui calculé lors du calcul de la statistique de test observée sur

nos jeux de données.

L'inconvénient de ces méthodes réside essentiellement dans le temps de calcul qu'elles nécessitent... il faut réaliser un grand nombre de permutations (c'est-à-dire générer un grand nombre de pseudo-jeux de données) pour avoir des résultats intéressants. Elles présentent néanmoins l'avantage non négligeable de permettre de calculer des p-valeurs sans hypothèse *a priori* sur la loi de la statistique de test.

Pour la statistique de différence des moyennes et pour la statistique de Turkheimer *et al.* , on étudiera plus longuement la loi à la fin de la section suivante.

Tests de comparaison multiple et sélection de modèles

2.2.1 Comparaison multiple et sélection de modèles

Le principe des procédures FDR a été développé dans le cadre des tests d'hypothèses multiples par Benjamini et Hochberg [8]. Etant donné un ensemble de p hypothèses parmi lesquelles il y en a p_0 de vraies et $p_1 = p - p_0$ de fausses, la méthode FDR identifie les hypothèses à rejeter, tout en gardant l'espérance du rapport du nombre de faux rejets par le nombre total de rejets sous un seuil q défini par l'utilisateur. Cette technique est appropriée surtout lorsque p est très grand comme dans le cadre des expériences de biopuces car elle est très économique au niveau calculatoire. Elle a été utilisée notamment dans le cadre du débruitage de signaux par décomposition ondelettes mais aussi dans le cadre de la génétique (Efron *et al.* 2001 [17] - Benjamini et Yekutieli [9]).
Dans cette partie, nous allons considérer de manière générale une famille de modèles $M_{\underline{\boldsymbol{\beta}}}$ indexés par un paramètre $\underline{\boldsymbol{\beta}} \in \mathbb{R}^p$ et interpréter la procédure FDR comme méthode d'estimation du sous-ensemble $I_1 \subseteq I_p \stackrel{def}{=} \{1, \ldots, p\}$ des composantes non-nulles de $\underline{\boldsymbol{\beta}}$. Le modèle $M_{\underline{\boldsymbol{\beta}}}$ est spécifié par :

$$M_{\underline{\boldsymbol{\beta}}} : \left| \begin{array}{ll} \beta_i \neq 0 \ , & \forall i \in I_1 \\ \beta_i = 0 \ , & \forall i \in C_{I_p}^{I_1} = I_p \backslash I_1 \end{array} \right.$$

Nous supposerons que $\underline{\boldsymbol{\beta}} \in \mathbb{R}^p$ peut être efficacement estimé. La stratégie adoptée sera donc :

1. Déterminer un estimateur de $\underline{\boldsymbol{\beta}} \in \mathbb{R}^p$.
2. Utiliser FDR pour estimer I_1.

Pour l'étape 2, remarquons que le problème d'estimation de I_1 peut être formulé comme celui du test de l'hypothèse multiple :

$$H_0 : \beta_1 = 0 \ , \ \ldots \ , \ \beta_p = 0$$

Toute méthode identifiant les hypothèses qui pourront être rejetées donnera une estimation de I_1.

Algorithme 2.2. Procédure FDR - Benjamini Hochberg

Etape 0 - Il faut d'abord déterminer des statistiques de tests $Z_1, \ldots, Z_p$ basées sur les estimateurs $\beta_1, \ldots, \beta_p$ de p-valeurs respectives $\pi_1, \ldots, \pi_p$ (on est dans le cas d'un test bilatéral, on a $\forall i$, $\pi_i = 2(1 - \Phi(|Z_i|))$.
Ensuite, pour tout $q \in]0, 1[$ fixé, effectuer les étapes suivantes :

Etape 1 - Ranger les p-valeurs par ordre croissant $\pi_{(1)} \leq \ldots \leq \pi_{(p)}$

Etape 2 - Calculer $k = \max\{i \mid \pi_{(i)} \leq \frac{i}{p}q\}$

Etape 3 - Rejeter $H_{0(j)} : \beta_{(j)} = 0$, $\forall j \in \{1, \ldots, k\}$ où $H_{0(j)}$ est l'hypothèse nulle associée à la p-valeur ordonnée $\pi_{(j)}$.
Si un tel k n'existe pas, on ne rejette aucune hypothèse.

Etape 4 - Estimer I_1 par $\hat{I}$, l'ensemble des indices des k premières p-valeurs ordonnées.

La procédure FDR, telle qu'elle est suggérée par Benjamini et Hochberg [8] est présentée dans l'algorithme 2.2.

Nous verrons plus tard que dans le cadre des biopuces l'estimation de I_1 peut se faire également à l'aide de techniques de pénalisation puisqu'il existe une relation importante entre le FDR et la pénalisation (*cf* Abramovich *et al.* [3] et Golubev [22]), malgré le fait que ces papiers ne s'intéressent qu'à l'estimation minimax de $\underline{\boldsymbol{\beta}}$ et non à l'estimation de I_1.

2.2.2 Convergence de l'estimateur FDR de I_0

Soit $0 \leq R \leq p$ le nombre total d'hypothèses nulles rejetées et soit $0 \leq V \leq R$ le nombre de rejets faits à tort. A condition que les statistiques de test $Z_1, Z_2, \ldots, Z_p$ soient indépendantes, Benjamini et Hochberg [8] montrent que ;

$$\mathbb{E}(Q) \leq \frac{p_0}{p} q \tag{2.2}$$

où

$$Q = \begin{cases} \frac{V}{R} & si\ R > 0 \\ 0 & sinon \end{cases}$$

et p_0 est le nombre de vraies hypothèses nulles.

Benjamini et Yekutieli [9] ont montré que cela reste vrai si le vecteur $\mathbf{Z} = (Z_1, \ldots Z_p)^T$ des statistiques de test sous H_0 vérifie la propriété PRDS suivante.

Définition 2.1. Propriété PRDS - Benjamini Yekutieli
Pour tout ensemble croissant D et pour tout $i \in I_0$, $\mathbb{P}(\mathbf{Z} \in D | Z_i = t)$ est non décroissante en t

Ils ont également montré que (1) reste vraie sans hypothèse particulière sur les Z_i, à condition de conduire la procédure avec $\frac{q}{\sum_{i=1}^{p} i^{-1}}$ à la place de q (Théorème 3.1 de [9]).
Comme $|I_1| = p_1$, la procédure FDR conduira à un estimateur $\hat{I}$ consistant de I_1 si et seulement si on a p_1 rejets ($R = p_1$) et qu'aucun de ces rejets n'est erronné ($V = 0$).
Montrer la consistance de $\hat{I}$ revient donc à montrer que :

$$\mathbb{P}(\hat{I} = I_1) = \mathbb{P}(R = p_1 \cap V = 0) \xrightarrow[n \to +\infty]{} 1$$

où n est un paramètre jouant le rôle de la taille d'échantillon dans le calcul des Z_i. Pour cela, nous nous restreignons à des estimateurs de $\underline{\boldsymbol{\beta}}$ qui, sous H_0, seront asymptotiquement normaux. En fait, on supposera dans la suite que :

Hypothèse 1 (HT)

$$Z_i - \mu_i \xrightarrow[n \to +\infty]{\mathcal{L}} \mathcal{N}(0, 1)$$

pour une suite μ_i, dépendante de n, telle que :
(a) $\limsup_{n\to+\infty} |\mu_i| = 0 \qquad \forall i \notin I_1$
(b) $\liminf_{n\to+\infty} \frac{|\mu_i|}{b_n} > 1 \quad \forall i \in I_1$ *pour une certaine suite* b_n *telle que* $b_n \xrightarrow[n\to\infty]{} +\infty$

Remarque. 1 Dans la plupart des cas, $Z_i = \hat{\beta}_i/e.t.(\hat{\beta}_i)$ et $\mu_i = \beta_i/e.t.(\hat{\beta}_i)$ où $e.t.(\hat{\beta}_i)$ désigne l'écart-type de $\hat{\beta}_i$. Quand la taille n de l'échantillon tend vers l'infini, $e.t.(\hat{\beta}_i)$ tend vers zéro. Dans ce cas, la condition (HT)(b) devient :

$$\liminf_{n\to+\infty} |\beta_i|/(e.t.(\hat{\beta}_i)b_n) > 1$$

pour une suite b_n telle que $b_n \xrightarrow[n\to\infty]{} +\infty$.

Grâce à (HT), on peut calculer les p-valeurs (asymptotiques) :

$$\pi_i = \mathbb{P}\{|\mathcal{N}(0,1)| \geq |Z_i|\} = 2(1 - \Phi(|Z_i|)),$$

où, sous H_0, $Z_i \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}(0,1)$ et Φ désigne la fonction de répartition pour une loi normale centrée réduite.

Notons tout d'abord que $\forall i \notin I_1,\ \beta_i = 0$ et donc :

$$\pi_i \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{U}[0,1]$$

puisque chaque π_i est calculé sous hypothèse nulle $H_{0(i)}$. Par contre, si $i \in I_1$, alors $\beta_i \neq 0$ et $\pi_i = 2(1 - \Phi(|Z_i|))$ ne converge pas nécessairement vers une distribution uniforme sur$[0,\ 1]$. Si maintenant, $\forall i \in I_1$, $|Z_i| \xrightarrow[n\to+\infty]{P} +\infty$ on a alors

$$\pi_i \xrightarrow[n\to+\infty]{P} 0$$

et donc on s'attendra à ce que les premières petites p-valeurs observées correspondent aux π_i tels que $i \in I_1$.
On a le résultat suivant :

Proposition 2.2. .
Soit $I_1 = \{j_1, \ldots, j_{p_1}\}$ *et* $R_n = (\{\pi_{(1)}, \ldots, \pi_{(p_1)}\} = \{\pi_{j_1}, \ldots \pi_{j_{p_1}}\})$.
Alors, sous les hypothèses (HT), on a :

$$\mathbb{P}(R_n) \to 1 \quad quand \quad n \to +\infty$$

PREUVE — D'après la condition (a) de (HT), on a :

$$\forall i \notin I_1 \ , \ \pi_i \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{U}[0,1]$$

La condition (b) de (HT) entraîne que :

$$\forall i \in I_1 \ , \ \pi_i \xrightarrow[n\to+\infty]{\mathcal{P}} 0$$

Maintenant, $\forall i \notin I_1$ et $\forall j \in I_1$, on a :

$$\lim_{\varepsilon\to 0} \limsup_{n\to\infty} \mathbb{P}(\pi_i < \pi_j) \leq \lim_{\varepsilon\to 0} \limsup_{n\to\infty} (\mathbb{P}(\pi_i \leq \varepsilon) + \mathbb{P}(\pi_j > \varepsilon)) = 0$$

De plus, on remarque que :

$$\mathbb{P}(R_n^C) \leq \sum_{j\in I_1} \sum_{i\in I_p\setminus I_1} \mathbb{P}(\pi_i < \pi_j)$$

D'où le résultat :

$$\lim_{n\to\infty} \mathbb{P}(R_n) = \lim_{n\to\infty} (1 - \mathbb{P}(R_n^C)) = 1 - \lim_{n\to\infty} \mathbb{P}(R_n^C) = 1$$

⊗

Cette proposition signifie que, sous des conditions appropriées, les p_1 premières p-valeurs ordonnées, $\pi_{(1)} \leq \ldots \leq \pi_{(p_1)}$, correspondent asymptotiquement aux π_j avec $j \in I_1$ alors que les $p_0 = p - p_1$ restantes correspondent aux π_j avec $j \notin I_1$.

Le résultat suivant établit que la procédure FDR est consistante si on choisit $q = q_n \xrightarrow[n\to+\infty]{} 0$. La vitesse de convergence dépendra du taux avec lequel $|\mu_i|$ tend vers l'infini.

Proposition 2.3. Consistance de la procédure FDR .
Si (HT) est vraie et si $q_n \xrightarrow[n\to+\infty]{} 0$ *satisfait*

$$\forall j \in I_1 \ , \ \lim_{n\to+\infty} \frac{1 - \Phi(|\mu_j|)}{q_n} = 0$$

Alors :

$$\lim_{n\to+\infty} \mathbb{P}(R = p_1 \cap V = 0) = 1$$

Remarque. 1 Une situation typique est $\mu_i = \frac{\beta_i}{e.t.(\hat{\beta}_i)}$ où $e.t.(\hat{\beta}_i) = O(\frac{1}{\sqrt{n}})$. Dans ce cas, on prend $b_n = \sqrt{n}$. Si b_n est un polynôme en n, on peut choisir $q_n = n^{-\alpha}$

avec $\alpha > 0$.

Remarque. 2 On peut remplacer la loi normale Φ dans (HT) par n'importe quelle loi symétrique F_0 et le résultat reste vrai, à condition de choisir q_n tel que :

$$\forall j \in I_1 \ , \ \lim_{n\to+\infty} \frac{1 - F_0(|\mu_j|)}{q_n} = 0$$

PREUVE — Par définition de Q, pour $p_1 \geq 1$, on a

$$\{R = p_1, V = 0\} = \{R = p_1, Q = 0\}.$$

On veut montrer que $\lim_{n\to+\infty} \mathbb{P}(R = p_1 \cap V = 0) = 1$. Or on a :

$$\begin{aligned} \mathbb{P}(R = p_1 \cap V = 0) &= \mathbb{P}(R = p_1 \cap Q = 0) \ , \ si \ p_1 \geq 1 \\ &= 1 - \mathbb{P}(R \neq p_1 \cup Q \neq 0) \end{aligned}$$

Par conséquent pour montrer que $\lim_{n\to+\infty} \mathbb{P}(R = p_1 \cap V = 0) = 1$, on va montrer que $\lim_{n\to+\infty} \mathbb{P}(R \neq p_1 \cup Q \neq 0) = 0$.

On a de façon évidente : $\mathbb{P}(R \neq p_1 \cup Q \neq 0) \leq \mathbb{P}(R \neq p_1) + \mathbb{P}(Q \neq 0)$.
Or, en conditionnant par R, on a :

$$\begin{aligned} \mathbb{P}(Q \neq 0) &= \mathbb{P}(Q \neq 0 \cap R \neq p_1) + \mathbb{P}(Q \neq 0 \cap R = p_1) \\ &= \mathbb{P}(R \neq p_1)\mathbb{P}(Q \neq 0/R \neq p_1) + \mathbb{P}(R = p_1)\mathbb{P}(Q \geq 1/R = p_1) \\ &\leq \mathbb{P}(R \neq p_1) + \mathbb{P}(QR \geq 1/R = p_1) \\ &\leq \mathbb{P}(R \neq p_1) + \mathbb{P}(Q \geq \frac{1}{p_1}) \end{aligned}$$

De plus, l'inégalité de Markov donne : $\mathbb{P}(Q \geq \frac{1}{p_1}) \leq p_1\mathbb{E}(Q)$
D'où :

$$\mathbb{P}(R \neq p_1 \cup Q \neq 0) \leq 2\mathbb{P}(R \neq p_1) + p_1\mathbb{E}(Q)$$

Le théorème de Benjamini-Yekutieli nous donne :

$$\mathbb{E}(Q) \leq \frac{p_0}{p} q$$

Ici, c'est bien de p_0, le vrai nombre d'hypothèses nulles, dont il s'agit.
On a alors, grâce à ce théorème :

$$\mathbb{P}(R \neq p_1 \cup Q \neq 0) \leq 2\mathbb{P}(R \neq p_1) + \frac{p_1 p_0}{p} q$$

On rappelle qu'on veut avoir :

$$\lim_{n\to+\infty} \mathbb{P}(R \neq p_1 \cup Q \neq 0) = 0$$

Le terme $\frac{p_1 p_0}{p} q$ tend vers 0 comme q.

– Dans le cas $p_1 \geq 1$ (celui qu'on vient de traiter), il suffit donc de montrer que :

$$\mathbb{P}(R \neq p_1) \xrightarrow[n\to\infty]{} 0$$

– Si on a $p_1 = 0$, alors $\mathbb{P}(R = p_1 \cap V = 0) = \mathbb{P}(R = 0 \cap V = 0) = \mathbb{P}(R = 0)$ qu'on veut voir tendre vers 1 quand $p \to +\infty$. Par conséquent, dans ce cas aussi, il suffit de montrer que $\mathbb{P}(R \neq p_1) \xrightarrow[n\to\infty]{} 0$.

Posons $q_p = \frac{q_n}{\sum_{i=1}^{p} i^{-1}}$. On a alors :

$$\{R = p_1\} = \{\pi_{(p)} > q_p \ , \ \pi_{(p-1)} > q_p \frac{p-1}{p} \ , \ \ldots \ , \ \pi_{(p_1+1)} > q_p \frac{p_1+1}{p} \ , \ \pi_{(p_1)} \leq q_p \frac{p_1}{p}\}$$

Or :

$$\mathbb{P}(R \neq p_1) = \mathbb{P}(\{R \neq p_1\} \cap R_n) + \mathbb{P}(\{R \neq p_1\} \cap R_n^C)$$

et :

$$\mathbb{P}(\{R \neq p_1\} \cap R_n^C) \leq \mathbb{P}(R_n^C)$$

On a :

$$\begin{aligned} \mathbb{P}(\{R \neq p_1\} \cap R_n) &= \mathbb{P}((\{\pi_{(p_1)} > q_p \frac{p_1}{p}\} \cup \bigcup_{j=p_1+1}^{p} \{\pi_{(j)} \leq q_p \frac{j}{p}\}) \cap R_n) \\ &\leq \mathbb{P}(\{\pi_{(p_1)} > q_p \frac{p_1}{p}\} \cap R_n) + \sum_{j=p_1+1}^{p} \mathbb{P}(\{\pi_{(j)} \leq q_p \frac{j}{p}\} \cap R_n) \end{aligned}$$

D'où :

$$\mathbb{P}(R \neq p_1) \leq \mathbb{P}(\{\pi_{(p_1)} > q_p \frac{p_1}{p}\} \cap R_p) + \sum_{j=p_1+1}^{p} \mathbb{P}(\{\pi_{(j)} \leq q_p \frac{j}{p}\} \cap R_p) + \mathbb{P}(R_p^C)$$

D'après la proposition 1, on a $\mathbb{P}(R_n^C) \xrightarrow[n\to\infty]{} 0$.
De plus :

$$\begin{aligned} \sum_{j=p_1+1}^{p} \mathbb{P}(\{\pi_{(j)} \leq q_p \frac{j}{p}\} \cap R_n) &\leq \sum_{j=p_1+1}^{p} \mathbb{P}(\{\pi_{(j)} \leq q_p\} \cap R_n) \\ &\leq \sum_{j \notin I_1} \mathbb{P}(\{\pi_j \leq q_p\}) \end{aligned}$$

Or :

$$\forall j \notin I_1 \ , \ \pi_j \xrightarrow[n\to\infty]{\mathcal{L}} \mathcal{U}[0,1]$$

donc

$$\forall j \notin I_1 \ , \ \mathbb{P}(\{\pi_j \leq q_p\} \xrightarrow[n\to\infty]{} q_p$$

En utilisant $|I_1| = p_1$,

$$\sum_{j \notin I_1} \mathbb{P}(\{\pi_j \leq q_p\}) = (p - p_1)q_p + o(1) \xrightarrow[n\to\infty]{} 0$$

Il nous reste donc à contrôler le dernier terme : $\mathbb{P}(\{\pi_{p_1} > q_p \frac{p_1}{p}\} \cap R_n)$.
On définit :

$$q_0 \stackrel{def}{=} q_p \frac{p_1}{2p} = \frac{q_n p_1}{2p \sum_{i=1}^{p} \frac{1}{i}}$$

Si R_n est vérifié, on a : $\pi_{(p_1)} = \max_{j \in I_1} \pi_j$. On a alors :

$$\begin{aligned}
\mathbb{P}(\{\pi_{(p_1)} > q_p \frac{p_1}{p}\} \cap R_n) &\leq \mathbb{P}(\max_{j \in I_1} \pi_j \geq 2q_0) \\
&\leq \sum_{j \in I_1} \mathbb{P}(\pi_j \geq 2q_0) \\
&\leq p_1 \max_{j \in I_1} \mathbb{P}(\pi_j \geq 2q_0) \\
&\leq p_1 \max_{j \in I_1} \mathbb{P}(1 - \Phi(|Z_j|) \geq q_0) \text{ car on a : } \pi_j = 2(1 - \Phi(|Z_j|))
\end{aligned}$$

Intéressons-nous maintenant au terme $\mathbb{P}(1 - \Phi(|Z_j|) \geq q_0)$ pour $j \in I_1$. On a :

$$\begin{aligned}
\mathbb{P}(1 - \Phi(|Z_j|) \geq q_0) &= \mathbb{P}(\Phi(|Z_j|) \leq 1 - q_0) \\
&= \mathbb{P}(|Z_j| \leq \Phi^{-1}(1 - q_0)) \\
&= \mathbb{P}(Z_j \leq \Phi^{-1}(1 - q_0)) - \mathbb{P}(Z_j \leq -\Phi^{-1}(1 - q_0))
\end{aligned}$$

Or avec l'hypothèse (HT), on a : $Z_j - \mu_j \xrightarrow[n\to\infty]{\mathcal{L}} \mathcal{N}(0, 1)$.
Donc : $\forall x$, $\mathbb{P}(Z_j \leq x) \xrightarrow[n\to\infty]{} \Phi(x - \mu_j)$
On en déduit :

$$\mathbb{P}(1 - \Phi(|Z_j|) \geq q_0) = \Phi(\Phi^{-1}(1 - q_0) - \mu_j) - \Phi(-\Phi^{-1}(1 - q_0) - \mu_j) + o(1)$$

Et donc :

$$\mathbb{P}(\{\pi_{(p_0)} > q_p \frac{p_1}{p}\} \cap R_n) \leq p_1 \max_{j \in I_1} \{\Phi(\Phi^{-1}(1-q_0) - \mu_j) - \Phi(-\Phi^{-1}(1-q_0) - \mu_j)\} + o(1)$$

Mais, avec HT (b), $\forall j \in I_0$, on a $\Phi^{-1}(1 - q_0) \xrightarrow[n\to\infty]{} +\infty$ et $|\mu_j| \xrightarrow[n\to\infty]{} +\infty$.
La divergence en $|\mu_j|$ sera plus rapide qu'en $\Phi^{-1}(1 - q_0)$ si et seulement si :

$$|\mu_j| \geq \Phi^{-1}(1 - q_0)$$

C'est-à-dire :

$$1 - \Phi(|\mu_j|) \leq q_0$$

Or, par hypothèse de la proposition :

$$\forall j \in I_0 \ , \ \lim_{n \to +\infty} \frac{1 - \Phi(|\mu_j|)}{q_n} = 0$$

et grâce à (HT) (b), on a :

$$\lim_{n\to+\infty} \frac{1-\Phi(|\mu_j|)}{q_0} \leq \lim_{n\to+\infty} K\frac{p\ logp}{q_n p_1} e^{-b_n^2} = 0$$

où K est une constante multiplicative. On conclut que :

$$\limsup_{n\to\infty} \mathbb{P}(\{\pi_{(p_1)} > q_p \frac{p_1}{p}\} \cap R_p) = 0$$

La démonstration est donc finie. ⊗

Commentaires sur les hypothèses de la proposition 2.3

On va ici s'interroger sur la validité des hypothèses (HT) pour les statistiques de test qui ont été proposées précédemment. Pour cela, il va falloir étudier les lois des statistiques de test proposées.

Pour la statistique de t-test, on peut vérifier sans problème les hypothèses. En effet, on rappelle le calcul de la statistique de t-test

$$Z_i = \frac{\overline{\mu}_{in}^1 - \overline{\mu}_{in}^2}{\sqrt{\frac{s_i^2}{n_1} + \frac{s_i^2}{n_2}}} \quad i \in \{1 \dots p\}$$

avec

$$s_i^2 = \frac{(n_1-1)(s_{in}^1)^2 + (n_2-1)(s_{in}^2)^2}{n_1+n_2-2}$$

Or, d'après le théorème central limite, on a

$$\overline{\mu}_{in}^k \overset{n_k\to+\infty}{\sim} \mathcal{N}\left(\mu_i^k, \frac{(\sigma_i^k)^2}{n_k}\right), \quad k \in \{1,2\}$$

où μ_i^k (resp. $(\sigma_i^k)^2$) désigne la vraie valeur de la moyenne (resp. la variance) pour le gène i et l'échantillon k.

Si on se place dans le cas asymptotique avec $n_1 \to +\infty$ et $n_2 \to +\infty$, on a alors, puisque les deux échantillons sont indépendants :

$$\overline{\mu}_{in}^1 - \overline{\mu}_{in}^2 \overset{n\to+\infty}{\sim} \mathcal{N}\left(\mu_i^1 - \mu_i^2, \frac{(\sigma_i^1)^2}{n_1} + \frac{(\sigma_i^2)^2}{n_2}\right)$$

Or dans le t-test standard, on fait l'hypothèse d'égalité des variances, et le s^2 défini est alors un estimateur de $(\sigma_i^1)^2$ et de $(\sigma_i^2)^2$.

On a alors, quand $n \to +\infty$:

$$Z_i - \frac{\mu_i^1 - \mu_i^2}{\sqrt{\frac{s_i^2}{n_1} + \frac{s_i^2}{n_2}}} \overset{n\to+\infty}{\sim} \mathcal{N}(0,1)$$

On se retrouve alors bien dans le cas mis en remarque après l'énoncé des hypothèses (HT) et de la propriété de consistance du FDR. On a $Z_i = \hat{\beta}_i / e.t.(\hat{\beta}_i)$ et on pose alors $\mu_i = \beta_i / e.t.(\hat{\beta}_i)$, c'est-à-dire qu'on pose

$$\mu_i = \frac{\mu_i^1 - \mu_i^2}{\sqrt{\frac{s_i^2}{n_1} + \frac{s_i^2}{n_2}}}$$

On a donc bien

$$Z_i - \mu_i \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}(0,1).$$

Si $i \notin I_1$, alors $\mu_i^1 = \mu_i^2$ et donc $\mu_i = 0$. La partie (a) des hypothèses (HT) est donc bien vérifiée.

Si $i \in I_1$, on utilise le fait que $e.t.(\hat{\beta}_i) = O(n/(n_1\ n_2))$. On prend alors $b_n = \sqrt{(n_1\ n_2)/n}$ et on peut choisir $q_n = ((n_1\ n_2)/n)^{-1/2}$. Si $n_1 = n_2$, cela revient à prendre $b_n = \sqrt{n}/2$ et à choisir $q_n = 2n^{-1/2}$.

Pour la statistique de différence des moyennes, on constate que les conditions pour avoir la consistance de la sélection de modèle par procédure FDR ne sont pas vérifiées. En effet, on a ici

$$Z_i = \overline{\mu}_{in}^1 - \overline{\mu}_{in}^2$$

et on a vu que

$$\overline{\mu}_{in}^1 - \overline{\mu}_{in}^2 \overset{n\to+\infty}{\sim} \mathcal{N}\left(\mu_i^1 - \mu_i^2, \frac{(\sigma_i^1)^2}{n_1} + \frac{(\sigma_i^2)^2}{n_2}\right)$$

Il est alors naturel de poser $\mu_i = \mu_i^1 - \mu_i^2$ puisqu'il s'agit de la limite de Z_i quand $n \to +\infty$, on aura alors bien la partie (a) des hypothèses (HT) qui sera vérifiée. En revanche, pour avoir la partie (b) des hypothèses, il faudrait avoir

$$\lim_{n\to+\infty} \frac{|\mu_i|}{b_n} > 1 \quad \forall i \in I_1$$

pour une certaine suite b_n telle que $b_n \xrightarrow[n\to\infty]{} +\infty$

Or μ_i est une constante qui ne dépend pas de n, donc pour toute suite b_n telle que $b_n \xrightarrow[n\to\infty]{} +\infty$, on aura $\liminf_{n\to+\infty} \frac{|\mu_i|}{b_n} = 0$.

Pour la statistique de Turkheimer, on n'arrive pas non plus à vérifier les hypothèses. Dans les calculs, on utilise des combinaisons linéaires entre la moyenne empirique $\overline{\mu}_{in}^k$ et la médiane empirique $\tilde{\mu}_{in}^k$. La loi asymptotique conjointe de la moyenne empirique $\overline{\mu}_n$ et de la médiane empirique $\tilde{\mu}_n$ a été étudiée dès le début du 19ème siècle par Laplace [29]. De façon générale, pour une distribution de fonction de répartition F, de densité f, de moyenne μ,de médiane ν et de variance σ^2, on a :

$$n^{1/2}\left[\overline{\mu}_n - \mu, \tilde{\mu}_n - \nu\right] \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}\left(0_{\mathbb{R}^2}, \Sigma\right)$$

où $\Sigma = (\sigma_{ij})$ désigne la matrice de covariance de taille 2×2 avec

$$\sigma_{11} = \sigma^2, \quad \sigma_{22} = \frac{1}{4f^2(\nu)}, \quad \sigma_{12} = \frac{\mathbb{E}|X - \nu|}{2f(\nu)}.$$

Dans le cas d'une distribution gaussienne, on a :

$$\sigma_{11} = \sigma^2, \quad \sigma_{22} = \sigma^2 \frac{\pi}{2}, \quad \sigma_{12} = \sigma^2.$$

Ainsi, une combinaison linéaire de la moyenne et de la médiane empirique suivra aussi une loi normale :

$$\lambda\overline{\mu}_n + (1-\lambda)\tilde{\mu}_n \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}\left(\lambda\mu + (1-\lambda)\nu \; ; \; \frac{1}{n}(\lambda^2\sigma_{11} + 2\lambda(1-\lambda)\sigma_{12} + (1-\lambda)^2\sigma_{22})\right)$$

Dans le cas d'une distribution gaussienne de moyenne μ et d'écart-type σ, on a :

$$\lambda\overline{\mu}_n + (1-\lambda)\tilde{\mu}_n \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}\left(\mu \; ; \; \frac{\sigma^2}{n}\left(\lambda^2 + 2\lambda(1-\lambda) + (1-\lambda^2)\pi/2\right)\right)$$

Ainsi, on peut connaître les lois de $(LCMM(X_{i\cdot}^1))_\lambda$ et de $(LCMM(X_{i\cdot}^2))_\lambda$, il s'agit de loi normales centrées respectivement en $\lambda\mu_i^1 + (1-\lambda)\nu_i^1$ et $\lambda\mu_i^2 + (1-\lambda)\nu_i^2$ et de variances que l'on peut calculer mais qui sont fonctions de certains paramètres inconnus comme l'écart-type ou la valeur de la densité en la médiane. Comme $(LCMM(X_{i\cdot}^1))_\lambda$ et $(LCMM(X_{i\cdot}^2))_\lambda$ sont des variables indépendantes, la différence suit également une loi normale d'espérance la différence des espérances et de variance la somme des variances. Si les distributions sont gaussiennes, on aura :

$$(LCMM(X_{i\cdot}^1))_\lambda - (LCMM(X_{i\cdot}^2))_\lambda \xrightarrow[n\to+\infty]{\mathcal{L}}$$

$$\mathcal{N}\left(\mu_i^1 - \mu_i^2 \; ; \; \left(\frac{(\sigma_i^1)^2}{n_1} + \frac{(\sigma_i^2)^2}{n_2}\right)\left(\lambda^2 + 2\lambda(1-\lambda) + (1-\lambda^2)\pi/2\right)\right)$$

Sous l'hypothèse nulle, c'est-à-dire si $i \notin I_1$, on a une loi normale centrée. Dans le cadre général d'une distribution non gaussienne, on a une expression pour la variance de la différence qui fait également intervenir les écart-types respectifs du groupe 1 et du groupe 2, mais aussi les densités pour les deux groupes en la médiane ν_i^k du gène i pour chaque groupe $k \in \{1, 2\}$, et l'espérance $\mathbb{E}|X_{i\cdot}^k - \nu_i^k|$. Pour estimer l'écart-type de la différence, il faudrait donc être en mesure d'estimer correctement tous ces paramètres. On pourrait alors utiliser cet écart-type estimé comme facteur d'échelle dans le calcul de la statistique de test.

Ce n'est pas ce qui a été fait par Turkheimer *et al.* , ils ont choisis volontairement de prendre l'écart-type empirique agrégé comme facteur d'échelle en raison disent-ils de sa robustesse pour les petites tailles d'échantillons, sans doute car l'estimation de la variance de $(LCMM(X_{i\cdot}^1))_\lambda - (LCMM(X_{i\cdot}^2))_\lambda$ pose des problèmes. Certes, si on est dans le cas gaussien, le calcul se fait simplement à partir des estimations des écart-types des deux échantillons et du λ calculé par bootstrap pour le gène i,et, si

on fait l'hypothèse d'égalité des variances, à partir de l'estimation s_i. Encore faut-il étudier alors la loi de la nouvelle statistique de test obtenue en divisant par un facteur d'échelle estimant la variance de la différence $(LCMM(X_{i\cdot}^1))_\lambda - (LCMM(X_{i\cdot}^2))_\lambda$.

Par conséquent, on a $(Z_i)_\lambda$ qui suit asymptotiquement une loi normale de variance approximativement de l'ordre de $1/n$ puisque le facteur d'échelle s_i^2 choisi pour le calcul de $(Z_i)_\lambda$ est de l'ordre d'une constante. Il est alors naturel de poser comme suite $\mu_i = \lambda(\mu_i^1 - \mu_i^2) + (1-\lambda)(\nu_i^1 - \nu_i^2)$. Pour tout $i \notin I_1$ on a alors bien la partie (a) de l'hypothèse (HT) qui sera vérifiée mais pour tout $i \in I_1$, μ_i est de l'ordre d'une constante, égale à un rapport entre la différence de caractéristiques centrales des deux échantillons et un facteur d'échelle qui estime l'écart-type des échantillons agrégés, ce qui implique qu'on ne pourra pas satisfaire la condition (b) de (HT).

Pour être dans les conditions de la proposition 2.3, il aurait fallu choisir comme facteur d'échelle un estimateur de l'écart-type de $(LCMM(X_{i\cdot}^1))_\lambda - (LCMM(X_{i\cdot}^2))_\lambda$. Ici, nous n'avons utilisé que la version de cette statistique de test telle qu'elle est proposée par Turkheimer *et al.* mais il pourrait être intéressant lors de travaux futurs de changer le facteur d'échelle pour se ramener aux conditions du théorème.

Cependant, pour la statistique de différence des moyennes comme pour la statistique de test proposée par Turkheimer *et al.* , on peut espérer pouvoir se ramener plus ou moins dans les conditions de la proposition dans la mesure où, dans les deux cas, on a une statistique de test qui suit une loi normale, dont l'écart-type tend vers 0 à une vitesse de l'ordre de $\sqrt{n}$. On peut en effet alors estimer une variance sur la statistique de test par exemple par méthode de rééchantillonnage et se replacer dans un cadre similaire à celui de la statistique de t-test en divisant la statistique de test calculée par ce facteur d'échelle. Remarquons quand même que dans le cas de la statistique de différence des moyennes, cela ne présentera pas grand intérêt dans la mesure où cela reviendra à se ramener à une sorte de statistique de t-test.

Remarque. Nous avons montré d'un point de vue théorique la consistance de la sélection de modèle par procédure FDR sous certaines hypothèses sur la statistique de test. Cependant, comme toujours dans le cadre des données de biopuces, on souffre du fléau de la dimension, on a un faible nombre n d'expériences de biopuces par rapport au nombre de gènes présents sur chaque biopuce p. Ainsi, en pratique on ne sera pas dans le cadre asymptotique puisque, pour les jeux de données réels que nous avons à notre disposition, n est de l'ordre de quelques dizaines. Nous utiliserons, pour les données réelles un seuil d'erreur q fixe, égal à 0.05.

Maintenant voyons sur quelques exemples le comportement de cette procédure de sélection de modèle.

2.2.3 Exemple d'application de la méthode

Nous allons ici donner un premier exemple d'application de la méthode proposée sur des données simulées. Les résultats de l'ensemble des méthodes proposées pour

la détection des gènes différentiellement exprimés feront l'objet d'une comparaison en fin de chapitre.

Nous nous intéressons uniquement ici à la performance de la méthode de sélection de modèle proposée. Dans un premier temps, nous allons simuler directement des données distribuées selon une loi normale, que nous utiliserons comme s'il s'agissait de la statistique de test calculée au préalable. Pour ces simulations, on travaillera avec un seuil $q \in \{0.01, 0.05, 0.1\}$ fixé, et on fera des simulations avec différents niveau de densité pour vérifier l'adaptabilité de la procédure. Dans un second temps, pour vérifier la consistance de la procédure quand $n \to +\infty$, nous simulerons des échantillons de taille $n_1 = n_2$ de plus en plus grands pour vérifer les performances de la méthode, en utilisant une statistique de t-test et en prenant $q = q_n$ qui tend vers 0 quand la taille d'échantillon tend vers l'infini.

2.2.3.1 Simulations avec q fixé

Il s'agit ici de donner une idée de la méthode sur un exemple assez simple. On va simuler un vecteur bruité avec un nombre réduit de composantes non nulles. Pour cela, on a utilisé un exemple cité dans [3].

On considère un vecteur de taille $p = 10\ 000$, et on a $p_1 = 10$ composantes non nulles. On prend $\mu_i = \mu_1 = 5.21$ pour $i = 1 \dots p_1/2 = 5$, $\mu_i = -\mu_1 = -5.21$ pour $i = p_1/2 \dots p_1 = 10$ et $\mu_i = 0$ si $i = 11 \dots p = 10\ 000$. On prend $\sigma_p^2 = 1$. On simule un vecteur $\underline{x} = (x_1, \dots, x_p)$ selon le modèle

$$x_i = \mu_i + \sigma_p \varepsilon_i \text{ avec } \varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}_p(0, 1)$$

A partir de ce vecteur $\underline{x}$, on crée un vecteur $\underline{y}$ qui est une permutation aléatoire de $\underline{x}$. Pour se placer dans le contexte du FDR et de la sélection de modèle, on va considérer que la statistique de test observée pour chacun des 10 000 gènes (puisque $p = 10\ 000$) est donnée par le vecteur $\underline{y}$. C'est à ce vecteur $\underline{y}$ qu'on va appliquer la méthode, en supposant ici que la statistique de test suit une loi normale centrée réduite sous l'hypothèse nulle (H_{0i} : gène i non différentiellement exprimé).

On simule 10 000 vecteurs $\underline{x}$ et donc 10 000 vecteurs $\underline{y}$ différents et à chaque fois on estime le modèle correspondant, c'est-à-dire l'ensemble des composantes non nulles de $\underline{y}$.

Pour un vecteur donné $\underline{y} = (y_1, \dots y_p)$, on calcule les p-valeurs $\pi_1, \dots \dots \pi_p$ selon la formule suivante :

$$\pi_i = 2(1 - \Phi(abs(y_i))).$$

On considère en fait que le vecteur $\underline{y}$ correspond aux valeurs observées d'une statistique de test dont on connaît la loi sous l'hypothèse nulle, ici une loi normale centrée réduite.

On range ces p-valeurs par ordre croissant $\pi_{(1)} \leq \dots \leq \pi_{(p)}$. On en déduit $k = \max\{i \mid \pi_{(i)} \leq \frac{i}{p} q\}$ et on rejette les hypothèses $H_{0(j)}$: $\beta_{(j)} = 0$ pour tout $j \in \{1 \dots k\}$. Si k n'existe pas, on rejette toutes les hypothèses. On prend $q = 5\%$.

On estime donc I_1 par $\hat{I}$, l'ensemble des indices des k premières p-valeurs ordonnées.

On refait ensuite deux études similaires, en changeant certains paramètres pour vérifier que la méthode s'adapte bien pour différentes proportions de composantes non nulles. On change aussi la moyenne des comosantes non nulles. En résumé, on a :

- première série de simulations : $p = 10\,000$, $p_1 = 10$, $\mu_1 = 5.21$
- deuxième série de simulations : $p = 10\,000$, $p_1 = 100$, $\mu_1 = 4.52$
- troisième série de simulations : $p = 10\,000$, $p_1 = 1\,000$, $\mu_1 = 3.11$

		Benjamini Hochberg		Benjamini Yekutieli	
p_1	q	FDR obs.	Puissance	FDR obs.	Puissance
10	0.01	0.0094	0.7729	0.0009	0.5909
	0.05	0.0496	0.8724	0.0047	0.7228
	0.10	0.0997	0.9076	0.0096	0.7745
100	0.01	0.0098	0.7086	0.0011	0.4808
	0.05	0.0493	0.8430	0.0051	0.6441
	0.10	0.0986	0.8898	0.0100	0.7106
1000	0.01	0.0091	0.3110	0.0009	0.0948
	0.05	0.0450	0.5521	0.0046	0.2297
	0.10	0.0900	0.6657	0.0948	0.3138

Table 2.2 – Résultats de simulation pour la sélection de modèles par méthode FDR. On a 10 000 simulations et $p = 10\,000$ gènes. On fait varier le seuil $q \in \{0.01, 0.05, 0.1\}$. p_1 désigne le vrai nombre de termes de moyenne non nulle. Pour $p_1 = 10$, on a $\mu_1 = 5.21$, pour $p_1 = 100$, on a $\mu_1 = 4.52$ et pour $p_1 = 1\,000$, on a $\mu_1 = 3.11$. On donne le FDR moyen et la puissance moyenne observés sur les 10 000 simulations.

Il convient maintenant d'analyser ces résultats. On remarque tout d'abord une nette différence entre les résultats obtenus par la méthode de Benjamini et Hochberg (BH) et celle de Benjamini et Yekutieli (BY). La méthode BY est bien plus contraignante, les taux de FDR observés sont de l'ordre de 10 fois inférieurs aux taux maximum autorisés. En fait, cette procédure a été introduite dans le cadre de données dépendantes selon une certaine structure mais ces jeux de données simulés sont en fait des échantillons, il est donc logique que BH, conçue pour des données indépendantes soit plus appropriée.

Les taux de FDR observées avec la méthode BH sont bons, très proches du seuil maximal autorisé, et ce pour les trois seuils utilisés, pour les trois proportions de composantes non nulles et pour les trois valeurs de μ_1. Cela montre que cette

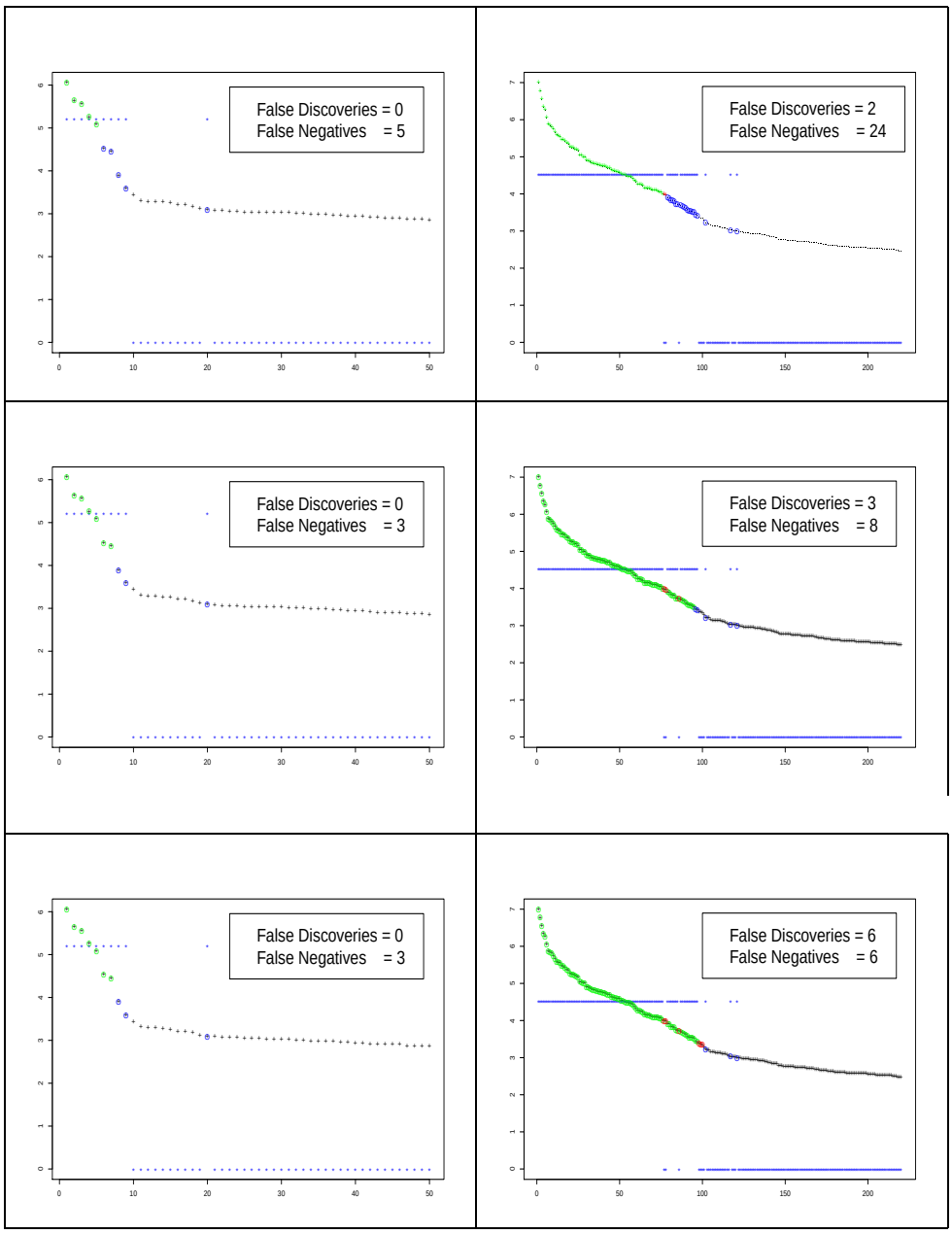

Figure 2.1 – Sélection de modèles par FDR. On représente ici les premières valeurs du vecteur ordonné des valeurs absolues des statistiques de test. De haut en bas : $q = 0.01,\ \ 0.05,\ \ 0.1$. On ne s'intéresse qu'à la procédure FDR par Benjamini et Hochberg ; à gauche avec $p_1 = 10$ et $\mu_1 = 5.21$, à droite avec $p_1 = 100$ et $\mu_1 = 4.52$. Les étoiles bleues désignent la vraie valeur de la moyenne correspondante. Les points entourés de vert désignent les vrais positifs (gènes détectés à raison), ceux entourés de rouge sont les faux positifs (détectés à tort) et ceux encerclés de bleu sont les faux négatifs (non détectés à tort).

méthode s'adapte très bien quelle que soit la proportion de composantes non nulles. Quand μ_1 diminue, c'est la puissance qui en pâtit, puisque les groupes sont davantage "mélangés", mais le FDR, est toujours contrôlé au taux souhaité.

2.2.3.2 Simulations avec $q = q_n \xrightarrow[n\to+\infty]{} 0$.

Ici, on génère deux échantillons de taille n_1 ou n_2 comportant chacun $p = 1\,000$ gènes. On fixe un niveau de densité égal à 10%, c'est-à-dire que sur les p gènes, 100 sont différentiellement exprimés entre les deux échantillons. Pour simplifier les choses, on prendra $n_1 = n_2 = n/2$. On veut vérifier de façon expérimentale la consistance de la procédure de sélection de modèles par FDR quand n tend vers $+\infty$.

On simule ces échantillons selon une loi gaussienne. Pour le premier échantillon, chacun des p gènes suit une loi normale centrée réduite $\mathcal{N}(0,1)$. Pour le deuxième échantillon, 900 gènes suivent une loi $\mathcal{N}(0,1)$, 50 une loi $\mathcal{N}(0.5,1)$ et 50 gènes suivent une loi $\mathcal{N}(-0.5,1)$. Sur ces échantillons simulés, on calcule une statistique de t-test, puis les p-valeurs correspondantes (ici on aura n suffisamment grand pour supposer que la loi sous l'hypothèse nulle est une loi normale centrée réduite).

Dans un premier temps, on applique alors la méthode de sélection de modèles proposée en considérant un seuil fixe $q = 0.05$. Dans un second temps, pour que le théorème de consistance de la procédure FDR s'applique, on prendra un seuil $q = q_n = 2n^{-1/2}$ qui tend vers 0 quand n tend vers $+\infty$.

On effectue les simulations pour $n_1 \in \{100, 500, 1\ 000, 5\ 000, 10\ 000\}$. Sur la figure 2.2, on représente le FDR moyen et la puissance moyenne observés sur 100 simulations en fonction de la taille de l'échantillon. On vérifie bien sur la figure la consitance de la procédure FDR proposée quand $n \to +\infty$.

On peut procéder de même en utilisant la procédure FDR de Benjamini et Yekutieli au lieu de celle de Benjamini et Hochberg, même si ici cela ne semble pas adapté puisqu'on simule des variables indépendantes. Cela revient à considérer comme seuil fixe $q = 0.05/\sum_{i=1}^{p} i^{-1}$ et comme seuil qui tend vers 0 quand $n \to +\infty$ $q = q_n = 2n^{-1/2}/\sum_{i=1}^{p} i^{-1}$. On observe alors un graphique avec des allures de courbes tout à fait similaires mais avec un FDR et une puissance observés plus faibles.

Conclusion

La méthode de sélection de modèles ainsi obtenue est donc satisfaisante, elle permet un bon contrôle du FDR moyen, et elle s'adapte bien à différents niveaux de densité de la statistique de test considéree. Comme cela a déjà été évoqué, il existe un lien important, quoique non immédiat, entre le FDR et la pénalisation. Nous allons maitenant envisager des techniques de sélection de modèles par pénalisation et comparer les résultats obtenus.

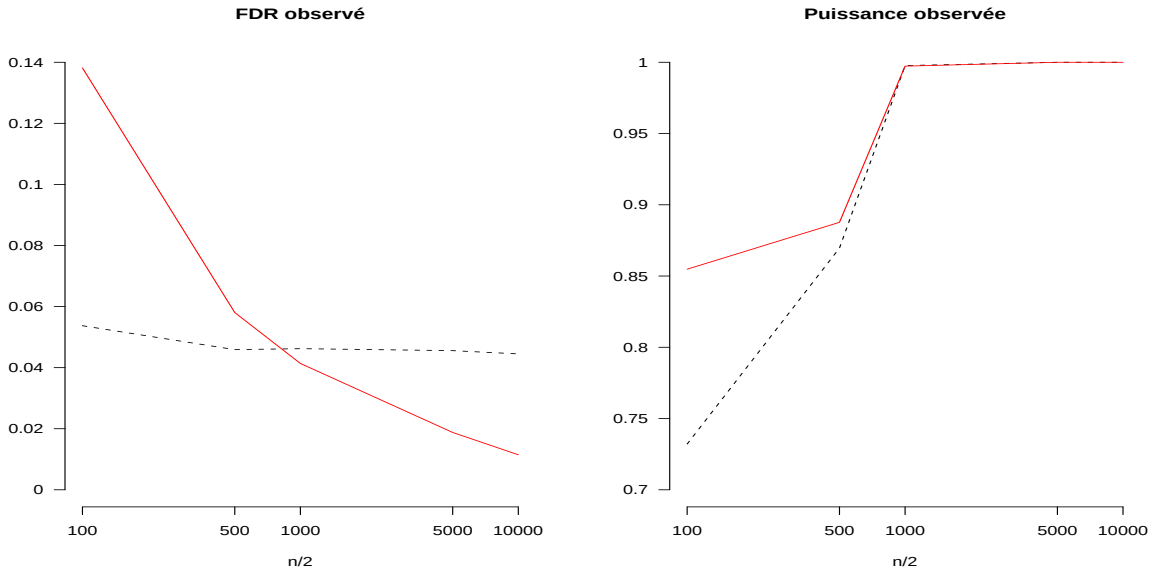

Figure 2.2 – Consistance de la procédure FDR quand n tend vers $+\infty$. On représente en échelle semi-logarithmique le FDR moyen observé (à gauche) et la puissance moyenne observée (à droite) sur 100 simulations. En trait plein, on représente les résultats obtenus en prenant $q = q_n = 2n^{-1/2}$ et en pointillés avec $q = 0.05$. Pour $q = q_n = 2n^{-1/2}$, on voit bien que le FDR tend vers 0 et la puissance tend vers 1, ce qui vérifie bien expérimentalement la consistance de la procédure.

Sélection de modèles et pénalisation

On reste ici dans le même contexte que dans le chapitre précédent et on considère le modèle de régression suivant :

$$Z_i = \mu_i + \sigma_p \varepsilon_i \text{ avec } \varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}_p(0,1)$$

avec σ_p inconnu. Ici, Z_i représente la statistique de test calculée pour le $i^{ème}$ gène. On suppose toujours que l'hypothèse HT est vérifiée (cf définition et discussion dans la section précédente).

Il s'agit ici de reconstruire le vecteur $\underline{\mu} = (\mu_1, \ldots, \mu_p)$ à partir de la statistique de test $\underline{Z} = (Z_1, \ldots, Z_p)$. Une chose importante dans tout problème statistique c'est de savoir quelle information a priori on se donne sur le vecteur que l'on cherche à estimer. Ici on considère le problème d'estimation d'un vecteur creux. Cela signifie d'une part que la dimension p du problème est assez grande et, d'autre part, que le nombre de composantes du vecteur estimé qui ont une grande valeur absolue est relativement petit par rapport à la dimension du problème et que les autres composantes ont des valeurs suffisamment faibles. Bien sûr, on ne sait pas quelles sont les composantes à grandes valeurs, et on ne connaît pas non plus le nombre de composantes qui sont très faibles. Si on connaissait ce degré de complexité du vecteur, le problème serait résolu différemment.

Ce genre de problème correspond tout à fait aux données de biopuces. On a calculé une statistique de test $\underline{Z}$ pour estimer l'expression différentielle des gènes. On a vu que si Z_i est significativement éloignée de zéro, on concluera que le i^{ème} gène est différentiellement exprimé, sinon, on concluera à un gène non différentiellement exprimé. Ainsi chercher les gènes différentiellement exprimés, c'est chercher les indices qui correspondent aux composantes de grande valeur absolue du vecteur creux $\underline{\mu}$. C'est pourquoi on cherche à estimer $\underline{\mu}$.

2.3.1 Le critère à minimiser

On définit $k_p < p$ et $\mathcal{L} = \{I \subset I_p \ ; \ \#I = k_I \leq k_p\}$. Notons pour tout $I \in \mathcal{L}$, $\underline{Z}_I$ le vecteur de $\mathbb{R}^p$ dont la ième composante vaut Z_i si $i \in I$ et 0 sinon.

Pour chaque $I \in \mathcal{L}$, en supposant que $I = J$, on calcule l'estimateur de maximum de vraisemblance de $\underline{\mu}$ noté $\underline{\hat{\mu}}_J$. Le problème est de choisir le « meilleur » $\hat{I}$ dans $\mathcal{L}$ et de prendre $\underline{\hat{\mu}} = \underline{\hat{\mu}}_{\hat{I}}$.

Remarque. Sous l'hypothèse $k_p < p$, on estime σ_p^2 par :

$$\hat{\sigma}^2_{I_{k_p}} = \frac{1}{p - \hat{k_p}} \sum_{i=\hat{k_p}+1}^{p} Z^2_{(i)} \tag{2.3}$$

Comme ce paramètre σ_p^2 va apparaître dans le critère que l'on va essayer de minimiser, il faut donc en avoir une estimation *a priori*, que l'on obtiendra en prenant comme nombre de composantes non nulles, k_p, le $\hat{k_p}$ obtenu avec une autre des méthodes proposées dans ce chapitre, de préférence une méthode pas trop conservative pour éviter de sur-estimer la variance. Si on sur-estime trop la variance, on perdra beaucoup en puissance. On estimera donc $\underline{\hat{\mu}}$ et $\hat{I}$ par minimisation de critère de log-vraisemblance pénalisée :

$$L_{pen}(\underline{Z}, \underline{\hat{\mu}}_J, \hat{\sigma}^2_{I_{k_p}}, k_J) = -L(\underline{Z}, \underline{\hat{\mu}}_J, \hat{\sigma}^2_{I_{k_p}}) + Pen(k_J)$$

où L est la log-vraisemblance et Pen la pénalité. On a :

$$L(\underline{Z}, \underline{\hat{\mu}}_J, \hat{\sigma}^2_{I_{k_p}}) = -\frac{||\underline{Z} - \underline{\hat{\mu}}_J||_2^2}{2 \times \hat{\sigma}^2_{I_{k_p}}}$$

Il reste à choisir la forme de la pénalité. Si on prenait une pénalité l_r, c'est-à-dire de la forme $\lambda||\underline{\mu}||_p$, on retrouverait des estimateurs familiers. Avec $p = 2$, on trouve $\hat{\mu}_i = (1 + \lambda)^{-1} t_i$, ce qui correspond à un estimateur à rétrécissement linéaire. Pour $p = 1$, on obtient $\hat{\mu}_i = sgn(t_i)(|t_i - \lambda/2)_+$, ce qui correspond à un seuillage doux. Enfin pour p=0, si on considère $||\mu||_0 = \#\{i; \mu_i \neq 0\}$, on a $\hat{\mu}_i = t_i \mathbb{1}_{|t_i| \geq \lambda}$, ce qui correspond à un seuillage dur (pour les notions de seuillage dur et de seuillage doux on poura se reporter à la section suivante 2.4.1.2). Avec ce genre de pénalité, on voit bien que le problème du choix du paramètre λ est déterminant.

Nous allons nous intéresser à d'autres formes de pénalité proposées dans la littérature. L'idée générale qui va servir de fil conducteur à la mise en place de cette pénalité est de tenir compte de la "complexité" du modèle. Par complexité, on entend ici niveau de densité du vecteur considéré. Ainsi, de façon générale, on aura une pénalité dépendante de la complexité du modèle retenue. Ce critère revient alors à chercher un compromis satisfaisant entre un critère de complexité du modèle et un critère de log-vraisemblance.

2.3.2 Pénalité proposée par Abramovich *et al.*

Il faut maintenant définir la fonction de pénalité à choisir. Ce choix est discuté par Abramovich *et al.* dans [3]. Leur proposition consiste à minimiser le critère :

$$||\underline{Z} - \underline{\hat{\mu}}_J||_2^2 + \sum_{j=1}^{k_J} w_j^2 \tag{2.4}$$

avec

$$w_k = \sigma_p w(\frac{q}{2}\frac{k}{p})$$

où w correspond à la fonction des quantiles gaussiens à droite, c'est-à-dire $w(\cdot) = \Phi^{-1}(1-\cdot)$ et q correspond à un seuil de contrôle du FDR. Si on se ramène aux notations précédentes, en utilisant l'estimateur $\hat{\sigma}_{I_{k_p}}$ de σ_p, on a donc comme pénalité Pen_1 :

$$Pen_1(k_J) = \frac{\sum_{j=1}^{k_J} w_j^2}{2\hat{\sigma}^2_{I_{k_p}}}$$

L'originalité de leurs travaux réside dans le lien qu'ils parviennent à établir entre le paramètre q de contrôle du FDR et l'atteinte ou pas d'un minimax asymptotique pour le critère qu'ils proposent. Ils prouvent que si on prend $q = q_p \to 0$ quand $n \to +\infty$, c'est-à-dire imposer un FDR asymptotiquement négligeable, permet d'attenidre la minimaxité asymptotique. Ils montrent également que prendre $q_p \to q > 1/2$ empêche la minimaxité de leur procédure. Et leurs résultats ne leur ont pas permis d'acquérir de certitudes sur ce qui se passe quand $q_p \to q \in]0, 1/2]$.

Voyons en quoi leur procédure permet de contrôler le FDR, car le lien n'est pas évident. Pour cela, nous appréhendons le contrôle du FDR d'une façon différente que celle évoquée jusqu'à présent dans ce manuscrit.

2.3.2.1 Une autre façon de voir le FDR

Si le FDR a été introduit pour la première fois par Yoav Benjamini et Yosef Hochberg [8], avec une proposition de procédure Step-Up pour le contrôler, dans le contexte de l'estimation, une procédure de seuillage qui réflete cette procédure Step-Up a d'abord été proposée par Abramovich et Benjamini [2]. Elle est présentée à partir des valeurs observées des variables et non des p-valeurs. C'est pourquoi nous la redétaillons ici pour mettre en évidence le lien avec la sélection de modèle pénalisée.

Soient $z_1, z_2, \dots z_p$ les valeurs observées de la statistique de test. On considère les observations liées à la statistique de rang associée $(z_{(i)})_{i=1\dots p}$:

$$|z|_{(1)} \geq |z|_{(2)} \geq \dots \geq |z|_{(k)} \geq \dots \geq |z|_{(p)}$$

et on les compare à la suite des quantiles à droite Gaussien $(w_k)_{k=1\dots p}$ définie précédemment. On définit k_{FDR} par

$$k_{FDR} = \max\{k \mid |z|_{(k)} \geq w_k\}$$

On seuille alors les données au seuil $\hat{w}_{k_{FDR}} = \hat{w}_F$ et on a l'estimateur $\hat{\mu}_F$ suivant

$$\underline{\hat{\mu}}_{F,k} = \begin{cases} z_k \;\; si \; |z_k| \geq \hat{w}_F \\ 0 \qquad\qquad\qquad sinon \end{cases}$$

Il est facile de montrer que si les Z_i suivent une loi gaussienne centrée sous l'hypothèse nulle alors les deux procédures, celle de Abramovich et Benjamini et celle et Benjamini et Hochberg, sont en fait très proches.

2.3.2.2 Lien entre la sélection de modèle pénalisée et le FDR

On cherche à minimiser le critère

$$||\underline{Z} - \underline{\hat{\mu}}_J||_2^2 + \sum_{j=1}^{k_J} w_j^2$$

Ce qui, par définition de $\underline{\hat{\mu}}_J$, est aussi égal à

$$\sum_{j=k_J+1}^{p} z_{(j)}^2 + \sum_{j=1}^{k_J} w_l^2$$

Si k_J est petit, c'est-à-dire si le vecteur $\underline{Z}$ est très creux, la pénalité est équivalente à $k_J w_{k_J}^2$. Cela fait alors penser à une pénalité l_0 mais avec un paramètre de régularisation λ remplacé par le quantile gaussien au carré correspondant à la complexité k_J du modèle adopté pour $\underline{\mu}$. Notons $\underline{\hat{\mu}}_2$ l'argmin du critère 2.4. $\underline{\hat{\mu}}_2$ est calculé par une règle de seuillage dur. En effet, si $\hat{k}_2$ minimise

$$S_k = \sum_{j=k+1}^{p} z_{(j)}^2 + \sum_{j=1}^{k} w_l^2$$

alors on a

$$\underline{\hat{\mu}}_2 = z_i \mathbb{1}_{|z_i| \geq w_{\hat{k}_2}}$$

C'est ainsi qu'on retrouve le lien avec la procédure FDR telle qu'elle est présentée dans la section 2.3.2.1 avec cependant une différence ; $\hat{k}_2$ est l'emplacement du minimum global de S_k alors qu'il est facile de montrer que l'indice $\hat{k}_F$ obtenu à la section 2.3.2.1 est l'emplacement du minimum local "le plus à droite" de S_k. De façon similaire, on peut définir $\hat{k}_G$ comme le minimum local de S_k "le plus à gauche" De façon évidente, on a

$$\hat{k}_G \leq \hat{k}_2 \leq \hat{k}_F$$

Ces notions de maximum local à droite ou à gauche pour une évaluation de l'indice $\hat{k}$ correspondent en fait aux deux familles de procédures proposées pour contrôler le FDR : procédures "Step-up" et procédures "Step-down".

Abramovich *et al.* ont pu vérifier lors d'expérimentations numériques qu'en pratique ces trois indices sont souvent confondus. Dans un but théorique, ils ont montré

que $\hat{k}_F - \hat{k}_G$ est uniformément assez petit sur les espaces I pour que les résultats de minimaxité obtenus sur $\hat{\mu}_2$ puissent être étendus à $\hat{\mu}_F$.

On a vu que si k_J est petit, la pénalité est équivalente à $k_J w_{k_J}^2$. Abramovich *et al.* montrent aussi que

$$\text{si } k = o(p), \quad w_k^2 \sim 2{\sigma_p}^2 log(\frac{2p}{q_p k}).$$

Dans le cadre des données de biopuces, on s'attend à avoir une faible proportion de gènes différentiellement exprimés ; on peut alors songer à simplifier la pénalité et à minimiser le critère

$$||\underline{Z} - \underline{\hat{\mu}}_J||_2^2 + 2k_J \ \hat{\sigma}_{I_p}^2 \ log(\frac{2p}{q_p k_J}).$$

Avec nos notations, cela signifie que l'on considère la pénalité Pen_2

$$Pen_2(k_J) = k_J \ log\left(\frac{2p}{q_p k_J}\right)$$

Cette pénalité est croissante en k_J c'est-à-dire que la pénalité augmente avec la complexité du modèle. Dans les simultions, on utilisera les pénalités Pen_1 et Pen_2 pour pouvoir comparer les résultats.

2.3.3 Méthode proposée par Golubev.

Golubev s'est aussi intéressé à ce problème de reconstruction d'un vecteur creux avec un bruit gaussien dans l'article [22]. Il propose également un critère de vraisemblance pénalisée. Il se place dans le cas où l'écart-type du bruit est connu et égal à 1. Mais il est assez facile d'adapter les résultats avec une variance différente de 1 (on estimera la variance de la même façon que dans la section précédente).

Dans un premier temps, il propose d'utiliser une pénalité "conservative", de la forme

$$Pen_3(k_J) = 2k_J log\frac{p}{k_J} + k_J log\left(e \ log\frac{ep}{k_J}\right)$$

où e n'est autre que la valeur de la fonction exponentielle en 1. C'est-à-dire choisir $\hat{k}$ de la façon suivante

$$\hat{k} = arg\min_k \left\{ \sum_{j=k+1}^{p} z_{(j)}^2 + 2\hat{\sigma}_{I_{k_p}}^2 \left(2k \ log\frac{p}{k} + k \ log\left(e \ log\frac{ep}{k}\right)\right) \right\}$$

Il montre dans son théorème 2 qu'avec une telle pénalité, on peut majorer l'erreur moyenne au carrée de façon assez satisfaisante mais sans toutefois obtenir un estimateur asymptotiquement minimax.

On peut apporter une modification assez simple à cette méthode de sélection de modèle pour établir une méthode de substitution avec pénalité conservative. Cela consiste à considérer la famille d'estimateurs :

$$\widehat{\mu}(k) = z_i \mathbb{1}_{\{|z_i|>v_k\}} , \quad v_k = \hat{\sigma}_{I_{k_p}} \sqrt{2log(p/k)}, \quad k = 1 \dots p$$

Donc pour un k connu, on aura l'estimateur correspondant du vecteur $\underline{\mu}$. Il reste à estimer le $\widehat{k}$ optimal. Pour cela on calcule :

$$\hat{k} = arg \min_k \left\{ ||\underline{z} - \widehat{\mu}(\underline{z}, k)||^2 + 2\hat{\sigma}^2_{I_{k_p}} \left(2k \; log\frac{p}{k} + k \; log\left(e \; log\frac{ep}{k} \right) \right) \right\}$$

On prend alors $\widehat{\mu} = \widehat{\mu}(\widehat{k})$. Ce seuil v_k qui apparaît ici n'est pas anodin. En effet, dans les problèmes où la proportion de composantes est connue, c'est-à-dire, si on a $p_1 = p^\beta$ composantes non nulles, on peut prendre un seuil $t_\beta = \sqrt{2(1-\beta)\log p} = \sqrt{2\log(p/p_1)}$. Ainsi, le seuil v_k représente le seuil à utiliser si on a effectivement k composantes non nulles. Ici, la méthode consiste donc à commencer par estimer ce nombre de composantes non nulles par $\widehat{k}$ et ensuite, d'utiliser ce $\widehat{k}$ pour trouver l'estimateur $\widehat{\mu}$.

Golubev prouve que l'estimateur obtenu par cette méthode de substitution avec pénalité conservative est asymptotiquement minimax.

2.3.4 Exemple d'application de ces deux méthodes

On reprend l'exemple utilisé en section 2.2.3 pour appliquer les méthodes de sélection de modèles pénalisées proposées. Ici, on ne fera toutefois que 1 000 simulations, le temps de calcul étant plus important. Il est à noter que dans nos simulations, on a une variance σ_p^2 connue et égale à 1. En pratique, la plupart du temps, on ne connaît pas cette variance. Ici, comme on est censé avoir simulé une statistique de test dont on connaît la loi sous hypothèse nulle, on supposera, comme on l'a fait dans l'application de la sélection de modèles non pénalisée, qu'on a σ_p connu égal à 1. Les résultats sont donnés tableau 2.3 pour la pénalisation proposée par Abramovich et al. et tableau 2.4 pour la pénalisation proposée par Golubev.

Pour les résultats obtenus avec la pénalité proposée par Abramovich *et al.* , on observe des résultats tout à fait conformes à ce qu'on pouvait espérer, du moins avec la première pénalité. On pourra tout de suite oublier la deuxième pénalité, qui était censée être une approximation de la première, mais qui est beaucoup trop conservative. On remarque que la méthode s'adapte très bien aux différentes proportions de composantes non nulles des vecteurs simulés pour rester à un FDR observé de l'ordre du seuil q imposé à chaque jeu de simulation. En ce qui concerne les approches proposées par Golubev, la pénalité conservative porte bien son nom. La puissance demeure très faible, alors que pour le premier jeu de simulation par exemple on a un gros écart entre la moyenne des composantes non nulles (5.21 en valeur absolue) et zéro. La méthode par substitution améliore les résultats. Ici on ne tient pas compte d'un seuil autorisé, il est donc difficile de comparer les performances de cette méthode avec la méthode d'Abramovich.

		Penalité 1		Pénalité 2	
p_1	q	FDR obs.	Puissance	FDR obs.	Puissance
10	0.01	0.0105	0.7684	0.0009	0.6466
	0.05	0.0513	0.8720	0.0115	0.7741
	0.10	0.1009	0.9070	0.0240	0.8242
100	0.01	0.0098	0.7107	0.0024	0.5675
	0.05	0.0504	0.8445	0.0127	0.7352
	0.10	0.0998	0.8900	0.0276	0.7992
1000	0.01	0.0089	0.3098	0.0023	0.1580
	0.05	0.0448	0.5505	0.0130	0.3613
	0.10	0.0898	0.6648	0.0282	0.4767

Table 2.3 – Résultats de simulation pour la sélection de modèles pénalisée par méthode d'Abramovich *et al.* , ici $\sigma_p = 1$ connu. On a 1 000 simulations et $p = 10\,000$ gènes. On fait varier le seuil $q \in \{0.01, 0.05, 0.1\}$. p_1 désigne le vrai nombre de termes de moyenne non nulle. Pour $p_1 = 10$, on a $\mu_1 = 5.21$, pour $p_1 = 100$, on a $\mu_1 = 4.52$ et pour $p_1 = 1\,000$, on a $\mu_1 = 3.11$. On donne le FDR moyen et la puissance moyenne observés sur les 10 000 simulations. La pénalité 1 est celle proposée par les auteurs, la pénalité 2 en est une approximation.

	Pen. Conservative		Méth. de Substitution	
p_1	FDR obs.	Puissance	FDR obs.	Puissance
10	0	0.2444	0.0190	0.8085
100	0.0003	0.3346	0.0462	0.5865
1000	0.0001	0.0143	0.0016	0.1218

Table 2.4 – Résultats de simulation pour la sélection de modèles pénalisée par méthode de Golubev, ici $\sigma_p = 1$ connu. On a 1 000 simulations et $p = 10\,000$ gènes. p_1 désigne le vrai nombre de termes de moyenne non nulle. Pour $p_1 = 10$, on a $\mu_1 = 5.21$, pour $p_1 = 100$, on a $\mu_1 = 4.52$ et pour $p_1 = 1\,000$, on a $\mu_1 = 3.11$. On donne le FDR moyen et la puissance moyenne observés sur les 10 000 simulations. On étudie la pénalité conservative et la méthode par substitution.

FDR-ondelettes et Ebayes threshold

2.4.1 La décomposition en ondelettes : une réponse à la dépendance des données

Les ondelettes sont apparues il y a quelques années et ont apporté un souffle nouveau au traitement du signal, notamment à celui des images. Les ondelettes permettent d'analyser et de repérer les discontinuités d'un signal à une ou deux dimensions à des échelles très différentes. Nous nous limiterons ici au cas d'un signal unidimensionnel puisque c'est celui qui nous intéressera par la suite. L'intérêt de la décomposition en ondelettes réside dans la représentation d'un signal observé par une superposition de fonctions « élémentaires » de façon adaptée. Pour obtenir ces représentations, des algorithmes rapides sont nécessaires. Une fois la décomposition faite, on aimerait pouvoir la simplifier de manière efficace en choisissant de façon appropriée seulement quelques composantes élémentaires. On peut voir cela comme une approximation ou une compression.

2.4.1.1 Décomposition de signaux en ondelettes

La transformée en ondelettes

La transformée en ondelettes est un outil qui décompose les données, les fonctions ou les opérateurs en composantes fréquentielles suivant une résolution adaptée à l'échelle. La transformée en ondelettes décompose le signal sur une famille d'ondelettes translatées et dilatées.

Une ondelette est une fonction ψ de $L^2(\mathbb{R})$ de moyenne nulle : $\int_{-\infty}^{+\infty} \psi(t)dt = 0$. Elle est normalisée : $\int |\psi(t)|^2 dt = 1$ et centrée au voisinage de $t = 0$.

Une famille d'éléments temps-fréquence s'obtient en dilatant d'un facteur a et en translatant d'un facteur b l'ondelette : $\psi_{a,b}(t) = \frac{1}{\sqrt{(a)}}\psi(\frac{t-b}{a})$, ces facteurs restent normalisés et on appelle transformée en ondelettes de $f \in L^2(\mathbb{R})$ au temps b et à l'échelle a :

$$W f_{a,b} = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{(a)}}\ \psi(\frac{t-b}{a})dt$$

Plus a est petit, moins l'ondelette est étendue temporellement.

Bases d'ondelettes orthogonales - Transformée dyadique

On agit maintenant sur les paramètres a et b pour analyser le signal. Dans le cadre d'une analyse multirésolution, on choisit a et b comme suit : $a = 2^j$ et $b = k2^j$ où j est le niveau de résolution. Ainsi les nouveaux paramètres sont k et j et on peut construire des ondelettes qui génèrent des bases orthonormées de $L^2(\mathbb{R})$ discrètes par :

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}\psi(\frac{t-2^j k}{2^j})}$$

Les ondelettes orthogonales dilatées de 2^j reproduisent les variations d'un signal à la résolution 2^{-j}. La construction de ces bases est ainsi liée à l'approximation multirésolution des signaux.

On définit alors les coefficients d'ondelette qui dépendent de j et de k :

$$C_{j,k} = \int_{-\infty}^{+\infty} f(t)\psi_{j,k}(t)dt$$

Ceci constitue une approche discrète des problèmes. C'est-à-dire que dans la réalité, l'espace fréquence-temps est parcouru de façon continue, donc lorsqu'on fait la transformation, les informations sont redondantes, il faut donc échantillonner l'espace : discrétiser. Les ondelettes sont toujours continues, mais les coefficients de la transformée sont dénombrables sur un intervalle de l'espace-temps.

Les approximations multirésolution calculent l'approximation d'un signal à diverses résolutions par projection orthogonale sur une famille d'espaces V_j appelés espaces d'approximation.

Une suite $\{V_j\}$ de sous-espaces fermés de $L^2(\mathbb{R})$ est une approximation multirésolution si elle vérifie pour tout f de cet espace :

$$\begin{cases} \forall (j,k) \in \mathbb{Z}^2, \quad f(t) \in V_j \Longleftrightarrow f(t-2^j k) \in V_j & (i) \\ \forall j \in \mathbb{Z}, \quad V_{j+1} \subset V_j & (ii) \\ \forall j \in \mathbb{Z}, \quad f(t) \in V_j \Longleftrightarrow f(\frac{t}{2}) \in V_{j+1} & (iii) \\ \lim\limits_{j \to +\infty} V_j = \bigcap\limits_{j=-\infty}^{+\infty} V_j = \{0\} & (iv) \\ \lim\limits_{j \to -\infty} V_j = \overline{\bigcup\limits_{j=-\infty}^{+\infty} V_j} = L^2(\mathbb{R}) & (v) \\ \exists\, \phi \in V_0 \text{ telle que } V_0 = \{f \in L^2(\mathbb{R}) : f(x) = \sum\limits_{k \in \mathbb{Z}} \alpha_k \phi(t-k)\} & (vi) \\ \{\phi(t-k)\}_{k \in \mathbb{Z}} \text{ est une base stable de } V_0 \end{cases}$$

La fonction ϕ est appelée la fonction d'échelle de l'analyse multirésolution.
Soit $\phi_{j,k}(t) = \frac{1}{\sqrt{2^j}}\phi(\frac{t-2^j k}{2^j})$. Alors $\{\phi_{j,k}\}_{k \in \mathbb{Z}}$ est une base de V_j pour tout j dans $\mathbb{Z}$.

La projection du signal sur les espaces V_j engendre des coefficients de décomposition. Ces coefficients vont représenter l'approximation du signal à une résolution donnée.

Pour les algorithmes de décomposition en ondelettes utilisés, il existe aussi la transformation inverse qui permet de reconstruire le signal d'origine à partir des coefficients d'ondelettes obtenus.

Donnons un exemple sur un signal c de taille $8 = 2^3$ Au départ, on a :

$$a = \begin{pmatrix} c_{3,0} \\ c_{3,1} \\ c_{3,2} \\ c_{3,3} \\ c_{3,4} \\ c_{3,5} \\ c_{3,6} \\ c_{3,7} \end{pmatrix}$$

Il s'agit du signal avec tous ses détails. Les étapes de décomposition sont :

$$\begin{pmatrix} c_{3,0} \\ c_{3,1} \\ c_{3,2} \\ c_{3,3} \\ c_{3,4} \\ c_{3,5} \\ c_{3,6} \\ c_{3,7} \end{pmatrix} \rightarrow \begin{pmatrix} c_{2,0} \\ c_{2,1} \\ c_{2,2} \\ c_{2,3} \\ \hline d_{2,0} \\ d_{2,1} \\ d_{2,2} \\ d_{2,3} \end{pmatrix} \rightarrow \begin{pmatrix} c_{1,0} \\ c_{1,1} \\ \hline d_{1,0} \\ d_{1,1} \\ d_{2,0} \\ d_{2,1} \\ d_{2,2} \\ d_{2,3} \end{pmatrix} \rightarrow \begin{pmatrix} c_{0,0} \\ \hline d_{0,1} \\ d_{1,0} \\ d_{1,1} \\ d_{2,0} \\ d_{2,1} \\ d_{2,2} \\ d_{2,3} \end{pmatrix}$$

L'algorithme de reconstruction permet de franchir les étapes dans l'autre sens.

2.4.1.2 Débruitage par seuillage des coefficients d'ondelettes

Nous avons vu qu'il était possible de réaliser une décomposition en ondelettes d'un signal puis de reconstruire ce signal à partir de ses coefficients d'ondelettes. Pourtant, cette technique n'aurait pas un grand intérêt si on ne modifiait pas ces coefficients, car alors le signal final serait en tout point identique au signal initial. L'idée c'est de garder les coefficients les plus caractéristiques ; on cherche donc à éliminer les détails les plus fins du signal. Cette technique de seuillage des coefficients d'ondelettes, initialement introduite par Donoho et Johnstone [14] [13], est très utilisée pour faire de la compression d'images, mais aussi, et c'est ce qui va nous intéresser ici, pour faire du débruitage de signal. Dans ce dernier cas, on ne garde que les coefficients les plus grands et on met les autres à zéro avant de reconstruire le signal. Le bruit correspond en général à des détails faibles donc il est éliminé par ce seuillage des coefficients d'ondelettes. On obtient alors un signal débruité.

En général, dans le cadre du débruitage, on suppose qu'on a des données de la forme :

$$Y_j = f(t_j) + \varepsilon_j, \qquad j = 1, \ldots, p$$

De façon usuelle, on a des t_j équidistants, p qui est égal à une puissance de deux ($p = 2^m$) et les ε_j sont indépendants et identiquement distribués de moyenne zéro et de variance σ^2. Cependant, en pratique, ces conditions peuvent ne pas être vérifiées, on peut avoir des t_j non équidistants, voire même aléatoires, un nombre de points qui n'est pas une puissance de deux et des bruits de variance non constante.
La démarche générale dans le cadre des méthodes non-linéaires est de faire la décomposition en ondelettes du signal, d'extraire les coefficients les plus significatifs par rétrécissement ou par seuillage et de débruiter en appliquant la transformée en ondelettes inverse sur les coefficients restants.

Seuillage dur ou « hard thresholding »

Le seuillage dur est celui qui est le plus « intuitif ». On se fixe un seuil T positif et on ne conserve que les coefficients d'ondelettes supérieurs à T, les autres sont mis à zéro. C'est-à-dire :

$$\theta(x) = \begin{cases} 0 & si \quad |x| \leq T \\ x & si \quad |x| > T \end{cases}$$

pour tout coefficient d'ondelettes x.

On a donc un seuillage des coefficients de la forme présentée figure 2.3 On es-

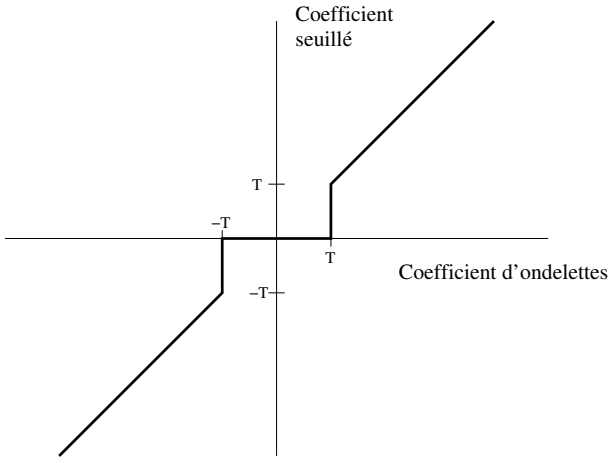

Figure 2.3 – Seuillage dur des coefficients d'ondelettes

timera ensuite le signal débruité par la transformation en ondelettes inverse des coefficients après seuillage.

Seuillage doux ou « soft thresholding »

Dans le cas du seuillage doux, on met toujours à zéro les coefficients inférieurs à un seuil T. Par contre, pour ceux supérieurs à T, on atténue l'amplitude des coefficients par la valeur du seuil afin de s'assurer d'avoir enlevé l'effet du bruit même pour les forts coefficients.

$$\theta(x) = \begin{cases} 0 & si \quad |x| \leq T \\ x - signe(x)T & si \quad |x| > T \end{cases}$$

Dans ce cas, la fonction de seuillage est continue comme on peut le voir figure 2.4.

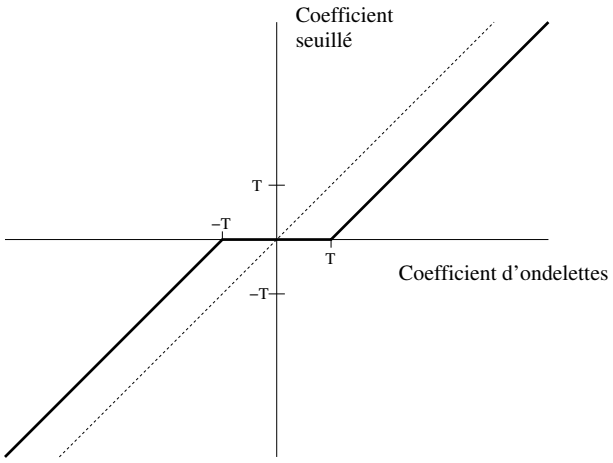

Figure 2.4 – Seuillage doux des coefficients d'ondelettes

Le coefficient seuillé sera donc plus petit que le coefficient du signal. Ce type de seuillage garantit que le signal obtenu sera toujours plus régulier que le signal de départ.

Propriétés et autre méthode

Quand on les compare, il semble que le seuillage dur donne des estimations avec une variance plus importante et le seuillage doux présente un biais plus important. C'est pourquoi une autre approche a été proposée, il s'agirait de combiner seuillage dur et doux pour bénéficier des avantages des deux méthodes :

$$\theta(x) = \begin{cases} 0 & si \quad |x| \leq T_1 \\ signe(x)\frac{T_2(|x|-T_1)}{T_2-T_1} & si \quad T_1 \leq |x| \leq T_2 \\ x & si \quad |x| > T_2 \end{cases}$$

Cette méthode donne la fonction de seuillage représentée figure 2.5.

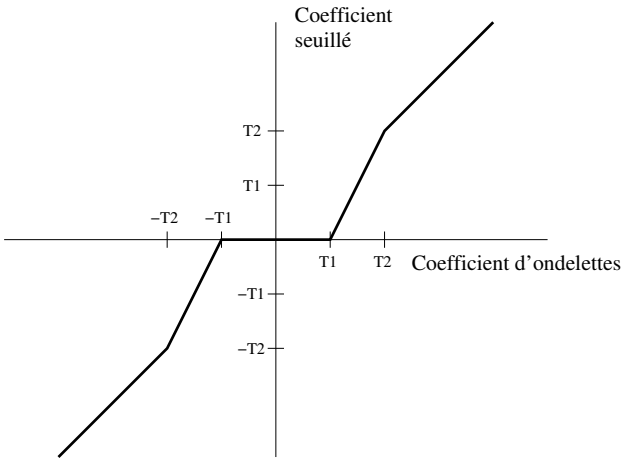

Figure 2.5 – Seuillage combiné des coefficients d'ondelettes

Quelque soit le type de seuillage choisi, le problème du choix du seuil T se pose, et il existe de nombreuses façons de fixer ce seuil. Pour les seuillages dur et doux, le choix le plus répandu consiste à utiliser le seuil "universel" proposé par Donoho et Johnstone [14], seuil pour lequel ils ont montré que le risque était proche de la borne inférieure. Ce seuil est de la forme :

$$T = s_d\sqrt{2log(n)}$$

où s_d une estimation de l'écart-type des coefficients d'ondelettes.

2.4.1.3 Application aux données de biopuces

En quoi les ondelettes peuvent-elles nous être utiles dans le cadre du traitement des données issues des biopuces à ADN ? En fait, leur réel atout réside dans leur propriété de décorréler les données. En effet, en biologie, on sait que les gènes sont dépendants entre eux, des réseaux de régulation souvent complexes les lient. Ainsi, en observant les niveaux d'expression des gènes via une expérience de biopuces, il paraît absolument déraisonnable de supposer que les niveaux d'expression sont indépendants entre les gènes. C'est pourtant bien souvent l'hypothèse qui est faite en pratique même si ce n'est pas toujours énoncé clairement. Pourquoi cette hypothèse ? Tout simplement parce que les outils statistiques les plus puissants ne sont souvent valables que dans les cas d'indépendance. Ainsi, le FDR initialement développé par Benjamini et Hochberg [8] l'était pour des données supposées indépendantes. Il a pu être étendu au cadre de données ayant une certaine structure de dépendance (Benjamini et Yekutieli [9]). Pourtant dans le cadre des biopuces, la structure de dépendance des données reste une inconnue. Ainsi l'idée est d'appliquer une décomposition en ondelettes à la statistique de test afin de décorréler les

données. Le principal problème qui se pose c'est que la décomposition en ondelettes a été développée pour faire du traitement de signal. Or, on ne peut pas vraiment parler de "signal" dans le cadre des données de biopuces. En effet, un signal suppose un ordre dans la disposition des données, on a par exemple des observations ordonnées selon une échelle temporelle. Or ici, on dispose d'un vecteur indexé par l'identifiant du gène, ces identifiants ne constituent pas un ensemble ordonné. Il faudra donc que les résultats obtenus soit invariants par permutation des indices des gènes.

Montrons cette décorrélation sur un exemple : on prend la statistique de t-test obtenue par comparaison entre le jeux de données normalisés des faibles et des forte doses de l'Institut Curie qui devrait nous permettre de détecter les gènes différentiellement exprimés entre ces deux conditions expérimentales. Sur la figure 2.6, on constate tout d'abord la corrélation initiale des données.
Sur les figures 2.7 et 2.8, on répresente désormais l'autocorrélation des coefficients

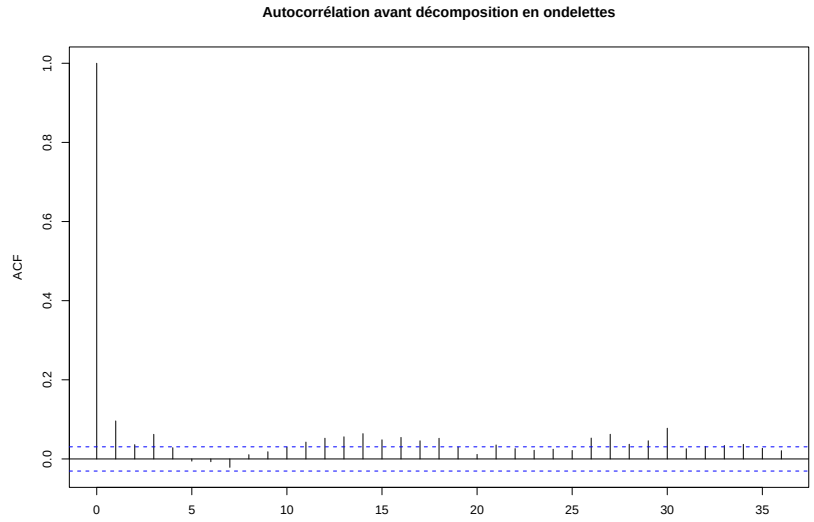

Figure 2.6 – Autocorrélation de la statistique de test avant décomposition en ondelettes. Il n'y aurait pas de corrélation si les traits verticaux (sauf le premier) ne dépassaient pas des limites horizontales. Ici, on voit bien qu'il y a dépendance entre les données.

d'ondelettes calculés sur la statistique de test à différentes échelles. On remarque une nette décorrélation des données.

La procédure proposée est présentée dans l'algorithme 2.3 Cette notion de procédure FDR « améliorée » (EFDR) est inspirée de Shen *et al.* dans [42]. L'idée, c'est d'augmenter la puissance de la procédure FDR. Rappelons la procédure FDR définie dans [8] :

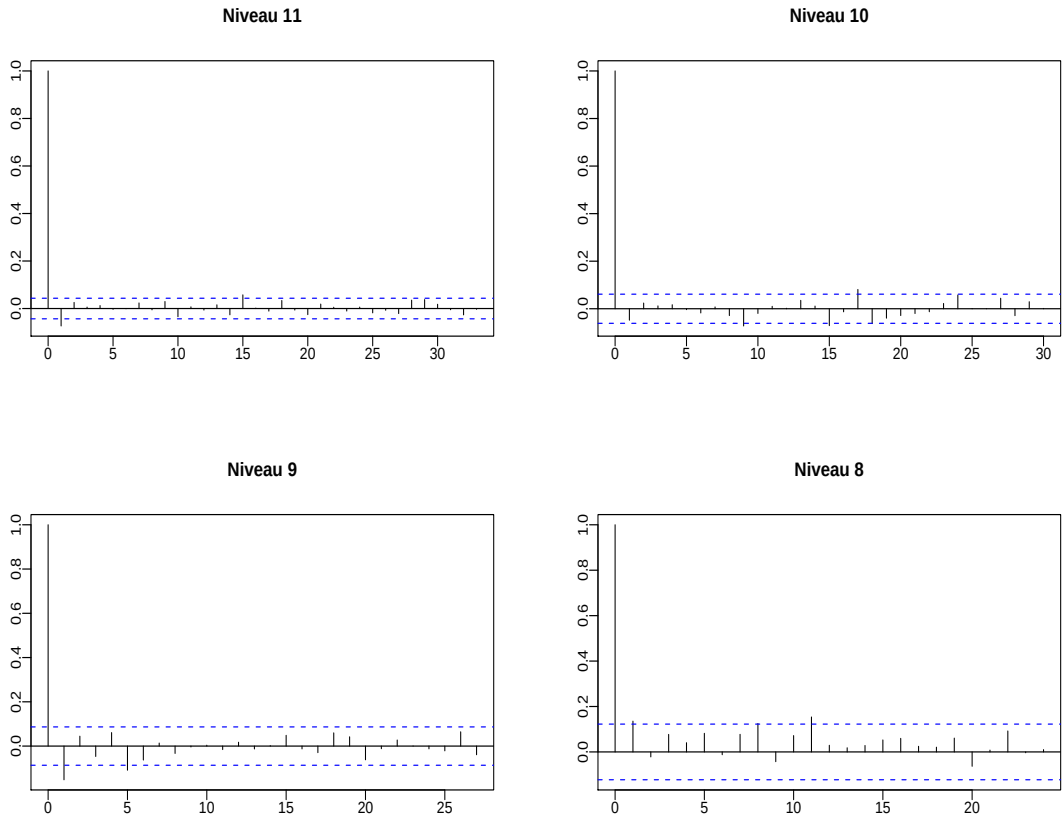

Figure 2.7 – Autocorrélation de la statistique de test après décomposition en ondelettes à différentes échelles (niveaux de détails 11 à 8). Les données commencent à se décorréler.

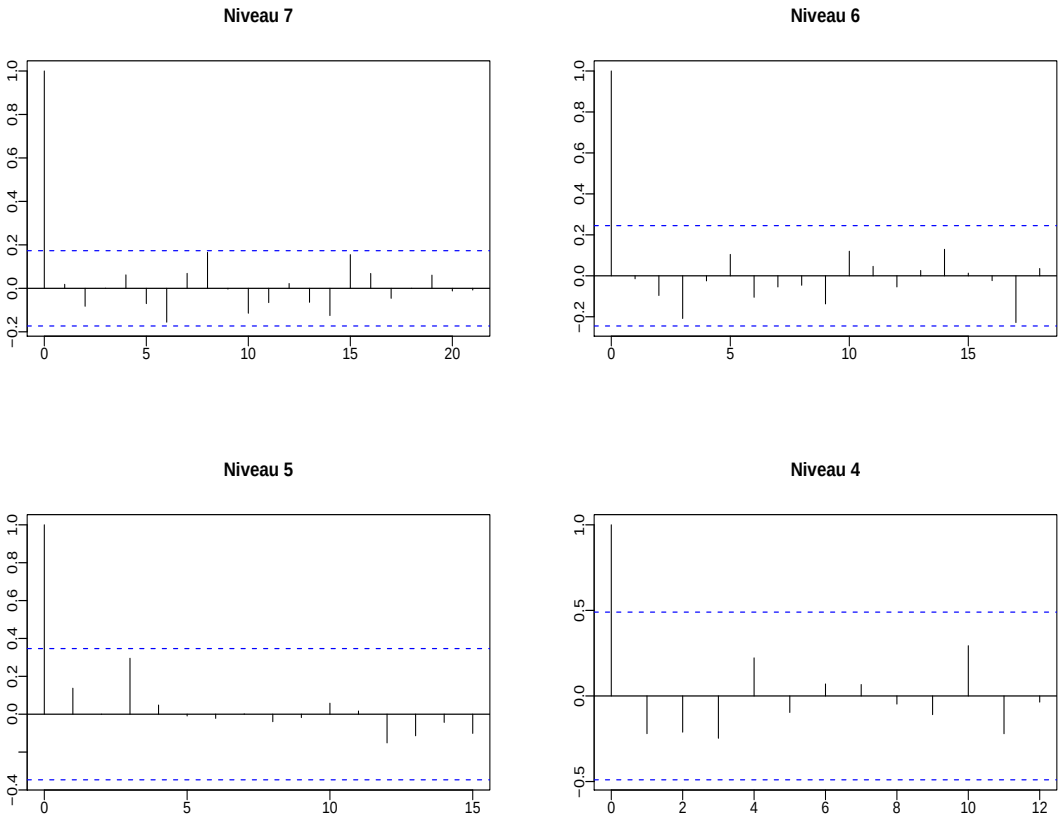

Figure 2.8 – Autocorrélation de la statistique de test après décomposition en ondelettes à différentes échelles (niveaux de détails 7 à 4). Dans ces niveaux où on garde peu de détails, on voit que les données sont bien décorrélées.

Algorithme 2.3. Procédure EFDR (Enhanced FDR)

Etape 1 - Calculer une statistique de test Z_i, $i = 1, \ldots, p$, où p désigne le nombre de gènes.
Etape 2 - Faire une décomposition en ondelettes sur ce signal pour décorréler.
Etape 3 - Faire une sélection optimale de p^* coefficients d'ondelettes via une méthode de seuillage.
Etape 4 - Appliquer une méthode FDR sur les p^* statistiques de test restantes.

1. Ranger les p-valeurs par ordre croissant $\pi_{(1)} \leq \ldots \leq \pi_{(p)}$
2. Calculer $k = \max\{i \mid \pi_{(i)} \leq \frac{i}{p}q\}$
3. Rejeter $H_{0(j)} : \beta_{(j)} = 0\ ,\ \forall j \in \{1, \ldots, k\}$ où $H_{0(j)}$ est l'hypothèse nulle associée à la p-valeur ordonnée $\pi_{(j)}$. Si un tel k n'existe pas, on ne rejette aucune hypothèse.

Nous avons vu que le théorème de Benjamini-Hochberg [8] permettait d'affirmer $FDR = \mathbb{E}(Q) \leq \frac{p_0}{p}q$, où p_0 désigne le vrai nombre d'hypothèses nulles et p le nombre total d'hypothèses. On contrôle donc le FDR au taux de $\frac{p_0}{p}q$ alors que le seuil fixé au départ était de q. Le test est alors plus conservatif que nécessaire. L'idée c'est de se rapprocher de cette valeur seuil de q, c'est-à-dire de détecter plus de gènes tout en conservant un taux d'erreur inférieur à q. Ici, nous allons donc appliquer la même procédure mais sur p^* hypothèses au lieu de p. Ainsi la statistique de test sera significative si $p_{(i)} \leq \frac{\alpha \cdot i}{p^*}$ au lieu de $p_{(i)} \leq \frac{\alpha \cdot i}{p}$, avec de plus $p^* < p$. On détectera donc plus d'hypothèses, et on aura comme FDR :

$$FDR = \mathbb{E}(Q) \leq \frac{p_0}{p^*}q$$

ce qui permet donc au FDR de se rapprocher d'avantage du seuil q et donc à la procédure de gagner en puissance.

Le problème qui se pose est celui de la puissance de deux. En effet, pour faire une décomposition en ondelettes, il faut avoir un signal de taille $p = 2^m$ avec m entier. La solution que nous avons choisie est de tronquer nos données, c'est-à-dire par exemple si on dispose de 10 000 gènes, nous n'allons garder que les $2^1 3 = 8\,192$ qui ont la plus grande statistique de test en valeur absolue. Cela pose toutefois un problème... cela change les résultats obtenus. Pour comparer les résultats avec ceux obtenus sans seuillage préalable des coefficients d'ondelettes, il faudra donc faire comme si la statistique de test totale observée était la préselection qu'on en a faite. C'est ce qu'on fera pour pouvoir étudier l'efficacité de la méthode tout en gardant à l'esprit que cette méthode ne sera en fait adaptée que si on dispose de biopuces avec un nombre total de gènes très proche d'un puissance de 2.

2.4.2 Méthodes empiriques bayésiennes de seuillage (Ebayes Threshold)

Il s'agit d'une méthode de seuillage empirique bayésien développée par Johnstone et Silverman (*cf* [27] et [26]) pour débruiter un signal considéré comme creux, en s'adaptant au niveau de densité de ce signal.

2.4.2.1 Contexte

On part d'un problème statistique classique : on s'intéresse à une séquence de paramètres μ_i pour chacun desquels on ne dispose que d'une observation bruitée X_i,

avec :

$$X_i = \mu_i + \varepsilon_i \qquad i \in 1, \ldots, p$$

où les ε_i suivent une loi normale centrée réduite.

Il est évident que, sans aucune information *a priori* sur les μ_i, on ne sera pas en mesure de les estimer très efficacement. La méthode proposée dans [27] exploite un possible caractère creux des données. Le seuillage constitue alors une approche naturelle de ce problème : pour un i donné, si la valeur absolue de X_i dépasse un certain seuil t alors on considère que cela correspond à un μ_i non nul, par contre, si $|X_i|$ est inférieure au seuil t, alors le coefficient $|\mu_i|$ est estimé comme étant nul.

$$\begin{cases} \hat{\mu} = 0 & si \quad |X_i| \leq t \\ \hat{\mu} \neq 0 & si \quad |X_i| > t \end{cases}$$

La qualité de l'estimation est assez sensible au choix du seuil, le meilleur choix dépendant des données du problème de départ. De façon générale, on choisira un seuil relativement élevé pour les signaux « creux » et un seuil plus faible pour les signaux plus « denses ». On peut espérer que les méthodes proposées estiment des seuils qui réflètent de façon stable la gradation des signaux creux aux signaux denses. Les articles de Johnstone et Silverman [27] et [26] montrent, par des considérations pratiques et théoriques, qu'une approche empirique bayésienne adaptée présente de bonnes propriétés.

On peut tout à fait adapter ce cadre de travail à la détection des gènes différentiellement exprimés. En effet, on cherche à différentier ce qu'on considèrera comme du bruit (gènes non exprimés) de ce qui est significatif (les gènes exprimés), en faisant par exemple un seuillage à partir de la statistique de test. On a un signal qui est supposé être réellement creux, c'est-à-dire que la plupart des gènes ne sont pas différentiellement exprimés et donc la plupart des μ_i sont nuls. Ainsi trouver les μ_i non nuls par une méthode empirique bayésienne de seuillage nous donnera la liste des gènes différentiellement exprimés.

2.4.2.2 Approches Bayésiennes

Dans un contexte Bayésien, la notion de signal creux est modélisée par une « distribution *a priori* » adaptée aux paramètres μ_i. On modélise les μ_i comme ayant des distributions *a priori* indépendantes données par le mélange :

$$f_{prior}(\mu) = (1 - w)\delta_0(\mu) + w\gamma(\mu)$$

δ_0 désigne le dirac en zéro, c'est-à-dire la distribution de la proportion $(1 - w)$ des μ_i qui sont nuls. On suppose que la partie non nulle de la distribution *a priori*, γ, est une densité unimodale symétrique fixée. Il y a plusieurs possibilités pour le choix de cette fonction notamment les distributions Laplace ou de quasi-Cauchy pour lesquelles les procédures peuvent être entièrement effectuées informatiquement. La loi de quasi-Cauchy est définie par la densité

$$\gamma(u) = (2\pi)^{-1/2} \left\{ 1 - |u| \frac{1 - \Phi(|u|)}{\phi(u)} \right\}$$

où Φ et ϕ sont respectivement la fonction de répartition et la densité de la loi normale centrée réduite. La loi de Laplace est quant à elle définie par la densité :

$$\gamma_a(u) = \frac{a}{2} exp(-a|u|)$$

où $a > 0$ est un paramètre d'échelle.

On suppose que μ suit la loi *a priori* de densité f_{prior} et que $X \sim \mathcal{N}(\mu, 1)$. On peut alors trouver la loi *a posteriori* de μ conditionnellement à $X = x$.
Soit $\hat{\mu}(x; w)$ la médiane de cette distribution *a posteriori* ; pour tout w fixé, $\hat{\mu}(x; w)$ sera une fonction monotone de x avec cette propriété de seuillage : il existe $t(w) > 0$ tel que $\hat{\mu}(x; w) = 0$ si et seulement si $|x| \leq t(w)$. C'est-à-dire qu'on peut ainsi définir une règle de seuillage liée à $\hat{\mu}(x; w)$ puisque :

$$\exists t(w) \ \ tel\ que\ :\ \hat{\mu}(x; w) = 0 \Longleftrightarrow |x| \leq t(w)$$

Une fois w fixé, il y a d'autres possibilités de choix de règles d'estimation, par exemple la moyenne *a posteriori*, $\tilde{\mu}(x; w)$, de μ pour $X = x$ donné ou bien un seuillage « dur » ou « doux » avec le seuil $t(w)$.
Pour une série d'observations, on peut appliquer la procédure Bayésienne séparément pour chaque observation X_i pour obtenir un estimateur du paramètre μ_i correspondant. C'est une procédure Bayésienne exacte si les X_i sont indépendants ; si les X_i ne sont pas complètements indépendants, on a une perte d'information dans la procédure d'estimation mais, si la dépendance reste faible, alors la méthode devrait donner des résultats raisonnables.

Le choix du poids de mélange w est très important. Si on suppose que les X_i sont indépendants, on peut alors estimer w en maximisant la vraisemblance.
Grâce à la densité *a priori* de μ et en utilisant l'hypothèse $X \sim \mathcal{N}(\mu, 1)$, c'est-à-dire $\varepsilon \sim \mathcal{N}(0, 1)$, on peut déterminer f_X, la densité marginale des observations X_i. En effet :

$$\left.\begin{array}{l} \mu \sim f_{prior} \\ \varepsilon \sim \phi \end{array}\right) \Longrightarrow X = \mu + \varepsilon \sim f_X = f_{prior} * \phi$$

où $*$ désigne le produit de convolution et ϕ la densité gaussienne standardisée.
On a donc :

$$f_X = (1 - w)\delta_0(\mu) * \phi + w\gamma(\mu) * \phi$$

De plus, $\delta_0(\mu) * \phi = \phi$ car la masse de Dirac en 0 est un élément neutre pour le produit de convolution.
On note $g = \gamma * \phi$, on a alors :

$$f_X(x) = (1 - w)\phi(x) + wg$$

On peut alors définir l'estimateur de maximum de vraisemblance marginale $\hat{w}$ de w comme étant la valeur qui maximise la log-vraisemblance marginale :

$$l(w) = \sum_{i=1}^{p} log\{(1 - w)\phi(X_i) + wg(X_i)\}$$

soumis à la contrainte sur w d'avoir un seuil qui satisfait $t(w) \leq \sqrt{2log(p)}$.
Pour les densités *a priori* que nous envisageons, $l'(w)$ est une fonction monotone, il est donc aisé de trouvé sa racine numériquement à condition que la fonction g s'y prête bien.
L'approche de base adoptée par Johnstone et Silverman est une approche Bayésienne Empirique. Il s'agit d'utiliser d'abord le jeu de données une fois pour calculer un estimateur $\hat{w}$ de w en maximisant la vraisemblance marginale. On injecte alors cette valeur $\hat{w}$ dans la densité *a priori* et on estime alors les paramètres μ_i par une procédure Bayésienne.
Maintenant que nous avons décrit l'esprit de la démarche, détaillons un petit peu les calculs. Tout d'abord, nous avons dit qu'une fois w estimé, la loi *a posteriori* de μ conditionnellement à $X = x$ pouvait être connue et ainsi aussi la médiane et la moyenne *a posteriori*.

- Probabilité *a posteriori* que le paramètre soit non nul
  Définissons w_{post} la probabilité *a posteriori* d'avoir un paramètre μ non nul conditionnellement à $X = x$.

$$w_{post} = \mathbb{P}(\mu \neq 0 | X = x) = \frac{wg(x)}{wg(x) + (1-w)\phi(x)} = \frac{1 + \beta(x)}{w^{-1} + \beta(x)}$$

  avec $\beta(x) = \dfrac{g(x)}{\phi(x)} - 1$.
  Ainsi il suffira d'être en mesure de calculer β pour avoir w_{post}.

- Moyenne *a posteriori*
  On définit f_1 :

$$f_1(\mu | X = x) = f(\mu | X = x, \mu \neq 0)$$

  alors la densité *a posteriori* f_{post} est :

$$f_{post}(\mu | X = x) = (1 - w_{post})\delta_0(\mu) + w_{post} f_1(\mu | x)$$

  Soit $\mu_1(x)$ la moyenne de la densité $f_1(\cdot|x)$. La moyenne *a posteriori* $\tilde{\mu}(x; w)$ est alors égale à $w_{post}\mu_1(x)$.

- Médiane *a posteriori*
  On cherche la médiane *a posteriori* $\hat{\mu}(x; w)$ de μ conditionnellement à $X = x$.
  Soit :

$$\tilde{F}_1(\mu | x) = \int_{\mu}^{\infty} f_1(u|x) du$$

  Si $x > 0$, on peut trouver $\hat{\mu}(x; w)$ grâce aux propriétés :

$$\begin{cases} \hat{\mu}(x; w) & = & 0 & \text{si } w_{post}\tilde{F}_1(0|x) \leq \frac{1}{2} \\ \tilde{F}_1(\hat{\mu}(x; w)|x) & = & \{2w_{post}(x)\}^{-1} & \text{sinon} \end{cases}$$

Remarquons que si $w_{post}(x) \leq \frac{1}{2}$, alors la médiane est forcément nulle, sans que d'autres calculs soient nécessaires.
Si $x < 0$, on utilise la propriété d'antisymétrie : $\hat{\mu}(-x; w) = -\hat{\mu}(x; w)$

– Poids de maximum de vraisemblance marginale L'expression explicite de la fonction g facilite le calcul du poids de maximum de vraisemblance marginale. On définit la fonction de score $S(w) = l'(w)$. Alors :

$$S(w) = \sum_{i=1}^{p} \frac{g(x_i) - \phi(x_i)}{(1-w)\phi(x_i) + wg(x_i)} = \sum_{i=1}^{p} \frac{\beta(x_i)}{1 + w\beta(x_i)} = \sum_{i=1}^{p} \frac{\beta_i}{1 + w\beta_i}$$

avec $\beta_i = \beta(x_i)$.

Soit w_{lo} défini tel que $\tau(w_{lo}) = \sqrt{2log(p)}$. Alors il est facile de montrer que $S(w)$ est une fonction décroissante de w pour w dans $[0, 1]$. De plus, l'estimateur de maximum de vraisemblance marginale de w sera donné par la racine de $S(w) = 0$ pour w dans l'intervalle $[w_{lo}, 1]$; il peut être trouvé par un algorithme de recherche binaire. Notons qu'une fois qu'on a calculé les β_i et la borne inférieure w_{lo}, l'algorithme ne dépendra pas de la distribution a priori.

Le seuil $\tau_{post}(w)$ défini à partir de la médiane *a posteriori* vérifie :

$$\mathbb{P}(\mu > 0 | X = \tau_{post}(w)) = 0.5$$

Ainsi $\tau_{post}(w)$ est la plus grande valeur des données observées pour laquelle on estimera que μ est nul, si on utilise la médiane pour établir la règle de seuillage.

2.4.2.3 La démarche pour une densité *a priori* donnée : Laplace

On va voir ce que donne concrètement la méthode quand la distribution *a priori* est une loi de Laplace.

Calcul des β_i

Pour être en mesure de calculer les β_i, il faut d'abord calculer $g(x)$ et $\beta(x)$.
La densité pour une loi de Laplace est :

$$\gamma_a(u) = \frac{a}{2} exp(-a|u|) \qquad a > 0$$

On a :

$$\begin{aligned}
g(x) &= (\gamma_a * \phi)(x) \\
&= \int_{\mathbb{R}} \gamma_a(x-t)\phi(t)dt \\
&= \int_{\mathbb{R}} \frac{a}{2} exp(-a|x-t|) \frac{1}{\sqrt{2\pi}} exp(-\frac{t^2}{2})dt \\
&= \int_{-\infty}^{x} \frac{a}{2} exp(-a(x-t)) \frac{1}{\sqrt{2\pi}} exp(-\frac{t^2}{2})dt + \int_{x}^{+\infty} \frac{a}{2} exp(-a(t-x)) \frac{1}{\sqrt{2\pi}} exp(-\frac{t^2}{2})dt \\
&= \frac{a}{2} exp(\frac{a^2}{2}) \left\{ exp(-ax) \int_{-\infty}^{x-a} \frac{1}{\sqrt{2\pi}} exp(-\frac{u^2}{2})du + exp(ax) \int_{x+a}^{+\infty} \frac{1}{\sqrt{2\pi}} exp(-\frac{v^2}{2})dv \right\} \\
&= \frac{a}{2} exp(\frac{a^2}{2}) \left\{ exp(-ax)\Phi(x-a) + exp(ax)(1-\Phi(x+a)) \right\} \\
g(x) &= \frac{a}{2} e^{\frac{a^2}{2}} \left\{ e^{-ax}\Phi(x-a) + e^{ax}\tilde{\Phi}(x+a) \right\}
\end{aligned}$$

avec $\tilde{\Phi} = 1 - \Phi$.

$$\begin{aligned}
\beta(x,a) &= \frac{g(x)}{\phi(x)} - 1 \\
&= \frac{\frac{a}{2} e^{\frac{a^2}{2}} \left\{ e^{-ax}\Phi(x-a) + e^{ax}\tilde{\Phi}(x+a) \right\}}{\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}} - 1 \\
&= \frac{a}{2} \left\{ \frac{\Phi(x-a}{\phi(x-a)} + \frac{\tilde{\Phi}(x+a)}{\phi(x+a)} \right\} - 1
\end{aligned}$$

On peut maintenant calculer $\beta(x,a)$.
Après avoir évalué les valeurs $\beta_i = \beta(x_i, a)$, pour être en mesure de calculer le poids de maximum de vraisemblance marginale pour un paramètre a fixé est la borne inférieure w_{lo}.

Moyenne *a posteriori*

Comme vu dans la partie précédente, avoir les β_i suffit au calcul de la probabilité *a posteriori* $w_{post}(x)$. Il reste alors à calculer la moyenne $\mu_1(x)$ de la distribution *a posteriori* de μ conditionnellement à $\mu \neq 0$. On a, d'après [27] :

$$f_1(\mu|x) = \begin{cases} e^{ax}\phi(\mu - x - a)/\{e^{-ax}\Phi(x-a) + e^{ax}\tilde{\Phi}(x+a)\} & if \ \ \mu \leq 0 \\ e^{-ax}\phi(\mu - x + a)/\{e^{-ax}\Phi(x-a) + e^{ax}\tilde{\Phi}(x+a)\} & if \ \ \mu > 0 \end{cases}$$

qui est une somme pondérée de distributions normales tronquées. Alors on peut montrer que, pour $x > 0$,

$$\mu_1(x) = x - a\frac{e^{-ax}\Phi(x-a) - e^{ax}\tilde{\Phi}(x+a)}{e^{-ax}\Phi(x-a) + e^{ax}\tilde{\Phi}(x+a)}$$

Ainsi la moyenne *a posteriori* est égale à $w_{post}(x)\mu_1(x)$

Médiane *a posteriori*

On suppose $x > 0$; Pour $\mu \geq 0$, on a :

$$\tilde{F}_1(\mu|x) = \frac{e^{-ax}\tilde{\Phi}(\mu - x + a)}{e^{-ax}\Phi(x-a) + e^{ax}\tilde{\Phi}(x+a)}$$

Alors, si la médiane *a posteriori* est positive, on aura :

$$\begin{aligned}\tilde{F}_1(\hat{\mu}(x;w)|x) &= \frac{wg(x) + (1-w)\phi(x)}{2wg(x)} \\ &= \frac{1}{aw}e^{-\frac{a^2}{2}}\phi(x)\frac{1 + w\beta(x)}{e^{-ax}\Phi(x-a) + e^{ax}\tilde{\Phi}(x+a)}\end{aligned}$$

Ce qui donne en simplifiant :

$$\tilde{\Phi}(\hat{\mu}(x;w) - x + a) = \frac{1}{aw}\phi(x-a)\{1 = w\beta(x)\},$$

d'où, grâce à la propriété $\tilde{\Phi}^{-1}(u) = -\Phi^{-1}(u)$, on a :

$$\hat{\mu}(x;w) = x - a - \Phi^{-1}(z_0)$$

avec $z_0 = a^{-1}\phi(x-a)\{w^{-1} + \beta(x)\}$.
On a alors :

$$\hat{\mu}(x;w) = max\left\{0, x - a - \Phi^{-1}(min\{1, z_0\})\right\}$$

Johnstone et Silverman ont développé une librairie « EbayesThresh » pour R [35] où toutes les procédures utiles sont définies, il y a des routines pour effectuer chacune des étapes intermédiaires de calcul. Pour l'utilisateur, la librairie se résume à la fonction principale, « ebayesthresh », pour laquelle il faut spécifier plusieurs paramètres. Elle prend en entrée le jeu de données bien entendu, mais également, entre autres, la distribution *a priori* à choisir parmi Laplace et quasi-Cauchy et la règle de seuillage à utiliser, à savoir médiane, moyenne, seuillage dur, seuillage doux ou même aucun seuillage si on cherche juste à estimer des paramètres. Notons aussi que, si dans la description de la méthode, le modèle a été présenté avec une variance connue égale à 1, en pratique, on peut entrer la variance en paramètre, mais bien souvent on ne la connaît pas. Dans leurs programmes, quand l'écart-type est inconnu, les auteurs l'estiment par l'écart médian absolu à la valeur zéro des $(X_i)_{i=1,\ldots,p}$, ce qui revient , dans la théorie, à calculer la valeur médiane des valeurs absolues des $(X_i)_{i=1,\ldots,p}$. En fait, ce n'est le cas qu'à un facteur d'échelle près puisque, dans R, par défaut, cet

écart-médian absolu est corrigé par un facteur d'échelle qui permet d'atteindre la consistance de l'écart médian absolu quand l'échantillon considéré est asymptotiquement normal. Ainsi, l'écart médian absolu tend vers l'écart-type quand on considère un échantillon gaussien avec $p \to +\infty$. Ce facteur d'échelle de correction est égal à 1.4826 soit approximativement $1/\Phi^{-1}(0.75)$ où Φ désigne la fonction de répartition de loi normale centrée réduite. Ce choix d'estimation de la variance n'est pas celui que nous avions fait pour les méthodes de sélection de modèle pénalisée, cependant, quand la variance sera inconnue, nous utiliserons en pratique l'estimation proposée par les auteurs.

La méthode appliquée aux données de biopuces
Cette méthode empirique bayésienne semble tout à fait adaptée dans le cadre des données de biopuces. L'hypothèse sous-jacente qui permet d'utiliser ces méthodes est de supposer que peu de gènes sont différentiellement exprimés. On peut faire un seuillage des p statistiques de test pour ne garder que les p^* les plus significatives et éventuellement appliquer par la suite une méthode de FDR si cela apporte un plus, ou si le nombre de gènes conservé après seuillage semble trop important.

2.4.3 Exemple d'application

Ici, les applications seront de deux sortes : une application sur des données simulées et une application sur des données réelles. Nous allons commencer par considérer les simulations proposées dans le cadre de la sélection de modèles, pénalisée ou pas, et appliquer les méthodes de seuillage bayésien proposées sans transformation d'ondelettes. Sur ces données simulées comme des vecteurs indépendants gaussiens, appliquer une décomposition en ondelettes pour décorréler les données n'aurait pas de sens. Pour tester la méthode EFDR, il faut utiliser des données réelles.

2.4.3.1 Données simulées

On reprend donc ici les mêmes choix de simulation que dans les sections 2.2.3 et 2.3.4. On applique alors les méthodes de seuillage empirique bayésien directement sur le vecteur simulé $\underline{y}$ qui représente la statistique de test obtenue. On utilise cette méthode avec les deux lois *a priori* prévue par la méthode : Laplace et quasi-Cauchy. On utilise les programmes de deux façons différentes : dans un premier temps en supposant la variance connue $\sigma_p = 1$ et dans un deuxième temps en la supposant inconnu (voir tableau 2.5) et en laissant le programme l'estimer par un écart médian absolu à la valeur zéro (voir tableau 2.6), et ce sur les mêmes tirages aléatoires. Pour cette méthode, il faut aussi préciser la règle de seuillage utilisée, ici on prendra la règle de seuillage définie par la médiane (*cf* 2.4.2).

Il est surprenant de constater ici, qu'à variance connue, les résultats se détériorent quand on augmente la densité du vecteur, alors que ce n'est pas le cas quand la variance est estimée. Quand on reprendra ces simulations pour comparer l'ensemble

	Cauchy		Laplace	
p_1	FDR obs.	Puissance	FDR obs.	Puissance
10	0.0485	0.8706	0.1126	0.9127
100	0.0714	0.8680	0.1356	0.9094
1000	0.1175	0.7094	0.3351	0.8731

Table 2.5 – Résultats de simulation pour la méthode de seuillage bayésien. On a 1 000 simulations et $p = 10\,000$ gènes. p_1 désigne le vrai nombre de termes de moyenne non nulle. Pour $p_1 = 10$, on a $\mu_1 = 5.21$, pour $p_1 = 100$, on a $\mu_1 = 4.52$ et pour $p_1 = 1\,000$, on a $\mu_1 = 3.11$. On donne le FDR moyen et la puissance moyenne observés sur les 1 000 simulations. On choisit une loi *a priori* de quasi-Cauchy ou de Laplace.Ici on suppose $\sigma_p = 1$ connu.

	Cauchy		Laplace	
p_1	FDR obs.	Puissance	FDR obs.	Puissance
10	0.0401	0.8675	0.1142	0.9116
100	0.0594	0.8543	0.1114	0.8961
1000	0.0259	0.46081	0.1007	0.6691

Table 2.6 – Résultats de simulation pour la méthode de seuillage bayésien. On a 1 000 simulations et $p = 10\,000$ gènes. p_1 désigne le vrai nombre de termes de moyenne non nulle. Pour $p_1 = 10$, on a $\mu_1 = 5.21$, pour $p_1 = 100$, on a $\mu_1 = 4.52$ et pour $p_1 = 1\,000$, on a $\mu_1 = 3.11$. On donne le FDR moyen et la puissance moyenne observés sur les 1 000 simulations. On choisit une loi *a priori* de quasi-Cauchy ou de Laplace.Ici on suppose σ_p inconnu.

des méthodes, on fera donc une estimation de la variance plutôt que de la supposer connue.

Les résultats obtenus ici par seuillage empirique bayésien avec estimation de la variance sont plutôt bons. On a un taux de faux positifs de l'ordre de 5% pour une loi *a priori* de quasi-Cauchy et de 10% pour une loi *a priori* de Laplace.

2.4.3.2 Jeu de données réel : eset12

Il s'agit d'un jeu de données provenant d'une expérience où seize gènes ont été injectés à différentes concentrations connues dans des hybridations différentes. Ces seize gènes sont :

[1]	« 37777_at »	« 684_at »	« 1597_at »	« 38734_at »	« 39058_at »	« 36311_at »
[7]	« 36889_at »	« 1024_at »	« 36202_at »	« 36085_at »	« 40322_at »	« 407_at »
[13]	« 1091_at »	« 1708_at »	« 33818_at »	« 546_at »		

Nous avons dans ce jeu de données deux populations et 12 répétitions pour chacune d'entre elles. On cherche à identifier les gènes différentiellement exprimés entre ces deux populations.

En tout, pour chaque biopuce on mesure l'expression de 12 626 gènes sur des puces HGU95a.

On va ici illustrer l'utilisation d'un seuillage des coefficients d'ondelettes avant d'appliquer une méthode de sélection des gènes comme FDR ou le seuillage empirique bayésien. On développe cet exemple en considérant la statistique de t-test. Notons que pour le calcul de la statistique de t-test, comme $n_1 = n_2$, la statistique de t-test standard et la statistique de t-test de Welch coïncident. On supposera que les variances sont égales et on considèrera donc que la statistique de test sous hypothèse nulle suit une loi de Student à $n_1 + n_2 - 2 = 22$ degrés de liberté. Ici, n est trop petit pour faire une approximation de la loi de Student par la loi normale.

Comme on a 12 626 gènes, et qu'on a besoin d'une puissance de 2 pour pouvoir faire la décomposition en ondelettes, on commence par sélectionner les $2^{13} = 8\,192$ gènes dont la stastitique de test est la plus grande en valeur absolue. Dans la suite de cette application, on ne considèrera que ces 8 192 valeurs de la statistique de test, considérant que tel était notre vecteur initial. Notons que dans cette sélection, on a conservé les 16 gènes qui sont réellement différentiellement exprimés.

On effectue alors une décomposition en ondelettes sur le vecteur des valeurs absolues des 8 192 statistiques de test sélectionnées, puis un seuillage sur les coefficients d'ondelettes avant de reconstruire le signal, après seuillage, par transformation inverse. Sur la figure 2.9, on peut voir les résultats obtenus après transformation inverse, selon le type de seuillage utilisé, en utilisant la statistique de t-test. Sur cette figure, on voit bien ressortir des "pics" sur la représentation graphique de la statistique de test. Les seuillages doux ou dur des coefficients d'ondelettes détériorent beaucoup ce signal. Par contre, après seuillage bayésien, le signal reconstruit semble avoir gardé les caractéristiques du signal initial mais avec un peu moins de bruit.

On n'aura pas réussi comme on l'espérait à avoir des coefficients à zéro, le nombre d'hypothèses à tester n'aura donc pas été réduit. Par contre, on peut espérer qu'avoir réduit le bruit facilitera la détection des gènes différentiellement exprimés.

Avant de réaliser une procédure FDR sur cette statistique de test reconstruite après seuillage des coefficients d'ondelettes, on va lui appliquer simplement une méthode de seuillage empirique bayésien avec une loi *a priori* de quasi-Cauchy. On va comparer les résultats obtenus avec ou sans seuillage préalable des coefficients d'ondelettes et préciser combien de ces gènes sont effectivement différentiellement exprimés.

Seuillage ondelettes	Nb gènes détectés	Nb vrai diff.	FDR obs.	Puissance
Aucun seuillage	20	14	0.300	0.875
Seuillage dur	7	7	0	0.438
Seuillage doux	3	3	0	0.188
Seuillage bayésien	16	13	0.188	0.812

Table 2.7 – Eset12 - résultats pour la méthode de seuillage bayésien selon le traitement préalable des coefficients d'ondelettes, après reconstruction. On choisit une loi *a priori* de quasi-Cauchy. Ce calcul est fait en considérant une statistique de test avec seulement 8 192 gènes. On donne le nombre de gènes détectés, et parmi ceux là le nombre de vrai gènes exprimés, ce qui permet de calculer un estimateur de FDR et une puissance de la détection.

On constate que les seuillages dur et doux nous mènent à une méthode de détection de gènes trop conservative. Le seuillage bayésien des coefficients d'ondelettes a, quant à lui, amélioré les résultats obtenus avec la statistique de test initiale dans la mesure où le FDR observé est moins important avec une puissance presqu'égale. Réduire le bruit a donc facilité la détection des gènes différentiellement exprimés par seuillage bayésien. voyons ce qu'il en est avec une procédure FDR.

Pour appliquer un test FDR sur cette statistique de test obtenue après seuillage des coefficients d'ondelettes et reconstruction, on calcule les p-valeurs selon une loi de Student à 22 degrés de liberté. Les résultats sont donnés sur le tableau 2.8 pour la procédure proposée par Benjamini et Hochberg et sur le tableau 2.9 pour la procédure proposée par Benjamini et Yekutieli. On ne s'intéresse plus ici aux seuillages dur et doux.

Pour les procédures FDR, on voit qu'on a plutôt tendance à améliorer légèrement les résultats en faisant un seuillage bayésien préalable des coefficients d'ondelettes.

Cette méthode sera donc intéressante quand on aura un nombre de gènes sur une biopuce qui est égal à une puissance de 2. Ici, nous avons pris les 8 192 plus grandes valeurs absolues des statistiques de test, mais c'était juste pour l'exemple, il paraît dangereux de sélectionner comme ça un certain nombre de données et ainsi de réduire de façon arbitraire le nombre d'hypothèses à tester. Nous allons

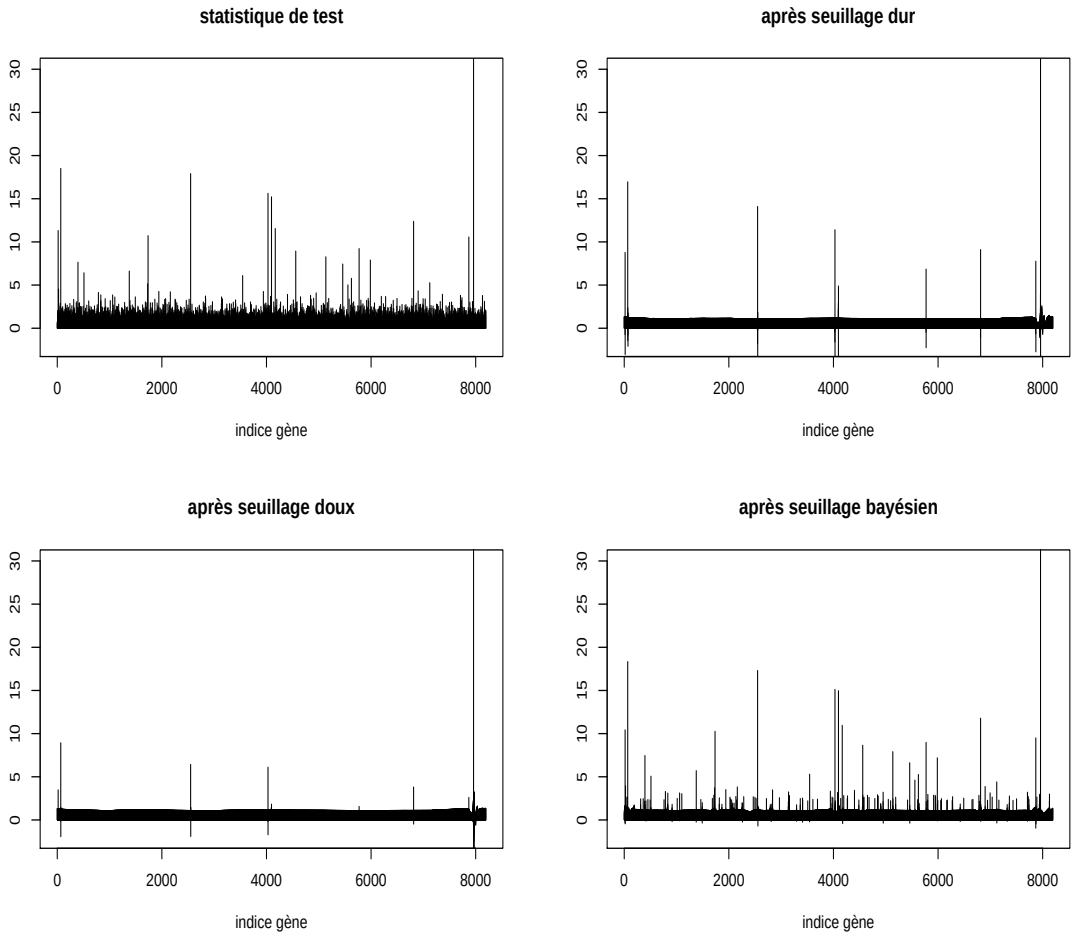

Figure 2.9 – Eset12 - Reconstruction de la statistique de t-test selon le seuillage des coefficients d'ondelettes. En haut à gauche, la statistique de t-test initiale avant toute transformation, en haut à droite, signal reconstruit après un seuillage dur des coefficients d'ondelettes, en bas à gauche, signal reconstruit après un seuillage doux des coefficients d'ondelettes,en haut à droite, signal reconstruit après un seuillage empirique bayésien (*a priori* quasi-Cauchy) des coefficients d'ondelettes. NB : ici, on a travaillé sur les valeurs absolues des statistiques de test.

Seuillage ondelettes	Nb gènes détectés	Nb vrai diff.	FDR obs.	Puissance
Aucun seuillage	23	15	0.348	0.938
Seuillage bayésien	20	14	0.300	0.875

Table 2.8 – Eset12 - résultats pour la procédure FDR de Benjamini et Hochberg avec ou sans seuillage préalable des coefficients d'ondelettes, procédure appliquée après reconstruction. Ce calcul est fait en considérant une statistique de test avec seulement 8 192 gènes. On donne le nombre de gènes détectés, et parmi ceux là le nombre de vrai gènes exprimés. Ce qui permet de calculer un estimateur de FDR et une puissance de la détection.

Seuillage ondelettes	Nb gènes détectés	Nb vrai diff.	FDR obs.	Puissance
Aucun seuillage	20	14	0.300	0.875
Seuillage bayésien	19	14	0.263	0.875

Table 2.9 – Eset12 - résultats pour la procédure FDR de Benjamini et Yekutieli avec ou sans seuillage préalable des coefficients d'ondelettes, procédure appliquée après reconstruction. Ce calcul est fait en considérant une statistique de test avec seulement 8 192 gènes. On donne le nombre de gènes détectés, et parmi ceux là le nombre de vrai gènes exprimés. Ce qui permet de calculer un estimateur de FDR et une puissance de la détection.

maintenant comparer l'ensemble des procédures proposées pour la détection des gènes différentiellement exprimés.

Applications et comparaison de l'ensemble des procédures

Pour comparer l'ensemble des méthodes de détection des gènes différentiellement exprimés, nous allons d'abord reprendre les simulations réalisées au sein de chaque section pour en faire une brève synthèse avant de nous intéresser à des données réelles : d'abord le jeu de données "eset12" déjà évoqué dans la section précédente puis un jeu de données de l'Institut Curie.

2.5.1 Données simulées

On reprend le protocole de simulation utilisé dans les applications des sections précédentes. On réalise ici 1 000 simulations et les différentes méthodes sont toutes appliquées sur les mêmes tirages aléatoires. On donne les résultats pour chacun des trois protocoles de simulation dans les tableaux 2.10, 2.11 et 2.12. Les résultats sont donnés pour plusieurs valeurs du seuil q, certaines procédures ne dépendent pas d'un seuil, on donne alors un seul résultat. Pour les procédures de sélection de modèles pénalisées, on a supposé la variance σ_p^2 connue et égale à 1.

Les résultats obtenus montrent bien la proximité entre les méthodes FDR proposée par Benjamini et Hochberg et la procédure de sélection de modèles pénalisée proposée par Abramovich *et al.* . Pour les différentes proportions de composantes non nulles et pour les différents seuils envisagés, les résultats sont presque identiques, et contrôlent le FDR au taux voulu. Les résultats de ces deux procédures avec un seuil $q = 0.05$ sont aussi très similaires aux résultats de la méthode de seuillage empirique bayésien avec un *a priori* de quasi-Cauchy, c'est un peu moins vrai pour le troisième jeu de simulation, où $p_1 = 1\ 000$ et où les composantes non nulles se distinguent moins facilement (puisque $\mu_1 = 3.11$). Comme on l'avait déjà remarqué, les méthodes proposées par Golubev sont plus conservatives, surtout celle avec pénalité conservative. Quant à la méthode de seuillage empirique bayésien avec un *a priori* de Laplace, c'est la méthode la plus puissante mais son FDR est un peu trop élevé. Elle pourra s'avérer intéressante si on privilègie la puissance au taux de faux positifs. Dans le tableau 2.13, on cherche à étudier l'influence de l'initialisation si on suppose σ_p^2 inconnu et qu'on utilise un $\hat{k}_p$ obtenu par une autre méthode pour l'estimer. Nous allons faire plusieurs initialisations différentes : sélection de modèle par procédure FDR ou seuillage empirique bayésien.

Méthode	FDR observé			Puissance observée		
	q=0.01	q=0.05	q=0.1	q=0.01	q=0.05	q=0.1
FDR BH	0.0104	0.0510	0.1018	0.7723	0.8714	0.9093
FDR BY	0.0015	0.0049	0.0104	0.5979	0.7200	0.7737
Pen. Abramovich 1	0.0104	0.0510	0.1013	0.7716	0.8714	0.9092
Pen. Cons. Golubev		0			0.2551	
Subst. Golubev		0.0201			0.8268	
Ebayes Cauchy		0.0526			0.8682	
Ebayes Laplace		0.1225			0.9136	

Table 2.10 – Résultats de simulation 1 pour la sélection de modèles pour les différentes méthodes. On a 1 000 simulations et $p = 10\ 000$ gènes. On fait varier le seuil $q \in \{0.01, 0.05, 0.1\}$. p_1 désigne le vrai nombre de termes de moyenne non nulle. Ici $p_1 = 10$, et $\mu_1 = 5.21$. On donne le FDR moyen et la puissance moyenne observés sur les 1 000 simulations.

Méthode	FDR observé			Puissance observée		
	q=0.01	q=0.05	q=0.1	q=0.01	q=0.05	q=0.1
FDR BH	0.0096	0.0485	0.0990	0.7103	0.8434	0.8905
FDR BY	0.0009	0.0050	0.0097	0.4803	0.6459	0.7121
Pen. Abramovich 1	0.0096	0.0485	0.0989	0.7101	0.8434	0.8905
Pen. Cons. Golubev		0.0002			0.3365	
Subst. Golubev		0.0132			0.7336	
Ebayes Cauchy		0.0578			0.8539	
Ebayes Laplace		0.1118			0.8956	

Table 2.11 – Résultats de simulation 2 pour la sélection de modèles pour les différentes méthodes. On a 1 000 simulations et $p = 10\ 000$ gènes. On fait varier le seuil $q \in \{0.01, 0.05, 0.1\}$. p_1 désigne le vrai nombre de termes de moyenne non nulle. Ici $p_1 = 100$, et $\mu_1 = 4.52$. On donne le FDR moyen et la puissance moyenne observés sur les 1 000 simulations.

Méthode	FDR observé			Puissance observée		
	q=0.01	q=0.05	q=0.1	q=0.01	q=0.05	q=0.1
FDR BH	0.0087	0.0456	0.0904	0.3099	0.5533	0.6664
FDR BY	0.0009	0.0044	0.0089	0.0951	0.2292	0.3128
Pen. Abramovich 1	0.0087	0.0456	0.0903	0.3098	0.5533	0.6663
Pen. Cons. Golubev		0.0482			0.0482	
Subst. Golubev		0.0296			0.4823	
Ebayes Cauchy		0.0251			0.4568	
Ebayes Laplace		0.1006			0.6750	

Table 2.12 – Résultats de simulation 3 pour la sélection de modèles pour les différentes méthodes. On a 1 000 simulations et $p = 10\ 000$ gènes. On fait varier le seuil $q \in \{0.01, 0.05, 0.1\}$. p_1 désigne le vrai nombre de termes de moyenne non nulle. Ici $p_1 = 1\ 000$, et $\mu_1 = 3.11$. On donne le FDR moyen et la puissance moyenne observés sur les 1 000 simulations.

	Pen. Abramovich 1		Subst. Golubev	
Initialisation	FDR obs.	Puissance	FDR obs.	Puissance
Fdr BH	0.0466	0.8400	0.0030	0.5875
ebayes Cauchy	0.0476	0.8412	0.0030	0.5865
ebayes Laplace	0.0512	0.8461	0.0030	0.5854

Table 2.13 – Comparaison des initialisations pour la sélection de modèles pénalisée par méthode d'Abramovich *et al.* et de Golubev (méthode de substitution). On a 1 000 simulations et $p = 10\ 000$ gènes. p_1 désigne le vrai nombre de termes de moyenne non nulle. On prend $p_1 = 100$, $\mu_1 = 4.52$ et, pour Abramovich *et al.* , $q = 0.05$. On donne le FDR moyen et la puissance moyenne observés sur les 1 000 simulations.

On constate que pour des initialisations avec une procédure FDR de Benjamini Hochberg avec un seuil $q = 0.05$ ou une initialisation par seuillage empirique bayésien avec une loi *a priori* de quasi-Cauchy, on obtient des résultats très similaires. Cela est cohérent puisqu'on a vu que ces deux procédures donnaient des résultats très proches. La procédure de seuillage empirique bayésien avec une loi *a priori* de Laplace a tendance à donner un $\hat{k}_p$ plus élevé, ce qui conduit à une sous-estimation de la variance dans les méthodes de sélection de modèles pénalisées et donc à la détection de davantage de gènes comme différentiellement exprimés.

2.5.2 Jeu de données "eset12"

Il s'agit du jeu de données décrit dans la section 2.4.3.2. Nous allons ici essayer les différentes techniques que nous avons décrites dans l'ensemble du chapitre pour évaluer leur qualité respective sur ce jeu de données. On a déjà étudié sur une sélection de gènes la performance de la méthode EFDR. On ne la ré-étudiera pas ici puisqu'on va conserver la totalité des 12 626 gènes.

On calcule trois statistiques de test différentes : la statistique de différence des moyennes, la statistique de t-test et la statistique de Turkheimer *et al.* .

Pour les méthodes où il est nécessaire de fixer un seuil α, on a pris $\alpha = 0.05$.

2.5.2.1 Statistique de différence des moyennes

Cette statistique de test consiste à faire une simple différence des moyennes. Le problème de cette statistique de test, c'est qu'on ne connaît pas sa loi sous l'hypothèse nulle. Pour les méthodes de type FDR, on calculera la statistique de test par méthode de rééchantillonnage. Pour les méthodes de type sélection de modèles pénalisée, il faudra utiliser un seuil initial pour obtenir une estimation de la variance $\hat{\sigma}^2_{I_{k_p}}$. Pour estimer la variance, on a ici pris le seuil initial obtenu par la procédure FDR. On donne les résultats dans le tableau 2.14. FDR BH désigne la procédure de Benjamini et Hochberg et FDR BY celle de Benjamini et Yekutieli.

En plus de ces résultats, on peut signaler une relative instabilité des p-valeurs obtenues par ré-échantillonnage : en effet, pour avoir des résultats plus fiables il faudrait faire un nombre de rééchantillonnages important. Or ceci est coûteux en temps de calcul. Dans ces applications, on a effectué 1 000 rééchantillonnages. Pourtant, cela ne suffit pas à stabiliser les résultats, puisque avec un deuxième calcul des p-valeurs par réchantillonnage, on a trouvé, pour les procédures FDR, 39 gènes différentiellement exprimés au lieu de 34 pour le premier calcul.

Pour les méthodes pénalisées, on a choisi une initialisation donnée. Expérimentalement, on observe qu'avec les méthodes proposées par Golubev, les résultats sont ici assez stables. Une initialisation de moyenne qualité n'est pas si grave. Cela est sans doute dû au fait qu'ici on a une très grande proportion de gènes non exprimés. Ainsi, si on en oublie quelques uns pour calculer la variance, le résultat ne change pas beaucoup. Et réciproquement, si on considère par erreur une poignée de gènes

Méthode	Nb gènes détectés	Nb vrai diff.	FDR obs.	Puissance
FDR BH	34	15	0.559	0.938
FDR BY	34	15	0.559	0.938
Pen. Abramovich 1	47	15	0.681	0.938
Pen. Cons. Golubev	24	13	0.375	0.813
Subst. Golubev	28	14	0.500	0.875
Ebayes Cauchy	106	16	0.849	1.00

Table 2.14 – Eset12 - statistique de différence des moyennes - résultats pour les méthodes de détection des gènes différentiellement exprimés. On donne le nombre de gènes détectés, et parmi ceux là le nombre de gènes réellement exprimés, ce qui permet de calculer un estimateur de FDR et une puissance de la détection.

différentiellement exprimés, on peut espérer que le biais qu'ils apporteront à la variance sera relativement faible compte tenu du nombre total de gènes considéré. Toutefois, cela ne suffit pas à expliquer la stabilité de la méthode. Pour la méthode proposée par Abramovich *et al.* , on observe une plus grande variabilité. En effet si pour la méthode conservative (resp. par substitution) de Golubev, on observait un nombre de gènes détectés compris entre 24 et 26 (resp. 28 et 32), pour des $\hat{k}_p$ compris entre 20 et 120, pour la même gamme de $\hat{k}_p$, la méthode d'Abramovich *et al.* détecte entre 45 et 64 gènes.

Pour cette statistique de test, c'est la méthode de Golubev avec pénalité conservative qui semble donner les meilleurs résultats.

2.5.2.2 Statistique de t-test

Les résultats sont résumés dans le tableau 2.15. Pour les procédures FDR, on considère que la statistique de test sous l'hypothèse nulle suit une loi de Student à $n-2 = 22$ degrés de liberté. Pour les procédures de sélection de modèle pénalisée, on a vu qu'on avait besoin d'un $\hat{k}_p$ initial pour estimer la variance $\hat{\sigma}_{I_{k_p}}$. Ici, la statistique de t-test sous hypothèse nulle est censée suivre une loi de Student à 22 degrés de liberté de variance $n/(n-2) = 1.1$. Pour les procédures de sélection de modèle pénalisée et le seuillage emirique bayésien, les résultats présentés dans le tableau ont donc été obtenus en supposant σ_p^2 connu et égal à 1.1. On donne dans le tableau 2.16 les résultats obtenus en supposant σ_p inconnu et en prenant comme initialisation le $\hat{k}_p$ obtenu par différentes procédures. On avait précisé dans la section 2.4.2.2 que, pour la méthode de seuillage empirique bayésien, les auteurs estimaient l'écart-type par l'écart médian absolu. On va donc regarder pour cette méthode les différences de résultats quand on utilise une estimation de σ_p^2 calculée à partir d'un $\hat{k}_p$ initial ainsi que ce qui se passe pour les autres méthodes si on utilise l'estimation de la variance proposée dans les algorithmes de seuillage bayésien (colonne *mad- median absolute*

deviation- du tableau). On constate ici que quelle que soit l'initialisation choisie, les résultats varient assez peu. Cela est dû au fait que l'on reste dans une gamme de bonnes initialisations, avec des méthodes valables. On notera toutefois que la méthode de sélection de modèles pénalisée avec la pénalité proposée par Abramovich *et al.* et le seuillage empirique bayésien sont plus sensibles à cette estimation de la variance.

En ce qui concerne les résultats du tableau 2.15, on constate des résultats globalement satisfaisants pour l'ensemble des méthodes. Un bémol toutefois pour la méthodes de sélection pénalisée d'Abramovich *et al.* qui a tendance à détecter trop de gènes. Notons qu'ici on ne donne pas de résultat pour la méthode de seuillage bayésien empirique avec une loi de Laplace *a priori* pour des raisons informatiques : cela génère une erreur d'optimisation dans le programme fourni par les auteurs.

Méthode	Nb gènes détectés	Nb vrai diff.	FDR obs.	Puissance
FDR BH	23	15	0.348	0.938
FDR BY	19	14	0.263	0.875
Pen. Abramovich 1	30	15	0.500	0.938
Pen. Cons. Golubev	20	14	0.300	0.875
Subst. Golubev	23	15	0.348	0.938
Ebayes Cauchy	25	15	0.400	0.938

Table 2.15 – Eset12 - statistique de t-test - résultats pour les méthodes de détection des gènes différentiellement exprimés. On donne le nombre de gènes détectés, et parmi ceux là le nombre de gènes réellement exprimés, ce qui permet de calculer un estimateur de FDR et une puissance de la détection.

Estimation de σ_p^2	$\hat{k}_p$= 19	$\hat{k}_p$=23	$\hat{k}_p$=40	*mad*
Pen. Abramovich 1	37 (15)	37 (15)	38 (15)	37 (15)
Pen. Cons. Golubev	21 (15)	21 (15)	22 (15)	22 (15)
Subst. Golubev	24 (15)	24 (15)	28 (15)	28 (15)
Ebayes Cauchy	38 (15)	40 (15)	44 (15)	40 (15)

Table 2.16 – Eset12 - statistique de t-test - résultats pour les méthodes de sélection de modèle pénalisées en fonction de l'estimation de la variance choisie. On donne le nombre de gènes détectés, et, entre parenthèses le nombre de vrai positifs.

Globalement, sur l'ensemble des méthodes considérées ici, la statistique de t-test apporte des résultats plus satisfaisants que la statistique de différence des moyennes. Voyons maintenant ce qu'il en est pour la statistique de test de Turkheimer *et al.* .

2.5.2.3 Statistique de Turkheimer *et al.*

On s'intéresse maintenant à la statistique de Turkheimer *et al.* . On ne dispose pas de la loi de cette statistique de test sous l'hypothèse nulle. Pour les procédures FDR, on utilise les p-valeurs obtenues par rééchantillonnage. Pour les procédures pénalisées, on doit estimer la variance, dans le tableau on a initialisé avec $\hat{k}_p = 30$ obtenu à partir des procédures FDR. Notons toutefois que si on considère une initialisation avec $\hat{k}_p = 23$ obtenu par seuillage bayésien empirique, les résultats ne changent pas, excepté pour la méthode d'Abramovich *et al.* pour laquelle on trouve alors 18 gènes différentiellement exprimés (dont 13 vrais).

Méthode	Nb gènes détectés	Nb vrai diff.	FDR obs.	Puissance
FDR BH	30	14	0.533	0.875
FDR BY	30	14	0.533	0.875
Pen. Abramovich	19	13	0.316	0.813
Pen. Cons. Golubev	12	11	0.083	0.688
Subst. Golubev	17	12	0.294	0.750
Ebayes Cauchy	23	14	0.391	0.875

Table 2.17 – Eset12 - statistique de Turkheimer *et al.* - résultats pour les méthodes de détection des gènes différentiellement exprimés. On donne le nombre de gènes détectés, et parmi ceux là le nombre de gènes réellement exprimés, ce qui permet de calculer un estimateur de FDR et une puissance de la détection.

Pour la plupart des méthodes on observe des résultats plutôt satisfaisants pour cette statistique de test. On peut toutefois noter que la pénalité conservative de Golubev, si elle limite réellement le taux de faux positifs, s'avère moins puissante que les autres. Les méthodes qui donnent le moins satisfaction ici sont les méthodes FDR. Comme ce sont les seules procédures où on a utilisé les p-valeurs calculées par rééchantillonnage, on peut penser que ce manque de performance est lié à une mauvaise qualité de la méthode de calcul des p-valeurs. Pour le vérifier on considère les procédures FDR d'une autre façon, celle présentée en section 2.3.2.1. Cela revient à considérer que la statistique de test sous hypothèse nulle suit une loi normale centrée et de variance σ_p^2 inconnue. On a alors besoin d'estimer cette variance, et pour cela on procède de même que pour les méthodes pénalisées. Nous avons essayé une telle procédure en essayant toute une gamme de valeurs initiales. Les résultats sont relativement stables : pour la procédure FDR de Benjamini-Hochberg, pour tout $\hat{k}_p$ entier compris entre 22 et 109, on détecte 19 gènes dont 13 réellement exprimés. Si on considère une valeur inférieure à 22, le résultat ne change pas beaucoup puisqu'alors on détecte 18 gènes dont 13 réellement exprimés. Pour la procédure FDR de Benjamini-Hochberg, pour tout $\hat{k}_p$ entier compris entre 5 et 64, on détecte 17 gènes dont 12 réellement exprimés. Les résultats deviennent alors comparables à

ceux obtenus avec les autres méthodes.

Toutefois cette statistique de test ne semble pas apporter une réelle amélioration à la statistique de t-test, et même au contraire puisque, quand on sélectionne le même nombre de gènes avec les deux statistiques, on a un gène différentiellement exprimé de plus dans l'ensemble des gènes détectés par statistique de t-test. Etant donné, son fort coût algorithmique, la statistique de Turkheimer *et al.* ne sera pas retenue ici.

2.5.3 Jeu de données "fortes doses" contre "faibles doses"

On considère ici un jeu de données de l'Institut Curie. On dispose de 34 profils d'expression répartis en deux populations. La première population est constituée de 24 biopuces correspondant à des levures ayant été exposées à de fortes doses d'irradiation (200 Gray (Gy)). La deuxième population se compose de 10 biopuces ayant été réalisées à partir de levures exposées à de faibles doses d'irradiation (de 7.5μGy à 36.2 Gy). Ici, on a un nobmre de gènes, p, égal à 6 327.

2.5.3.1 Statistique de différence des moyennes

Les résultats obtenus sont donnés dans le tableau 2.18. Pour les procédures de sélection FDR, les résultats donnés sont ceux obtenus en considérant la statistique de test obtenue par ré-échantillonnage.

Méthode	Nb gènes détectés
FDR BH	2299
FDR BY	1244
Pen. Abramovich 1	91 - 537 - 1424
Pen. Cons. Golubev	16 - 303 - 3163
Subst. Golubev	43 - 125 - 184
Ebayes Cauchy	141
Ebayes Laplace	221

Table 2.18 – Faibles doses et fortes doses - statistique de différence des moyennes - résultats pour les méthodes de détection des gènes différentiellement exprimés. Quand plusieurs valeurs sont données, cela correspond à plusieurs initialisations $\hat{k}_p$ possibles pour l'estimation de la variance. Ici, de gauche à droite, avec $\hat{k}_p = 141, 1244, 2299$.

2.5.3.2 Statistique de t-test

On fait ici l'hypothèse d'égalité des variances et on considère un t-test standard. Sous l'hypothèse nulle, la statistique de test suit une loi de Student à 32 degrés de liberté. Ici, on est dans les conditions d'approximation de la loi de Student par une loi normale mais on préfère toutefois, pour plus de précision, travailler avec la loi de Student. Ici la variance sous hypothèse nulle est égale à $32/30 = 1.0667$. C'est cette variance comme valeur connue de σ_p dans l'ensemble des procédures.

Méthode	Nb gènes détectés
FDR BH	2307
FDR BY	1128
Pen. Abramovich 1	2379
Pen. Cons. Golubev	6327
Subst. Golubev	2082
Ebayes Cauchy	6327
Ebayes Laplace	6327

Table 2.19 – Faibles doses et fortes doses - statistique de t-test - résultats pour les méthodes de détection des gènes différentiellement exprimés.

Ici, on remarque que dans l'ensemble beaucoup de gènes, voire tous, sont détectés comme étant différentiellement exprimés. Signalons de plus que pour les méthodes de seuillage empirique bayésien, on a aussi cherché à détecter des gènes différentiellement exprimés en supposant la variance inconnue et en laissant le programme estimer l'écart-type par l'écart médian absolu à la valeur zéro. Pour les deux lois *a priori* possibles, on ne détecte alors aucun gène. Cela montre bien qu'il est ici très difficile de fixer un seuil entre gènes différentiellement exprimés et gènes non différentiellement exprimés et que l'estimation de la variance devient difficile dans un tel cadre.

2.5.3.3 Statistique de Turkheimer *et al.*

Les résultats obtenus sont donnés dans le tableau 2.20. Pour les procédures de sélection FDR, les résultats donnés sont ceux obtenus en considérant la statistique de test obtenue par rééchantillonnage. Précisons ici que si on change l'estimation de la variance pour les procédures de seuillage bayésien, en utilisant un $\hat{k}_p$ initial on change beaucoup les résultats. Ainsi, avec un *a priori* de Laplace, on détecte tous les gènes que l'on prenne $\hat{k}_p = 702$ ou $\hat{k}_p = 1519$. Pour un *a priori* de quasi-Cauchy, on détecte 69 ou 1 226 gènes.

Les résultats laissent plutôt perplexes dans l'ensemble car ils sont très irréguliers. Ici, les méthodes pénalisées sont très sensibles à l'initialisation, c'est très certainement parce que pour ce jeu de données, les gènes différentiellement exprimés sont

Méthode	Nb gènes détectés
FDR BH	1519
FDR BY	702
Pen. Abramovich 1	5 - 296
Pen. Cons. Golubev	1 - 1
Subst. Golubev	0 - 48
Ebayes Cauchy	0
Ebayes Laplace	0

Table 2.20 – Faibles doses et fortes doses - statistique de t-test - résultats pour les méthodes de détection des gènes différentiellement exprimés. Quand plusieurs valeurs sont données, cela correspond à plusieurs initialisations $\hat{k}_p$ possibles pour l'estimation de la variance. Ici, de gauche à droite, avec $\hat{k}_p = 702$ ou $\hat{k}_p = 1519$.

moins séparés des autres gènes que dans le cas du jeu de données précédent, eset12. En conséquence, les méthodes manquent de stabilité. Les méthodes peinent ici à choisir un seuil de détection, il semble qu'il n'y a pas des différentiellement exprimés et des non différentiellement exprimés mais toute une gamme de niveau d'expression différentielle qui va de l'un à l'autre, ce qui n'est pas tellement étonnant dans le cadre de données réelles. Si on devait choisir, on aurait plutôt tendance à dire qu'il y a beaucoup de gènes différentiellement exprimés, entre 1200 et 2500.

Conclusion du deuxième chapitre

Le problème de la détection des gènes différentiellement exprimés n'est pas simple. Dans ce chapitre, nous avons étudié et proposé plusieurs méthodes de détection des gènes différentiellement exprimés. Pour le choix de la statistique de test, la statistique de t-test semble la plus adaptée. Une fois la statistique de test calculée, le choix d'un seuil de détection pour les gènes différentiellement exprimés est un problème bien plus complexe. En effet, les méthodes proposées donnent dans l'ensemble de très bons résultats sur les données simulées, ce qui est normal puisque ces méthodes ont été développées justement dans ce cadre. On a pu constater aussi que, dans le cadre du jeu de données "eset12", les résultats étaient très bons. Les méthodes de sélection de modèles par procédure FDR, la sélection de modèle avec la pénalité proposée par Abramovich *et al.* ou la méthode de substitution de Golubev donnent les meilleurs résultats. Il s'agit ici d'un jeu de données plus ou moins idéal, où on connaît à l'avance les gènes différentiellement exprimés ce qui permet d'évaluer les performances des méthodes utilisées. Ce jeu de données a été construit comme tel, en injectant certains gènes pour les rendre différentiellement exprimés. Dans des jeux de données plus complexes, la détection s'avère nettement plus difficile, et déterminer une règle de décision devient plus difficile. Dans le jeu de données de l'Institut Curie, on a simplement voulu comparer deux conditions expérimentales différentes. Ici, il semble qu'il n'y ait pas des gènes différentiellement exprimés et des gènes non différentiellement exprimés, mais plutôt des gènes suivant toute la gamme des expressions possibles. Il est alors difficile d'aller au-delà d'un simple classement des niveaux d'expression établi à partir de la statistique de test considérée. Selon l'objectif de l'utilisateur on pourra vouloir détecter un faible nombre de gènes différentiellement exprimés mais avec une très grande certitude ou détecter beaucoup plus de gènes, quite à faire de nombreuses erreurs, et essayer de retrouver des familles de gènes répondant à des fonctions cellulaires identiques pour essayer d'inférer l'activation d'une réaction cellulaire à un traitement.

Chapitre 3

Réduction de dimension et classification supervisée

Introduction

Un des enjeux du traitement statistique des données issues des biopuces à ADN est l'analyse discriminante à but décisionnel. D'un point de vue statistique, ce grand nombre de covariables devant un petit nombre d'observations rend l'analyse discriminante difficile. Une façon de contourner ce « fléau de la dimension » consiste à réduire cette dimension. Sans préalablement réduire la dimension, les méthodes standards de statistique en classification supervisée, même utilisables, ne sont pas très performantes. En particulier, un des problèmes est lié à une forte multicolinéarité des p régresseurs. Les solutions des équations intervenant dans les méthodes traditionnelles peuvent ne plus être uniques et devenir instables. Même si on peut utiliser tous les gènes, il ne semble pas souhaitable de le faire. En effet, utiliser tous les gènes introduit du bruit via des gènes qui ont un faible pouvoir discriminant, dégradant ainsi les performances des méthodes utilisées. Dans ce cas, les méthodes de réduction de dimension peuvent s'avérer utiles.

Une approche semi-paramétrique a été proposée par Antoniadis *et al.* [6]. Ils suggèrent d'utiliser la méthode MAVE (acronyme de Minimum Average Variance Estimation, Xia *et al.* [48]) pour réduire la dimension avant d'appliquer une régression logistique paramétrique ou non paramétrique. La procédure MAVE est fondée sur un critère de moindres carrés local combiné à une estimation non paramétrique par polynômes locaux de la fonction de régression. Bien qu'applicable aux modèles linéaires généralisés (GLM), cette méthode n'exploite pas la structure particulière de ces modèles à savoir la relation entre espérance et variance. Par ailleurs, il est bien connu que, dans le cadre des GLM, les critères basés sur les moindres carrés et ceux basés sur la vraisemblance ne coincident que dans le cas particulier des modèles gaussiens.

Dans le cadre de cette thèse, j'ai développé, en collaboration avec Sophie Lambert-Lacroix, une approche de type semi-paramétrique qui utilise des estimateurs par vraisemblance locale dans les modèles généralisés en indice simple. Cette dernière est comparable à MAVE : le critère basé sur les moindres carrés est remplacé par un critère sur la vraisemblance locale, exploitant ainsi la structure particulière des modèles généralisés.

Modèles linéaires généralisés et notations

Nous allons commencer par introduire quelques notations et définir les modèles linéaires généralisés (GLM) canoniques. Nous évoquerons aussi les méthodes d'inférence non paramétrique dans les GLM.

3.1.1 Notations

Pour $\underline{k} = (k_1, \ldots, k_p)$ un vecteur de $\mathbb{N}^p$, on introduit les notations suivantes :

$$|\underline{k}| = \sum_{j=1}^{p} k_j$$

$$\underline{k}! = \prod_{j=1}^{p} k_j!$$

$$\forall \underline{x} \in \mathbb{R}^p \ , \ \underline{x}^{\underline{k}} = \prod_{j=1}^{p} x_j^{k_j}$$

Soit f une fonction ayant au moins q dérivées partielles continues dans $\mathbb{R}^p$. Pour $\underline{k} \in A_q = \{\underline{k}; |\underline{k}| \leq q\}$, la dérivée partielle $\partial^{|\underline{k}|} f(\underline{x}) / \partial \underline{x}^{\underline{k}}$ est notée $D^{\underline{k}} f(\underline{x})$. On note aussi ∇ l'opérateur du gradient. Pour une matrice A, on notera A^T sa transposée et $|A|$ son déterminant.

3.1.2 Modèles linéaires généralisés

Soit $(\underline{X}_1^T, Y_1), \ldots, (\underline{X}_n^T, Y_n)$ un échantillon aléatoire indépendant issu de la loi de $(\underline{X}^T, Y)$, où Y est la réponse scalaire associée au vecteur de covariables $\underline{X} \in \mathbb{R}^p$. On suppose que la densité conditionnelle de Y sachant $\underline{X} = \underline{x}$ appartient à la famille exponentielle canonique (voir MacCullagh et Nelder [31])

$$f_{Y/\underline{X}}(y) = \exp\{\frac{y\theta(\underline{x}) - b(\theta(\underline{x}))}{a(\phi)} + c(y, \phi)\} \tag{3.1}$$

où $a(\cdot)$, $b(\cdot)$ et $c(\cdot,\cdot)$ sont des fonctions connues. Le paramètre $\theta(\cdot)$ est appelé paramètre canonique et ϕ est appelé paramètre de dispersion. Sous le modèle 3.1, on peut montrer que

$$\mathbb{E}(Y|\underline{X}=\underline{x}) = b'(\theta(\underline{x})) = \mu(\underline{x}), \quad \mathrm{Var}(Y|\underline{X}=\underline{x}) = a(\phi)b''(\theta(\underline{x})),$$

où b' et b'' désignent respectivement les dérivées première et seconde de b relativement au paramètre canonique θ. Dans les modèles linéaires généralisés paramétriques, certaines transformations de la fonction de régression $\mu(\underline{x}) = \mathbb{E}(Y|X=\underline{x})$ sont supposées être linéaires en les covariables

$$g(\mu(\underline{x})) = \eta(\underline{x}) = \underline{x}^T\gamma. \tag{3.2}$$

La fonction g est appelée fonction de lien. Le choix de $g = (b')^{-1}$ permet d'identifier le prédicteur linéaire $\eta(\underline{x})$ avec le paramètre canonique $\theta(\underline{x})$, et ce lien particulier est appelée lien canonique.

Dans un cadre plus général, quand la vraisemblance n'est pas complètement connue, il est seulement possible de spécifier la relation entre la moyenne et la variance des observations comme suit :

$$\mathrm{Var}(Y|\underline{X}=\underline{x}) = a(\phi)V(\mu(\underline{x})) \tag{3.3}$$

pour une fonction positive connue V et la quasi-vraisemblance est définie par la propriété suivante

$$\frac{\partial Q(u,y)}{\partial u} = \frac{y-u}{V(u)}. \tag{3.4}$$

Alors les procédures qui utilisent la log-vraisemblance doivent être remplacées par des procédures analogues dans lesquelles la fonction de quasi-vraisemblance remplace la fonction de vraisemblance.

La log-vraisemblance associée au modèle (3.1) est un cas spécial de fonction de quasi-vraisemblance avec $V(\cdot) = a(\phi)b'' \circ (b')^{-1}(\cdot)$. Par exemple, quand $V(\mu) = (1-\mu)\mu$, la méthode de quasi-vraisemblance coïncide avec la méthode de log-vraisemblance dans le cas d'un modèle de Bernoulli.

3.1.3 Inférence dans les GLM : approche non paramétrique

On rappelle ici la méthode des polynômes locaux introduite par Fan et Gijbels [18].

L'idée est d'approximer la fonction η localement par une fonction polynômiale d'ordre q,

$$\eta(u) \sim \sum_{\underline{k}\in A_q} D^{\underline{k}}\eta(x)\ (u-x)^{\underline{k}} = \sum_{\underline{k}\in A_q} a_{\underline{k}}\ (u-x)^{\underline{k}},$$

pour u dans un voisinage de x. Par définition, $A_q = \{\underline{k}, |\underline{k}| \leq q\}$ et $D^{\underline{k}}f(x)$ désigne la dérivée partielle $\partial^{|\underline{k}|}f(x)/\partial x^{\underline{k}}$.

Soient K^p un noyau p-dimensionnel avec une matrice de taille de fenêtres de lissage H et $K_H^p(\cdot) = \det(H)^{-1}K^p(H^{-1} \times \cdot)$ le changement d'échelle de K^p. La vraisemblance locale est une vraisemblance pondérée, avec les poids $K_H^p(\underline{X}_i - x)$. On note $\mathcal{L}(u, Y)$ la fonction log-vraisemblance avec $\eta(x)$ remplacé par u ; la log-vraisemblance locale est définie par

$$\sum_{i=1}^{n} \mathcal{L}\left(\sum_{\underline{k}\in A_q} a_{\underline{k}}\ (\underline{X}_i - x)^{\underline{k}}, Y_i\right)\quad K_H^p(\underline{X}_i - x). \tag{3.5}$$

La méthode des polynômes locaux conduit à $\widehat{D^{\underline{k}}\eta}(x) = \underline{k}!\hat{a}_{\underline{k}}(x)$, où $\{\hat{a}_{\underline{k}}(x), \underline{k} \in A_q\}$ maximise le critère (3.5) comme fonction de $\{a_{\underline{k}}, \underline{k} \in A_q\}$. En particulier, on a

$$\hat{\eta}(x) = \hat{a}_{(0,\ldots,0)}(x), \qquad \widehat{\nabla\eta}(x) = (\hat{a}_{e_1}(x), \ldots, \hat{a}_{e_p}(x))^T.$$

Les estimateurs $\hat{a}_{\underline{k}}$ sont déterminés par un algorithme de type moindres carrés itératif (IRLS, proposé par Green [23]) avec une matrice de plan d'expérience et une matrice de poids appropriée. Par ailleurs, Fan et Gijbels [18] décrivent plusieurs méthodes pour déterminer la taille de fenêtre.

Le problème de la méthode des polynômes locaux même en petite dimension est celui du "fléau de la dimension". Ce problème se réfère au fait qu'un voisinage local en grande dimension n'est plus vraiment local. En effet un voisinage avec un pourcentage de points donné peut se révéler très grand en dimension supérieure à un. Une façon de contourner ce problème consiste à réduire la dimension. C'est ce à quoi nous allons nous intéresser dans la section suivante.

Réduction de dimension dans les GLM

On envisage une méthode de réduction de dimension basée sur une approche semi-paramétrique utilisant les modèles linéaires généralisés en indice simple. Après avoir introduit ces modèles, nous donnons la méthode d'estimation utilisant la vraisemblance locale. Nous supposons de plus que le modèle est suffisamment régulier pour que la fonction de lien g soit un C^1-difféomorphisme.

3.2.1 Modèles linéaires généralisés en indice simple

Pour venir à bout du problème de dimension, il est assez répandu de commencer par projeter toutes les covariables $\underline{X}$ sur un espace linéaire engendré par les covariables et ensuite d'ajuster une courbe non paramétrique à ces combinaisons linéaires. Ce principe conduit aux modèles en indice simple :

$$Y = \tilde{\mu}(\underline{\beta}^T \underline{X}) + \varepsilon \tag{3.6}$$

avec $\mathbb{E}[\varepsilon|\underline{X}] = 0$ presque sûrement. Par définition des modèles linéaires généralisés, $Y = \mu(\underline{X}) + \varepsilon$. Dans les modèles en indice simple, on suppose donc l'existence d'une direction $\underline{\beta} \in \mathbb{R}^p$ et d'une fonction $\tilde{\mu} : \mathbb{R} \mapsto \mathbb{R}$ telle que $\mu(\underline{X}) = \tilde{\mu}(\underline{\beta}^T \underline{X})$ (notons que l'on a $\mu : \mathbb{R}^p \mapsto \mathbb{R}$).

Naturellement, l'échelle de $\underline{\beta}^T \underline{X}$ dans $\tilde{\mu}(\underline{\beta}^T \underline{X})$ peut être choisie de façon arbitraire : pour tout $c > 0$, $(\underline{\beta}, \mu_0(\cdot))$ et $(c\underline{\beta}, \mu_0(\cdot/c))$ nous conduisent à la même fonction de régression. Pour rendre le paramètre identifiable, on pose $\underline{\beta} = \mathbb{E}(\nabla\mu(\underline{X}))$. En effet, nous avons :

$$\mathbb{E}(\nabla\mu(\underline{X})) = \mathbb{E}(\nabla g^{-1}(\eta_0(\underline{\beta}^T \underline{X}))) = \mathbb{E}((g^{-1})'(\eta_0(\underline{\beta}^T \underline{X})))\underline{\beta} = c\underline{\beta}.$$

3.2.2 Méthode d'estimation : algorithme GSIM

On cherche à estimer $\underline{\beta}$ et $\tilde{\eta} = g(\tilde{\mu})$. Puisque $\underline{\beta} = \mathbb{E}[\nabla\mu(\underline{X})]$ et $\mu = g^{-1}(\eta)$, nous avons

$$\underline{\beta} = \mathbb{E}\left[(g^{-1})'(\eta(\underline{X}))\ \nabla\eta(\underline{X})\right].$$

L'idée que nous allons développer est d'estimer η et $\nabla\eta$ par la méthode des polynômes locaux à partir de la fonction de vraisemblance conditionnelle (3.1). A ce stade, nous n'utilisons pas la structure en indice simple de notre modèle. Alors on peut estimer $\underline{\beta}$ par la variance empirique des variables $(g^{-1})'(\hat{\eta}(\underline{X}_i))\widehat{\nabla\eta}(\underline{X}_i)$. Pour finir, on régresse Y_i sur $\hat{\underline{\beta}}^T\underline{X}_i$ par polynômes locaux pour obtenir $\widehat{\tilde{\eta}}$ et $\hat{\mu}(x) = g^{-1}(\widehat{\tilde{\eta}}(\hat{\underline{\beta}}^T x))$. Plus précisément la procédure GSIM pour estimer $\underline{\beta}$ et $\tilde{\eta}$ est décrite par l'algorithme suivant 3.1. Dans cet algorithme, les $\underline{e}_i$ sont les vecteurs de $\mathbb{R}^p$ tels que $\{\underline{e}_i\}_j = 1$ si $i = j$, 0 sinon, pour i et $j \in 1 \dots p$.

Algorithme 3.1. Algorithme GSIM - Estimation de β et η

Etape A - `Pour` $j = 1, \dots, n$, `calculer`

$$\hat{\eta}(\underline{X}_j) = \hat{a}_{(0,\dots,0)}(\underline{X}_j), \quad \widehat{\nabla\eta}(\underline{X}_j) = (\hat{a}_{\underline{e}_1}(\underline{X}_j) \dots, \hat{a}_{\underline{e}_p}(\underline{X}_j))^T,$$

`obtenus en maximisant`

$$\sum_{i=1}^{n} \mathcal{L}\left(\sum_{\underline{k}\in A_q} a_{\underline{k}}\ (\underline{X}_i - \underline{X}_j)^{\underline{k}}, Y_i\right) K_H^p(\underline{X}_i - \underline{X}_j), \tag{3.7}$$

`par rapport à` $a_{\underline{k}}$, $\underline{k} \in A_q$. `On pose`

$$\hat{\underline{\beta}} = \sum_{i=1}^{n} (g^{-1})'(\hat{\eta}(\underline{X}_i))\hat{\nabla}\eta(\underline{X}_i).$$

Etape B - `Déterminer` $\widehat{\eta}(x) = \hat{a}_0$ `en maximisant`

$$\sum_{i=1}^{n} \mathcal{L}\left(a_0 + a_1\hat{\underline{\beta}}^T(\underline{X}_i - x), Y_i\right) K_{h_B}^1\left(\hat{\underline{\beta}}^T(\underline{X}_i - x)\right),$$

`par rapport à` a_0 `and` a_1.

3.2.3 Problèmes de dimension, préselection des covariables et pénalisation

La maximisation du critère (3.7) peut être lue comme la recherche d'un maximum de vraisemblance pondérée, de poids $\left(K_H^p(\underline{X}_i - \underline{X}_j)\right)_i$ et de matrice de plan d'expérience particulière, dont le nombre de colonnes augmente avec q. En pratique, la maximisation est résolue par un algorithme de type IRLS. Or le maximum n'existe pas nécessairement.En particulier, dans le cadre des modèles de régression logistique, le maximum de vraisemblance ne peut jamais exister si le rang de la matrice de plan

d'expérience est égal à n. Dans le cas particulier des applications de biopuces, le rang de la matrice de plan d'expérience est, en pratique, égal à n (puisque le nombre de gènes est très grand devant le nombre d'individus). Pour remédier à ce problème, deux approches sont envisageables. La première consiste à faire une pré-sélection de gènes. La seconde consiste à introduire un terme de régularisation dans le critère (3.7), par exemple de type Ridge. Nous allons nous intéresser aux deux approches en vue d'une application à des données de biopuces.

Pour pré-sélectionner les gènes, on utilise la méthode de seuillage empirique bayésien (avec un *a priori* Laplace) introduite en section 2.4.2 appliquée ici à la statistique de test obtenue en calculant pour chaque gène la différence des moyennes entre les deux classes. Cette méthode risque de nous amener à sélectionner un peu trop de gènes, mais on préfère en conserver quand même un certain nombre pour que la réduction de dimension ait un intérêt.

La méthode nous permet de faire un seuillage des p statistiques de test pour ne garder que les p^* gènes les plus significatifs. Ce nombre p^* est adaptatif, et dépend donc de l'ensemble d'apprentissage choisi. Souvent, le nombre de gènes sélectionnés est encore trop grand. Dans ce cas, on donne en entrée de l'algorithme une valeur p_{max} qui représentera le nombre maximum de covariables à conserver.

De façon expérimentale (voir section 3.4.2.1), on observe des résultats d'autant plus mauvais que le nombre de gènes sélectionnés augmente. Pour le jeu de données Colon, par exemple, on en arrive à obtenir les meilleurs résultats pour un nombre de gènes égal à deux. Or, là le problème de réduction de dimension perd quand même de son intérêt et ne nous permet plus de mesurer la qualité de notre approche de compression.

En fait, le bruit induit par l'augmentation du nombre de gènes n'est pas compensé par la pondération qui n'a pas assez d'influence sur la variabilité introduite par le nombre de gènes. Pour obtenir plus de stabilité, il faut prendre une grande valeur de h. En effet, quand h augmente, on perd le caractère local, mais surtout le terme de biais augmente alors que le terme de variance diminue, on observera donc une stabilisation des résultats quand h augmentera.

On voudrait garder beaucoup de gènes pour pouvoir conserver plus d'information, tout en conservant un biais assez faible et en contrôlant la variance. Ceci peut-être obtenu en introduisant un terme de pénalité dans la procédure IRLS.

Reste à choisir la nature de cette pénalité. Les données de biopuces, en plus de souffrir du "fléau" de la dimension, posent aussi le problème de la multicolinéarité des régresseurs. Une pénalité de type ridge permettra de répondre à ce problème. De plus cette pénalité est relativement aisée à contrôler au niveau numérique. Dans la première étape de notre algorithme, celle qui va être pénalisée, on ne s'intéresse pas à l'estimation de la constante - qui est l'objectif de la deuxième étape - mais à l'estimation du gradient qui nous fournira la direction de projection. Il semble donc logique d'utiliser une pénalité de type Ridge qui porte sur toutes les composantes autres que la constante. Seifert et Gasser [40] introduisent une pénalité de type Ridge dans l'estimation du gradient dans le cas p=1 et q=1. Ils indiquent qu'une telle pénalité permet aussi de remédier aux problèmes de résolution liés en partie

à l'existence éventuelle (et même souhaitable en classification) de "clusters" ou regroupements d'individus, c'est-à-dire aussi de régions "creuses" ce qui est quasiment inévitable dès que le nombre de covariables commence à augmenter. Quand on fait de la classification, on s'attend justement à trouver des "groupes" d'individus. Ces problèmes de résolution interviennent au moment d'inverser la matrice $\underline{X}^T W_{IRLS} \underline{X}$ (où W désigne la matrice de poids utilisée dans l'étape IRLS). C'est pourquoi on peut penser qu'une telle pénalité serait toujours utile si on disposait d'un grand nombre d'individus.

L'étape A de l'algorithme GSIM pénalisé revient alors à maximiser le critère :

$$\sum_{i=1}^{n} \mathcal{L}\left(\sum_{\underline{k} \in A_q} a_{\underline{k}} \, (\underline{X}_i - \underline{X}_j)^{\underline{k}}, Y_i \right) K_H^p(\underline{X}_i - \underline{X}_j) - \frac{1}{2} \lambda {a_{\underline{k}}}^T \Sigma^2 a_{\underline{k}}, \tag{3.8}$$

où Σ^2 est la matrice diagonale de même dimension que $a_{\underline{k}}$ à laquelle on a mis un zéro en position $(1,1)$ et les variances empiriques de $\mathbf{X}$ sur le reste de la diagonale, et ceci pour ne pas pénaliser aussi la constante. λ est le paramètre de régularisation. $\mathbf{X}$ désigne ici la matrice $n \times p$ dont les colonnes sont les p covariables considérées Ainsi, le problème d'inversion disparaîtra, on calculera $\underline{X}^T W_{IRLS} \underline{X} + \lambda R$ au lieu de $\underline{X}^T W_{IRLS} \underline{X}$. On notera GSIM_λ la version pénalisée de la procédure GSIM et $\hat{\underline{\beta}}_\lambda$ l'estimateur de $\underline{\beta}$ qui en résulte.

L'inconvénient de l'introduction de cette pénalité est qu'il nécessite de choisir le paramètre de régularisation λ en plus des paramètres H et h_B déjà introduits dans la procédure GSIM initiale.

3.2.4 Implantation et choix des paramètres pour la procédure GSIM

Dans la première étape, il faut fixer l'ordre q de l'approximation polynomiale. En pratique, on choisit un ajustement linéaire (q=1) comme cela est fait dans la méthode rOPG proposée par Xia *et al.* [48]. Quelques essais ont été faits avec $q = 2$, mais les résultats ne s'avéraient pas meilleurs alors qu'on pouvait espérer, en augment l'ordre du développement, améliorer l'estimation du gradient comme indiqué dans [50] ; notons aussi qu'augmenter q accroît considérablement la complexité du problème (augmentation exponentielle de la taille de la matrice de design) et donc le temps de calcul. D'autre part, pour réduire le fléau de la dimension, nous utilisons dans l'étape A un noyau produit, ce qui conduit à choisir une matrice H diagonale. Les noyaux considérés sont gaussiens. Des noyaux de type Epanechnikov ont aussi été essayés et ont fourni des résultats en tous points comparables à ceux obtenus avec des noyaux gaussiens.

Pour l'instant, une éventuelle standardisation de la matrice $\mathbf{X}$ n'a pas été évoquée. Notons Σ^2 la matrice diagonale dont le i-ème terme diagonal est égal à la variance empirique associée à la i-ème covariable. Comme il est d'usage dans une

pénalité de type Ridge, on utilise la norme induite par Σ^2 pour les coefficients relevant du gradient. Cela revient à pénaliser le gradient fortement dans les directions les plus variables. La matrice de plan d'expérience standardisée est donnée par $\mathbf{X}_s = (\mathbf{X} - \mathbb{1}_n \mathbb{1}_n^T \mathbf{X}/n)\Sigma^{-1}$. Soit $\underline{\hat{\beta}}_s^{\lambda}$ l'estimateur correspondant à $\mathbf{X}_s$ avec une taille de fenêtre H_s et avec dans le terme de pénalité la norme induite par $\mathbb{R}_{1,p+1}$; on peut montrer que, pour $H_s = H\Sigma^{-1}$, on a $\underline{\hat{\beta}}^{\lambda} = \Sigma^{-1}\underline{\hat{\beta}}_s^{\lambda}$. Puisque GSIM est invariante par standardisation en colonnes, on effectuera les calculs avec la matrice standardisée. Cela nous permettra d'avoir dans le terme de pénalité une matrice R égale à l'identité avec un zéro substitué au 1 en position $(1,1)$. Si les covariables sont standardisées, il est alors naturel de choisir $H_s = h_A \mathrm{Id}_p$, ce qui réduit le nombre d'hyperparamètres à déterminer, ce qui est un gain considérable et motive cette standardisation des données.

La procédure nécessite le choix d'un paramètre de lissage à deux niveaux différents. Dans la première étape, on cherche à estimer η ainsi que son gradient et la taille de fenêtre h_A doit être optimale pour cet objectif. De plus pour GSIM_λ, on doit également déterminer l'hyperparamètre λ. Pour la procédure GSIM non pénalisée nous n'avons, dans la première étape, que le paramètre h_A à déterminer. On choisit alors l'hyperparamètre h_A par utilisation de la L-curve (*cf* Hansen et O'Leary [24]). Pour h_A appartenant à une grille donnée, on calcule l'opposé de la log-vraisemblance obtenue pour h_A comme une fonction du logarithme de h_A et on choisit la valeur de h_A qui correspond au "coude" de la fonction. On effectue une interpolation par BSpline et on en utilise la dérivée seconde pour repérer automatiquement ce coude.

En ce qui concerne la procédure GSIM_λ, on choisit une approche validation croisée simultanément en h_A et λ. Cette approche est coûteuse en temps de calcul, mais ce choix simultané de paramètres n'est pas classique et nous n'avons pas, à l'heure actuelle, trouvé de solution plus satisfaisante. Précisons ici la méthode que nous avons utilisée pour choisir nos hyperparamètres par validation croisée, et ce pour toutes les méthodes où une validation croisée est nécessaire (GSIM_λ et les méthodes auxquelles on la compare). On considère que notre population de n individus est divisée en un ensemble d'apprentissage et un ensemble de test. Pour un ensemble d'apprentissage et pour une valeur des paramètres donnée, on effectue une procédure « LeaVe One Out » (notée LVO). Cela consiste à retirer successivement chacun des n individus de l'ensemble d'apprentissage pour faire tourner l'algorithme, puis de faire une prédiction sur l'individu qui avait été initialement retiré et de comparer cette prédiction à la réalité. Ainsi pour une procédure LVO donnée, on a un nombre d'erreur compris entre 0 et n. Cela permet d'obtenir pour chaque valeur du paramètre (ou vecteur de paramètres) un nombre d'erreur de la procédure LVO. On choisit alors l'hyperparamètre ou la combinaison des hyperparamètres qui donne le plus faible nombre d'erreurs.

Lorsque nous faisons un choix d'hyperparamètre par validation croisée et qu'un prétraitement des données est nécessaire (ce qui peut enlever un certain nombre de covariables), nous effectuons le prétraitement avant de rentrer dans la boucle LVO

car nous avons le droit d'utiliser tout l'ensemble d'apprentissage pour choisir les valeurs optimales des paramètres. Cela est d'ailleurs préférable dans la mesure où les paramètres que nous recherchons doivent être optimaux pour cette matrice de données après prétraitement et non pour les differentes matrices que l'on peut avoir dans la procedure LVO, si l'on refait un prétraitement à chaque nouvel individu enlevé de l'ensemble d'apprentissage.

Mais la validation croisée présente un inconvénient majeur, celui du temps et du coût de calcul. En effet pour chaque valeur de h_A envisagée, on doit faire tourner n fois l'algorithme sur des problèmes de taille $(n-1) \times p$ pour calculer le taux d'erreur du LVO.

Dans la seconde étape, on cherche à estimer $\tilde{\eta}$ et h_B doit être optimal pour cette tâche. Pour ce choix plus standard, on utilisera la méthode de "plug-in" de Fan et Gijbels [18].

Notons enfin que la procédure GSIM$_\lambda$ est invariante par reparamétrisation utilisant la décomposition en valeurs singulières, permettant ainsi de réduire considérablement le temps de calcul. Cette reparamétrisation est détaillée dans Fort et Lambert-Lacroix [20].

3.2.5 Comparaison avec d'autres procédures

Nous allons tout dabord comparer notre méthode à des méthodes de discrimination "classiques", telles que la règle de discrimination linéaire diagonale (DLDA) et quadratique diagonale (DQDA) ; et la règle des k-plus proches voisins pour la métrique euclidienne classique (kNN), où le nombre de plus proches voisins est choisi par validation croisée. Dudoit *et al.* [15] conseillent l'utilisation de ces méthodes simples, notamment DLDA, pour la classification supervisée de biopuces.

D'un point de vue conceptuel, il est aussi particulièrement intéressant de comparer nos résultats avec ceux obtenus avec la procédure rOPG. En effet, la procédure GSIM est en quelque sorte une extension aux GLM de la méthode OPG proposée par Xia *et al.* [48].

La méthode OPG vise à estimer κ ($\kappa << p$) directions orthonormales engendrant l'espace estimé de *effective dimension reduction* (EDR). La procédure OPG avec $\kappa = 1$, est analogue à l'étape A de GSIM, avec un critère de moindres carrés à la place de la log-vraisemblance (*i.e.* ce qui, conceptuellement, revient à considérer que les observations sont gaussiennes). La direction est estimée comme le vecteur propre associé à la plus grande valeur propre de l'estimation empirique de la matrice $\mathbb{E}[\nabla\mu(\underline{X})\nabla\mu(\underline{X})^T]$.

Dans Xia *et al.* [48], cette procédure n'est pas appliquée à des problèmes en grande dimension. Quand $n \leq p$, pour estimer les différents gradients, la méthode revient à projeter selon une certaine géométrie un vecteur de taille n sur un espace engendré par les colonnes d'une matrice de plan d'expérience de rang n. Ainsi ces projections ne dépendent plus de la fenêtre de lissage. Cela équivaut à considérer le modèle en indice simple $\mu(\underline{X}) = \alpha + \underline{\beta}^T\underline{X}$, et en estimer la direction $\underline{\beta}$ par

maxixum de vraisemblance. Cette approche ne semble pas très intéressante dans la situation considérée ici. C'est pourquoi, dans le même esprit que pour GSIM, nous proposons d'adapter la méthode OPG en introduisant une pénalité de type Ridge dans le critère. Par ailleurs, dans les programmes proposés par les auteurs, il y avait déjà un terme de stabilisation numérique que l'on peut voir comme une pénalité Ridge avec un paramètre de régularisation fixé à 0.0001. Notre approche pour OPG consiste à faire de cette constante un paramètre λ.

Dans un but de classification, nous avons appliqué, après l'étape correspondant à l'estimation de la direction de projection $\underline{\beta}$, la même étape B que l'algorithme GSIM correspondant à l'estimation de $\widehat{\eta}$. Seule la méthode d'estimation de la direction de projection diffère par rapport à GSIM, permettant ainsi de mesurer l'intérêt de la prise en compte de la relation entre espérance et variance dans les GLM. Notons que comme pour GSIM, la fenêtre correspondant à l'étape A et λ sont choisis simultanément par validation croisée ; la fenêtre associée à l'étape B est déterminée par plug-in. Par ailleurs Xia *et al.* [48] définissent une version raffinée (procédure rOPG) de la manière suivante. On rajoute une étape A' après l'étape A par l'itération jusqu'à stabilisation des instructions suivantes. Poser $\hat{\underline{\beta}}^{(0)} = \hat{\underline{\beta}}$ et pour la k-ième itération, $\hat{\underline{\beta}}^{(k)}$ est obtenu comme à l'étape A mais avec les poids $K_h^1(\hat{\underline{\beta}}^{(k-1)T}(\underline{X}_i - \underline{X}_j))$ dans (3.7) à la place de $K_H^p(\underline{X}_i - \underline{X}_j)$. En effet, dans les modèles linéaires généralisés en indice simple, la fonction η est constante dans les directions orthogonales à la direction $\underline{\beta}$. On peut donc prendre une fenêtre plus large dans ces directions, ce que fait le noyau $K_h^1(\underline{\beta}^T(\cdot - \underline{X}_j))$.

Notons que cette règle de discrimination est différente de ce qui est fait par Antoniadis *et al.* [6] puisque la réduction de dimension se fait par la procédure rOPG qui se trouve être une version "simplifiée" de MAVE. Nous avons observé que les résultats sont comparables à ceux obtenus avec la règle d'Antoniadis *et al.* [6] ; nous avons choisi de présenter les résultats de rOPG parce que cette méthode est plus directement reliée à l'étape A de GSIM.

Nous signalons également que pour une des études de jeux de données réels, les résultats ont été obtenus pour la méthode rOPG sans choix d'un paramètre de régularisation par validation croisée. Dans ce cas, nous avons utilisé directement les programmes des auteurs ce qui revient aussi à fixer $\lambda = 0.0001$, cela est dû à la récente implantation de rOPG avec un choix d'un paramètre λ, les changements n'ont pas pu être faits pour tous les jeux de données mais les résultats devraient rester comparables. Pour les jeux de données concernés, nous signalerons alors que $\lambda = 0.0001$.

En ce qui concerne l'implantation, les méthodes OPG et rOPG sont stables par reparamétrisation utilisant la décomposition en valeurs singulières. Seule la méthode OPG est stable à la standardisation en colonne, cependant nous choisissons de standardiser la matrice pour se placer dans les mêmes conditions que GSIM.

La méthode dans le cas asymptotique : étude théorique

Avant de nous intéresser à l'application aux données de biopuces, nous allons, dans un premier temps, étudier les qualités théoriques de convergence de l'algorithme dans le cas standard c'est-à-dire quand $n >> p$. Dans ces conditions, on considère la version non pénalisée de la méthode, GSIM. Nous allons démontrer la consistance de l'estimateur $\underline{\hat{\beta}}$ obtenu sous certaines hypothèses.

Nous allons commencer par introduire quelques notations et imposer des conditions de régularité. Soit $l_i(u,v) = (\partial^i/\partial u^i)\mathcal{L}(g^{-1}(u),v)$, l_i est linéaire en v pour u fixé et de plus :

$$l_1(\eta(\underline{x}),\mu(\underline{x})) = 0$$

$$l_2(\eta(\underline{x}),\mu(\underline{x})) = -\rho(\underline{x}),$$

où $\rho(\underline{x}) = \{g'(\mu(\underline{x}))^2 V(\mu(\underline{x}))\}^{-1}$.

Dans un souci de clarté, le paramètre h_A introduit dans l'étape A de l'algorithme $GSIM$ sera, dans cette section, simplement noté h.

On suppose que le vecteur $\underline{X}$ des covariables a une densité f à support compact $S_{\underline{X}}$. On suppose de plus que les conditions suivantes sont vérifiées :

Hypothèses GSIM

1. La fonction $l_2(u,v) < 0$ pour $u \in \mathbb{R}$ et v prenant les valeurs possibles de la variable réponse.
2. La fonction $l_1(u,v)$ est bornée, la fonction $(g^{-1})''$ est bornée.
3. Les fonctions $f(\cdot)$, $D^{\underline{k}}\eta(\cdot)$, $|\underline{k}| = q+1$, $D^{\underline{k}}f(\cdot)$ et $D^{\underline{k}}\rho(\cdot)$ pour $|\underline{k}| = 1$, sont uniformément continues pour $\underline{x} \in S_{\underline{X}}$.
4. $\rho(\underline{x}) \neq 0$, $f(\underline{x}) \neq 0$, $\mathrm{Var}(Y|\underline{X} = \underline{x}) \neq 0$, et $g'(\mu(\underline{x})) \neq 0$, pour $\underline{x} \in S_{\underline{X}}$. On suppose aussi
$$\inf_{\underline{x} \in S_{\underline{X}}} (\rho(\underline{x}) f(\underline{x})) > 0.$$
5. Le noyau K^p est une densité de probabilité à support compact S_{K^p}. On prend une matrice de largeur de fenêtre $H = h\mathrm{Id}_p$. On suppose que l'on a $h = cn^{-\alpha}$, où c est une constante, et que nh^{p+2} tend vers l'infini quand n tend vers l'infini. Cette dernière condition nous donne la contrainte $\alpha < 1/(p+2)$.

6. On suppose qu'il existe $\delta > 0$, tel que :

$$\frac{1-2\delta}{2q+p+2} < \alpha < \frac{1-2\delta}{p+2} \tag{3.9}$$

Remarque. Pour cette dernière condition, cela revient à prendre $h = O(n^{-\alpha})$, avec α appartenant à un certain intervalle. Cette condition n'est pas vraiment contraignante, et permet de choisir une largeur de fenêtre h convenable.

Théorème 3.1. *Consistance de l'estimateur $\hat{\underline{\beta}}$*
Sous les conditions qui précèdent, $\hat{\underline{\beta}}$ est un estimateur consistant de $\underline{\beta}$ c'est-à-dire que l'on a :

$$\forall \varepsilon > 0, \quad \lim_{n\to\infty} \mathbb{P}(|\hat{\underline{\beta}} - \underline{\beta}| > \varepsilon) = 0$$

PREUVE — Pour obtenir des propriétés sur l'estimateur $\hat{\underline{\beta}}$, on va tout d'abord s'intéresser aux propriétés des estimateurs de $D^{\underline{k}}\eta(\underline{x})$.

La première partie de la preuve du théorème peut être établie en utilisant des arguments similaires à ceux du cas univarié (voir Fan *et al.* [19] et Fan et Gijbels [18]).

On considère l'estimateur normalisé $\hat{\underline{a}}^*(\underline{x})$, il s'agit d'un vecteur qui a pour dimension le cardinal de A_q et dont la composante $\underline{k} \in A_q$ est donnée par :

$$c_n^{-1} h^{|\underline{k}|} [\hat{a}_{\underline{k}}(\underline{x}) - D^{\underline{k}}\eta(\underline{x})/\underline{k}!],$$

où $c_n = (nh^p)^{-1/2}$. Il est à noter que pour simplifier l'écriture, ce ne sont pas les notations usuelles qui sont utilisées ici pour l'indexation des vecteurs. Les indices 1 à cardinal de A_q sont ré-indexés en $\underline{k} \in A_q$. Quand cela est nécessaire, nous utilisons les mêmes notations d'indexation pour les matrices.

On voit facilement que $\hat{\underline{a}}^*(\underline{x})$ maximise l'expression :

$$\sum_{i=1}^{n} \mathcal{L}\left[g^{-1}\left(\bar{\eta}(\underline{x}, \underline{X}_i) + c_n \underline{a}^{*T} R(\underline{x}, \underline{X}_i)\right), Y_i\right] K^p(H^{-1}(\underline{X}_i - \underline{x})).$$

comme fonction de $\underline{a}^*$, où

$$\bar{\eta}(\underline{x}, \underline{X}_i) = \eta(\underline{x}) + \sum_{\underline{k} \in A_q \setminus A_0} D^{\underline{k}}\eta(\underline{x})(\underline{X}_i - \underline{x})^{\underline{k}}/\underline{k}!$$

et

$$R(\underline{x}, \underline{X}_i) = \left\{(H^{-1}(\underline{X}_i - \underline{x}))^{\underline{k}}\right\}_{\underline{k} \in A_q}.$$

De manière équivalente, $\hat{\underline{a}}^*(\underline{x})$ maximise

$$\mathcal{L}_n(\underline{a}^*) = \sum_{i=1}^{n} \left(\mathcal{L}\left[g^{-1}\left(\bar{\eta}(\underline{x}, \underline{X}_i) + c_n \underline{a}^{*T} R(\underline{x}, \underline{X}_i)\right), Y_i\right] - \mathcal{L}\left[g^{-1}\left(\bar{\eta}(\underline{x}, \underline{X}_i)\right), Y_i\right]\right) K^p(H^{-1}(\underline{X}_i - \underline{x})).$$

La condition (1) implique que la fonction $\mathcal{L}_n$ est concave en $\underline{a}^*$. Un développement de Taylor de $\mathcal{L}([g^{-1}(\cdot), Y_i]$ nous donne

$$\mathcal{L}_n(\underline{a}^*) = W_{\underline{x}}^T \underline{a}^* + \frac{1}{2}\underline{a}^{*T} A_n \underline{a}^* + \frac{c_n^3}{6}\sum_{i=1}^{n} q_3(\eta_i, Y_i)(\underline{a}^{*T} R(\underline{x}, \underline{X}_i))^3 K^p(H^{-1}(\underline{X}_i - \underline{x})), \tag{3.10}$$

où η_i est compris entre $\bar{\eta}(\underline{x}, \underline{X}_i)$ et $\bar{\eta}(\underline{x}, \underline{X}_i) + c_n \underline{a}^{*T} R(\underline{x}, \underline{X}_i)$,

$$W_{\underline{x}} = c_n \sum_{i=1}^{n} l_1\left[\bar{\eta}(\underline{x}, \underline{X}_i), Y_i\right] R(\underline{x}, \underline{X}_i) K^p(H^{-1}(\underline{X}_i - \underline{x})),$$

et

$$A_n = (c_n)^2 \sum_{i=1}^{n} l_2\left[\bar{\eta}(\underline{x}, \underline{X}_i), Y_i\right] R(\underline{x}, \underline{X}_i)^T R(\underline{x}, \underline{X}_i) K^p(H^{-1}(\underline{X}_i - \underline{x})).$$

Soit $\mathcal{D}$ l'ensemble défini par $\{\underline{u}; \underline{x} + H\underline{u} \in S_{\underline{X}}\} \cap S_{K^p}$. On définit aussi

$$\Sigma_{\underline{x}} = \left\{\rho(\underline{x}) f(\underline{x}) \nu_{\underline{l}+\underline{k}}\right\}_{\underline{l}\in A_q, \underline{k}\in A_q}.$$

où $\nu_{\underline{l}} = \int_{\mathcal{D}} \underline{u}^{\underline{l}} K^p(\underline{u}) d\underline{u}$. On a le lemme suivant :

Lemme 3.2. *Sous les conditions qui précédent,*

$$\mathcal{L}_n(\underline{a}^*) = W_{\underline{x}}^T \underline{a}^* - \frac{1}{2}\underline{a}^{*T} \Sigma_{\underline{x}} \underline{a}^* + o_P(1), \tag{3.11}$$

uniformément en $\underline{x} \in S_{\underline{X}}$.

PREUVE — On commence par prouver que $A_n = -\Sigma_{\underline{x}} + o_P(1)$. Cela peut être montré en utilisant le fait que, pour $\underline{l}$ et $\underline{k}$ dans A_q,

$$(A_n)_{\underline{l},\underline{k}} = (\mathbb{E}A_n)_{\underline{l},\underline{k}} + O_P\left[\left\{\mathrm{Var}\,(A_n)_{\underline{l},\underline{k}}\right\}^{\frac{1}{2}}\right].$$

Dans l'expression ci-dessus, la moyenne est égale à

$$(\mathbb{E}A_n)_{\underline{l},\underline{k}} = n c_n^2 \int_{S_{\underline{X}}} l_2([\bar{\eta}(\underline{x}, \underline{X}_1), \mu(\underline{X}_1)]\, f(\underline{X}_1) K^p(H^{-1}(\underline{X}_1 - \underline{x})) d\underline{X}_1.$$

On effectue le changement de variable $\underline{u} = H^{-1}(\underline{X}_1 - \underline{x})$ et on obtient :

$$(\mathbb{E}A_n)_{\underline{l},\underline{k}} = n c_n^2 h^p \int_{\mathcal{D}} l_2\left[\bar{\eta}(\underline{x}, \underline{x} + H\underline{u}), \mu(\underline{x} + H\underline{u})\right] f(\underline{x} + H\underline{u}) K^p(\underline{u}) \underline{u}^{\underline{l}+\underline{k}} d\underline{u}.$$

On a $c_n = (nh^p)^{-1/2}$ donc $n c_n^2 h^p = 1$ d'où :

$$(\mathbb{E}A_n)_{\underline{l},\underline{k}} = \int_{\mathcal{D}} l_2\left[\bar{\eta}(\underline{x}, \underline{x} + H\underline{u}), \mu(\underline{x} + H\underline{u})\right] f(\underline{x} + H\underline{u}) K^p(\underline{u}) \underline{u}^{\underline{l}+\underline{k}} d\underline{u}.$$

Comme le support de K^p est compact, on a :

$$\bar{\eta}(\underline{x}, \underline{x} + H\underline{u}) = \eta(\underline{x} + H\underline{u}) - \sum_{|\underline{k}|=q+1} D^{\underline{k}}\eta(\underline{x})(H\underline{u})^{\underline{k}}/\underline{k}! + o(h^{q+1}),$$

uniformément en $\underline{x}$.En utilisant un développement limité de l_2 au voisinage de $(\eta(\underline{x}+H\underline{u}), \mu(\underline{x}+H\underline{u}))$, on obtient

$$(\mathbb{E}A_n)_{\underline{l},\underline{k}} = -\int_{\mathcal{D}} \rho(\underline{x}+H\underline{u}) f(\underline{x}+H\underline{u}) K^p(\underline{u}) \underline{u}^{\underline{l}+\underline{k}} d\underline{u} + o(h^q).$$

On effectue maintenant un développement de Taylor sur ρf au voisinage de $\underline{x}$:

$$(\rho f)(\underline{x}+H\underline{u}) = (\rho f)(\underline{x}) + \sum_{\underline{k}\in A_1} D^{\underline{k}}(\rho f)(\underline{x}) \frac{(H\underline{u})^{\underline{k}}}{\underline{k}!} + o(h). \tag{3.12}$$

Ce qui permet d'avoir :

$$\begin{aligned} (\mathbb{E}A_n)_{\underline{l},\underline{k}} &= -\rho(\underline{x}) f(\underline{x}) \int_{\mathcal{D}} K^p(\underline{u}) \underline{u}^{\underline{l}+\underline{k}} d\underline{u} + O(h), \\ \textit{i.e.} \quad (\mathbb{E}A_n)_{\underline{l},\underline{k}} &= -\rho(\underline{x}) f(\underline{x}) \nu_{\underline{l}+\underline{k}} + O(h) = -\Sigma_{\underline{x}} + O(h), \end{aligned}$$

uniformément en $\underline{x}$. Des arguments similaires permettent de montrer que $\mathrm{Var}\,(\{A_n\}_{\underline{l},\underline{k}}) = O((nh^p)^{-1})$. On a donc

$$A_n = -\Sigma_{\underline{x}} + O(h) + O_P((nh^p)^{-1}).$$

En utilisant la condition $nh^{p+2} \to \infty$, on obtient le résultat intermédiaire

$$A_n = -\Sigma_{\underline{x}} + o_P(1).$$

Grâce au lemme d'approximation quadratique (voir [32]), l'équation 3.11 est vraie uniformément en $\underline{a}^* \in C$ pour un ensemble C compact et on peut appliquer le lemme A.1 de [10] qui donne

$$\sup_{\underline{x}\in S_{\underline{X}}} \left|\hat{\underline{a}}^* - \Sigma_{\underline{x}}^{-1} W_{\underline{x}}\right| \xrightarrow[n\to\infty]{P} 0.$$

$\otimes$

Maintenant nous allons calculer les deux premiers moments de $W_{\underline{x}}$. Plus précisément, nous avons le lemme suivant.

Lemme 3.3. *Sous les conditions qui précédent,*

$$\mathbb{E}\left(\{W_{\underline{x}}\}_{\underline{l}}\right) = \rho(\underline{x}) f(\underline{x}) \left[\left(nh^{2q+2+p}\right)^{\frac{1}{2}} \sum_{|\underline{k}|=q+1} \frac{D^{\underline{k}}\eta(\underline{x})}{\underline{k}!} \nu_{\underline{l}+\underline{k}} \right.$$

$$\left. + \left(nh^{2q+4+p}\right)^{\frac{1}{2}} \left(\sum_{|\underline{k}|=q+1} \frac{D^{\underline{k}}\eta(\underline{x})}{\underline{k}!} \sum_{|\underline{m}|=1} \frac{D^{\underline{m}}(\rho f)(\underline{x})}{(\rho f)(\underline{x})} \nu_{\underline{l}+\underline{k}+\underline{m}} + \sum_{|\underline{k}|=q+2} \frac{D^{\underline{k}}\eta(\underline{x})}{\underline{k}!} \nu_{\underline{l}+\underline{k}} \right) \right]$$

$$+ o\left(\left[nh^{2p+4+d}\right]^{\frac{1}{2}}\right)$$

$$Cov\left(\{W_{\underline{x}}\}_{\underline{l},\underline{k}}\right) = \frac{f(\underline{x})\, Var\left(Y/\underline{X}=\underline{x}\right)}{\left[V\{\mu(\underline{x})\} g'\{\mu(\underline{x})\}\right]^2} \int_{\mathcal{D}} K^p(\underline{u})^2 \underline{u}^{\underline{l}+\underline{k}} d\underline{u} + o(1).$$

Remarque. Dans le lemme précédent, on a conservé deux termes pour exprimer l'espérance de $\{W_{\underline{x}}\}$, un terme en $O(nh^{2q+2+p})$ et le deuxième terme en $O(nh^{2q+4+p})$. Le deuxième terme est pourtant négligeable devant le premier ; en fait, des problèmes peuvent subvenir selon l'ordre q du développement. Si on utilise un noyau symétrique (ce qui est souvent le cas), on peut obtenir des moments $\nu_{\underline{l}+\underline{k}}$ qui sont nuls. Ainsi les $\nu_{\underline{l}+\underline{k}}$ seront nuls si on prend un ordre q pair et on aura alors besoin du deuxième terme. Inversément, avec q impair, c'est le deuxième terme de l'expression qui va s'annuler. Dans la suite de la démonstration du théorème, on aura besoin de l'ordre de grandeur de cette espérance pour effectuer ensuite une majoration. On pourra donc utiliser un $O(nh^{2q+2+p})$.

PREUVE — Par un développement de Taylor,

$$\begin{aligned} l_1\left[\bar{\eta}(\underline{x}, \underline{x}+H\underline{u}), \mu(\underline{x}+H\underline{u})\right] &= l_1\left(\eta(\underline{x}+H\underline{u}), \mu(\underline{x}+H\underline{u})\right) \\ &+ \left[\bar{\eta}(\underline{x}, \underline{x}+H\underline{u}) - \eta(\underline{x}+H\underline{u})\right] l_2\left(\eta(\underline{x}+H\underline{u}), \mu(\underline{x}+H\underline{u})\right) \\ &+ O\left(\left[\bar{\eta}(\underline{x}, \underline{x}+H\underline{u}) - \eta(\underline{x}+H\underline{u})\right]^2\right) \end{aligned}$$

Or on a $l_1\left(\eta(\underline{x}+H\underline{u}), \mu(\underline{x}+H\underline{u})\right) = 0$, $l_2\left(\eta(\underline{x}+H\underline{u}), \mu(\underline{x}+H\underline{u})\right) = -\rho(\underline{x}+H\underline{u})$ et de plus

$$\bar{\eta}(\underline{x}, \underline{x}+H\underline{u}) - \eta(\underline{x}+H\underline{u}) = -\sum_{q+1 \le |\underline{k}| \le q+2} D^{\underline{k}}\eta(\underline{x})(H\underline{u})^{\underline{k}}/\underline{k}! + O(h^{q+2}).$$

D'où

$$l_1\left[\bar{\eta}(\underline{x}, \underline{x}+H\underline{u}), \mu(\underline{x}+H\underline{u})\right] = \rho(\underline{x}+H\underline{u}) \sum_{q+1 \le |\underline{k}| \le q+2} \frac{D^{\underline{k}}\eta(\underline{x})}{\underline{k}!} [H\underline{u}]^{\underline{k}} + o(h^{q+2}),$$

et on obtient, en faisant le même changement de variable que dans la démonstration du lemme précédent

$$\mathbb{E}\left(\{W_{\underline{x}}\}_{\underline{l}}\right) = c_n^{-1}\left[\sum_{q+1\leq|\underline{k}|\leq q+2} h^{|\underline{k}|}\frac{D^{\underline{k}}\eta(\underline{x})}{\underline{k}!}\int_{\mathcal{D}} f(\underline{x}+H\underline{u})\rho(\underline{x}+H\underline{u})\underline{u}^{\underline{l}+\underline{k}}K^p(\underline{u})d\underline{u}+o(h^{q+2})\right].$$

On effectue alors un développement limité de la fonction ρf au voisinage de $\underline{x}$ (voir équation 3.12) :

$$\begin{aligned}\mathbb{E}\left(\{W_{\underline{x}}\}_{\underline{l}}\right) &= c_n^{-1}\left[(\rho f)(\underline{x})\sum_{q+1\leq|\underline{k}|\leq q+2} h^{|\underline{k}|}\frac{D^{\underline{k}}\eta(\underline{x})}{\underline{k}!}\nu_{\underline{l}+\underline{k}}\right.\\ &\quad\left.+\sum_{q+1\leq|\underline{k}|\leq q+2} h^{|\underline{k}|+1}\frac{D^{\underline{k}}\eta(\underline{x})}{\underline{k}!}\sum_{|\underline{m}|=1} D^{\underline{m}}(\rho f)(\underline{x})\nu_{\underline{l}+\underline{k}+\underline{m}}\right]+o(c_n^{-1}h^{q+2}),\end{aligned}$$

On a donc

$$\begin{aligned}\mathbb{E}\left(\{W_{\underline{x}}\}_{\underline{l}}\right) &= c_n^{-1}(\rho f)(\underline{x})\left[h^{q+1}\sum_{|\underline{k}|=q+1}\frac{D^{\underline{k}}\eta(\underline{x})}{\underline{k}!}\nu_{\underline{l}+\underline{k}} + h^{q+2}\sum_{|\underline{k}|=q+2}\frac{D^{\underline{k}}\eta(\underline{x})}{\underline{k}!}\nu_{\underline{l}+\underline{k}}\right.\\ &\quad\left.+h^{q+2}\sum_{|\underline{k}|=q+1}\frac{D^{\underline{k}}\eta(\underline{x})}{\underline{k}!}\sum_{|\underline{m}|=1}\frac{D^{\underline{m}}(\rho f)(\underline{x})}{(\rho f)(\underline{x})}\nu_{\underline{l}+\underline{k}+\underline{m}}\right] + o(c_n^{-1}h^{q+2}).\end{aligned}$$

Or $c_n^{-1}h^{q+1} = (nh^{2q+2+p})^{\frac{1}{2}}$ et $c_n^{-1}h^{q+2} = (nh^{2q+4+p})^{\frac{1}{2}}$ ce qui nous conduit directement au premier résultat.

La covariance entre la $\underline{l}^{ème}$ et la $\underline{k}^{ème}$ composante de $W_{\underline{x}}$ est

$$\mathbb{E}(\{W_{\underline{x}}\}_{\underline{l}}\{W_{\underline{x}}\}_{\underline{k}}) + O(h^{2q+2+p}).$$

On a

$$\mathbb{E}(\{W_{\underline{x}}\}_{\underline{l}}\{W_{\underline{x}}\}_{\underline{k}}) = nc_n^2\mathbb{E}\left(l_1^2\left[\bar{\eta}(\underline{x},\underline{X}_1),Y_1\right]R(\underline{x},\underline{X}_1)_{\underline{l}}R(\underline{x},\underline{X}_1)_{\underline{k}}\left[K^p\left(H^{-1}(\underline{X}_1-\underline{x})\right)\right]^2\right).$$

Or $\bar{\eta}(\underline{x},\underline{X}_1) = \eta(\underline{x}) + O(h^q)$, on fait une développement limité de la fonction l_1 selon la première variable au voisinage de $\eta(\underline{x})$:

$$l_1\left[\bar{\eta}(\underline{x},\underline{X}_1),Y_1\right] = l_1\left[\eta(\underline{x}),Y_1\right] + l_1'\left[\eta(\underline{x}),Y_1\right]O(h^q) = l_1\left[\eta(\underline{x}),Y_1\right] + O(h^q).$$

d'où

$$l_1^2\left[\bar{\eta}(\underline{x},\underline{X}_1),Y_1\right] = l_1^2\left[\eta(\underline{x}),Y_1\right] + O(h^q).$$

On a alors, sachant aussi que $nc_n^2 = h^{-p}$

$$\begin{aligned}\mathbb{E}(\{W_{\underline{x}}\}_{\underline{l}}\{W_{\underline{x}}\}_{\underline{k}}) &= h^{-p}\mathbb{E}\left[l_1^2\left(\eta(\underline{x}),Y_1\right)R(\underline{x},\underline{X}_1)_{\underline{l}}R(\underline{x},\underline{X}_1)_{\underline{k}}\left[K^p\left(H^{-1}(\underline{X}_1-\underline{x})\right)\right]^2\right]\\ &+ h^{-p}O(h^q)\mathbb{E}\left[R(\underline{x},\underline{X}_1)_{\underline{l}}R(\underline{x},\underline{X}_1)_{\underline{k}}\left[K^p\left(H^{-1}(\underline{X}_1-\underline{x})\right)\right]^2\right].\end{aligned}$$

Intéressons-nous au deuxième terme du membre de droite de cette équation que nous noterons T_2.

$$T_2 = h^{-p}O(h^q)\int_{S_{\underline{X}}} R(\underline{x},\underline{X}_1)_{\underline{l}}R(\underline{x},\underline{X}_1)_{\underline{k}}\left[K^p\left(H^{-1}(\underline{X}_1-\underline{x})\right)\right]^2 f(\underline{X}_1)d\underline{X}_1.$$

On fait le changement de variable $\underline{u} = H^{-1}(\underline{X}_1 - \underline{x})$ et on obtient :

$$\begin{aligned}T_2 &= O(h^q)\int_{\mathcal{D}} \underline{u}^{\underline{k}+\underline{l}}\left[K^p(\underline{u})\right]^2 f(\underline{x}+H\underline{u})d\underline{u}\\ &= O(h^q) = o(1).\end{aligned}$$

D'où

$$\mathbb{E}(\{W_{\underline{x}}\}_{\underline{l}}\{W_{\underline{x}}\}_{\underline{k}}) = h^{-p}\mathbb{E}\left(l_1^2\left(\eta(\underline{x}),Y_1\right)R(\underline{x},\underline{X}_1)_{\underline{l}}R(\underline{x},\underline{X}_1)_{\underline{k}}\left[K^p\left(H^{-1}(\underline{X}_1-\underline{x})\right)\right]^2\right)+O(h^q).$$

On utilise ensuite la définition de l_1, on fait un développement limité de ρf au voisinage de $\underline{x}$, et avec $\rho(\underline{x}) = \{g'(\mu(\underline{x}))^2 V(\mu(\underline{x}))\}^{-1}$, on arrive au deuxième résultat du lemme.

$\otimes$

Maintenant que les propriétés de $\hat{\underline{a}}^*$ ont été étudiées, on revient à la démonstration du théorème sur la consistance de l'estimateur $\hat{\underline{\beta}}$. On s'intéresse donc au terme $\hat{\underline{\beta}}-\underline{\beta}$. Par définition on a :

$$\underline{\beta} = \mathbb{E}\left[(g^{-1})'(\eta(\underline{X}))\nabla\eta(\underline{X})\right]$$

$$\widehat{\underline{\beta}} = \frac{1}{n}\sum_{i=1}^{n}(g^{-1})'(\hat{\eta}(\underline{X}_i))\widehat{\nabla\eta}(\underline{X}_i)$$

On a donc :

$$\hat{\underline{\beta}} - \underline{\beta} = \frac{1}{n}\sum_{i=1}^{n}\left[(g^{-1})'(\hat{\eta}(\underline{X}_i))\widehat{\nabla\eta}(\underline{X}_i) - \mathbb{E}\left[(g^{-1})'(\eta(\underline{X}))\nabla\eta(\underline{X})\right]\right].$$

On cherche alors à faire apparaître le terme $(g^{-1})'(\eta(\underline{X}_i))\nabla\eta(\underline{X}_i)$, on a

$$\begin{aligned}\hat{\underline{\beta}} - \underline{\beta} &= \frac{1}{n}\sum_{i=1}^{n}\left[(g^{-1})'(\eta(\underline{X}_i))\nabla\eta(\underline{X}_i) - \mathbb{E}\left[(g^{-1})'(\eta(\underline{X}))\nabla\eta(\underline{X})\right]\right]\\ &+ \frac{1}{n}\sum_{i=1}^{n}\left[(g^{-1})'(\hat{\eta}(\underline{X}_i))\widehat{\nabla\eta}(\underline{X}_i) - (g^{-1})'(\eta(\underline{X}_i))\nabla\eta(\underline{X}_i)\right].\end{aligned}$$

Le théorème central limite donne

$$\frac{1}{n}\sum_{i=1}^{n}\left[(g^{-1})'(\eta(\underline{X}_i))\nabla\eta(\underline{X}_i) - \mathbb{E}\left[(g^{-1})'(\eta(\underline{X}))\nabla\eta(\underline{X})\right]\right] = o_P(1).$$

On s'intéresse donc au deuxième terme $(g^{-1})'(\hat{\eta}(\underline{X}_i))\widehat{\nabla\eta}(\underline{X}_i) - (g^{-1})'(\eta(\underline{X}_i))\nabla\eta(\underline{X}_i)$. On fait un développement limité de $(g^{-1})'((\hat{\eta})(\underline{x}))$ au voisinage de $\eta(\underline{x})$

$$(g^{-1})'(\hat{\eta}(\underline{x})) = (g^{-1})'(\eta(\underline{x})) + (g^{-1})''(\eta(\underline{x}))(\hat{\eta}(\underline{x}) - \eta(\underline{x})) + O([\hat{\eta}(\underline{x}) - \eta(\underline{x})]^2).$$

On définit

$$\varepsilon_n(\underline{x}) = (g^{-1})''(\eta(\underline{x}))\nabla\eta(\underline{x})(\hat{\eta}(\underline{x}) - \eta(\underline{x})) + (g^{-1})'(\eta(\underline{x}))(\widehat{\nabla\eta}(\underline{x}) - \nabla\eta(\underline{x}))$$

et

$$r_n(\underline{x}) = (g^{-1})'(\hat{\eta}(\underline{x}))\widehat{\nabla\eta}(\underline{x}) - (g^{-1})'(\eta(\underline{x}))\nabla\eta(\underline{x}) - \varepsilon_n(\underline{x}).$$

Alors

$$r_n(\underline{x}) \;=\; (g^{-1})''(\eta(\underline{x}))(\hat{\eta}(\underline{x}) - \eta(\underline{x}))(\widehat{\nabla\eta}(\underline{x}) - \nabla\eta(\underline{x})) + \widehat{\nabla\eta}(\underline{x})O([\hat{\eta}(\underline{x}) - \eta(\underline{x})]^2).$$

L'erreur d'estimation devient donc

$$\underline{\hat{\beta}} - \underline{\beta} = \frac{1}{n}\sum_{i=1}^{n}\left[\varepsilon_n(\underline{X}_i) + r_n(\underline{X}_i)\right] + o_P(1)$$

Pour avoir le résultat, il suffit donc de montrer que

$$\frac{1}{n}\sum_{i=1}^{n} r_n(\underline{X}_i) = o_P(1) \;\;, \qquad \frac{1}{n}\sum_{i=1}^{n}\varepsilon_n(\underline{X}_i) = o_P(1) \tag{3.13}$$

La première condition (terme en $r_n(\underline{x})$ de 3.13) sera vérifiée si on a la condition :

$$\sup_{\underline{x}\in S_{\underline{X}}} |r_n(\underline{x})| = o_P(1).$$

Or $\widehat{\eta}(\underline{x}) = \widehat{D^{\underline{k}}\eta}(\underline{x})$ avec $|\underline{k}| = 0$ et $\widehat{\nabla\eta}(\underline{x}) = \widehat{D^{\underline{k}}\eta}(\underline{x})$ avec $|\underline{k}| = 1$. Il suffit donc d'avoir la condition suivante :

$$\sup_{\underline{x}\in S_{\underline{X}}} |(\hat{\eta}(\underline{x}) - \eta(\underline{x}))(\widehat{D^{\underline{k}}\eta}(\underline{x}) - D^{\underline{k}}\eta(\underline{x}))| = o_P(1), \text{ pour } |\underline{k}| = 0 \text{ et } |\underline{k}| = 1 \tag{3.14}$$

à condition qu'on puisse admettre que $\widehat{\nabla\eta}(\underline{x})$ est borné pour n grand, ce qui sera assuré dans la suite car on sera amenés à montrer que $\sup_{\underline{x}\in S_{\underline{X}}} |\nabla\eta(\underline{x}) - \widehat{\nabla\eta}(\underline{x})| = o_P(1)$.

Pour montrer la condition (3.14), montrons que, pour $|\underline{k}| = 0$ ou 1, on a

$$\forall \varepsilon > 0, \quad \sup_{\underline{x} \in S_{\underline{X}}} |\widehat{D^{\underline{k}}\eta}(\underline{x}) - D^{\underline{k}}\eta(\underline{x})| = o_P(n^{-(1-\alpha p)/2+\alpha|\underline{k}|+\varepsilon}).$$

On considère Δ_n une discrétisation de l'hypercube $S_{\underline{X}}$ suffisamment fine pour avoir :

$$\sup_{\underline{x} \in S_{\underline{X}}} \inf_{\underline{x}' \in \Delta_n} \left| \Sigma_{\underline{x}}^{-1} W_{\underline{x}} - \Sigma_{\underline{x}'}^{-1} W_{\underline{x}'} \right| = o_P(1).$$

Comme on l'a vu dans le lemme 3.2, on a :

$$\sup_{\underline{x} \in S_{\underline{X}}} \left| \underline{\hat{a}}^* - \Sigma_{\underline{x}}^{-1} W_{\underline{x}} \right| \xrightarrow[n \to \infty]{P} 0.$$

Or on a aussi, par définition de $\hat{a}^*(\underline{x})_{\underline{k}}$:

$$\hat{a}^*(\underline{x})_{\underline{k}} = \frac{c_n^{-1} h^{|\underline{k}|}}{\underline{k}!} \left[\widehat{D^{\underline{k}}\eta}(\underline{x}) - D^{\underline{k}}\eta(\underline{x}) \right].$$

D'où, en utilisant aussi $c_n = (nh^p)^{-1/2}$, on obtient, pour $|\underline{k}| = 0$ ou 1 :

$$\begin{aligned}
\sup_{\underline{x} \in S_{\underline{X}}} \left| \widehat{D^{\underline{k}}\eta}(\underline{x}) - D^{\underline{k}}\eta(\underline{x}) \right| &= \frac{1}{h^{|\underline{k}|}\sqrt{nh^p}} \sup_{\underline{x} \in S_{\underline{X}}} \hat{a}^*(\underline{x})_{\underline{k}} \\
&= \frac{1}{h^{|\underline{k}|}\sqrt{nh^p}} \left\{ \sup_{\underline{x} \in S_{\underline{X}}} |(\Sigma_{\underline{x}}^{-1} W_{\underline{x}})_{\underline{k}}| + o_P(1) \right\} \\
&= \frac{1}{h^{|\underline{k}|}\sqrt{nh^p}} \left\{ \sup_{\underline{x} \in \Delta_n} |(\Sigma_{\underline{x}}^{-1} W_{\underline{x}})_{\underline{k}}| + o_P(1) \right\}.
\end{aligned}$$

Si $D_{\underline{k},\underline{l}} = \sup_{\underline{x} \in S_{\underline{X}}} (\Sigma_{\underline{x}}^{-1})_{\underline{k},\underline{l}}$, on sait, grâce à la contrainte $\inf_{\underline{x} \in S_{\underline{X}}} (\rho(\underline{x}) f(\underline{x})) > 0$, que tous les $D_{\underline{k},\underline{l}}$ sont finis. Donc, en utilisant les résultats du lemme 3.3, on a :

$$\begin{aligned}
\sup_{\underline{x} \in S_{\underline{X}}} \left| \widehat{D^{\underline{k}}\eta}(\underline{x}) - D^{\underline{k}}\eta(\underline{x}) \right| &\leq \sum_{\underline{l} \in A_q} D_{\underline{k},\underline{l}} \sup_{\underline{x} \in \Delta_n} \left| \frac{1}{h^{|\underline{k}|}\sqrt{nh^p}} (W_{\underline{x}})_{\underline{l}} - \mathbb{E}\left[\frac{1}{h^{|\underline{k}|}\sqrt{nh^p}} (W_{\underline{x}})_{\underline{l}} \right] \right| \\
&+ O(h^{q+1-|\underline{k}|}) + o_P(\frac{1}{h^{|\underline{k}|}\sqrt{nh^p}}). \qquad (3.15)
\end{aligned}$$

Il est à noter que dans l'expression précédente, le terme $O(h^{q+1-|\underline{k}|})$ correspond à l'ordre de grandeur de l'espérance de $W_{\underline{x}}$. Comme signalé dans une remarque après l'énoncé du lemme 3.3, on peut éventuellement, selon la parité de q avoir un terme plus fin en $O(h^{q+2-|\underline{k}|})$. Mais pour la majoration, le terme en $O(h^{q+1-|\underline{k}|})$ est suffisant.

On note

$$d_{\underline{x}} = \left| \frac{1}{h^{|\underline{k}|}\sqrt{nh^p}} (W_{\underline{x}})_{\underline{l}} - \mathbb{E}\left[\frac{1}{h^{|\underline{k}|}\sqrt{nh^p}} (W_{\underline{x}})_{\underline{l}} \right] \right|.$$

Rappelons maintenant la définition de $W_{\underline{x}}$

$$W_{\underline{x}} = c_n \sum_{i=1}^{n} l_1 \left[\bar{\eta}(\underline{x}, \underline{X}_i), Y_i \right] R(\underline{x}, \underline{X}_i) K^p(H^{-1}(\underline{X}_i - \underline{x})).$$

On note

$$\Psi(\underline{x}, \underline{X}_i, Y_i) = l_1 \left[\bar{\eta}(\underline{x}, \underline{X}_i), Y_i\right] R(\underline{x}, \underline{X}_i) K^p(H^{-1}(\underline{X}_i - \underline{x})).$$

On a donc

$$W_{\underline{x}} = c_n \sum_{i=1}^{n} \Psi(\underline{x}, \underline{X}_i, Y_i)$$

et

$$d_{\underline{x}} = \left| \sum_{i=1}^{n} \left\{ \frac{1}{nh^{|\underline{k}|+p}} \left\{\Psi(\underline{x}, \underline{X}_i, Y_i)\right\}_{\underline{l}} - \mathbb{E}\left[\frac{1}{nh^{|\underline{k}|+p}} \left\{\Psi(\underline{x}, \underline{X}_i, Y_i)\right\}_{\underline{l}} \right] \right\} \right|.$$

On va maintenant utiliser l'inégalité de Bernstein pour obtenir, pour $\tau > 0$ donné et $\underline{l} \in A_q$:

$$\mathbb{P}\left(d_{\underline{x}} > \tau\right) \leq 2\, exp\left(\frac{-\tau^2}{2\sum_{i=1}^{n} Var\left(\frac{1}{nh^{|\underline{k}|+p}} \left\{\Psi(\underline{x}, \underline{X}_i, Y_i)\right\}_{\underline{l}} \right) + \frac{2}{3nh^{|\underline{k}|+p}} M\tau} \right).$$

où M est une constante telle que

$$\mathbb{P}\left(\left\{\Psi(\underline{x}, \underline{X}_i, Y_i)\right\}_{\underline{l}} - \mathbb{E}\left[\left\{\Psi(\underline{x}, \underline{X}_i, Y_i)\right\}_{\underline{l}} \right] \leq M \right) = 1.$$

On pose, pour $v > 0$, $\varepsilon > 0$

$$\tau = n^{-(1-\alpha p)/2 + \alpha|\underline{k}| + \varepsilon} v.$$

L'inégalité de Bernstein devient alors

$$\mathbb{P}\left(d_{\underline{x}} > n^{-(1-\alpha p)/2+\alpha|\underline{k}|+\varepsilon} v\right) \quad \leq$$

$$2\, exp\left(\frac{-n^{1+\alpha p+2\alpha|\underline{k}|+2\varepsilon}}{2\sum_{i=1}^{n} Var\left(\frac{1}{h^{|\underline{k}|+p}} \left\{\Psi(\underline{x}, \underline{X}_i, Y_i)\right\}_{\underline{l}} \right) + \frac{2n}{3h^{|\underline{k}|+p}} M n^{-(1-\alpha p)/2+\alpha|\underline{k}|+\varepsilon} v} \right).$$

Mais, grâce au lemme 3.3, on sait que $Var(\{W_{\underline{x}}\}_{\underline{l}} = O(1)$, ce qui implique

$$Var\left(\frac{1}{h^{|\underline{k}|+p}} \left\{\Psi(\underline{x}, \underline{X}_i, Y_i)\right\}_{\underline{l}} \right) = O(n^{\alpha p + 2\alpha|k|}).$$

Par conséquent, on obtient, pour le terme de droite de l'inégalité obtenue, une expression de la forme $2exp(-C \times n^{2\varepsilon})$, où C est une constante positive. Le premier terme du membre de droite de l'inégalité (3.15) est donc majoré par

$$2 \sum_{\underline{l} \in A_q} D_{\underline{k},\underline{l}} exp(-Cn^{2\varepsilon})$$

qui tend vers 0 quand $n \to +\infty$. On a donc vu que

$$\mathbb{P}\left(d_{\underline{x}} > \tau\right) \leq 2exp(-Cn^{2\varepsilon}).$$

On peut toujours écrire, pour tout τ positif :

$$|d_{\underline{x}}| \leq \tau \mathbb{1}_{|d_{\underline{x}}|\leq\tau} + |d_{\underline{x}}|\mathbb{1}_{|d_{\underline{x}}|>\tau}.$$

Or, en reprenant l'expression de τ utilisée précédemment, la probabilité de l'événement $\{|d_{\underline{x}}| > \tau\}$ a été majorée grâce à l'inégalité de Bernstein par un terme en $2exp(-Cn^{2\varepsilon})$. On obtient ainsi :

$$\sup_{\underline{x}\in S_{X_m}} |d_{\underline{x}}| \leq O_P(n^{-(1-\alpha p)/2+\alpha|\underline{k}|+\varepsilon}).$$

C'est-à-dire, pour tout $\varepsilon > 0$

$$\begin{aligned}\sup_{\underline{x}\in S_{\underline{X}}} \left|\widehat{D^{\underline{k}}\eta}(\underline{x}) - D^{\underline{k}}\eta(\underline{x})\right| &\leq O_P(n^{-(1-\alpha p)/2+\alpha|\underline{k}|+\varepsilon}) + O(h^{q+1-|\underline{k}|}) + o_P(\frac{1}{h^{|\underline{k}|}\sqrt{nh^p}}) \\ &\leq O_P(n^{-(1-\alpha p)/2+\alpha|\underline{k}|+\varepsilon}) + O(h^{q+1-|\underline{k}|}) + o_P(n^{-(1-\alpha p)/2+\alpha|\underline{k}|}) \\ &\leq O_P(n^{-(1-\alpha p)/2+\alpha|\underline{k}|+\varepsilon}) + O(h^{q+1-|\underline{k}|}).\end{aligned}$$

L'inégalité précédente est vraie pour tout $\varepsilon > 0$. On prend $\varepsilon = \delta$, où δ est le nombre apparaissant dans l'inégalité (3.9). On utilise ce δ pour le terme avec $|\underline{k}| = 0$ et pour le terme avec $|\underline{k}| = 1$. Maintenant on va montrer que c'est le terme $O_P(n^{-(1-\alpha p)/2+\alpha|\underline{k}|+\delta})$ qui domine. On veut donc avoir :

$$h^{q+1} = o(n^{-(1-\alpha p)/2+\delta}).$$

Cela sera vrai si et seulement si

$$\alpha > \frac{1-2\delta}{2q+p+2}$$

pour $|\underline{k}| = 0$ ou 1. δ a justement été choisi pour vérifier cette condition. On en déduit, pour $|\underline{k}| = 0$ ou 1,

$$\sup_{\underline{x}\in S_{\underline{X}}} |(\hat{\eta}(\underline{x}) - \eta(\underline{x}))(\widehat{D^{\underline{k}}\eta}(\underline{x}) - D^{\underline{k}}\eta(\underline{x}))| \leq O_P(n^{-1+\alpha(p+|\underline{k}|)+2\delta}). \qquad (3.16)$$

A ce stade, on veut que le membre de droite de l'inégalité (3.16) tende vers 0 quand n tend vers l'infini pour $|\underline{k}| = 0$ ou 1. Pour cela, il faudra avoir

$$\alpha < \frac{1-2\delta}{p+|\underline{k}|}.$$

Dans les cas qui nous intéressent, à savoir $|\underline{k}| = 0$ et $|\underline{k}| = 1$, cette condition est vérifiée car elle est moins stricte que l'hypothèse (3.9) du théorème.
Ainsi on a montré que

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} r_n(\underline{X}_i) = o_P(1).$$

Il reste à s'intéresser au terme $\frac{1}{n}\sum_{i=1}^{n}\varepsilon_n(\underline{X}_i)$ (*cf* condition (3.13)). On veut avoir

$$\frac{1}{n}\sum_{i=1}^{n}\varepsilon_n(\underline{X}_i) = o_P(1).$$

Il suffira donc de montrer que

$$\sup_{\underline{x}\in S_{\underline{X}}}|\varepsilon_n(\underline{x})| = o_P(1).$$

Par définition, on a

$$\varepsilon_n(\underline{x}) = (g^{-1})''(\eta(\underline{x}))\nabla\eta(\underline{x})(\hat{\eta}(\underline{x}) - \eta(\underline{x})) + (g^{-1})'(\eta(\underline{x}))(\widehat{\nabla\eta}(\underline{x}) - \nabla\eta(\underline{x})).$$

Grâce aux calculs précédents et aux hypothèses du théorème, on peut écrire

$$\begin{aligned}\sup_{\underline{x}\in S_{\underline{X}}}|\varepsilon_n(\underline{x})| &\leq O_P(n^{-(1-\alpha p)/2+\delta}) + O_P(n^{-(1-\alpha p)/2+\alpha+\delta}) \\ &\leq O_P(n^{-(1-\alpha p)/2+\alpha+\delta}).\end{aligned}$$

Et grâce à la condition $\alpha < (1-2\delta)(p+2)$, on obtient le résultat

$$\sup_{\underline{x}\in S_{\underline{X}}}|\varepsilon_n(\underline{x})| = o_P(1).$$

Comme on a montré que

$$\hat{\underline{\beta}} - \underline{\beta} = \frac{1}{n}\sum_{i=1}^{n}[\varepsilon_n(\underline{X}_i) + r_n(\underline{X}_i)] + o_P(1),$$

on a donc bien

$$\hat{\underline{\beta}} - \underline{\beta} = o_P(1).$$

⊗

Remarque.

On a montré la convergence de l'estimateur $\hat{\underline{\beta}}$, mais sans toutefois parvenir à trouver un taux de convergence satisfaisant. Pour y parvenir, il faudrait sans doute essayer de raffiner la démonstration. Nous avons déjà essayé de montrer qu'on pouvait avoir une vitesse en $\sqrt{n}$ mais nous sommes à chaque fois arrivés à des contraintes du genre $q > p/2$. Quand on voit qu'en pratique on utilise des développements d'un ordre guère plus élevé que 1 ou 2, on se rend compte qu'une telle contrainte est tout simplement inacceptable.

Conclusion

Ainsi, on a montré la convergence de l'estimateur $\underline{\hat{\beta}}$ obtenu grâce à l'étape A de la méthode GSIM. Nous n'avons pas eu le temps, lors de ces travaux, de nous intéresser de près à la convergence réelle de la méthode sur des simulations dans le cas standard. Ces simulations nous permettraient de mieux appréhender le comportement de l'estimateur, et nous permettrait d'ailleurs d'avoir une idée de l'ordre de grandeur de la vitesse de convergence. C'est un travail qu'il faudra effectuer dans la suite des travaux. En effet, l'objectif premier a ici été l'application aux données de biopuces, c'est-à-dire bien loin du cas standard où on aurait pu constater des propriétés asymptotiques.

Résultats sur des données de biopuces

Nous considérons dans ce chapitre l'application de l'algorithme pénalisé GSIM_λ aux données de biopuces relevant de l'analyse discriminante entre deux groupes. Des résultats avec GSIM sans pénalité, avec un nombre de gènes restreint, seront aussi présentés et ce uniquement pour le jeu de données "Colon" qui est celui que nous avons le plus étudié.

Pour être en mesure de comparer la méthode à d'autres méthodes proposées dans la litérature, nous nous sommes tout d'abord intéressés aux jeux de données classiquement utilisés dans ce genre de problématique et par la suite aux données dont nous disposions à l'Institut Curie. Comme nous allons le voir les résultats obtenus sont tout à fait satisfaisants, et ce en dépit de la réduction en indice simple seulement. Le prétraitement des données va dépendre du jeu de données considéré.

3.4.1 Mode de validation des résultats

Pour tester l'algorithme sur de vrais jeux de données on va utiliser une technique de rééchantillonnage : on effectue 100 subdivisions aléatoires des données en un ensemble d'apprentissage et un ensemble de test. On calcule la direction de projection β sur les individus de l'ensemble d'apprentissage dont on connaît la classe, et on utilise les résultats obtenus pour prédire la classe, considérée comme inconnue, des individus de l'ensemble de test.

Le taux d'erreur est calculé comme la moyenne du taux d'erreur sur les individus de l'ensemble de test, moyenne calculée sur l'ensemble des 100 subdivisions.

3.4.2 Résultats expérimentaux

3.4.2.1 Colon

Ce jeu de données est constitué de 62 profils d'expression issus de deux populations : 40 tissus tumoraux et 22 tissus sains. Chaque profil comporte 2000 niveaux d'expression. On trouvera dans Alon *et al.* [4] une description complète de ces données.

Pré-traitement des données

Avant toute analyse statistique ces données "brutes" sont pré-traitées selon un protocole comportant une étape de seuillage, de filtrage et de transformation logarithmique (Dudoit *et al.* [15]).

Le seuillage consiste à ne conserver que les valeurs comprises entre 100 et 16000. Le filtrage a pour but d'éliminer avant tout traitement les gènes dont l'expression est trop uniforme et dont la variation n'est pas significative par rapport à la précision de mesure des niveaux d'expression. Pour chaque gène, on relève la valeur minimale s_{min} et la valeur maximale s_{max} du niveau d'expression parmi les tissus disponibles et on ne garde que les gènes tels que $s_{max}/s_{min} > 5$ et $s_{max} - s_{min} > 500$. Et on effectue une transformation logarithmique en base 10 des données. Pour appliquer l'algorithme GSIM on va aussi avoir à procéder à une sélection de gènes via une méthode de seuillage bayésien. Pour que la statistique de test correspondant à la différence des moyennes entre les deux classes pour chaque gène soit significative, on procède au préalable à une standardisation "en ligne" des données ce qui revient à normaliser chaque biopuce en retranchant sa moyenne et en divisant par son écart-type. Cette normalisation peut sembler "grossière" par rapport aux méthodes exposées dans le chapitre 1, cependant nous ne disposons pas des informations nécessaires pour mettre en œuvre d'autres méthodes.

Pour ce jeu de données, le fichier d'apprentissage est constitué de 41 échantillons et chaque sous-population y est représentée dans la même proportion que dans la population totale.

Paramètres d'implantation

Pour GSIM, sur ce jeu de données, on montre l'évolution de la méthode non pénalisée à la méthode pénalisée, avec les choix de paramètres correspondant. Pour les résultats avec la procédure non pénalisée, le paramètre de lissage h_A est choisi par L-curve à partir d'une grille de 20 points de l'intervalle $[0,5;90]$ log-linéairement espacés. On gardera le même choix pour la procédure pénalisée à λ fixé. Par contre, quand il faut choisir λ, le choix de h_A par L-curve n'est plus possible et on fait alors une validation croisée sur l'ensemble d'apprentissage simultanément en h_A et en λ sur 5 points de l'intervalle $[7,90]$ pour h_A et sur 5 points de $[0.01,30]$ pour λ ; dans les deux cas, les points sont log-linéairement espacés. Notons que l'on a ici bien réduit la taille des grilles, et ce à cause du coût algorithmique de la validation croisée.

Pour rOPG, seuls les résultats de la version pénalisée de la méthode sont présentés ici. On effectue alors le choix de h_A et de λ simultanément par validation croisée sur 5 points de l'intervalle $[0.5,10]$ pour h_A et sur 5 points de $[0.01,30]$ pour λ ; dans les deux cas, les points sont log-linéairement espacés. Les valeurs envisagées pour h_A sont différentes selon la méthode considérée (GSIM ou rOPG). En effet, la fenêtre pour rOPG correspond à un noyau de dimension 1, puisque dans la version raffinée de OPG le noyau est déterminé sur les covariables projetées.

Pour kNN, le nombre de voisins est déterminé par validation croisée sur l'ensemble d'apprentissage dans la grille des entiers $\{1, 2, 3, \ldots, 20\}$.

Résultats sans pénalisation

On sélectionne un certain nombre de gènes par seuillage bayésien. On limite à p_{max} le nombre de gènes sélectionnés. Les résultats du taux d'erreur sur l'ensemble de test par rééchantillonnage sont fournis dans le tableau 3.1 pour différentes valeurs de p_{max}.

p_{max}	2	3	4	5	6	8	10	20	30	39
moy	0.185	0.197	0.213	0.215	0.207	0.216	0.227	0.234	0.267	0.264
std	0.080	0.079	0.081	0.087	0.088	0.099	0.091	0.089	0.094	0.088

Table 3.1 – `Colon.` Moyenne et écart-type du taux d'erreur dans le fichier test en fonction du nombre limite de gènes considérés. Ici, on utilise un critère non pénalisé.

Sur ce tableau, on peut vérifier que, comme annoncé en section 3.2.3, les résultats se dégradent rapidement quand le nombre de gènes augmente. On vérifie aussi sur les figures 3.1 et 3.2 qu'augmenter le paramètre h permet de stabiliser les résultats.

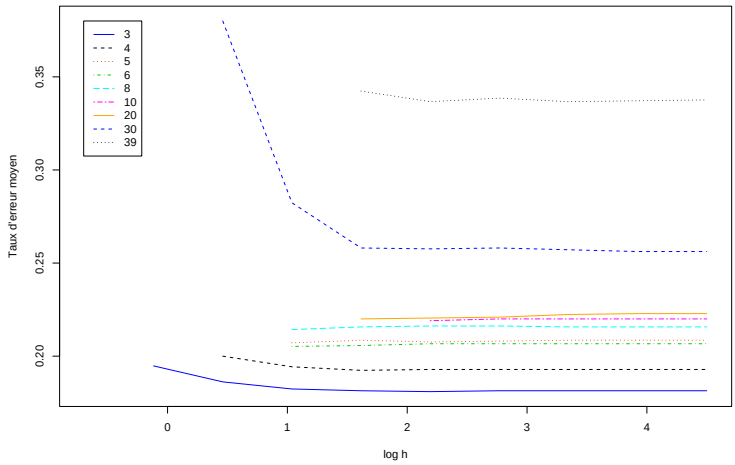

Figure 3.1 – Taux d'erreur moyen en fonction de log(h). On remarque ici que le taux d'erreur moyen se stabilise quand h devient grand.

Nous allons maintenant étudier les résultats obtenus avec une pénalité de type Ridge.

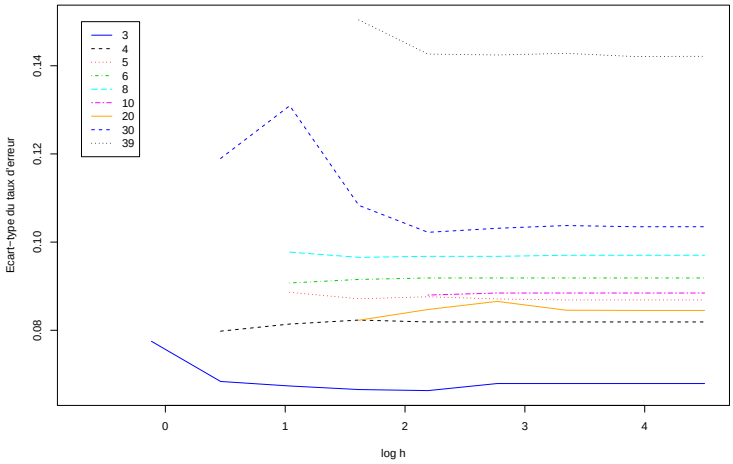

Figure 3.2 – Ecart-type du taux d'erreur en fonction de log(h). On remarque ici que l'écart-type du taux d'erreur se stabilise quand h devient grand.

Résultats avec pénalisation de type Ridge

Nous répétons les mêmes simulations que dans la partie précédente, mais cette fois en ajoutant une pénalité de type Ridge. L'objectif ici est de montrer que l'ajout d'une pénalité améliorent nettement les résultats. Nous ne nous intéressons donc pas tout de suite au choix du paramètre λ, nous étudierons son influence sur les résultats ultérieurement, et nous prenons dans la première partie de ces résultats $\lambda = 5$. On trouvera les résultats dans le tableau 3.2.

p_{max}	2	3	4	5	6	8	10	20	30	40
moy	0.174	0.171	0.169	0.162	0.155	0.145	0.145	0.142	0.140	0.139
std	0.070	0.060	0.062	0.062	0.062	0.069	0.065	0.063	0.063	0.063

Table 3.2 – `Colon`. Moyenne et écart-type du taux d'erreur dans le fichier test en fonction du nombre limite de gènes considérés. Ici, on a introduit une pénalité de type Ridge, en fixant $\lambda = 5$.

On observe une nette amélioration des résultats grâce à l'introduction de cette pénalité, mais l'autre avantage c'est qu'on peut maintenant se permettre d'augmenter réellement le nombre de gènes considérés, et même de considérer tous les gènes. En fait, nous allons maintenant comparer deux approches différentes pour plusieurs valeurs possibles de λ : dans un premier temps, nous conserverons tous les gènes et dans un deuxième temps, nous effectuerons une pré-sélection des covariables, en gardant toutes les covariables sélectionnées par la méthode de seuillage empirique

bayésien. Les résultats sont présentés dans le tableau 3.3.

λ	0.0001		1		5		10	
	p_{max}	p_{lim}	p_{max}	p_{lim}	p_{max}	p_{lim}	p_{max}	p_{lim}
moy	0.155	0.167	0.154	0.139	0.155	0.136	0.155	0.138
std	0.051	0.057	0.056	0.062	0.060	0.061	0.060	0.066

λ	20		30		40	
	p_{max}	p_{lim}	p_{max}	p_{lim}	p_{max}	p_{lim}
moy	0.157	0.140	0.156	0.140	0.160	0.140
std	0.063	0.068	0.065	0.067	0.064	0.066

Table 3.3 – `Colon.` Moyenne et écart-type du taux d'erreur dans le fichier test en fonction du paramètre λ de la pénalité. Ici, on compare les résultats selon que l'on a considéré tous les gènes (p_{max}) ou un nombre limité de gènes (p_{lim}).

On constate que, même si les résultats restent comparables, ils sont légèrement meilleurs quand on considère un nombre réduit de gènes. Notons que le nombre de gènes sélectionnés par la méthode seuillage bayésien varie entre 19 et 72 selon l'ensemble d'apprentissage considéré. Ces résultats laissent penser que la méthode de sélection de gènes permet de conserver l'essentiel de l'information discriminante du jeu de données alors que considérer tous les gènes apportera plus de bruit que d'information discriminante supplémentaire.

Dans tous les cas, l'ajout de la pénalité a permis de fortement stabiliser les résultats. En effet, dans cette version pénalisée de l'algorithme, seuls quelques individus ont tendance à être très souvent mal classés, des individus qui posent des problèmes quelles que soient les méthodes utilisées dans la littérature ; il y a une forte suspition d'erreur d'étiquetage pour ces quelques individus. Ainsi on obtient un taux de mauvais classement de pratiquement 100% pour ces quelques individus et un taux de mauvais classement très proche de 0 pour les autres individus, alors que les différences entre les taux de mauvais classements des individus étaient moins nettes dans le cas du critère non pénalisé.

De plus, on constate que les résultats ne sont pas très sensibles au choix du λ. Il semble juste déconseillé de prendre un λ vraiment trop petit.

Comparaison avec d'autres procédures.

On a donc constaté l'intérêt de l'introduction d'un terme de pénalité dans notre critère. Nous allons maintenant faire varier le paramètre de régularisation λ pour choisir, à chaque nouvel ensemble d'apprentissage, le couple (h_A, λ) le plus approprié, par validation croisée sur l'ensemble d'apprentissage. On ne fait ici aucune autre sélection de gènes que celle qui est faite par filtrage pendant le prétraitement des données. Les résultats obtenus sont donnés table 4 et figure 3.3.

	DLDA	DQDA	kNN	rOPG	$\texttt{GSIM}_\lambda$
moy	0.144	0.154	0.204	0.153	0.148
std	0.057	0.064	0.071	0.060	0.056

Table 3.4 – Colon. Etude resampling : moyenne et écart-type du taux d'erreur dans le fichier test. Pour la méthode kNN, le nombre de voisins choisi est en moyenne de 5.06.

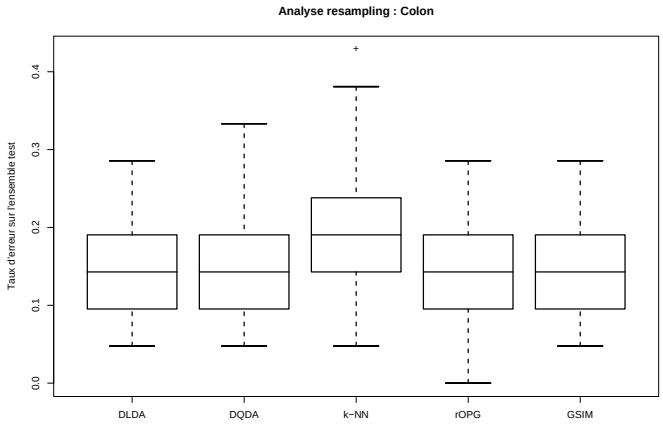

Figure 3.3 – Colon. Boxplot du taux d'erreur de classification moyen sur l'ensemble test.

Les conclusions que l'on peut tirer de ce tableau de résultats et du boxplot ne sont pas très tranchées. En effet, exceptée la méthode kNN qui donne des résultats plutôt moins satisfaisants, toutes les méthodes semblent plus ou moins équivalentes au niveau des résultats, cela tient sans doute aussi à la nature assez particulière des données. De plus, la méthode kNN présente un inconvénient majeur : on se trouve souvent confronté à des problèmes d'indécision, quand le k-voisinage contient autant de voisins d'une classe que de l'autre. Ce phénomène d'indécision rend la méthode instable, les résultats n'étant alors pas reproductibles.

On constate que GSIM améliore légèrement les résultats obtenus avec rOPG. La méthode DLDA donne ici de bons résultats. Notons toutefois que les méthodes DLDA et DQDA sont très sensibles au prétraitement. En effet, si dans le prétraitement on omet la standardisation en lignes des données, on multiplie par 2 le taux d'erreur obtenu pour DLDA (on passe à 28% d'erreur) alors que les méthodes GSIM et rOPG restent relativement stables (elles passent respectivement à 15.3% et 20.4% d'erreur).

3.4.2.2 Leukemia

Ce jeu de données est constitué de 72 profils d'expression issus de deux populations : 47 tissus atteints de Leucémie lymphoblastique aiguë (ALL) et 25 tissus atteints de Leucémie myéloïde aiguë (AML). Il est à noter que ce jeu de données peut aussi être considéré comme problème de discrimination multiclasses dans la mesure où les 47 tissus ALL se subdivisent en deux populations selon que les cellules analysées sont de type B (38 cas) ou de type T (9 cas). Chaque profil comporte 7129 niveaux d'expression de gènes. On trouvera dans [21] une description complète de ces données. On effectue ici exactement le même pré-traitement que pour le jeu de données Colon. En sortie du prétraitement, il reste environ 3000 à 3500 gènes selon l'ensemble d'apprentissage considéré. On utilise les mêmes grilles en h_A et en λ.

	DLDA	DQDA	KNN	GSIM_λ
moy	0.035	0.049	0.053	0.038
std	0.031	0.041	0.036	0.032

Table 3.5 – Leukemia en binaire. Etude resampling : moyenne et écart-type du taux d'erreur dans le fichier test. Pour la méthode kNN, le nombre de voisins choisi est en moyenne de 2.86.

Ici, la comparaison des méthodes mène à peu près aux mêmes conclusions que pour le jeu de données de Colon. On remarque de bonnes performances des méthodes GSIM_λ et DLDA.

3.4.2.3 Données de l'Institut Curie

Fortes doses contre faibles doses

On reprend le jeu de données déjà étudié dans le chapitre précédent : faibles doses contre fortes doses. Ici, comme on dispose de toutes les données brutes, une vraie normalisation a pu être appliquée. On a normalisé les données par une méthode de lowess par bloc avec remise à l'échelle et lissage des résidus. On n'effectue donc aucun prétraitement sur les données si ce n'est de remettre les ratios normalisés en échelle logarithmique.

Ce jeu de données est constitué de 34 biopuces issues de deux populations : 24 obtenus à partir de levures exposées à de fortes doses d'irradiation γ et 10 exposées à de faibles doses. Pour chaque biopuce, on a mesuré l'expression de 6327 gènes. En raison du grand nombre de gènes, il pourra s'avérer intéressant de réduire le nombre de gènes en entrée de l'algorithme, ne serait-ce que pour réduire les temps de calculs.

Pour l'analyse de ce jeu de données, le fichier d'apprentissage est constitué de 23 échantillons et chaque sous-population y est représentée dans la même proportion que dans la population totale à savoir 16 individus exposés à de fortes doses et 7 individus exposés à de faibles doses.

Etant donné le faible nombre d'individus, l'analyse par échantillons tests peut sembler abusive. On fera donc aussi une étude "LeaVe One out" (LVO), c'est-à-dire qu'on enlèvera tour à tour chacun des individus pour faire l'apprentissage sur l'ensemble des $n-1$ individus restants, et on prédira ensuite sur cet individu qui avait été oté. On effectue donc une prédiction pour chacun des 34 individus, on obtiendra alors un nombre d'erreurs compris entre 0 si on a bien classé tous les individus et 34 si on les a tous mal classés.

Pour l'instant les seuls résultats disponibles sont pour $\lambda = 5$ qui semble une valeur assez adaptée, et h_A est fixé par $L - curve$ sur une grille étendue (de 0.5 à 200) et assez fine (10 points log-linéairement espacés). On a fait ce choix car cela permet d'obtenir des résultats beaucoup plus vite qu'en validation croisée. Dans le tableau 3.6 (resp. 3.7), on observe les résultats obtenus en fonction du nombre limite de gènes fixé pour l'analyse par LVO (resp. ré-échantillonage).

nb gènes / Méthode	10	30	50	100	p_{lim}	p_{max}
$GSIM_\lambda$	0	1	1	1	1	0
DLDA	0	1	1	1	1	1
DQDA	0	1	2	2	1	1
kNN	0	0	0	0	1	0

Table 3.6 – `Faibles et fortes doses, analyse LVO.` Nombre d'erreurs en fonction du nombre limite de gènes considérés. Ici, on a utilisé une pénalité de type Ridge, en fixant $\lambda = 5$.

Méthode \ nb gènes		10	30	50	100	p_{lim}	p_{max}
GSIM	moy	0.022	0.038	0.035	0.031	0.028	0.015
	std	0.048	0.062	0.056	0.050	0.046	0.048
DLDA	moy	0.020	0.028	0.025	0.027	0.028	0.030
	std	0.047	0.051	0.046	0.047	0.044	0.053
DQDA	moy	0.018	0.031	0.029	0.027	0.023	0.031
	std	0.045	0.052	0.051	0.044	0.041	0.055
kNN	moy	0.008	0.019	0.033	0.031	0.036	0.036
	std	0.029	0.047	0.061	0.062	0.064	0.063

Table 3.7 – `Faibles et fortes doses, analyse resampling.` Moyenne du taux d'erreur dans le fichier test en fonction du nombre limite de gènes considérés. Ici, on a utilisé une pénalité de type Ridge, en fixant $\lambda = 5$.

Les résultats obtenus sont très bons, et ce pour toutes les méthodes, même avec un petit nombre de gènes. Cela montre que ces deux jeux de données se discriminent très bien. Il est donc difficile d'utiliser ces résultats pour comparer les méthodes. On notera toutefois qu'ici, contrairement aux exemples précédents, la méthode kNN donne plutôt les meilleurs résultats (avec un nombre moyen de voisins allant de 1.1 à 1.56 selon le nombre de gènes). Pour l'étude par rééchantillonnage, on constate aussi que GSIM devient la meilleure méthode quand on conserve tous les gènes, ce qui montre que cette méthode résiste mieux au bruit.

D'un point de vue biologique, il est intéressant de regarder de plus près quels sont les échantillons mal classés. Dans l'analyse LVO, quand il y a au moins une erreur de commise, l'échantillon 25 en fait systématiquement partie. Cela devient particulièrement révélateur quand on considère qu'il s'agit, parmi les individus soumis à de faibles doses d'irradiations, de l'individu qui a reçu l'irradiation la plus importante. Il n'est donc pas surprenant de constater que c'est celui qui est le plus difficile à classer, même si la dose d'irradiation reçue reste nettement inférieure à celles utilisées pour les fortes doses. Un autre jeu de données est mal classé deux fois, et seulement par la méthode DQDA, il s'agit d'un individu de la classe "fortes doses". Dans l'analyse par ré-échantillonage, le jeu de données "faible dose" ayant reçu la plus forte dose d'irradiation, faibledose1, est mal classé de façon quasi-systématique. Trois autres échantillons sont parfois mal classés : parmi eux, les deux jeux de données de faibles doses, les plus irradiés après faibledose1, ce qui n'est pas anodin. Le dernier individu qui est parfois mal classé est le même que pour l'analyse LVO.

Faibles doses, non irradiés, formaldéhyde.

On a donc vu qu'on parvenait assez facilement à faire la différence entre des individus exposés à de fortes doses de rayonnement et à de faibles doses de rayonnement. On va maintenant reprendre le jeu des faibles doses et regarder si on parvient

tout aussi bien à le différencier d'un jeu de données correspondant à des individus non irradiés (12 lames). Cette comparaison a un intérêt biologique : en effet, si on a vu que les réactions induites dans la transcription n'était pas les même selon la dose d'irradiation, on est en droit de se demander si une réaction se déclenche effectivement en présence de faibles radiations. Etant donné que le jeu de données faibles correspond à toute une gamme de faibles expositions, on peut éventuellement s'attendre à avoir du mal à classer les individus les moins exposés, et peut-être même à faire apparaître, même si ce n'est pas la technique appropriée pour cela, un seuil de réaction.

Dans le cadre de cet exemple, on s'intéressera également au jeu de données "formaldéhyde" (7 lames) qui correspond à des levures ayant été exposées à doses assez faibles à un agent génotoxique. On verra ainsi si la réaction à une agression est spécifique de cette agression (irradiation ou formaldéhyde) et si l'exposition à ce formaldéhyde a réellement des conséquences au niveau de la transcription de l'ARNm au sein des cellules.

De la même façon que précédemment, les données ayant été normalisées aucun autre prétraitement que la log-transformation des données ne sera nécessaire.

Dans le cadre de ce jeu de données, on ne s'intéressera qu'au cas où on prend tous les gènes (environ 6 000). On peut envisager deux approches différentes, une où l'on considère chaque ensemble de jeux de données contre un autre, et l'autre où l'on considère chacun des jeux de données contre les deux autres. On essaiera les deux, on aura donc les expériences suivantes à faire :

- Non irradiés vs. faibles doses
- Non irradiés vs. formaldéhyde
- Formaldéhyde vs. faibles doses
- Non irradiés vs. les deux autres
- Faibles doses vs. les deux autres
- Formaldéhyde vs. les deux autres

Comme les jeux de données sont encore une fois assez restreints en nombre d'individus, on se contentera ici de faire une analyse LVO. Les résultats sont obtenus pour $\lambda = 5$ qui semble une valeur assez adaptée, h_A est toujours choisi par $L-curve$. Pour $rOPG$, le choix de h_A est fait par validation croisée et λ est fixé à la valeur 0.0001 (*cf* remarque à la fin de la section 3.2.5).

Les résultats obtenus ne font pas vraiment ressortir de méthode meilleure qu'une autre. DLDA commet plutôt plus d'erreurs que les autres, KNN oscille du meilleur au pire selon les jeux de données et GSIM, rOPG et DQDA fournissent en moyenne des résultats parmi les meilleurs observés. Pour les comparaisons un à un, on voit que la discrimination la meilleure semble être entre les données "formaldéhyde" et "faibles doses", montrant que les réactions induites par ces deux traitements sont différentes. La séparation se fait aussi assez bien entre "faibles doses" et "non irradiés" ce qui montre bien que même une faible irradiation peut avoir des conséquences sur l'organisme. La plus grande incertitude réside dans la distinction entre formaldéhyde et non irradiés (*i.e.* non traités), les résultats étant plutôt mauvais.

Méthode / Expérience	n total	$GSIM_\lambda$	rOPG	DLDA	DQDA	kNN
Non irradiés vs. faibles doses	22	3	3	3	3	7
Non irradiés vs. formaldéhyde	19	5	3	6	7	3
Formaldéhyde vs. faibles doses	17	0	1	1	1	1
Non irradiés vs. les deux autres	29	10	9	12	7	8
Faibles doses vs. les deux autres	29	2	3	3	4	7
Formaldéhyde vs. les deux autres	29	5	4	4	6	3

Table 3.8 – `Non irradiés contre faibles doses, analyse LVO.` Nombre d'erreurs en fonction du nombre limite de gènes considérés. Ici, on a utilisé une pénalité de type Ridge, en fixant $\lambda = 5$, tous les gènes sont considérés.

En ce qui concerne les résultats quand on compare un traitement aux deux autres, les résultats sont très mauvais quand ce sont les non irradiés que l'on tente de discriminer des autres. Cela n'est pas tellement surprenant : nous venons de voir que, des trois jeux de données, les plus différents sont "faibles doses" et "formaldéhyde". Or ici, ces deux jeux de données viennent justement d'être réunis comme s'ils appartenaient à une seule et même classe. Ainsi on comprend bien que les différences au sein de cette classe risquent d'être plus importantes qu'entre les deux classes que nous avons artificiellement créees. Les "faibles doses" étaient celles qu'on distinguait déjà le mieux des deux autres classes, qui, par contre, ne se différenciaient pas très bien, entre elles. Il n'est donc pas tellement étonnant de retrouver un nombre d'erreur assez faible quand on compare ce jeu de données aux deux autres, réunis alors en une seule et même classe. Les données "formaldéhyde" se discriminent aussi relativement bien, même si le nombre d'erreurs reste relativement important : environ 5 erreurs sur 29 jeux de données.

Regardons maintenant quels sont les individus mal classés, pour les situations où nous n'avons pas trop d'erreurs. Dans le cas de la discrimination entre "faibles doses" et "formaldéhyde", les méthodes rOPG, DQDA et KNN, classent mal un individu de la classe "formaldéhyde", un des deux individus de ce jeu de données ayant été le moins exposé. La méthode DLDA classe mal l'individu "faibledose1", l'individu qui était déjà mal classé dans l'étude précédente (fortes doses contre faible doses d'irradiation). Pour la classification entre "faibles doses" et "non irradiés", cet individu est également systématiquement mal classé.

Conclusion du troisième chapitre

Nous avons proposé une méthode de discrimination pour la classification de données de biopuces à ADN fondée principalement sur une méthode semi-paramétrique de réduction de dimension, basée sur la maximisation d'un critère de vraisemblance locale dans les modèles linéaires généralisés. Cette méthode fournit une réponse satisfaisante pour la classification de ces données qui relèvent du cadre statistique de la grande dimension, le nombre de covariables étant très grand devant le nombre d'échantillons.

D'un point de vue théorique, nous avons montré la convergence de l'estimateur de la direction de projection obtenu à l'étape A de notre procédure.

D'un point de vue pratique, les résultats obtenus sur des données réelles sont stables et tout à fait satisfaisants, et ce malgré le modèle en indice simple utilisé. Pour tous les jeux de données traités ici, la méthode GSIM s'est avérée performante et fournit des résultats comparables ou meilleurs à ceux obtenus via des approches plus classiques.

Perspectives

▶ **Simulation de données et biopuces et normalisation.**

Les perspectives de ce chapitre se situent essentiellement dans la partie simulation de données. Des progrès peuvent être faits, notamment en prenant en compte plus de paramètres pour générer un jeu de données. Ainsi les bruits additifs et multiplicatifs inhérents à un jeu de données devraient être générés aléatoirement, à chaque nouvelle simulation, selon une loi estimée à partir de données réelles. Il serait aussi intéresant d'étudier comment choisir les paramètres c_{ind} et c_{repr} qui fixent en fait l'écart entre le nuage des gènes non exprimés et les nuages des différentiellement exprimés. Le problème c'est que pour améliorer le modèle de simulation il faudrait disposer d'une série de données pour lesquelles on connaît les différentiellements exprimés afin de pouvoir étudier les paramètres du modèle et les courbes de forme séparement pour chaque catégorie de gène (non exprimés, induits, réprimés), ce qui est assez irréaliste.

▶ **Détection des gènes différentiellement exprimés.**

Dans ce chapitre, il y a encore plusieurs voies à explorer, et ce à plusieurs niveaux. Les méthodes proposées ici donnent des résultats satisfaisants sur les simulations, mais sur un vrai jeu de données tout est devenu plus compliqué. Il faut sans doute envisager de nouvelles statistiques de test, peut-être mieux adaptées à ce type de données bien spécifiques. On peut aussi s'en doute réutiliser l'idée développée pour la statistique de Turkheimer *et al.* , en essayant d'estimer un facteur d'échelle, par rééchantillonnage par exemple, de telle façon qu'ensuite on ait de bonnes propriétés pour la statistique de test standardisée. Un travail important peut aussi être fait sur le calcul des p-valeurs à partir des statistiques de test, et éventuellement des données. Enfin, on pourrait aussi considérer d'autres taux d'erreur à contrôler, comme le $pFDR$ (positive FDR) proposé par Storey [11] ou le "local FDR" comme le font Aubin *et al.* dans [7].

▶ **Réduction de dimension et classification supervisée**.

D'un point de vue théorique, il serait intéressant de reprendre la démonstration de la convergence de l'estimateur de la direction de projection pour obtenir un taux

de convergence. Une autre perspective de ce travail sera d'appliquer la méthode GSIM à des données simulées pour vérifier la qualité de la méthode dans le cas standard, c'est-à-dire quand $n >> p$.

D'un point de vue pratique, d'autres extensions de la méthode sont déjà en cours pour permettre une meilleure classification des données réelles. Tout d'abord, les échantillons observés appartenant souvent à plus de deux classes, il serait particulièrement intéressant d'étendre la méthode au cas multiclasse. Des travaux ont déjà commencé à ce sujet et les résultats sont tout à fait encourageants. De plus, dans les données réelles, il est rare qu'on ne s'intéresse qu'à deux classes d'individus. Ainsi, dans les jeux de données étudiées précédemment, on a vu que "Leukemia" peut aussi se diviser en trois classes, et pour la dernière étude sur les données de l'Institut Curie, il aurait été naturel de traiter le problème avec une modélisation en trois classes. Indépendamment du nombre de classes, on peut espérer de meilleurs résultats pour la méthode GSIM_λ en indice multiple, c'est-à-dire étendue à plusieurs directions de projection. Ceci fait également l'objet de travaux en cours.

Liste des figures

Liste des tables

Liste des algorithmes

Bibliographie

1. Felix Abramovich, Anestis Antoniadis, Theofanis Sapatinas et Brani Vidakovic. Optimal testing in functional analysis of variance models. Mars 2002. (33)

2. Felix Abramovich et Yoav Benjamini. Thresholding of wavelet coefficients as multiple hypotheses testing procedures. Dans A. Antoniadis (ed.), *Wavelets and Statistics*, volume 103, pages 5–14. Springer Verlag Lecture Notes in Statistics, 1995. (113)

3. Felix Abramovich, Yoav Benjamini, David Donoho et Iain Johnstone. Adapting to Unknown Sparsity by controlling the False Discovery Rate. Rapport technique, Department of Statistics, Stanford University, Stanford, 2000. (accessible à http ://www-stat.stanford.edu/∼imj). (93, 105, 112)

4. Uri Alon, Naama Barkai, Daniel A. Notterman, Kurt Gish, Suzanne Ybarra, David H. Mack et Arnold J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, 96(12), 6745–6750, 1999. (185)

5. Umberto Amato, Anestis Antoniadis et Gérard Grégoire. Independant component discriminant analysis. *International Mathematical Journal*, pages 735–753, 2002. (52)

6. Anestis Antoniadis, Sophie Lambert-Lacroix et Frédérique Leblanc. Effective Dimension Reduction Methods for Tumor - Classification using gene Expression Data. *Bioinformatics*, 19(5), 563–570, 2003. (157)

7. Julie Aubert, Bar-Hen Avner, Jean-Jacques Daudin et Stéphane Robin. Determination of the differentially expressed genes in microarray experiments using local fdr. *Bioinformatics*, 5(125), 2004. (199)

8. Yoav Benjamini et Yosef Hochberg. Controlling the false discovery rate : a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57, 289–300, 1995. (82, 83, 93, 95, 113, 124, 125, 128)

9. Yoav Benjamini et Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29 (4), 1165–1188, 2001. (93, 95, 124)

10. Raymond J. Carroll, Jianqing Fan, Irène Gijbels et Matt P. Wand. Generalized partially linear single index models. *Journal of the American Statistical Association*, 92, 477–489, 1997. (174)

11. Storey John D. A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B*, 64, 479–498, 2002. (199)

12. Ralph B. D'Agostino et Edward E. Cureton. A class of simple linear estimators of the standard deviation of the normal distribution. *Journal of the American Statistical Association*, 68, 207–210, 1973. (88)

13. David L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3), 613–627, 1995. (121)

14. David L. Donoho et Iain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3), 425–455, 1994. (121, 123)

15. Sandrine Dudoit, Jane Fridlyand et Terence P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Stastical Association*, 97, 77–87, 2002. (168, 185)

16. Sandrine Dudoit, Juliet P. Shaffer et Jennifer C. Boldrick. Multiple Hypothesis Testing in Microarray Experiments. *Statistical Science*, 18(1), 71–103, 2003. (82)

17. Bradley Efron, Robert Tibshirani, John D. Storey et Virginia Tusher. Empirical Bayes Analysis of a Microarray Experiment. *Journal of the American Stastical Association*, 96(456), 1151–1160, 2001. (93)

18. Jianqing Fan et Irène Gijbels. *Local Polynomial Modelling and its Applications*. Chapman and Hall, London, 1996. (160, 168, 172)

19. Jianqing Fan, Nancy Heckman et Matt P. Wand. Local polynomial kernel regression for generalized linear models and quasi-likelihood functions. *Journal of the American Stastical Association*, 90, 141–150, 1995. (172)

20. Gersende Fort et Sophie Lambert-Lacroix. Classification using Partial Least Squares with Penalized Logistic Regression. *Bioinformatics*, 21(7), 1104–1111, 2005. (168)

21. Todd R. Golub, Donna K. Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P. Mesirov, Hilary Coller, Mignon L. Loh, James R. Downing,

Mark A. Caligiuri, Clara D. Bloomfield et Eric S. Lander. Molecular Classification of Cancer : Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286(5439), 531–537, 1999.

22. G. K. Golubev. Reconstruction of Sparse Vectors in White Gaussian Noise. *Problems of Information Transmission*, 38(1), 75–91, 2002. (93, 115)

23. Peter J. Green. Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and some Robust and Resistant Alternatives. *Journal of the Royal Statistical Society, Series B*, 46(2), 149–192, 1984. (161)

24. Per Christian Hansen et Dianne P. O'Leary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM Journal on Scientific Computing*, 14(6), 1487–1503, 1993. (167)

25. Yosef Hochberg et Ajit C. Tamhane. *Multiple Comparison Procedures*. Wiley, 1987. (82)

26. Iain M. Johnstone et Bernard W. Silverman. Needles and straw in haystacks : Empirical Bayes estimates of possibly sparse sequences. *The Annals of Statistics*, 32(4), 1594–1649, 2004. (128, 129)

27. Iain M. Johnstone et Bernard W. Silverman. Empirical Bayes selection of wavelet thresholds. *The Annals of Statistics*, 33, 2005. A paraître. (128, 129, 133)

28. Charles Kooperberg, Thomas G. Fazzio, Jeffrey J. Delrow et Tsukiyama Toshio. Improved Background Correction for Spotted DNA Microarrays. *Journal of Computational Biology*, 9(1), 55–66, 2002. (22)

29. Pierre S. de Laplace. *Deuxième supplément à la Théorie Analytique des Probabilités*. Courcier, Paris, 1818. (102)

30. Christian Léger et Joseph P. Romano. Bootstrap Choice of Tuning Parameters. *Annals of the Institute of Statistical Mathematics*, 42(4), 709–735, 1990. (88)

31. Peter McCullagh et John A. Nelder. *Generalized Linear Models. 2nd ed.* New-York : Chapman & Hall, 1989. (159)

32. David Pollard. Asymptotics for least absolute deviation regression estimators. *Econometric Theory*, 7, 186–199, 1991. (174)

33. Katherine S. Pollard et Mark J. van der Laan. Multiple testing procedures for gene expression data : an investigation of test statistics and null distributions with consequences for the permutation test. Rapport technique, University of California, Berkeley, 2002. (86, 90)

34. Katherine S. Pollard et Mark J. van der Laan. Resampling-based Methods for Identification of Significant Subsets of Genes in Expression Data. Rapport technique 121, University of California, Berkeley, 2002. (86, 90)

35. R Development Core Team. *R : A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2004. ISBN 3-900051-00-3. (134)

36. J. 0. Ramsay et Bernard W. Silverman. *Functional Data Analysis*. New York : Springer-Verlag, 1997. (33)

37. Brian D. Ripley. *Spatial Statistics*. Wiley, 1981. (24)

38. David M. Rocke et Blythe Durbin. A Model for Measurement Error for Gene Expression Arrays. *Journal of Computational Biology*, 8, 559–567, 2001. (55, 56)

39. Ester Samuel-Cahn. Combining unbiased estimators. *The American Statistician*, 48(1), 34–36, 1994. (87)

40. Burkhardt Seifert et Theo Gasser. Finite-Sample Variance of Local Polynomials : Analysis« and Solutions. *Journal of the American Statistical Association*, 91(433), 267–275, 1996. (165)

41. Juliet P. Shaffer. Multiple hypothesis testing. *Annual Review of Psychology*, 46, 561–584, 1995. (82)

42. Xiaotong Shen, Hsin-Cheng Huang et Noel Cressie. Nonparametric Hypothesis Testing for a Spatial Signal. *Journal of the American Stastical Association*, 97(460), 1122–1140, 2002. (125)

43. Charles J. Stone, Mark Hansen, Charles Kooperberg et Young K. Truong. Polynomial splines and their tensor products in extended linear modeling (with discussion). *Annals of Statistics*, 25, 1971–1470, 1997. (33)

44. Federico Turkheimer, Karen Pettigrew, Louis Sokoloff et Kathleen Schmidt. A minimum variance adaptive technique for parameter estimation and hypothesis testing. *Communications in Statistics-Simulation and Computation*, 28, 931–956, 1999. (88)

45. Federico Turkheimer, Karen Pettigrew, Louis Sokoloff, Carolin B. Smith et Kathleen Schmidt. Selection of an Adaptative Test Statistic for Use with Multiple Comparison Analyses of Neuroimaging Data. *Neuroimage*, 12, 219–229, 2000. (87)

46. Federico Turkheimer, Louis Sokoloff, Karen Pettigrew et Kathleen Schmidt. A new general purpose minimum variance algorithm for the analysis and modeling of biological data. *Neuroimage*, 3, S101, 1996. (88)

47. Peter H. Westfall et S. Stanley Young. *Resampling-Based Multiple Testing*. Wiley, 1993. (90)

48. Yingcun Xia, Howell Tong, W.K. Li et Li-Xing Zhu. An adaptive estimation of dimension reduction space. *J. R. Stat. Soc., Ser. B*, 64(3), 363–410, 2002. (157, 166, 168)

49. Yee H. Yang, Sandrine Dudoit, Percy Luu et Terence P. Speed. Normalization for cDNA Microarray Data. Dans *SPIE BiOS*. San Jose, California, Janvier 2001. (29, 32)

50. Lu Zhan-Qian. Multivariate locally weighted polynomial fitting an partial derivative estimation. *Journal of Multivariate Analysis*, 59, 187–205, 1996. (166)

Analyse statistique des données issues des biopuces à ADN — Résumé.

Cette thèse est consacrée à l'analyse statistique des données issues des biopuces à ADN. Nous nous intéressons ici à trois problématiques liées aux données du transcriptôme.

Dans un premier chapitre, nous étudions le problème de la normalisation des données dont l'objectif est d'éliminer les variations parasites entre les échantillons des populations pour ne conserver que les variations expliquées par les phénomènes biologiques. Nous présentons plusieurs méthodes existantes pour lesquelles nous proposons des améliorations. Pour guider le choix d'une méthode de normalisation, une méthode de simulation de données de biopuces est mise au point.

Dans un deuxième chapitre, nous abordons le problème de la détection de gènes différentiellement exprimés entre deux séries d'expériences. On se ramène ici àun problème de test d'hypothèses multiples. Plusieurs approches sont envisagées : sélection de modèles et pénalisation, méthode FDR basée sur une décomposition en ondelettes des statistiques de test ou encore seuillage bayésien.

Dans le dernier chapitre, nous considérons les problèmes de classification supervisée pour les données de biopuces. Pour remédier au problème du "fléau de la dimension", nous avons développé une méthode semi-paramétrique de réduction de dimension, basée sur la maximisation d'un critère de vraisemblance locale dans les modèles linéaires généralisés en indice simple. L'étape de réduction de dimension est alors suivie d'une étape de régression par polynômes locaux pour effectuer la classification supervisée des individus considérés.

Mots clés : biopuces, test d'hypothèses multiples, sélection de variables, modèles linéaires généralisés, régression semi-paramétrique.

* * *

Statistical analysis of microarray data — Abstract.

This dissertation is dedicated to the statistical analysis of microarray data. We consider three issues linked to the transcriptome data.

In the first chapter, we study the problem of data normalisation; its purpose is to eliminate the parasite differences between samples, so as to retain only those variations that are due to biological phenomena. We present several existing normalisation methods and we propose improvements for some of them. Furthermore, in order to guide the choice among those methods, we develop a procedure to simulate microarray data.

In the second chapter, we deal with the detection of differentially expressed genes between two series of experiments, an issue that we assimilate to a multiple hypothesis testing problem. Several approaches are studied: model selection and penalty, FDR method based on a wavelet decomposition of the test statistics and Bayesian thresholding.

In the last chapter, we consider the problem of supervised classification of microarray data. To cope with the high-dimensionality issue, we develop a semiparametric method for dimension reduction, based on the maximisation of a local likelihood criterion in generalized linear single-index models. The dimension reduction step is then followed by a local polynomial regression step, in order to perform the supervised classification of the given individuals.

Key words: microarrays, multiple hypothesis testing, variable selection, generalized linear models, semiparametric regression.