NNT : 2016SACLE056

Thèse de doctorat de l'Université Paris-Saclay préparée Université d'Evry

Ecole doctorale n°577
Structure et dynamique des systèmes vivants
Spécialité de doctorat : Sciences de la vie et de la santé

par

Mme Morgane Pierre-Jean

Développement de méthodes statistiques pour l'analyse du nombre de copies d'ADN en cancérologie

Thèse présentée et soutenue à Évry, le 2 Décembre 2016.

Composition du Jury :

M. F. Radvanyi, Directeur de Recherche, Institut Curie, (Président du jury)
Mme. A-L. Boulesteix, Professeur, LMU- München, (Rapporteur)
M. J-P. Vert, Directeur de Recherche, Mines ParisTech/Institut Curie, (Rapporteur)
Mme N. Zhang, Professeur Associé, University of Pennsylvania, (Rapporteur)
M. C. Dalmasso, Maitre de Conférence, Université d'Evry Val d'Essonne, (Examinateur)
Mme. G. Marot, Maitre de Conférence, Université Lille 2/INRIA, (Examinatrice)
Mme. C. Matias, Directrice de Recherche, CNRS, (Directrice de thèse)
M. P. Neuvial, Chargé de Recherche, CNRS, (Co-Directeur de thèse)

À Jean-Baptiste et Marius,
To Cassandre

"Essentially, all models are wrong,
but some are useful"
George E. P. Box,
Empirical Model-Building and Response Surfaces

Contents

Remerciements

Je tiens très sincèrement à remercier Pierre pour avoir eu cru en moi durant ces 4 années. Pour m'avoir proposé un stage de M2 qui me correspondait parfaitement. Pour ta patience, ta disponibilité et bien évidemment ta bienveillance. Merci aussi pour ta compréhension dans mes choix de vie que d'autres personnes n'auraient sans doute pas acceptés, merci pour ta bonne humeur et tes conseils toujours très pertinents. Merci pour ton calme imperturbable, ta sérénité quand moi je paniquais même si tu as fini par déteindre un petit peu sur moi. Merci de m'avoir donné la chance de faire cette thèse ici, de m'avoir guidé tout au long de cette thèse. Mes mots seront donc sans doute insuffisants pour te montrer toute ma reconnaissance et le plaisir que j'ai eu à travailler avec toi.

Je voudrais ensuite remercier mes rapporteurs, Anne-Laure Boulesteix, Nancy Zhang et Jean-Philippe Vert qui ont bien voulu consacrer un peu de leur temps pour rapporter ma thèse. Merci à tous pour vos questions et remarques pertinentes qui ont fait avancer ce travail de thèse.

Merci à tous les membres du jury qui ont assisté à ma soutenance de thèse. Je voudrais remercier Mr. François Radvanyi, président du jury, d'avoir accepté de lire une thèse en statistique et pour son point de vue de biologiste. Je voudrais également remercier Cyril, d'avoir accepté de faire parti de ce jury et pour avoir été le premier utilisateur de mon premier package mais aussi pour avoir assisté à mes nombreuses répétitions... Merci pour ta gentillesse, tous tes conseils scientifiques mais aussi sportifs. Enfin, merci pour les pauses café matinales avant d'attaquer la journée qui vont beaucoup me manquer. Merci aussi Guillemette, d'avoir terminé cette aventure avec moi en faisant parti du jury. Je voudrais également te remercier de m'avoir intégrée à Lille pendant presqu'un an malgré la distance. Ce petit passage à Lille m'a permis de grandir et de découvrir de nouvelles choses. Enfin, merci Catherine d'avoir accepté d'être ma directrice de thèse et d'avoir réglé les problèmes administratifs quand j'étais dépassée, merci d'avoir assisté à mes comités de thèse et à la soutenance.

J'ai eu l'occasion de travailler avec plusieurs personnes au cours de ma thèse et je voudrais remercier Guillem pour ses conseils et sa disponibilité. C'est un plaisir de travailler avec toi. Merci aussi à Julien de m'avoir initiée au Lasso. Merci pour tes supports de cours qui m'ont fait gagner du temps. Merci pour ton aide très efficace quand j'étais bloquée. Je voudrais aussi remercier, Henrik Bengtsson pour la collaboration à San Francisco et m'avoir accueillie au café Strada et également Alain, de m'avoir accueillie à Lille pendant un an. Enfin, je voudrais remercier toute l'équipe de Curie pour sa bonne humeur et de m'avoir permis de travailler avec eux.

Merci Christophe, toi qui te soucis toujours du bien-être des membres du labo en particulier des thésards et pour accueillir comme il se doit chaque nouvel arrivant. Tu es vraiment quelqu'un d'attentionné puisque tu vois les choses bien avant les autres... Merci aussi pour les cafés du matin qui vont beaucoup me manquer.

Un grand merci à Margot pour supporter mon tempérament de râleuse tous les jours. C'était vraiment un plaisir de partager le bureau avec toi. Merci pour ta gentillesse, ta bonne humeur, merci pour ta patience quand je te pose des questions idiotes. Merci aussi Franck, pour tes conseils en python mais aussi pour les bonbons et la bonne humeur! Merci à tous les deux pour ce bureau d'accueil des âmes en perditions entre la cuisine et leurs bureaux.

Merci à Benjamin, d'avoir partagé mon bureau et qui m'a permis de faire une collaboration avec l'institut Curie. Tu n'étais pas beaucoup à Evry mais tu fais partie intégrante du laboratoire. Bon courage pour la fin! Merci également à Maurice parce que tu sais me taquiner sans pousser le bouchon trop loin, tu as toujours su te rendre disponible quand j'avais des problèmes d'informatique! Merci pour ta bonne humeur. Et tout comme Margot, pour partager mes humeurs de râleuse lorsque le RER est en panne.

Je remercie aussi Michèle qui est toujours aux petits soins pour les thésards et pour tout le monde, Carène pour ta bonne humeur et tes bonnes pâtisseries, tu m'as aussi beaucoup conseillé et je t'en remercie, merci aussi pour les petits cadeaux à Marius et de venir courir avec moi le midi. Claudine pour nous avoir fait rire malgré toi de nombreuses fois, tu égayais vraiment le laboratoire. Merci de m'avoir racompagnée chez moi quand j'avais des problèmes de RER et pour nos séances de course à pieds.

Merci aussi Anne-Sophie de m'avoir entendu râler sur les étudiants qui ne venait pas en cours et de m'avoir permis de trouver des solutions. Merci aussi Marie-Luce de veiller sur les services d'enseignement des thésards pour que ce soit le plus confortable possible. Merci aussi Abass de m'avoir fait confiance pour les TDs et de toujours te soucier si je vais bien quand tu passes devant le bureau. Merci Valérie pour ton efficacité en toutes circonstances. Une pensée pour toi, Bernard, merci pour tous les conseils que tu m'as donnés depuis le début de l'aventure à Stat et Génome. Tu es parti trop tôt et tu nous manques, j'aurais tellement aimé que tu sois là pour lire ce manuscrit.

D'abord Sarah, ensuite Virginie et maintenant Florent je vous ai pas mal embêtées il faut dire que vous étiez sur le chemin de l'imprimante et que j'aime bien discuter ... Le chemin entre mon bureau et l'imprimante pouvait être long... Merci pour ces longues discussions. Merci à toutes les deux d'être venues courir avec moi. Virginie et Florent bonne continuation.

Un très très grand merci à Jean-Michel et Magali pour les soirées jeux qui m'ont fait décompresser après de dures journées. Merci aussi pour le badminton qui m'a permis de nombreuses fois de me défouler même si je perdais à chaque fois. J'ai copié sur toi en faisant un bout de chou pendant ma thèse! Tu étais bien plus qu'un collègue et le bureau est bien vide depuis ton départ, il n'y plus personne pour me charier. Merci d'avoir pris des nouvelles pendant la rédaction du manuscrit et de me donner des nouvelles de votre famille.

Edith, la deuxième bretonne du labo pour un temps, merci pour tous tes conseils et pour avoir partagé le bureau avec toi pendant un temps. Pour toutes nos discussions, j'aurais sans doute le plaisir de travailler avec toi et j'en suis plus que ravie.

Plus généralement, merci à tous les membres du LaMME et plus particulièrement à l'équipe SG.

Merci au groupe d'Athlé de la génopole parce qu'entre deux pages de rédaction ça fait du bien de courir et de se défouler.

Une petite parenthèse ENSAI. Ma petite Perrine, l'aventure à commencer depuis bien longtemps (7 ans déjà). Je te remercie pour ton soutien depuis l'ENSAI, pour avoir été là dans les moments les plus durs. Sans toi le Master n'aurait pas eu le même goût. A nos soirées Mac do/grey's anatomy d'après exams et projets. Merci d'être là tout simplement et d'avoir fait le déplacement pour la soutenance. Merci aussi à Claire, Coralie, Alice, Joséphine, Graziella et Constance pour vos encouragements et pour nos soirées entre filles qui font du bien quand ça va mal et quand ça va bien. Merci à François Coquet qui m'a fait confiance depuis mon entrée à l'ENSAI en tant que cobaye.

Merci ma petite Chloé pour ta joie de vivre que je te connais depuis ma première année à l'université. Tu avais prédit lorsqu'on était en première année que je serai toujours à l'université dans 10 ans et bien tu vois tu n'as pas tout à fait tord.

Enfin, pour ma famille. Ma maman pour son éternelle gentillesse et qui prend toujours soin des autres avant de prendre soin d'elle. Merci ma petite Maman de m'avoir toujours poussée pour aller toujours plus haut même si c'était dur. Merci pour tout ce que nous donne chaque jour même si tu es loin. Merci pour ton amour inconditionnel et tous tes bons petits plats et les légumes que tu nous donnes quand on rentre le week end. Mon père qui je crois est très fier de ses filles peu importe les circonstances et qui m'a toujours encouragée dans tous mes choix de vie. Ma grande soeur et Louis pour vos énergies respectives. Ca n'a pas toujours été rose entre nous mais c'est la vie et je t'aime quand même grande soeur! Merci à ma grand-mère à qui j'ai pendant longtemps raconté ma vie à l'université. Merci Louise d'avoir consacré du temps pour corriger les dernières fautes d'anglais. Merci à toute ma famille et ma belle-famille en général pour m'avoir encouragée sans toujours savoir ce que je faisais. Je voudrais remercier particulièrement Annie, ma "jolie-maman", qui a toujours su trouver les mots et les petites attentions pour me réconforter dans les moments les plus difficiles. Merci pour ces discussions tard le soir quand on rentre en Bretagne et merci d'être venue m'écouter. Un merci tout particulier à ma tante Odette qui m'a toujours poussée à faire ce que je voulais et qui m'a toujours soutenue. Mon rayon de soleil plein d'énergie : ma petite Camillou, restes comme tu es et ne changes surtout pas.

Pour conclure ces lignes de remerciements, merci à mon fils Marius pour tous les moments magiques que tu me fais vivre au quotidien, pour ton sourire d'enfant et ton rire, pour tes câlins qui réconfortent après une dure journée. Un immense merci à Jean-Baptiste, mon amour depuis 9 ans qui me supporte et me fait rire et sans qui la vie serait beaucoup moins agréable... Impossible d'imaginer la vie sans toi. Merci de m'écouter râler pendant des heures, merci pour tes bras protecteurs, pour tes encouragements, merci d'avoir cru en moi. Merci de partager ma vie, merci de me donner autant de bonheur et de joie de vivre.

Chapter 1

General introduction

1.1 Context in molecular biology

All living beings are composed of cells with predefined genetic functions. The Deoxyribonucleic acid (DNA) is the molecule, located in all cells, that carries most of the genetic instructions. Thanks to these instructions, cells are able to grow, develop, function and reproduce. The DNA sequence can be seen as an oriented sequence of characters from an alphabet composed of only four different letters (A, T, C, and G). These letters are in fact molecules which are nitrogenous bases and where A is Adenine molecule, T is Thymine, C is Cytosine and G is Guanine. The DNA is arranged in chromosomes (Fig. 1.1). Each cell of the human body is generally composed of pairs of chromosomes, each inherited from one parent, and each existing in two DNA copies. The chromosomes are divided into 22 autosomal pairs and 1 sexual one (Figure 1.2). Each chromosome is composed of several genes with specific functions, where a gene may be defined as a part of the DNA sequence that contains instructions to allow the cell to function correctly [Alberts et al., 2010].

Genetic information exists at several levels that it is possible to measure: genome, epigenome, transcriptome, proteome and metabolome [Ritchie et al., 2015]. At the genome level, only the DNA molecule is studied in its original form. Epigenome studies focus on cellular and physiological phenotypic trait variations due to external or environmental factors. These variations can switch genes on and off and affect how cells express genes [Moore, 2015]. Transcriptome studies mainly focus on gene expressions and proteome studies focus on protein expressions. Metabolites are molecules that have various functions, including fuel generation, structure, signaling, stimulation and inhibition of enzymes, the catalysis activity of their own, defense, and interactions with other organisms. The metabolomic is the study of the chemical processes that involve the metabolites [Daviss, 2005].

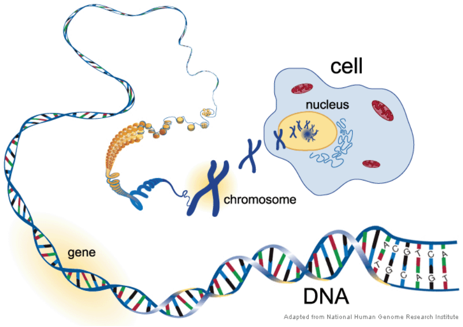


Figure 1.1 – From the cell to the DNA. (Property of US National Human Genome Research Institute)

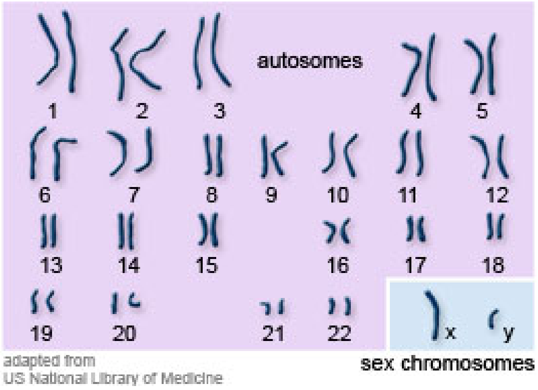


Figure 1.2 – 22 pairs of numbered chromosomes (called autosomes) plus one pair of sex chromosomes (the X and Y chromosomes). (Property of National Library of Medicine)

However, mutations frequently occur during cell life. Most of these mutations are repaired by the molecule itself but some are too severe to be repaired and produce serious damages at the level of the cell regulation. Common mutations include insertions and deletions of parts of the DNA sequence. When these mutations occur during cell life and are not repaired, they may be responsible for cancers. In the next section, we explain in detail the alterations that can occur in tumoral genomes.

1.2 Cancer

Cancer is a heterogeneous disease [Marusyk and Polyak, 2010] which is characterized by an abnormal proliferation of cells. The notion of heterogeneity in cancerology can be defined at least at two different levels:

- Differences between tumors of the same disease in different patients (inter-tumour heterogeneity)
- Differences between cancer cells within a single tumor of one patient (intra-tumour heterogeneity).

First, tumor cells are often located in a given tissue but can eventually spread to other parts of the body by the intermediate of the blood. The result of this phenomenon is metastatic cancer which may compromise complete remission of patients. Genetic and epigenetic differences found between the cancer cells within a single primary tumor could explain why some cells remain resistant to drugs and cause relapses [Wagle et al., 2011]. An important goal of current cancer research is to better understand the underlying mechanisms and to characterize genomic mutations within tumor cells in order to improve diagnosis and prognosis and eventually provide best-adapted drugs for each patient. In this way, the Cancer Genome Atlas Research Network (TCGA [Weinstein et al., 2013]) and also the International Cancer Genome Consortium (ICGC [Zhang et al., 2011])have led inter-tumor heterogeneity studies. Inter-tumor studies collect one sample per patient, typically from solid tumors, and perform analysis to compare alterations profiles across the patient.

Studies are performed at several levels of the genomic information. But, it is well established that cancer cells come from damages at the levels of genes responsible for the regulation of the cell division [Cooper, 2000], and as a result, several mechanisms are modified. Cancer cells often originate from different subclones harboring specific mutational, expression or DNA copy number profiles [Navin et al., 2010, Rasmussen et al., 2011, Oesper et al., 2013]. The intra-tumor heterogeneity challenges our understanding of the mechanisms of oncogenesis and disease progression. The inter-tumor heterogeneity studies are older and started by study expression data of tumor from several patients [Badea, 2008, Inamura et al., 2005, Collisson et al., 2011]. Dissecting the genetic heterogeneity of tumors and cancers could help to refine the molecular subtyping of tumors, predicting drug resistance and building personalized treatment options [Marusyk et al., 2012].

In this thesis, we focus on the changes at DNA copy number level. In fact, it is well known that gains and losses of parts or entire chromosomes [Hanahan and Weinberg, 2011] can be observed in cancer (an example of karyotype from bladder cancer is shown in Fig. 1.3). These modifications can directly or indirectly affect the gene expression levels. In consequence, some cell functions may be incorrectly regulated. For example, multiple oncogenes can be duplicated at DNA copy number levels and aggravate the proliferation of tumor cells. Generalities about the technologies developed to analyze DNA copy number alterations are detailed in the next section.

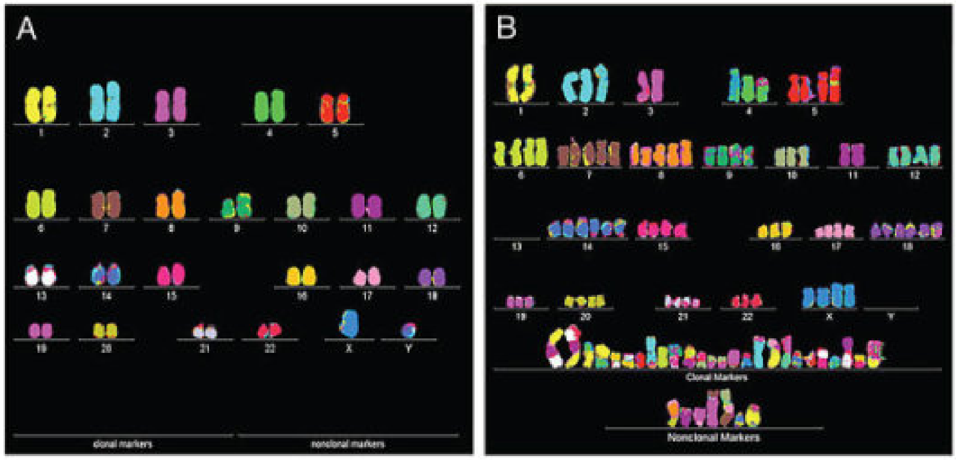


Figure 1.3 – A normal human karyotype (left) where each color corresponds to a chromosome (humans have precisely two copies of each chromosome). A bladder cancer cell karyotype (right) where extra copies of some chromosomes are present (which characterize cancer cells). Source: https://www.sciencedaily.com/releases/2011/07/110726163519.html

1.3 Quantifying genomic features of tumor cells

To study the genome (not only in cancer research), high throughput molecular biological techniques have been developed in the last twenty years, and several efficient technologies have been designed to identify biomarkers involved in genetic diseases and cancers. DNA microarrays and high throughput sequencing technologies are two kinds of emergent technologies developed to quantify the various levels of genetic information. DNA microarrays and since ten years high throughput sequencing technologies (Whole genome sequencing (WGS) and Whole exome sequencing (WES)) are used to detect biomarkers involved in genetic diseases and cancers. In this section, we briefly explain the technologies that can be used in the context of cancerology to measure the DNA copy number in tumor cells.

1.3.1 DNA microarray data

DNA microarrays were first produced by the company Affymetrix in 1991 but they starting being used in 1995 [Schena et al., 1995]. Microarrays were first used to measure gene expression and more particularly to compare expression in tumor cells to expression in normal cells. Then, those to analyze DNA copy number have been designed at the same time in 1992 [Kallioniemi et al., 1992]. In 1998, microarrays were developed to genotype multiple regions of the genome (called loci) [Wang et al., 1998]. Data from two types of DNA microarrays have been studied in this thesis: Comparative Genomic Hybridization (CGH) arrays and Single nucleotide polymorphism (SNP) arrays.

CGH arrays have been developed to measure the total DNA copy number at predefined loci on the genome with a high-resolution scale. The principle is the following: DNA from a reference and a test are collected and labeled with two different fluorophores. Then, cleaning and scanning of the arrays have been performed in order to do image processing by an image analysis software (see Fig. 1.4). The signal obtained from these arrays after the image processing is summarized by the ratio between the amount of the test and reference DNA. An example of signals obtained from CGH arrays are presented in section 1.4.

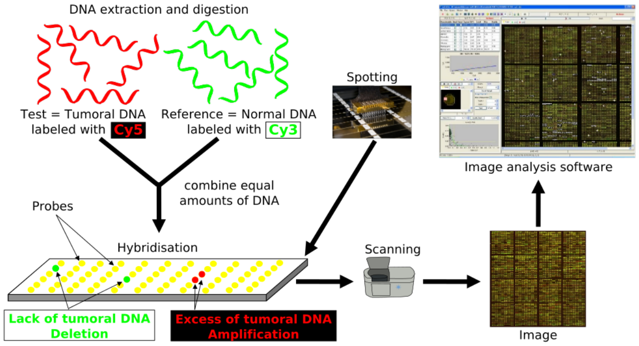


Figure 1.4 – Principle array-CGH (extract from [Barillot et al., 2012])

SNP arrays. A Single-Nucleotide Polymorphism (SNP) (pronounced snip) is defined by a variation of the DNA sequence occurring at a single genome position. It is characterized by a nucleotide - A, T, C, or G - which differs between members of a population (or between paired chromosomes in an individual). For instance, the DNA fragment sequence at the top of Fig. 1.5 is AAGCCTA and AAGCTTA for the DNA fragment sequence on the bottom. In most cases, we say that there are two alleles : C and T (denoted arbitrarily by allele A and B). In most cases, SNPs only have two alleles.

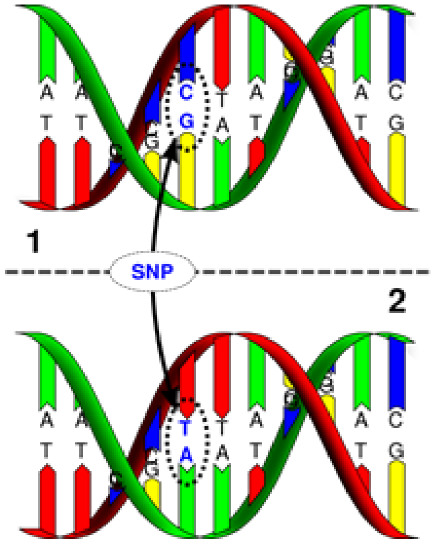


Figure 1.5 – What is a SNP? Source: http://www.dnabaser.com/articles/SNP/SNP-single-nucleotide-polymorphism.html

SNP arrays measure allele quantities at a large number of predefined loci. The SNP arrays are used to study small variations between whole genomes [Visscher et al., 2012]. Indeed, it is possible to compare if a SNP appears more frequently in a population which suffers from a particular disease than in a healthy population. However, SNP arrays can also be used to study genetic abnormalities in cancer. In fact, by measuring intensities of alleles at predefined loci, it is possible to deduce the DNA copy number but also the genotypes AA or BB (the homozygous) or AB (the heterozygous) of each locus.

Formally, for each $j = 1, \ldots, J$, let us denote by θ_{Aj} and θ_{Bj} the signal intensities measured at SNP j for alleles A and B, respectively. θ_{Aj} and θ_{Bj} are proportionate to the allele quantity.

We define the first dimension of the signal by the total DNA copy number that is proportionate to SNP j by $\theta^t_j = \theta^t_{Aj} + \theta^t_{Bj}$ (sum of quantity of allele A and allele B in the tumor sample denoted t). If a reference sample is available, it is possible to measure the total DNA copy number in the tumor sample by :

$$c_j = 2 \times \frac{\theta^t_j}{\theta^r_{Aj} + \theta^r_{Bj}} \tag{1.1}$$

where θ^r_{Aj} and θ^r_{Bj} are the intensities at the same location in the reference sample denoted r.

The second dimension of the signal from SNP arrays is the B allele fraction (BAF). The BAF is defined at SNP j by:

$$b_j^t = \frac{\theta_{Bj}^t}{\theta_{Aj}^t + \theta_{Bj}^t} \tag{1.2}$$

and is between 0 and 1.

For example, in a normal cell with two DNA copies, BAF is close to 0 if only allele A is observed, 1 if only allele B is observed, and 0.5 if both of allele A and B are observed. Thereafter, the heterozygous in the *germline* refers to the genotype in the normal cells of the patient.

1.3.2 High-throughput sequencing data

DNA sequencing is usually used to determine the sequence from single gene to entire genomes. This technology provides ordered sequences of nucleotides present at the level of DNA or Ribonucleic acid (RNA) and allows us to detect mutations typically linked to diseases at a higher resolution than DNA microarrays.

Whole genome sequencing (WGS)

High-throughput sequencing (HTS) is a recent technology which allows us to sequence DNA and RNA much quicker and cheaper than Sanger sequencing. Whole genome sequencing consists in collecting DNA sample and then determining the identity of the nucleotides (A, T, C, G) that compose the genomes of a living being. The first step is to cut the whole DNA sequence into short fragments between 10bp and 100bp. After a step of replication, the start and the end of the all replicated fragments are sequenced and read (Fig. 1.6). Then, each fragment, called read, is aligned to a reference sequence. The depth of sequencing is the number of times a nucleotide is read during the process. For instance, a depth of 100x means that in mean the number of reads with a part aligned at this position is close to 100.

Whole exome sequencing (WES)

However, the exons (Fig. 1.7) represent only 1,5% of the human genome [Venter et al., 2001] are short DNA sequences that lead to transcripts. In addition, the disease-causing variants are in majority in exons. Before performing the sequencing, the coding portion of the genome is *captured* and then sequenced (see Fig. 1.8 for more details). As a result, it is possible to sequence more deeply a small part of the genome for a same cost than the whole genome.

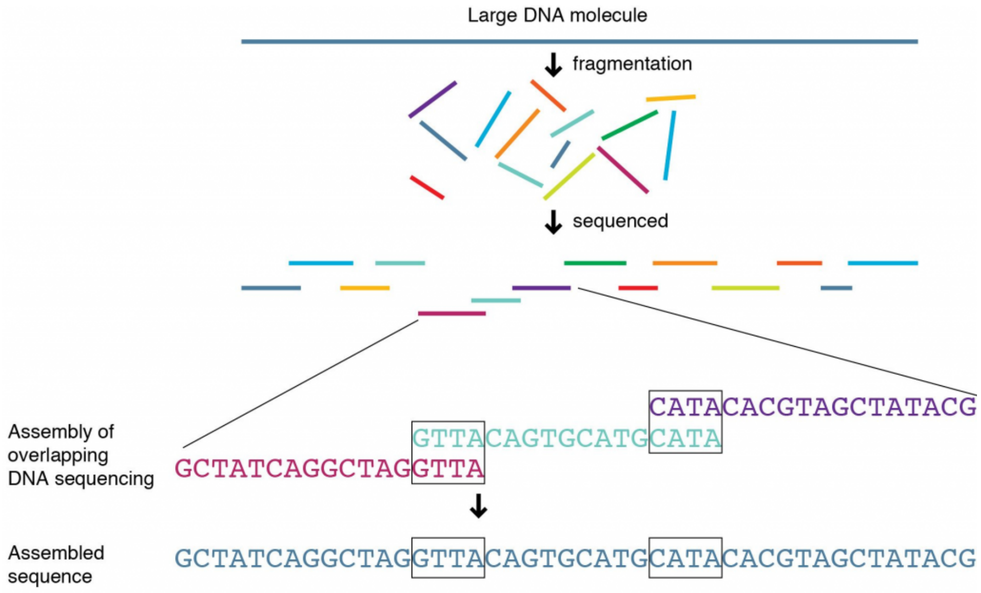


Figure 1.6 – Whole genome sequencing (Property of US National Human Genome Research Institute)

In cancer research, this technology allows us to recover both DNA copy number alterations,in addition to mutations which can drive the evolution of cancer and somatic events like translocations. In order to be able to analyze this kind of data, we have to adapt our statistical methods.

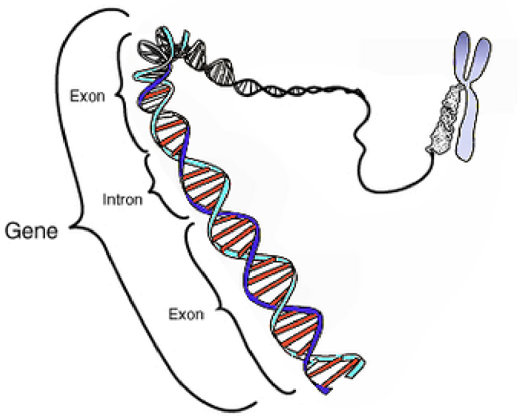


Figure 1.7 – Gene (Property of US National Human Genome Research Institute)

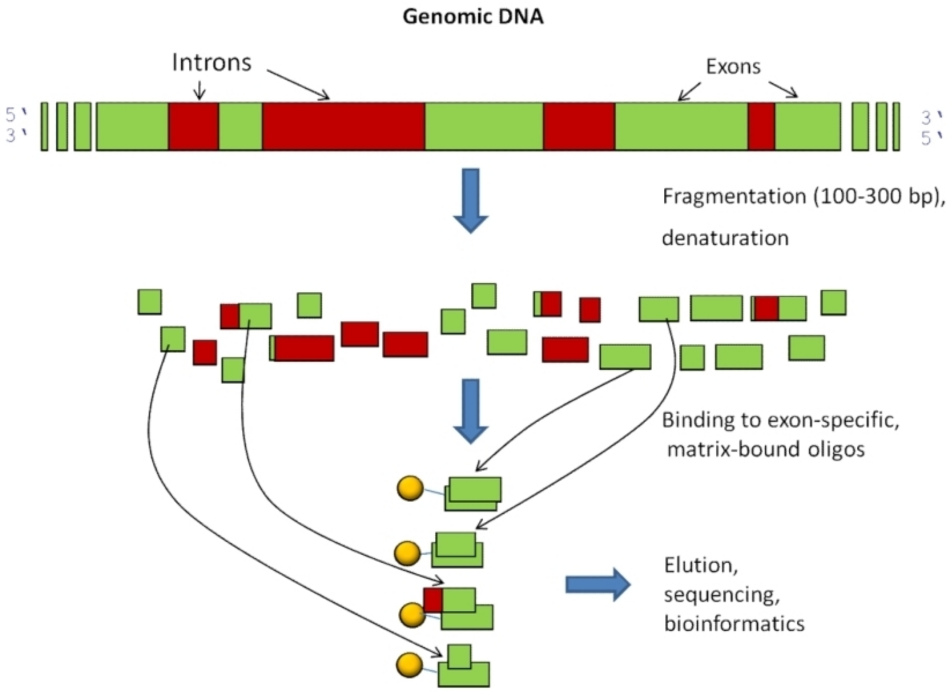


Figure 1.8 – Whole exome sequencing Source: http://biol1020-2012-2.blogspot.fr/2012/08/a-new-breast-cancer-susceptibility-gene.html

1.4 DNA copy number data in cancerology

In this thesis, we focused on genomic alterations in tumor cells at the level of the DNA copy number. This section describes the notations and variables used along the manuscript.

1.4.1 Total copy number and B allele fraction

As explained in section 1.2, in tumor cells, parts of a chromosome of various sizes (from kilobases to a chromosome arm) may be deleted, or copied several times. As a result, DNA copy numbers in tumor cells are piecewise constant along the genome. For illustration, Figure 1.9 displays an example of copy number signals that may be obtained from SNP-array data. Red vertical lines represent change points. In this particular example, the first region [0-2200] is normal, the second one [2200-6100] is a region where one of the parental chromosomes has been duplicated, and the third one [6100-10000] is a region of uniparental disomy called also Copy neutral loss of heterozygosity (cn-LOH), that is, a region where one of the parental chromosomes has been duplicated and the other one deleted. The top panel represents estimates of the Total copy number (TCN) (denoted by c). The bottom panel represents estimates of BAF (denoted by b). We refer to section 1.3.1 for an explanation of how these estimates may be obtained and to [Neuvial et al., 2011] for normalization of these quantities. In the normal region [0-2200], the total copy number is centered around two copies and allelic ratios have three modes centered at 0, 1/2 and 1. These modes correspond to homozygous SNPs AA ($b = 0$) and BB ($b = 1$), and heterozygous SNPs AB ($b = 1/2$). We note that in the second region where the tumor has 3 copies of DNA, the average observed signal is substantially below the true copy number. This is due to the presence of normal cells in the "tumor sample", a phenomenon known as *normal contamination* which shrinks the observed signals toward two copies of DNA. We can refer to [Neuvial et al., 2011] for a more detailed explanation of this phenomenon and other sources of non-calibration in DNA copy number signals, such as the ploidy of the tumor. One important observation is that change points occur at the same position in both dimensions. This is explained by the fact that a change in only one of the parental copy numbers is reflected in both c and b. Therefore, it makes sense to analyze both dimensions of the signal jointly in order to identify change points. In the following, we denote by J the number of loci and respectively by c_j and by b_j the total copy number and the B allele fraction at the locus j for all $j = 1, \dots, J$.

Note that, the second dimension (BAF) makes it possible the detection of cn-LOH events from a normal region. This corresponds to the third region in Fig. 1.9). Biolog-

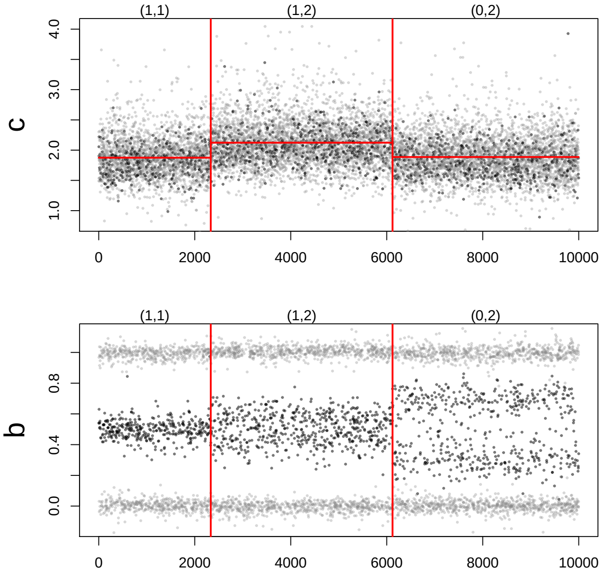


Figure 1.9 – Example SNP array data. Total copy numbers (c), allelic ratios (b) along 10,000 genomic loci. Red vertical lines represent change points, and red horizontal lines represent mean signal levels between two change points. SNPs that are heterozygous in the germline are colored in black; all of of the other loci are colored in gray.

ically, one parental copy has been lost when the other parental copy has been gained. Parental copy number estimation is detail in section 1.4.3.

1.4.2 DoH transformation

In order to facilitate the separation between the different altered regions (called mathematically segmentation), allelic ratios (b) are generally transformed into unimodal signals, as originally proposed in [Staaf et al., 2008]. This transformation is motivated by the fact that allelic ratios can be symmetrized ("folded") and that SNPs that are homozygous in the germline (these SNPs are plotted in gray in Figure 1.9) can be discarded as they carry very little information about copy-number changes. Following [Bengtsson et al., 2010], we define the Decrease of heterozygosity (DoH) as :

$$d_j = 2\left|b_j - \frac{1}{2}\right| \tag{1.3}$$

only for SNPs that are heterozygous in the germline, which is essentially a rescaled version of the "mirrored/folded BAF" defined by [Staaf et al., 2008]. After this trans-

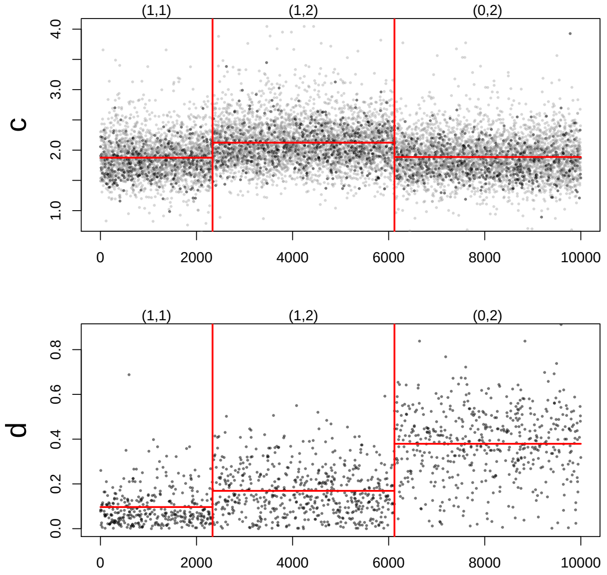


Figure 1.10 – Example SNP array data along 10,000 genomic loci, after transformation of allelic ratios (b) into decrease in heterozygosity (d), following [Bengtsson et al., 2010, Staaf et al., 2008]. Red vertical lines represent change points, and red horizontal lines represent mean signal levels between two change points. SNPs that are heterozygous in the germline are colored in black; all of of the other loci are colored in gray.

formation, DNA copy numbers can be considered as a bivariate, piecewise-constant signal, as illustrated by Figure 1.10.

It should be emphasized at this stage that because the proportion of heterozygous markers among SNPs is generally of the order of 1/3 for a given sample, the number of informative markers is several times larger for c than for d.

1.4.3 Parental copy number computation

[Neuvial et al., 2011] proposes to estimate the parental copy numbers (maternal and paternal copies) from the DoH estimation described in the previous section. Considering a SNP j which is heterozygous in the germline, the minor and major copy numbers at j are defined as the smallest and the largest of the two parental chromosomes. They can be estimated as:

$$\begin{cases} c^1_j = c_j\,(1-d_j)\,/2 \\ c^2_j = c_j\,(1+d_j)\,/2 \end{cases} \tag{1.4}$$

where c_j is still the total copy number at j, $d_j = 2|b_j - 1/2|$ is still DoH (see Eq.

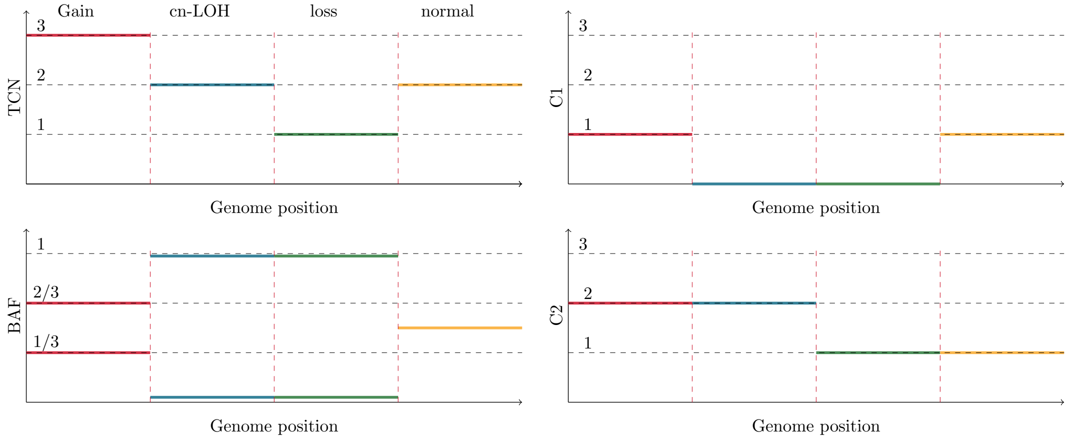


Figure 1.11 – TCN, BAF and minor and major copy number representations along the genome

1.3) . By definition, minor and major copy number have the following nice properties: $c_j = c_j^1 + c_j^2$, $d_j = (c_j^2 - c_j^1)/c_j$ and $c_j^1 \leq c_j^2$. The interpretation in terms of Loss of Heterozygosity (LOH) is also very simple : for instance a minor copy close to 0 corresponds to LOH alteration and $c_j^1 = c_j^2$ corresponds to allelic balance.

The list below describes the common different copy number states in terms of minor and major copy number denoted by the vector (c_1, c_2), where c_1 corresponds to the minor copy number, and c_2 corresponds to the major copy number [Neuvial et al., 2011].

- (1,1): normal (one copy from each parent)
- (0,1): hemizygous deletion (loss of one parental copy)
- (0,2): copy-neutral LOH (loss of one parental copy and gain of the other)
- (1,2): single copy gain

A graphical representation without noise for this four common types of alterations is shown on Figure 1.11.

1.4.4 Features DNA copy number data

It is clear by looking at Fig. 1.9 and 1.10 that DNA copy number profiles have particular features. The first one is that DNA copy number data sin tumor cells are piecewise constant along the genome. Then, the second one shared generally by genomic data is the high dimension. As we said in the section 1.3, it is possible to quantify genomic information at a large number of loci along the genome. Therefore, nowadays, microarrays contains around 10^6 observations and HTS can reach around the billion$3 * 10^9$ observations (a whole human genome).

In addition, tumor samples are not composed of only one type of cells but several, indeed it is usual that there is a contamination by a non-negligible proportion of normal cells when the sample is taken off. This induces difficulties to identify the altered regions 1.10.

1.5 Statistical and bioinformatical issues

The two technologies (microarrays and sequencing) described in the section 1.3 require bioinformatical and statistical methods at several levels. Indeed, several artifacts from microarrays and HTS may disturb the estimations of the DNA copy numbers. These experimental artifacts lead to systematic biases that it is necessary to correct. For instance, for CGH-arrays, the most common is the spatial artifact. Indeed, boundary effects but also regional shifts and systematic variations are often observed [Reimers and Weinstein, 2005]. For HTS, the most common bias is due to the percentage of GC-content that influences the read alignment on the reference sequence [Benjamini and Speed, 2012]. These issues are often dealt with bioinformatic methods but also statistical models.

Microarray and sequencing data can be used to explore several fields. For instance, [Beerenwinkel et al., 2014] have recently reviewed mathematical models for cancer evolution. Indeed, cancer can be seen as an evolutionary process with specific features. The tumors display an abnormal copy number of chromosomes, an elevated mutation rate, and several rearrangements of chromosomes for instance. Therefore, several phylogenetic methods that take into account particular features of tumor have been developed to study the clonal evolution of cancers [Chowdhury et al., 2013, Greenman et al., 2012]. Recently, an evolutionary study of ovarian cancers has shown a correlation between genetic heterogeneity, patient survival, and drug resistance [Schwarz et al., 2014].

In this thesis, we focus on the detection of the copy number alterations that can be observed in tumor cells. The improving of the detection of the alterations can lead to better understand the tumor evolution by integrating the discoveries in the phylogenetic models cited previously. Detecting alterations can also highlight genetic biomarkers linked to the patient survival or the drug resistance as in [Schwarz et al., 2014, Jiang et al., 2016].

To discover genomic biomarkers, it is necessary to develop statistical methods able to deal with features of microarrays and NGS described in 1.4.4. This thesis is structured in three parts: the first one is about segmentation models, the second one is about the model on heterogeneity and the third one is about bioinformatic considerations to deal with real data.

In the first part, we start by introducing the statistical models that are usually used to segment this kind of data namely c, b and d, or even c_1 and c_2 signals (Chapter 2). The aim of segmentation methods is to recover the genome location where the altered regions are not the same before and after these points. Microarray and NGS produced a large quantity of information at the scale of kilobase even at the scale of the unit base on the genome. Therefore, these methods require to be statistically efficient to discover relevant biomarkers correctly but also efficiently in terms of computation, both in terms of time and space complexity. After extending the univariate segmentation methods to apply them simultaneously to c and d signals, we present a new strategy to evaluate the gain using both c and d signals to recover alterations in DNA copy number signals in Chapter 3. The last chapter of this section aims to present a new method that does not require to transform b signals to d signals (Chapter 4).

Then, in the second part, we focused on the discovery of the tumoral heterogeneity by dealing with several samples simultaneously. After a chapter that briefly introduces the models used to study tumoral heterogeneity (Chapter 5), we present a new model in Chapter 6. This model has the particularity to be able to be applied to discover intra or inter-tumoral heterogeneity on microarrays or HTS data. We present two applications to two different kinds of data sets (Chapter 7). The first one is a public data set where data is from microarrays. We attempt to infer intra-tumoral heterogeneity from several samples of the same patient. The second one is a collaboration with Institut Curie. We analyze heterogeneity from several patients suffering from a particular breast cancer. We dealt with WES data for this study.

To finish, the last part summarizes some contributions at a bioinformatic level to solve problems come across the dealing with real data analysis. Indeed, through the applications of the developed heterogeneity model on real data, several issues have been raised at normalization level. The two chapters of this part are therefore focused on normalization of data. Chapter 8 deals with the estimation of the DoH signals in absence of a normal reference from microarrays data and is linked to the first application. Chapter 9 is focused on the normalization of the WES data to get TCN, BAF and DoH signals as from microarrays.

Part I

Joint segmentation methods

Table of Contents

Chapter 2

DNA copy number segmentation

This chapter introduces the segmentation models to detect alteration in DNA copy number signals. After a brief review of the main methods to segment DNA copy number signals, we present the univariate models in a second section. In a third section, we described the approach that we have considered to segment jointly the TCN and the DoH. Finally, we present the standard approach to select the best model in the case of segmentation models.

2.1 Typology of copy number segmentation methods

This section is about the segmentation models previously described by [Neuvial et al., 2011] and [Zhang, 2010]. In the last twenty years, many different methods have been proposed for the analysis of DNA copy number profiles. Most of them may be classified into four categories: methods based on Hidden Markov Models (HMM), multiple change-point methods, fused lasso-based methods and recursive segmentation methods.

1. HMM-based approaches rely on the idea that the recovered DNA copy number should be discrete and that these different levels can be modeled using a small number of HMM states. A typical example of such an HMM is the work of [Fridlyand, 2004]. For the specific case of SNP array analysis in cancer samples, several dedicated HMM have been proposed [Sun et al., 2009, Greenman et al., 2010, Chen et al., 2011] (Section 2.2.1).

2. Multiple change-point methods assume that the observed signal is affected by abrupt changes and that between these breaks the signal should be homogenous [Picard et al., 2005] (2.2.3).

3. Recursive segmentation approaches rely on the intuitive idea that a segmentation can be recovered by recursively cutting the signal into two or more pieces. A typical example of such an recursive approach is the work of [Olshen et al., 2004] (2.2.4).

4. Methods based on a fused lasso penalty rely on the idea that, in most cases, two successive measurements should have the same estimate. This is encoded by a L_1 penalty on successive differences. The recovered signal is guaranteed to be piecewise constant. A typical example of such a fused model is the work of [Tibshirani et al., 2005]. This class of methods can be viewed as solving a convex relaxation of the multiple change point problem. (2.2.5)

The above classification is by no means exhaustive (see for example [Hupé et al., 2004, Ben-Yaacov and Eldar, 2008]), but summarizes the most common approaches linked to the work of this thesis. In the next section, we present the main classical models.

2.2 Univariate models

2.2.1 Classical Hidden Markov Models

We present in this section the HMM of [Fridlyand, 2004]. The HMM assume that there exist a certain number K of hidden states. Some examples of considered hidden states are described in the following table 2.1:

k	TCN	States
1	0	Full deletion
2	1	Single deletion
3	2	Normal
4	3	Single copy gain
5	4	Double copy gain

Table 2.1 – Example of considered Hidden states in HMM for segmentation models

We denote these hidden states by $(\mathcal{S}_1, \ldots, \mathcal{S}_K)$. Then, we denote by q_j the actual state at the position j on the genome and by $\pi_k = \mathbb{P}(q_1 = \mathcal{S}_k)$, the probability that the first locus is in state k, this is, in fact, the marginal distribution of the hidden states.

The transition matrix corresponds to the probability to move from one state l to another one m and is defined by:

$$a_{lm} = \mathbb{P}(q_{j+1} = \mathcal{S}_m | q_j = \mathcal{S}_l), \tag{2.1}$$

for all $1 \leq j \leq J - 1$ and $1 \leq l, m \leq K$.

[Fridlyand, 2004] assume that it is the $\log 2$ of the TCN in state K which follows a gaussian distribution with unknown mean and variance:

$$\forall j \in \mathcal{S}_k, log2(c_j) \sim \mathcal{N}(\mu_k, \sigma_k^2), \tag{2.2}$$

Note that μ_k here represents the true hidden copy number that is an integer (Table 2.1).

Therefore, the HMM is characterized by three parameters:

- Initial state probabilities denoted by $\pi = (\pi_1, \dots, \pi_K)$
- The transition matrix denoted by A
- The collection of the distribution in each state denoted by B

We denote the parameter vector by $\lambda = (\pi, A, B)$

The optimal values of λ is found by maximizing the likelihood $L(\lambda|\mathbf{c})$ of the parameters given the vector of values $\mathbf{c}$ in several steps.

1. π is initialized by placing a majority of the weight on the normal state and distributing the remaining probability uniformly among all other states.
2. Similarly, the initialization of A is done by assigning a high probability of remaining in the same state and low non-zero probabilities to transitioning between states.
3. Identifying the optimal state sequence associated with a given vector, for each observation c_j. Choosing the state s_l which is individually most likely.
4. Finally, they re-estimate model parameters λ to maximize $L(\lambda|\mathbf{c})$ using an EM algorithm.

The HMM take the advantage of determining both the position of breakpoints (when the states change) but also the status of each region (gain, loss, normal). In this thesis, we make the choice to split the two steps and to focus on the first one that is to detect with a high accuracy the breakpoints and in a second step performing the calling (attribute a status to segments).

2.2.2 Multiple change-point model

We introduce the basic model for DNA copy number signals, which is widely used in the literature by several methods [Hupé et al., 2004, Olshen et al., 2004, Hautaniemi et al., 2003, Picard et al., 2005]. It assumes that total DNA copy number profiles are piecewise constant in the mean. Statistical model for S change points at $m_S = (t_1, \dots, t_S)$ can be written as follows:

$$\forall j = 1, \dots, J \qquad c_j = \gamma_j + \epsilon_j \tag{2.3}$$

where $\forall s \in \{1, \dots, S+1\}, \forall j \in [t_{s-1}, t_s[\qquad \gamma_j = \mathit{\Gamma}_s$ with the following convention $t_0 = 1$ and $t_{S+1} = J + 1$

We consider the homoscedastic model: $\epsilon_j \sim \mathcal{N}(0, \sigma^2)$, the variance does not depend on the region. However, the variance of copy number signals is, in fact, increasing with their mean. Therefore, an heteroscedastic model where $\sigma = \sigma_s$ is more realistic. A common practice in applications is to transform the raw copy number signals using $\sqrt{.}$ or log(.) transformations, in order to stabilize the variance of the signal. Then, the homoscedastic model may make sense. We refer to [Picard et al., 2005] for a discussion on the estimation of the homoscedastic vs the heteroscedastic model.

Assuming that the model is gaussian and homoscedastic, the parameters of the model are the change-point locations denoted by the segmentation $m_S = t_1, \dots, , t_S$, the true copy number $\mathit{\Gamma}_1, \dots, \mathit{\Gamma}_{S+1}$, and the variance of the model σ^2. The likelihood of the model is :

$$\begin{aligned}
\mathcal{L}_S(\mathbf{c}, m_S, \mathit{\Gamma}, \sigma^2) &= \prod_{s=1}^{S+1} \prod_{j=t_{s-1}}^{t_s - 1} f(c_j; \gamma_j, \sigma^2) && (2.4)\\
&= \prod_{s=1}^{S+1} \prod_{j=t_{s-1}}^{t_s - 1} \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(c_j - \gamma_j)^2\right\} && (2.5)
\end{aligned}$$

By log transformation, we obtain the following equation:

$$\ell_S(m_S) = -J \log\left(\sigma\sqrt{2\pi}\right) - \frac{1}{2\sigma^2} \sum_{s=1}^{S+1} \sum_{j=t_{s-1}}^{t_s - 1} (c_j - \gamma_j)^2 \tag{2.6}$$

Then, the best segmentation into $S + 1$ segments is defined by:

$$\widehat{m}_S = \underset{m \in \mathcal{M}(S)}{\arg\max}\, \ell_S(m_S) \tag{2.7}$$

where $\mathcal{M}(S)$ is the set of all possible segmentations in S change-points in J-length signals.

And, considering that σ^2 is well estimated by the estimator proposed by [Hall et al., 1990] in a previous step, maximizing the log-likelihood is finally equivalent to:

$$\widehat{m}_S = \underset{m_S \in \mathcal{M}(S)}{\arg\min} \sum_{s=1}^{S+1} \sum_{j=t_{s-1}}^{t_s - 1} (c_j - \gamma_j)^2 \tag{2.8}$$

Then, once S and m_S are known, we deduce the estimations of $\mathit{\Gamma}_s$ for each $s = 1, \ldots, S$:

$$\widehat{\mathit{\Gamma}}_{t_{s-1}, t_s} = \widehat{\mathit{\Gamma}}_s \quad = \quad \frac{1}{\widehat{t}_s - \widehat{t}_{s-1}} \sum_{j=t_{s-1}}^{t_s - 1} c_j \tag{2.9}$$

Here, we can note that the classical model assumes that the signals are Gaussian. However, Gaussian assumption for TCN and DoH can be wrong and could be lead to poor performance of this model. Chapter 3 shows a way to evaluate performance based on a realistic simulation framework. This simulation framework allows checking if the Gaussian assumption is nevertheless acceptable.

The theoretical time complexity of an exhaustive search to partition J points into S segments is in $\mathcal{O}(J^S)$. In the genomic context, space and time complexity is an actual challenge due to the large size J of vector $\mathbf{c}$. Several strategies are used to solve the problem: Dynamic Programming (DP) algorithm, binary strategies, and convex relaxation that enable to reduce the complexity.

2.2.3 Dynamic programming algorithm

In this section, we describe the principle of univariate DP algorithm [Bellman, 1961, Auger and Lawrence, 1989]. In our context, DP is used to solve the model described previously in order to segment DNA copy number The main advantage of the DP algorithm is the exact resolution of the problem based on the model described by equation 2.3 with a reduction of the complexity.

If we denote the cost of a segment between positions u and v by $\mathcal{C}(u, v)$ such that:

$$\forall\, 1 \leq u < v \leq J, \quad \mathcal{C}(u, v) = \sum_{j=u}^{v-1} (c_j - \widehat{\mathit{\Gamma}}_{uv})^2 \tag{2.10}$$

where $\widehat{\Gamma}_{uv} = \dfrac{1}{v-u} \displaystyle\sum_{j=u}^{v-1} c_j$ is the mean of TCN in the segment between u and v (Eq. 2.9). We define the cost of the best segmentation of the TCN signal between positions 1 and J into S segments by:

$$\mathbf{C}_S(1, J) = \min_{m_S \in \mathcal{M}(S)} \sum_{s=1}^{S+1} \mathcal{C}(t_{s-1}, t_s). \tag{2.11}$$

Estimates are the same as for the maximizer of likelihood.

The cost of the best segmentation into 2 parts of segment (u, v) is defined by:

$$\mathbf{C}_2(u, v) \quad = \quad \min_j \{\mathcal{C}(u, j) + \mathcal{C}(j, v)\} \tag{2.12}$$

And, the cost of the best segmentation into 3 parts of segment (u, v) is defined by:

$$\mathbf{C}_3(u, v) \quad = \quad \min_j \{\mathbf{C}_2(u, j) + \mathcal{C}(j, v)\} \tag{2.13}$$

Recursively, the cost of the best segmentation into s parts of segment (u, v) is defined by::

$$\forall\, 1 \leq s \leq S \quad \mathbf{C}_s(u, v) \quad = \quad \min_{u \leq j \leq v} \{\mathbf{C}_{s-1}(u, j) + \mathcal{C}(j, v)\} \tag{2.14}$$

The argmin of 2.14 provides the last best position that enables to move from $s - 1$ to the s segments.

Complexity is reduced to $\mathcal{O}(SJ^2)$ instead of $\mathcal{O}(J^S)$ in time and we deduce the following algorithm by applying the update rules:

```
Compute $\mathcal{C}_1(u, v)$ for all $u$ and $v$ in $\{1, \ldots, J\}$
for $s = 2$ to $S - 1$ do
    for $j = s$ to $J$ do
        $\mathbf{C}_s(1, j) = \min_{j' \leq j}\{\ \mathbf{C}_{s-1}(1, j') + \mathcal{C}_1(j', j+1)\ \}$
        $j^* = \arg\min_{j' \leq j}\{\ \mathbf{C}_{s-1}(1, j') + \mathcal{C}_1(j', j+1)\ \}$ the last best position to
        segment $(1, j)$ in $s$ segments.
```

Algorithm 1: Basic use of Dynamic Programming

In fact, the last j^* is the best position to move from $S - 1$ segments to S segments for the TCN signal 1 and J, therefore, we perform a back tracking to recover all the best positions j^* at each step.

Nevertheless, even if this trick allows to exactly solve the problem, it has a cost in space because without dynamic programming the cost is of the order of $\mathcal{O}(1)$ while for dynamic programming the cost is of $\mathcal{O}(SJ^2)$. However, [Rigaill, 2015, Cleynen et al., 2014] has developed a DP algorithm that enables to reduce the average time-complexity in $\mathcal{O}(J\log(J))$ and suffers from a worst-case complexity of $\mathcal{O}(J^2)$. In the same line, another version of regularized dynamic programming has been explored by [Killick et al., 2012] who designed the PELT procedure. It provides the best segmentation over all segmentations with a penalty of λ per change-point with an $\mathcal{O}(J)$ complexity in time if the number of change-points is linear in n. On the one hand, with PELT it is not straightforward to efficiently solve Eq. (2.11) for each $1 \leq S \leq S$, which is precisely the goal we pursue. On the other hand the complexity of the pruning inside PELT depends on the true number of change-points and number of dimensions. In particular for a small number of change-points it is quadratic.

2.2.4 Basic binary segmentation

Binary methods are also used to reduce complexity and solve this kind problem by approximation [Olshen et al., 2004, Yang, 2012, Fryzlewicz, 2012]. The principle is to cut the signal into two segments at the best location and do this recursively at each step to finally reach S segments in the signal. Let us recall that the likelihood ratio statistic for testing the null hypothesis of no change against the alternative that there is exactly one change at an unknown location [Sen and Srivastava, 1975] is given by $\max_{1\leq j\leq J}|Z_j|$ in the case of homoscedastic model , where

$$Z_j = \frac{\frac{\widetilde{c}_j}{j} - \frac{\widetilde{c}_j - \widetilde{c}_J}{J-j}}{\sqrt{\frac{1}{j} + \frac{1}{J-j}}}, \tag{2.15}$$

where $\widetilde{c}_j$ denotes the partial sum $\widetilde{c}_j = \sum_{1\leq l\leq i} c_l$ for $1 \leq j \leq J$. By using the previous notations 2.9, this statistic is equal to:

$$Z_j = \frac{\widehat{\Gamma}_{1j+1} - \widehat{\Gamma}_{j+1,J+1}}{\sqrt{\frac{1}{j} + \frac{1}{J-j}}}, \tag{2.16}$$

Connection to the likelihood We explain in detail the origin of Z_j in the case of we deal with one-dimensional signals. Let us test the null hypothesis $\mathcal{H}_0$: there is no change-point in the signal versus $\mathcal{H}_1$: there is exactly one change-point at the location t in the signal.

Consider the likelihood of the model $\mathcal{M}_1$:

$$\mathcal{L}_1(t, c_1, \dots, c_J, \varGamma_1, \varGamma_2) = \frac{1}{\sqrt{2\pi\sigma^2}^J} \exp\left\{\sum_{j=1}^{t} \frac{(c_j - \varGamma_1)^2}{2\sigma^2} + \sum_{j=t+1}^{J} \frac{(c_j - \varGamma_2)^2}{2\sigma^2}\right\} \tag{2.17}$$

And the likelihood of the model $\mathcal{M}_0$

$$\mathcal{L}_0(c_1, \dots, c_J, \varGamma) = \frac{1}{\sqrt{2\pi\sigma^2}^J} \exp\left\{\sum_{j=1}^{J} \frac{(c_j - \varGamma_0)^2}{2\sigma^2}\right\} \tag{2.18}$$

Remember that, the maximum-likelihood estimators of equation (2.17) are:

$$\widehat{\varGamma}_1 = \widehat{\varGamma}_{1,t+1} = \frac{1}{t}\sum_{j=1}^{t} c_j$$

and

$$\widehat{\varGamma}_2 = \widehat{\varGamma}_{t+1,J+1}\frac{1}{J-t}\sum_{j=t+1}^{J} c_j$$

and the maximum-likelihood estimator of (2.18) is:

$$\widehat{\varGamma}_0 = \widehat{\varGamma}_{1,J+1} = \frac{1}{J}\sum_{j=1}^{J} c_j$$

The likelihood ratio statistic is defined by :

$$RV = \frac{\mathcal{L}_1}{\mathcal{L}_0}$$

And by log transformation:

$$\log(RV) = \ell_1 - \ell_0$$

where ℓ_1 and ℓ_0 are the log-likelihood.

Then,

$$\begin{aligned}\log(RV) &= \sum_{j=1}^{t}\frac{(c_j-\widehat{\mathit{\Gamma}}_1)^2}{2\sigma^2}+\sum_{j=t+1}^{J}\frac{(c_j-\widehat{\mathit{\Gamma}}_2)^2}{2\sigma^2}\\ &\quad -\sum_{j=1}^{J}\frac{(c_j-\widehat{\mathit{\Gamma}}_0)^2}{2\sigma^2} \qquad (2.19)\end{aligned}$$

Therefore

$$\begin{aligned}2\sigma^2\log(RV) &= \sum_{j=1}^{t}(c_j-\widehat{\mathit{\Gamma}}_1)^2+\sum_{j=t+1}^{J}(c_j-\widehat{\mathit{\Gamma}}_2)^2\\ &\quad -\sum_{j=1}^{J}(c_j-\widehat{\mathit{\Gamma}}_0)^2 \qquad (2.20)\end{aligned}$$

After some simplifications by using the König-Huygens formula and using definitions of $\widehat{\mathit{\Gamma}}_1$, $\widehat{\mathit{\Gamma}}_2$ and $\widehat{\mathit{\Gamma}}_0$, we obtain :

$$2\sigma^2\log(RV)=\left(-t\widehat{\mathit{\Gamma}}_1^2-(J-t)\widehat{\mathit{\Gamma}}_2^2+J\widehat{\mathit{\Gamma}}_0^2\right) \qquad (2.21)$$

Lemma 1. $2\sigma^2\log(RV)=\left(-\frac{t(J-t)}{J}\left(\widehat{\mathit{\Gamma}}_1-\widehat{\mathit{\Gamma}}_2\right)^2\right)$

The proof is given in Appendix A.

Therefore, we have,

$$\begin{aligned}Z_t &= \sqrt{-2\sigma^2\log(RV)}=\sqrt{\frac{\left(\widehat{\mathit{\Gamma}}_1-\widehat{\mathit{\Gamma}}_2\right)^2}{\frac{J}{t(J-t)}}}\\ &= \left|\frac{\left(\widehat{\mathit{\Gamma}}_1-\widehat{\mathit{\Gamma}}_2\right)}{\sqrt{\frac{1}{t}+\frac{1}{J-t}}}\right| \qquad (2.22)\end{aligned}$$

We define the first change-point by

$$t_1=\underset{1\le t\le J}{\arg\max}\, Z_t^2.$$

Lemma 2. *The first change-point is in fact the maximizer of likelihood ratio.*

Then we compute Z_t for each $1 \leq t < t_1$ and $t_1 < t \leq J$, and Z_{t_1} is set to 0. At this step, we get two possible change-points: t_2 on the first segment and t_2' on the second segment.

The second step is to compute for each segment the gain in the meaning of the RSE (root square error) achieved by the two change-points at location t :

$$\text{gain}(t) = RSE(\text{no change-point}, t) - RSE(\text{one change-point,t})$$

Where

$$RSE(\text{no change-point}, t_2) = \sqrt{\tfrac{1}{t_1} \textstyle\sum_{j=1}^{t_1} \left(c_j - \widehat{\Gamma}_{1,t_1+1}\right)^2} = \sqrt{\tfrac{1}{t_1} \mathcal{C}(1, t_1+1)}$$

with $\widehat{\Gamma}_{1,t_1+1} = \dfrac{1}{t_1} \displaystyle\sum_{j=1}^{t_1} c_j$

$$RSE(\text{one change-point}, t_2) =$$
$$\sqrt{\frac{1}{t_1}\left(\sum_{j=1}^{t_2}\left(c_j - \widehat{\Gamma}_{1,t_2+1}\right)^2 + \sum_{j=t_2+1}^{t_1}\left(c_j - \widehat{\Gamma}_{t_2+1,t_1+1}\right)^2\right)}$$

Bellow is the detailed explanation of the steps of recursive binary segmentation in Algorithm 2.

```
for $k \leftarrow 1, \dots S$ do
    Compute $Z_j$ for each $\min(1,t) \leq j < t_{k-1}$ and $t_{k-1} < j \leq \max(J,t)$,;
    where $t$ is the set of selected change-points at each previous step;
    and $\forall j \in \{1, \dots, J\}$ $Z_j$ is equal to 0.;
    At this step, there are $k$ possible change-points : two change-points from this
    step $t_k$ and $t_k'$, and $k-2$ change-points which were not selected at previous
    steps;
    Compute for the two new segments the gain brings by the change-points
    gain($t_k$) and gain($t_k'$);
    Compare with the $k-2$ previous gains computed.;
    The next change-point $k$ is the one which is in the segment where the gain is
    the greatest. In other terms $t = \underset{t \in \{t_1, \dots, t_k, t_k'\}}{\arg\max}$ gain($t$);
```

Algorithm 2: Basic recursive binary segmentation

Efficiently implemented, recursive binary segmentation reaches a time complexity of order $\mathcal{O}(Jlog(S))$. In space, this algorithm requires to save only the partial sums i.e. algorithm is in $\mathcal{O}(J)$.

2.2.5 Convex relaxation

To reduce the complexity to recover change-points in TCN profiles [Harchaoui and Lévy-Leduc, 2008] demonstrates that the problem can be written as a convex optimization problem.

As γ is a J piecewise constant vector, maximizing the likelihood define in equation 2.4 after estimation of σ^2 is exactly equivalent to solve:

$$\min_{\gamma \in \mathbb{R}^J} \|\mathbf{c} - \gamma\|_2^2 \text{ subject to } \sum_{j=1}^{J-1} \mathbb{1}_{\gamma_{j+1} \neq \gamma_j} \leq S, \tag{2.23}$$

where $\mathbb{1}$ is the indicator function and $\|.\|_2$ indicates the euclidean norm for vectors. Equation 2.23 minimizes the quadratic error to look for the best approximation of signal $\mathbf{c}$ subject to the constraint that there are exactly S breakpoints in the signal.

The presence of singularities at the boundary induces the variable selection by exactly zeroing some coefficients. But this problem is non-convex and hard to optimize. The following equation is the convex-relaxation formulation of Eq. 2.23:

$$\min_{\gamma \in \mathbb{R}^J} \frac{1}{2}\|\mathbf{c} - \gamma\|_2^2 \text{ subject to } \sum_{j=1}^{J-1} |\gamma_{j+1} - \gamma_j| \leq \mu, \tag{2.24}$$

$$\min_{\gamma \in \mathbb{R}^J} \frac{1}{2}\|\mathbf{c} - \gamma\|_2^2 + \lambda \sum_{j=1}^{J-1} |\gamma_{j+1} - \gamma_j|, \tag{2.25}$$

For a fixed λ, the $\gamma \in \mathbb{R}^J$ is still piecewise constant. Note that parameter λ controls S, the number of change-points.

Formulation as a LASSO problem [Harchaoui and Lévy-Leduc, 2008] also demonstrated that it is possible to reformulate this problem as a simple lasso regression. Let : $\beta_j = \gamma_{j+1} - \gamma_j \qquad \forall j = 1, ..., (J-1)$.

Then : $\gamma_j = \gamma_1 + \sum_{i=1}^{j-1} \beta_i$, with the following convention $\gamma_0 = 1$.

and γ can be written as follows $\mathbb{1}_{J,1}\gamma_1 + \mathbf{X}\beta$ where $\mathbf{X}$ is a $J \times (J-1)$ matrix and $\mathbf{X}_{j'j} = 1$ for $j' > j$ and 0 otherwise.

$$\min_{\gamma_1 \in \mathbb{R}, \beta \in \mathbb{R}^{J-1}} \frac{1}{2}\|\mathbf{c} - \mathbf{X}\beta - \mathbb{1}_{J,1}\gamma_1\|_2^2 + \lambda \sum_{j=1}^{J-1} |\beta_j|, \tag{2.26}$$

The solution of Eq. 2.26 in γ_1 is given by:

$$-\mathbb{1}_{1,J}(\mathbf{c} - \mathbf{X}\beta - \mathbb{1}_{J,1}\gamma_1) = 0 \tag{2.27}$$

$$\Leftrightarrow \quad \gamma_1 = \mathbb{1}_{1,J}(\mathbf{c} - \mathbf{X}\beta)/J \tag{2.28}$$

Then, by optimizing the equation in β :

$$\min_{\beta \in \mathbb{R}^{J-1}} \frac{1}{2}\|\mathbf{c} - \mathbf{X}\beta - \mathbb{1}_{J,1}\gamma_1\|_2^2 + \lambda \sum_{j=1}^{J-1} |\beta_j|, \tag{2.29}$$

The jump vector β can be obtained as a solution of 2.30 by injecting 2.28 in 2.29:

$$\min_{\beta \in \mathbb{R}^{J-1}} \frac{1}{2}\|\bar{\mathbf{c}} - \bar{\mathbf{X}}\beta\|_2^2 + \lambda \sum_{j=1}^{J-1} |\beta_j|, \tag{2.30}$$

where $\bar{\mathbf{c}}$ and $\bar{\mathbf{X}}$ are obtained from $\mathbf{c}$ and $\mathbf{X}$ by centering each column.

Finally, we get a standard lasso regression model.

$$\widehat{\beta} = \arg\min_{\beta \in \mathbb{R}^{J-1}} \frac{1}{2}\|\bar{\mathbf{c}} - \bar{\mathbf{X}}\beta\|_2^2 + \lambda\|\beta\|_1, \tag{2.31}$$

where $\|.\|_1$ is the ℓ_1 norm.

Illustration To illustrate the fact that problem 2.24 induces also sparsity as the problem 2.23, we take $J = 3$, then $\beta_1 = \gamma_2 - \gamma_1$ and $\beta_2 = \gamma_3 - \gamma_2$. The problem becomes a standard linear regression model where it is the coefficient β that is penalized (Eq. 2.31).

The Figure 2.1 shows the geometrical forms of the ℓ_0 (left) and ℓ_1 (right) penalizations. It is clear that the ℓ_1-norm induces sparsity in addition to be convex.

If the matrix $\mathbf{X}$ is orthonormal the solution of the lasso problem is easy to compute [Tibshirani, 1996]. However, in our case, $\mathbf{X}$ is not orthonormal and there is no explicit solution. A practical solution is to use the Least Angle Regression LAR algorithm [Efron et al., 2004, Harchaoui and Lévy-Leduc, 2008]. Indeed, the principle of this algorithm is to select step by step the most meaningful variable (active-set method). In our case, this is equivalent to select the most relevant breakpoints and it is easy to stop the algorithm at a predefined number of segments.

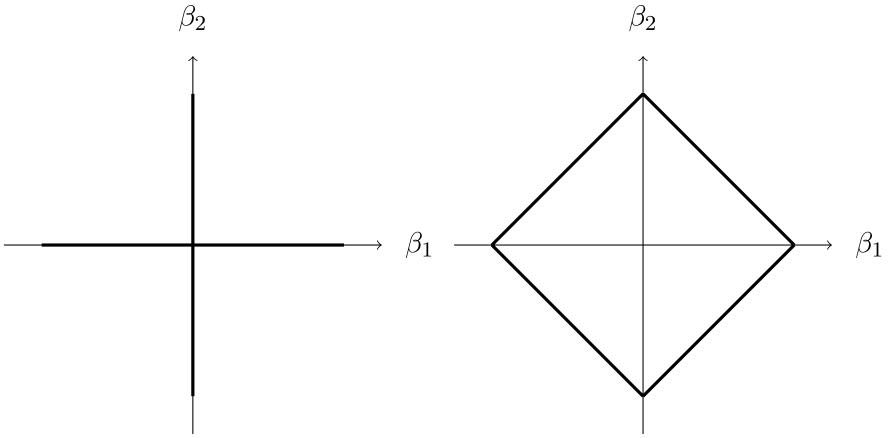


Figure 2.1 – Contour of the feasible solutions by $\|\beta\|_0 \leq 1$ (left) and $\|\beta\|_1 \leq 1$ (right) when $\beta \in \mathbb{R}^2$

2.2.6 The two-step strategies

[Gey and Lebarbier, 2008, Harchaoui and Lévy-Leduc, 2008] use a hybrid strategy to catch change-points in TCN signals. This two-step approach consists in:

- quickly identifying a set of candidate change points by taking account the two dimensions
- pruning this list using DP.

To quickly identify the candidate breakpoints, [Harchaoui and Lévy-Leduc, 2008] use fused LASSO while [Gey and Lebarbier, 2008] uses binary segmentation. Both of the two methods prune the list of candidates by using DP.

Therefore, the number of explored breakpoints is very small compared to the total number of loci ($S \ll J$). The only difference with the classical dynamic programming is that segments are defined by the breakpoints recovered by the lasso or by recursive binary segmentation. This step of DP enables us to recover the best segmentations in $s \in \{1, \ldots, S\}$ breakpoints, in terms of likelihood. Then, the time complexity of the dynamic programming is of the order of $O(S^3)$ for S change-points. The overall complexity is therefore $O(S^3 + J\log(J)S)$ and $O(S^3 + JS)$ for RBS and Fused Lasso respectively.

2.3 Two dimensional methods

In this section, after a few words on complexity, we describe two methods to perform jointly the segmentation of TCN and BAF signals. Indeed, we saw in the introduction that change-points occur exactly at the same location and take into account these two dimensions could increase the detection power.

2.3.1 Complexity

All presented methods are extensible to detect breakpoints in multiple dimension but at a non-negligible cost. It is, for instance, possible to extend the DP algorithms. However, the trick of [Rigaill, 2015] that permits to reduce the complexity is not applicable to multidimensional signals. Therefore, when the dimension of the signal is larger than one, the computation takes $\mathcal{O}(SJ^2)$ in time and $\mathcal{O}(SdJ^2)$ in space, where d is the dimension of the signal. [Mosen-Ansorena and Aransay, 2013] develops a method to use both TCN and DoH signals by using the trick of [Huber et al., 2006] that is to constraint the length of segments. Therefore, the complexity is reduced to $\mathcal{O}(JL)$ where L is the maximal length of individual segments. However, we notice that L requires to be greater that $\frac{J}{S}$ in algorithm to provide a segmentation. As consequence, the time complexity remains quadratic in $\mathcal{O}(J^2/S)$. [Chen et al., 2011] have also developed a HMM to segment parental copy number signals.

Recursive binary segmentation and convex relaxation methods are easily extensible to multidimensional signals with a lesser cost in terms of time complexity. Both algorithms have a low space complexity in $\mathcal{O}(J)$. It is, then, possible to extend the two-steps strategies to reach a space complexity in $\mathcal{O}(J)$. Time computations takes for RBS and convex relaxation $\mathcal{O}(J\log(S)d)$ and $\mathcal{O}(JS \times d)$, respectively.

Based on the work of [Gey and Lebarbier, 2008, Harchaoui and Lévy-Leduc, 2008, Vert and Bleakley, 2010], we applied the hybrid strategy to catch change-points simultaneously in TCN and BAF (section 2.2.6).

We first present the group Fused Lasso uses by [Vert and Bleakley, 2010] to recover quickly breakpoints in TCN and BAF based on convex relaxation method (section 2.3.2. Then, we extend the binary segmentation described in [Gey and Lebarbier, 2008] in order to include the BAF information (section 2.3.3).

2.3.2 Group Fused Lasso Segmentation

This section presents the method developed by [Vert and Bleakley, 2010] but applying on TCN and DoH signals. Indeed, in the original article, the aim was to detect change-points from CGH arrays in a multisample data set from various patients. In our case of the joint segmentation of the B allele fraction and the total DNA copy number signals, change-points occur by definition exactly at the same location in the two signals. In section 2.2.5, we show how Group Fused Lasso methods can be used to perform the joint segmentation of **c** and **d**.

Group Fused Lasso method

Let $\mathbf{Y} = (\mathbf{c} \quad \mathbf{d})$ the matrix that contains the TCN and the DoH signals as described in section 1.4. Therefore the matrix size is $J \times 2$, each row j of the matrix is a couple of TCN and DoH at locus j. Note that we assume that the SNPs are ordered along the genome.

$$\begin{cases} \forall j = 1, \dots, J & c_j = \gamma_j + \epsilon_j^{(1)} \\ \forall j = 1, \dots, J & d_j = \delta_j + \epsilon_j^{(2)}, \end{cases}$$

where $\epsilon^{(d)} \sim \mathcal{N}(0, \sigma_d^2)$, for $d = 1, 2$.

We search $\varPhi = (\gamma \quad \delta)$, the matrix containing the true values and such that **c** and **d** satisfy the following equation:

$$\min_{\varPhi \in \mathbb{R}^{J \times 2}} \frac{1}{2} \|\mathbf{Y} - \varPhi\|_2^2 \text{ s.t. } \sum_{j=1}^{J-1} \mathbb{1}_{\gamma_{j+1} \neq \gamma_j} \leq S \text{ and } \sum_{j=1}^{J-1} \mathbb{1}_{\delta_{j+1} \neq \delta_j} \leq S \tag{2.32}$$

The convex relaxation trick of the above equation is:

$$\min_{\varPhi \in \mathbb{R}^{J \times 2}} \frac{1}{2} \|\mathbf{Y} - \varPhi\|_2^2 + \lambda \sum_{j=1}^{J-1} \left((\gamma_{j+1} - \gamma_j)^2 + (\delta_{j+1} - \delta_j)^2 \right)^{\frac{1}{2}} \tag{2.33}$$

In other terms, the aim is to recover two piece-wise constant functions under the constraint that the breakpoints occur exactly at the same location. In addition we constrain functions to have a small number of jumps.

[Vert and Bleakley, 2010] added weights depending on the position along the genome allows to reduce bound effects. For a $h_j = \sqrt{\frac{J}{j(J-j)}}$, the first change-point recovered by the algorithm will be the same as in binary segmentation. It provides a guarantee on the consistence of the first change-point induced by the Gaussian likelihood.

$$\min_{\Phi \in \mathbb{R}^{J \times 2}} \frac{1}{2} \|\mathbf{Y} - \Phi\|_2^2 + \lambda \sum_{j=1}^{J-1} \frac{\left((\gamma_{j+1} - \gamma_j)^2 + (\delta_{j+1} - \delta_j)^2\right)^{\frac{1}{2}}}{h_j} \tag{2.34}$$

As we demonstrate on the univariate case in section 2.2.5, it is possible to reformulate the problem 2.34 as a Group Lasso regression model. Group lasso is able to force the coefficients of the same group to be null or not simultaneously. In this case, this allows to detect change-point with a high probability but as well to detect change-point which are visible on only one of the two dimensions.

Algorithm

It exists a corresponding between the λ parameter in the Lasso regression model and the number of jumps in the signal. Indeed, the smaller λ, the larger the number of jumps. As the parametrization of λ is difficult even for classical regression problem and not intuitive in our case, we use the trick described in [Vert and Bleakley, 2010] that is using group LARS algorithm to solve group LASSO. This method consists in defining a number of non-zero components which corresponds to the number of breakpoints in the model (active set method). In this way, it is possible to segment the signal for a fixed number of segment and not for a fixed value of λ.

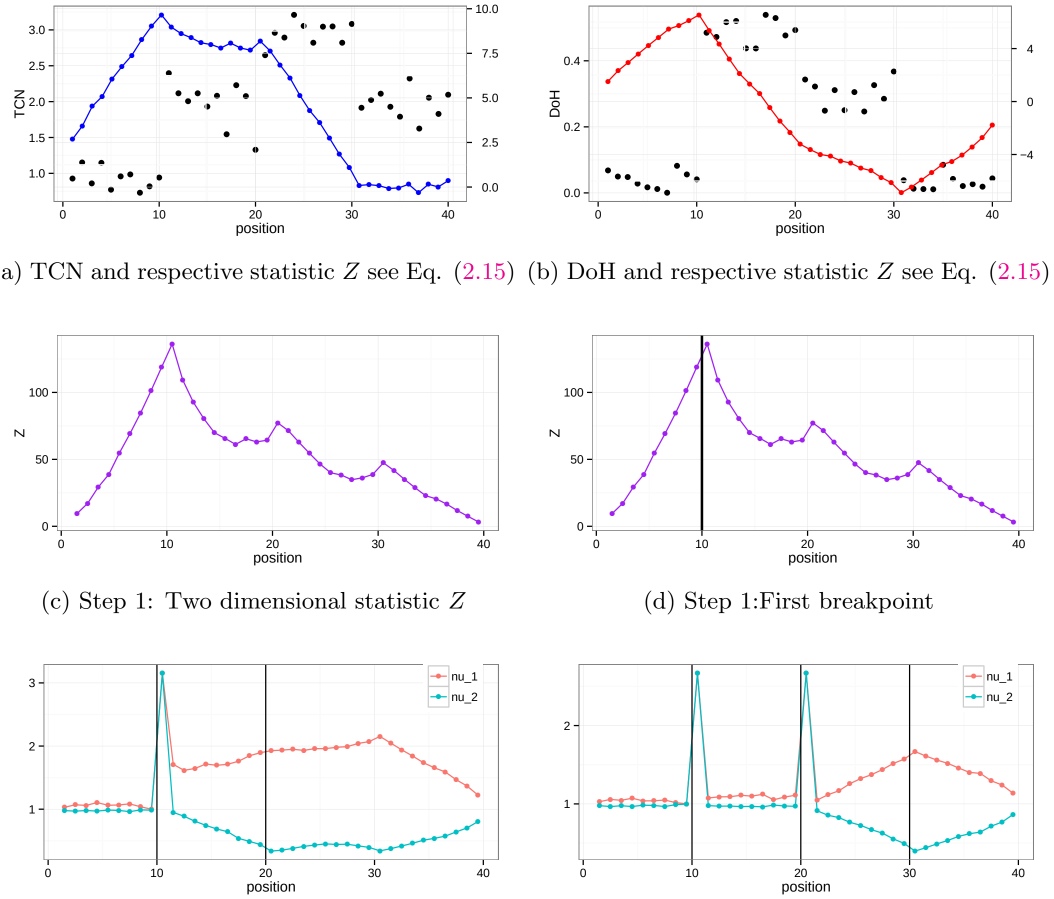
The first step is to center the matrices $\mathbf{Y}$ and $\mathbf{X}$ defined above in the previous section. In our case, remember that $\mathbf{Y}$ is a $J \times 2$ matrix and $\mathbf{X}$ is a $J \times (J-1)$ matrix.

The predefined number of breakpoints that we look for in the signal is denoted S.

The resolution of the problem is presented by Algorithm 7 in Appendix B.

Illustration of the algorithm

A toy illustration is shown in Figure 2.2. The graphic on the bottom left represents a 40 points TCN signal with three change-points at locations 10, 20 and 30. The blue curve represents the likelihood ratio statistic for this univariate signal. Then, the graphic on the bottom right represents the DoH signal for heterozygous SNP with the same three change-points. The red curve represents the likelihood ratio statistic for this univariate signal. We simulated here the same number of points in the two dimensions (TCN and DoH) (see Fig. 2.2a and 2.2b), the start point (first breakpoint) is, in fact, the breakpoint recovered by the univariate likelihood ratio test (Fig. 2.2c and 2.2d). Then, we represent the two solutions of Eq. B.2 that is quadratic, the breakpoint added in the active set is the one for which the solution is the lower (Fig. 2.2e and 2.2f).



(a) TCN and respective statistic Z see Eq. (2.15)

(b) DoH and respective statistic Z see Eq. (2.15)

(c) Step 1: Two dimensional statistic Z

(d) Step 1:First breakpoint

(e) Step 2: The two solutions of the quadratic Equation B.2

(f) Step 3: The two solutions of the quadratic Equation B.2

Figure 2.2 – Illustration of GFLARS algorithm. Top panel: TCN and DoH signals with their respective Z statistic (2.15). Middle panel is multi-dimensional statistic Z. Bottom panel contains.

Limitations

Due to Eq. B.2, the first limitation that we noted is the impossibility to deal with SNPs arrays which contain probes that only measure TCN. In addition, DoH is defined only for the heterozygous SNPs and the method is applicable only for this signal. By consequence, we lose a large quantity of information because on average there is only one-third of heterozygous SNP in arrays. We noted that the first change-point is, in fact, the first point of binary segmentation by maximum likelihood. For this reason, we extended the original recursive binary segmentation to the two dimensions and used tricks to deal with rows where there is no DoH information.

2.3.3 Two dimensional recursive binary segmentation

We have seen in section 2.2.4 the binary segmentation when the signal is univariate. In this section, we show that it is easy to extend the binary segmentation to multidimensional signals. Let assume that d is the dimension of the signal. For $d > 1$, as we assumed that the errors of the d dimensions are independent of each other, the likelihood ratio statistic becomes $\max_{1 \leq t \leq J} ||Z_t||_2^2$, however, now a Z_t is a d-dimensional vector.

Although the objective of this method is to take into account the two dimensions, the TCN dimension has many more informative probes than the dimension of allelic ratio and it is not so obvious. Indeed, in the DoH dimension, only heterozygous SNP are informative. To compute the joint likelihood ratio statistic we need that J be the same in the two dimensions. An easy approach is to say that when i is only in one of the two dimensions (for example in the first), Z_t^1 is computed and Z_t^2 is equal to $Z_{t'}^2$ (where $t' < t$) the previous one where there exists a value for the second dimension. On other terms Z_t^2 provides no information compared to $Z_{t'}^2$.

If the data are multivariate, the likelihood for the model $\mathcal{M}_0$ can be written as :

$$
\begin{aligned}
\mathcal{L}_0(\mathbf{y}_1, \ldots, \mathbf{y}_J, \gamma) &= \Pi_{j=1}^{J} \frac{1}{\sqrt{(2\pi)^2 \varSigma}} \exp\left\{\frac{1}{2}(\mathbf{y}_j - \varPhi_j)^T \varSigma^{-1} (\mathbf{y}_j - \varPhi_j)\right\} \\
&= \frac{1}{\sqrt{(2\pi)^2 \varSigma}^J} \exp\left\{\frac{1}{2}\sum_{j=1}^{J}(\mathbf{y}_j - \varPhi_j) \varSigma^{-1} (\mathbf{y}_j - \varPhi_j)\right\} \qquad (2.35)
\end{aligned}
$$

With $\forall i \in \{1, \ldots, J\} \quad \mathbf{y}_j = (c_j \quad d_j) \in \mathbb{R}^2$, $\varPhi_j = (\gamma_j \quad \delta_j) \in \mathbb{R}^2$, and $\varSigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} = \sigma^2 I_2$

Considering that the matrix $\mathbf{Y}$ is standardized.

The log-likelihood is defined by :

$$\ell_0(\mathbf{Y}, \Phi) = -\frac{J}{2}\log\left((2\pi)^2\sigma^2\right) + \frac{1}{2\sigma^2}\left\{\sum_{j=1}^{J}\|\mathbf{y}_j - \Phi_j\|_2^2\right\} \tag{2.36}$$

We can do the same calculations as in the section 2.2.4 to recover the test statistic:

$$\begin{aligned}\|Z_t\|_2^2 = -2\sigma^2\log(RV) = &\left(\frac{k(J-k)}{J}\left(\widehat{\gamma}^{(1)} - \widehat{\gamma}^{(2)}\right)^2\right)\\ &+\left(\frac{k(J-k)}{J}\left(\widehat{\delta}^{(1)} - \widehat{\delta}^{(2)}\right)^2\right)\end{aligned} \tag{2.37}$$

where

$$\widehat{\gamma}^{(1)} = \frac{1}{t}\sum_{j=1}^{t} c_j \qquad \widehat{\delta}^{(1)} = \frac{1}{t}\sum_{j=1}^{t} d_j$$

$$\widehat{\gamma}^{(2)} = \frac{1}{J-t}\sum_{j=t+1}^{J} c_j \qquad \widehat{\delta}^{(2)} = \frac{1}{J-t}\sum_{j=t+1}^{J} d_j$$

Then we deduce:

$$\|Z_t\|_2^2 = \frac{\left(\widehat{\gamma}^{(1)} - \widehat{\gamma}^{(2)}\right)^2 + \left(\widehat{\delta}^{(1)} - \widehat{\delta}^{(2)}\right)^2}{\frac{1}{t} + \frac{1}{J-t}}$$

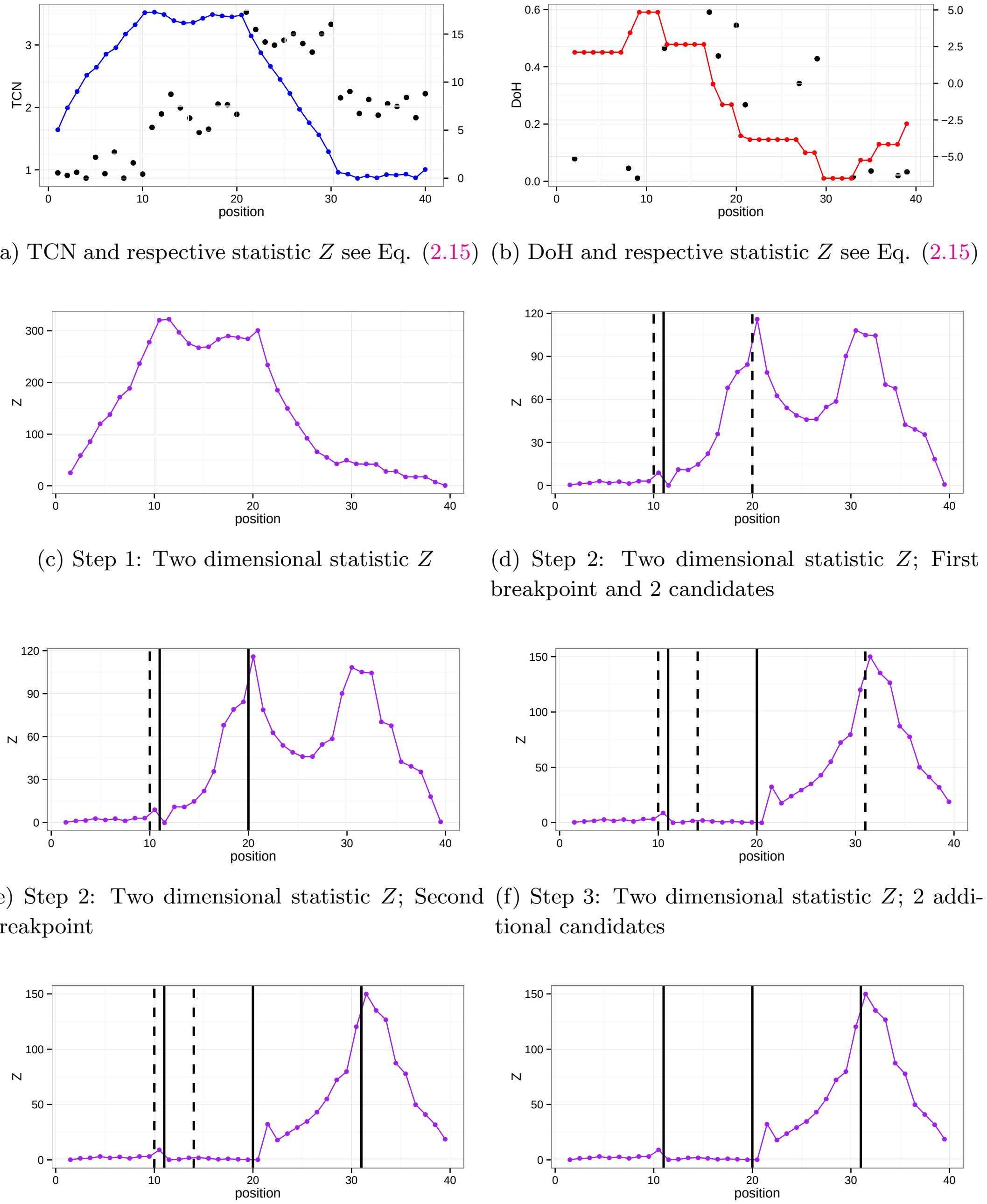
Illustration of algorithm

Another toy illustration is shown on figure 2.3 to illustrate the RBS algorithm. The graphic on the bottom left represents a 40 points TCN signal with three true change-points at location 10, 20 and 30 (Fig. 2.3a). The blue curve represents the likelihood ratio statistic for this univariate signal. Then, the graphic on the bottom right represents the DoH signal for heterozygous SNP with the same three change-points (Fig. 2.3b). The red curve represents the likelihood ratio statistic for this univariate signal. The graphic under the copy number signal is $\|Z_t\|^2$, the aggregation of the blue and red curves (Fig. 2.3c). So, at the first step, there is only one solution for the first change-point, and the maximum of $\|Z_t\|^2$ is between the 10^{th} and the 11^{th}, so we define the next change-points at location 10. Then, $\|Z_{10}\|^2$ is equal to 0 at step 2, compute $\|Z_t\|^2$ separately between 1 and 9 and between 11 and 40 and there are two possible change-points, the one which brings the greatest gain in terms of RSE is selected, gain on the left of the first change-point (at location 10) is equal to 0.02 and gain on the right is equal to 2.25, so the next change-point is at location 20 (Fig. 2.3d and 2.3e). At step 3, $\|Z_{20}\|^2 = 0$. To finish, compute $\|Z_t\|^2$ between locations 11 and 20 and between location 21 and 40 (Fig. 2.3f). There are three solutions, the change-point at location 9 still provides a gain of 0.02, the change-point at location 31 a gain of 3.99, and the last change-point at location 13 provides a gain of 0.12 (Fig. 2.3g) . At step 4, three change-points were selected at locations 10, 20 and 30, the original change-points (Fig. 2.3h).

Features of the approach

As the convex relaxation method through the trick of LARS algorithm, we have only a single tuning parameter. Indeed, the only parameter of the method is the original number S of candidate change-points to retrieve by RBS and Group Fused LARS. By the design of the method, we note that larger values of S can be expected to yield better quality segmentation. In addition, by construction, the proposed hybrid approach is fast. The time complexity of RBS is in order of $O(dJ\log(S))$. For GFLars, time complexity is in order of $\mathcal{O}(JdS)$ but can be reduced in $\mathcal{O}(dJS)$. Therefore for both methods, choosing a value of S larger than the expected number of change-points is not time-consuming and can provide a better segmentation.

To conclude, this approach brings a huge advantage compared to convex relaxation method presented in the previous section, since it takes full advantage of uneven resolutions. Indeed, on the one hand, RBS takes into account of the full resolution of TCN dimension, on the other hand, it takes into account all informative probes of DoH dimension.



(a) TCN and respective statistic Z see Eq. (2.15)

(b) DoH and respective statistic Z see Eq. (2.15)

(c) Step 1: Two dimensional statistic Z

(d) Step 2: Two dimensional statistic Z; First breakpoint and 2 candidates

(e) Step 2: Two dimensional statistic Z; Second breakpoint

(f) Step 3: Two dimensional statistic Z; 2 additional candidates

(g) Step 4: Two dimensional statistic Z Third breakpoint

(h) Step 4: Two dimensional statistic Z with final segmentation

Figure 2.3 – Illustration of RBS algorithm.

2.4 Model selection

By construction, when S (the number of segments) increases the likelihood increases too. Therefore, $\sum_{s=1}^{S+1}\sum_{j=t_{s-1}}^{t_s-1}(c_j-\gamma_j)^2$ decreases because the more segments there are, the better estimate adjusts to the signal. As for classical regression models with a large number of covariates, it is required to add a function depending on S to penalize models with a large number of segments. The two classical criteria used to penalize the likelihood are the Akaike Information Criterion (AIC) and the Bayes Information Criterion (BIC) defined respectively by the two following equations:

$$pen_{AIC}(S) = 2S \tag{2.38}$$

$$pen_{BIC}(S) = \frac{1}{2}\log(J)2K. \tag{2.39}$$

However, [Picard et al., 2005] and [Zhang and Siegmund, 2007] have highlighted that the two criteria do not take into account the complexity of the visited models and lead to oversegmented models (with a too high number of segments).

[Birgé and Massart, 2001] and [Lebarbier, 2005] adapted the penalty for segmentation model, that is given by the following formula:

$$pen(S) = \frac{CS}{J}\left(c_1\log\left(\frac{J}{S}\right)+c_2\right). \tag{2.40}$$

Extensive simulation experiments of [Lebarbier, 2005] suggested the values $c_1 = 2$ and $c_2 = 5$. Thereafter in this thesis, we use this form of the penalization quoted previously with theses values.

$$pen(S) = \frac{CS}{J}\left(\log\left(\frac{J}{S}\right)+\frac{c_2}{c_1}\right) = \frac{CS}{J}\left(\log\left(\frac{J}{S}\right)+2.5\right) \tag{2.41}$$

An efficient data-driven way of estimating $C*$, the slope-heuristic method has been also proposed in [Lebarbier, 2005]. This method consists in detecting a large jump in the segment number selected between two very close constants . The best constant is equal to twice the values of the largest jump (see Figure 2.4).

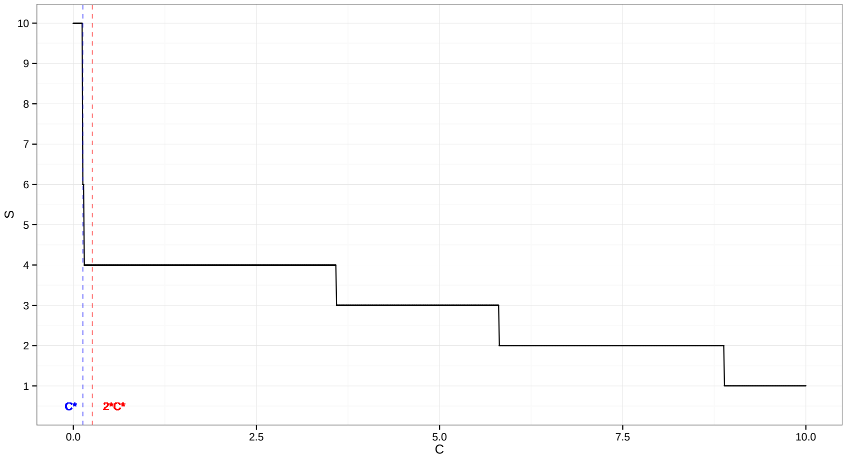


Figure 2.4 – Model selection $\widehat{S}$ in function penalty constant denote C

2.5 Conclusion

After a review of existing univariate models to segment total DNA copy number signals, we proposed two bivariate methods that are versatile for each problem of joint segmentation with an underlying piecewise constant structure in the mean. In addition, these two methods can be used on a univariate signal as well. The recursive binary segmentation, as well as group fused Lasso, are two methods that provide quickly a predefined set of change-points. Group Fused Lasso method was initially implemented in Matlab. We ported this implementation of Group Fused Lasso and RBS as an `R` package `jointseg` that is freely available from github. Possible improvements and extensions can be provided. The implementation is mostly in `R` (all except dynamic programming) but could be in `C++` in order to improve the time performance. The aim will be to keep the `R` interface in order to stay user-friendly. Model selection could also be improved. Indeed, it is well established that errors are not gaussian for TCN and DoH signals and model selection is quite sensible to this hypothesis. This violated assumption could lead to a wrong selection of the number of breakpoints. A way to make TCN more gaussian is to use logarithm or even root-square transformation. This could also improve the set of breakpoints $\widehat{m}_S$ because all models are based on the gaussian hypothesis.

The performance evaluation of all the methods is presented in the next chapter. We demonstrate on several frameworks the utility of using the two dimensions to segment the DNA copy number data. We develop a simulation framework based on real data (Chapter 3) to compare our new approaches to the existing ones. In Chapter 4, we explore a non-parametric method that uses kernel methods and does not require to transform the B allele fraction signal into DoH.

In the context of cancerology and recovery of change-points in the DNA copy number signals, joint segmentation could be applied to segment DNA copy number signals of the same patient but from different platforms as CGH, SNP arrays, sequencing. Another natural application is the segmentation of different probe types (CNV, heterozygous SNP, homozygous SNP).

Chapter 3

Performance evaluation of DNA copy number segmentation methods

3.1 Background

This chapter is extracted from the publication in Briefings in Bioinformatic. We take on the position of biologists or bioinformatician who wanted to analyze changes in DNA copy numbers from high-throughput technologies. Various methods have been proposed in the past decade for analyzing such data. From a practitioner's point of view, it is quite difficult to find which method is best for a given scientific question. In fact, it is likely that the overall difficulty of the problem depends on the context (technology, type of cancer, percentage of tumor cells). It is also likely that certain methods are more appropriate for certain contexts. Therefore, it is important to take this context into account when evaluating a set of methods, in order to 1) get a sense of the overall difficulty of the problem when interpreting the results and 2) choose appropriate methods for this context. Typically, a practitioner chooses among available data analysis methods or calibrates their parameters using a trial and error approach. A limitation of such an approach is that it is subjective, hardly reproducible and non quantitative.

The present work tackles this problem by proposing a reproducible framework for evaluating the performance of existing segmentation methods for identifying change-points from DNA copy number profiles from cancer patients. As any performance evaluation strategy, addressing this question requires the definition of three objects:

1. data with known “truth”;
2. methods to be compared;
3. criteria for performance assessment.

In this chapter, we propose such a definition and illustrate how it may be used to compare segmentation methods. The main contributions of this work are

- a framework to generate realistic DNA copy-number profiles with known “truth”. This framework is generic and may be applied to any copy number data set;
- a framework to address the question of which SNP array data segmentation method performs best, depending on biologically relevant parameters.

These frameworks are implemented in the `R` packages `jointseg` and `acnr`.

The rest of this chapter is organized as follows. We start by describing our proposed data generation framework (Section 3.2). Then, we describe the pipeline we use for evaluating segmentation methods (Section 3.3). Finally, the result of our comparison study on two data sets are reported in Section 3.4.

3.2 Generating data with known “truth”

3.2.1 Review of existing approaches

A number of data generation mechanisms have been proposed in the context of performance evaluation of DNA copy number analysis in cancer samples, either in comparison studies [Willenbrock and Fridlyand, 2005, Mosén-Ansorena et al., 2012, Lai et al., 2005, Hocking et al., 2013], or in papers describing new analysis tools. The generation of data with known “truth” can be done using either simulated or real data, both of which have advantages and drawbacks.

At first glance, simulated data are more appealing than real data because (i) “truth” is known with no ambiguity, (ii) the level of difficulty of the problem can be tuned as desired, and (iii) a large number of simulated data sets can be generated. As most DNA copy number segmentation methods rely on a Gaussian model (see Section 2.2), their performance is usually assessed using Gaussian simulations (see, for example, [Picard et al., 2005, Zhang and Siegmund, 2007]). While we do not question the usefulness of model assumptions for building statistical methods and for testing implementations, we believe that performance evaluation should as much as possible avoid relying on on a particular model. A recent study which compared several approaches for segmenting univariate DNA copy number profiles using the multiple change point approach showed that the best performing methods on Gaussian simulations performed quite poorly on real data [Rigaill et al., 2013, Table 3]. In the remainder of this section, we briefly review some existing approaches that have tried to take the best of both the "simulated data" and the "real data" worlds:

An automatically annotated data set [Willenbrock and Fridlyand, 2005]. The authors analyzed real data using one particular segmentation method to generate "truth". They then used resampling to generate realistic copy-number profiles, where (Gaussian) noise was added in order to control the signal-to-noise ratio of the data set. Two drawbacks of this approach are that the notion of "truth" depends on the chosen segmentation method, and that the problem difficulty is not driven by biological considerations.

A dilution series [Staaf et al., 2008]. In order to address the latter point, [Staaf et al., 2008] have produced a dilution data set, where DNA from a lung cancer cell line is mixed with matched blood DNA from the same patient with varying (and known) mixture proportion (see description in section E.1). Therefore, the fraction of tumor cells in the mixture controls the difficulty of the problem. The "truth" is a panel of regions whose DNA copy number status in the cell line (normal, gain, hemizygous deletion, copy-neutral LOH . . .) is known. This evaluation method has been accepted as a *de facto* standard and has been used in several subsequent papers, including [Chen et al., 2011, Olshen et al., 2011, Rancoita et al., 2010].

An important drawback of this evaluation framework is that it focuses on a very limited number of regions (ten), which results in very little discrimination between most methods in realistic settings. For example, four of the six methods compared in [Olshen et al., 2011] reach maximum sensitivity in all 10 regions for tumor cell fractions greater than 25%. In practice, samples with less than 50% are rarely analyzed, in particular because the performance of most methods typically decreases severely when the fraction of tumor cells is less than 75%. We also note that sensitivity and specificity are evaluated separately in [Staaf et al., 2008], and this weakness has been perpetuated in all subsequent papers based on the same evaluation framework.

A manually annotated data set [Hocking et al., 2013]. The authors analyzed hundreds of neuroblastoma array-CGH profiles in order to define regions containing breakpoints (true signals), and regions not containing breakpoints (false signals). This data set is freely distributed on CRAN[1]. Based on this large data set with known truth, the authors have performed a comprehensive comparison of segmentation methods for array-CGH data based on ROC curves. A drawback of this evaluation framework is that once a particular data set is chosen, it is not possible to tune the signal-to-noise ratio of the problem. Moreover, annotating a new data set is a challenging task, because it has to be large enough to contain a set of change-points that discriminate between competing segmentation methods.

A simulation model [Mosén-Ansorena et al., 2012]. The authors designed a complex simulation model to generate "realistic" copy-number profiles. This model is implemented in the R package `CnaGen`, which is available from the authors' web page[2]. The simulation model depends on 24 parameters[3]. Some of them are directly driven by biological considerations, such as the percentage of tumor cells in the sample or intra-tumor heterogeneity. We empirically found it difficult to find a combination of parameters that yield realistic copy-number profiles. This may be due to the fact that the underlying data generation model is Gaussian. Table 3.1 summarizes the features of approaches reviewed above.

Reference	Willenbrock and Fridlyand	Staaf et al.	Hocking et al.	Mosén-Ansorena et al.	This chapter
Based on real biological data?	√	√	√	-	√
Noise level based on a biological parameter?	-	√	-	√	√
Data generation possible?	√	-	-	√	√
Available as an R package?	√	-	√	√	√

Table 3.1 – Features of existing frameworks for real copy number data with known "truth".

3.2.2 Proposed data generation mechanism

Based on these considerations, we propose an original data generation framework which aims at combining the advantages of all of the above-mentioned existing approaches. Two necessary and sufficient ingredients for generating a copy-number profile of length J are:

- truth, in the form of S breakpoint positions (out of $J-1$ intervals between two successive loci) and $S+1$ copy-number state labels for all $S+1$ regions between two consecutive breakpoints;

[1] http://cran.r-project.org/web/packages/neuroblastoma/

[2] http://web.bioinformatics.cicbiogune.es/cnagen/

[3] `CnaGen` version 2.1.

- signal, in the form of locus-level data. For SNP arrays, this is generally a $J \times 3$ matrix of total copy numbers ($\mathbf{c}$), allelic ratios ($\mathbf{b}$), and germline genotypes.

Our proposed approach is described below.

Generation of "truth"

When breakpoints and region labels are not user-supplied, we propose the following approach for generating them:

breakpoints: given a signal length J, draw S breakpoint positions uniformly out of the $J-1$ possible intervals between successive data points (vertical red lines in Figure 3.1);

region labels: draw $S+1$ region labels from a pre-defined set of copy-number state labels, such as normal, gain of one copy, hemizygous deletion, homozygous deletion, copy-neutral LOH (labels on top of each plot in Figure 3.1). By default, all region labels are equiprobable, but the user may provide a vector of probabilities for each desired region label. By default, successive regions are constrained in such a way that only one of the two parental copy numbers changes at the breakpoint. Not adding such a constraint would be equivalent to allowing two distinct biological events to occur at the same genomic position, which is possible in theory but rarely observed in practice.

Generation of locus-level data

Given breakpoint positions and region labels, we generate a copy-number profile as follows: for each region of size n_R between two breakpoints, we sample n_R data points from a real copy-number data corresponding to this type of region.

The data generation mechanism therefore relies on real data where the underlying region label is (assumed to be) known. We have made available two such "real data sets with known truth" in the package: each of them corresponds to a different SNP array platform (Affymetrix or Illumina), and both of them are taken from dilution series, consisting of mixtures of DNA from a tumor cell line and from blood cells originating from the same patient, with varying mixture proportions. For both data sets, we have selected several genomic regions which are representative of the diversity of copy-number states that are typically observed in tumor samples. Contrary to [Willenbrock and Fridlyand, 2005], these labels do not rely on any automatic segmentation or calling method. Both data sets are described in Appendix E.1.1.

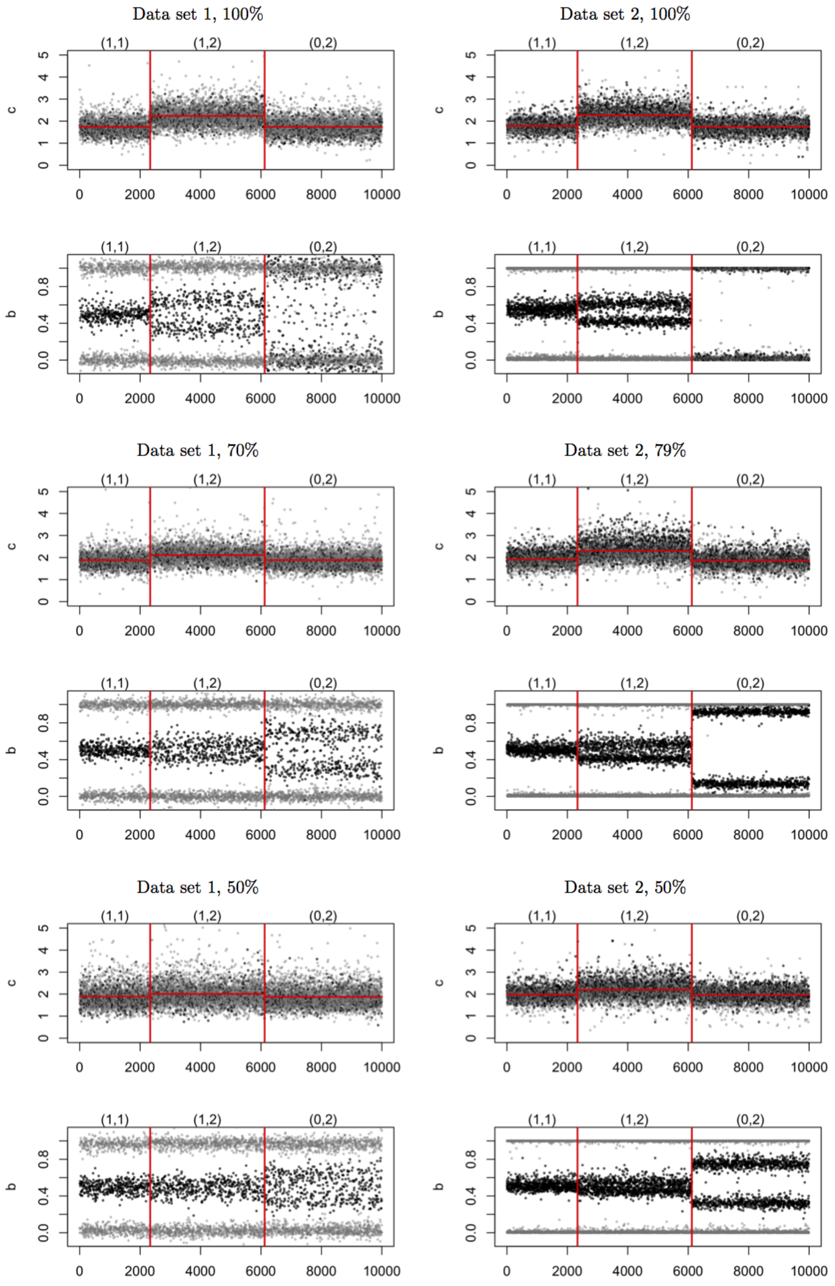


Figure 3.1 – Illustration of the variety of copy-number profiles that can be generated from the same "truth" as in Figure 1.9. Each block of two plots corresponds to total copy numbers (c) and allelic ratios (b) for one particular combination of fraction of tumor cells (in rows) and data set (in columns). Red vertical lines represent change points. SNPs that are heterozygous in the germline are colored in black; all of of the other loci are colored in gray.

3.2.3 Features of the proposed data generation mechanism

Our proposed data generation mechanism enjoys the following features:

- simplicity: small number of required parameters, all of which have a clear biological interpretation. In particular, for a given data set, the noise level is governed by the fraction of tumor cells. This is illustrated by Figure 3.1;

- flexibility: the user may specify breakpoint positions and region labels directly, if desired. Therefore, it is also possible to generate profiles with the same underlying "truth", but with different SNR, as illustrated by Figure 3.1;

- reliability: copy-number regions were identified using the profiles with 100% tumor cells. In these profiles, the region labels may be defined manually unambiguously. Because the same tumor cell line is used for the dilutions series from a given platform, the regions identified on the profiles with 100% tumor cells can also be considered as ground truth for the profiles with less tumor cells, where direct manual identification would have been more problematic;

- versatility: the design choice of separating "truth" generation from locus-level data generation implies that it is relatively easy to:
  - annotate a new data set. Although dilution series are not publicly available for all possible platforms, it is also possible to annotate representative profiles from a given data set. Moreover, annotating a new data set is not time-expensive, as one only needs to identify a few copy-number regions.
  - extend the framework to other data types (for example array-CGH or high-throughput exome capture or whole genome sequencing) is straightforward: only a set of annotated data is required.

3.3 Evaluation pipeline

Now that we have a framework to generate data, we describe how to evaluate the performance of segmentation methods.

3.3.1 Benchmark

Synthetic copy-number profiles were generated as described in Section 3.2:

region-level "truth" : Each profile contains $J = 200,000$ loci in copy number signal and $S = 20$ breakpoints. We chose to impose the constraint that on average, 90% of segments are either normal (1,1), copy-neutral LOH (0,2), single copy-gain (1,2) or hemizygous deletion (0,1). The remaining 10% of regions are given less common copy-number states, such as homozygous deletion, or balanced duplication. These parameters were inspired by our experience with SNP array data from The Cancer Genome Altas (TCGA), especially on ovarian cancers, where normal regions and regions of copy-neutral LOH, single copy-gain, and hemizygous deletion are fairly common, while other types of alterations are much more rare [Cancer Genome Atlas Research Network and others, 2011].

locus-level data: for each of $B = 50$ such "truth" profiles, corresponding locus-level data are then generated for 100%, 70% and 50% of tumor cells for data set 1, and 100%. 79% and 50% of tumor cells for data set 2. These percentages are among those available from the dilution series from which real data was extracted, see Appendix E.1.1. Pure tumor samples (100%) are typically observed in studies about tumor cell lines, while percentages as low as 50% are typically observed in primary tumors.

3.3.2 Preprocessing

We log-transformed total copy numbers to stabilize their variance and smoothed outliers using `smooth.CNA` [Olshen et al., 2004] as it improved segmentation results for all methods. Allelic ratios were converted to (unimodal) decrease in heterozygosity (d) as described in Section 1.4.

3.3.3 Compared segmentation methods

We evaluated different types of methods belonging to the different classes described in Section 2.1: multiple change-point, recursive, fused, and HMM-based methods. These methods are described in Table 3.2, where we mention which of them are able to process both signal dimensions (c and d) or only one of them.

				Time (s)		
Name	R package	function	dims	J=10^4	J=10^5	Ref
			Multiple change-point			
DP	cghseg	segmeanCO	1d	0.24	2.37	Rigaill
CST	cnaStruct	segment	2d	120	fail	Mosen-Ansorena and Aransay
DP	jointseg	doDynamicProgramming	2d	140	fail	
			Recursive			
CBS	DNAcopy	segment	1d	0.34	1.69	Venkatraman and Olshen
PSCBS	PSCBS	segmentByPairedPSCBS	2d	1.04	4.00	Olshen et al.
RBS	jointseg	doRBS	2d	0.15	1.15	Gey and Lebarbier
			Fused			
GFLars	jointseg	doGFLars	1d	0.29	3.70	Harchaoui and Lévy-Leduc
GFLars	jointseg	doGFLars	2d	0.08	0.60	Bleakley and Vert
			HMM			
PSCN	PSCN	segmentation	2d	7.25	73	Chen et al.

Table 3.2 – List of DNA copy number segmentation methods evaluated.

Not all of these methods were implemented in R. We ported from Matlab GFLseg[4] to R the implementation of multi-dimensional dynamic programming and the group-fused LARS [Bleakley and Vert, 2011], and we implemented recursive binary segmentation [Gey and Lebarbier, 2008] in R. In practice, as recommended by [Gey and Lebarbier, 2008, Harchaoui and Lévy-Leduc, 2008, Bleakley and Vert, 2011], both group-fused LARS and recursive binary segmentation are used to quickly identify a list of *candidate* change points, which is then pruned using dynamic programming (Chapter 2).

All of the compared methods are reasonably fast and memory-efficient, except those based on two-dimensional dynamic programming (DP): cnaStruct and our implementation of DP in R . Indeed, two-dimensional DP is quadratic in time and memory and thus cannot handle profiles of size $J = 10^5$. It may be surprising that the two-dimensional version of GFLars is faster than its one-dimensional counterpart. This is a consequence of the fact that the number of informative markers is several times larger for (c) than for (d) (as explained in Section 1.4). As the implementation of GFLars does not handle missing values, the 2d version of GFLars was applied to non-missing entries in (c, d), while the 1d version was applied to a much longer signal (all (c) entries). This phenomenon does not happen for other two-dimensional segmentation methods as their implementation does handle missing values.

3.3.4 Criteria for performance evaluation

Comparison studies typically assess the performance of DNA copy number analysis methods either in terms of their ability to accurately identify breakpoint locations [Lai et al., 2005, Hocking et al., 2013], copy-number states [Staaf et al., 2008, Mosén-Ansorena et al., 2012], or both [Willenbrock and Fridlyand, 2005]. This chapter focuses on the former only, because we are interested in comparing segmentation methods. The problem of evaluating strategies for calling copy-number states is left for future work.

As our proposed data generation framework provides copy number profiles with known "truth", a natural way to evaluate the performance of a given method is to cast the problem of breakpoint detection as a binary classification problem. Specifically, for each generated copy number profile, we know where the true breakpoints are located. The number of true positives TP is the number of true breakpoints for which at least one breakpoint is detected closer than a given tolerance parameter. The number of false positives FP is defined as FP=P-TP, where P is the number of "positives", that is, the total number of detected breakpoints. With this definition, whenever a method identifies two or more breakpoints within the tolerance area of a true breakpoint, one

[4] Available at http://cbio.ensmp.fr/~jvert/svn/GFLseg/html.

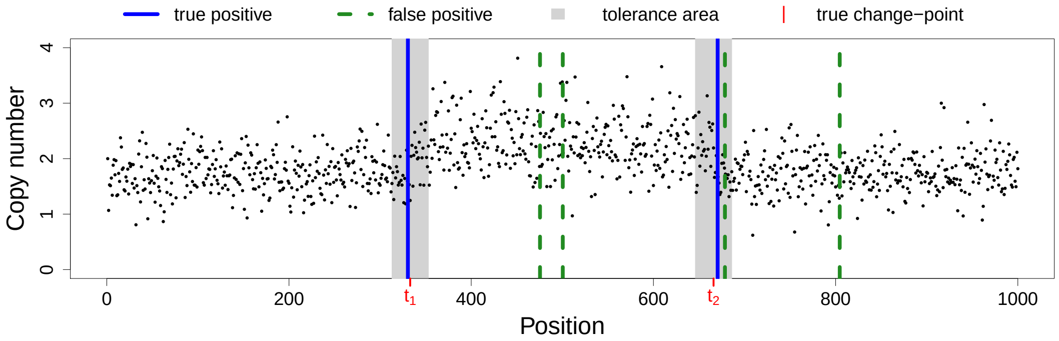


Figure 3.2 – Definition of false positive and true positive to build performance evaluation.

of these breakpoints counts as a true positive, while all others count as false positives. This definition of true and false positives is illustrated by Figure 3.2, where gray areas highlight tolerance areas around the true change-points, whose positions are identified as t_1 and t_2 on the x axis. In this example, breakpoints were detected in both shaded areas, therefore the number of true positives (solid blue lines) is two. There are four false positives (dashed green lines): one in a gray area where there is already one true positive, and three which are not within the tolerance area of any true breakpoint. Alternative definitions of true and false positives may be considered. Some of these alternatives are implemented in the `jointseg` package, including one in which a second breakpoint found within a tolerance area is not counted as a false positive. We chose to stick with the above-described evaluation (where such breakpoints are called false positives) in order not to favor methods such as the (group) fused lasso that tend to systematically find multiple breakpoints very close to each other, which is generally inconsistent with the biology of cancers.

Related works. A similar definition of true and false positives is used in [Willenbrock and Fridlyand, 2005], although the authors do not mention how the above case of multiple breakpoints within the tolerance area is handled. Another related approach has been proposed in [Hocking et al., 2013]. There, copy-number profiles are real, array-CGH profiles for which regions containing a breakpoint and regions containing no breakpoints have been delineated by experts. The main difference is that only a subset of the "true" and "false" breakpoints are annotated, and that the tolerance parameter cannot be tuned without the expert re-annotating the data set. Finally, a similar type of evaluation has been used by [Lai et al., 2005], at the the locus level instead of the breakpoint level. This locus-level based evaluation method tends to favor segmentation methods that accurately identify large altered regions, even if they fail to detect breakpoints delineating smaller altered regions.

3.3.5 ROC-based evaluation

Usually, each method provides a segmentation and its associated set of breakpoints. This can be translated into a measure of sensitivity and specificity using the above definition of true and false positives. However, the methods have to be compared at the same specificity or sensitivity level in order for this comparison to be fair. Ideally, we would like to compute a Receiver Operator Characteristic (ROC) curve for each method. In order to do this, one needs to explore a large set of possible segmentations with varying sensitivity and specificity, obtained by exploring the set of tuning parameters of each method. Such an exhaustive exploration is tedious and time consuming as soon as the number of parameters is larger than 2 or 3, and may lead to over-optimistic results. To overcome this problem, we adopted the following strategy: for any given method m, we recovered a segmentation in k_m change points using default parameters, and we retrieved for each $k \in \{1 \dots k_m\}$ the best k subset of these k_m using dynamic programming. Another possible strategy would be to sort the k_m change points according to a measure of confidence.

One could be worried that the range of explored sensitivity/specificity is highly variable across methods. In practice, our experience is that the default parameters of a method generally tend to over-segment the data and that typically, most of the true change points are found, at the cost of a more or less large number of false positives. This is in agreement with [Hocking et al., 2013].

3.4 Results

3.4.1 Quantifying problem difficulty for known change points

Segmentation methods rely on a statistic to quantify the biological difference between any two regions. Based on this statistic, they aim at locating a good set of regions or equivalently, of change points. This location problem is combinatorial in nature. In this section, we try to quantify this biological difference independently of this combinatorial problem. In order to do this, we assume that change point positions are given *a priori* and we compare the power to call a change using total copy numbers (c) or allelic signals (d) for different types of change points. In order to perform this power study, we need to formally define the notion of power, or signal-to-noise ratio (SNR), between copy number regions. We chose a definition of SNR which is consistent with our proposed data-generation mechanism, in which DNA copy number data from a given region are sampled from a population which represents the corresponding copy-number state (see Section 3.2.3). Let us consider two regions and label by "0" and "1" the copy number state of two regions. For univariate signals (c or d), a natural definition of SNR is the (squared) Z statistic of the comparison between the sample means of region "0" and region "1":

$$\mathrm{SNR}(c) = \frac{(\bar{c}_0 - \bar{c}_1)^2}{\sigma_{c,0}^2/n_0 + \sigma_{c,1}^2/n_1} \tag{3.1}$$

$$\mathrm{SNR}(d) = \frac{\left(\bar{d}_0 - \bar{d}_1\right)^2}{\sigma_{d,0}^2/n_0^\star + \sigma_{d,1}^2/n_1^\star}\,, \tag{3.2}$$

where n_i is the total number of loci in region i, $\bar{c}_i$ and $\sigma_{c,i}$ are the sample mean and population standard deviation of total copy numbers in state i and $\bar{d}_i, \sigma_{d,i}$ are the sample mean and population standard deviation of the decrease in heterozygosity in state i. Note that the decrease in heterozygosity is only defined for SNPs that are heterozygous in the germline, whereas the total copy number is defined for all loci. Therefore, $\bar{d}_i$ is calculated based on $n_i^\star$ heterozygous SNPs, while $\bar{c}_i$ is calculated based on all n_i loci. For a given DNA sample, the fraction of heterozygous SNPs among those present on the microarray is typically close to 1/3; moreover, data set 1 contains not only SNP probes but also non-polymorphic loci, with a 1:1 ratio. As a result, the fraction $n_i^\star/n_i$ is approximately 1/6 for data set 1 and 1/3 for data set 2. A natural extension of this definition of SNR to the two-dimensional case of the statistic (c, d) is

$$\mathrm{SNR}(c, d) = \left(\bar{c}_0 - \bar{c}_1, \bar{d}_0 - \bar{d}_1\right)(S_0 + S_1)^{-1}\left(\bar{c}_0 - \bar{c}_1, \bar{d}_0 - \bar{d}_1\right)'\,, \tag{3.3}$$

where S_i is the population covariance matrix of the bivariate vector (c, d), that is $S_i = \begin{pmatrix} \sigma_{c,i}^2/n_i & \tau_{cd,i}/n_i^\star \\ \tau_{cd,i}/n_i^\star & \sigma_{d,i}^2/n_i^\star \end{pmatrix}$ with $\tau_{cd,i}$ the covariance between c and d in state i. In practice, the population parameters for copy-number state i (that is, $\sigma_{d,i}$, $\tau_{cd,i}$, and $\sigma_{d,i}$) are calculated from the annotated data. The sample parameters ($\bar{c}_i$ and $\bar{d}_i$) are calculated from samples of n_i and $n_i^\star$ loci, respectively. Note that SNR(c) and SNR(d) are comparable with each other since they follow (non-centered) χ^2 distributions with 1 degree of freedom under the null hypothesis of no breakpoint between state 0 and state 1.

By definition, SNR is an increasing function of the length of each flanking segment. For $i \in \{0, 1\}$, we chose $n_i = 500$. $n_i^\star$ depends on the proportion of heterozygous SNPs in the sample; as explained above, it is very close to $n_0/6$ for data set 1 and $n_0/3$ for data set 2. Therefore, the length of the flanking regions essentially acts as a

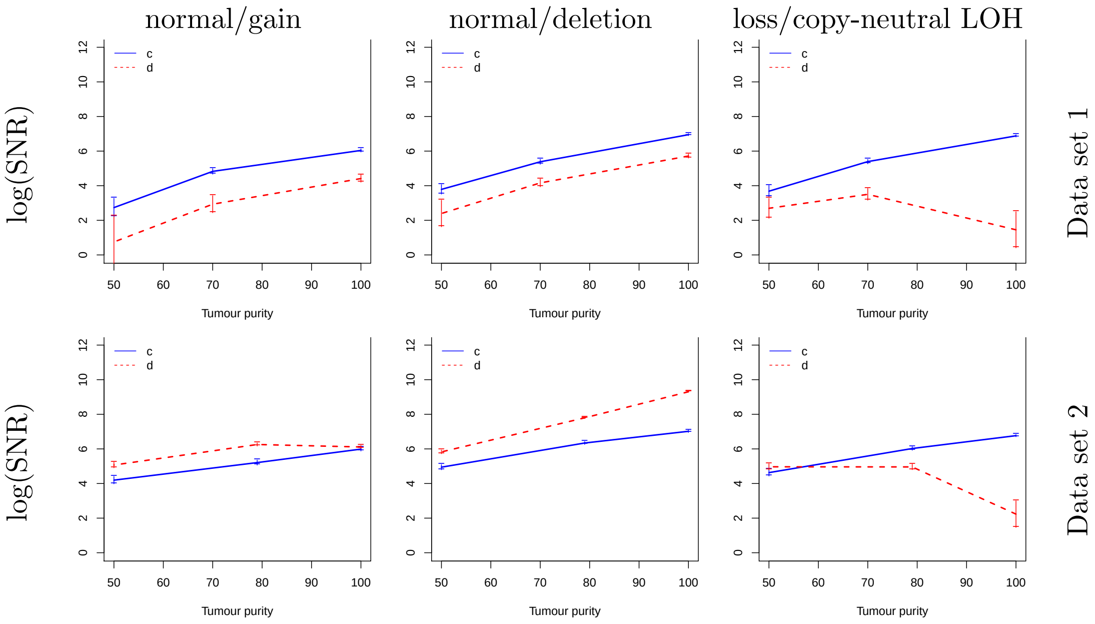


Figure 3.3 – Average log(SNR) and corresponding standard errors across 100 samples as a function of the percentage of tumor cells for total copy numbers (c, solid blue lines) and allelic ratios (d, dashed red lines). Each column corresponds to a type of copy number transition. Each row corresponds to a given data set.

constant scaling factor across all transitions and settings. Therefore, SNR only reflects differences between the underlying copy number states. Figure 3.3 shows the average (and standard error) of log(SNR) across 100 samplings for three levels of tumor purity level, for three common types of copy number transitions for data set 1 (top panel) and data set 2 (bottom panels). Several conclusions may be drawn:

- **Difficulty generally increases with normal contamination**: SNR generally increases with the percentage of tumor cells. This is true for all types of transitions for c. For d, the only situation in which SNR is not an increasing function of tumor purity is the case of transitions between loss and copy-neutral LOH (Figure 3.3, rightmost column). This is expected theoretically because both of these states correspond to LOH in the tumor cells of the sample, implying that the true d in these cells is 1. In presence of normal cells, d estimates in both states are shrunk d toward 0, but in a state-specific way (see [Bengtsson et al., 2010, Figure 4] for a detailed explanation of this phenomenon);

- **SNR levels depend on the type of copy number transition** for a given data set (that is, for a given row in Figure 3.3). This holds for both statistics (c or d). Note that in the case of c, this is unexpected, as all plotted transitions correspond to a one-copy gain.

- **Possibly low power.** Note that in some cases (e.g. data set 1, (a) and (c)), the computed SNR is lower than 2. Under the null hypothesis of no difference in mean levels, SNR follows a centered $\chi^2(1)$ distribution, so that this range of observed SNR correspond to p-values of the order of 1%, which is not low considering the large number of data points ($n_i = 500$).
- **Neither c or d is always the best statistic**. For a given type of transition (that is, for a given column in Figure 3.3) and a given statistic, the trend in SNR is comparable across data sets. However, the relative power of c with respect to d is much higher for data set 1 than for data set 2. This is directly related to the above-mentioned difference between ratios $n_i^\star/n_i$ of the number of informative loci for each statistic.

In this subsection, we assessed the intrinsic difficulty of calling a change point if the positions to test are known *a priori*. This study suggests that c and d are complementary sources of information, implying that change point detection methods should ideally take both of them into account. This study also sheds light on the fact that low percentages of tumor cells severely impacts SNR. In the remaining subsections, we assess the ability of segmentation methods to recover the true location of change points.

3.4.2 Robustness of the evaluation to the tolerance parameter

Our first goal was to check the influence of the tolerance parameter on the methods' performance. Our simulations were run using data generation as described in section 3.3.1. We computed partial areas under the ROC curves (pAUC) with a number of false positives between 0 and 10. Mean and 95% confidence intervals of pAUCs across simulation runs were calculated for each method for 5 values of the tolerance parameter (1, 2, 5, 10 and 20). For example, a tolerance of 5 means that a breakpoint is considered correct if it lies within 5 data points of the true breakpoints (see section 3.3.4 for more details). These results are reported in Figure 3.4 in the scenario without normal contamination. Similar results were observed for other scenarios.

Increasing tolerance clearly increases pAUC for all methods. This is the case even in the arguably "simple" scenario where no normal cells are present. However, in most cases, the ranking of all methods is not affected by tolerance. Based on these results,

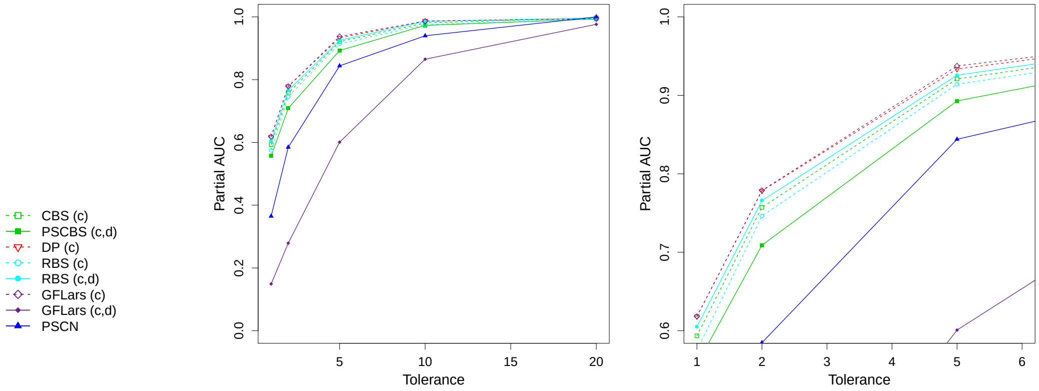


Figure 3.4 – Method performance increase with the tolerance parameter for both data sets. Partial AUC for FP ≤ 10 for data set 1 and 100% tumor cells.

we decided to report only pAUC for one particular value of tolerance: 5 loci on each side of the breakpoints.

3.4.3 Joint segmentation generally increases performance

This section aims at comparing the quality of segmentations obtained using total copy numbers only (c), allelic ratios only (d), and both of them (c, d) and how the quality of the segmentation is affected by the purity of the sample. As explained in section 3.4.1, it is typically expected that localization of the breakpoints is easier using both dimensions of the signal. In order to do so, we compared 6 scenarios corresponding to two data sets and three levels of purity (high, intermediate and low). Table 3.3 reports the pAUC of the best (c), (d) and (c, d) methods for data set 1 and 2, respectively. Detailed results for all methods are presented in Table 3.4.

For both data sets it is quite clear that performance in terms of pAUC severely deteriorates when the level of contamination increases. (c) methods perform better than (d) methods for high level of purity. For example in the case of data set 2 the minimum difference in pAUC between (c) and (d) is 19% for high level (Table 3.4). For an intermediate level of purity, for data set 1 (c) outperforms (d) with a minimum pAUC difference of 41% and for data set 2 (c) is similar to (d). For a low level of purity, the pAUCs are low or very low for both data sets; for data set 1, (c) outperforms (d) with a minimum pAUC difference of 6%; for data set 2, (d) outperforms (c) with a minimum pAUC difference of 15%. These observations are in agreement with the results of Section 3.4.1. The difference between data sets 1 and 2 can be explained by the fact that the proportion of informative markers is different, namely around 1/6 and 1/3, respectively. This low proportion of informative markers also explains the poor performance of GFLars (c, d) (which could also be seen in Figure 3.4), as the current implementations of 2d GFLars do not handle missing values in one of the dimensions.

Not all (c, d) methods outperform (c)-only and (d)-only methods. For example, for data set 1 and 100%, although PSCBS has good performance, it is outperformed by 2 to 5 % by all (c) methods. However, as can be seen in Table 3.3, there are always several (c, d) approaches among top performers.

3.4.4 Choosing the appropriate method for a given context

In practice, when analyzing SNP array data, biostatisticians and bioinformaticians will choose one particular method to perform data segmentation. This choice is often *ad hoc* and based on personal experience. Our purpose here is not to make a comparison of all existing segmentation methods, but to compare relevant candidates in different classes of approaches. In the settings that we have considered it seems that RBS (c, d) performs very well. However, the point of our framework is not to select once and for all a best segmentation tool, but rather to justify the use of one method for one particular type of scenario (cancer type, cellularity, data set). In particular, we make no claim about the performance of RBS for other data sets.

	Data set 1			Data set 2		
Statistic	100%	70%	50%	100%	79%	50%
(c, d)	0.93	0.63	0.22	0.97	0.95	0.75
(c)	0.94	0.64	0.18	0.96	0.89	0.49
(d)	0.35	0.18	0.10	0.71	0.84	0.67

Table 3.3 – Best pAUC across methods for each combination of statistic, data set and percentage of tumor cells.

3.4.5 Heterogeneity of breakpoint detection difficulty

An important question when using a biostatistical or bioinformatic tool is to assess its ability to recover events and to know which events they are likely to find and which of them are harder to detect. In Table 3.3 it can be seen that the pAUC is never at 100%. This is not necessarily surprising as the signal is quite noisy. In fact considering noise level the pAUC is quite high. Figure 3.5 demonstrates that (as could be expected) missed change-points are those for which we have a low signal to noise ratio (the right

		Data set 1			Data set 2		
Statistic	Method	100%	70%	50%	100%	79%	50%
(c, d)	PSCBS	0.89	0.60	0.16	0.97	0.88	0.51
	GFLars	0.60	0.42	0.14	0.97	0.91	0.60
	RBS	0.93	0.63	0.22	0.97	0.95	0.75
(c)	CBS	0.92	0.59	0.16	0.91	0.84	0.45
	GFLars	0.94	0.64	0.18	0.96	0.89	0.49
	RBS	0.91	0.62	0.17	0.90	0.84	0.48
	cghseg	0.93	0.61	0.18	0.95	0.88	0.49
(d)	CBS	0.35	0.17	0.10	0.71	0.83	0.64
	GFLars	0.35	0.18	0.10	0.71	0.84	0.66
	RBS	0.34	0.17	0.09	0.69	0.83	0.65
	cghseg	0.35	0.18	0.10	0.70	0.84	0.67

Table 3.4 – pAUC by for each combination of method, statistic, data set and percentage of tumor cells.

panel is darker than the left panel). However, the signal to noise ratio substantially depends on the type of change-point. Typically, in Figure 3.5 the column corresponding to the (0,2)-(1,2) transition is much darker than that of the (1,1)-(1,2) transition. This is confirmed by Table 3.5, which indicates that for a high level of normal contamination in data set 2, the proportion of missed (1,1)-(1,2) change-points is greater than 1/2.

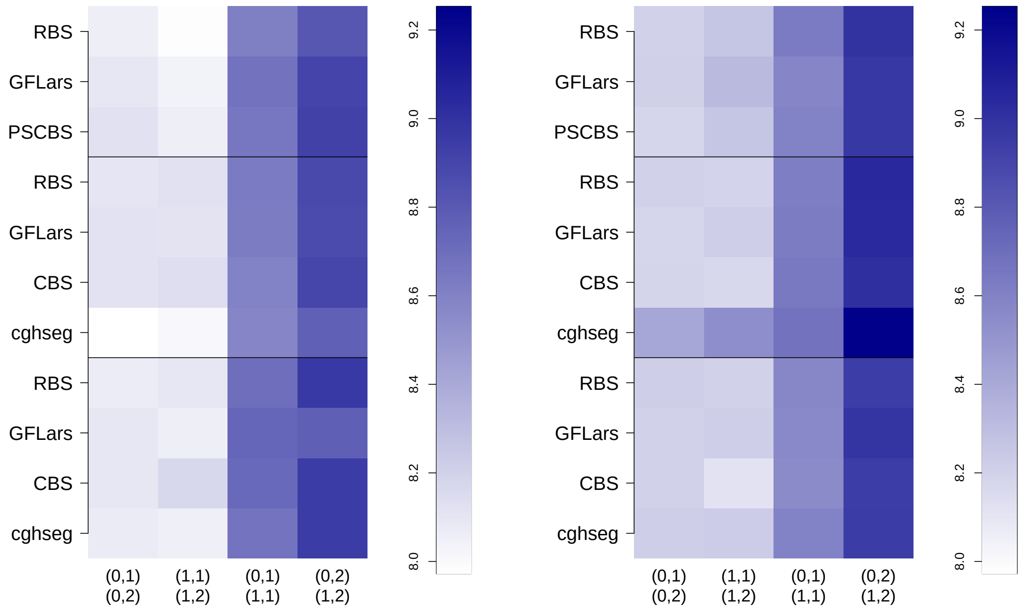


Figure 3.5 – log(SNR) for missed (left) and caught (right) breakpoints for four types of breakpoints on data set 2 with 50% normal cell contamination.

Statistic	Method	(0,1)-(0,2)	(1,1)-(1,2)	(0,1)-(1,1)	(0,2)-(1,2)
(c, d)	RBS	**0.40**	**0.47**	**0.32**	0.31
	GFLars	0.51	0.66	0.44	0.34
	PSCBS	0.55	0.63	0.51	0.47
(c)	RBS	0.57	0.69	0.52	0.63
	GFLars	0.54	0.70	0.45	0.58
	CBS	0.59	0.71	0.52	0.62
	cghseg	0.66	0.79	0.55	0.69
(d)	RBS	0.49	0.54	0.39	0.24
	GFLars	0.49	0.51	0.34	**0.20**
	CBS	0.51	0.49	0.41	0.23
	cghseg	0.51	0.51	0.38	0.23

Table 3.5 – Proportion of missed breakpoints by method, statistic and type of copy-number transition (data set 2, 50% of tumor cells).

3.5 Summary and discussion

We have developed a framework to assess the performance of various DNA copy number segmentation methods. A critical aspect of this framework is that it generates realistic copy-number profiles by resampling real SNP array data. This allows us to study a large number of scenarios without relying on a particular statistical model. It is our opinion that this framework is simple to use as it depends on few parameters, all of which have a straightforward biological interpretation. An R package is available and we believe that our proposed data generation scheme can be used readily as well as applied to other data sets and technologies. It is also possible to extend the set of segmentation methods compared, as explained in the package documentation. In this chapter, we illustrated the usage of this framework on two SNP array data sets from Affymetrix and Illumina.

We were able to identify which technological and biological parameters drive the performance of segmentation methods. First, it appears that the percentage of tumor cells in the sample plays a critical role: for a percentage lower than 70%, it is probably hopeless to recover the whole set of breakpoints with a high accuracy. We emphasize the relevance of the considered range of cellularity for applications: we expect that tumor cell lines should be well represented by the 100% setting, while the 50% is not unusual for clinical practice. Second, it seems that different microarray technologies might lead to different performances. Specifically, the ratio between the number of informative allelic probes (heterozygous SNPs) to the total number of probes is a crucial aspect, particularly for a high level of normal contamination. Finally, not all methods achieve similar performance across the scenarios that we have considered. Interestingly, we show that methods that take advantage of both signal dimensions are generally but not always better than those using only one of them. This variability between segmentation methods may be attributed to some extent to the biological and technological contexts, in the sense that some methods might be more adapted to certain scenarios.

Our framework provides a way to critically evaluate the performance of segmentation methods, and therefore to rationally select one or several of them for a particular data set. Such a quantitative assessment is also useful for interpretation. For example, we showed that even in favorable scenarios, performances are not perfect. Furthermore, perhaps unexpectedly, we showed that copy number transitions involving the gain or loss of a single DNA copy are not equally easy to recover, meaning that the proportion of different types of copy number transitions recovered by a particular segmentation method may not be directly interpretable.

Chapter 4

Non-parametric segmentation method using kernels

4.1 Introduction

This chapter is a collaborative work with Alain Célisse, Guillemette Marot, and Guillem Rigaill. We recently submitted this work to the *Computational Statistics and Data Analysis (CSDA)* journal. In this Chapter, we attempt to use kernel tricks in order to develop a new non-parametric segmentation procedure. This method allows us to free ourselves from the DoH transformation described in section 1.4.2. We bring several contributions to the computational aspects and the statistical performance of the kernel change-point procedure introduced by [Arlot et al., 2012] to segment separately then jointly TCN and BAF.

The model presented in this chapter is similar to the model described in Chapter 2. However, instead of considering change-points in the mean of the signal, we consider change-points in the whole distribution of the signal. Indeed, this assumption is more realistic for BAF signals for which it is clear changes don't occur only in the mean. Indeed, the distribution is multimodal for each segment. The number of the modes but also their values, and the variance vary from a segment to another. For instance, on Fig. 4.1, it is easy to observe three modes in the normal region (1,1) respectively for the three statuses of SNPs (AA, BB, and AB). However, for the gain region (1,2) where we have to observe four statuses respectively for (AAA, AAB, ABB, and BBB), modes are more difficult to distinct but it seems that variance has increased. For the third region (cn-LOH) denoted (0,2) in terms of parental copy number, both the number and the location of the modes have changed compared to the gain region. All these

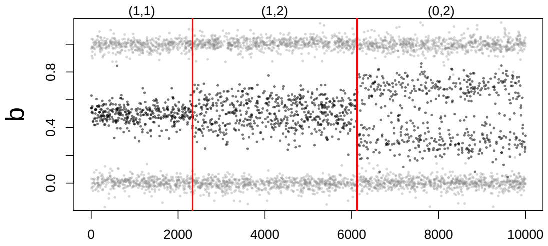


Figure 4.1 – BAF signal

observations had motivated us to o propose a non-parametric method to detect these different types of changes at the same time.

This chapter describes a new algorithm to simultaneously perform the dynamic programming step of [Harchaoui and Cappé, 2007] and also compute the required elements of the cost matrix on the fly. As a consequence, this algorithm has a complexity of order $\mathcal{O}(SJ^2)$ in time and $\mathcal{O}(SJ)$ in space (including both the dynamic programming and the cost matrix computation). This improved space complexity comes without an increased time complexity, which is a key point for genome analysis. However, to deal with larger data sets the time complexity is still high and we develop a new algorithm based on a low-rank approximation to the Gram matrix. This computational improvement is possible at the price of an approximation, which leads to (almost) the best segmentations from 1 to S segments with a complexity of order $\mathcal{O}(Sp^2J)$ in time and $\mathcal{O}((S+p)J)$ in space, where p is the rank of the approximation. Finally, we adapt the model selection of [Lebarbier, 2005] described in section 2.4 to our case and illustrate its good empirical statistical performance. Before describing the kernel segmentation method, we need to introduce some essential notions on kernels [Shawe-Taylor and Cristianini, 2004, Schölkopf et al., 2004].

Let $(\mathrm{X}_1, \ldots, \mathrm{X}_J) \in \mathcal{X}$ a signal of length J within there are some changes in the distribution. This signal can either represent the total DNA copy number or the B allele fraction. Then, we consider a positive definite kernel $k \;:\; \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. $\mathcal{H}$ is its associated Reproducible kernel hilbert space (RKHS), and $\Phi \;:\; \mathcal{X} \mapsto \mathcal{H}$ is the canonical feature map defined by $\Phi(x) = k(x, \bullet)$ (a function of $\mathcal{H}$). There exists a strong link between the canonical feature map and the kernel function since $\forall x, y \in \mathcal{H}$: $\langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}} = k(x, y)$.

Representation in RKHS $\mathcal{H}$ For every $j \in \{1, \ldots, J\}$ and $\mathrm{X}_j \in \mathcal{X}$, we define :

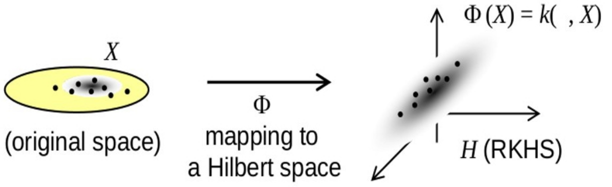


Figure 4.2 – Mapping initial data to hilbert space $\mathcal{H}$

$$\mathrm{Y}_j = \varPhi(\mathrm{X}_j) \in \mathcal{H}$$

and the mean element $\mu_j^\star \in \mathcal{H}$ of the distribution of X_j.

$$\forall f \in \mathcal{H} \quad \langle \mu_j^\star, f\rangle_{\mathcal{H}} = \mathbb{E}_{\mathrm{Y}_j} \langle \mathrm{Y}_j, f\rangle_{\mathcal{H}} = \mathbb{E}_{\mathrm{X}_j} \langle \varPhi(\mathrm{X}_j), f\rangle_{\mathcal{H}}$$

There is a strong connection between the mean element $\mu_j^\star$ and the distribution of X_j denoted $\mathbb{P}_{\mathrm{X}_j}$. Indeed, for particular kernels (namely characteristic kernels), a change in the distribution of X_j implies a change in the mean element $\mu_j^\star$.

$$\mathbb{P}_{\mathrm{X}_j} \neq \mathbb{P}_{\mathrm{X}_j} \Rightarrow \mu_j^\star \neq \mu_j^\star$$

Kernel examples: Several usual characteristic kernels are defined below if $\mathcal{X} = \mathbb{R}$.

- Gaussian kernel: $k(x, y) = \exp\left\{\frac{-\|x-y\|^2}{\delta}\right\}$
- Exponential kernel : $k(x, y) = \exp\left\{\frac{-|x-y|}{\delta}\right\}$
- Polynomial kernel: $k(x, y) = (\delta_0 + \delta_1\langle x, y\rangle)^{\delta_2}$ with $\delta = (\delta_0, \delta_1, \delta_2)$

We denote the associated Gram matrix associated by $\mathbf{K} = \{\mathbf{K}_{i,j}\}_{1\leq i,j\leq J}$, where $\mathbf{K}_{i,j} = k(\mathrm{X}_i, \mathrm{X}_j)$. The model and algorithms presented in the following can be applied for each type of different type of kernels. However, to discover breakpoints in the DNA copy number signals, we focused on characteristic kernels and more particularly on the gaussian one. After describing the model and the algorithms (sections 4.2 and 4.3), a short section described how to combine two kernels (4.4). Then, we present the model selection used in this case (section 4.5). Finally, we present global performance of the procedure at the end of the chapter 4.6.

4.2 Model in RKHS

4.2.1 Definition

In this section, we define the statistical model in the RKHS $\mathcal{H}$ based on notations presented in the previous section. The statistical model for S change points at $(t_1, \ldots, t_S)$ can be written as follows:

$$\forall j \in \{1, \ldots, J\} \qquad \mathrm{Y}_j = \mu^\star_j + \epsilon_j \tag{4.1}$$

where $\forall s \in \{1, \ldots, S+1\}, \forall j \in [t_{s-1}, t_s[\qquad \mu^\star_j = \mu^\star_s$

With this model, a change-point in the mean of Y_j is a change-point in the distribution of the X_j. Methods to detect change-points in the mean are well known and have been described previously in Chapter 2. We can proceed in the same way in $\mathbb{R}$ as in the RKHS $\mathcal{H}$ and perform the change-point detection in the mean of Y_j. We denote $\mathrm{Y} = (\mathrm{Y}_1, \ldots, \mathrm{Y}_J)$ and $\mathcal{H}^J$ $\mu^\star = (\mu^\star_1, \ldots, \mu^\star_J)$ elements of $\mathcal{H}^J$ with Euclidean structure given by $\|f - g\|^2_{\mathcal{H}^J} = \sum_{j=1}^{J} \|f(j) - g(j)\|^2_{\mathcal{H}}$ for every $f, g \in \mathcal{H}^J$.

4.2.2 Estimation of the mean element

As we said previously, we assume that $\mu^\star = (\mu^\star_1, \ldots, \mu^\star_J) \in \mathcal{H}^J$ is piecewise constant as is the distribution of the signal $(\mathrm{X}_1, \ldots, \mathrm{X}_J)$. We denote m a particular segmentation of the signal, and s a segment from this segmentation m.

The estimation of the $\mu^\star$ on a segment denoted s is defined by equation 4.2:

$$\hat{\mu}_m(s) = \frac{1}{Card(s)} \sum_{j \in s} \mathrm{Y}_j = \frac{1}{n_s} \sum_{j \in s} \varPhi(\mathrm{X}_j) \tag{4.2}$$

$\hat{\mu}_m(s)$ is the mean in space $\mathcal{H}$ of all the element Y_j that are in segment denoted s.

4.2.3 Best segmentation with S segments

Let $\mathcal{M}_J(S)$ all possible segmentations with S segments. We denote $\hat{m}_S$ the best segmentation with S segments as:

$$\hat{m}_S = \underset{m_S \in \mathcal{M}_J(S)}{\arg\min} \|\mathrm{Y} - \hat{\mu}_{m_S}\|^2_{\mathcal{H}^J} \tag{4.3}$$

$$\|\mathrm{Y} - \hat{\mu}_{m_S}\|^2_{\mathcal{H}^J} = \|\mathrm{Y}\|^2_{\mathcal{H}^J} + \|\hat{\mu}_{m_S}\|^2_{\mathcal{H}^J} - 2\langle \mathrm{Y}, \hat{\mu}_{m_S} \rangle_{\mathcal{H}^J} \tag{4.4}$$

We explain in detail the computation of each element of the equation 4.4 separately.

The first term is the norm of the vector Y in the space $\mathcal{H}$.

$$\begin{aligned}
\|\mathrm{Y}\|^2_{\mathcal{H}^J} &= \sum_{j=1}^{J} \|\mathrm{Y}_j\|^2_{\mathcal{H}} = \sum_{j=1}^{J} \langle \mathrm{Y}_j, \mathrm{Y}_j \rangle_{\mathcal{H}} = \sum_{j=1}^{J} \langle \Phi(\mathrm{X}_j), \Phi(\mathrm{X}_j) \rangle_{\mathcal{H}} \\
&= \sum_{j=1}^{J} k(\mathrm{X}_j, \mathrm{X}_j) \qquad (4.5)
\end{aligned}$$

The second one is the norm of the estimation of the mean element for the segmentation m_S.

$$\begin{aligned}
\|\hat{\mu}_{m_S}\|^2_{\mathcal{H}^J} &= \sum_{j=1}^{J} \|\hat{\mu}_{m_S}(j)\|^2_{\mathcal{H}} = \sum_{j=1}^{J} \langle \hat{\mu}_{m_S}(j), \hat{\mu}_{m_S}(j) \rangle_{\mathcal{H}} \\
&= \sum_{j=1}^{J} \langle \frac{1}{n_{s(j)}} \sum_{j' \in s(j)} \mathrm{Y}'_j, \frac{1}{n_{s(j)}} \sum_{l \in s(j)} \mathrm{Y}_l \rangle_{\mathcal{H}} \\
&= \sum_{j=1}^{J} \frac{1}{n^2_{s(j)}} \sum_{j' \in s(j)} \sum_{l \in s(j)} \langle \mathrm{Y}'_j, \mathrm{Y}_l \rangle_{\mathcal{H}} = \sum_{s \in m_S} \frac{n_s}{n_s^2} \sum_{j,l \in s} \langle \mathrm{Y}_j, \mathrm{Y}_l \rangle_{\mathcal{H}} \\
&= \sum_{s \in m_S} \frac{1}{n_s} \sum_{j,l \in s} k(\mathrm{X}_j, \mathrm{X}_l) \qquad (4.6)
\end{aligned}$$

where $s(j)$ is the unique segment which contains element j from segmentation m_S.

To finish, we compute the scalar product between the estimation of the mean element $\hat{\mu}_{m_S}$ and the vector Y.

$$\begin{aligned}
\langle \mathrm{Y}, \hat{\mu}_{m_S} \rangle_{\mathcal{H}^J} &= \sum_{j=1}^{J} \langle \mathrm{Y}_j, \hat{\mu}_{m_S}(j) \rangle_{\mathcal{H}} = \sum_{j=1}^{J} \langle \mathrm{Y}_j, \frac{1}{n_{s(j)}} \sum_{j' \in s(j)} \mathrm{Y}_{j'} \rangle_{\mathcal{H}} \\
&= \sum_{j=1}^{J} \frac{1}{n_{s(j)}} \sum_{j' \in s(j)} \langle \mathrm{Y}_{j'}, \mathrm{Y}_j \rangle_{\mathcal{H}} = \sum_{s \in m_S} \frac{1}{n_s} \sum_{j',j \in s} \langle \mathrm{Y}_j, \mathrm{Y}_{j'} \rangle_{\mathcal{H}} \\
&= \sum_{s \in m_S} \frac{1}{n_s} \sum_{j',j \in s} k(\mathrm{X}_{j'}, \mathrm{X}_j) \qquad (4.7)
\end{aligned}$$

We note that elements 4.6 and 4.7 are equal and hence,

$$\begin{aligned}
\|\mathrm{Y} - \hat{\mu}_{m_S}\|^2_{\mathcal{H}^J} &= \|\mathrm{Y}\|^2_{\mathcal{H}^J} - \|\hat{\mu}_{m_S}\|^2_{\mathcal{H}^J} \\
&= \sum_{j=1}^{J} k(\mathrm{X}_j, \mathrm{X}_j) - \sum_{s \in m_S} \frac{1}{n_s} \sum_{j', j \in s} k(\mathrm{X}_{j'}, \mathrm{X}_j)
\end{aligned}$$

To conclude:

$$\hat{m}_S = \underset{m_S \in \mathcal{M}_J(S)}{\arg\min} \sum_{j=1}^{J} k(\mathrm{X}_j, \mathrm{X}_j) - \sum_{s \in m_S} \frac{1}{n_s} \sum_{j', j \in s} k(\mathrm{X}_{j'}, \mathrm{X}_j) \tag{4.8}$$

This computation will be done for several number of segments ($S = 1, \ldots, S_{max}$) with dynamic programming.

4.3 Algorithms

In this section, we present new algorithms mainly proposed by Guillem Rigaill to solve (4.8). This description permits to better understand how kernel segmentation works.

The first step is to compute on the fly the elements of the cost matrix required by DP algorithm to reduce complexity. Indeed, by reordering loops in algorithm 1, we obtain a new exact algorithm which has a reduced complexity in space of order $\mathcal{O}(SJ)$ instead of $\mathcal{O}(SJ^2)$. This algorithm is described in details in 4.3.1. However, the time complexity remains the same and when J is large, $J > 10^6$, the algorithm is very consuming. Thus, we also provide an approximate algorithm but substantially faster. This algorithm reaches a time complexity of order $\mathcal{O}(SJ)$. The trick is to use a low-rank approximation to the Gram matrix and in a second step use binary segmentation heuristic (4.3.2). This allows to deal with very large data sets (when J is larger that 10^6 for instance).

As explain in section 2.2.3, the main computational cost is due to (4.8) and consists in recovering the best segmentation with S segments. That is solving (4.9) :

$$\mathcal{L}_s(1, J+1) = \min_{m_S \in \mathcal{M}_J(S)} \|\mathrm{Y} - \hat{\mu}_{m_S}\|^2_{\mathcal{H}^J}. \tag{4.9}$$

Standard dynamic programming algorithm applying to kernel segmentation

Let m_S denote a segmentation in S segments (with the convention that $t_1 = 1$ and $t_{S+1} = J + 1$). For any $1 \leq s \leq S$, the segment $\{t_s, ..., t_{s+1} - 1\}$ of the segmentation m_S has a cost that is equal to

$$\mathcal{C}_1(t_s, t_{s+1}) = \sum_{j=t_s}^{t_{s+1}-1} k(\mathrm{X}_j, \mathrm{X}_j) \; - \; \frac{1}{t_{s+1} - t_s} \sum_{j=t_s}^{t_{s+1}-1} \sum_{j'=t_s}^{t_{s+1}-1} k(\mathrm{X}_j, \mathrm{X}'_j). \tag{4.10}$$

In terms of the cost defined above, the total cost of the segmentation m_S is given by:

$$\|\mathrm{Y} - \hat{\mu}_{m_S}\|^2_{\mathcal{H}^J} = \sum_s \mathcal{C}_1(t_s, t_{s+1}),$$

which is clearly *segment additive* [Harchaoui and Cappé, 2007, Arlot et al., 2012] and a reformulation of Eq. 4.8.

Then it is easy to apply algorithm 1 described in the introduction. This algorithm, which is the one used by [Harchaoui and Cappé, 2007], suffers from two main limitations. First it assumes that the $\mathcal{C}_1(j, j')$ have already been computed, and does not take into account the resulting computational cost. Second, it stores all $\mathcal{C}_1(j, j')$ in a $\mathcal{O}(J^2)$ matrix, which is memory expensive.

A quick inspection of the algorithm reveals that the main step at Line 3 requires $O(j)$ operations (assuming the $\mathcal{C}_1(j, j')$s have been already computed). Therefore with the two **for** loops we get a complexity of $\mathcal{O}(SJ^2)$ in time. Note that without any particular assumption on the kernel $k(\cdot, \cdot)$, computing $\|Y - \hat{\mu}_{m_S}\|^2_{\mathcal{H}^J}$ for a given segmentation m_S is already of order $\mathcal{O}(J^2)$ in time since it involves summing over a quadratic number of terms of the Gram matrix (see Eq. (4.10)). Therefore there is no hope to solve (4.9) exactly in less than quadratic time without additional assumptions on the kernel.

From Eq. (4.10) let us also remark that computing each $\mathcal{C}_1(j, j')$ $(1 \leq j' < j \leq n)$ naively requires itself a quadratic number of operations, hence a $\mathcal{O}(J^4)$ time complexity for computing the whole cost matrix. Then the dynamic programming step (Line 3 of Algorithm 1) is not the limiting factor in that case and the overall time complexity of Agorithm 1 is $\mathcal{O}(SJ^4)$.

Finally let us also emphasize that this high computational burden is not specific of detecting change-points with kernels. It is rather representative of most learning procedures based on positive semi-definite kernels and the associated Gram matrix [Bach, 2013].

4.3.1 Improved use of dynamic programming for kernel methods

Reducing space complexity From Algorithm 1, let us first remark that each $\mathcal{C}_1(j,j')$ is used several times along the algorithm.

A simple idea to avoid that is to swap the two **for** loops in Algorithm 1. This leads to the following modified Algorithm 3, where each column $\mathcal{C}_1(\cdot, j'+1)$ of the cost matrix is only used once unlike Algorithm 1.

```
for $j' = 2$ to $J$ do
    for $s = 2$ to $\min(j', S)$ do
        $\mathbf{L}_{s,j'+1} = \min_{j \leq j'} \{ \; \mathcal{L}_{s-1,j} + \mathcal{C}_1(j, j'+1) \; \}$;
```

Algorithm 3: Improved space complexity

Importantly, this swap does not change the output of the algorithm and does not induce any additional calculations. In addition, at step j' of the first **for** loop we do not need the whole $J \times J$ cost matrix to be stored, but only the column $\mathcal{C}_1(\cdot, j'+1)$ of the cost matrix. This column is of size at most $\mathcal{O}(J)$. Storing only this column leads to a much improved $\mathcal{O}(SJ)$ space complexity.

Algorithm 3 finally requires to store coefficients $\{\mathcal{L}_{s,j}\}_{1 \leq s \leq S,\ 2 \leq j \leq J}$ that are computed along the algorithm as well as successive column vectors $\{\mathcal{C}_1(\cdot, j)\}_{2 \leq j \leq J}$ (of size at most J) of the cost matrix. This leads to an overall complexity of $\mathcal{O}(SJ)$ in space. The only remaining problem is to compute these successive column vectors efficiently. Let us recall that a naive implementation is prohibitive: each coefficient of the column vector can be computed in $\mathcal{O}(J^2)$, which would lead to $\mathcal{O}(J^3)$ to get the whole column.

Iterative computation of the columns of the cost matrix The last trick of our final algorithm is the efficient computation of each column vector $\{\mathcal{C}_1(\cdot, j)\}_{2 \leq j \leq J}$. We explain how to iteratively compute each vector in linear time.

First at all, it can be easily observed that Eq. (4.10) can be written as follows

$$\{\mathcal{C}_1(\cdot, j)\}_{2 \leq j \leq J} = \sum_{i=j}^{j'-1} \left(k\left(\mathrm{X}_i, \mathrm{X}_i\right) - \frac{A_{i,j'}}{j'-j} \right) = D_{j,j'} - \frac{1}{j'-j} \sum_{i=j}^{j'-1} A_{i,j'},$$

where $D_{j,j'} = \sum_{i=j}^{j'-1} k\left(\mathrm{X}_i, \mathrm{X}_i\right)$ and

$$A_{i,j'} = -k(\mathrm{X}_i, \mathrm{X}_i) + 2 \sum_{j=i}^{j'-1} k(\mathrm{X}_i, \mathrm{X}_j).$$

Secondly, both $D_{j,j'}$ and $\left\{A_{i,j'}\right\}_{i\leq j'}$ can be iteratively computed from j' to $j'+1$ by use of the two following equations:

$$D_{j,j'+1} = D_{j,j'} + k(\mathrm{X}_{j'}, \mathrm{X}_{j'}), \qquad \text{and} \qquad A_{i,j'+1} = A_{i,j'} + 2k(\mathrm{X}_{j'}, \mathrm{X}_{j'}),\ \forall i \leq j',$$

with $A_{j'+1,j'+1} = -k(\mathrm{X}_{j'}+1, \mathrm{X}_{j'}+1)$. Therefore as long as computing $k(x_i, x_j)$ is in $\mathcal{O}(1)$, updating from j' to $j'+1$ requires $\mathcal{O}\left(j'\right)$ operations. Note that for many classical kernels, computing $k(x_i, x_j)$ is indeed in $\mathcal{O}(1)$. For example if $x_i \in \mathbb{R}^q$ with q a constant larger or equal to 1 and $k(\cdot,\cdot)$ denotes the Gaussian kernel, each evaluation of $k(x_i, x_j)$ has a $\mathcal{O}(q) = \mathcal{O}(1)$ time complexity. If q is not negligible, the last example illustrates that the resulting time complexity is only increased by a multiplicative factor.

This update rule leads us to the following Algorithm 4, where each column

$$\mathcal{C}_1(\cdot, j'+1)$$

in the first **for** loop is computed only once:

```
for $j' = 2$ to $S$ do
    Compute the $(j'+1)$-th column $\mathcal{C}_1(\cdot, j'+1)$ from $\mathcal{C}_1(\cdot, j')$;
    for $s = 2$ to $\min(j', S)$ do
        $\mathbf{L}_{s,j'+1} = \min_{j\leq j'}\{\mathbf{L}_{s-1,j} + \mathcal{C}_1(j, j'+1)\}$;
```

Algorithm 4: Improved space and time complexity

From a computational point of view, each step of the first **for** loop in Algorithm 4 requires $\mathcal{O}(j')$ operations to compute $\mathcal{C}_1(\cdot, j'+1)$ and at most $\mathcal{O}(Sj')$ additional operations to perform the dynamic programming step at Line 4. Then the overall complexity is $\mathcal{O}(SJ^2)$ in time and $\mathcal{O}(SJ)$ in space. This should be compared to the $\mathcal{O}(SJ^4)$ time complexity of the naive calculation of the cost matrix and to the $\mathcal{O}(J^2)$ space complexity of the standard Algorithm 1 from [Harchaoui and Cappé, 2007].

4.3.2 Heuristic

In the previous section 4.3.1, we described an improved algorithm based on carefully combining dynamic programming and the computation of the cost matrix elements. This new algorithm (Algorithm 4) provides the exact solution to the optimization problem given by Eq. (4.9). However without any further assumption on the underlying kernel, this algorithm only achieves the complexity $\mathcal{O}(J^2)$ in time, which is a clear limitation for signals larger than 10^6. Note also that this limitation results from the use of positive semi-definite kernels (and related Gram matrices) and cannot be improved by existing algorithms to the best of our knowledge. However, we can remark that for some kernels it is possible to reduce this time complexity. For instance, for the linear one $k(x, y) = \langle x, y\rangle_{\mathbb{R}^s}\,,\ x, y \in \mathbb{R}^s$, one can use the following trick

$$\sum_{1\leq i\neq j\leq J} k(\mathrm{X}_i, \mathrm{X}_j) = \sum_{1\leq i\neq j\leq J} \langle \mathrm{X}_i, \mathrm{X}_j\rangle_{\mathbb{R}^S} = \sum_{1\leq i\leq J} \left\langle \mathrm{X}_i, \sum_{j=1}^{J} \mathrm{X}_j - \mathrm{X}_i \right\rangle \tag{4.11}$$
$$= \left\| \sum_{i=1}^{J} \mathrm{X}_i \right\|^2 - \sum_{i=1}^{J} \|\mathrm{X}_i\|^2 \,,$$

where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^s$.

The aim of this section is to describe a general strategy (that is applying to any kernel) relying on a low-rank approximation to the Gram matrix [Williams and Seeger, 2001, Smola and Schölkopf, 2000, Fine et al., 2001]. This approximation allows to considerably reduce the computation time by exploiting (4.11). Note however that the resulting procedure achieves this lower time complexity at the price of only providing an approximation to the exact solution to (4.9) (unlike the algorithm described in the previous section 4.3.1).

Low-rank approximation to the Gram matrix

The key idea is to follow the same strategy as the one described by [Drineas and Mahoney, 2005] to derive a low-rank approximation to the Gram matrix $\mathbf{K} = \{\mathbf{K}_{i,j}\}_{1\leq i,j\leq J}$, where $\mathbf{K}_{i,j} = k(\mathrm{X}_i, \mathrm{X}_j)$.

Assuming $\mathbf{K}$ has rank $\mathrm{rk}(\mathbf{K}) \ll J$, we could be tempted to compute the best rank approximation to $\mathbf{K}$ by computing the $\mathrm{rk}(\mathbf{K})$ largest eigenvalues (and corresponding eigenvectors) of $\mathbf{K}$. However such computations induce a $\mathcal{O}(J^3)$ time complexity which is prohibitive.

Instead, [Drineas and Mahoney, 2005] suggests applying this idea on a square submatrix of $\mathbf{K}$ with size $p \ll J$. For any subsets $\mathcal{I}, \mathcal{I}' \subset \{1, \ldots, J\}$, let $\mathbf{K}_{\mathcal{I},\mathcal{I}'}$ denote the sub-Gram matrix with respectively row and column indices in $\mathcal{I}$ and $\mathcal{I}'$. Let $\mathcal{I}_p \subset \{1, \ldots, J\}$ denote such a subset with cardinality p, and consider the sub-Gram matrix $\mathbf{K}_{\mathcal{I}_p,\mathcal{I}_p}$ which is of rank $r \leq p$. Further assuming $r = p$, the best rank p approximation to $\mathbf{K}_{\mathcal{I}_p,\mathcal{I}_p}$ is $\mathbf{K}_{\mathcal{I}_p,\mathcal{I}_p}$ itself. This leads to the final approximation to the Gram Matrix $\mathbf{K}$ [Drineas and Mahoney, 2005, Bach, 2013] by

$$\widetilde{\mathbf{K}} = \mathbf{K}_{\mathcal{I}_J,\mathcal{I}_p}\ \mathbf{K}^{+}_{\mathcal{I}_p,\mathcal{I}_p}\ \mathbf{K}_{\mathcal{I}_p,\mathcal{I}_J}\,,$$

where $\mathcal{I}_J = \{1, \ldots, J\}$, and $\mathbf{K}^{+}_{\mathcal{I}_p,\mathcal{I}_p}$ denotes the pseudo-inverse of $\mathbf{K}_{\mathcal{I}_p,\mathcal{I}_p}$. Further considering the SVD decomposition of $\mathbf{K}_{J_p,J_p} = \mathbf{U}'\Lambda\mathbf{U}$, for an orthonormal matrix $\mathbf{U}$, we can rewrite

$$\widetilde{\mathbf{K}} = \mathbf{Z}'\mathbf{Z}, \qquad \text{with} \quad \mathbf{Z} = \Lambda^{-1/2}\mathbf{U}\ \mathbf{K}_{J_p,I_n} \in \mathcal{M}_{p,J}(\mathbb{R}).$$

Note that the resulting time complexity is $\mathcal{O}(p^2J)$, which is smaller than the former $\mathcal{O}(J^3)$ as long as $p = o(\sqrt{J})$. This way, columns $\{Z_i\}_{1\leq i\leq J}$ of $\mathbf{Z}$ act as new p-dimensional observations and each $\widetilde{\mathbf{K}}_{i,j}$ can be seen as the inner product between two vectors of $\mathbb{R}^p$, that is

$$\widetilde{\mathbf{K}}_{i,j} = Z_i'Z_j. \tag{4.12}$$

The main interest of this approximation is that, using Eq. (4.11), computing the cost of a segment s of length n_s has a complexity $\mathcal{O}(n_s)$ in time unlike the usual $\mathcal{O}(n_s^2)$ that holds with general kernels.

Note that choosing the set $\mathcal{I}_p$ of columns/rows leading to the approximation $\widetilde{\mathbf{K}}$ is of great interest in itself for at least two reasons. First from a computational point of view, the p columns have to be selected following a process that does not require to compute the J possible columns beforehand (which would induce an $O(J^2)$ time complexity otherwise). Second, the quality of $\widetilde{\mathbf{K}}$ to approximate $\mathbf{K}$ crucially depends on the rank of $\widetilde{\mathbf{K}}$ that has to be as close as possible to that of $\mathbf{K}$, which remains unknown for computational reasons. However such questions are out of scope of the present paper and we refer interested readers to [Williams and Seeger, 2001, Drineas and Mahoney, 2005, Bach, 2013] where this point has been extensively discussed.

Binary segmentation heuristic

Since the low-rank approximation of the Gram matrix detailed in Section 4.3.2 leads to deal with finite dimensional vectors in $\mathbb{R}^p$ (4.12), the change-point detection problem described in Section 4.2 amounts recovering abrupt changes of the mean of a p-dimensional time-series. Therefore any existing algorithm usually used to solve this problem in the p-dimensional framework can be applied. A review of such algorithms is presented in Chapter 2. However we will mention only a few of them to highlight their drawbacks and motivate our choice. Let us also recall that our purpose is to provide an efficient algorithm allowing: (i) to (approximately) solve Eq. (4.9) for each $1 \leq s \leq S$, and (ii) to deal with large sample sizes ($J > 10^6$).

The first algorithm is the usual version of constrained dynamic programming [Auger and Lawrence, 1989]. As presented in Chapter 2, a second possible algorithm is a standard heuristic (*binary segmentation*) for approximately solving Eq. (4.9) for each $1 \leq s \leq S$. This iterative algorithm computes the new segmentation m_{S+1} with $S+1$ segments from m_S by splitting one segment of m_S into two new ones without modifying other segments. More precisely considering the set of change-points $m_S = \{t_1, \ldots, t_S\}$, binary segmentation provides

$$m_{S+1} = \underset{m \in \mathcal{M}(S+1) | t \cap m_S = m_S}{\arg\min} \left\{ \|\mathrm{Y} - \hat{\mu}_m\|^2_{\mathcal{H}^J} \right\}.$$

The overall time complexity of binary segmentation for recovering approximate solutions to (4.9) for all $1 \leq s \leq S$ is only $\mathcal{O}\left(\log(S)J\right)$ on average and $\mathcal{O}\left(SJ\right)$ at worse (Section 2.2.4).

An important remark is that binary segmentation only achieves this reduced time complexity provided computing the cost of one segment has a complexity linear in its length. This is precisely what has been allowed by the low-rank matrix approximation summarized by Eq. (4.12). Otherwise with a quadratic complexity for computing the cost of one segment, binary segmentation would suffer an overall time complexity of order $\mathcal{O}(SJ^2)$.

4.4 Combination of kernels

As we said in the previous chapter, the performance of segmentation method is better if we use simultaneously the TCN and the DoH. In the case of kernel segmentations, we would like to segment jointly the TCN and the BAF signals. Then, a first nice property of kernels is that linear combinations of kernels are kernels. By consequences, it is easy to build a convex combination of the best kernel to segment TCN and the best one of the BAF. Indeed, $\forall x = (x_1, x_2) \in \mathcal{X}_1 \times \mathcal{X}_2$ and $y = (y_1, y_2) \in \mathcal{X}_1 \times \mathcal{X}_2$

$$k(x, y) = \alpha k_1(x_1, y_1) + (1 - \alpha) k_2(x_2, y_2)$$

However, an important issue will be to calibrate α and the respective parameters of kernels k_1 and k_2 for the each dimension. All algorithms presented in the previous section can be applied on this two-dimensional kernel.

4.5 Model selection

This section recalls notions presented in 2.4 expanded to the kernel framework. To find the best number of segments we can use ideal criteria if we work on simulated profiles (i.e. the true segments and the distribution of the signal are known) or otherwise a penalized criteria defined in [Birgé and Massart, 2001]. Ideal criteria is the minimization of the quadratic risk defined as follow :

$$\hat{S} = \underset{s=1,\dots,S}{\arg\min} \frac{1}{n} \|\mu^\star - \hat{\mu}_{\hat{m}_s}\|^2_{\mathcal{H}^J}$$

And the penalized criteria is the minimization of the empiral risk with a penalty which depends of number of segments d.

$$\hat{S} = \underset{s=1,\dots,S}{\arg\min} \frac{1}{n} \|\mathrm{Y} - \hat{\mu}_{\hat{m}_s}\|^2_{\mathcal{H}^J} + pen(s)$$

4.5.1 Ideal criterion

We define the quadratic risk as :

$$\begin{aligned} L(\hat{\mu}_{\hat{m}_S}) &= \|\mu^\star - \hat{\mu}_{\hat{m}_S}\|^2_{\mathcal{H}^J} \\ &= \|\mu^\star\|^2_{\mathcal{H}^J} + \|\hat{\mu}_{\hat{m}_S}\|^2_{\mathcal{H}^J} - 2\langle \mu^\star, \hat{\mu}_{\hat{m}_S}\rangle_{\mathcal{H}^J} \end{aligned}$$

We start by computation of the norm in $\mathcal{H}^J$ of the mean element $\mu^\star$. Then, we have to pay attention that Y_j is an element of $\mathcal{H}$. Therefore, there is no reason that the expectation is on Y_j and we need to take a copy of Y_j denote Y'_j and which follow the same distribution that Y_j. To estimate the double expectation, we need to have several random variables X_j^b which have the same distribution X_j. Then, we estimate the expectation by mean on the $B = 100$ random variables X_j^b.

$$\begin{aligned} \|\mu^\star\|^2_{\mathcal{H}^J} &= \sum_{j=1}^J \|\mu_j^\star\|^2_{\mathcal{H}} = \sum_{j=1}^J \langle \mu_j^\star, \mu_j^\star\rangle_{\mathcal{H}} = \sum_{j=1}^J \mathbb{E}_{\mathrm{Y}_j} \langle \mathrm{Y}_j, \mu_j^\star\rangle_{\mathcal{H}} \\ &= \sum_{j=1}^J \mathbb{E}_{\mathrm{Y}_j,\mathrm{Y}'_j} \langle \mathrm{Y}_j, \mathrm{Y}'_j\rangle_{\mathcal{H}} = \sum_{j=1}^J \mathbb{E}_{X_j,X'_j} k(X_j, X'_j) \\ &\approx \frac{1}{B(B-1)} \sum_{b\neq b'} \sum_{j=1}^J k(X_j^b, X_j^{b'}) \end{aligned}$$

Then, we compute the norm in $\mathcal{H}^J$ of the estimation of $\mu^\star$ denote $\hat{\mu}_{\hat{m}_S}$.

$$
\begin{aligned}
\|\hat{\mu}_{\hat{m}_S}\|^2_{\mathcal{H}^J} &= \sum_{j=1}^{J} \|\hat{\mu}_{\hat{m}_S}(j)\|^2_{\mathcal{H}} = \sum_{j=1}^{J} \langle \hat{\mu}_{\hat{m}_S}(j), \hat{\mu}_{\hat{m}_S}(j)\rangle_{\mathcal{H}} \\
&= \sum_{j=1}^{J} \langle \frac{1}{n_{s(j)}} \sum_{j \in s(j)} \mathrm{Y}_j, \frac{1}{n_{s(j)}} \sum_{l \in s(j)} \mathrm{Y}_l \rangle_{\mathcal{H}} \\
&= \sum_{j=1}^{J} \frac{1}{n^2_{s(j)}} \sum_{j \in s(j)} \sum_{l \in s(j)} \langle \mathrm{Y}_j, \mathrm{Y}_l \rangle_{\mathcal{H}} = \sum_{s \in \hat{m}_S} \frac{n_s}{n_s^2} \sum_{j,l \in s} \langle \mathrm{Y}_j, \mathrm{Y}_l \rangle_{\mathcal{H}} \\
&= \sum_{s \in \hat{m}_S} \frac{1}{n_s} \sum_{j,l \in s} k(\mathrm{X}_j, \mathrm{X}_l)
\end{aligned}
$$

To finish, the scalar product between $\mu^\star$ and $\hat{\mu}_{\hat{m}_S}$. We need one copy of the distribution of X_j for the same reasons than previously since in the sum on l and j, $l = j$. We estimate expectation by mean on the $B = 100$ random variables X_j^b which have the same distribution than X_j.

$$
\begin{aligned}
\left\langle \mu^\star, \hat{\mu}_{\hat{m}_S} \right\rangle_{\mathcal{H}^J} &= \sum_{j=1}^{J} \left\langle \mu_j^\star, \hat{\mu}_{\hat{m}_S}(j) \right\rangle_{\mathcal{H}} = \sum_{l=1}^{J} \frac{1}{n_{s(l)}} \sum_{j \in s(l)} \left\langle \mu_l^\star, \mathrm{Y}_j \right\rangle_{\mathcal{H}} \\
&= \sum_{l=1}^{J} \frac{1}{n_{s(l)}} \sum_{j \in s(l)} \mathbb{E}_{\mathrm{Y}'_{\mathrm{l}}} \left\langle \mathrm{Y}'_l, \mathrm{Y}_j \right\rangle_{\mathcal{H}} \\
&= \sum_{s \in \hat{m}_S} \frac{1}{n_s} \sum_{l,j \in s} \mathbb{E}_{\mathrm{Y}'_{\mathrm{l}}} \left\langle \mathrm{Y}'_l, \mathrm{Y}_j \right\rangle_{\mathcal{H}} \\
&\approx \frac{1}{B} \sum_{b \in \mathcal{B}} \sum_{s \in \hat{m}_S} \frac{1}{n_s} \sum_{l,j \in s} k(\mathrm{X}_l^b, \mathrm{X}_j)
\end{aligned}
$$

Then, we get three terms to compute the quadratic risk :

$$
\|\mu^\star\|^2_{\mathcal{H}^J} \approx \frac{1}{B(B-1)} \sum_{b \neq b'} \sum_{j=1}^{J} k(\mathrm{X}_j^b, \mathrm{X}_j^{b'}) \tag{4.13}
$$

$$
\|\hat{\mu}_{\hat{m}_S}\|^2_{\mathcal{H}^J} = \sum_{s \in \hat{m}_S} \frac{1}{n_s} \sum_{j,l \in s} k(\mathrm{X}_j, \mathrm{X}_l) \tag{4.14}
$$

$$
\langle \mu, \hat{\mu}_{\hat{m}_S} \rangle_{\mathcal{H}^J} \approx \frac{1}{B} \sum_{b \in \mathcal{B}} \sum_{s \in \hat{m}_S} \frac{1}{n_s} \sum_{l,j \in s} k(\mathrm{X}_l^b, \mathrm{X}_j) \tag{4.15}
$$

4.5.2 A penalized criterion

However, for real data, it is impossible to compute the ideal criterion (quadratic risk) in order to select the right number of segment. As for RBS and Group Fused Lasso method, we use the penalized empirical risk (see section 2.4 and equation 4.17).

$$pen(s) = \frac{Cs}{J}\left(c_1 \log\left(\frac{J}{s}\right) + c_2\right). \tag{4.16}$$

For the moment, we use the values $c_1 = 2$ and $c_2 = 5$ as in the initial paper of [Lebarbier, 2005].

To summarize:

$$S^* = \arg\min_{s \in 1:S} \|\mathrm{Y} - \hat{\mu}_{\hat{m}_s}\|^2_{\mathcal{H}^J} + 2 * pen(s). \tag{4.17}$$

where $\hat{m}_s$ is the best segmentation in s segments.

4.5.3 Criterion to compare segmentation

In a first intention, we wanted to know which is the parameter value δ that provides the best segmentations. The first idea was to compare directly the quadratic risk in the RKHS provided by the algorithms as it was done for the model selection. However, a brief review of literature leads us to conclude that norms from different RKHS are in fact not comparable. This is the subject of the following section. From [Zhang and Zhao, 2011] and lemmas in Appendix C, we can deduce inclusions between two RKHS from two different Gaussian kernels. Therefore, the norms from two different Gaussian kernels are not equivalent and therefore the respective quadratic and empirical risks are not comparable. This section informs us that it is impossible to remain in the RKHS space to select the best parameter. In order to compare segmentations provided by two kernels when the number of segments is fixed, we have to go back to the initial space of observation. We define another performance criterion to measure the quality of segmentations for each value of the parameter.

The quality of segmentations is measured by the gap between the segmentation $\hat{m}$ and the true segmentation m^* by using the Frobenius distance as follows. First, for any segmentation $m = (t_1, t_2, \ldots, t_S, t_{S+1})$, let us introduce a matrix $M = \{M_{i,j}\}_{1 \leq i,j \leq J}$ such that

$$M_{i,j} = \sum_{k=1}^{D+1} \frac{\mathbb{1}_{(t_{k-1} \leq i,j < t_k)}}{t_k - t_{k-1}},$$

where $\mathbb{1}_{(t_{k-1} \leq i,j < t_k)} = 1$, if $i, j \in [t_{k-1}, t_k[\cap \mathbb{N}$, and 0 otherwise. Let us now consider the matrix M^* defined from the true segmentation m^* in the same way. Then, the Frobenius distance between segmentations $\hat{m}$ and m^* is given, through the distance between matrices M and M^*, by

$$d_F^2 (m, m^*) = \|M - M^*\|_F^2 = \sum_{i,j=1}^{J} \left(M_{i,j} - M^*_{i,j}\right)^2 .$$

4.6 Results on the realistic simulated framework

This section is divided into three parts. The first one presents the results to calibrate δ of the gaussian kernel in order to get the best segmentation. Then, we present performance of the model selection directly in the RKHS space. To finish, the results of the global performance of the procedure are presented in a simulation framework. Note that all simulations are performed with `jointseg` and `acnr` packages on the realistic framework.

4.6.1 Selection of parameter

In this section, we present the results of the influence of δ on the segmentation. For this part, we perform segmentation for several short profiles and we compare the performance of the segmentation with the true number of segments (section 4.5.3).

In an easy framework (100% of tumor cells), it is obvious that the value of the δ parameter to segment TCN is not crucial. Contrary to the more difficult frameworks (with more contamination by normal cells) when δ increases the quality of segmentations is deteriorated. For both frameworks (79% and 50%), the value which provides the best segmentation quality is around 1. For this reason, we fixed $\delta = 1$ for the extensive experiments that we performed in the following.

The grid of values for the BAF segmentation is completely different, this probably due to the fact that BAF is always between 0 and 1. However, we observe the same shapes of curves, i.e when the value of parameter increases the quality of segmentations decreases. The same phenomenon is observed when the parameter is too small. In this case, it appears that the best parameter is around 0.005 for the three settings of simulations. This value corresponds in fact to the optimal bandwidth selected by cross-validation in order to estimate density with a gaussian kernel.

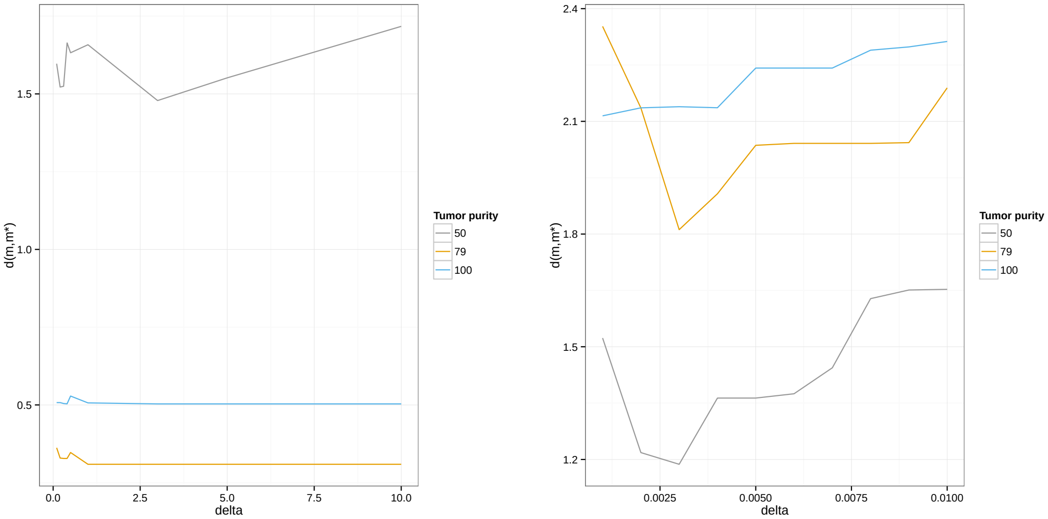


Figure 4.3 – Influence of δ on three levels of difficulty (TCN on the left, BAF on the right)

4.6.2 Selection of parameter p to approximate kernel

In this section, we present the results of the influence of the parameter p on the segmentation in terms of partial AUCs. We use the same method as in section 3.3.3 to determine true positive and false positive. Results are presented for the TCN segmentation (Illumina technology) and for a tolerance parameter equal to 10. Initial points to approximate the Gram matrix are taken uniformly between the minimum and the maximum point of the TCN signal. The grid of p is between 1 to 100 initial points. We simulated TCN 50 profiles with 20000 points and we evaluate the performance for each p through the pAUC. On Fig. 4.4, we have the mean of pAUC on the 50 profiles for three dilution series (100%, 79% and 50% of tumor cells). It appears that taking $p = 10$ seems to be a good compromise between the performance of segmentation and the performance in terms of times for the three levels of difficulty.

4.6.3 Model selection

It is not obvious that usual model selection criteria perform correctly for model selection in a RHKS. For this reason, we implemented the cost function in the RKHS as described in the section 4.5.1 and the penalized cost function 4.5.2 and we compare their ability to select the best number of segments on simulations. Indeed, on simulations, we know the truth i.e the position of breakpoints required to compute the quadratic risk. We simulate profiles that contain 3 breakpoints i.e 4 segments and we segmented them

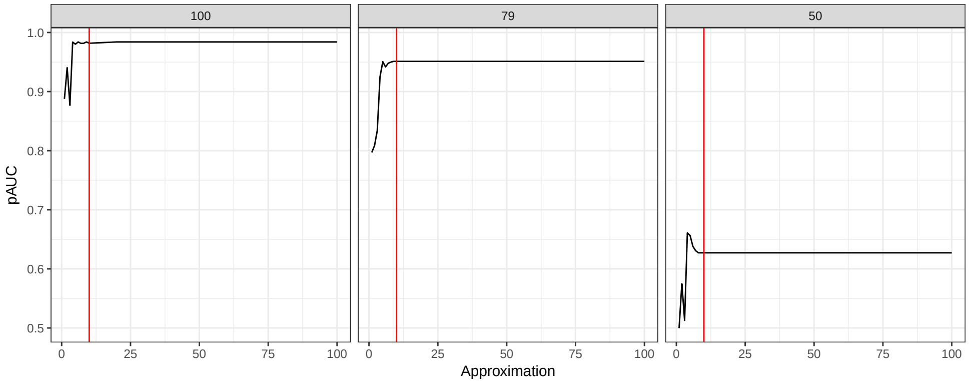


Figure 4.4 – Experiments to choose the level of approximation (p) (Illumina example)

and selected the correct number of segments with the penalized criterion. For each annotated data described in the previous chapter, we apply the segmentation algorithm on several profiles with various levels of contamination by normal cells. The results are presented in Fig. 4.5. It is clear that the method of penalized criteria directly applied in the RKHS perform as well as for the three different levels of difficulty (100, 70 and 50% of tumor cells in the samples) to select the right number of segments either for Affymetrix or Illumina.

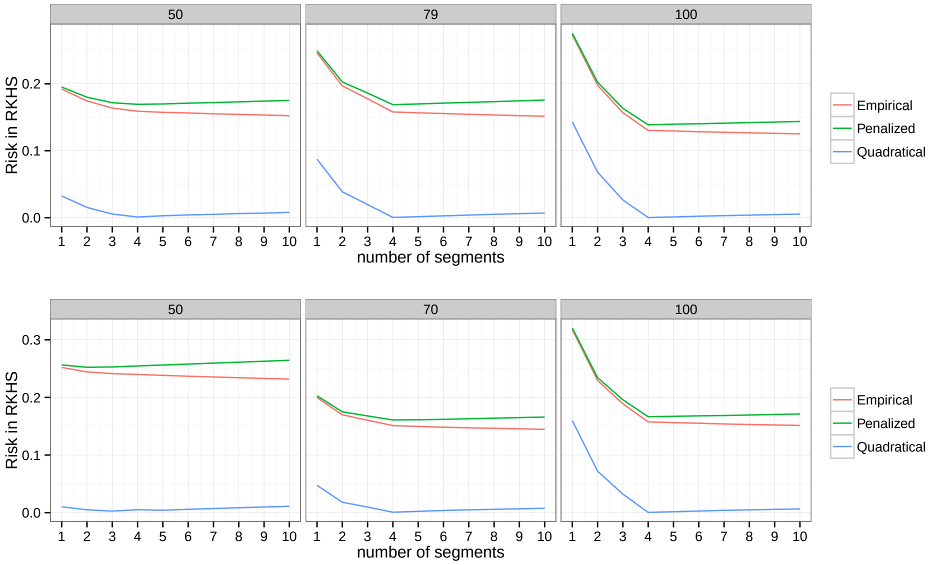


Figure 4.5 – Model selection illustration on three levels of difficulty (Illumina on the top and Affymetrix on the bottom)

4.6.4 Performance of segmentations

We expand the simulation framework by simulating 50 DNA copy number profiles of length 20000 containing 10 breakpoints following the method proposed in section 3.2.2 were simulated to measure the performance of the kernel procedure. The breakpoints are random as well as the status of altered regions. Profiles are simulated with 100%, 79% and 50% of tumor cells (Illumina data set).

As we said in the previous chapter, the best results are provided by RBS method, therefore we only compare the kernel method to this method . We plot the results in terms of partial AUCs of exact kernel segmentation, the heuristic of kernel segmentation and recursive binary segmentation. For each type of methods, we apply it on BAF/ (respectively for kernel and RBS method), TCN and jointly on DoH/BAF and TCN. In order to compute partial AUC, we use the same method as in section 3.3.3 to determine true positive and false positive.

The results of performance across the three values of purity are presented in Fig. 4.6. Each curve is the pAUC curve along the tolerance parameter as described in section 3.3.3. The figure shows that using kernel segmentation allows detecting some changes that probably not occur in the mean but in higher order in the distribution. Indeed, pAUC is notably higher if we use the kernel segmentation procedure. For all settings (100%, 79% and 50% of tumor cells), exact kernel segmentation using both of TCN and BAF outperforms all methods. It is clear that for easy the framework, using kernel improves performance by recovering more precisely the change points. For the two others, RBS using simultaneously TCN and DoH does as well as kernel method. In addition, RBS is really faster than Kernel segmentation due to a lower complexity of computation.

After exploring results, we realized that the symmetrization of BAF signal provided best results when we used the approximation of the algorithm. For this reason, the results of the performance of heuristic kernel algorithm on BAF is, in fact, the results of the performance on the DoH which includes all SNPs (homozygous and heterozygous). Nevertheless, even with this transformation, all other methods outperform the heuristic kernel algorithm on BAF. This is likely the reason why kernel approximation algorithm on the two dimensions is not as good as the exact one.

However, an essential point is that it is not necessary to know the germinal status of SNPs to perform the segmentation in the case of kernels and to transform BAF into DoH in the case of exact algorithm unlike to RBS that uses DoH on heterozygous SNPs. To summarize, kernel segmentation procedure can provide good results in term of segmentations with no data transformation (TCN to log(TCN) and BAF to DoH) but is clearly slower than RBS.

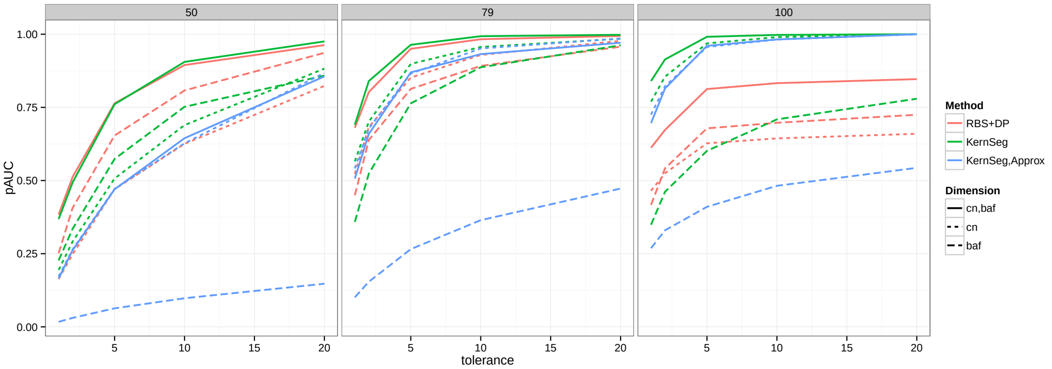


Figure 4.6 – Partial AUC for Kernel segmentation

4.7 Conclusion

To conclude, in this chapter, we presented a new non-parametric segmentation method in order to be free from the transformation of the BAF signal to DoH. This method has the particularity to be independent of the type of breakpoints of the signals. Indeed, we are able to detect change-points either in the mean or in the higher orders of the distribution if the kernel is well chosen. By using the kernel trick, it has been possible to combine TCN and BAF signals to perform a joint segmentation method. Performance is at least as good as one of the joint RBS method and in addition, we produce efficient algorithms in terms of complexity and quality of segmentations. We show that heuristic algorithm performs as well as the exact one in some frameworks. But some other experiments are required to tune the δ parameters and the approximation level.

The joint kernel segmentation takes the advantage that it is not necessary to know the germinal status of SNPs before doing the segmentation. This issue of to not knowing the germinal status is quite common and raises some new problems to perform segmentation and estimations of parental copy numbers for example. This topic is tackled in Chapter 8.

To conclude the proposed method provides a usable procedure to segment the DNA copy number signals from microarrays or even NGS. In addition, after a good selection of δ, we show that classical the model selection directly applied in the RKHS space provided good results. An R package is currently in development and will be soon available.

Part II

Tumor clonality inference using copy number data

Table of Contents

Chapter 5

Discovering heterogeneity in cancers

This Chapter aims to present some existing models to discover heterogeneity in cancers.

5.1 Introduction

Even if recovering of breakpoints in TCN and BAF signals with accuracy is still a challenge, determining the status of each segment/region is another essential point. We have seen that HMM methods (section 2.2.1) perform jointly the segmentation and the region status inference. Here, we do not develop this kind of methods but we make the choice to perform the calling separately from the segmentation.

The first reason of the difficulty to attribute an alteration type (Table 2.1) to each segment is the proportion of normal cells in the samples that shrinks the total copy number toward 2, but also the cellularity. Intra-tumoral heterogeneity of samples can also disturb the inference of the calling of region status. Single cell methods enable to assess tumor heterogeneity without the effects of mixed cell population [Navin et al., 2011] but in reality, the single-cell sequencing data is noisier than the sequencing directly performed on solid tumors. This method is expensive and most laboratories still sequence the DNA at the level of bulk tumors. Therefore, instead of observing only one type of cell, we observe a mixture of normal/stromal cells, and one or several types of cancer cells. The problem of inferring tumor subclones from microarray or sequencing genomic data has received a lot of attraction in the past few years [Beerenwinkel et al., 2014]. The global aim of these methods is to reconstruct the observed profiles by inferring subclones and weights. We present in the following how to model the linear combination to explain total DNA copy number profiles.

5.2 Matrix factorization problem

In this section, we work with a $n \times J$ matrix $\mathbf{Y}$ that is composed of n DNA copy number profiles of length J. Therefore, let y_{ij} denote the total copy number signal at locus $j \in \{1, \ldots, J\}$ for sample $i \in \{1, \ldots, n\}$. This model states that for each sample i, the corresponding copy-number profile $\mathbf{Y}_i = (y_{i1}, \ldots, y_{iJ})$ can be expressed as a weighted linear combination of p latent profiles $\mathbf{Z}_k$ plus some noise, that is,

$$\mathbf{Y}_i \quad = \quad \sum_{k=1}^{p} w_{ik}\mathbf{Z}_k + \mathbf{E}_i \,, \text{ for } i = 1, \ldots, n \tag{5.1}$$

where $\mathbf{Z}_k = (z_{k1}, \ldots, z_{kJ})$ is the copy-number profile of the k-th latent profile for $k = 1, \ldots, p$, w_{ik} is the weight of latent profile k in sample i, and $\mathbf{E}_i = (\epsilon_{i1}, \ldots, \epsilon_{iJ})$ is the vector of reconstruction errors for sample i. Both matrices $\mathbf{W}$ and $\mathbf{Z}$ are unknown and estimated by minimizing the total squared errors 5.2:

$$\sum_{i=1}^{n} \|\mathbf{Y}_i - \sum_{k=1}^{p} w_{ik}\mathbf{Z}_k\|_2^2 \tag{5.2}$$

The matrix form of the Eq. 5.2 is:

$$\|\mathbf{Y} - \mathbf{WZ}\|_{\mathrm{F}}^2 \tag{5.3}$$

where

- $\mathbf{Y}$ is the $n \times J$ matrix of copy-number signals for each sample,
- $\mathbf{W}$ is the $n \times p$ matrix of weights,
- $\mathbf{Z}$ is the $p \times J$ matrix of latent profiles,

and $\|.\|_{\mathrm{F}}$ the Frobenius Norm.

This is a standard method to reduce the dimensionality of a matrix and discover subgroups within patients with the same disease. This method permits to select only the most relevant information in meaningful components by adding some constraints.

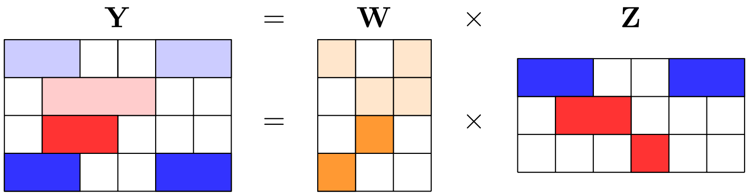


Figure 5.1 – Modelisation of heterogeneity

The figure 5.1 shows a toy illustration of the heterogeneity model. Here, the matrix **Y** which is observed is, in fact, the factorization of the two matrices **W** and **Z**. Patterns are identifiable in matrix **Z**. These patterns could be for instance loss (blue tiles) and gain (red tiles) in a DNA copy number latent profile. The matrix **W** represents, in fact, the composition of each sample that we observed in **Y**.

Generally, to solve a matrix factorization problem, it is necessary to add constraints on the dictionary defined by **Z**. For example in our context, for interpretability reasons, we would like that copy number latent profiles present only a few alterations, in other terms **Z** is required to be sparse (few components are non-zero). This kind of problem has been already solved and is named sparse dictionary learning [Olshausen and Field, 1997], we present the model in section 5.3.

Then, in the context of cancerology and tumoral heterogeneity discovery, the Non-negative matrix factorization (NMF) model is the most common model [Brunet et al., 2004]. Indeed, the first studies on cancer heterogeneity focused on gene expression and methylation data [Qi et al., 2009, Taslaman and Nilsson, 2012, Schwalbe et al., 2013], which are non-negative. Its aim is to factorize the matrix **Y** into two non-negative matrices **W** and **Z**, NMF model is presented in 5.4.

Finally, in the context of DNA copy number, the observed mixture of several latent profiles can be seen as if **W** represents the proportion of each copy number latent profile. [Cutler and Breiman, 1994] have proposed to constrain each row of **W** to have sum one and **W** to be non-negative. This method is called archetypal analysis in the literature. This type of constraints enables to give a biological signification to **W** (composition of the sample). We present this method in 5.5.

5.3 Sparse dictionary learning

The sparse dictionary learning is a matrix factorization method that aims at finding a sparse representation of the input data **Y** in the form of a linear combination of basis elements as well as those basis elements themselves. The input data set in our context is $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_n]^T, \mathbf{y}_i \in \mathbb{R}^J$. We wish to find a dictionary $\mathbf{Z} \in \mathbb{R}^{p \times J}$ and a representation $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_n]^T, \mathbf{w}_i \in \mathbb{R}^p$ such that both $\|\mathbf{Y} - \mathbf{WZ}\|^2$ is minimized and the representations $\mathbf{w}_i$ are sparse enough. This can be formulated as the following optimization problem:

$$min_{\mathbf{W} \in \mathbb{R}^{np}, \mathbf{Z} \in \mathcal{I}} \sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{w}_i \mathbf{Z}\|^2 + \lambda \|\mathbf{w}_i\|_0 \,, \tag{5.4}$$

where $\mathcal{I} \equiv \mathbf{Z} \in \mathbb{R}^{p \times J} : \|z_j\|_2 \leq 1 \; \forall j = 1, ..., J$

The constraint $\mathbf{Z} \in \mathcal{I}$ is required to prevent the atoms of $\mathbf{Z}$ to reach arbitrarily high values allowing for arbitrarily low (but non-zero) values of $\mathbf{w}_i$.

The minimization problem above is not convex because of the ℓ_0- pseudo-norm and solving it is NP-hard [Tillmann, 2015]. ℓ_1-norm is known to ensure sparsity and therefore the above problem 5.4 can be relaxed into a convex optimization problem with respect to each of the variables $\mathbf{Z}$ and $\mathbf{W}$ in case the other one of them is fixed.

The dictionary $\mathbf{Z}$ defined is “undercomplete“ if $p < n$ or ”overcomplete” in case $p > n$.

5.4 Non-negative matrix factorization

NMF uses the model defined by 5.3 but with some positivity constraints on the two matrices $\mathbf{W}$ and $\mathbf{Z}$. In other terms, $\mathbf{z}_k = (z_{k1}, \ldots, z_{kJ}) \succeq 0$ for $k = 1, \ldots, p$ and $w_{ik} \geq 0$ for $k = 1, \ldots, p$ and for $i = 1, \ldots, n$.

The equivalent matrix form is written below:

$$\left(\hat{\mathbf{W}}, \hat{\mathbf{Z}}\right) = \underset{\mathbf{W} \in \mathbb{R}^{np}, \mathbf{Z} \in \mathbb{R}^{pJ}}{\arg\min} \|\mathbf{Y} - \mathbf{WZ}\|^2 \quad \text{s.t.} \quad \mathbf{W} \succeq 0 \quad \text{and} \quad \mathbf{Z} \succeq 0 \tag{5.5}$$

Several optimized algorithms have been developed this last twenty years [Lee and Seung, 2001, Brunet et al., 2004, Kim and Park, 2007, Badea, 2008] to solve NMF on gene expression data from microarrays, and a R package implements several algorithms to solve NMF problems [Gaujoux and Seoighe, 2010]. The most common types of methods to solve NMF are: multiplicative update rules, alternating least square and alternating nonnegative least squares. We briefly present the update rules to solve NMF.

All algorithms begin by initializing the matrix $\mathbf{Z}$ by taking p rows from the matrix $\mathbf{Y}$, then the multiplicative updating rules are defined by the following equations:

$$w_{ik} \rightarrow w_{ik} \frac{(\mathbf{YZ}^T)_{ik}}{(\mathbf{WZZ}^T)_{ik}} \tag{5.6}$$

and

$$z_{kj} \rightarrow w_{ik} \frac{(\mathbf{W}^T\mathbf{Y})_{kj}}{(\mathbf{W}^T\mathbf{WZ})_{kj}} \tag{5.7}$$

Even if, this method was not used directly, we choose to present the NMF because it was the first method used in cancerology to infer heterogeneity jointly from several samples.

5.5 Archetypal analysis

The archetypal analysis uses the same constraint than NMF on matrix $\mathbf{W}$ i.e. to be positive, but in addition, the sum of each row of matrix $\mathbf{W}$ is forced to be equal to 1. This constraint is useful if we see the observed samples of the matrix $\mathbf{Y}$ as a convex combination of subclones. Then, the matrix $\mathbf{W}$ can be seen as a proportion matrix. The last constraint imposed by the archetypal analysis is that the components of matrix $\mathbf{Z}$ is also a convex combination of the elements of $\mathbf{Y}$.

More precisely, for fixed $\mathbf{z}_1, \ldots, \mathbf{z}_p$ we have:

$$\mathbf{z}_k = \sum_{i=1}^{n} \beta_{ki} \mathbf{y}_i, \qquad k = 1, \ldots p$$

with $\beta_{ki} \geq 0$ and $\sum_{i=1}^{n} \beta_{ki} = 1$.

Then, $\{w_{ik}\}$, $k = 1, \ldots, p$ are defined as the minimizers of

$$\|y_{i\bullet} - \sum_{k=1}^{p} w_{ik} z_{k\bullet}\|^2$$

under the constraints $w_{ik} \geq 0$ and $\sum_k w_{ik} = 1$.

The archetypal patterns are defined by the minimization of the Residual sum of squares (RSS) computed on the whole set of samples:

$$RSS \quad = \quad \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{k=1}^{p} w_{ik} z_k\|^2 \tag{5.8}$$

The RSS can also be written:

$$RSS = \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{k=1}^{p} w_{ik} \sum_{i'=1}^{n} \beta_{ki'} \mathbf{y}_{i'}\|^2 \tag{5.9}$$

The problem of archetypal analysis is to find $\mathbf{W}$ and β by minimizing the equation (5.9). The algorithm of archetypal analysis alternates between finding w_{ik} and β. It begins by initializing the matrix of latent profiles $\mathbf{Z}$ and coefficient w_{ik} and β. Then, solve the n convex least squares problems:

$$\min_{\mathbf{w}_i} \|\mathbf{y}_i - \mathbf{w}_i\mathbf{Z}\|_2^2 \text{ subject to } w_{ik} \geq 0 \text{ and } \sum_{k=1}^{p} w_{ik} = 1 \tag{5.10}$$

Then, update latent profiles $\tilde{\mathbf{Z}}$. And, solve the p convex least squares problems:

$$\min_{\beta_k} \|\mathbf{z}_k - \mathbf{w}_k\mathbf{Y}\|_2^2 \text{ subject to } \beta_k \geq 0 \text{ and } \sum_{i=1}^{n} \beta_{ki} = 1 \tag{5.11}$$

Then, update latent profiles $\tilde{\mathbf{Z}} = \beta\mathbf{Y}$.

The main drawback of this model is that latent profiles are a convex combination of the observed profiles and biologically it is a strong hypothesis. This means that subclones are directly observed, which is unlikely. In Chapter 6 were inspired by sparse dictionary learning for sparse constraints on the latent profiles and by the archetypal analysis for the constraints on the weight matrix.

5.6 A few words on identifiability

A major issue for this type of reconstruction is that even ignoring the noise in the input profile, there may be multiple ways to define the weights and the subclones that correspond to the same reconstructed profile when neither the weights nor the subclones are observed. From a statistical viewpoint, this can be phrased as an *identifiability* problem [Behr and Munk, 2015].

This issue of identifiability can be overcome by adding constraints to the model parameters and two major directions have been considered in the literature. One possibility is to constrain the latent profiles to belong to a pre-determined alphabet (e.g. $\{0, 1, 2, \ldots 10\}$). Necessary and sufficient conditions have recently been given for the problem to be identifiable under this constraint [Behr and Munk, 2015], and the same authors are working on an application to DNA copy number profiles [Behr et al., 2016]. Although true DNA copy numbers indeed are integers, restricting the latent profile values to a finite alphabet may not always be realistic in practice. Indeed, copy number signals measured from microarray or sequencing technologies are not directly proportional to true copy numbers due to non-linearities induced by the biological assays including saturation effects [Skvortsov et al., 2007].

New methods attempted to infer intra-tumor heterogeneity using copy number from HTS or SNP data by integrating mutation information [Ha et al., 2014, Oesper et al., 2013, Roth et al., 2014, Li and Li, 2014]. [Jiang et al., 2016] pointed out the fact that these methods do not take into account several samples at the same time except PyClone. However, PyClone does not enable copy number alterations to be subclonal. Deal with several profiles is a possibility to overcome the above-mentioned identifiability issue. Models assume that several profiles are observed and share the same set of latent profiles. From our perspective, this assumption is both justified by the underlying biology, and weak enough to be coherent with the observed copy number signals. This is the reason why we work under this assumption. We assume that several DNA copy number profiles are observed and that each of these profiles is a mixture of the same subclones (possibly with weight 0 for some of the subclones). This model is able to tackle either the intra or inter-tumoral heterogeneity by dealing with several samples.

5.7 Conclusion

This chapter aims to introduce the classical models of the matrix factorization that is the direction that we have chosen to analyze heterogeneity. We have seen that is possible to add several constraints on these models. This makes it possible to add biological priors in the form of various constraints. For example, a sample can be seen as a mixture of several cells and weights represent the proportion of each type of cell (archetypal analysis constraint). Then, we would like that latent profiles present a few alterations, in particular it could be interesting to add a fused LASSO constraint (sparse dictionary learning constraint).

Therefore, our contribution in this part deals with discovering heterogeneity from DNA copy number data by extending the existent models. We add constraints that provide a more realistic biological sense and integrate the BAF signal (Chapter 6). The goal is to discover characteristics of the resistant subclones in DNA copy number data. All information from SNP array data i.e. B allele fraction and total copy number at each SNPs is integrated by the intermediate of the PSCN (parent specific copy number) information (section 1.4.3). We are currently implementing the method as an R package named InCaSCN (Inferring cancer subclones using DNA copy number) to discover heterogeneity. We have this method to two real data sets that contain several samples at various time points and locations for the same patient (Chapter 7). A paper with J. Chiquet and P. Neuvial is currently in preparation.

Chapter 6

Inferring cancer subclones using parental DNA copy numbers

6.1 Introduction

The objective of this Chapter is to present our proposed approach to study cancer heterogeneity by using copy number alterations (CNA). That way, the model will be applicable either on the array-based Comparative Genomic Hybridization (aCGH), Single Nucleotide polymorphism (SNP) microarrays, whole exome sequencing (WES) or whole genome sequencing (WGS). We attempt to respond to the question of reconstructing the underlying subclones and the corresponding weights from a series of DNA copy number profiles measured by the technologies cited above.

This model presented here is applicable to samples from a same patient that have been taken at various time or spatial points [Schwarz et al., 2015], but also to samples from an homogeneous group of several patients. This assumption has already been made in the literature [Nowak et al., 2011, Masecchia et al., 2013]. Our model may be seen as an extension of these approaches, with the following original contributions:

1. leveraging the allelic signals available from SNP array or sequencing data in order to explicitly integrate parent-specific copy numbers [Olshen et al., 2011] in the model;

2. making the mixing weights interpretable as such by modeling each profile as a *convex combination* of latent profiles;

3. modeling tumor clonality at the level of copy number segments (not individual

probes), which is the level of information at which such events occur.

Our model has similarities with the model recently proposed in [Jiang et al., 2016] except that it enables us to analyze several samples to explore both intra-heterogeneity and inter-heterogeneity contrary to the model of [Jiang et al., 2016]. Indeed, this model is restricted to the first type of heterogeneity because its first aim is to build phylogenetic trees in order to understand the tumor history for a single patient.

The model that we propose in this Chapter is inspired by dictionary learning methods (see section 5.2) as models of [Nowak et al., 2011, Masecchia et al., 2013] and the convex combination in our model of the latent profiles to explain samples is inspired of archetypal analysis introduced in section 5.5.

We formulate the problem of estimating the parameters of the model as an optimization problem and propose an iterative algorithm to estimate these parameters. We assess the performance of this approach using realistic simulations based on real DNA copy number data [Pierre-Jean et al., 2015]. We also applied the model to two different kinds of real data sets (Chapter 7).

6.2 Model

6.2.1 Basic model

Our starting point is a latent feature model described by 5.1. This models has been used to model multi-sample aCGH data [Masecchia et al., 2013, Nowak et al., 2011].

Importantly, the latent profiles are the same for all samples, that is, $\mathbf{Z}_k$ does not depend on the sample index i. This assumption is crucial to make the model identifiable (see Section 5.6 for a brief discussion on identifiability).

Figure 6.1 represents the phenomenon of intra-tumoral heterogeneity. Figure 6.1(a) shows two tumor samples (green and yellow circles) composed of a mixture of "subclones" according to (5.1). In this example, for the top sample, the fraction of normal cells is $w_{10} = 20\%$, and the sample is a mixture of two tumor subclones with proportions $w_{11} = 60\%$, $w_{12} = 20\%$. For the bottom sample, the fraction of normal cells is $w_{20} = 40\%$, and the sample is a mixture of two tumor subclones with proportions $w_{21} = 30\%$, $w_{22} = 30\%$. For both samples, the corresponding (noiseless) copy-number profiles are displayed in Figure 6.1(b). The copy-number profiles $\mathbf{Y}_i$ of samples $i = 1, 2$ are given by the linear convex combination: $\mathbf{Y}_i = w_{i1}\mathbf{Z}_1 + w_{i2}\mathbf{Z}_2 + w_{i0}\mathbf{Z}_0$, for $i = 1, 2$ and where $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are the profiles of the two tumor latent profiles (top two lines), and the normal latent profile $\mathbf{Z}_0$.

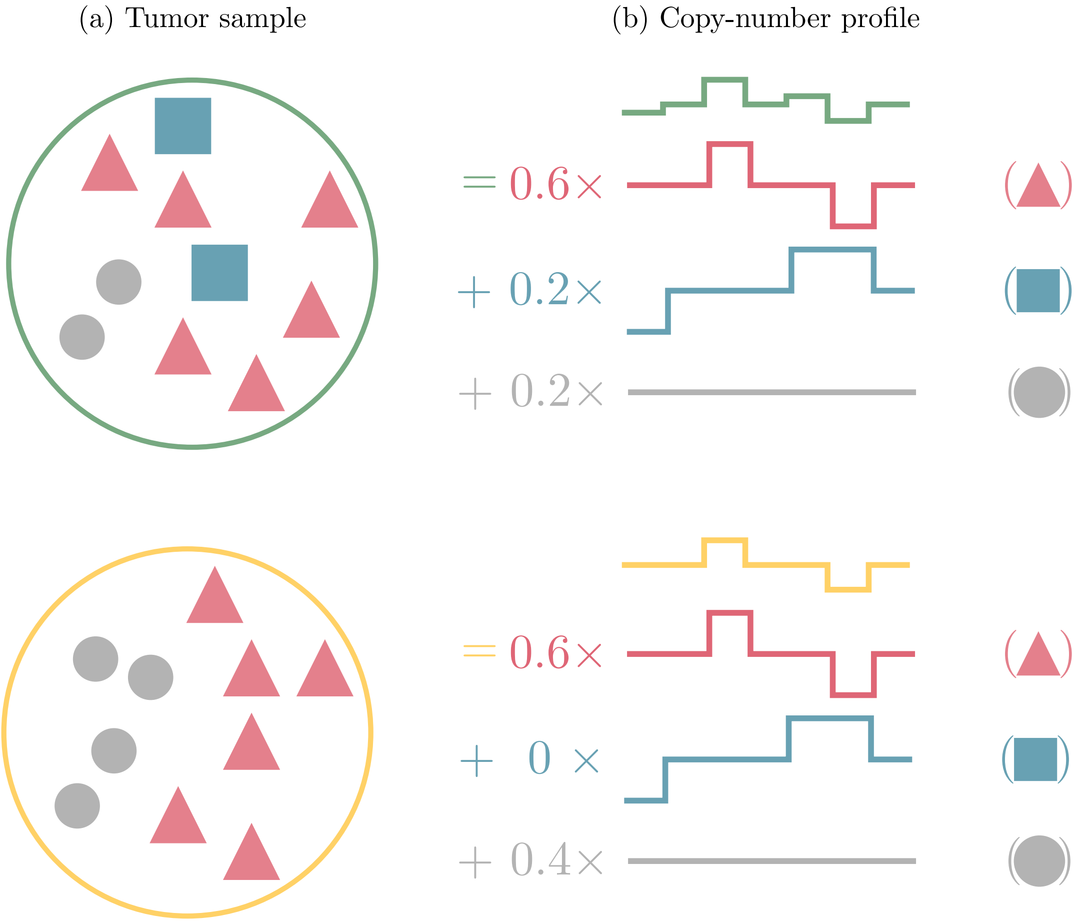


Figure 6.1 – (a) Heterogeneous tumor samples with a collection of normal cells (gray discs) and two cancer subclones (red triangles and blue squares). (b) A corresponding (noiseless) copy number profiles, written according to model (5.1) as a linear combination of latent profiles. This Figure is adapted from [Nowak et al., 2011].

6.2.2 State of the art

Total DNA copy number models In this section we describe the methods proposed by [Masecchia et al., 2013] and [Nowak et al., 2011]. In both papers, the parameters (w_{ik}) and (z_{kj}) are estimated by minimizing the total squared error of model (5.1) under additional constraints. Minimizing this loss function is equivalent to maximizing the likelihood of model (5.1) when the errors are independent and Gaussian. In [Nowak et al., 2011], a fused lasso constraint on the rows of $\mathbf{Z}$ encourages the latent profiles to be piecewise constant. Moreover, an ℓ_2 constraint on the weights is added in order to make the weights identifiable. The optimization problem considered in [Nowak et al., 2011] is then:

$$\min_{w_{ik}, z_{kj}} \sum_{i=1}^{n} \sum_{j=1}^{J} \left(y_{ij} - \sum_{k=1}^{p} w_{ik} z_{kj} \right)^2 + \sum_{k=1}^{p} \left(\mu \sum_{j=1}^{J} |z_{kj}| + \lambda \sum_{j=1}^{J-1} |z_{k(j+1)} - z_{kj}| \right)$$
$$\text{s.t.} \quad \sum_{i=1}^{n} w_{ik}^2 \leq 1 \quad \forall k = 1, \dots, p \quad (6.1)$$

The following equivalent matrix form is more compact:

$$\min_{\mathbf{W} \in \mathbb{R}^{np}, \mathbf{Z} \in \mathbb{R}^{Jp}} \left\{ \|\mathbf{Y} - \mathbf{W}\mathbf{Z}\|^2 + \mu \|\mathbf{Z}\|_1 + \lambda \left\| \mathbf{D}\mathbf{Z}^\top \right\|_1 \right\}$$
$$\text{s.t.} \quad \mathbf{W}_i \mathbf{W}_i^\top \leq 1 \quad \forall i = 1, \dots n, \quad (6.2)$$

where

- $\mathbf{Y}$ is the $n \times J$ matrix of copy-number signals for each sample,
- $\mathbf{W}$ is the $n \times p$ matrix of weights for each latent profile, and $\mathbf{W}_i$ its i-th row,
- $\mathbf{Z}$ is the $p \times J$ matrix of copy-number signals for each archetype,
- $\mathbf{D}$ is a $(J-1) \times J$ matrix for the first order differences:

$$\mathbf{D} = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix}$$

Two practical limitations of the above optimization problem are pointed out in [Masecchia et al., 2013]. First, the absence of positivity constraint on the weights makes it difficult to interpret the weights and the latent profiles themselves. Indeed, a copy number gain and a copy number loss can be explained by the same latent profile with weights of opposite signs, whereas they correspond to two markedly different biological events. Second, because the method is applied for each chromosome independently, it does not take advantage of the information of co-occurrence of alterations on two different chromosomes. In order to address these limitations, [Masecchia et al., 2013] have proposed to add a positivity constraint on the weights, and to incorporate location-dependent weights (θ_j) into the fused penalty in order to encourage breakpoints between chromosomes to be detected. The optimization problem considered in [Masecchia et al., 2013] is then:

$$\min_{\mathbf{W}\in\mathbb{R}^{np},\mathbf{Z}\in\mathbb{R}^{Jp}} \left\{ \|\mathbf{Y}-\mathbf{W}\mathbf{Z}\|^2 + \mu \|\mathbf{Z}\|_1 + \lambda \left\| \theta \mathbf{D}\mathbf{Z}^\top \right\|_1 \right\}$$
$$\text{s.t.} \quad \mathbf{W}_i\mathbf{W}_i^\top \leq 1, \quad \mathbf{W}_i \succeq 0 \quad \forall i = 1, \ldots n, \quad (6.3)$$

where $\theta \in \mathbb{R}^{L-1}$ encode fixed weights which make it possible to consider several chromosomes simultaneously. In practice, these weights are set to 1 for most loci, except near chromosome boundaries where they are “close to zero” [Masecchia et al., 2013]. Another difference with [Nowak et al., 2011] is the use of a structured penalty on the weights w_{ik} in order to induce sparsity along the set of weights associated to each sample separately. From our perspective, an important practical limitation of this method is that as the weights w_{ik} may not be interpreted as the proportion of latent profile k in sample i, because $\sum_k w_{ik}$ is not constrained to be 1. This issue cannot be addressed by a simple renormalization of the weights after estimation of $\mathbf{W}$: the constraint $\sum_k w_{ik} = 1$ must be incorporated into the estimation procedure.

The above models [Masecchia et al., 2013, Nowak et al., 2011] have been used to analyze *total* copy numbers, as measured by aCGH. However, SNP array and sequencing (WES or WGS) techniques not only measure total copy numbers but also carry information about the LOH status of a SNP or a region. From a biological perspective, it is, for instance, crucial to be able to differentiate a normal region with two copies from a copy-neutral LOH (cn-LOH) regions. Cn-LOH events are for instance implied in the inactivation of tumor suppressor genes as well as the activation of oncogenes in microsatellite stable and microsatellite instable tumors [Melcher et al., 2011, Andersen et al., 2006, Melcher et al., 2007].

For this reason, it is important to use the LOH information that is generally encoded at the level of Single Nucleotide Polymorphisms (SNPs) by *allelic ratios*, also known as BAF for “B allele fraction” [Staaf et al., 2008]. Recently, [Jiang et al., 2016] proposed a model that explores the intra-tumoral heterogeneity using sequencing data and infer phylogenetic tree of the tumor evolution and at the *parent-specific copy-number* (PSCN) level. PSCNs, also known as minor and major copy numbers, correspond to the number of DNA copies of each of the two inherited chromosomes [Olshen et al., 2011, Chen et al., 2011, Neuvial et al., 2011, Greenman et al., 2010]. For details on the computation of parental copy number signals, we refer to the section 1.4.3.

Parental copy number model To address the problem of intra-tumor heterogeneity, [Jiang et al., 2016] have developed a new model that integrates both copy number and mutation information from multi-sample HTS data. The data set is composed of several samples at various time-point or at multiple locations for a single patient. The model aims to identify the subpopulations within a tumor, determine the mutation profiles of the subpopulations and infer the tumor's phylogenetic tree. Authors provide an available implementation as an `R` package named `Canopy` (copy number and single nucleotide alteration analysis of tumor phylogeny). A Bayesian framework is used and enables to assess the quality of inference.

The model assumes that both minor and major copy numbers follow the basic model (5.1). That is, minor and major DNA copy number profiles for sample $i \in \{1, \ldots, n\}$ are modeled as:

$$y_{ijm} \quad = \quad \sum_{k=1}^{p} w_{ik} z_{kjm} + \epsilon_{im} \quad \text{for} \quad m = 1, 2\,, \quad \text{and} \quad j = 1, \ldots J\,, \tag{6.4}$$

where z_{kj1} is the minor copy number for the k-th latent profile at location l, z_{kj2} is the major copy number for the k-th latent profile at location j. The weights are assumed to be the same for minor and major copy numbers, as these weights correspond to the proportion of each latent profile. Therefore, summing (6.4) for $m \in \{1, 2\}$, the original latent model (5.1) is recovered.

This model aims to discover heterogeneity from HTS data, consequently, J can be large (WGS data). To reduce the dimension and to improve the signal to noise ratio, The analysis is performed at a segment level. Let us denote the minor and the major copy number matrices by $\mathbf{Y}_1 \in \mathbb{R}^{S \times n}$ and $\mathbf{Y}_2 \in \mathbb{R}^{S \times n}$ respectively after segmentation. Matrices are composed of n samples and S segments. The equation 6.5 is the equivalent matrix form of 6.4 after segmentation of the parental copy number signals.

$$\mathbf{Y}_m = \mathbf{Z}_m \mathbf{W}, \ \text{for } m = 1, 2 \tag{6.5}$$

where $\mathbf{Z}_1 \in \mathbb{N}^{S \times p}$ and $\mathbf{Z}_2 \in \mathbb{N}^{S \times p}$ are respectively integer-valued minor and major copy numbers matrices by assuming that there exists p subclones among the samples. $\mathbf{W} \in \mathbb{R}^{p \times n}$ is the sample proportion matrix. Because, $\mathbf{Y}_m$ are estimated on the segments (mean of the observations by segment), the model assumes that minor and major copy are normally distributed:

$$\mathbf{Y}_m \sim \mathcal{N}\left(\mathbf{Z}_m, \epsilon_m\right) \ \text{for } m = 1, 2 \tag{6.6}$$

For the modelisation of the mutations, the matrix containing the observed number of mutant sequences is denoted by $\mathbf{B} \in \mathbb{N}^{n \times L}$ where n is still the number of samples and L the total number of observed mutations. The number of mutations follows a binomial distribution. For the i-th sample and the l Single-nucleotide alteration (SNA),

$$r_{il} \sim \text{Binomial}\left(T_{il}, \text{VAF}_{il}\right) \tag{6.7}$$

where T_{il} is the total count for sample i at position l and VAF_{il} the probability of the position to be a Variant allele frequency (VAF).

6.2.3 Proposed model

Considering the advantages and the drawbacks of each method, we used a similar model that [Jiang et al., 2016]: namely, we assume that both minor and major copy numbers follow the basic model (5.1). We wish to constrain each latent profile to be piecewise constant, with a relatively small number of breakpoints (few alterations per latent profiles). We also constrain the weights for each sample to sum up to 1, so that the weight w_{ik} may be interpreted as the proportion of latent profile k in sample i. For instance, some breakpoints will be observed only for a subset of features. In this case where deconvolution is applied to minor and major copy number, this also makes sense because it's common that only one of the two parental copies changes at each breakpoints [Neuvial et al., 2011]. As a result, we propose to solve the following optimization problem:

$$\min_{\mathbf{W}\in\mathbb{R}^{np},\mathbf{Z}_m\in\mathbb{R}^{Jp}} \left\{ \sum_{m=1}^{2} \|\mathbf{Y}_m - \mathbf{W}\mathbf{Z}_m\|^2 + \lambda_m \left\|\mathbf{D}\mathbf{Z}_m^\top\right\|_1 \right\}$$
$$\text{s.t.} \quad \mathbb{1}_p^\top \mathbf{W}_i = 1, \quad \mathbf{W}_i \succeq 0 \quad \forall i = 1, \ldots n, \tag{6.8}$$

where $\mathbf{W}$ and $\mathbf{D}$ are defined as above and

- $\mathbf{Y}_m = (y_{ijm})_{i=1\ldots n, j=1\ldots J}$ is the $n \times J$ matrix of minor (for $m = 1$) and major (for $m = 2$) copy number signals for each sample $i = 1 \ldots n$,
- $\mathbf{Z}_m = (z_{kjm})_{k=1\ldots p, j=1\ldots J}$ is the $p \times J$ matrix of minor (for $m = 1$) and major (for $m = 2$) copy numbers for each archetype $k = 1 \ldots p$.

Note that as breakpoints may occur more often in one of the minor or the major copy number signal than in the other one, we do not constrain the optimal penalty coefficients λ_m to be equal for $m = 1$ and $m = 2$.

Segmentation step As the model [Jiang et al., 2016], we perform the analysis at a segment level in order to reduce dimension and improve the signal to noise ratio. By using the package `jointseg` [Pierre-Jean et al., 2015], it is possible to jointly segment TCN and DoH on the whole dataset, that is, simultaneously on all samples. We use the fast and efficient Recursive Binary Segmentation (RBS) method to find candidate breakpoints [Gey and Lebarbier, 2008], and prune these candidates using dynamic programming [Harchaoui and Lévy-Leduc, 2008, Bleakley and Vert, 2011] and section. 2.2.6). We compute the average TCN and DoH within each segment s for $s = 1, \dots S$, where $S - 1$ is the number of breakpoints, as:

$$c_{is} = \frac{1}{|s|} \sum_{j \in s} c_{ij} \text{ for } \ ,$$

$$d_{is} = \frac{1}{|s|} \sum_{j \in s} d_{ij} \text{ for } \ ,$$

where $i = 1, \dots, n$ denotes the i-th sample and $|s|$ the number of data points in segment s. Then minor and major copy number estimations by segments are defined by,

$$y_{is1} = c_{is}(1 - d_{is})/2$$

and

$$y_{is2} = c_{is}(1 + d_{is})/2,$$

respectively.

We point out here that contrary to the model in [Jiang et al., 2016], our proposed model can be applied on data from microarray but also to discover inter-tumoral heterogeneity.

The table 6.1 summarizes the features of methods reviewed above.

Reference	Nowak et al.	Masecchia et al.	Jiang et al.	Our method
Intra-heterogeneity?	√	√	√	√
Inter-heterogeneity?	√	√	-	√
Microarray application?	√	-	-	√
HTS application?	√	-	√	√
Mutation information?	-	-	√	-
Using BAF information?	-	-	√	√
Whole genome application?	-	√	√	√
Available as an R package?	√	-	√	√

Table 6.1 – Features of existing models.

6.3 Estimation procedure

6.3.1 Algorithm

Several types of inference methods could be considered to tackle the problem of inferring latent profiles and weights in our model. We used the alternative optimization of the different parameters when the others are fixed. This is a common method used for NMF models but also for archetypal analysis (see section 5.4).

Even though the optimization problem (6.8) is not jointly convex in $(\mathbf{W}, \mathbf{Z}_1, \mathbf{Z}_2)$, it is convex with respect to each variable $\mathbf{W}$ or $\mathbf{Z}_1$ or $\mathbf{Z}_2$ when the other two are held fixed. A natural way of optimizing the cost function 6.8 is, therefore, to alternate the minimization between $\mathbf{W}$ and $\mathbf{Z}_1$ and $\mathbf{Z}_2$, fixing one of the parameters and optimizing with respect to the others. This is what we describe in Algorithm 5.

```
Data: Minor copy number profile $\mathbf{Y}_1$, major copy number profile $\mathbf{Y}_2$, penalty
      coefficients $\lambda_1$ and $\lambda_2$;
Parameters : Number of latent profiles $p$, maximal number of iterations $T$;
Result: Minor and major latent profiles denoted by $\hat{\mathbf{Z}}_1$ and $\hat{\mathbf{Z}}_2$; weight matrix
        $\hat{\mathbf{W}}$;
Initialization: Clustering on total copy numbers $\mathbf{Y} = \mathbf{Y}_1 + \mathbf{Y}_2$ by considering
                loci as variables ;
Selection of $p$ clusters ;
$\mathbf{Z}_1^{(0)}$: mean of minor copy number profiles by each cluster ;
$\mathbf{Z}_2^{(0)}$: mean of major copy number profiles by each cluster ;
for $t \leftarrow 1, \ldots T$ do
    $\mathbf{W}^{(t)} \leftarrow \operatorname*{arg\,min}_{\mathbf{W} \in \mathbb{R}^{np}} \sum_{m=1}^{2} \left\| \mathbf{Y}_m - \mathbf{W}\mathbf{Z}_m^{(t-1)} \right\|^2 \quad \text{s.t.} \quad \mathbb{1}_p \mathbf{W}_i = 1, \quad \mathbf{W}_i \succeq 0,;$
    $\mathbf{Z}_1^{(t)} \leftarrow \operatorname*{arg\,min}_{\mathbf{Z}_1 \in \mathbb{R}^{Sp}} \|\mathbf{Y}_1 - \mathbf{W}^{(t)}\mathbf{Z}_1\|^2 + \lambda_1 \|\mathbf{D}\mathbf{Z}_1^\top\|_1;$
    $\mathbf{Z}_2^{(t)} \leftarrow \operatorname*{arg\,min}_{\mathbf{Z}_2 \in \mathbb{R}^{Sp}} \|\mathbf{Y}_2 - \mathbf{W}^{(t)}\mathbf{Z}_2\|^2 + \lambda_2 \|\mathbf{D}\mathbf{Z}_2^\top\|_1;$
```

Algorithm 5: Estimation of minor and major copy number latent profiles and weight matrix.

We show in section 6.3.3 that the update of $\mathbf{W}^{(t)}$ involves solving a linear inverse problem, and that the updates of $\mathbf{Z}_1^{(t)}$ and $\mathbf{Z}_2^{(t)}$ can be reduced to a Lasso problem. Therefore, Algorithm 5 may be implemented using standard optimization tools. Our implementation relies on the R packages **limSolve** and **glmnet** [den Meersche et al., 2009, Friedman et al., 2010].

6.3.2 Initialization

As noted above, as problem (6.8) is not jointly convex in ($\mathbf{W}$, $\mathbf{Z}_1$, $\mathbf{Z}_2$), we have no formal guarantee that Algorithm 5 will reach a global minimum. We emphasize that this was also the case for the optimization problems (6.2) and (6.3) studied in [Nowak et al., 2011, Masecchia et al., 2013]. One possible way to overcome this issue is to run the algorithm with a range of different initial values (such as p randomly selected rows of $\mathbf{Y}$), and choose the final estimate that minimizes the loss. Instead, [Nowak et al., 2011] proposed to initialize with the p first principal components of $\mathbf{Y}$. Here, we choose to initialize our algorithm by performing a clustering with p classes on the matrix $\mathbf{Y}$ by considering loci as variables. We use a hierarchical clustering using Ward's method [Ward Jr, 1963] computed on $\mathbf{Y}$. Letting $\mathcal{C}_k$ be the k-th cluster, the initial values for Z_1 and Z_2 are:

$$z_{ksm}^{(0)} = \frac{1}{Card(\mathcal{C}_k)} \sum_{i \in \mathcal{C}_k} y_{ism} \text{ for } m = 1, 2$$

We have compared the performance of this initialization with the above-mentioned approach with multiple randomly selected starting points. We now show that our proposed initialization algorithm provided fairly good results in term of minimization of loss while remaining substantially faster than the other option.

We perform experiments on a toy example (30 simulated profiles with 5000 loci and 6 breakpoints in latent profiles). For a model with 4 latent profiles and one couple of (λ_1, λ_2), performing the random initialization over 100 replications takes approximately 70s on a standard laptop contrary to clustering initialization that takes 0.84s. Assuming that we would like to explore 10 values for each λ_1 and λ_2, the random initialization takes around 2 hours just for 4 latent profiles while clustering initialization takes around 2 minutes.

Results in terms of loss provided by the clustering initialization are close to the average of the loss provided by random initialization. Consequently, using this clustering to initialize Z is a good compromise between time and performance (Fig. 6.2). It is obvious that using clustering to initialize the algorithm will be less time consuming if we deal with larger data set for instance several chromosomes (namely potentially more breakpoints).

A solution to be the closest to the optimum of (6.8) could be to perform first clustering initialization, choose the best couple (λ_1, λ_2) and run another time the algorithm by taking random initialization.

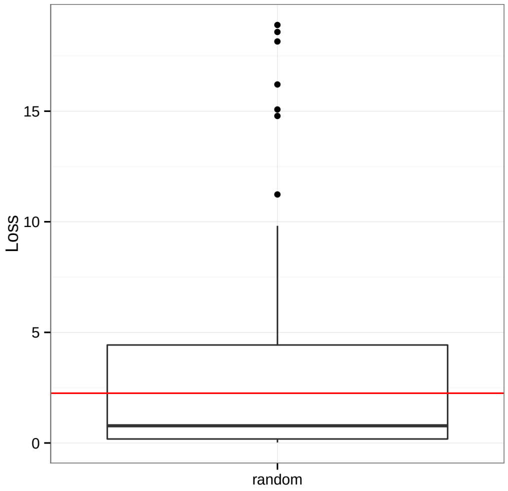


Figure 6.2 – A comparison between 100 random initializations (boxplot) and the clustering initialization (red line) in terms of loss (Toy example).

6.3.3 Solving the optmization problem (6.8)

Solving in W Assume that $\mathbf{Z}_m$ are given. We need to solve (6.8) in $\mathbf{W}$, which is equivalent to solving

$$\min_{\mathbf{W}\in\mathbb{R}^{np}} \sum_{m=1}^{M} \|\mathbf{Y}_m - \mathbf{W}\mathbf{Z}_m\|^2 \quad \text{s.t.} \quad \mathbb{1}_p^\top \mathbf{W}_i = 1, \quad \mathbf{W}_i \succeq 0, \tag{6.9}$$

The weights of each patient can be treated independently. We thus consider patient i with weights $\omega \triangleq \mathbf{W}_i$ and drop the corresponding index for clarity purpose. The corresponding optimization problem is

$$\min_{\omega\in\mathbb{R}^{p}} \sum_{m=1}^{M} \left\|\mathbf{y}_m - \mathbf{Z}_m^\top \omega\right\|^2 \quad \text{s.t.} \quad \mathbb{1}_p^\top \omega = 1, \quad \omega \succeq 0, \tag{6.10}$$

where $\mathbf{y}_m \in \mathbb{R}^S$ are the CNV at each locus in signal m for patient i. The above problem is equivalent to

$$\arg\min_{\omega\in\mathbb{R}^{p}} \left\|\tilde{\mathbf{y}} - \tilde{\mathbf{Z}}\omega\right\|^2 \quad \text{s.t.} \quad \mathbb{1}_p^\top \omega = 1, \quad \omega \succeq 0, \tag{6.11}$$

where $\tilde{\mathbf{y}} = (\mathbf{y}_1, \ldots, \mathbf{y}_M) \in \mathbb{R}^{SM}$ and $\tilde{\mathbf{Z}} = (\mathbf{Z}_1, \ldots, \mathbf{Z}_M) \in \mathbb{R}^{SM\times p}$ are obtained by stacking the M components of $\mathbf{y}$ and $\mathbf{Z}$.

Problem (6.11) is a least-squares problem with equality constraint plus inequality constraints for the non-negativity of the coefficient. This is a linear inverse problem that can be solved in `R` with the package **limSolve**.

Solving in Z

Lemma 3. *The problem of solving* (6.8) *for a fixed* $\mathbf{W}$ *can be cut into two independent LASSO problems in* $(\mathbf{Z}_1, \mathbf{Z}_2)$.

The proof of lemma 3 is given in Appendix D.

6.3.4 Parameter selection

The optimal values of λ_1, λ_2 and p are unknown. This section is inspired of the approaches used in [Nowak et al., 2011, Masecchia et al., 2013]. We propose to (i) select the appropriate values of λ_1 and λ_2 for a fixed p and (ii) choose p. For each p, we explore a grid of values for λ_s.

In order to select the best model for a fixed number of latent profiles p, we use the Bayesian Information Criterion (BIC) [Schwarz et al., 1978] adapted to our context. We search to minimize :

$$(nS) \times \log\left(\frac{\|\mathbf{Y} - \widehat{\mathbf{W}}\widehat{\mathbf{Z}}\|^2}{nS}\right) + k(\widehat{\mathbf{Z}})\log(nS) \tag{6.12}$$

where $\widehat{\mathbf{Z}} = \widehat{\mathbf{Z}}_1 + \widehat{\mathbf{Z}}_2$ corresponds to the estimated total copy number latent profiles, and $k(\widehat{\mathbf{Z}}) = \sum_{m=1}^{2}\sum_{k=1}^{p}\sum_{s=1}^{S-1} \mathbb{1}_{\{\hat{z}_{k(s+1)m} \neq \hat{z}_{ksm}\}}$ is the total number of breakpoints across all latent profiles. This criterion enables us to strike a balance between over-fit and under-fit models. Once the optimal values of the tuning parameters are selected for each value of p, the next step is to choose the best number of latent profiles. We use the percentage of variation explained (PVE) for each p. The PVE is defined by the following:

$$\mathrm{PVE}(p) = 1 - \frac{\|\mathbf{Y} - \widehat{\mathbf{W}}\widehat{\mathbf{Z}}\|^2}{\|\mathbf{Y} - \overline{\mathbf{Y}}\|^2}, \tag{6.13}$$

where $\overline{\mathbf{Y}}$ is a $n \times S$ matrix whose i-th row is identically equal to $\sum_{s=1}^{S} y_{is}/S$, that is, the mean total copy number signal of sample i. Following [Nowak et al., 2011], we assume that PVE(p) reaches a plateau when p is larger than the optimal number of latent profiles in the model. Therefore, we propose to select the value of p at which the PVE reaches a plateau.

6.3.5 Implementation

The method proposed in this Chapter is implemented in an `R` package named InCaSCN (Inferring Cancer Subclone using copy number signals). The package includes scripts to reproduce the experiments described in this chapter but is not yet publicly available. The vignette of the `R` package is presented in the last part of the thesis in Appendix E.3.

6.4 Simulation study

6.4.1 Setting

We simulated 100 data sets composed of 30 tumor samples and 5 latent profiles with alterations with the following characteristics:

- Each matrix $\mathbf{W}$ is different for the 100 data sets but the latent profiles are the same across the data sets.
- The simulated samples before segmentation contain $J = 24,000$ loci, including one-third of heterozygous SNPs.
- Each tumor sample is composed of a mixture of at least two different latent profiles including normal.

To simulate latent profiles, we have annotated a new dataset with a very low contamination by normal cells. Then, we proceeded by resampling as explained in Chapter 3 to create the latent profiles.

We simulate, the latent profiles with the following characteristics:

- Each latent profile is represented at a proportion between 20 and 60%.
- For two different latent profiles, the change-points do not occur at the same locations except for two simulated latent profiles.

We present the simulated latent profiles in Figure 6.3. The breakpoints of latent profiles 3 and 6 occur at the same locations, however, the middle region is cn-LOH in latent profile 3, and normal in latent profile 6. These two types of regions cannot be distinguished by looking at the TCN signal but identifiable from parental copy numbers (see section 1.4.3).

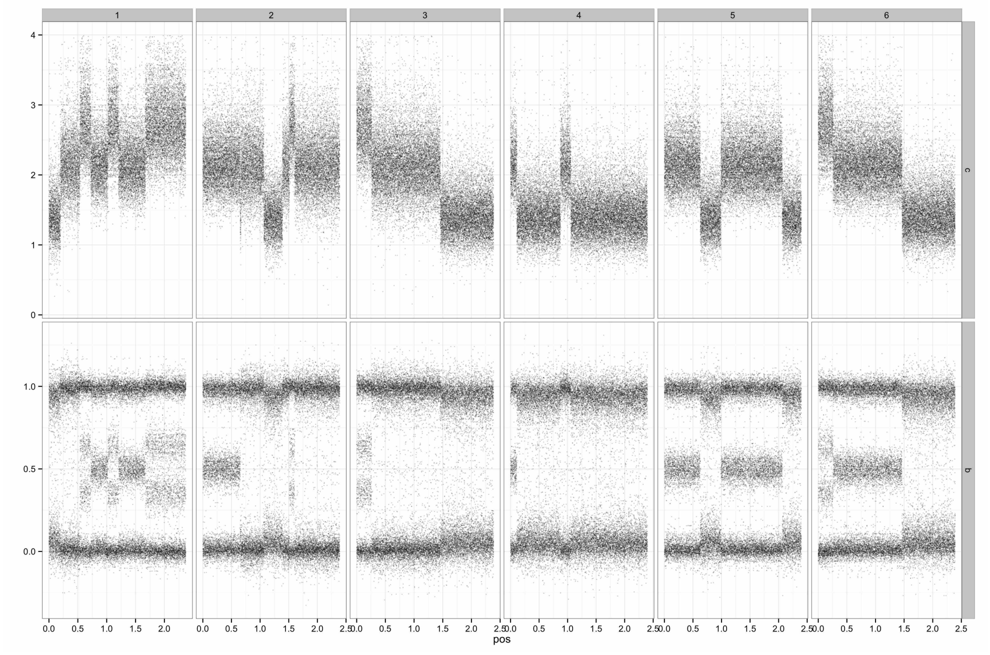


Figure 6.3 – Simulated latent profiles

Two examples of simulated total copy number profiles are presented in Fig. 6.4. These examples are the noisy versions of the profiles shown in Fig. 6.1 (b). Vertical red lines represent the true segmentation, while green and yellow segments represent the mean after a joint segmentation of the two samples. Even if the segmentation is the same for the two samples, and therefore the yellow sample has to be oversegmented, the yellow sample appears to get the correct number of segments. Indeed, the means of the yellow sample before and after the breakpoints that are not true breakpoints in this signal are very close and seem to be same.

6.4.2 Parameters of the model

For InCaSCN, for λ_1 and λ_2 parameters, we choose a grid of 10 values between 10^{-6} and 10^{-5}. This range of values is small because we are not in a high dimensional case, since $n > p$ and S is small. The number of candidate latent profiles p varies between 2 and 14.

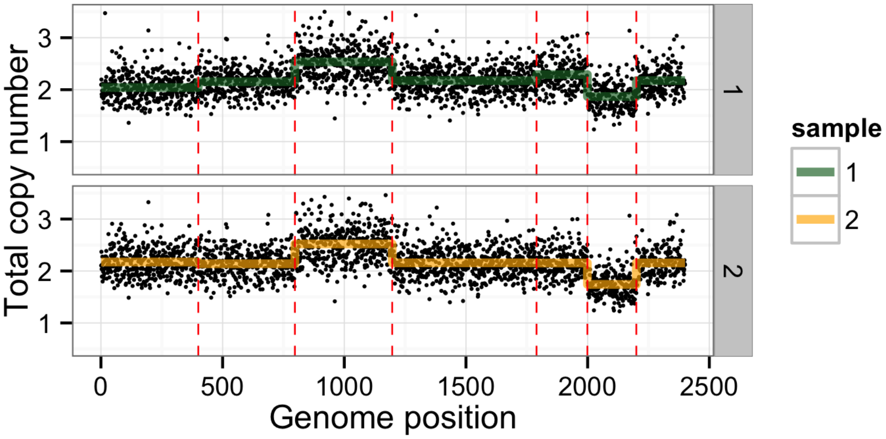


Figure 6.4 – Example of a simulated profile.

6.4.3 Comparison with other methods

Our algorithm has been applied to the matrices $(\mathbf{Y}_1, \mathbf{Y}_2)$ (minor and major copy number signals), but also on the corresponding total copy number estimates $\mathbf{Y} = \mathbf{Y}_1 + \mathbf{Y}_2$ for direct comparison with existing methods. We compare the results of InCaSCN with those of FLLAT[1]. Our initial objective was to compare with e-FLLAT as well, but the available implementation was not applicable to profiles of length $J = 24,000$[2].

Therefore, we compared performance of three methods namely:

- InCaSCN on parental copy number profiles
- FLLAT on total copy number profiles

[1] FLLAT is implemented as an R package available from CRAN: https://cran.r-project.org/web/packages/FLLat/index.html.

[2] No implementation of e-FLLAT is currently available from the authors' website or public repositories. Upon our request the authors gave us access to a repository containing 64 Python scripts and C code implementing their approach. Despite our efforts in obtaining results with this implementation, we found that it was too time and memory-consuming to process profiles of length $J = 24,000$. Our attempts with profiles of length $2,400$ were not successful either, even after pre-segmentation of these profiles. For comparison, it took more than 24 hours to obtain the results of the method for only 3 candidate values of p on the example data set provided with this implementation ($J = 500$ loci, $n = 30$ input copy-number profiles). The very high memory footprint of this implementation (dozens of gigabytes of RAM) also prevented us from running this method in parallel on several data sets. The maintainer confirmed that the current implementation was indeed unexpectedly time and memory-consuming.

- InCaSCN on total copy number profiles

In this section, we present a setting of simulations to show the ability of InCaSCN method of inferring accurately the latent parental copy number profiles and the weight matrix. Performance evaluation was done at three levels. The first one is the ability to recover the correct number of latent profiles that composed the data set ($\hat{p}$), The second one is the ability to reconstruct samples ($\hat{\mathbf{Y}}$). However, even if it is crucial to well recover the observed profiles, recover them entirely depends on the inference of weights and latent profiles. Thus, it is important to also evaluate how these two components are similar to the true ones. Therefore the last one is the ability to properly recover the weights and the latent profiles ($\hat{\mathbf{W}}$ and $\hat{\mathbf{Z}}$).

Evaluation at the level of reconstructed profiles

For each value of p, for InCaSCN, we keep only the best combination $(\lambda_1^p, \lambda_2^p)$ to compute the PVE defined by equation (6.13) as explained in section 6.3.4. On the other hand, FLLAT proposes to compute the PVE for each p for only one particular set of coefficients of penalties. Then, after the choice of p, the algorithm is run one more time in order to tune the penalty coefficients. The PVE measures the percentage of variation explained by the model.

Evaluation at the level of inferred weights

One possible way to evaluate the quality of the inference of weights is to compare $\mathbf{W}$ and $\hat{\mathbf{W}}$ by directly comparing their values. However, it is not guaranteed that latent profiles are in the same order in $\mathbf{W}$ and $\hat{\mathbf{W}}$. For this reason, we compute a correlation measure described by equation 6.14:

$$corr(\hat{w}_{\bullet k'}, w_{\bullet k}) = \frac{1}{n} Card(\{i \in \{1, \dots n\} \text{ such that } |\hat{w}_{ik'} - w_{ik}| < \epsilon\}) \tag{6.14}$$

where $k' = 1, \dots, \hat{p}$ and $k = 1, \dots, p$ and we consider $\epsilon = 0.1$ (variation of 10% between the truth and the estimation). Then, for each $w_{\bullet k}$ there exists a $\hat{w}_{\hat{k}}$ such that $hatk = \arg\max_{k'} corr(\hat{w}_{\bullet k'}, w_{\bullet k})$. We define a loss as:

$$\mathbf{C}(\mathbf{W}, \hat{\mathbf{W}}) = \sum_{k=1}^{p} \|w_{\bullet k} - \hat{w}_{\bullet \hat{k}}\|_2^2 \tag{6.15}$$

We consider another one metric based on the fact that one of the purposes of this model is to detect heterogeneity in tumor samples. A way to achieve this goal is to look at the weight matrix that reflects the composition of each sample. We compute a rand index [Rand, 1971] between clustering on $\mathbf{W}$ (that contains the true composition of profiles) and clustering on $\hat{\mathbf{W}}$ for each method: $rand(\mathbf{W}, \hat{\mathbf{W}})$. We use the euclidean distance computed on matrices $\mathbf{W}$ and $\hat{\mathbf{W}}$ associated to the Ward's method [Ward Jr, 1963]. The number of clusters is the number of latent profiles namely $\hat{p}$ and p for $\hat{\mathbf{W}}$ and $\mathbf{W}$ respectively. This evaluation method does not require to perform the matching between the inferred and true latent profiles. It also take it possible to compare the two matrices $\hat{\mathbf{W}}$ and $\mathbf{W}$ even if $\hat{p} \neq p$.

Evaluation at the level of inferred latent profiles

Recover the actual composition of the observed profiles is important but another important point is to recover the good features of each latent profiles. Indeed, the final aim is to detect genetic biomarkers that may be linked to the resistance to the drugs for example. Therefore, detecting the relevant losses and gains is crucial to help biologists to develop original targeted treatments.

We perform a ROC curve based evaluation on latent profiles. FLLAT does not segment initial data so that the signal length of $\hat{\mathbf{Z}}$ is the same than the signal length of $\mathbf{Y}$ i.e J. In our case, because of the segmentation step, it requires to "expand" each latent profile that composes the matrix $\hat{\mathbf{Z}}$. In other terms, instead of dealing with a $p \times S$ matrix, we perform the evaluation of the expanded $p \times J$ matrix.

We collect each region simulated as altered (gain, loss or cnLOH), and check whether these regions can be considered as altered in estimated latent profiles, with a tolerance parameter that varies between 0 and 1. In each region simulated as altered, for FLLat and InCaSCN on TCN, we check if TCN is larger or smaller than the basal level of a normal region with a certain tolerance.

A toy example is shown in Fig. 6.5 to illustrate how observations are partitioned (considering the basal level of a normal region equal to 2) in true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). We define the negatives as loci considered as conversely by the considered method (filled and empty blue circles) i.e. all loci that are in the tolerance area (gray band). All loci out of the area are considered as altered (filled and empty red circles). Then, all loci that are in a region that is considered as normal in annotated data set and that are in the gray area are TN (blue filled circles). The loci that are in a region considered as not normal in the annotated data set and that are not in the gray area are considered as TP (red filled circles).

Note that in this evaluation, we do not distinguish losses from gains but only altered regions from normal regions. This is in favor of FLLAT because the model can consider a gain as a loss in a tumor sample by using a negative weight.

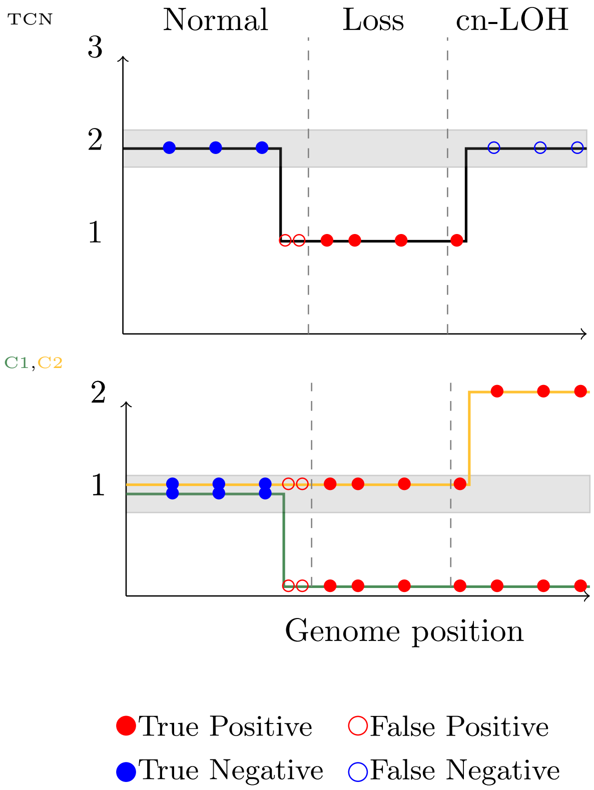


Figure 6.5 – Definition of True Positive (TP), False Positive (FP) and False Negative (FN). Grey area defined the tolerance area for which the loci within are considered as negative.

6.4.4 Results

Good performance for the model selection

The plateau of PVE curves begins at $\hat{p} = 5$ for the both FLLat and InCaSCN. This means that after $\hat{p} = 5$ latent profiles adding another one does not improve the reconstruction of the matrix $\mathbf{Y}$. We recover all latent profiles in most cases for the three methods. For the three methods, the normal latent profile is not recovered. This is probably due to the weak proportion of the normal component in samples (around 20%) or by the fact that the other inferred latent profiles are shrink toward the basal levels. This component is not essential to recover genetic biomarkers associated with the drug

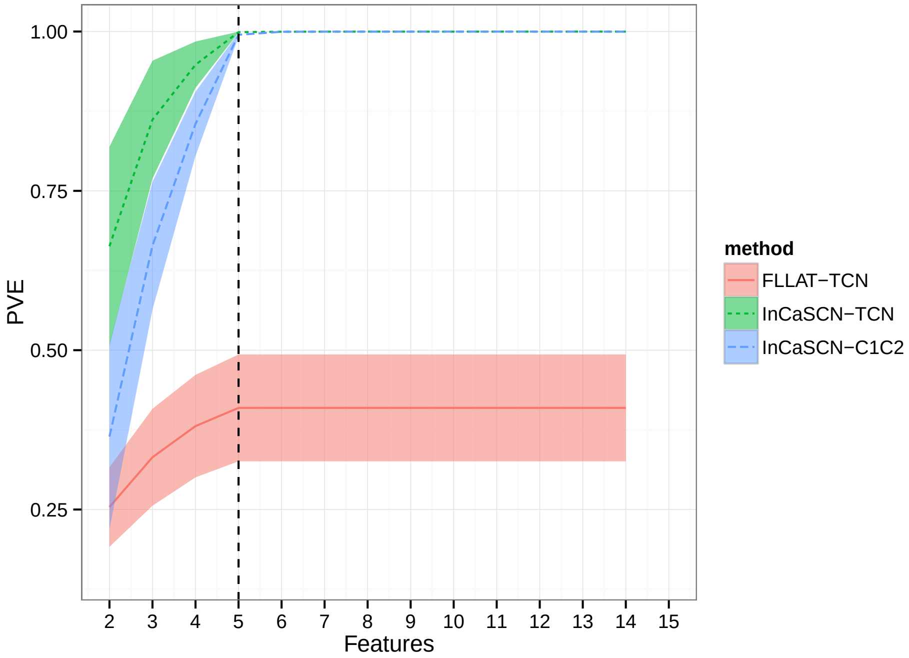


Figure 6.6 – Percentage of variation explained (PVE) as a function of the number of latent profiles p

resistance. Improvement about this drawback will be discussed in the conclusion of this Chapter.

Reconstruction of profiles is better with InCaSCN

We can also conclude from Fig. 6.6 that InCaSCN perfectly explains the variance of the observations, unlike FLLAT. After further analysis, we conclude that it is probably due to the tuning of the penalty coefficients that makes latent profiles too flat. For these experiments, we have used the default tuning of the package. However, the results of PVE from FLLat in our simulations are coherent with the results presented in the initial article [Nowak et al., 2011].

Better estimation and interpretation of weights by using InCaSCN

In this section, we present the ability of the InCaSCN method to recover the simulated weights. Even if in our simulations the weights are all positive, and positive weights are admissible in the FLLat model, this method is not able to estimate them properly

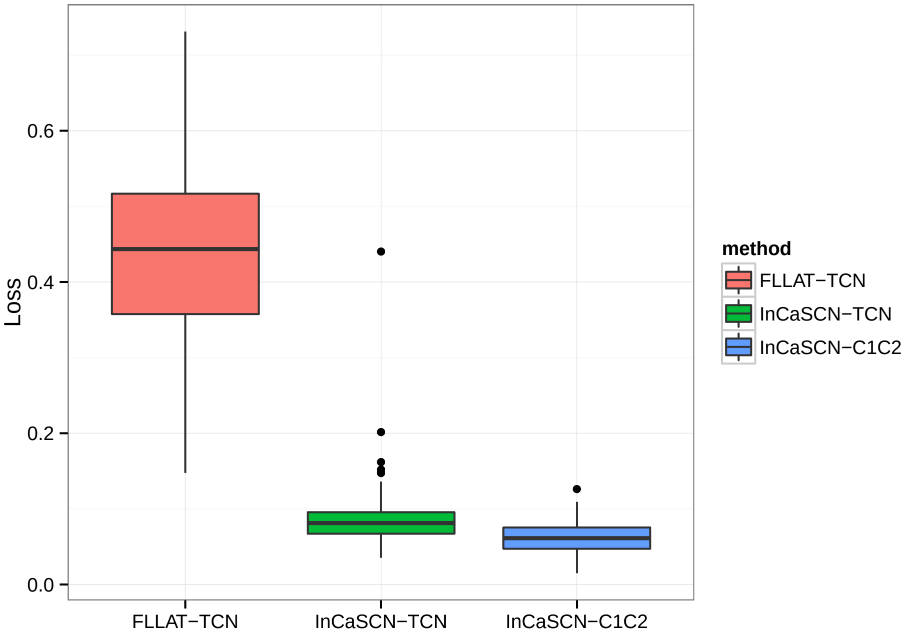


Figure 6.7 – Distribution of loss of matrix weight: $\mathbf{C}(\mathbf{W}, \hat{\mathbf{W}})$

(Fig. 6.7). In fact, the weight matrix of FLLAT appears to be far from the truth, unlike weights inferred by InCaSCN. The weights from InCaSCN are pretty close to the true ones even when only the TCN is used (Fig. 6.7). We present the results of the rand index between the clustering on inferred weights and on the true ones. The performance of InCaSCN and FLLat are presented in Fig. 6.8. Globally, the methods applied on TCN provide similar results and using parental copy numbers in our model clearly improves the clustering. To conclude, by comparing InCaSCN on parental copy number profiles and InCaSCN on TCN profiles, it is clear that it is very useful to include the BAF information in our model. This huge difference in performance between the two models is probably due to the simulated latent profiles 3 and 6 that are not identifiable when only TCN is used.

Inferred latent profiles from InCaSCN recover the true alterations.

Finally, the last point of this evaluation is to confirm that inferred latent profiles are well reconstructed. We summarize the ROC curves introduced in section 6.4.3 of each simulation data set by their respective AUC (area under the curve). The closer AUC is to 1, the better is the detection of alterations. The performance of InCaSCN and

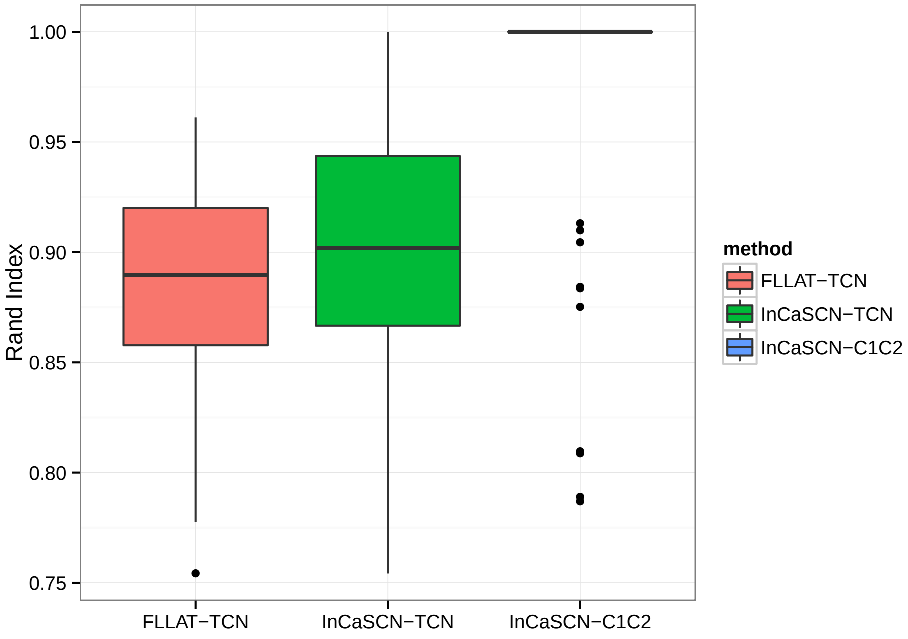


Figure 6.8 – Rand index distributions across 100 simulations between clustering on $\mathbf{W}$ and $\hat{\mathbf{W}}$ for each method.

FLLat are presented in Fig. 6.9 for their respective inferred latent profiles. First, InCaSCN is better than FLLAT even when only the TCN is used. This can be due to the poor calibration of penalties that force latent profiles to be equal to 0 at a large number of loci. Then, it is clear that using parental copy numbers helps to recover more alterations. This gain between InCaSCN applied on TCN and applied on parental copy numbers is probably due to all cn-LOH regions detected with parental copy numbers but not with total copy number.

6.4.5 Summary of the results

InCaSCN enables to recover both simulated latent profiles when the weight is high enough and weights with a small error. The results on realistic simulation framework are very promising for the application to real data sets. In addition, it was no obvious that our proposed model would be able to recover the true latent profiles. Indeed, because of saturation phenomenon on microarrays, ploidy and cellularity of samples, for instance, the TCN of a normal region or copy neutral LOH is not perfectly centered at 2 like TCN of a region gain is not centered at 3 (see Fig. 8.6 in Chap. 8). This

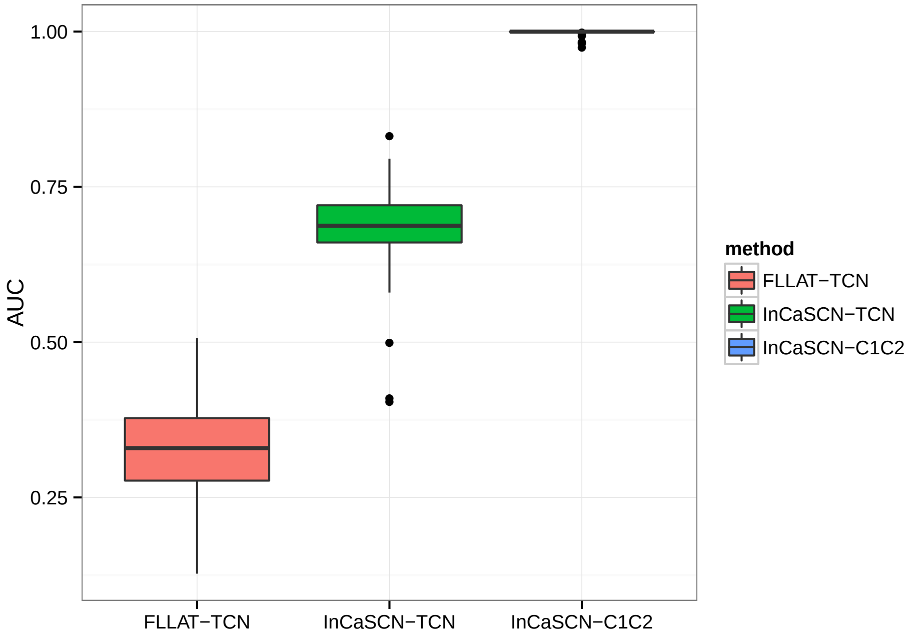


Figure 6.9 – AUC distribution across 100 simulations (detection of alterations in latent profiles).

could disturb inference of latent profiles and then the inference of weights. In the next chapter, we present the results of InCaSCN on two real data sets and we show that inference of heterogeneity is useful to improve discovering of the composition of the sample (weight inference) and of molecular targets (latent profiles inference).

6.5 Conclusion

In this Chapter, we proposed a method that we called InCaSCN (Inferring Cancer Subclone by using Copy Number) that is an extension of previous latent models. Our model integrates the BAF information and new more interpretable constraints. Our simulations show that our model outperforms the existing method at several levels but also that it is crucial to take into account the BAF information to discover the cn-LOH regions. In addition, the model gives a sparse representation of the observed signal within the meaning that the starting signal is of length J is explained at the end by a vector of weights of length $p \ll J$. By means of the segmentation step and fused constraints in the model, the latent profiles contain only a few breakpoints that is an advantage for the biological interpretation. The segmentation step gives also the advantage to get a fast algorithm that enables to apply it on the whole genome and therefore discover co-occurring alterations in several chromosomes. We conclude that this statistical model may help to more accurately discover frequent alterations and also characteristic subgroups of patients.

An important point that could be improved is that the level of sparsity is the same for all latent profiles. This implies the same number of breakpoints between two latent profiles but there is no reason to have this constraint. A solution to deal with this issue will be to penalize differently each latent profile even if it could be time-consuming. Indeed, this induces the tuning of a larger number of penalty coefficients. Another improvement could be the inclusion of a new parameter in the model in order to take into account the different values of the ploidy and the cellularity for each sample.

In the `R` package `Canopy`, toy data sets are available and it could be interesting to compare the results of the two methods. As the package is very recent, this comparison has not been done yet.

To summarize, InCaSCN is relatively efficient and fast due to the segmentation step and can be applied to real data sets either with a large number of patient or a large number of biomarkers. We applied the model on two real data experiments in the next chapter.

Chapter 7

Application to real data

In this chapter, we present two applications to real data of InCaSCN method presented in the previous chapter. The first application is from a public data set on ovarian cancer. The second one is from collaboration with Institut Curie on breast cancer. These two applications aim to demonstrate that is possible to use the model in real cases and to illustrate what kind of conclusions can be drawn from these analyses.

7.1 Application to high grade serous ovarian cancer

7.1.1 A brief introduction to ovarian cancer

High grade serous ovarian cancer (HGSOC) is the most common one among all ovarian cancer cases (70%) but unfortunately also the most malignant. Recent researches highlighted that finally the majority of the HGSOC starts in the fallopian tube and not in the ovary. Because of this location, tumor cells spread quickly through the entire abdomen and by the time symptoms often appear when cancer reached a high stage. Therefore, the survival for patients is often weak. The overall survival of women with HGSOC has not changed in over 50 years [Kurman, 2013], and some screening studies carried out over the past two decades have failed to provide a survival benefit. At a genetic level, HGSOC is generally characterized by a high frequency of somatic DNA copy number variations. A recent study of the intra-tumor genetic heterogeneity in the HGSOC demonstrated that specific copy number alterations can occur in subclonal populations and lead to resistant metastases [Schwarz et al., 2015].

7.1.2 Study design and data

The public data set studied in [Schwarz et al., 2015] is composed of 177 samples at various time points and anatomic locations for 18 patients. Therefore, there are several samples for each patient. The DNA copy number profiles were obtained from Affymetrix Genome-Wide SNP6.0 arrays. The raw data are available under the GEO accession number GSE40546.

In this application, we focus on patient 8 of this study for which 11 samples are available. The patient was treated with neo-adjuvant and responded partially to the drug. Physicians observed a reduction of 24% of the primary tumor.

In this data set, no matched normal samples were publicly available. For this reason, we normalized the total DNA copy number signals by dividing the tumor samples of patient 8 by a weighted mean of the TCN signals from the other samples. The weights were chosen such that each patient has the same weight regardless of its number samples. Then DoH signals were estimated within each segment after a segmentation done on total copy number signals by the method described in Chapter 8.

7.1.3 Tumoral heterogeneity results

After performing the joint segmentation of the 11 tumor samples chromosome by chromosome, we have recovered 221 segments across the 22 autosomal chromosomes. We applied InCaSCN and we selected 4 latent profiles according to Fig. 7.1. With 4 latent profiles, the model explains the matrix of TCN signals at 98% .

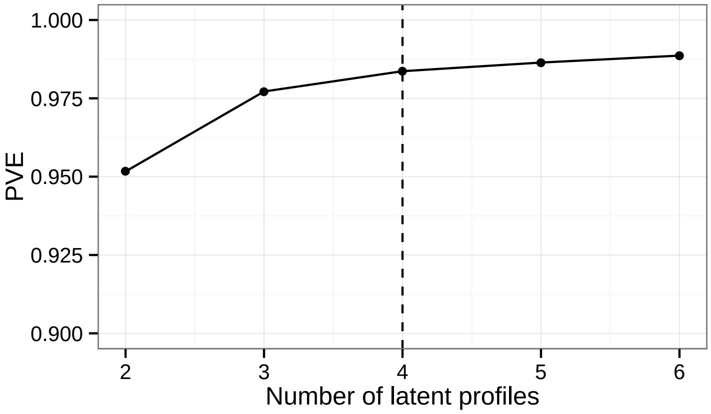


Figure 7.1 – PVE of InCaSCN model on a public the data set GSE40546

The results show that the InCaSCN model helps to highlight clones that can characterize clonal expansion. Indeed, we see in Fig. 7.2 that ascites and ovarian tissues have typical alterations and not systematically present in omemtum tissues.

The alterations in ascites may come from a resistant clone to the drug and which was initially in the primary tumor located in omemtum and ovarian tissues. The ovarian sample is mostly composed of two latent profiles, whose one is only present is exclusive to the ovarian tissue (latent profile 3). The latent profile 3 may be very specific to ovarian tissue and non-resistant to the drugs. Indeed, this latent profile is not present in relapses, same conclusions may be drawn for latent profiles 1 and 2 (Fig. 7.2), they are not present in the relapses either.

In conclusion, for this patient, there may exist only one resistant “clone “to the drugs that led to a relapse. The “clone” (latent profile 4) is characterized by large and high frequent amplifications across the whole genome. We can also note that there is a large part at beginning (before 15Mb) of chromosome 9 that is amplified following by a small cn-LOH region that is not present in the other latent profiles (Fig. 7.3). Therefore it may be interesting to explore if there are not known genes that can be responsible for the resistance.

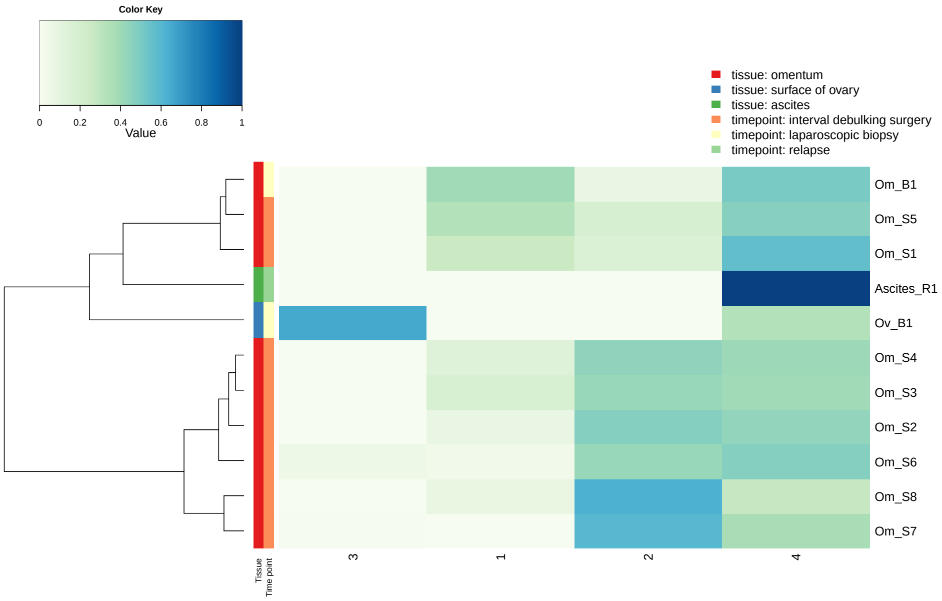


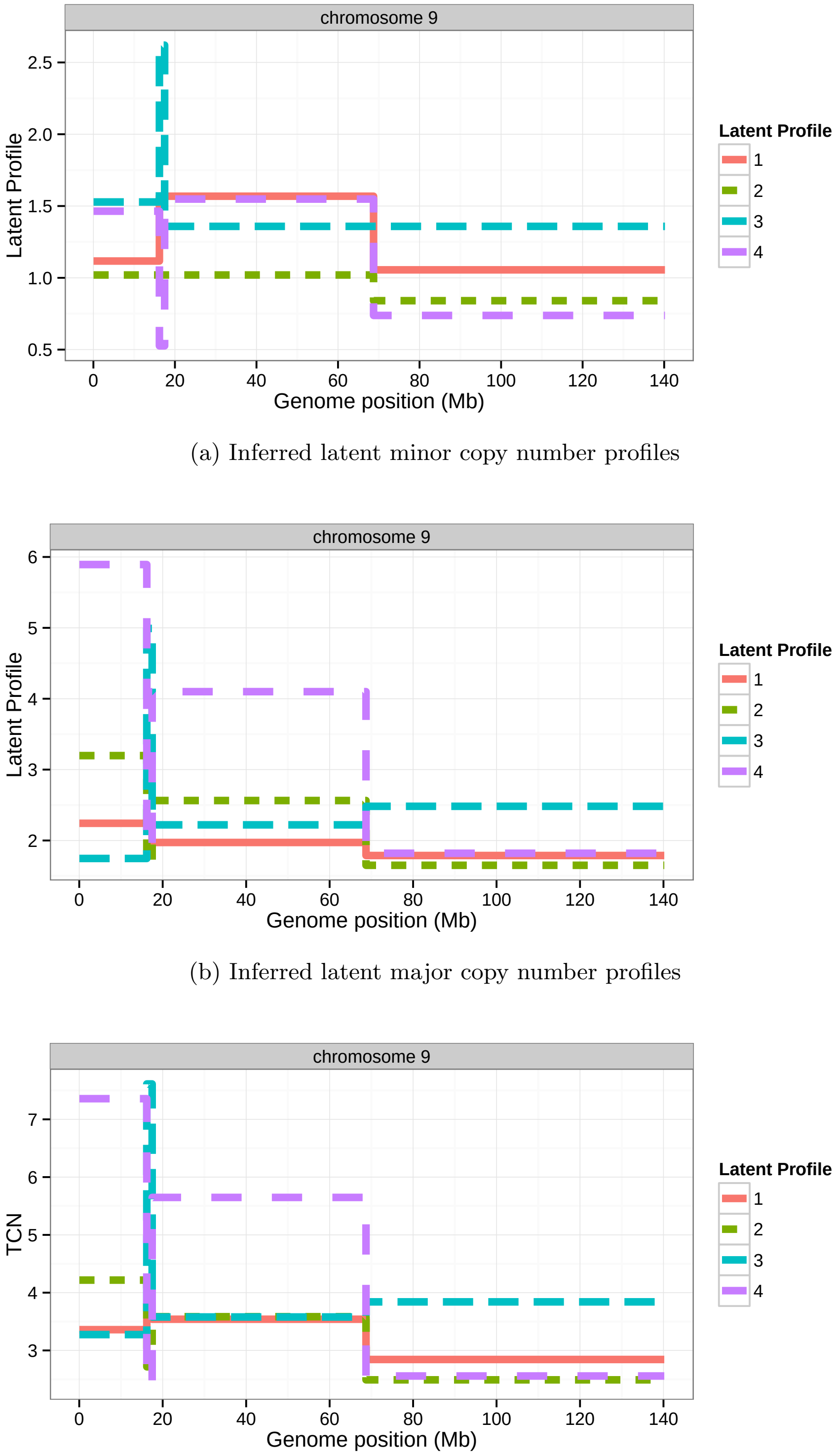
Figure 7.2 – Inferred weight matrix for patient 8 from dataset GSE40546

7.1.4 Short conclusion

The resulting latent profiles are difficult to interpret in terms of major and minor copy number in this example. This is probably due to the fact that jumps occur at the same location in the two dimensions (minor and major copy signals) that is biologically unlikely.

Another possible reason is the poor quality of the DoH estimations. As we do not dispose of the true genotypes, it was difficult to normalize BAF signal to obtain clean signals in terms of minor and major copy numbers. Therefore, the minor and major copy number signals could be not well estimated in some regions and jumps can occur in the original signals.

However, to conclude, the model provides a weight matrix that seems to be easy to interpret.



(a) Inferred latent minor copy number profiles

(b) Inferred latent major copy number profiles

(c) Inferred latent total copy number profiles

Figure 7.3 – Inferred latent profiles of chromosome 9 for patient 8 from dataset GSE40546 (top left:minor copy number, top right:major copy number and bottom:total copy number)

7.2 Collaboration with Institut Curie

This section summarizes a collaboration with the Fabien Reyal's team of Institut Curie started in 2014. The aim of this project is to better understand the different mechanisms of resistance of Triple negative breast cancer (TNBC) to classical therapies.

As TNBC is a very heterogeneous cancer, one important question to achieve this goal is highlight subgroups from DNA and RNA sequencing data. A similar work has recently performed in [Burstein et al., 2015]. After a brief introduction on the TNBC, we describe the study design and the WES data used. Then, we apply the InCaSCN model in order to characterize DNA copy number profiles of TNBC. First, we apply the model to TCN signals, and then to parental copy number signals.

7.2.1 Introduction to breast cancer

Nowadays, patients with a breast cancer can have a good chance to be cured. However, it is well known that there exist several breast cancers subtypes that require adapted treatments. The most common subtype is the Estrogen receptor (ER)-positive (80%). The drugs help to stop tumor cell growth by blocking estrogen hormone receptors. The most second frequent one is the Progesterone receptor (PR)-positive breast cancer, tumor cells grow in response to the progesterone hormone and the treatments help by blocking the associated receptors. The third most frequent one concerns patients with an amplification of the HER2 gene. This proto-oncogene helps to control how a healthy breast cell grows, divides and repairs. Patients who suffer from these three types of breast cancer have generally a good prognosis with the adapted hormonal therapies.

The rest of the breast cancers but the most aggressive cancer is the TNBC for which all receptors (ER, PR, and HER2 gene) are negative. It doesn't exist any specific and appropriate drugs to cure patients with this kind of cancer. Therefore, the survival prognosis of the patients suffering from TNBC is very poor. Previously, it was assumed that TNBC had specific features compared to the other breast cancers and that there existed a homogeneity within the patients suffering from TNBC. However, several studies have been conducted to analyze genomic profiles and it has been suggested that TNBC is, in fact, a heterogeneous disease and that it existed subgroups or subtypes. [Burstein et al., 2015] investigated 198 mRNA expression and DNA profiles of uncharacterized TNBC and concluded that TNBC may be divided into four stable subtypes by their expression profiles:

- luminal androgen receptor (AR; LAR),

- mesenchymal (MES),
- basal-like immunosuppressed (BLIS)
- basal- like immune-activated (BLIA)

In contrary, DNA profiles only produced two major groups (LAR and MES/BLIS/BLIA). These four subtypes may lead to various prognoses.

7.2.2 Study design and data

Design

Institut Curie selected 16 patients who have been affected by a TNBC. First, a micro-biopsy of the Primary tumor (PT) has been done at diagnosis, then all patients received 8 cycles of a combination of Anthracycline and Taxanes as neoadjuvant chemotherapy before surgery, where the Residuals (RES) of the tumor has been removed. In addition, for some patients, some tumor cells have migrated to the Lymph node (LN). Samples to study characteristics of TNBC have been collected at level of PT, RES and LN when it was possible. The residual samples are available for all patients, the PT sample of the patient 34 (due to a mistake in the labeling) is finally not available and 6 patients had metastases in the lymph node. Paired normal samples are available for only 12 patients (indeed, at the starting of the study, the paired normal samples were not available but they were one year after the tumor ones (2015), all reference samples are not available for a reason of non-consent or of the death of patients).

Available data

Institut Curie produced sequencing data from DNA and RNA samples in order to analyze features of the TNBC. Indeed, this enables to study simultaneously mutations, DNA copy number or gene expression profiles. In this study, both RNAseq data and WES data for each PT, RES and LN sample are available. We refer to section 1.3.2 for more details on this sequencing technology. We only deal with the WES data in the following.

The samples have been sequenced with Illumina Hi-Seq 2500 pair-end technology aligned on the reference genome hg19 with the BWA algorithm [Li and Durbin, 2009]. The sequencing depth was 100x for all the tumor samples and 30x for the paired normal ones with 100bp-length reads (short DNA sequences of length 100 bp). This means that on average, there are 100 reads that are aligned to each targeted position.

Institut Curie supplied us with 38 `bam` files for the tumor samples and 12 `bam` files for the normal samples. The `bam` files are the binary format of `sam` files. `sam` are TAB-delimited text formats which are the outputs from aligners that read FASTQ files and assign the sequences to a position with respect to a known reference genome (here hg19).

Our goal in this study was to take into account tumor heterogeneity in order to discover common DNA copy alterations that may explain drug resistance. Estimating DNA copy number from WES data without paired normal was an issue that few bioinformatic tools address. We explain in detail this part in Chapter 9. However, we succeeded to estimate DNA copy number signals for each sample (even for those for which the paired references were not available) but also DoH signals. This enabled us to compute parental copy number signal estimates.

7.2.3 Heterogeneity exploration on total copy number

We attempted to discover common alterations between patients of this study. In this section, we applied our model proposed in Chapter 6 on the TCN since CNVkit algorithm has provided quite good estimations of total DNA copy number profiles (Chapter 9).

We segment all 38 available samples jointly using RBS+DP algorithm described in Chapter 2. Then, we applied the InCaSCN model proposed model in Chapter 6 to infer heterogeneity for 3 values of λ (the penalty coefficient for the latent copy number profiles) and for a number of latent profiles varying between 5 and 19. For each value of p, the model provides a PVE (Percentage of variation explained, see section 6.3.4 for details) larger than 90%. If the poor result of PVE_{14} is omitted, it seems that the plateau of the PVE curve starts at $p = 13$ and in addition, the PVE is larger that 95%, therefore we select the model with $p = 13$ latent profiles (Fig. 7.4).

The results at a level of weight matrix are presented in Fig. 7.5. We can note that the samples are grouped by patients. Patients 36 and 56 are clearly grouped by their triplet contrary to patient 40 where the residual sample is far from LN and PT. This may be due to a new mutation that occurred after the initial treatment. It is well known that neo-adjuvant treatments have more difficulty to reach the lymph nodes, and as a result metastases in lymph nodes is likely closer to PT sample than to RES sample. For patient 7, PT sample which is not grouped with LN and RES. However, all 3 samples are in the same branch of the tree and there are no significantly supplementary latent profiles in the LN and RES. Then, it seems that no major event occurred between before and after the drug intake for the copy number perspective.

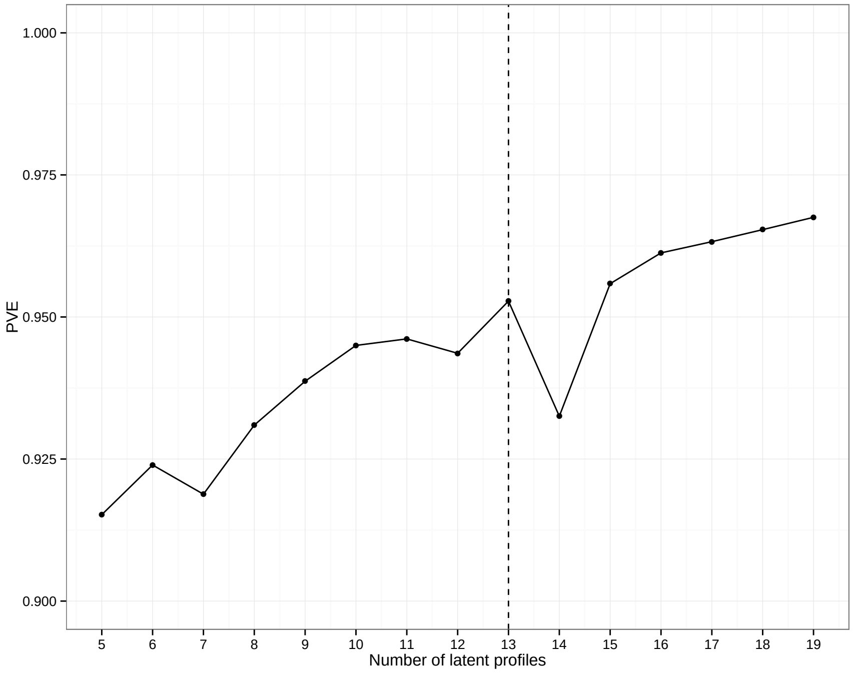


Figure 7.4 – PVE results of InCaSCN method on WES data from Institut Curie

However, latent profiles 2, 7 and 13 are common to several patients. Hence, it could be interesting to further study these latent profiles. Note that the PT sample of the patient 34 is completely isolated. As explained in the table, in fact, PT34 does not belong to the patient 34. If we look at the correlation between latent profiles 2 and 7 it appears that, in fact, they are very similar. Alterations in these latent profiles are probably common to the whole dataset. This was also the conclusion of authors in article [Burstein et al., 2015] for instance for CDKN2A gene, a tumor suppressor gene, which is deleted. This gene is between 21,97 and 22 Mb on the chromosome 9, and latent profile 13 is well characterized by a large region that has been lost (Fig.7.6) .

To summarize, two groups appear clearly after the clustering on the weight matrix, where the first group is characterized by the latent profile 13. Then, it appears that the groups are done at the patient level. Some latent profiles reach high values. However, these extreme values are probably compensated by low weights. Estimations of TCN from WES data likely contain high extreme values despite filtering.

This was the first analysis that has been conducted on this data set. Then, it has been possible to compute parental copy numbers thanks to the method described in Chapter 9. We are going to present the results of InCaS-CN model on the parental

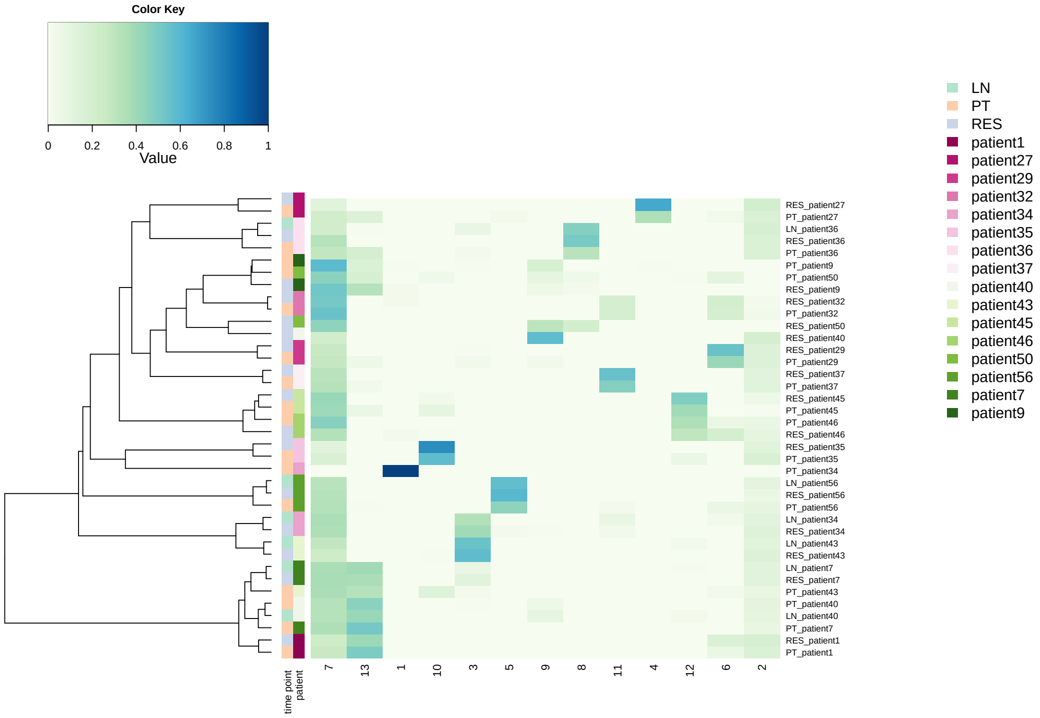


Figure 7.5 – Heatmap of weight matrix from InCaSCN method on WES data from Institut Curie

copy number signals in the next section.

7.2.4 Heterogeneity exploration on parental copy numbers

By means of the `sequenza` tool, it has been possible to compute the DoH and therefore the parental copy number signals (see Chapter 9). The DoH signals have been computed for the 12 patients with paired normal samples. The resolution scale of DoH was lower than the TCN. We perform the joint segmentation step on the TCN signals and we compute the DoH and the parental copy number signals at a segment level. We applied the heterogeneity model proposed in the previous Chapter to the parental copy number signals of a sub-data set. As for the TCN model, we selected 13 latent profiles according to the Fig. 7.7. The selected model reaches a PVE of 88% in this case.

At a first look of the inferred weight matrix, it appears that there is only one latent profile common across the patients (Fig.7.8). In addition, as for the model on the TCN, the patients are grouped together. In other terms, PT, RES, and LN (when it is available) samples of same patients are grouped in the same cluster. This is the case for

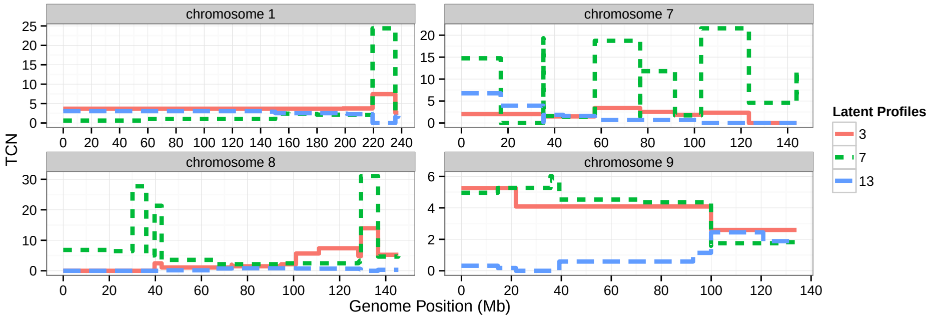


Figure 7.6 – Latent profiles extract from InCaSCN method on WES data from Institut Curie.

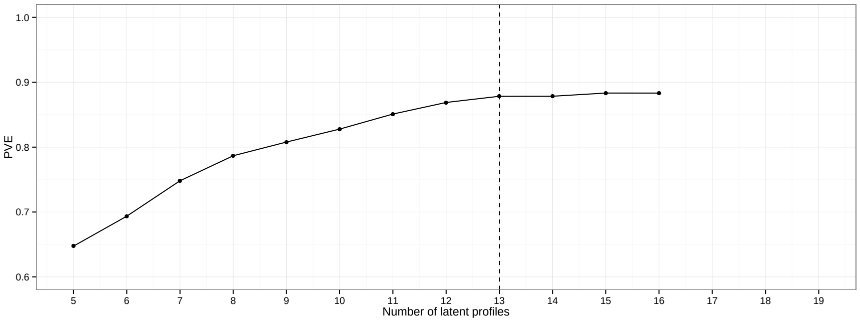


Figure 7.7 – PVE results of InCaSCN method on WES data from Institut Curie

the patients 56 and 36 for instance. PT and RES are grouped in patients 32, 43, 35, 29, 1 and 27 . As expected, PT of patient 34 does not share latent profiles with the others. In the same way, as the conclusion from the model on TCN, PT and LN samples of patient 40 are not grouped with its respective RES sample. Therefore, RES_{40} is may be composed of one clone from PT that was resistant to the drugs. In addition, it is clear that the proportion of the resistant latent profile increases in RES.

PT and RES of the patient 50 are not grouped but share a common latent profile. A similar conclusion than for the patient 40 can be drawn. For these two patients, it seems that the resistant clone is already present in PT and becomes largely predominant in RES.

To summarize these conclusions, we highlight the fact that DNA copy number profiles of TNBC data set are really heterogeneous and there are a few alterations common between patients.

By looking at some latent profiles (Fig.7.9), it appears that each has its own characteristics in terms of copy number variations.

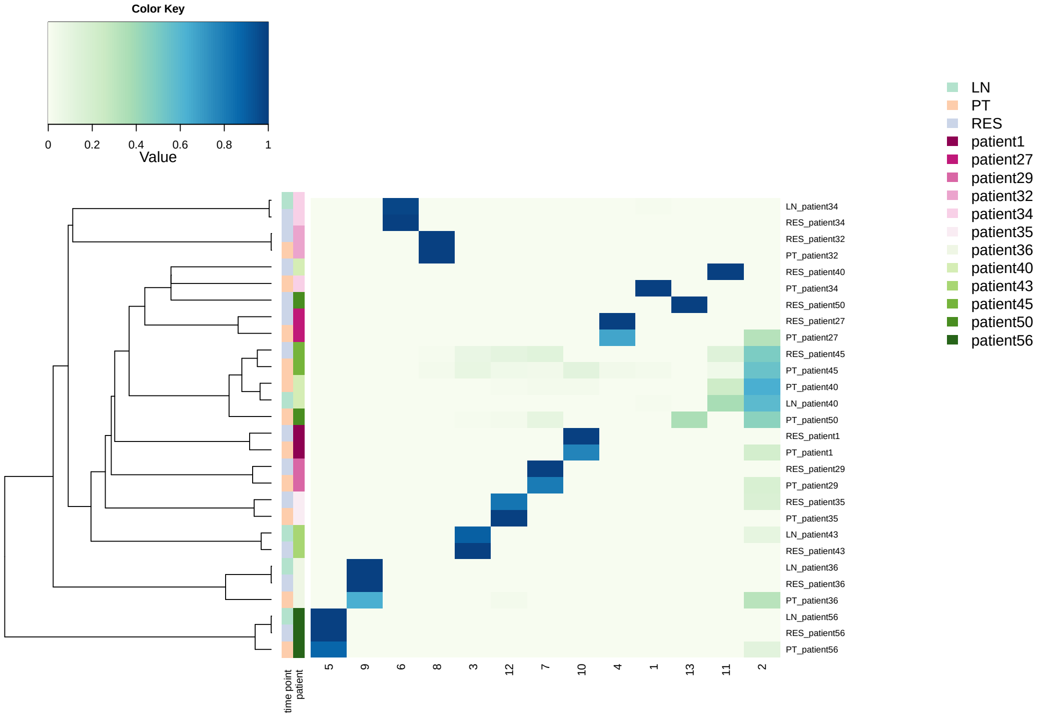


Figure 7.8 – Heatmap of weight matrix from InCaSCN method on WES data from Institut Curie run on parental copy numbers

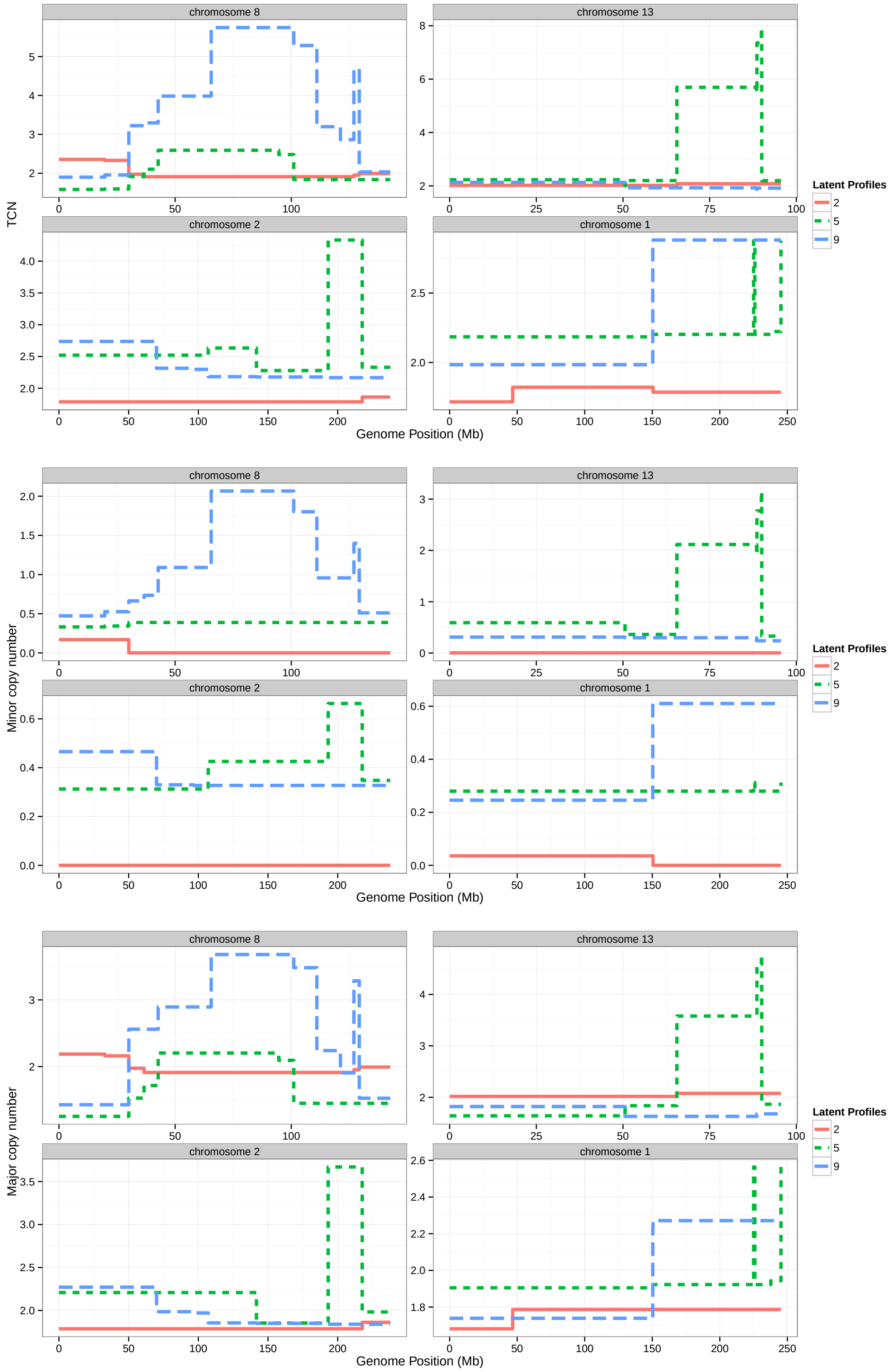


Figure 7.9 – Latent profiles extracted from InCaSCN method on WES data from Institut Curie (TCN on top, C1 on the middle and C2 on the bottom)

7.3 Conclusion and perspectives

For the first application, in addition to a similar phylogenetic tree to those shown in [Schwarz et al., 2015], our model is able to detect a resistant latent profile that was already present in the primary tumor. The exploration of these latent profiles could provide new genetic biomarker targets and help to develop original adapted drugs.

This chapter also summarizes a work done in collaboration with Institut Curie. This collaboration enabled to apply our developed model on a completely new raw data set. However, it is difficult to draw relevant biological conclusions at this stage. Although some patients were grouped after clustering, it appears that TNBC remains a heterogeneous disease for which it is difficult to highlight relevant biomarkers. Similar conclusions have been drawn from the analysis of the RNAseq data performed by Benjamin Sadacca.

Therefore, a key point of this Chapter is that we were able to apply the InCaSCN model on two different kinds of data sets. First, data from two technologies have been used (microarray and sequencing). Then, the model has been used to attempt to explain both intra and inter-tumoral heterogeneity. It could be interesting for the second one type of heterogeneity to apply the model to a study with a similar design than those of Institut Curie but on a less heterogeneous cancer type.

For both real data sets, we discovered that the inferred parental copy numbers present breakpoints simultaneously in minor and major copy number signals. However, such phenomenon is unlikely because it means that two major alterations occur at the same time. As a result, as mentioned in the Chapter 6, some modifications like the penalties that control the level of sparsity of the latent profiles could be added. We could also force jumps of latent profiles to occur only on the minor of major copy number. This phenomenon was not observed in the simulation study and may be that these results are due to the approximation on the estimations of the DoH in both of cases or to a poor segmentation.

We are currently writing an article that presents InCaSCN model and an application to a real data set.

Part III

Bioinformatic pipelines

Table of Contents

Chapter 8

Estimation of DoH in absence of normal reference

In absence of the knowledge of germline genotype of each locus, it is not possible to compute the DoH. Indeed, DoH is only computed on the heterozygous SNPs. Therefore, without DoH estimations, it is impossible to estimate the parental copy number signals and to use InCaSCN model described in Chapter 6 to infer tumoral heterogeneity.

It is not uncommon, in cancer research, that the normal blood samples are not available because of non-consent of patients to sequence or genotype their genomes or because normal tissue samples were not taken at the time of the study. For instance in the case of retrospective or tumor cell lines studies.

It was the case of [Schwarz et al., 2015] data set used in Chapter 7, where the data set was only composed of tumor samples. In order to compensate the absence of normal samples and therefore to not lost information from SNP arrays, we developed a method to estimate the DoH by regions and then the parental copy number signals. Indeed, the InCaSCN model only requires segment-level PSCN (parent specific copy number) estimates and not necessarily at each locus.

This chapter is divided into four sections. The first one describes how to estimate the DoH at a segment-level and the assumptions that are required. The second one deals with the detection of the normal region that is one of the main issues in the estimation of DoH in absence of a normal reference. Then, we attempt to check whether the assumptions of the model are realistic and whether the estimations are close to the reality. Some bias issues have been raised at the end of the section and we attempt to correct them in a next section. The last section is about experiments on another data set from another technology.

The work done in of this Chapter is in progress and several issues have not been yet fixed.

8.1 Estimation of DoH

We would like to estimate the DoH at a segment-level. Then, in the following, a segment is denoted by s and we denote by $n_s = Card(s)$ the number of SNPs in segment s, n_s^{AA}, n_s^{AB} and n_s^{BB} are the number of SNPs AA, AB and BB in segment s respectively. Note that, $n_s = n_s^{AA} + n_s^{BB} + n_s^{AB}$.

Then, we define the quantity δ_j for either heterozygous and homozygous SNPs j by:

$$\overline{\delta}_j = 2 \times |b_j - 1/2| \tag{8.1}$$

where b_j is the BAF for loci $j = 1, \ldots, J$.

$$\overline{\delta}_s = \pi_s^{AA}\overline{\delta}_s^{AA} + \pi_s^{BB}\overline{\delta}_s^{BB} + \pi_s^{AB}\overline{\delta}_s^{AB} = \frac{1}{n_s}\sum_{j \in s} \delta_j \tag{8.2}$$

$$\overline{\delta}_s^{AB} = \frac{\overline{\delta}_s - \pi_s^{AA}\overline{\delta}_s^{AA} - \pi_s^{BB}\overline{\delta}_s^{BB}}{\pi_s^{AB}} \tag{8.3}$$

where :

- $\pi_s^{AA} = \frac{n_s^{AA}}{n_s}$,
- $\pi_s^{BB} = \frac{n_s^{BB}}{n_s}$
- and $\pi_s^{AB} = \frac{n_s^{AB}}{n_s}$

are respectively the proportion of SNPs AA, BB and AB in the segment s for $s = 1, \ldots S$, and

- $\overline{\delta}_s^{AA} = \frac{1}{n_s^{AA}} \sum_{j \in s} \delta_j^{AA}$,
- $\overline{\delta}_s^{BB} = \frac{1}{n_s^{BB}} \sum_{j \in s} \delta_j^{BB}$

- and $\overline{\delta}_s^{AB} = \frac{1}{n_s^{AB}} \sum_{j \in s} \delta_j^{AB}$

are respectively the mean of δ for SNPs AA, BB and AB in the segment s. Note that δ^{AB} is in fact the DoH.

We assume that the proportion of the homozygous (π^{AA} and π^{BB}), and heterozygous SNPs (π_s^{AB}) are the same regardless of the type of region (gain, loss, normal,..). Then, we can estimate the proportion of homozygous SNPs by detecting a normal region and selecting SNPs with a BAF larger or lower than given threshold values. These two thresholds are dependent on the technology or even of the sample itself. The normal regions have the particularity that BAF signals have only three modes that are centered at 1 (BB SNPs), 0 (AA SNPs) and 1/2 (AB SNPs). The Fig. 8.1 shows the raw BAF signal of the annotated data set from `acnr` package (Affymetrix technology). The first region is a normal one where it appears that the number of modes is three with the values quoted previously. The main difficulty is to detect these kinds of regions, we present a way to detect these regions in section 8.2.

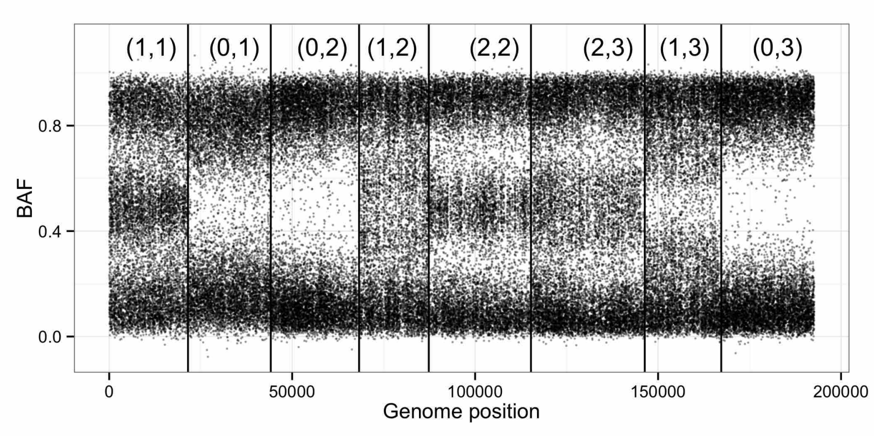


Figure 8.1 – BAF signal

Then, once we know a normal region, we are able to estimate the mean of $\overline{\delta}_N^{AA}$ and $\overline{\delta}_N^{BB}$ for a normal region. Assuming that the distribution of δ for homozygous SNPs is constant along the genome, we replace $\overline{\delta}_s^{AA}$ and $\overline{\delta}_s^{BB}$ in (8.3) respectively by their estimations in the normal region ($\overline{\delta}_N^{AA}$ and $\overline{\delta}_N^{BB}$).

$$\hat{\delta}_s^{AB} = \frac{\overline{\delta}_s - \pi_N^{AA}\overline{\delta}_N^{AA} - \pi_s^{BB}\overline{\delta}_N^{BB}}{1 - (\pi_N^{AA} + \pi_N^{BB})} \tag{8.4}$$

Finally, the parental copy number estimations are equal to:

$$\begin{cases} y_{s1} = \bar{c}_s \times \left(1 - \hat{\delta}_s^{AB}\right)/2 \\ \\ y_{s2} = \bar{c}_s \times \left(1 + \hat{\delta}_s^{AB}\right)/2 \end{cases} \tag{8.5}$$

where $\bar{c}_s$ is the average of the TCN in the segment s.

To summarize, we have made three assumptions to estimate DoH that are:

A1 No breakpoints in the segments

A2 Proportion of the heterozygous and homozygous SNPs is the same along the genome (independent of the region).

A3 The distribution of the homozygous SNPs is the same regardless of the region.

8.2 Detection of a normal region after the segmentation

The estimation of DoH that is $\hat{\delta}_s^{AB}$ by segment can be done with the formula 8.4 provided that we are able to identify a normal region. The two essential points are recovered the right segmentation in the signal but also a normal region in order to compute the DoH densities of the homozygous SNPs. To find the normal region after segmentation, two choices are possible:

- the first one is the annotation by the user
- the second one is an automated detection of the normal region in the data set.

We present in this section the second proposed option.

After performing the segmentation on CN for example, it is possible to estimate in each segment the distribution of the BAF. Distribution in each segment may be considered as a mixture model parameterized by the mean, the variance of each normal distribution and their respective proportion (Fig. 8.1).

However, the germline SNPs AA, BA, AB, and BB produce different kind of mixture model within an altered region. For instance, within a loss region, only two modes are observed: for distribution for SNPs A and B because of the loss of one parental copy. A huge feature of the BAF signal in a normal region is that the mixture model has only three components: modes for the SNPs AA, BB and only one mode for the SNPs AB and BA. Therefore, a trick to find a normal region is to look at which segments have a number of modes equal to three and especially where the proportions are balanced between the three modes i.e close to 1/3 for each component. However, in the case of a balanced gain (2,2) regions the BAF, the distribution is the same than in the normal regions. Therefore, a second step consists of selecting the segment with the smallest mean of CN. The pipeline of this selection is presented on algorithm 6. After the segmentation step, for each segment, we perform a kmeans algorithm with euclidean distance for a number of clusters between 2 and 4. Then, we select the best number of cluster for this segment. We repeat these steps for each segment of the BAF signal. Finally, only the segment with three modes are selected and the normal region is finally those with the smallest mean of CN.

```
Data: Segmented BAF data;
Parameters: Number of segment $S$;
Result: Distribution for each segment $s$: number of modes, partition for each
        loci in segment and mean of CN;
for $s \leftarrow 1, \dots S$ do
    Use Kmeans method for segment $s$ with euclidean distance ;
    Choose the best number of clusters;
    Partition between clusters for each loci in $s$;
    Compute the proportion for each component of the distribution;
    Compute the mean of the CN within segment $s$;
```

Algorithm 6: Estimation of the distribution within each segment.

In this way, we illustrate the result of the algorithm on raw Affymetrix data on Fig. 8.2. For the instance, here, only three segments can be labeled as a normal region (1,1), (2,2) and (2,3) but by checking the CN values within segments, the regions (2,2) and (2,3) are naturally excluded.

The topic of the next section is to check if these assumptions can be considered as true.

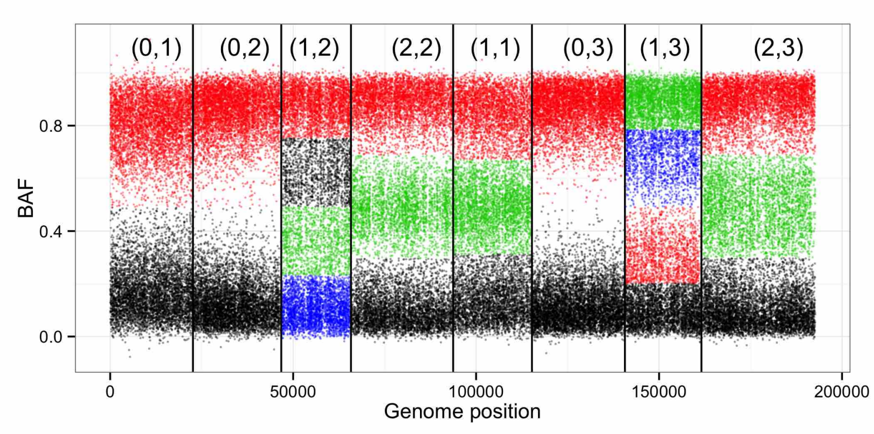


Figure 8.2 – BAF signal (coloured by clustering)

8.3 Checking the assumptions

In order to explore and validate the assumptions done in the section 8.1, we used the annotated data set described in Chapter 3 where homozygous SNPs and the various type of altered regions are known. The illustrations of this section are specific to the Affymetrix technology and could be completely different for Illumina and Agilent technologies. The results of this section have been done on the Affymetrix data set from GSE29172. This work is currently in development and we plan to explore the assumptions on other data sets of `acnr` package.

8.3.1 Assumption A1

The first assumption is that the distributions of the BAF and TCN are constant within the segments. To be sure that it is the case, a solution is to oversegment the signals. We have seen in the previous part of this thesis that there exist a lot of various methods to segments these data. Even if we do not know the true number of breakpoints, a solution is to oversegment the signal by reasonably assuming that the true ones will be in the list.

8.3.2 Assumption A2

We first plot the proportion of SNPs AA, BB and AB for each type of regions (Fig. 8.3). By looking at Fig. 8.3, assuming that the proportions are the same along the genome seems to make sense.

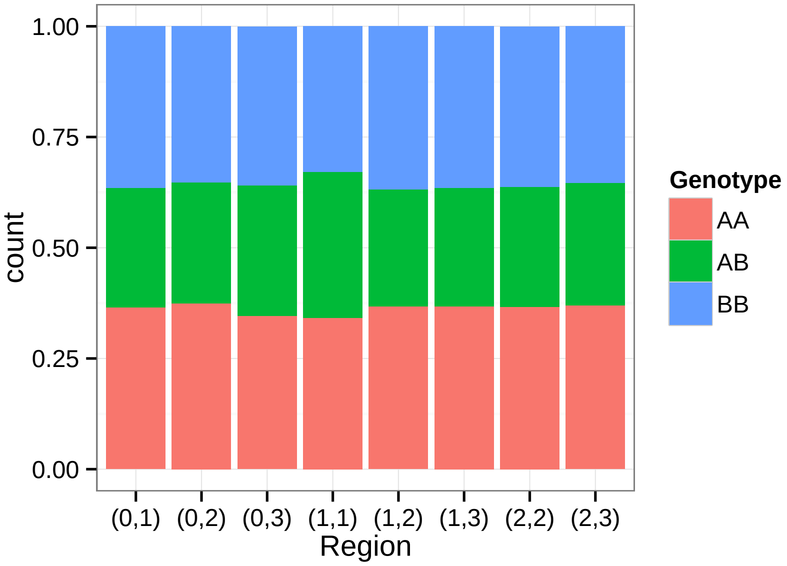


Figure 8.3 – Proportion of genotypes in each type of region

To confirm this first observation that the proportions of the homozygous are the same between the region, we perform a chi-square test between the genotype variable for homozygous status and the altered region. We conclude that the two variables are independent with a p-value equal to 0.30.

	(0,1)	(0,2)	(0,3)	(1,1)	(1,2)	(1,3)	(2,2)	(2,3)
AA	2305	2472	2399	2267	2418	2521	2177	2468
BB	2302	2331	2493	2178	2421	2506	2154	2357

Table 8.1 – Number of loci per genotype in each altered region

To summarize, we can estimate the proportions π^{AA} and π^{BB} by the proportions estimated in the normal region. The third assumption that we have made is that the distributions of δ^{AA} and δ^{BB} are the same along the genome. This is the subject of the next section.

8.3.3 Assumption A3

In the section 8.1, we have assumed that the density, but especially the means of δ for the homozygous SNPs are the same across the different types of altered regions. If it is not the case, the estimation of DoH and then of the parental copy numbers may be biased.

In the annotated data set presented in Chapter 3, it is easy to extract for each region the BAF profiles and only keep the homozygous SNPs. Then, it is possible to plot the density of the BAF for each type of region. We also plot the density of DoH in order to check if the mirroring usually used by [Staaf et al., 2008] doesn't bias the estimations. Although the distributions of the BAF and δ are similar overall, it is clear that the density of homozygous SNPs is not identical across the different regions and it is more obvious for the loss (0,1) where SNPs AA and BB have a BAF further of 0 and 1 respectively (Fig. 8.4 and 8.5).

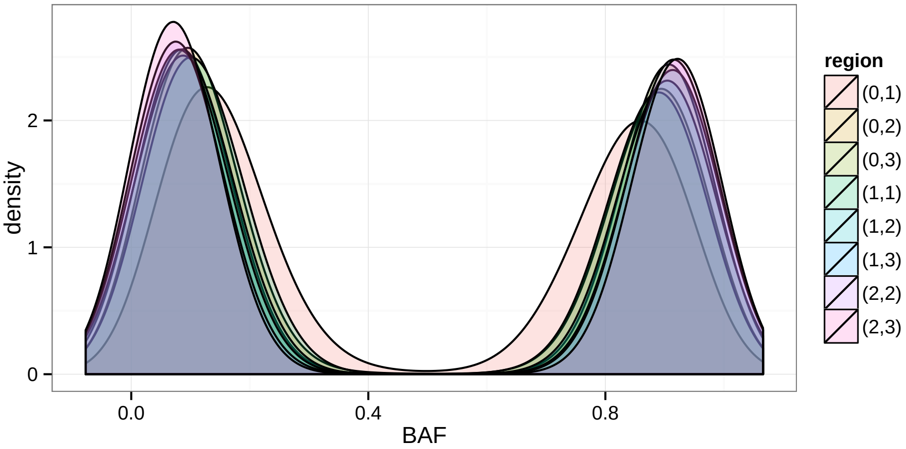


Figure 8.4 – Density of BAF for homozygous SNPs by annotated regions

In addition, it seems that the DoH and TCN values increase at the same time. To explore this new assumption, we plot the observed values of DoH of the homozygous SNPs against the observed values of TCN within each region (Fig. 8.6). The DoH ranges between 0.65 for the loss region (0,1) and 0.80 for the gain region (2,3) i.e. one gain of one parental copy and two gains of the other one. In this way by replacing δ^{AA} and δ^{BB} of all regions by δ_N^{AA} and δ_N^{BB} respectively, in formula (8.3) is likely to produce a bias in estimations.

We proposed a mean to correct the produced bias in the section 8.4.

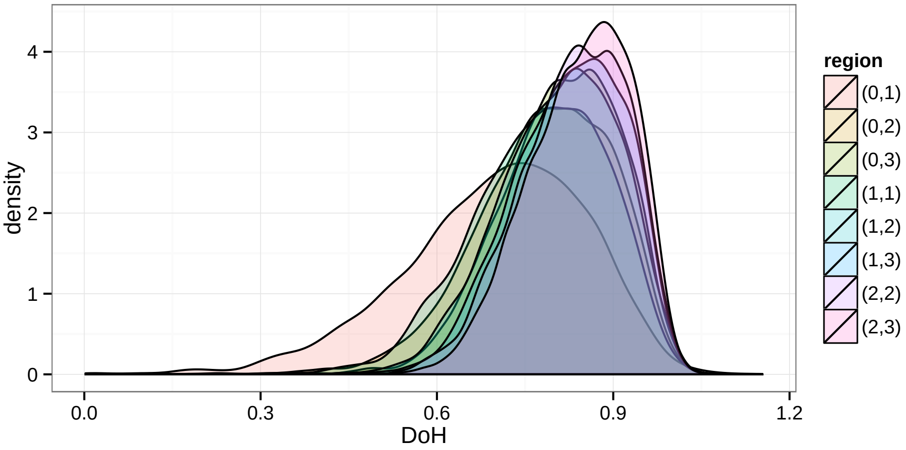


Figure 8.5 – Density of δ^{Hom} by annotated regions

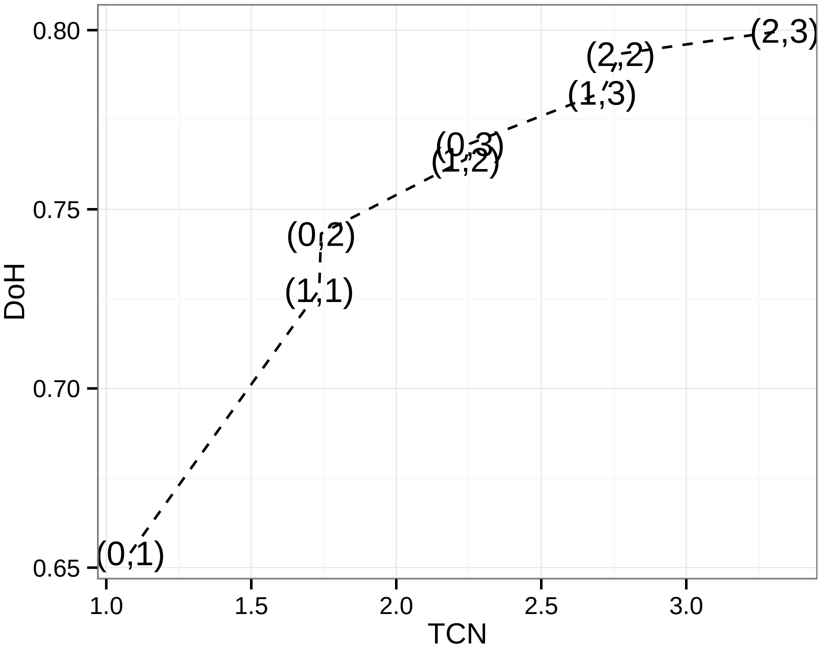


Figure 8.6 – Link between δ^{Hom} of homozygous SNP and CN

8.4 Tentative to correct the bias

We see in the section 8.3.3, that replacing δ^{AA} and δ^{BB} by their estimations in the normal region can produce a bias. However, it could be possible to perform a calibration that is completely technology dependent in order to reduce bias in estimations of the DoH.

We have seen that δ^{Hom} is dependent to the values of TCN (Fig. 8.6). We denote by θ_j^A and θ_j^B the quantities of allele A and B at locus j respectively. Then b_j the B allele fraction can be written:

$$b_j = \frac{\theta_j^B}{\theta_j^B + \theta_j^A},$$

The B allele fraction signal for SNPs AA in a region s is close to 0 and therefore $\theta_s^B \sim \epsilon$ with ϵ to be estimated. Therefore $\theta_s^A \sim c_s$ and then,

$$\begin{aligned}
\delta_s^{AA} &= 2 \times \left| \frac{\epsilon}{\epsilon + c_s} - \frac{1}{2} \right| \\
&= 2 \times \left| \frac{2\epsilon - \epsilon - c_s}{2(\epsilon + c_s)} \right| \\
&= \left| \frac{2\epsilon - \epsilon - c_s}{\epsilon + c_s} \right| \\
&= \left| \frac{\epsilon - c_s}{\epsilon + c_s} \right| \\
&= \frac{c_s - \epsilon}{c_s + \epsilon}
\end{aligned}$$

The same equations can be written for the SNPs BB, the B allele fraction signal for SNPs BB in a region s is close to 1 and therefore $\beta_s \sim \epsilon$ with ϵ to be estimated. Generally we can write δ_s^{Hom} for homozygous SNPs by:

$$\delta_s^{Hom} = \frac{c_s - \epsilon}{c_s + \epsilon} \tag{8.6}$$

where c_s is the average of the TCN in the region s.

The last step is to calibrate the value of ϵ. In our case, we try to tune ϵ to get the best fit to the curve presented in Fig. 8.6 and the curve defined by Eq. 8.6. For this data set, the best ϵ is around 0.3. To estimate $\hat{\delta}^{AB}$ we used δ_s^{Hom} (Eq. 8.6).

Given that, we knew the labels of regions it is easy to know the breakpoints in the signal. Then, we plot the DoH estimations computed within each region and compare it with the truth when genotypes are known. As the breakpoints are known, it is obvious that the TCN is not at all biased. The red circles are $\hat{\delta}_s^{AB}$ for each region by replacing the δ^{AA} and δ^{BB} by their estimations in the normal region (Eq. 8.4) . Compare to the truth (blue squares), it is clear that these estimations are really biased. Then, we plot the second version of the estimation of $\hat{\delta}^{AB}$ (green triangles). Estimations produced by the second version is better than the first one. Indeed, estimations are closer to the truth but in addition, the LOH regions are aligned at the same value on the y-axis. To conclude, we have been able to produce an estimation of the DoH relatively few biased and takes the advantage to avoid the detection of a normal region in the sample. However, these results may be entirely dependent on the technology (Affymetrix) and of the data set. Therefore the transformation used to drop the bias may not work in other cases.

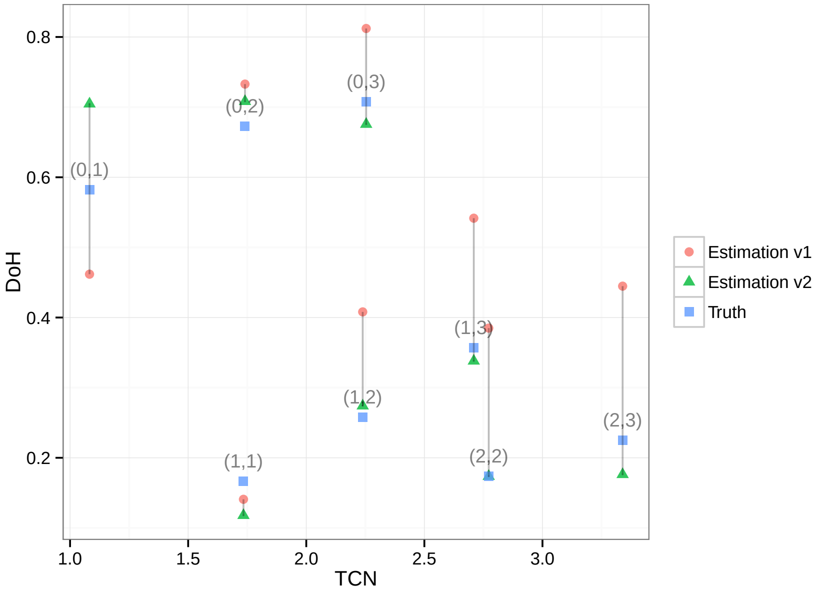


Figure 8.7 – DoH along TCN (red disks are the estimation after segmentation and blue triangle are the observed values).

8.5 Experiments on Illumina data set

In `acnr` package, we annotated in total three data sets. Among these data sets, one is from the technology Illumina. Illumina pre-processing pipeline enables to produce very clean BAF signals (Fig. 8.8). Therefore, the detection of a normal region is easy compared to data from Affymetrix technology that is noisier. Here, the clustering step on BAF to detect the normal region performs well and we recover the region annotated as normal.

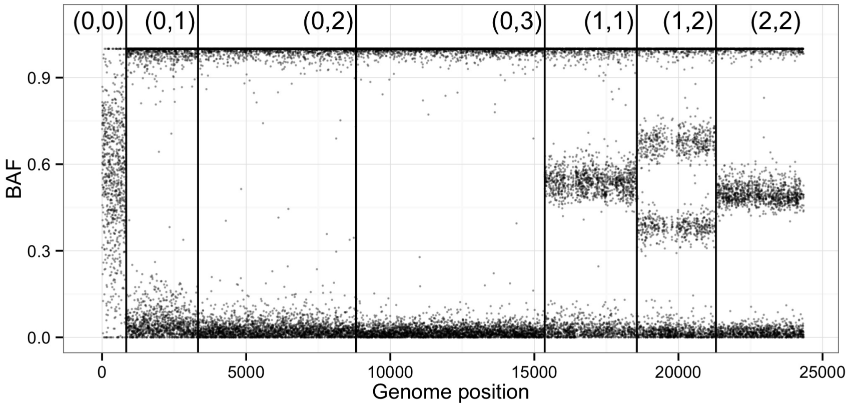


Figure 8.8 – Example SNP array data (Illumina technology)

As previously, we ask if the same link exists between δ^{Hom} and the TCN for this data set than those observed for Affymetrix technology. Except for the homozygous deletion (loss of both parental copies) region, the variation of δ^{Hom} in function of the c is minor compared to those observed previously for Affymetrix (Fig. 8.9). We fit the model described in 8.6 with $\epsilon = 0.04$.

Estimations of DoH are presented in Fig. 8.10 where v1 is done by using 8.4 and v2 is done by using 8.6. Except for the homozygous deletion, there is no notable variation between the estimation v1 and the estimation v2.

In conclusion, the estimation of the DoH on Illumina technology is useful to estimate DoH in LOH regions. Indeed, for the LOH regions, because it is hard to split the homozygous SNPs from the heterozygous with a threshold, the pipeline that we proposed can help.

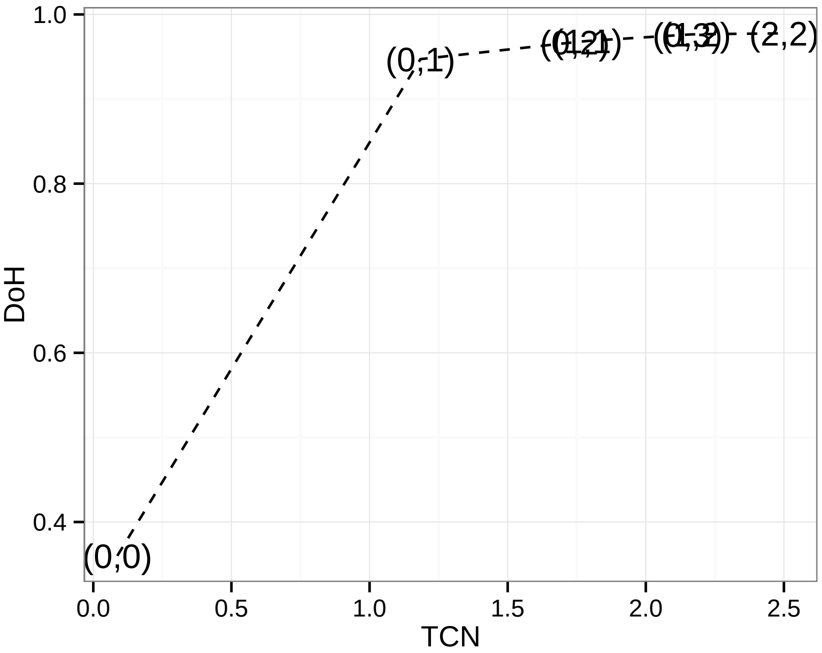


Figure 8.9 – Link between δ^{Hom} SNP and CN (Illumina technology)

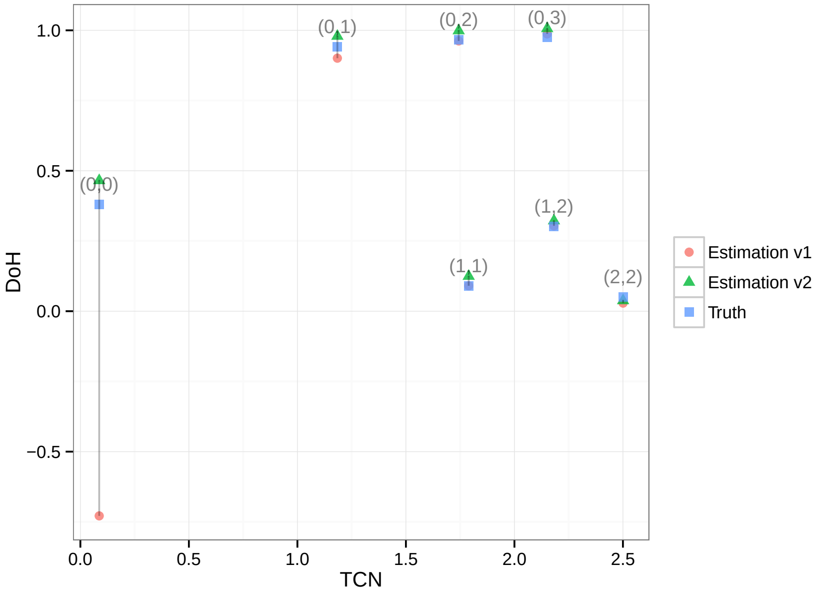


Figure 8.10 – DoH along TCN (red disks are the estimation after segmentation and blue triangle are the observed values).

8.6 Conclusion

In this chapter, we presented a way to estimate DoH and therefore parental copy number signals when no paired normal is available at a segment level. We have seen that some biases exist in the estimations and we attempted to drop them. However, as we said, the trick used for this data set is likely to fail on other ones. To solve this problem, further experiments will be conducted on several annotated data sets from other technologies. The key point is the tuning of the parameter ϵ that can be very different between the data sets and the technology.

The clustering step uses to detect the normal region and especially the choice of the number of clusters is also ad-hoc to the data. The threshold to define the homozygous SNPs in the normal region is also dependent on technology and one sample. A solution to this problem could be to include an annotation step done by the user. In fact, the user could define the normal region by himself by defining positions on the genome. This step requires a visualization step by a fast and user-friendly interface. This kind of interface is currently in development in the lab.

In this way, it is clear that enhancements need to be added to this pipeline in order to estimate properly the DoH signals at a segment level. Then, we may propose this pipeline to the bioinformatic community as an `R` package form for instance.

Chapter 9

Pipeline to estimate DNA copy number from sequencing data

9.1 Introduction

The aim of this chapter is to present tools to estimate TCN and DoH from WES data. It was an issue raised when we had to deal with the data from Institut Curie. A brief review of the literature to compare tools implemented to estimate copy number from WES data was performed by an intern Jimmy Carrillo. At the end of the two-month internship, he provided a pipeline in python in order to estimate both the TCN and the DoH from WES data. This pipeline allows us to quickly get data under the same form than microarrays. With this tool, it has been possible to apply the heterogeneity model on WES data from Institut Curie. In this chapter, we present some results on the estimation of TCN and DoH.

9.2 Estimation of Copy number and B allele fraction

Many tools have been developed to estimate copy number data from WGS or WES technologies. Codex [Jiang et al., 2015], ExomeCNV [Sathirapongsasuti et al., 2011], CoNIFER [Krumm et al., 2012], VarScan 2 [Koboldt et al., 2012], EXCAVATOR [Magi et al., 2013], VEGAWES [Anjum et al., 2015], CONVEX [Amarasinghe et al., 2013], Control-FREEC [Boeva et al., 2012] and CNVkit [Talevich et al., 2016]. Recently [Kadalayil et al., 2014] has performed a review of existing method to analysis copy number data from WES. However, most of these tools require a normal-paired sample in order to normalize properly copy number data. Contrary to the analyze of the total DNA copy number, only a few tools have been developed to estimate BAF from WES data [Favero et al., 2015].

9.2.1 Copy number

We performed the copy number estimation with the CNVkit algorithm. This algorithm takes into account all reads that are aligned out of the targets (here exons) but also enables to build a pooled of normal samples in order to get a unique copy number reference sample. Pooled normal samples can be very useful in our case because only 12 normal paired samples have been available for a total of 16 patients. In consequences, for each patient, we can normalize their tumor sample and get an estimation of the total copy number. Another advantage of the pooling is the decreasing of the noise that is probably due to the depth of sequencing (30x) for the normal sample. In addition, if we analyze all the tumor samples simultaneously it is important to have the same normal reference for all samples.

Figure 9.1 shows copy number estimations for PT, RES and Normal samples for the first patient and the chromosome 1. This figure highlights that for patient 1 (top of the figure) the residual and primary tumor are similar in term of alterations even if alterations are more obvious in the residual, it could mean that it was only one resistant clone to the drug in the primary tumor and that PT contained more normal cells in the sample. The same phenomenon is observed for the samples of patient 36 (bottom of the figure), that are globally similar to PT, RES, and LN.

The results that have obtained suggest that patients have potentially copy number profiles completely different and there are no common alterations between profiles. By looking at the normal profiles, the figure suggests that the normalization is enough properly done because copy number is centered on 2 and there is no wave effect due to the GC-content. Indeed, GC-content inside or outside exons affect the read alignments [Benjamini and Speed, 2012].

9.2.2 B allele fraction

Two main strategies are possible to determine the B allele fraction. The first one is from the package `sequenza` that attempts to discover new SNPs (where there is a variation of nucleotides at a locus) by the intermediate of the usual tools `samtools` to deal with sequencing data. The main drawback of using all positions is the time consuming because it requires to read the file at each position and count the number

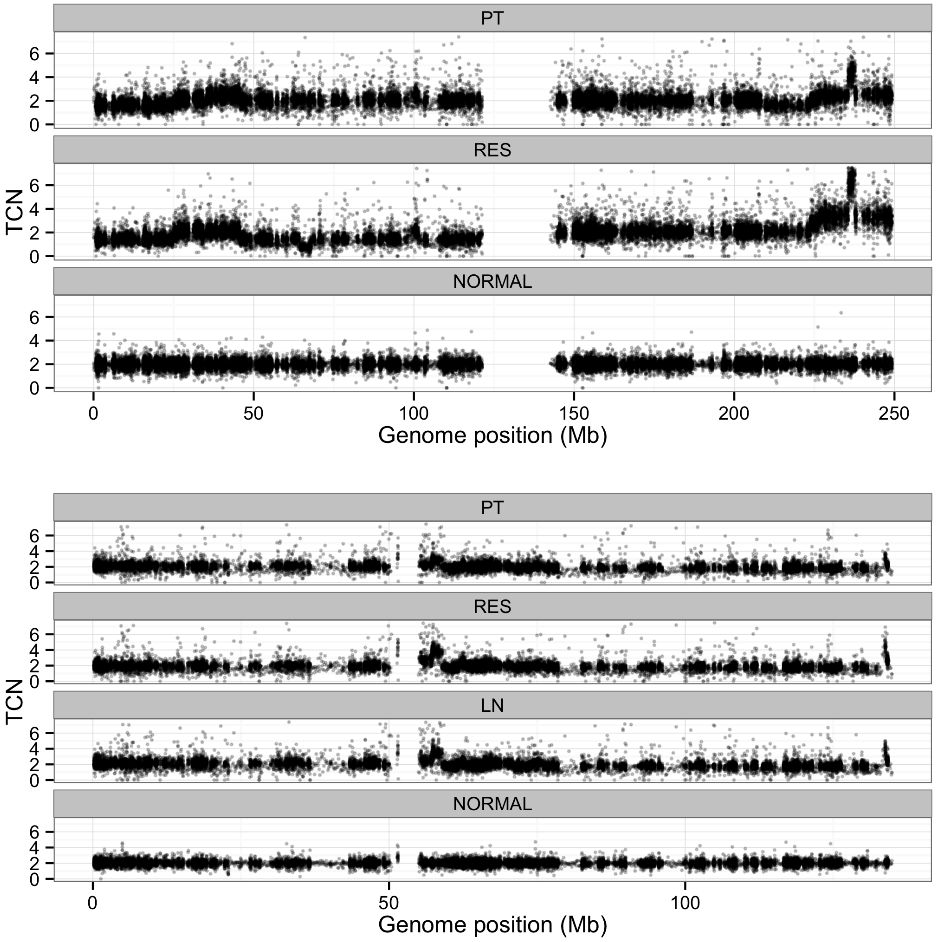


Figure 9.1 – Copy number profiles of two patients of NACRE project (chromosome 1 and 11)

of aligned reads.

Then, we started a collaboration with Henrik Bengtsson during a three-week stay at UCSF, we attempted to compute DoH from whole exome sequencing data by using known SNPs. Indeed, SNPs uses by microarray are nucleotide variations well known in the population. This approach is included in the package `aroma.seq` in the function `countNucleotides`. In this way, it is possible to exclude all other positions and compute DoH only for loci where there are known SNPs. To get more information, we attempted to use the database of Ensembl [Yates et al., 2016] or dbSNP [Sherry et al., 2001]. However, these two databases contain a huge number of indexed SNPs and it requires a computing server with a large RAM. It is possible to get around this issues by selecting only SNPs that are in exons, but this step hasn't been done yet.

However, we present in this section some results from `sequenza` and the results of the first experiences with `countnucleotides`.

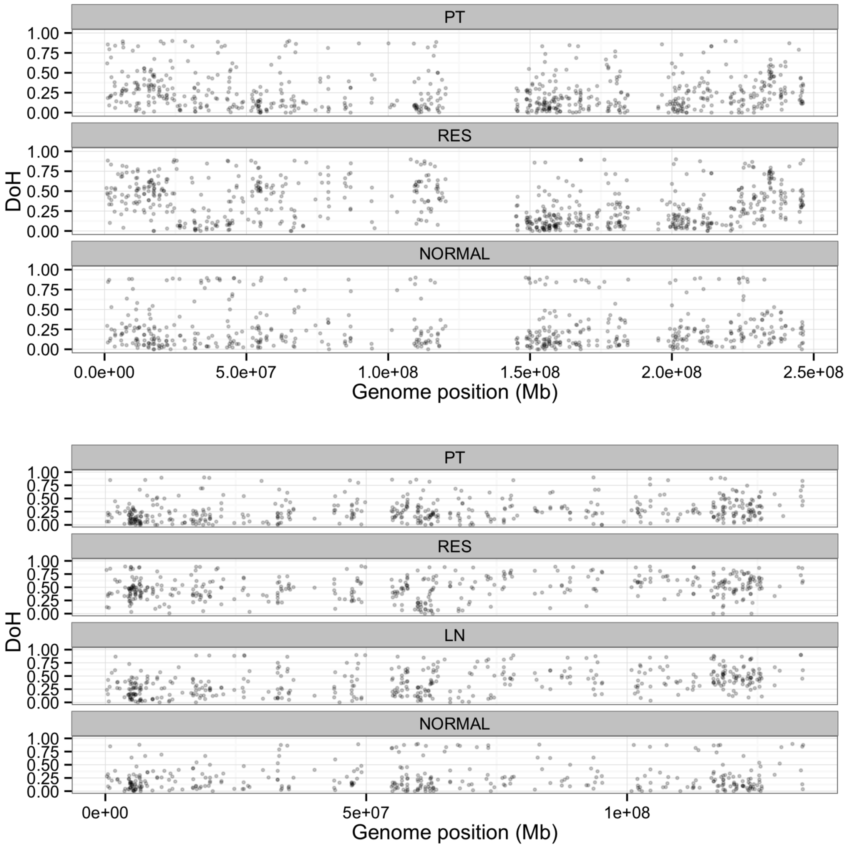


Figure 9.2 – DoH profiles for two patients of NACRE project (chromosome 1 and 11), countnucleotides method

Even after filtering (remove all loci where the number of aligned reads are less than 10), the estimation of the DoH is not evident because of the variable depth sequencing due to GC-content for instance. Therefore the DoH signal remains noisy for `countnucleotides` method (Fig. 9.2). Indeed, it is difficult to get an optimal estimation of the B allele fraction when the number of aligned reads at a locus is poor. Another one drawback is the difficulty in differentiating the mutations from sequencing errors or from the true SNPs. It is also the reason for which we have preferred to work on the predefined SNPs over a first phase. Results from sequenza `sequenza` (Fig. 9.3) seems to be less noisy and the alterations seem to be more visible after the removal of the

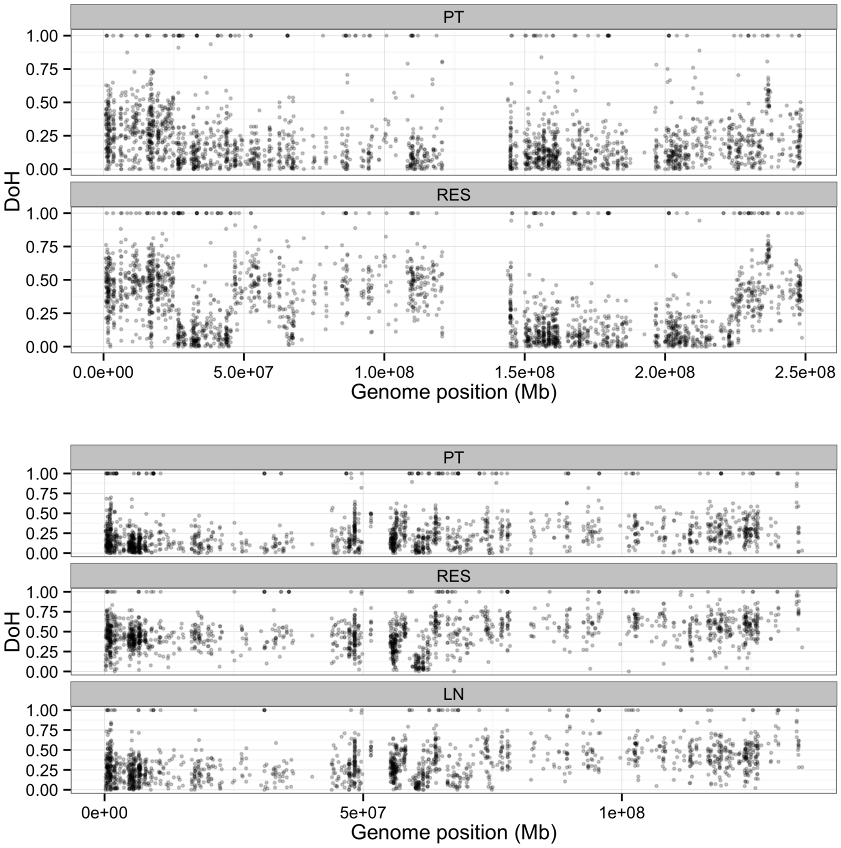


Figure 9.3 – DoH profiles for two patients of NACRE project (chromosome 1 and 11), sequenza method

homozygous SNPs.

As the estimations of the BAF or DoH are not really satisfying and the data integration not evident between CNVkit and sequenza.

9.3 ExCoBAF pipeline

This pipeline was developed by an intern that we have supervised during two months. He combined both `CNVkit` and `sequenza` tools in order to get both TCN and DoH signal from WES data. This pipeline was written in `python` and included several `R scripts`. As the pipeline included several steps that take a non-negligible time, we chose to save the intermediate files in order to save precious time. For example, for the CNVkit tool that computes a pooled reference, if the reference is already done it is not necessary to compute it for each tumor sample. In addition, the pipeline allows saving files in a predefined structure in order to be more user-friendly. First, the user has to create two directories that contain respectively the bam files and the required annotation files i.e (`fasta` file of the reference organism, `bed` file that contains the position of targets in our case exons).

```
bamData/
  NACRE/
    HomoSapiens/
      S1.tumor.bam
      S2.tumor.bam
      S1.normal.bam
      S2.normal.bam

annotationData/
  organisms/
    HomoSapiens/
      hg19.fa
      AgilentSureSelect,hg19.bed
```

From the initial `bed` file, CNVkit create two new files named `target.bed` and `antitarget.bed`. This step permits to cut off the reads aligned on targeted areas and the ones out of targets. Indeed, normalization is not performed in the same way for the two kinds of reads (on and off targets).

```
annotationData/
  organisms/
    HomoSapiens/
      AgilentSureSelect,hg19,target.bed
      AgilentSureSelect,hg19,antitarget.bed
```

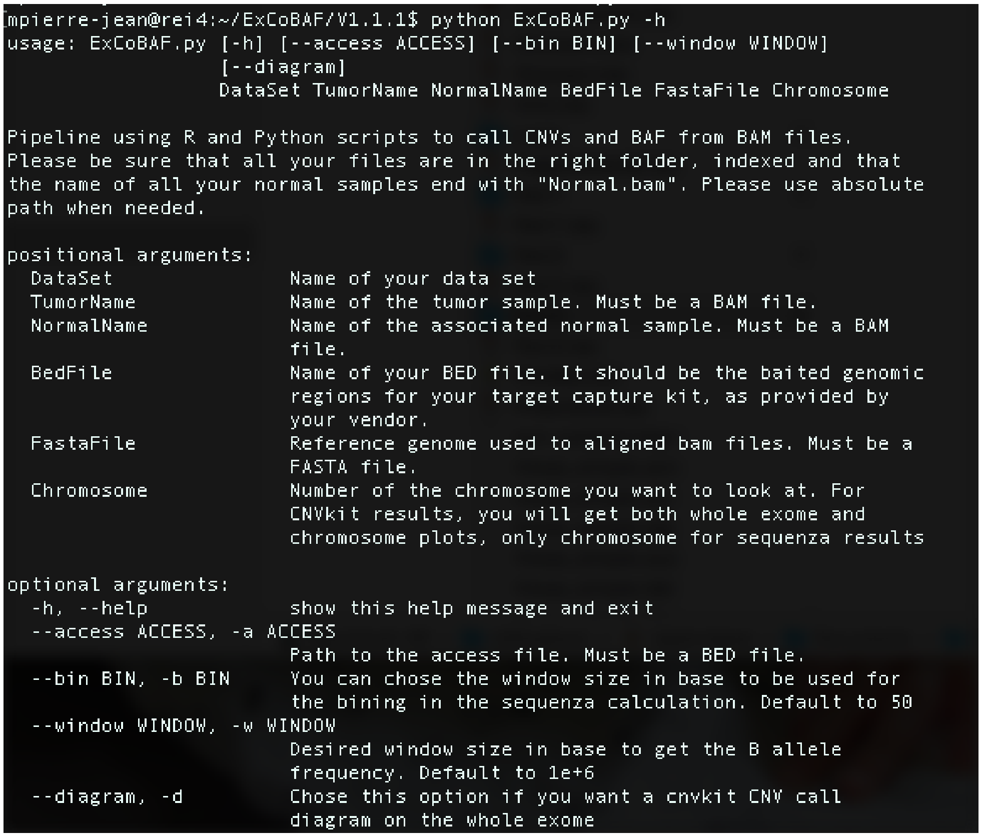
After the CNVkit step, two directories are created to save two different types of files (`cnn` and `cnr`). The `cnn` files contain the count of reads within bins of 200 kb-length and `cnr` contain $log2(TCN)$ within bins of 200 kb i.e after normalization.

```
cnvkitData/
  NACRE,cnn/
    HomoSapiens/
      S1.tumor,target.cnn
      S1.tumor,antitarget.cnn
      ...
      S1.normal,target.cnn
      S1.normal,antitarget.cnn
      ...
      reference.cnn

cnvkitData/
  NACRE,cnr/
    HomoSapiens/
      S1.tumor.cnr
      S2.tumor.cnr
      S1.normal.cnr
      S2.normal.cnr
```

Sequenza part creates on pileup file by tumor sample. A pileup format describes the base-pair information at each chromosomal position, it facilitates SNP/indel calling to compute the BAF from sequencing data. Then sequenza summarizes the pileup files into seqz files that are readable in R. The intern includes a part that transforms seqz files to **rds** files that are compressions of the seqz files after binning the BAF. The window of the binning is adjustable by the user.

```
pileupData/
  NACRE/
    HomoSapiens/
      S1.tumor.pileup.gz
      ...

sequenzaData/
  NACRE,seqz/
    HomoSapiens/
      S1.tumor.seqz
      ...
  NACRE,baf/
    HomoSapiens/
      S1.tumor.rds
      ...
```

An overview of the help of the pipeline is shown in Fig. 9.4.



```
mpierre-jean@rei4:~/ExCoBAF/V1.1.1$ python ExCoBAF.py -h
usage: ExCoBAF.py [-h] [--access ACCESS] [--bin BIN] [--window WINDOW]
                  [--diagram]
                  DataSet TumorName NormalName BedFile FastaFile Chromosome

Pipeline using R and Python scripts to call CNVs and BAF from BAM files.
Please be sure that all your files are in the right folder, indexed and that
the name of all your normal samples end with "Normal.bam". Please use absolute
path when needed.

positional arguments:
  DataSet               Name of your data set
  TumorName             Name of the tumor sample. Must be a BAM file.
  NormalName            Name of the associated normal sample. Must be a BAM
                        file.
  BedFile               Name of your BED file. It should be the baited genomic
                        regions for your target capture kit, as provided by
                        your vendor.
  FastaFile             Reference genome used to aligned bam files. Must be a
                        FASTA file.
  Chromosome            Number of the chromosome you want to look at. For
                        CNVkit results, you will get both whole exome and
                        chromosome plots, only chromosome for sequenza results

optional arguments:
  -h, --help            show this help message and exit
  --access ACCESS, -a ACCESS
                        Path to the access file. Must be a BED file.
  --bin BIN, -b BIN     You can chose the window size in base to be used for
                        the bining in the sequenza calculation. Default to 50
  --window WINDOW, -w WINDOW
                        Desired window size in base to get the B allele
                        frequency. Default to 1e+6
  --diagram, -d         Chose this option if you want a cnvkit CNV call
                        diagram on the whole exome
```

Figure 9.4 – ExCoBAF help

9.4 Parental copy number estimations

In order to apply the heterogeneity model, we have to estimate the parental copy number signals for each sample. As the total copy number signals from CNVkit tool contain more observations than BAF signals, we first segment jointly all TCN signals in order to get more accurate change-points.

However, as we saw in the introduction chapter, we have to know the germinal status (heterozygous or homozygous) of each SNPs. The sequenza tool is able to estimate the germinal status and therefore it is possible to compute the DoH. Actually, we compute the DoH by segment and then we deduce the parental copy number signals by segments. Estimates by segment are sufficient to apply the heterogeneity model. However, in some segments, there are no heterozygous SNPs and therefore no parental copy number estimations. For the moment, our model doesn't deal with missing values and to get around this issue, we remove the segments with no estimation of parental copy number signals. We have noted that it was often very small segment. An example of the smoothed parental copy number signals for the patient 1 is shown in Fig. 9.5. In this example, it appears that some change-points are not well detected and may be model selection is too stringent. Some further exploration will perform to explain and fix this issue.

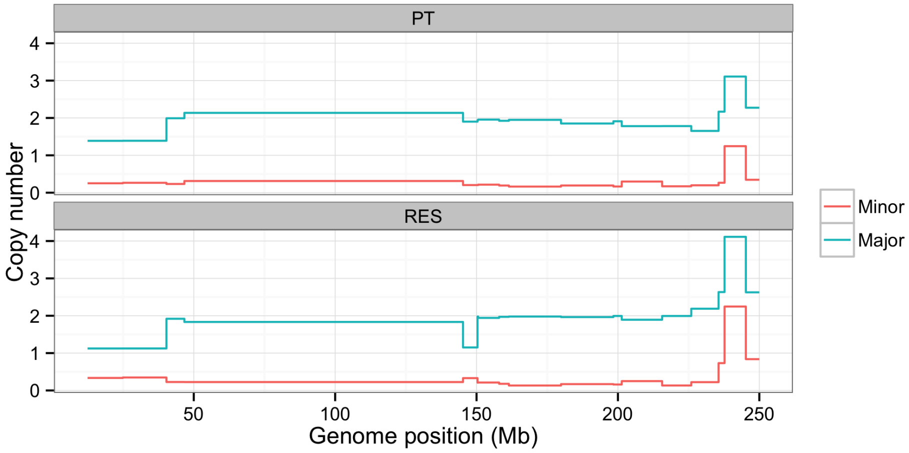
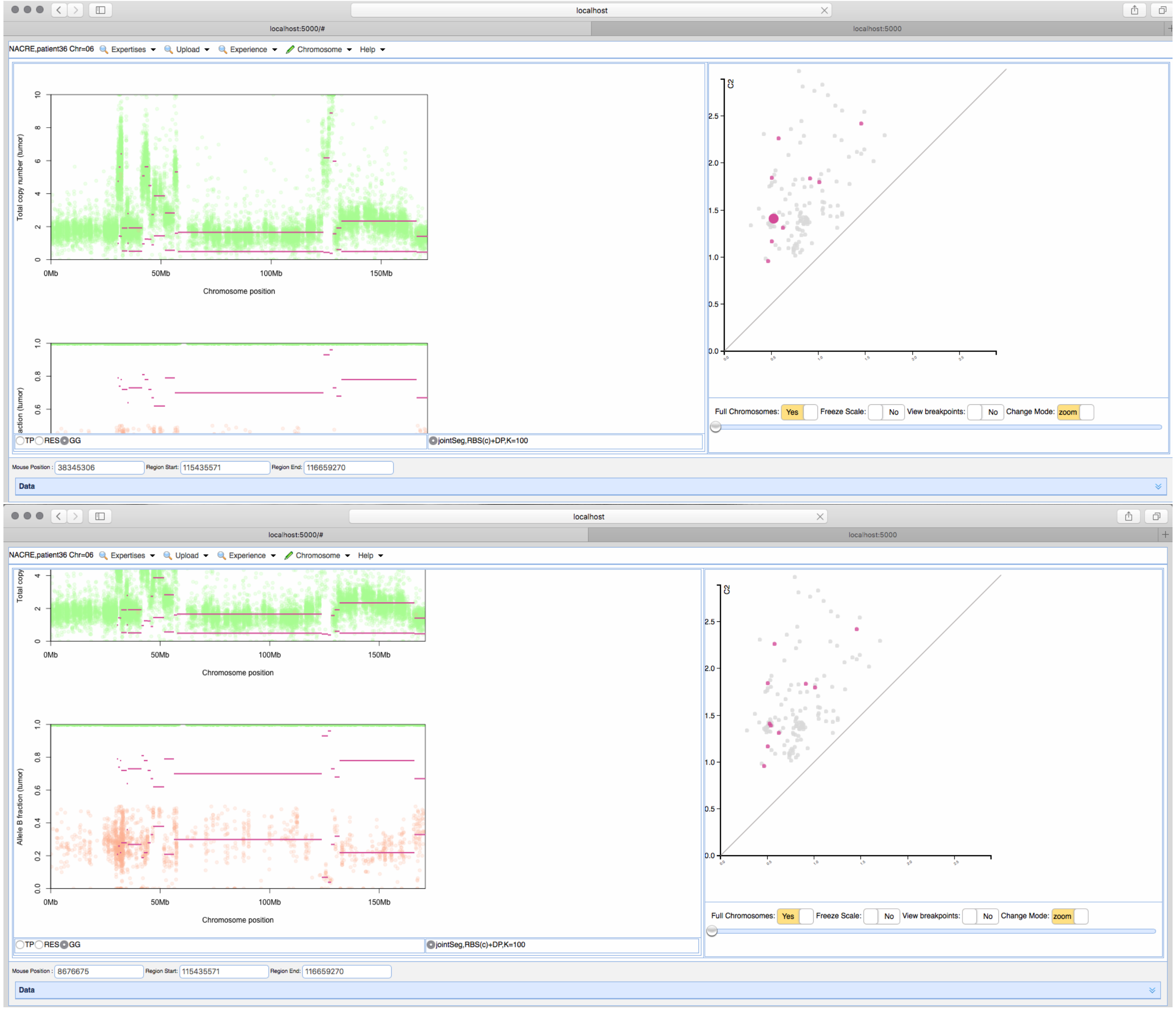


Figure 9.5 – Joint segmentation chromosome 1 of Patient 1 sample

9.5 Integration to a visualization tool

This section presents a visualization interface currently developed in the lab by an engineer (Franck Samson). The developed interface aims to enable a quick visualization of TCN and BAF firstly from microarray and then from WES across several samples. A quick visualization could be used to annotate a new data set to perform simulations as explained in chapter 3 but also annotated particular regions as “normal”which is necessary to estimate DoH without normal reference sample (Chapter 8).Trough the development pipeline ExCoBAF and R scripts, it has been possible to visualize simultaneously TCN and DoH from WES data with the interface.

An overview of the interface is presented in Fig. 9.6. It is possible thanks to the interface to switch from a sample to another for the same patient (for instance PT to RES to LN). But also annotate some regions (see Fig.9.7). We plan to add new functionalities to the interface in the next months as described previously. For instance, be able to annotate normal region in order to estimate DoH in absence of a reference sample. The functionality of annotating and is currently in development is independent of the visualization of sequencing data.



NACRE,patient36 Chr=06

○TP ○RES ●GG

●jointSeg,RBS(c)+DP,K=100

Full Chromosomes: Yes Freeze Scale: No View breakpoints: No Change Mode: zoom

Mouse Position : 38345306 Region Start: 115435571 Region End: 116659270

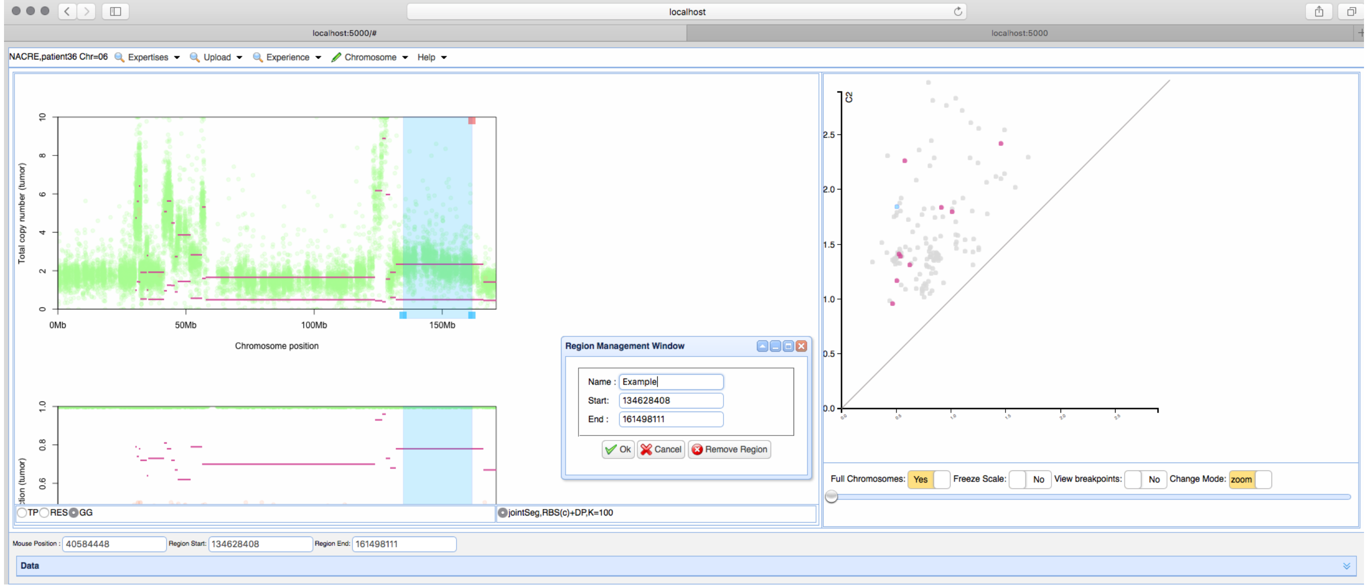
○TP ○RES ●GG

●jointSeg,RBS(c)+DP,K=100

Full Chromosomes: Yes Freeze Scale: No View breakpoints: No Change Mode: zoom

Mouse Position : 8676675 Region Start: 115435571 Region End: 116659270

Figure 9.6 – Example of interface to visualize the DNA copy number data



Name : Example
Start: 134628408
End : 161498111

○ TP ○ RES ● GG

● jointSeg,RBS(c)+DP,K=100

Full Chromosomes: Yes
Freeze Scale: No
View breakpoints: No
Change Mode: zoom

Mouse Position : 40584448
Region Start: 134628408
Region End: 161498111

Figure 9.7 – Example of interface to visualize the DNA copy number data

9.6 Conclusion

In conclusion, in this chapter we presented the pipeline developed by an intern in the lab in order to deal easier with the data provided by Institut Curie. From this pipeline, it is possible to apply the joint segmentation method developed in the first part of this thesis and then to provide an estimation of segmented parental copy number signals. Once, these estimations have been done, it has been possible to apply the heterogeneity model described in the second part of the thesis.

However, it is clear that the model selection is too stringent and requires to be more flexible in order to enable more change-points and to not miss evident ones. A possible amelioration could be to annotate the data set from Institut Curie with the interface developed in the lab. Indeed, it is easy to browse from a sample to another.

Conclusion and perspectives

Genomic data analysis is emergent in cancer research since several years in order to attempt developing of personalized treatments. Study the genomic features of the tumors can improve diagnosis as well as prognosis of patients suffering from cancer. Cancer is a heterogeneous disease that is difficult to characterize, in particular at a genetic level. Genomic data analysis requires the development of efficient statistical methods able to deal with the large volume of data that is produced.

This Ph.D. thesis is focused on the analysis of DNA copy number in cancerology. The aim was to provide statistical methods but also efficient software in term of time and memory in order to deal with data from microarrays and HTS. The final aim is to characterize the tumors and give some indications and guidelines to the drug resistance. This last chapter summarizes the main points developed in this thesis and provides some general conclusions and as well as perspectives on this work.

General conclusion

The work performed here deals with several fields: statistics, bioinformatic, genomics and cancerology. All along the manuscript, we have attempted to pay attention to introducing the basic concepts of each domain. For this reason, the introduction aims to present general genomic concepts and cancer. Indeed, cancer is a well-known disease by all the population but only by the name and the poor prognosis that it recalls. Few people know that huge genetic alterations can be present in tumor cells are decisive for the prognosis. The thesis is particularly focused on one level of genetic alterations, namely DNA copy number alterations. We briefly present at the end of the chapter the statistical and bioinformatical issues of this kind of data.

The introduction of the first part begins by an introduction to the univariate statistical models to segment DNA copy number signals and our contributions to segment jointly TCN and BAF. Then, we showed that taking into account all information from these technologies is essential because it improves the efficiency of the methods. We implemented R packages in order to easily reproduce the performance evaluation study. The vignettes of the packages are presented in Appendix.

In a second part, we studied the intra and inter-tumoral heterogeneity by using multi-sample data sets. Several strategies are possible to infer tumoral heterogeneity and we have chosen to assume that tumor samples from the same patient or the same disease could share common alterations. We extended an existing model by adding constraints with more biological sense and that includes the BAF information through the parental copy number signals. We provided an R package named InCaSCN for this model, that also enables to reproduce the results of simulations. Despite the good results produced by the simulations, we wanted to know if the method could be used in practice and we applied it on two different data sets. The first one is a public data set and the second one arises from a collaboration with Institut Curie in order to investigate heterogeneity in triple-negative breast cancers. We dealt with microarray data for the first data set and HTS data for the second one. The study designs were different for the two applications. The first application deals with intra-tumoral heterogeneity: several tumor samples from the same patient. And the second one deals with inter-tumoral heterogeneity: several samples from different patients suffering from TNBC. We have been able to draw a few conclusions with a biological sense. For instance, it seems quite clear that some clones are resistant to treatments and other ones are not. Indeed, it appears that clones are present in primary tumor and not in relapses samples.

During this Ph.D. thesis, we encountered issues to deal with real data. For instance, when we wanted to analyze the public data set, no reference sample was available. As we wanted to apply the InCaSCN method on parental copy number signals, we had to develop a strategy to compensate the lack of reference and get parental copy number estimations by segments. Currently, because some improvements are required, there is no package available to apply this strategy but we are working on making it available on a free platform as soon as possible. Then, Institut Curie provided us sequencing data and we had to deal with this new type of data. For WES data, through the work of an intern, we combined several tools in order to estimate both DNA copy number and BAF (quantities that we were used to dealing with). By means of this implemented pipeline, we have been able to apply InCaSCN method to the WES data from Institut Curie.

All along this thesis we have attempted to propose efficient statistical methods to deal with the particularities of new genomic technologies but also with the features of cancers. Every time, we proposed a model with as most biological sense as possible coupled with available and efficient implementations.

Perspectives

Even if the manuscript resulted in several conclusions, we also highlighted some drawbacks that could lead to interesting research perspectives.

Improvement of the heterogeneity model

The first improvement can be done on InCaSCN model that it has been developed to investigate heterogeneity. After applying the model on real data, some of the conclusions produced were difficult to interpret. For instance, it is unlikely that breakpoints occur both in minor and major copy number. Therefore, it could be interesting to add this constraint in the model and to check if the performance is still as good as before.

Another important point is that we do not provide proofs that our model converge to a global minimum. Some mathematical proofs and experiments will be required to be sure that the model provides an acceptable solution. For instance, we show that the way to initialize the algorithm is not perfect and requires improvements.

The other parameters that we did not take into account explicitly in the model are the ploidy and the cellularity. These parameters vary from a sample to another one and adding a constant to estimate them directly in the model could also improve the performance and provide conclusions with a better biological meaning.

Improvement of the pipeline to estimate DoH

As we said in the previous section, we encountered some issues when we had to deal with the real data. To deal with public data of [Schwarz et al., 2015], we developed a short pipeline to estimate DoH in absence of a reference sample. It is clear that we have to improve this pipeline in order to distribute it to the bioinformatic community. We proposed a solution to remove the bias induced by assumptions that are not really respected, but this solution depends on the sample and technology. Extensive experiments on various data sets can be led in order to help us to provide a general estimation of DoH by segments which will better.

TCGA application

During the internship of the M1 student that we supervised, we asked to The cancer genome atlas (TCGA) several data sets in order to test the pipeline that enables to compute both TCN and BAF from WES data. TCGA enables us to use WES data from head and neck cancer, pleuropulmonary blastoma and breast cancer. These data sets not only contain WES data but also SNPs arrays. It could be interesting to compare estimations from WES and estimations from arrays. This is a new recent project that we did not have time to develop. The data set on breast cancer aims to discover subtypes, therefore we could also apply the InCaSCN model on it and confirm their subtypes.

Data integration

Finally, Institut Curie has two different types of sequencing data (RNAseq and DNAseq). It could be very interesting to perform a joint analysis of this two kind of data in order to may discover common points between patients or some relations between the RNAseq samples and those of the DNAseq for the same patient. A first conclusion from boths kind of data (RNAseq and DNAseq) is that tumoral genetic alterations are very specific to the patients. In conclusion, TNBC is a very heterogeneous disease.

All this work aims to help biologists and physicians to develop adapted treatments for each type of cancer by the intermediate of statistical and bioinformatic tools.

Appendix A

Proof of Lemma 1

Proof.

$$2\sigma^2 \log(RV) = \left(-t\widehat{\varGamma}_1^2 - (J-t)\widehat{\varGamma}_2^2 + J\widehat{\varGamma}_0^2\right) \tag{A.1}$$

We can write $\widehat{\varGamma}_0$ as a function of $\widehat{\varGamma}_1$ and $\widehat{\varGamma}_2$

$$\widehat{\varGamma_0} = \frac{1}{J}\left(t\widehat{\varGamma}_1 + (J-t)\widehat{\varGamma}_2\right) \tag{A.2}$$

So, by replacing $\widehat{\varGamma_0}$ in A.1 by the expression defined by A.2

$$2\sigma^2 \log(RV) = \left(-\frac{Jt}{J}\widehat{\varGamma}_1^2 - \frac{J(J-t)}{J}\widehat{\varGamma}_2^2 + \frac{1}{J}\left(t\widehat{\varGamma}_1 + (J-t)\widehat{\varGamma}_2\right)^2\right) \tag{A.3}$$

By developing the last term of A.4 :

$$\begin{aligned}2\sigma^2 \log(RV) = \Bigg(&-\frac{Jt}{J}\widehat{\varGamma}_1^2 - \frac{J(J-t)}{n}\widehat{\varGamma}_2^2 + \frac{t^2}{J}\widehat{\varGamma}_1^2 \\ &+ \frac{(J-t)^2}{J}\widehat{\varGamma}_2^2 + 2\frac{t(J-t)}{n}\widehat{\varGamma}_1\widehat{\varGamma}_2\Bigg)\end{aligned} \tag{A.4}$$

So,

$$\begin{aligned}2\sigma^2 \log(RV) = \Bigg(&-\frac{t(J-t)}{J}\widehat{\varGamma}_1^2 - \frac{(J-t)(J-(J-t))}{J}\widehat{\varGamma}_2^2 \\ &+2\frac{t(J-t)}{J}\widehat{\varGamma}_1\widehat{\varGamma}_2\Bigg)\end{aligned} \tag{A.5}$$

To finish:

$$2\sigma^2 \log(RV) = \left(-\frac{t(J-t)}{J}\left(\widehat{\mathit{\Gamma}}_1 - \widehat{\mathit{\Gamma}}_2\right)^2\right) \tag{A.6}$$

□

Appendix B

LARS algorithm for the Group Fused Lasso of (Vert and Bleakley)

```
Initialization: $k = 1$ ;
$\mathcal{A}_0 = \varnothing$ no change-point in the signal, in the following $\mathcal{A}_k$ will be the active set at step $k$ (i.e the $k$ breakpoints).;
Remember that $\beta$ is the matrix of successive differences and which is an $(J-1) \times 2$ matrix. First column is the difference on the TCN and the second one is the difference on the DoH. ;
$c^{[0]} = \mathbf{X}^T\mathbf{Y}$, that is easy to compute by using structure of matrix $\mathbf{X}$;
First change-point $\mathcal{A}_1$ is the position which maximizes the gain by cutting exactly at this position (the larger difference between the mean at the left and mean at the right of this change-point).;
for $k \leftarrow 1, \ldots S$ do
    $\mathcal{A}_k$ active set at step $k$;
    Compute the direction $v$ which is an $(J-1) \times 2$ matrix such that $w_{\mathcal{A}_k^c} = 0$ and
        $w_{\mathcal{A}_k} = (X'_{\mathcal{A}_k} X_{\mathcal{A}_k})^{-1} c^{[k-1]}_{\mathcal{A}_k}$        (B.1)
    where $X_{\mathcal{A}_k}$ is a sub-matrix of $X$ composed of columns with indexes in $\mathcal{A}_k$ and $c^{[k-1]}_{\mathcal{A}_k}$ is sub-matrix of $c^{[k-1]}$ composed of rows with indexes in $\mathcal{A}_k$.;
    $w$ is in fact a $\beta^{OLS}$ where some variables are not active.;
    $a = \mathbf{X}'\mathbf{X}w$ for all active indexes, by construction $a_{\mathcal{A}_k} = c^{[k-1]}_{\mathcal{A}_k}$;
    Then, for each index $j$ that is not in $\mathcal{A}_k$, we look for the most correlated variable $X_j$ in the direction $w$. This variable will be included in $\mathcal{A}_{k+1}$. The correlation is determine by the variable $\nu_j \in [0,1]$ such that:
        $\|c^{[k-1]}_j - \nu_j a_j\|_2^2 = \|c^{[k-1]}_{j'} - \nu_j a_{j'}\|_2^2$        (B.2)
    where $j'$ is chosen arbitrarily in the active set, but in the implementation we define $j' = \arg\max_{j \in \mathcal{A}_k} \|c^{[k-1]}_j\|_2^2$;
    Then, $\nu_{j*} = \min_{j \notin \mathcal{A}_k \text{ et } \nu_j > 0} \nu_j$;
    Finally, $\mathcal{A}_{k+1} = \mathcal{A}_k \cup \{j*\}$ and $c^{[k]} = c^{[k-1]} - \nu_{j*} a$ are updated.
```

Algorithm 7: LARS algorithm.

Appendix C

Inclusion of RKHS

C.1 Generalities

In this section we recover the results of Corollary 3.14 in [Steinwart et al., 2006]. Following lemmas and propositions are from [Zhang and Zhao, 2011].

Lemme 1. *Let K, G two kernels on $\mathcal{X}$, $\mathcal{H}_K \subseteq \mathcal{H}_G$ if and only if there exists a positive constant $\lambda \leq 0$ such as $K \ll \lambda G$ ($\lambda G - K$ is still a kernel on $\mathcal{X}$).*

Lemme 2. *For two kernels K, L on $\mathcal{X}$*

- *$K \ll L$ if and only if $\mathcal{H}_K \subseteq \mathcal{H}_L$ and $\|f\|_{\mathcal{H}_L} \leq \|f\|_{\mathcal{H}_K}$.*
- *$K \ll \lambda L$ if and only if $\mathcal{H}_K \subseteq \mathcal{H}_L$ and $\|f\|_{\mathcal{H}_L} \leq \sqrt{\lambda}\|f\|_{\mathcal{H}_K}$.*

Then, with lemme 2

Proposition 1. *Let K, G two kernels on $\mathcal{X}$ with $\mathcal{H}_K \subseteq \mathcal{H}_G$ then $K \ll \lambda(K,G)G$ with $\beta(K,G) = \sqrt{\lambda(K,G)}$ ($\|f\|_{\mathcal{H}_G} \leq \beta(K,G)\|f\|_{\mathcal{H}_K}$ with $f \in \mathcal{H}_K$).*

Theorem 1. *u,v two positive functions of $L^1(\mathbb{R}^d)$*

Let K, G define by :

- *$K(x,y) = \int_{\mathbb{R}^d} e^{i\langle x-y,\xi\rangle} u(\xi) d\xi$*
- *$G(x,y) = \int_{\mathbb{R}^d} e^{i\langle x-y,\xi\rangle} v(\xi) d\xi$*

then $\mathcal{H}_K \subseteq \mathcal{H}_G$ if and only if $\left\{t \in \mathbb{R}^d : u(t) > 0, v(t) = 0\right\} = \emptyset$ and $\frac{u}{v}$ is bounded on $\left\{t \in \mathbb{R}^d : v(t) > 0\right\}$ $\lambda(K,G)$ = supremum of $\frac{u}{v}$ on $\left\{t \in \mathbb{R}^d : v(t) > 0\right\}$

C.2 Application to the Gaussian kernel

Let

$$G_\delta(x,y) \quad = \quad \exp\left(-\frac{\|x-y\|^2}{\delta}\right) = \int_{\mathbb{R}^d} e^{i\langle x-y,\xi\rangle} g_\delta(\xi) d\xi, \quad x,y \in \mathbb{R}^d, \delta > 0$$

with

$$g_\delta(\xi) \quad := \quad \left(\frac{\sqrt{\delta}}{2\sqrt{\pi}}\right)^d \exp\left(-\frac{\delta\|\xi\|^2}{4}\right), \quad \xi \in \mathbb{R}^d$$

Take $0 < \delta_1 < \delta_2$ and $G_{\delta_1}, G_{\delta_2}$ the associated kernels.

Then

$$\mathcal{H}_{G_{\delta_2}} \subseteq \mathcal{H}_{G_{\delta_1}} \quad \Leftrightarrow \quad \left\{t \in \mathbb{R}^d, g_{\delta_2}(t) > 0, g_{\delta_1}(t) = 0\right\}$$

We have

$$g_{\delta_1}(\xi) = 0 \quad \Leftrightarrow \quad \exp\left(-\frac{\delta_1\|\xi\|^2}{4}\right) = 0 \text{ avec } \delta_1 < 0$$

Hence

$$\left\{t \in \mathbb{R}^d, g_{\delta_1}(\xi) = 0, g_{\delta_2}(\xi) > 0\right\} \quad = \quad \emptyset$$

Hence

$$\mathcal{H}_{G_{\delta_2}} \quad \subseteq \quad \mathcal{H}_{G_{\delta_1}}$$

Now, checking that $\frac{u}{v}$ is bounded:

Computing λ:

$$\frac{g_{\delta_2}}{g_{\delta_1}}(\xi) \quad = \quad \left(\sqrt{\frac{\delta_2}{\delta_1}}\right)^d \exp\left(-\frac{(\delta_2-\delta_1)}{4}\|\xi\|^2\right)$$

As $0 < \delta_1 < \delta_2$, on a $-(\delta_2 - \delta_1) < 0$

Hence,

$$\begin{aligned}
\exp\left(-\frac{(\delta_2-\delta_1)}{4}\|\xi\|^2\right) &< 1, \quad \forall \xi \in \mathbb{R}^d \\
\left(\sqrt{\frac{\delta_2}{\delta_1}}\right)^d \exp\left(-\frac{(\delta_2-\delta_1)}{4}\|\xi\|^2\right) &< \left(\sqrt{\frac{\delta_2}{\delta_1}}\right)^d \quad \forall \xi \in \mathbb{R}^d \\
\frac{g_{\delta_2}}{g_{\delta_1}}(\xi) &< \left(\sqrt{\frac{\delta_2}{\delta_1}}\right)^d \quad \forall \xi \in \mathbb{R}^d
\end{aligned}$$

Hence $\lambda = \left(\frac{\delta_2}{\delta_1}\right)^{\frac{d}{2}}$

Appendix D

Proof of Lemma 3

Proof. Suppose that $\mathbf{W}$ is fixed. We wish to solve (6.8) in $\mathbf{Z}_m$. By dropping the index m for clarity, we can consider solving the problem

$$\min_{\mathbf{Z}\in\mathbb{R}^{pS}} \|\mathbf{Y}-\mathbf{W}\mathbf{Z}\|^2 + \lambda \left\|\mathbf{D}\mathbf{Z}^\top\right\|_1 , \tag{D.1}$$

Now, by simple matrix algebra and properties of the vectorization operator, let us show that this can be written as an equivalent Lasso problem.

First, use the equivalence between the trace and the Frobenius norm, and operate the change of variable $\tilde{\mathbf{Z}} = \mathbf{D}\mathbf{Z}^\top$ (that is, $\mathbf{D}^{-1}\tilde{\mathbf{Z}} = \mathbf{Z}^\top$)

$$\|\mathbf{Y}-\mathbf{W}\mathbf{Z}\|^2 + \lambda \left\|\mathbf{D}\mathbf{Z}^\top\right\|_1 = \operatorname{Tr}\left\{\left(\mathbf{Y}-\mathbf{W}\tilde{\mathbf{Z}}^\top\mathbf{D}^{-\top}\right)\left(\mathbf{Y}-\mathbf{W}\tilde{\mathbf{Z}}^\top\mathbf{D}^{-\top}\right)^\top\right\} + \lambda\left\|\tilde{\mathbf{Z}}\right\|_1$$

Now, use the rules $\operatorname{Tr}(\mathbf{A}\mathbf{B}^\top) = \operatorname{vec}(\mathbf{A})^\top \operatorname{vec}(\mathbf{B})$ to show that

$$\begin{aligned}
\operatorname{Tr}\left\{\left(\mathbf{Y}-\mathbf{W}\tilde{\mathbf{Z}}^\top\mathbf{D}^{-\top}\right)\left(\mathbf{Y}-\mathbf{W}\tilde{\mathbf{Z}}^\top\mathbf{D}^{-\top}\right)^\top\right\} + \lambda\left\|\tilde{\mathbf{Z}}\right\|_1 &= \\
\operatorname{vec}\left(\mathbf{Y}-\mathbf{W}\tilde{\mathbf{Z}}^\top\mathbf{D}^{-\top}\right)^\top \operatorname{vec}\left(\mathbf{Y}-\mathbf{W}\tilde{\mathbf{Z}}^\top\mathbf{D}^{-\top}\right) &+ \lambda\left\|\operatorname{vec}(\tilde{\mathbf{Z}})\right\|_1 \\
= \left(\mathbf{y}-\operatorname{vec}(\mathbf{W}\tilde{\mathbf{Z}}^\top\mathbf{D}^{-\top})\right)^\top \left(\mathbf{y}-\operatorname{vec}(\mathbf{W}\tilde{\mathbf{Z}}^\top\mathbf{D}^{-\top})\right) &+ \lambda\|\tilde{\mathbf{z}}\|_1 ,
\end{aligned}$$

where we denoted by $\mathbf{y} = \operatorname{vec}(\mathbf{Y})$ and $\tilde{\mathbf{z}} = \operatorname{vec}(\tilde{\mathbf{Z}})$.

Using that $\operatorname{vec}(\mathbf{A}\mathbf{X}\mathbf{B}) = (\mathbf{B}^\top \otimes \mathbf{A})\operatorname{vec}(\mathbf{X})$, one has

$$\begin{aligned}&\left(\mathbf{y} - \mathrm{vec}(\mathbf{W}\tilde{\mathbf{Z}}^\top\mathbf{D}^{-\top})\right)^\top \left(\mathbf{y} - \mathrm{vec}(\mathbf{W}\tilde{\mathbf{Z}}^\top\mathbf{D}^{-\top})\right) + \lambda \left\|\tilde{\mathbf{z}}\right\|_1 \\ &\qquad = \left(\mathbf{y} - \left(\mathbf{D}^{-1} \otimes \mathbf{W}\right)\tilde{\mathbf{z}}^\top\right)^\top \left(\mathbf{y} - \left(\mathbf{D}^{-1} \otimes \mathbf{W}\right)\tilde{\mathbf{z}}^\top\right) + \lambda \left\|\tilde{\mathbf{z}}^\top\right\|_1\end{aligned}$$

$$\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \|\beta\|_1$$

with $\beta = \tilde{\mathbf{z}}^\top$ and $\mathbf{X} = \mathbf{D}^{-1} \otimes \mathbf{W}$. We recognize a Lasso problem. Note that $\mathbf{D}^{-1}$ is an upper triangular matrix filled with -1. We get back the original $\mathbf{Z}$ by first reconstituting $\tilde{\mathbf{Z}}$ by stacking β row-wise, then multiply $\tilde{\mathbf{Z}}$ by $\mathbf{D}^{-1}$. This latter operation can be done efficiently with cumulative sums. □

Appendix E

R packages

This chapter aims to present packages developed to analyze DNA copy number data. In a first section, we present the strategy used to compare the segmentation methods introduced in chapter 3. In order to measure performance of DNA copy number segmentation methods, we developed a pipeline to build synthetic copy number profiles from real data set. The first step is to annotate a real data set, then by resampling build synthetic profiles by using only few parameters as the number of breakpoints and the length of signal. This implementation is included in the `R` package `jointseg`. In a second section, we show how to used the `R` package `jointseg` that groups several segmentation methods through a unified interface. To finish, we present the last package `incas-cn` developed in order to analyze heterogeneity from a multi-sample data set. This package is under development and will be soon publicly available.

E.1 Generating realistic copy number profiles

E.1.1 Annotated data set

We annotated altered regions of two data sets from a public database to form that we call annotated data set. This annotated data set is included in an `R` package named `acnr` for "annotated copy number regions". This package is only used to save the data. We describe the characteristics of the two data sets below.

Data set 1 We have worked with a lung cancer data [Rasmussen et al., 2011], for which raw data is accessible at NCBI GEO database [Edgar et al., 2002], accession GSE29172. DNA from patient-matched lung cancer and blood cell lines `NCI-H1395` and

NCI-BL1395 were mixed to simulate tumor tissue with 30, 50, 70, 100% cancer cells. DNA was analyzed on Affymetrix SNP6.0 microarray. Data were normalized using ASCRMAv2 [Bengtsson et al., 2009] followed by TumorBoost [Bengtsson et al., 2010]. For the sake of reproducibility, the R scripts that were written to normalize this data set are distributed in the jointseg package, together with the normalized data itself.

Data set 2 We have also worked with a breast cancer data [Staaf et al., 2008], for which raw data is accessible at NCBI GEO database [Edgar et al., 2002], accession GSE11976. DNA from patient-match breast cancer cell line (HCC1395) and its match normal HCC1395BL were mixed to simulate tumor tissue with 14, 34, 50, 79, 100% cancer cells. DNA was analyzed on Illumina HumanCNV370-Duov1 microarrays. We obtained the BAF-normalized and summarized data as calculated by the Illumina BeadStudio software [Illumina, inc, 2006, Illumina, inc, Peiffer et al., 2006]

Description of annotated copy-number regions The list below describes the different copy number states available for data generation. They are labeled as a pair (c_1, c_2), where c_1 corresponds to the minor copy number (the smallest of the two parental copy numbers), and c_2 corresponds to the major copy number (the largest of the two) [Neuvial et al., 2011].

(1,1): normal (one copy from each parent)

(0,1): hemizygous deletion (loss of one parental copy)

(0,0): homozygous deletion (loss of both parental copies)

(0,2): copy-neutral LOH (loss of one parental copy and gain of the other)

(0,3): loss of one parental copy and gain of two copies from the other parent)

(1,2): single copy gain

(1,3): unbalanced two-copy gain (gain of two copies from the same parent)

(2,2): balanced two-copy gain (gain of one copy from each parent)

(2,3): three-copy gain (gain of one copy from each parent, and two copies from the other parent)

CN state	(0,1)	(0,2)	(0,3)	(1,1)	(1,2)	(1,3)	(2,2)	(2,3)	(0,0)
Data set 1	22615	24135	25405	21539	19048	20903	27924	31098	0
Data set 2	2492	5484	6545	3196	2746	0	3044	0	838

Table E.1 – Size of annotated copy-number regions for each of the 2 data sets.

E.1.2 Resampling

As we said in the introduction of the chapter, we first proceed by annotating a data set. The Fig. E.1 (left) shows two annotated regions for a same sample but on two different chromosomes. The red area correspond to a copy-neutral LOH region and the yellow one to a normal region.

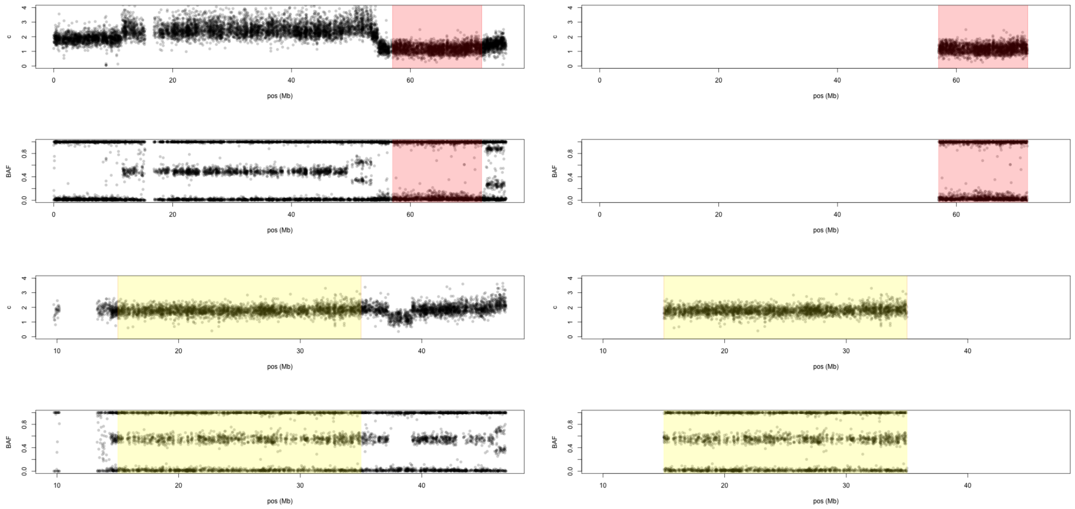


Figure E.1 – Example of annotated regions

Then, in a second step, we isolate these annotated regions to build the database in package `acnr` (Fig. E.1 right). This permits to get a light database that contains only the interesting altered regions.

To finish, the red and yellow areas presented in Fig. E.2 are built from a resampling of the red and yellow areas of Fig. E.2. In this simulation, we force the status of the regions as well as the position of breakpoints,. However, the user can let the package attribute a label to each region and place the breakpoints randomly. Therefore, the required parameters are the number of breakpoints and the length of the simulated signal.

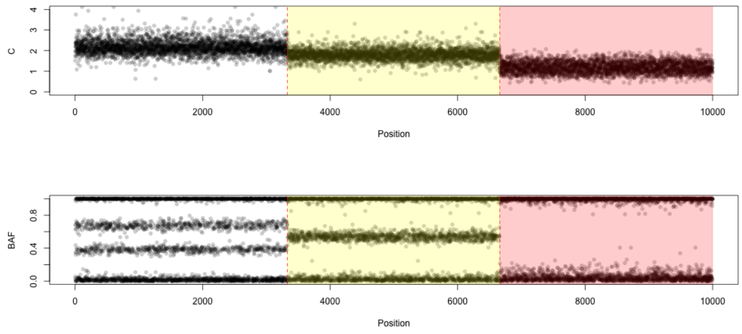


Figure E.2 – Example of a simulated profile

E.1.3 Vignette

In this subsection, we present the vignette of the `jointseg` package that illustrates how to generate a variety of copy-number profiles from the same biological "truth". Such profiles have been used to compare the performance of segmentation methods [Pierre-Jean et al., 2015].

```
library(jointseg)
```

The parameters are defined as follows:

```
n <- 1e4                                        ## signal length
bkp <- c(2334, 6121)                            ## breakpoint positions
regions <- c("(1,1)", "(1,2)", "(0,2)")  ## copy number regions
```

```
ylims <- cbind(c(0, 5), c(-0.1, 1.1))
colG <- rep("#88888855", n)
hetCol <- "#00000088"
```

For convenience we define a custom plot function for this vignette:

```
plotFUN <- function(dataSet, tumorFraction) {
    regDat <- loadCnRegionData(dataSet=dataSet, tumorFraction=tumorFraction)
    sim <- getCopyNumberDataByResampling(n, bkp=bkp, regions=regions, regData=regDat)
    dat <- sim$profile
    wHet <- which(dat$genotype==1/2)
    colGG <- colG
    colGG[wHet] <- hetCol
    plotSeg(dat, sim$bkp, ylims=ylims, col=colGG)
}
```

Affymetrix data

```
ds <- "GSE29172"
```

```
tf <- 1
plotFUN(ds, tf)
```

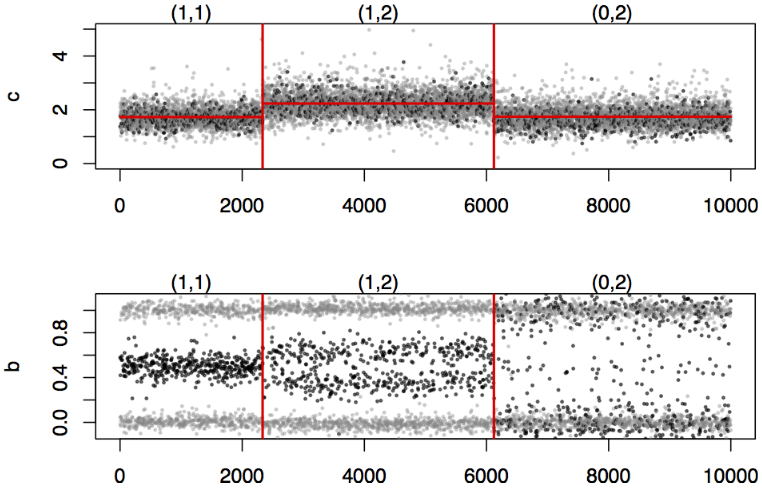


Figure E.3 – Data set GSE29172, 100% tumor cells

Another resampling

```
tf <- 1
plotFUN(ds, tf)
```

For 70% of tumor cells

```
tf <- .7
plotFUN(ds, tf)
```

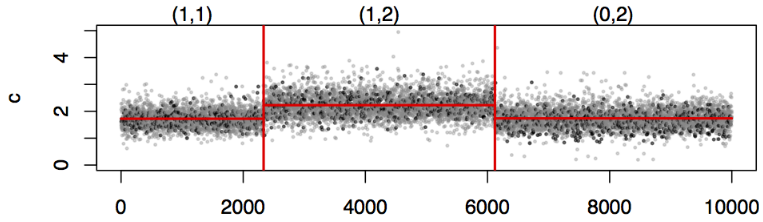
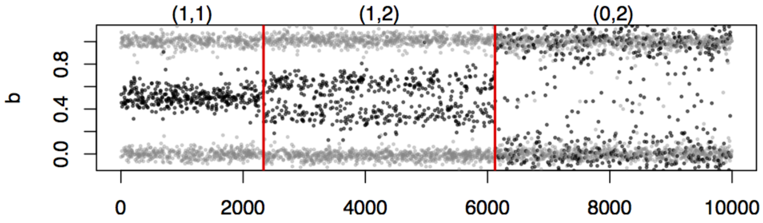



Figure E.4 – Data set GSE29172, 100% tumor cells (another resampling)

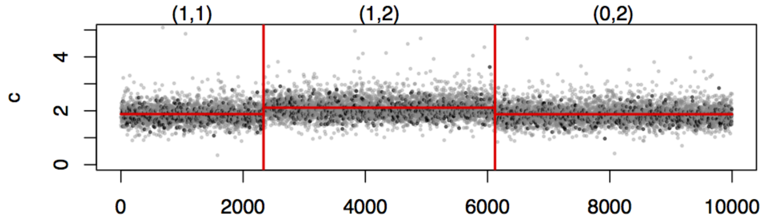
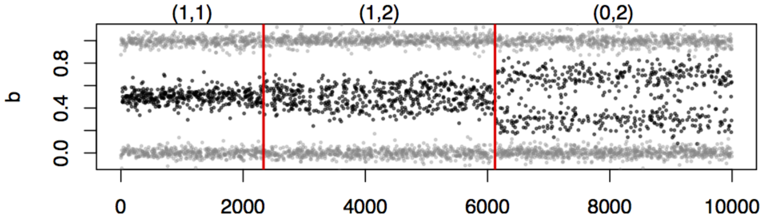



Figure E.5 – Data set GSE29172, 70% tumor cells

For 50% of tumor cells

```
tf <- .5
plotFUN(ds, tf)
```

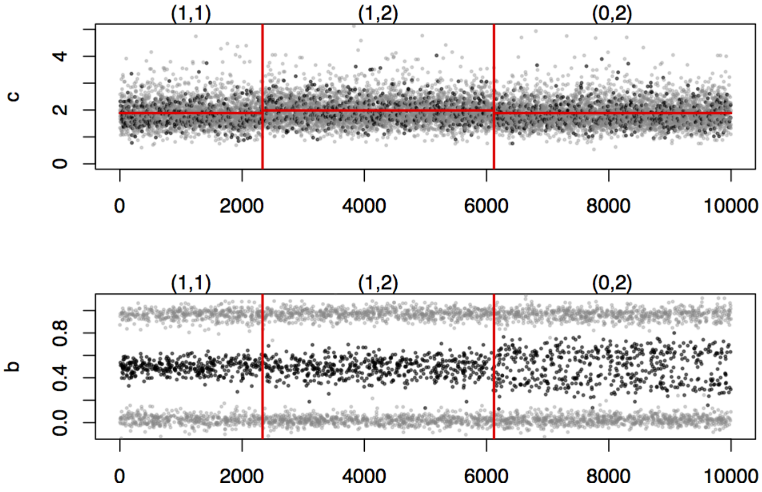


Figure E.6 – Data set GSE29172, 50% tumor cells

Illumina data

```
ds <- "GSE11976"
```

```
tf <- 1
plotFUN(ds, tf)
```

```
tf <- 1
plotFUN(ds, tf)
```

```
tf <- .79
plotFUN(ds, tf)
```

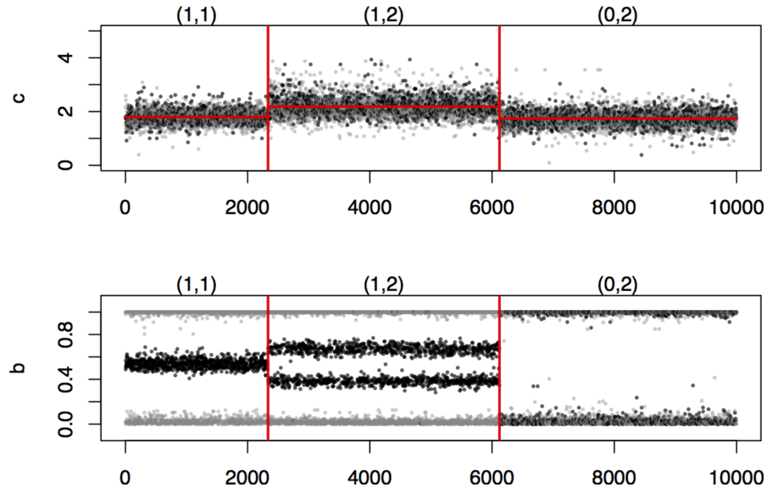


Figure E.7 – Data set GSE11976, 100% tumor cells

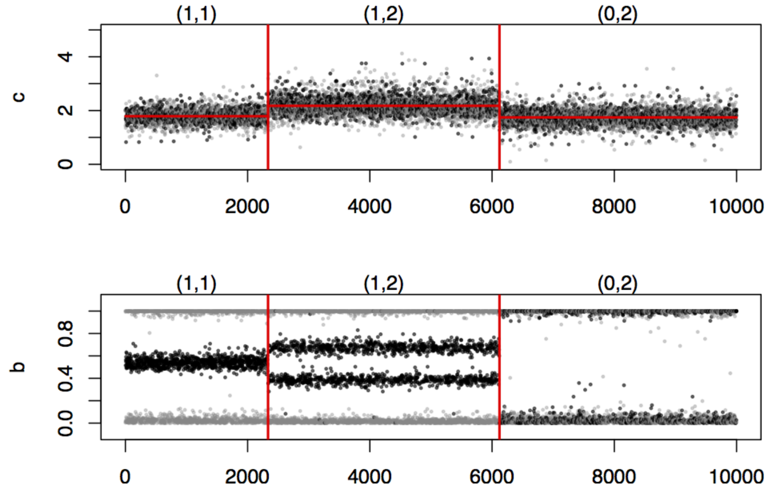


Figure E.8 – Data set GSE11976, 100% tumor cells (another reasampling

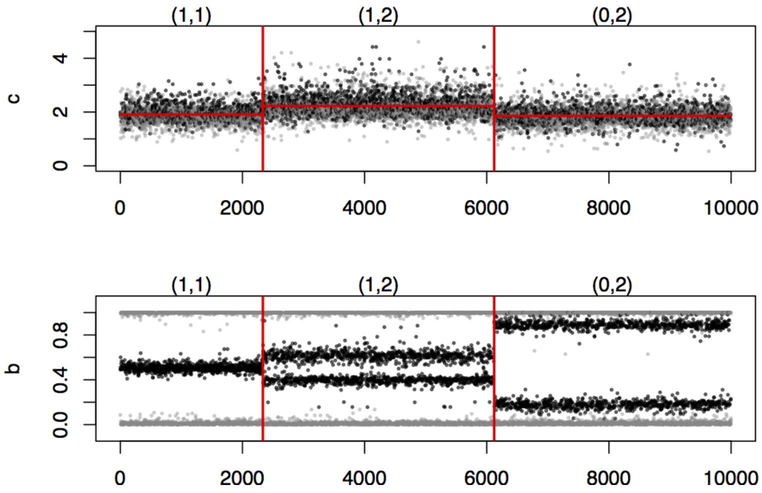


Figure E.9 – Data set GSE11976, 79% tumor cells

```r
tf <- .5
plotFUN(ds, tf)
```

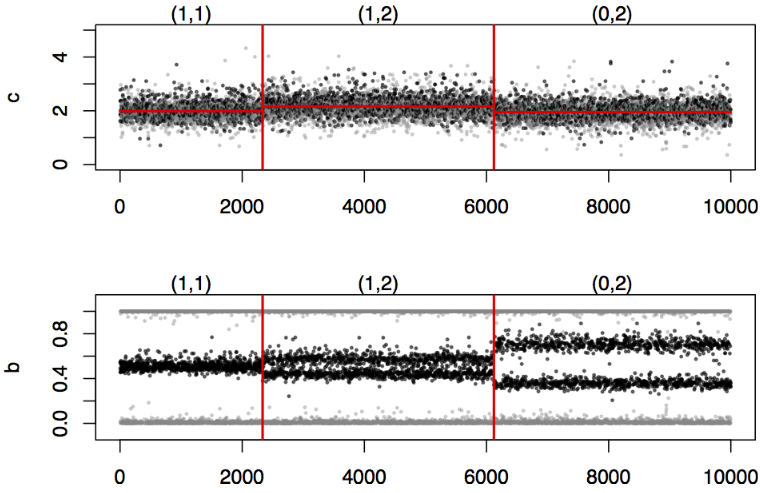


Figure E.10 – Data set GSE11976, 50% tumor cells

E.2 Joint segmentation vignette

In this section we describe how to use the `jointseg` package to partition bivariate DNA copy number signals from SNP array data into segments of constant parent-specific copy number.

We demonstrate the use of the PSSeg function of this package for applying two different strategies. Both strategies consist in first identifying a list of candidate change points through a fast (greedy) segmentation method, and then to prune this list is using dynamic programming [Bellman, 1961]. The segmentation method presented here is Recursive Binary Segmentation (RBS, [Gey and Lebarbier, 2008]). We refer to chapter 3 for a more comprehensive performance assessment of this method and other segmentation methods.

The package is presented in the form of vignette.

E.2.1 Preparing data to be segmented

PSSeg requires normalized copy number signals, in the form of total copy number estimates and allele B fractions for tumor, the (germline) genotype of SNP. Loci are assumed to come from a single chromosome and to be ordered by genomic position.

For illustration, we show of such a data set may be created from real data. We use data from a public SNP array data set, which is distributed in the acnr package (on which the jointseg package depends).

```
library(jointseg)
```

```
data <- loadCnRegionData(dataSet="GSE29172", tumorFraction=1)
str(data)
```

```
## 'data.frame':    104720 obs. of  4 variables:
##  $ c       : num  1.137 1.304 0.959 0.832 1.09 ...
##  $ b       : num  1.034 0.44 0.954 NaN NaN ...
##  $ genotype: num  0.5 0.5 1 NA NA 1 NA NA NA 1 ...
##  $ region  : chr  "(0,1)" "(0,1)" "(0,1)" "(0,1)" ...
```

This data set consists of copy number signals from length(unique(data[["region"]])) types of genomic regions:

```
table(data[["region"]])
```

```
##
## (0,1) (0,2) (0,3) (1,1) (1,2) (1,3) (2,2) (2,3)
## 13090 13090 13090 13090 13090 13090 13090 13090
```

These regions are coded as (C_1, C_2), where C_1 denotes the minor copy number and C_2 denotes the major copy number, i.e. the smallest and the largest of the two parental copy numbers (see e.g. [Neuvial et al., 2011] for more detailed definitions). For example, $(1, 1)$ corresponds to a normal state, $(0, 1)$ to an hemizygous deletion, $(1, 2)$ to a single copy gain and $(0, 2)$ to a copy-neutral LOH (loss of heterowygosity).

```
idxs <- sort(sample(1:nrow(data), 2e4))
plotSeg(data[idxs, ])
```

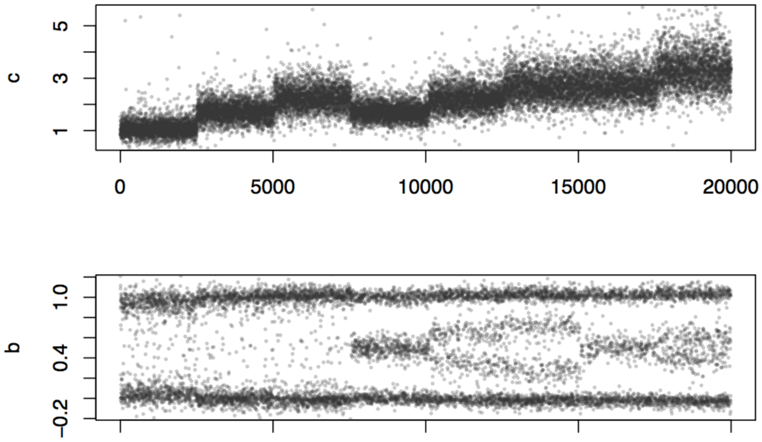


Figure E.11 – Annotated data set

These real data can then be used to create a realistic DNA copy number profile of user-defined length, and harboring a user-defined number of breakpoints. This is done using the `getCopyNumberDataByResampling` function. Breakpoint positions are drawn uniformly) among all possible loci. Between two breakpoints, the copy number state corresponds to one of the types of regions in `data`, and each data point is drawn with replacement from the corresponding true copy number signal from the region. More options are available from the documentation of `getCopyNumberDataByResampling`.

```
K <- 10
bkp <- c(408,1632,3905, 5890,6709, 10481, 12647,14089,17345,18657)
len <- 2e4
sim <- getCopyNumberDataByResampling(len, bkp=bkp, minLength=500, regData=data)
datS <- sim$profile
str(datS)
```

```
## 'data.frame':    20000 obs. of  4 variables:
```

```
##  $ c       : num  2.36 2.46 2.19 2.61 2.1 ...
##  $ b       : num  0.015 NaN NaN 0.683 NaN NaN NaN NaN 0.04 NaN ...
##  $ genotype: num  0 NA NA 0.5 NA NA NA NA 0 NA ...
##  $ region  : chr  "(1,2)" "(1,2)" "(1,2)" "(1,2)" ...
```

The resulting copy-number profile is plotted below.

```
plotSeg(datS, sim$bkp)
```

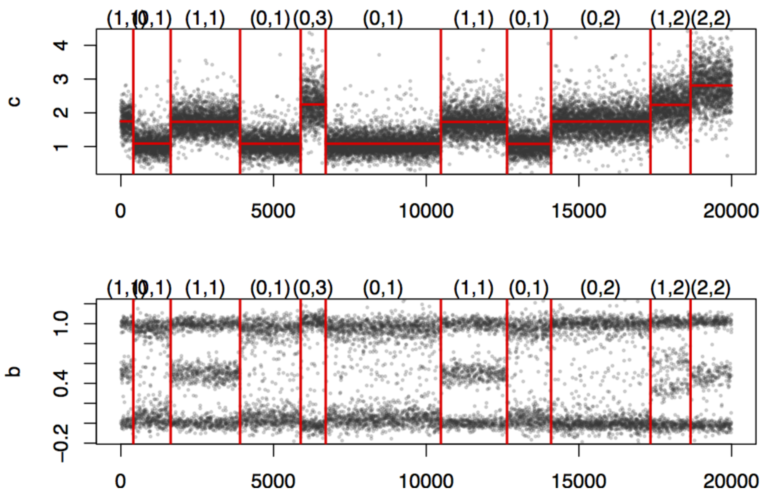


Figure E.12 – Copy number profile

E.2.2 Preprocessing

We advise the following (typical) preprocessing before segmentation:

1. log-transform total copy numbers in order to stabilize their variance; this step improve segmentation results for all methods.

   ```
   datS$c <- log2(datS$c)-1
   ```

2. smooth single point outliers as suggested by [Olshen et al., 2004]. This step is controlled by the `dropOutliers` option in the `PSSeg` function, which internally calls the `smooth.CNA` function of the `DNAcopy package`. The default value for this option is `TRUE`.

3. convert allelic ratios to (unimodal) decrease in heterozygosity (d), as initially suggested by [Staaf et al., 2008]. This step is performed internally in the `PSSeg` function.

E.2.3 PSSeg segmentation using RBS

We can now use the `PSSeg` function to segment signals. The method consists in three steps:

1. run a fast (yet approximate) segmentation on these signals in order to obtain a set of (at most hundreds of) candidate change points. This is done using Recursive Binary Segmentation (RBS) [Gey and Lebarbier, 2008];
2. prune the obtained set of change points using dynamic programming [Bellman, 1961]
3. select the best number of change points using a model selection criterion proposed by [Lebarbier, 2005]

E.2.4 Initial segmentation and pruning

```
resRBS <- PSSeg(data=datS, K=2*K, method="RBS", stat=c("c", "d"), profile=TRUE)
```

Note that this is fast:

```
resRBS$prof[, "time"]
```

```
## segmentation        dpseg
##         0.22         0.00
```

E.2.5 Plot segmented profile

To plot the PSSeg segmentation results together with the true breakpoints, do :

```
plotSeg(datS, list(true=sim$bkp, est=resRBS$bestBkp))
```

E.2.6 Results evaluation

The `PSSeg` function returns the original segmentation (by `RBS`), the result of the pruning step, and the best model (among those selected by dynamic programming) according to the criterion proposed by [Lebarbier, 2005].

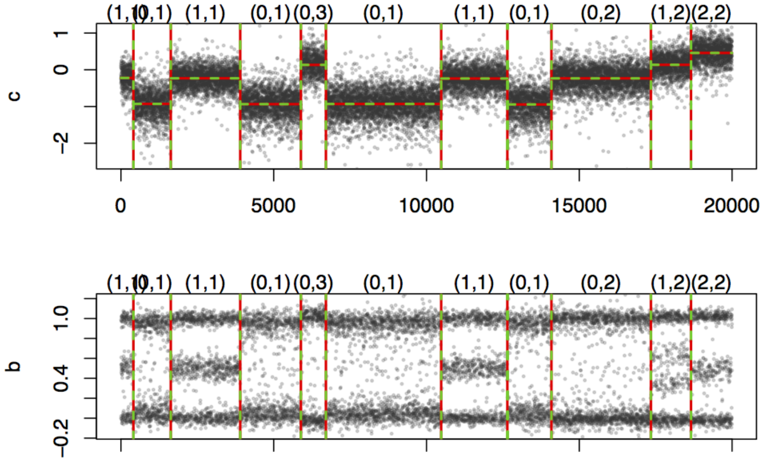


Figure E.13 – Copy number profile with RBS segmentation

The quality of the best segmentation can be assessed as follows. The number of true positives (TP) is the number of true change points for which there exists a candidate change point closer than a given tolerance `tol`. The number of false positives is defined as the number of true negatives (all those which are not change points) for which the candidate change points are out of tolerance area and those in tolerance area where there already exists a candidate change point. %The true negative rate (TNR) is defined as 1-FPR. % True negative are defined as the midpoints of intervals between true change points (augmented by points 0 and $n + 1$, where n is the number of loci. The true negative rate (TNR) is the proportion of true negatives for which there is no candidate change point closer than `tol`. By construction, $TP \in \{0, 1, \cdots, S\}$ where S is the number of true change points.

```
print(getTpFp(resRBS$bestBkp, sim$bkp, tol=5))
```

```
## TP FP
##  7  3
```

Obviously, this performance measure depends on the chosen tolerance:

```
perf <- sapply(0:10, FUN=function(tol) {
    getTpFp(resRBS$bestBkp, sim$bkp, tol=tol,relax = -1)
})
print(perf)
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
## TP    2    4    4    6    7    7    8    8    8     8     8
## FP    8    6    6    4    3    3    2    2    2     2     2
```

E.3 InCaSCN vignette

E.3.1 Introduction

This section provides a brief tutorial on using the `InCaSCN` package. The package implements a constraint dictionary learning problem to recover subclones across several patients or samples from SNP data. The InCaSCN model is designed to identify regions of copy number variation (CNV) in multi-sample SNPs data. More details on this model are given in Chapter 6.

E.3.2 Create Simulations

In order to permit the users to reproduce results of the Chapter 6, the package included the functions that permit to simulate heterogeneous samples. Therefore, the package permits to create an artificial dataset from real data. The first step is to load an annotated dataset saved in `acnr` package.

Then, after defining characteristics, we can create artificial latent profiles.

```
dataAnnotTP <- loadCnRegionData(dataSet="GSE11976", tumorFrac=1)
dataAnnotN <- loadCnRegionData(dataSet="GSE11976", tumorFrac=0)
len <- 500*10
nbClones <- 3
bkps <- list(c(100,250)*10, c(150,400)*10,c(150,400)*10)
regions <-list(c("(0,1)", "(0,2)","(1,2)"), c("(1,1)", "(0,1)","(1,1)"), c("(0,2)", "(0,
datSubClone <- buildSubclones(len, dataAnnotTP, dataAnnotN, nbClones, bkps, regions)
```

Example with a second dataset.

```
dataAnnotTP <- loadCnRegionData(dataSet="GSE13372", tumorFraction=1)
dataAnnotN <- loadCnRegionData(dataSet="GSE13372", tumorFraction=0)
datSubClone2 <- buildSubclones(len, dataAnnotTP, dataAnnotN, nbClones, bkps, regions)
```

Once subclones are created, it is also easy to generate a matrix W in order to build mixtures.

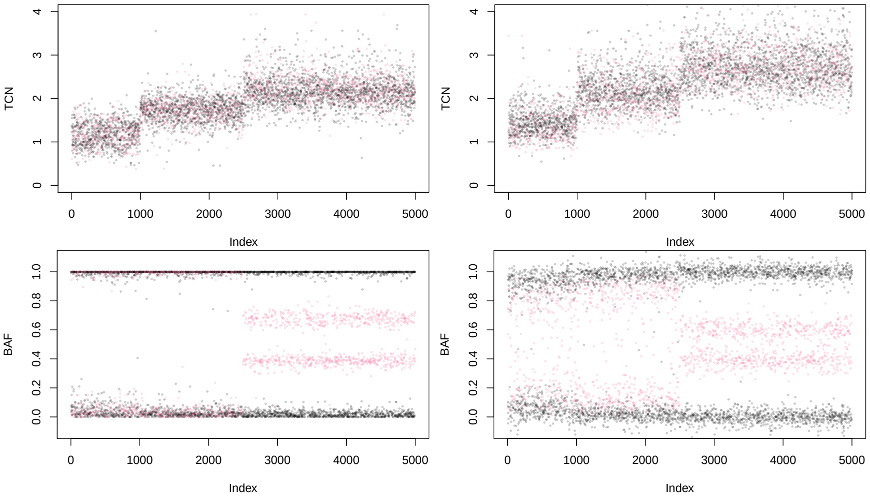


Figure E.14 – Subclone examples with two different annotated data sets

```r
W = getWeightMatrix(70,30, nb.arch = 3, nb.samp = 20)
dat <- apply(W, 1, mixSubclones, subClones=datSubClone, fracN=NULL)
str(dat[[1]])
```

```
## 'data.frame':    5000 obs. of  7 variables:
##  $ c1      : num  NA NA 0.473 0.472 NA ...
##  $ c2      : num  NA NA 1.48 1.4 NA ...
##  $ tcn     : num  1.33 1.85 1.95 1.87 1.93 ...
##  $ dh      : num  NA NA 0.515 0.495 NA ...
##  $ genotype: num  0 1 0.5 0.5 1 1 0 0.5 1 0 ...
##  $ chr     : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ pos     : int  1 2 3 4 5 6 7 8 9 10 ...
```

Note that dat is a list of data frame with the following required columns : `c1,c2,tcn,dh,genotype`

E.3.3 Run InCaSCN model

Then it is easy to run InCaSCN model on the data. We choose the same grid for λ_1 and λ_2 and a grid from 2 to 6 for the number of subclones. Intermediate results for each model are automatically saved in a directory named by default `resultsInCaSCN`. It is possible to change the name of directory by set the `output.dir` parameter. For each p `InCaSCN`, we only saved the results of the model provided by the combination (λ_1, λ_2) which minimize the BIC. Because the PVE becomes too poor, we automatically stop the algorithm at $p = 4$ and we do not consider the following models.

```
lambda1.grid <- lambda2.grid <- c(0.005,0.001)
casRes <- InCaSCN(dat,lambda1.grid, lambda2.grid, nb.arch.grid = 2:6)
```

```
## [1] 2
## [1] 3
## [1] 4
```

```
casResTCN <- InCaSCN(dat,lambda1.grid, lambda2.grid, nb.arch.grid = 2:6, stat="TCN")
```

```
## [1] 2
## [1] 3
## [1] 4
```

The next step is to choose the best p (number of subclones). In this example, it seems that the best is $\hat{p} = 4$ (which is the true number of subclones).

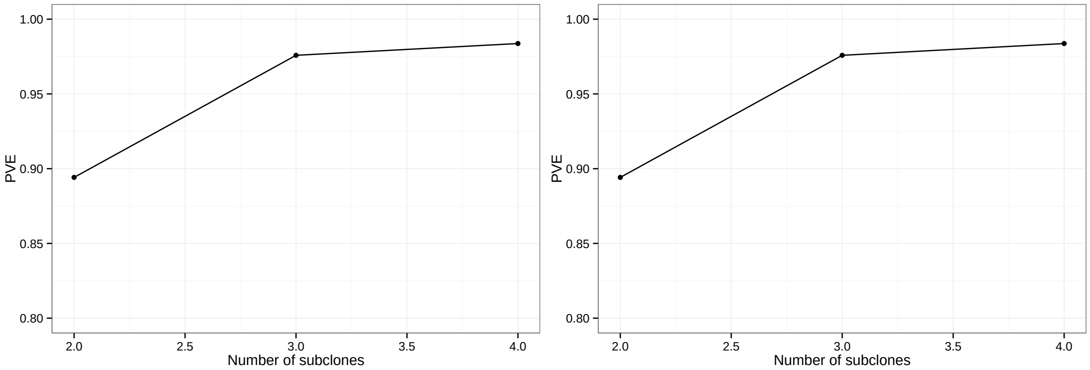


Figure E.15 – PVE (TCN and C1,C2)

We can compare the true and the estimated matrices of the weights. Even if the computation is not perfect, we can easily recover a classification close to the truth with the inferred weight matrix.

```
## List of 5
##  $ BIC  : num -211
##  $ PVE  : num 0.983
##  $ res  :List of 5
##   ..$ Z    : num [1:7, 1:3] 1.95 1.95 1.54 1.55 1.55 ...
##   ..$ Z1   : num [1:7, 1:3] 0.884 0.884 0.409 0.42 0.421 ...
##   ..$ Z2   : num [1:7, 1:3] 1.07 1.07 1.13 1.13 1.13 ...
##   ..$ W    : num [1:20, 1:3] 0.266 0.718 0.375 0.438 0.342 ...
##   ..$ Y.hat:List of 2
##   .. ..$ Y1: num [1:20, 1:7] 0.434 0.758 0.487 0.527 0.5 ...
##   .. ..$ Y2: num [1:20, 1:7] 1.4 1.1 1.38 1.35 1.33 ...
##  $ param:List of 3
##   ..$ nb.arch: int 3
##   ..$ lambda1: num 0.001
##   ..$ lamda2 : num 0.001
##  $ bkp  :List of 1
##   ..$ : num [1:6] 990 1500 1508 1508 2502 ...
```

By looking at the estimations of the latent profiles, we recover the simulated latent profiles with their respective specific alterations (Table E.2 and Fig.E.17).

Latent Profile	Region 1	Region 2	Region 3	True latent Profile
1	(0,1)	(0,2)	(1,2)	1
2	(1,1)	(0,1)	(1,1)	2
3	(0,2)	(0,1)	(1,1)	3

Table E.2 – Summarize of alterations for estimations of latent profiles

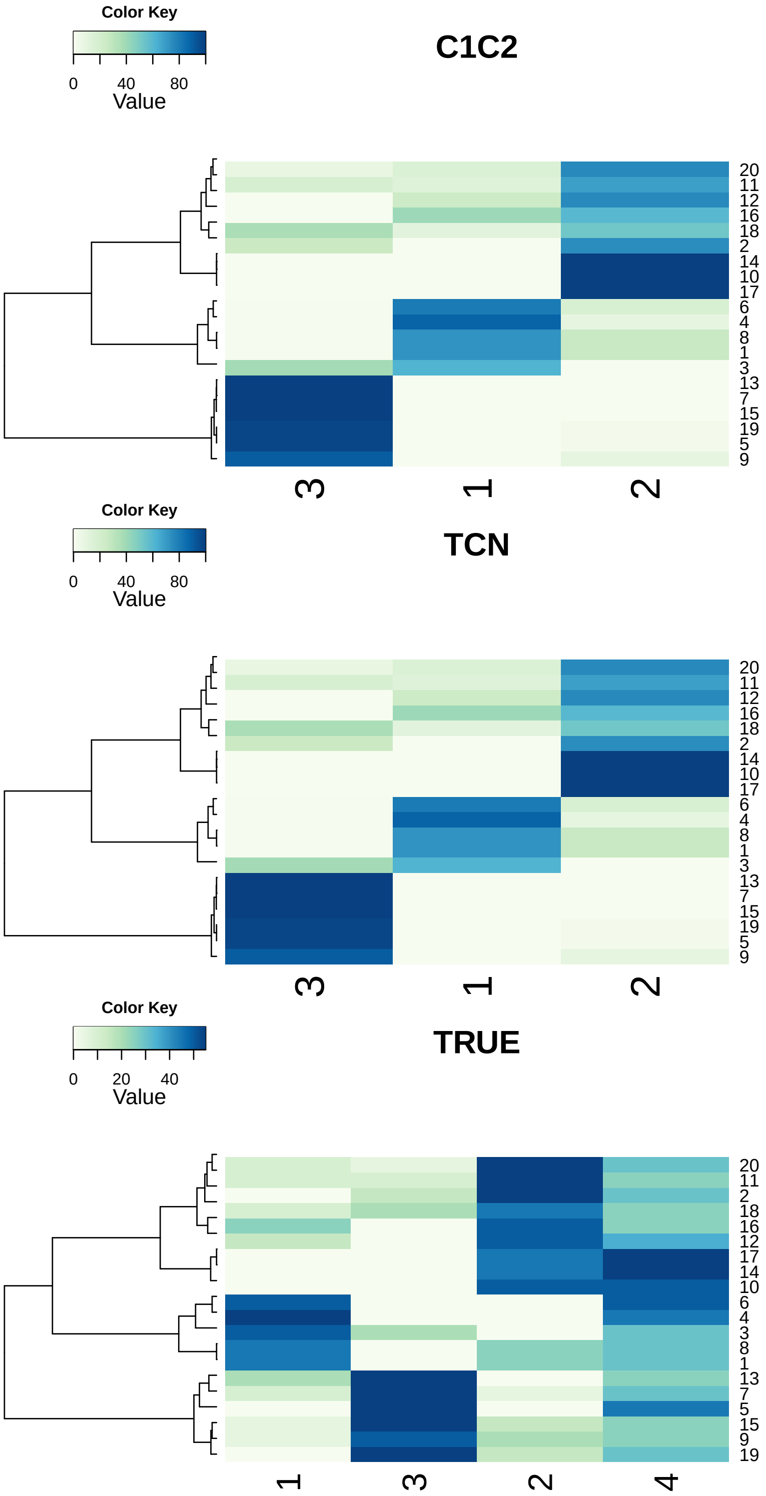


Figure E.16 – Weight matrix (C1-C2, TCN and Truth)

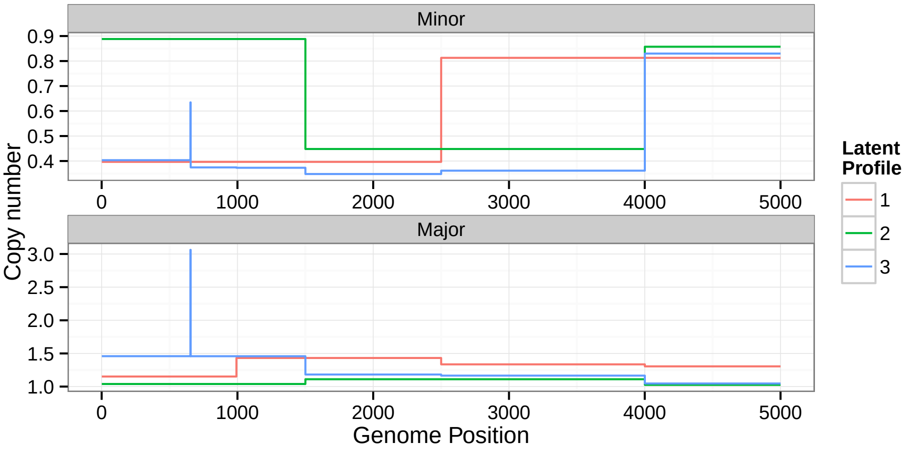


Figure E.17 – Estimated Latent profiles

E.4 Conclusion

We present in this chapter the `R` packages implemented during this thesis. All these packages aim to analyze the DNA copy number data from SNPs and CGH arrays. With adaptive `R` scripts, it is also possible to run segmentation method but also InCaSCN method on sequencing data as it has been presented in section 7.2. In addition, all packages presented here are or will be soon available at `https://github.com/mpierrejean`. For all packages, we make sure to produce user-friendly packages for the bionformatic and biostatistic community with reproducible simulations.

Glossary

Anthracycline Anticancer drug developed to treat several types of cancer. 127

FASTQ A text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores. 128

neoadjuvant Chemotherapy treatment performed before surgical extraction of a tumor. 127

read A raw sequence that comes off a sequencing machine. A read may consist of multiple segments. For sequencing data, reads are indexed by the order in which they are sequenced.. 7, 127

Taxanes First drugs to treat non-hormone responsive breast cancer. 127

transcript A sequence of RNA produced by transcription.. 7

Acronyms

AIC Akaike Information Criterion. 42

BAF B allele fraction. 7, 10, 13

BIC Bayes Information Criterion. 42

CGH Comparative Genomic Hybridization. 5

cn-LOH Copy neutral loss of heterozygosity. 10

DNA Deoxyribonucleic acid. 1, 5, 7

DoH Decrease of heterozygosity. 11, 12, 139

DP Dynamic Programming. 25, 33, 70

ER Estrogen receptor. 126

HGSOC High grade serous ovarian cancer. 121

HTS High-throughput sequencing. 7, 14

LN Lymph node. 127

LOH Loss of Heterozygosity. 13, 100

NMF Non-negative matrix factorization. 91

PR Progesterone receptor. 126

PT Primary tumor. 127

RES Residuals. 127

RKHS Reproducible kernel hilbert space. 66

RNA Ribonucleic acid. 7

RSS Residual sum of squares. 93, 94

SNA Single-nucleotide alteration. 102

SNP Single nucleotide polymorphism. 5–7, 95

TCGA The cancer genome atlas. 171

TCN Total copy number. 10, 13

TNBC Triple negative breast cancer. 126, 170

VAF Variant allele frequency. 102

WES Whole exome sequencing. 4, 126, 153

WGS Whole genome sequencing. 4, 153

Contributions

Publications

Morgane Pierre-Jean, Julien Chiquet, and Pierre Neuvial.
A new latent model to infer copy number profiles,
In preparation.

Alain Celisse, Guillemette Marot, **Morgane Pierre-Jean**, and Rigaill Guillem.
New efficient algorithms for multiple change-points detection with kernels,
Submitted to *Computational Statistics and Data Analysis (CSDA).*

Morgane Pierre-Jean, Guillem Rigaill, and Pierre Neuvial.
Performance evaluation of DNA copy number segmentation methods.
Briefings in bioinformatics, 16(4):600–615, 2015.

Seminars and communications

Nov 2016 Non parametric DNA copy number segmentation using kernels,
Poster at Statistical Analysis of Massive Genomic Data, Evry

Apr 2015 Non parametric DNA copy number segmentation using kernels,
Statistics and Genomics Seminar invited by Sandrine Dudoit, UC Berkeley (USA)

Jan. 2015 Non parametric DNA copy number segmentation using kernels,
Poster at SMPGD (Statistical Methods for Post-Genomic Data), München

June. 2013 R package jointSeg: Segmentation de données génomiques en cancérologie,
2èmes Rencontres R, Lyon

May. 2013 Change-point detection with kernel methods : application to DNA copy number signals,
45èmes journées de la SFDS, Toulouse

Jan. 2013 Joint segmentation of total DNA copy number and allelic ratios in cancer samples,
SMPGD (Statistical Methods for Post-Genomic Data), Amsterdam

Nov. 2012 Segmentation de données génomiques en cancérologie,
Journée Annuelle du groupe Biopharmacie et Santé de la SFDS, Paris

Software

2016 `InCaSCN`, `R`: This package enables to infer heterogeneity from DNA copy numbers signals.

Jan. 2013 `acnr`, `R` package: This package contains annotated data set used to evaluate performance of segmentation methods. `https://github.com/mpierrejean/acnr`

Jan. 2013 `jointseg`, `R` package: This package implements functions to quickly segment multivariate signals into piecewise constant profiles and a framework to generate realistic copy-number profiles. A typical application is the joint segmentation of total DNA copy numbers and allelic ratios obtained from Single Nucleotide Polymorphism (SNP) microarrays in cancer studies. `https://github.com/mpierrejean/jointSeg`

Bibliography

B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular biology of the cell*. New York: Garland Science, 2010. 1

K. C. Amarasinghe, J. Li, and S. K. Halgamuge. CoNVEX: copy number variation estimation in exome sequencing data using HMM. *BMC bioinformatics*, 14(Suppl 2):S2, 2013. 153

C. L. Andersen, C. Wiuf, M. Kruhoffer, M. Korsgaard, S. Laurberg, and T. F. Orntoft. Frequent occurrence of uniparental disomy in colorectal cancer. *Carcinogenesis*, 28(1):38–48, 2006. doi: 10.1093/carcin/bgl086. URL `http://carcin.oxfordjournals.org/content/28/1/38.abstract`. 101

S. Anjum, S. Morganella, F. D'Angelo, A. Iavarone, and M. Ceccarelli. VEGAWES: variational segmentation on whole exome sequencing for copy number detection. *BMC bioinformatics*, 16(1):315, 2015. 153

S. Arlot, A. Celisse, and Z. Harchaoui. Kernel change-point detection. Technical Report hal-00671174, HAL, 2012. URL `http://hal.archives-ouvertes.fr/hal-00671174`. 65, 71

I. E. Auger and C. E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of mathematical biology*, 51(1):39–54, 1989. 25, 75

F. Bach. Sharp analysis of low-rank kernel matrix approximations. In *In Proc. COLT, 2013*, 2013. 71, 74, 75

L. Badea. Extracting gene expression profiles common to colon and pancreatic adenocarcinoma using simultaneous nonnegative matrix factorization. In *Pacific Symposium on Biocomputing*, volume 290, pages 279–290. Citeseer, 2008. 3, 92

E. Barillot, L. Calzone, P. Hupe, J.-P. Vert, and A. Zinovyev. *Computational systems biology of cancer*. CRC Press, 2012. 5

N. Beerenwinkel, R. F. Schwarz, M. Gerstung, and F. Markowetz. Cancer Evolution: Mathematical Models and Computational Inference. *Systematic Biology*, 2014. doi: 10.1093/sysbio/syu081. URL http://sysbio.oxfordjournals.org/content/early/2014/11/24/sysbio.syu081.abstract. 14, 89

M. Behr and A. Munk. Identifiability for Blind Source Separation of Multiple Finite Alphabet Linear Mixtures. *arXiv preprint arXiv:1505.05272*, 2015. 94

M. Behr et al. Multiscale inference for blind demixing with applications in cancer genetics. Oral communication at SMPGD 2016, Lille, France, 2016. 94

R. Bellman. On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4(6):284, 1961. 25, 192, 195

E. Ben-Yaacov and Y. C. Eldar. A fast and flexible method for the segmentation of aCGH data. *Bioinformatics*, 24(16):i139—-i145, Aug. 2008. 22

H. Bengtsson, P. Wirapati, and T. P. Speed. A single-array preprocessing method for estimating full-resolution raw copy numbers from all Affymetrix arrays including GenomeWideSNP 5 & 6. *Bioinformatics*, 27(17):2149–2156, 2009. 184

H. Bengtsson, P. Neuvial, and T. P. Speed. TumorBoost: Normalization of allele-specific tumor copy numbers from a single pair of tumor-normal genotyping microarrays. *BMC bioinformatics*, 11(1):245, May 2010. 11, 12, 58, 184

Y. Benjamini and T. P. Speed. Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic acids research*, page gks001, 2012. 14, 154

L. Birgé and P. Massart. Gaussian model selection. *Journal of the European Mathematical Society*, 3:203–268, 2001. ISSN 1435-9855. doi: 10.1007/s100970100031. URL http://dx.doi.org/10.1007/s100970100031. 42, 77

K. Bleakley and J.-P. Vert. The group fused Lasso for multiple change-point detection. Technical report, http://hal.archives-ouvertes.fr/hal-00602121/en, June 2011. 53, 54, 104

V. Boeva, T. Popova, K. Bleakley, P. Chiche, J. Cappo, G. Schleiermacher, I. Janoueix-Lerosey, O. Delattre, and E. Barillot. Control-FREEC: a tool for assessing copy number and allelic content using next-generation sequencing data. *Bioinformatics*, 28(3):423–425, 2012. 153

J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the national academy of sciences*, 101(12):4164–4169, 2004. 91, 92

M. D. Burstein, A. Tsimelzon, G. M. Poage, K. R. Covington, A. Contreras, S. A. Fuqua, M. I. Savage, C. K. Osborne, S. G. Hilsenbeck, J. C. Chang, et al. Comprehensive genomic analysis identifies novel subtypes and targets of triple-negative breast cancer. *Clinical Cancer Research*, 21(7):1688–1698, 2015. 126, 129

Cancer Genome Atlas Research Network and others. Integrated genomic analyses of ovarian carcinoma. *Nature*, 474(7353):609–615, 2011. 52

H. Chen, H. Xing, and N. R. Zhang. Estimation of parent specific DNA copy number in tumors using high-density genotyping arrays. *PLoS Computational Biology*, 7(1): e1001060, 2011. 21, 34, 47, 53, 101

S. A. Chowdhury, S. E. Shackney, K. Heselmeyer-Haddad, T. Ried, A. A. Schäffer, and R. Schwartz. Phylogenetic analysis of multiprobe fluorescence in situ hybridization data from tumor cell populations. *Bioinformatics*, 29(13):i189–i198, 2013. 14

A. Cleynen, M. Koskas, E. Lebarbier, G. Rigaill, and S. Robin. Segmentor3IsBack: an R package for the fast and exact segmentation of Seq-data. *Algorithms for Molecular Biology*, 9:6, 2014. 27

E. A. Collisson, A. Sadanandam, P. Olson, W. J. Gibb, M. Truitt, S. Gu, J. Cooc, J. Weinkle, G. E. Kim, L. Jakkula, et al. Subtypes of pancreatic ductal adenocarcinoma and their differing responses to therapy. *Nature medicine*, 17(4):500–503, 2011. 3

G. Cooper. *The Cell: A Molecular Approach*. 2nd edition. Sunderland (MA), 2000. URL http://www.ncbi.nlm.nih.gov/books/NBK9963/. 3

A. Cutler and L. Breiman. Archetypal analysis. *Technometrics*, 36(4):338–347, 1994. 91

B. Daviss. Growing pains for metabolomics: the newest'omic science is producing results–and more data than researchers know what to do with. *The Scientist*, 19(8): 25–29, 2005. 2

K. V. den Meersche, K. Soetaert, and D. V. Oevelen. xsample(): An R Function for Sampling Linear Inverse Problems. *Journal of Statistical Software, Code Snippets*, 30(1):1–15, 2009. ISSN 1548-7660. URL http://www.jstatsoft.org/v30/c01. 105

P. Drineas and M. W. Mahoney. On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. *JMLR*, 6(2153–2175), 2005. 74, 75

R. Edgar, M. Domrachev, and A. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210, 2002. 183, 184

B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004. 32

F. Favero, T. Joshi, A. M. Marquard, N. J. Birkbak, M. Krzystanek, Q. Li, Z. Szallasi, and A. C. Eklund. Sequenza: allele-specific copy number and mutation profiles from tumor sequencing data. *Annals of Oncology*, 26(1):64–70, 2015. 154

S. Fine, K. Scheinberg, N. Cristianini, J. Shawe-taylor, and B. Williamson. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001. 74

J. Fridlyand. Hidden Markov models approach to the analysis of array-CGH data. *Journal of Multivariate Analysis*, 90(1):132–153, July 2004. ISSN 0047259X. doi: 10.1016/j.jmva.2004.02.008. URL http://linkinghub.elsevier.com/retrieve/pii/S0047259X04000260. 21, 22, 23

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010. 106

P. Fryzlewicz. Wild Binary Segmentation for Multiple Change-Point Detection. *Annals of Statistics, to appear*, 2012. 27

R. Gaujoux and C. Seoighe. A flexible R package for nonnegative matrix factorization. *BMC bioinformatics*, 11(1):367, 2010. 92

S. Gey and E. Lebarbier. Using CART to detect multiple change points in the mean for large sample. Technical report, Statistics for Systems Biology research group, 2008. 33, 34, 53, 54, 104, 192, 195

C. D. Greenman, G. Bignell, A. Butler, S. Edkins, J. Hinton, D. Beare, S. Swamy, T. Santarius, L. Chen, S. Widaa, P. A. Futreal, and M. R. Stratton. PICNIC: an algorithm to predict absolute allelic copy number variation with microarray cancer data. *Biostatistics*, 11(1):164–175, 2010. 21, 101

C. D. Greenman, E. D. Pleasance, S. Newman, F. Yang, B. Fu, S. Nik-Zainal, D. Jones, K. W. Lau, N. Carter, P. A. Edwards, et al. Estimation of rearrangement phylogeny for cancer genomes. *Genome research*, 22(2):346–361, 2012. 14

G. Ha, A. Roth, J. Khattra, J. Ho, D. Yap, L. M. Prentice, N. Melnyk, A. McPherson, A. Bashashati, E. Laks, et al. TITAN: inference of copy number architectures in clonal cell populations from tumor whole-genome sequence data. *Genome research*, 24(11):1881–1893, 2014. 95

P. Hall, J. Kay, and D. Titterinton. Asymptotically optimal difference-based estimation of variance in nonparametric regression. *Biometrika*, 77(3):521–528, 1990. 25

D. Hanahan and R. A. Weinberg. Hallmarks of cancer: the next generation. *cell*, 144 (5):646–674, 2011. 3

Z. Harchaoui and O. Cappé. Retrospective multiple change-point estimation with kernels. In *IEEE Workshop on Statistical Signal Processing*, 2007. 66, 71, 73

Z. Harchaoui and C. Lévy-Leduc. Catching change-points with lasso. *Advances in Neural Information Processing Systems*, 20:161–168, 2008. 31, 32, 33, 34, 53, 54, 104

A. R. Hautaniemi, S. Kauraniemi, P. Yli-Harja, O. Astola, J. Wolf, and M. Kallioniemi. A CGH-plotter: MATLAB toolbox for CGH-data analysis. *Bioinformatics*, 13(1714–1715), 2003. 24

T. Hocking, G. Schleiermacher, I. Janoueix-Lerosey, V. Boeva, J. Cappo, O. Delattre, F. Bach, and J.-P. Vert. Learning smoothing models of copy number profiles using breakpoint annotations. *BMC Bioinformatics*, 14(1):164, 2013. 46, 47, 48, 54, 55, 56

W. Huber, J. Toedling, and L. M. Steinmetz. Transcript mapping with high-density oligonucleotide tiling arrays. *Bioinformatics*, 22(16):1963–1970, 2006. doi: 10.1093/bioinformatics/btl289. URL `http://bioinformatics.oxfordjournals.org/content/22/16/1963.abstract`. 34

P. Hupé, N. Stransky, J.-P. Thiery, F. Radvanyi, and E. Barillot. Analysis of array-CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, 20 (18):3413–3422, 2004. 22, 24

Illumina, inc. BeadStudio Genotyping Module v3.2. User Guide. 184

Illumina, inc. Illumina's Genotyping Data Normalization Methods. White paper, 2006. 184

K. Inamura, T. Fujiwara, Y. Hoshida, T. Isagawa, M. H. Jones, C. Virtanen, M. Shimane, Y. Satoh, S. Okumura, K. Nakagawa, et al. Two subclasses of lung squamous cell carcinoma with different gene expression profiles and prognosis identified by hierarchical clustering and non-negative matrix factorization. *Oncogene*, 24(47): 7105–7113, 2005. 3

Y. Jiang, D. A. Oldridge, S. J. Diskin, and N. R. Zhang. CODEX: a normalization and copy number variation detection method for whole exome sequencing. *Nucleic acids research*, 43(6):e39–e39, 2015. 153

Y. Jiang, Y. Qiu, A. J. Minn, and N. R. Zhang. Assessing intratumor heterogeneity and tracking longitudinal and spatial clonal evolutionary history by next-generation sequencing. *Proceedings of the National Academy of Sciences*, 113(37):E5528–E5537, 2016. doi: 10.1073/pnas.1522203113. URL `http://www.pnas.org/content/113/37/E5528.abstract`. 15, 95, 98, 101, 102, 103, 104

L. Kadalayil, S. Rafiq, M. J. Rose-Zerilli, R. J. Pengelly, H. Parker, D. Oscier, J. C. Strefford, W. J. Tapper, J. Gibson, S. Ennis, et al. Exome sequence read depth methods for identifying copy number changes. *Briefings in bioinformatics*, page bbu027, 2014. 153

A. Kallioniemi, O. P. Kallioniemi, D. Sudar, D. Rutovitz, J. W. Gray, F. Waldman, and D. Pinkel. Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors. *Science*, 258(5083):818–821, 1992. 5

R. Killick, P. Fearnhead, and I. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012. 27

H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23 (12):1495–1502, 2007. 92

D. C. Koboldt, Q. Zhang, D. E. Larson, D. Shen, M. D. McLellan, L. Lin, C. A. Miller, E. R. Mardis, L. Ding, and R. K. Wilson. Varscan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome research*, 22 (3):568–576, 2012. 153

N. Krumm, P. H. Sudmant, A. Ko, B. J. O'Roak, M. Malig, B. P. Coe, A. R. Quinlan, D. A. Nickerson, E. E. Eichler, N. E. S. Project, et al. Copy number variation detection and genotyping from exome sequence data. *Genome research*, 22(8):1525–1532, 2012. 153

R. Kurman. Origin and molecular pathogenesis of ovarian high-grade serous carcinoma. *Annals of Oncology*, 24(suppl 10):x16–x21, 2013. 121

W. R. Lai, M. D. Johnson, R. Kucherlapati, and P. J. Park. Comparative analysis of algorithms for identifying amplifications and deletions in array-CGH data.

Bioinformatics (Oxford, England), 21(19):3763–70, Oct. 2005. ISSN 1367-4803. doi: 10.1093/bioinformatics/bti611. URL http://www.ncbi.nlm.nih.gov/pubmed/16081473. 46, 54, 55

E. Lebarbier. Detecting multiple change-points in the mean of gaussian process by model selection. *Signal Process.*, 85(4):717–736, Apr. 2005. ISSN 0165-1684. doi: 10.1016/j.sigpro.2004.11.012. URL http://dx.doi.org/10.1016/j.sigpro.2004.11.012. 42, 66, 79, 195

D. D. Lee and H. S. Seung. Algorithms for Non-negative Matrix Factorization. In *Advances in neural information processing systems*, pages 556–562, 2001. 92

B. Li and J. Z. Li. A general framework for analyzing tumor subclonality using SNP array and DNA sequencing data. *Genome biology*, 15(9):473, 2014. 95

H. Li and R. Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009. doi: 10.1093/bioinformatics/btp324. URL http://bioinformatics.oxfordjournals.org/content/25/14/1754.abstract. 127

A. Magi, L. Tattini, I. Cifola, R. D'Aurizio, M. Benelli, E. Mangano, C. Battaglia, E. Bonora, A. Kurg, M. Seri, et al. EXCAVATOR: detecting copy number variants from whole-exome sequencing data. *Genome biology*, 14(10):1, 2013. 153

A. Marusyk and K. Polyak. Tumor heterogeneity: causes and consequences. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1805(1):105–117, 2010. 3

A. Marusyk, V. Almendro, and K. Polyak. Intra-tumour heterogeneity: a looking glass for cancer? *Nature Reviews Cancer*, 12(5):323–334, 2012. 3

S. Masecchia, S. Salzo, A. Barla, and A. Verri. A dictionary learning based method for aCGH segmentation. In *Proceedings of the European Symposium on Artificial Neural Networks*, 2013. 97, 98, 99, 100, 101, 104, 106, 108

R. Melcher, O. Al-Taie, T. Kudlich, E. Hartmann, S. Maisch, C. Steinlein, M. Schmid, A. Rosenwald, T. Menzel, W. Scheppach, et al. SNP-array genotyping and spectral karyotyping reveal uniparental disomy as early mutational event in MSS-and MSI-colorectal cancer cell lines. *Cytogenetic and genome research*, 118(2-4):214–221, 2007. 101

R. Melcher, E. Hartmann, W. Zopf, S. Herterich, P. Wilke, L. Muller, E. Rosler, T. Kudlich, O. Al-Taie, A. Rosenwald, T. Katzenberger, B. Scholtka, S. Seibold, D. Rogoll, W. Scheppach, M. Scheurlen, and H. Luhrs. LOH and copy neutral LOH (cnLOH) act as alternative mechanism in sporadic colorectal cancers with chromosomal and microsatellite instability. *Carcinogenesis*, 32(4):636–642, 2011. doi: 10.1093/carcin/bgr011. URL http://carcin.oxfordjournals.org/content/32/4/636.abstract. 101

D. S. Moore. *The developing genome: An introduction to behavioral epigenetics*. Oxford University Press, 2015. 1

D. Mosen-Ansorena and A. M. Aransay. Bivariate segmentation of SNP-array data for allele-specific copy number analysis in tumour samples. *BMC Bioinformatics*, 14 (1):84, 2013. ISSN 1471-2105. doi: 10.1186/1471-2105-14-84. URL http://www.biomedcentral.com/1471-2105/14/84. 34, 53

D. Mosén-Ansorena, A. Aransay, and N. Rodríguez-Ezpeleta. Comparison of methods to detect copy number alterations in cancer using simulated and real genotyping data. *BMC bioinformatics*, 13(1):192, 2012. 46, 48, 54

N. Navin, A. Krasnitz, L. Rodgers, K. Cook, J. Meth, J. Kendall, M. Riggs, Y. Eberling, J. Troge, V. Grubor, et al. Inferring tumor progression from genomic heterogeneity. *Genome research*, 20(1):68–80, 2010. 3

N. Navin, J. Kendall, J. Troge, P. Andrews, L. Rodgers, J. McIndoo, K. Cook, A. Stepansky, D. Levy, D. Esposito, et al. Tumour evolution inferred by single-cell sequencing. *Nature*, 472(7341):90–94, 2011. 89

P. Neuvial, H. Bengtsson, and T. P. Speed. Statistical analysis of single nucleotide polymorphism microarrays in cancer studies. In *Handbook of Statistical Bioinformatics*, Springer Handbooks of Computational Statistics, pages 225–255. Springer, 1st edition, Mar. 2011. 10, 12, 13, 21, 101, 103, 184, 193

G. Nowak, T. Hastie, J. R. Pollack, and R. Tibshirani. A fused lasso latent feature model for analyzing multi-sample aCGH data. *Biostatistics*, page kxr012, 2011. 97, 98, 99, 100, 101, 104, 106, 108, 115

L. Oesper, A. Mahmoody, and B. J. Raphael. THetA: inferring intra-tumor heterogeneity from high-throughput DNA sequencing data. *Genome Biol*, 14(7):R80, 2013. 3, 95

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997. 91

A. B. Olshen, E. S. Venkatraman, R. Lucito, and M. Wigler. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4): 557–572, 2004. 22, 24, 27, 52, 194

A. B. Olshen, H. Bengtsson, P. Neuvial, P. T. Spellman, R. A. Olshen, and V. E. Seshan. Parent-specific copy number in paired tumor-normal studies using circular binary segmentation. *Bioinformatics*, 27(15):2038–2046, Aug 2011. 47, 53, 97, 101

D. A. Peiffer, J. M. Le, F. J. Steemers, W. Chang, T. Jenniges, F. Garcia, K. Haden, J. Li, C. A. Shaw, J. Belmont, S. W. Cheung, R. M. Shen, D. L. Barker, and K. L. Gunderson. High-resolution genomic profiling of chromosomal aberrations using Infinium whole-genome genotyping. *Genome Res.*, 16(9):1136–1148, Sept. 2006. doi: 10.1101/gr.5402306. URL `http://dx.doi.org/10.1101/gr.5402306`. 184

F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J.-J. Daudin. A statistical approach for array-CGH data analysis. *BMC bioinformatics*, 6:27, Jan. 2005. ISSN 1471-2105. doi: 10.1186/1471-2105-6-27. URL `http://www.ncbi.nlm.nih.gov/pubmed/15705208`. 21, 24, 42, 46

M. Pierre-Jean, G. Rigaill, and P. Neuvial. Performance evaluation of DNA copy number segmentation methods. *Briefings in Bioinformatics*, 16(4):600–615, 2015. doi: 10.1093/bib/bbu026. URL `http://bib.oxfordjournals.org/content/16/4/600.abstract`. 98, 104, 186

Q. Qi, Y. Zhao, M. Li, and R. Simon. Non-negative matrix factorization of gene expression profiles: a plug-in for BRB-ArrayTools. *Bioinformatics*, 25(4):545–547, 2009. 91

P. M. Rancoita, M. Hutter, F. Bertoni, and I. Kwee. An integrated Bayesian analysis of LOH and copy number data. *BMC bioinformatics*, 11(1):321, 2010. 47

W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971. 113

M. Rasmussen, M. Sundstrom, H. Goransson Kultima, J. Botling, P. Micke, H. Birgisson, B. Glimelius, and A. Isaksson. Allele-specific copy number analysis of tumor samples with aneuploidy and tumor heterogeneity. *Genome Biol*, 12(10):R108–R108, 2011. 3, 183

M. Reimers and J. N. Weinstein. Quality assessment of microarrays: visualization of spatial artifacts and quantitation of regional biases. *Bmc Bioinformatics*, 6(1):1, 2005. 14

G. Rigaill. A pruned dynamic programming algorithm to recover the best segmentations with 1 to k_max change-points. *Journal de la Société Française de Statistique*, 156 (4):180–205, 2015. 27, 34, 53

G. Rigaill, T. D. Hocking, F. Bach, and J.-P. Vert. Learning sparse penalties for change-point detection using max margin interval regression. *Proceedings of the 30th International Conference on Machine Learning,* JMLR W&CP, 28(3):172–180, 2013. 47

M. D. Ritchie, E. R. Holzinger, R. Li, S. A. Pendergrass, and D. Kim. Methods of integrating data to uncover genotype-phenotype interactions. *Nature Reviews Genetics*, 16(2):85–97, 2015. 1

A. Roth, J. Khattra, D. Yap, A. Wan, E. Laks, J. Biele, G. Ha, S. Aparicio, A. Bouchard-Côté, and S. P. Shah. PyClone: statistical inference of clonal population structure in cancer. *Nature methods*, 11(4):396–398, 2014. 95

J. F. Sathirapongsasuti, H. Lee, B. A. Horst, G. Brunner, A. J. Cochran, S. Binder, J. Quackenbush, and S. F. Nelson. Exome sequencing-based copy-number variation and loss of heterozygosity detection: ExomeCNV. *Bioinformatics*, 27(19):2648–2654, 2011. 153

M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467, 1995. 5

B. Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel methods in computational biology*. MIT press, 2004. 66

E. C. Schwalbe, D. Williamson, J. C. Lindsey, D. Hamilton, S. L. Ryan, H. Megahed, M. Garami, P. Hauser, B. Dembowska-Baginska, D. Perek, et al. DNA methylation profiling of medulloblastoma allows robust subclassification and improved outcome prediction using formalin-fixed biopsies. *Acta neuropathologica*, 125(3):359–371, 2013. 91

G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2): 461–464, 1978. 108

R. F. Schwarz, A. Trinh, B. Sipos, J. D. Brenton, N. Goldman, and F. Markowetz. Phylogenetic quantification of intra-tumour heterogeneity. *PLoS Comput Biol*, 10 (4):e1003535, 2014. 14

R. F. Schwarz, C. K. Ng, S. L. Cooke, S. Newman, J. Temple, A. M. Piskorz, D. Gale, K. Sayal, M. Murtaza, P. J. Baldwin, et al. Spatial and temporal heterogeneity in high-grade serous ovarian cancer: a phylogenetic analysis. *PLoS Med*, 12(2):e1001789, 2015. 97, 121, 122, 134, 139, 171

A. Sen and M. Srivastava. On tests for detecting change in mean. *The Annals of Statistics*, 3(1):98–108, 1975. 27

J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004. 66

S. T. Sherry, M.-H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and K. Sirotkin. dbSNP: the NCBI database of genetic variation. *Nucleic acids research*, 29(1):308–311, 2001. 155

D. Skvortsov, D. Abdueva, C. Curtis, B. Schaub, and S. Tavare. Explaining differences in saturation levels for affymetrix genechip® arrays. *Nucleic acids research*, 35(12): 4154–4163, 2007. 95

A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. 2000. 74

J. Staaf, D. Lindgren, J. Vallon-Christersson, A. Isaksson, and et al. Segmentation-based detection of allelic imbalance and loss-of-heterozygosity in cancer cells using whole genome SNP arrays. *Genome Biol*, 9(9):R136, Oct. 2008. 11, 12, 47, 48, 54, 101, 146, 184, 194

I. Steinwart, D. Hush, and C. Scovel. An explicit description of the reproducing kernel hilbert spaces of gaussian rbf kernels. *IEEE Transactions on Information Theory*, 52(10):4635–4643, 2006. 177

W. Sun, F. A. Wright, Z. Tang, S. H. Nordgard, P. V. Loo, T. Yu, V. N. Kristensen, and C. M. Perou. Integrated study of copy number states and genotype calls using high-density SNP arrays. *Nucl. Acids Res.*, 37(16):5365–5377, Sept. 2009. 21

E. Talevich, A. H. Shain, T. Botton, and B. C. Bastian. CNVkit: Genome-Wide Copy Number Detection and Visualization from Targeted DNA Sequencing. *PLoS Comput Biol*, 12(4):e1004873, 2016. 153

L. Taslaman and B. Nilsson. A framework for regularized non-negative matrix factorization, with application to the analysis of gene expression data. *PloS one*, 7(11): e46331, 2012. 91

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 32

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2005.00490.x. URL `http://dx.doi.org/10.1111/j.1467-9868.2005.00490.x`. 22

A. M. Tillmann. On the computational intractability of exact and approximate dictionary learning. *IEEE Signal Processing Letters*, 22(1):45–49, 2015. 92

E. S. Venkatraman and A. B. Olshen. A faster circular binary segmentation algorithm for the analysis of array-CGH data. *Bioinformatics*, 23(6):657–663, Mar. 2007. ISSN 1460-2059 (Electronic). doi: 10.1093/bioinformatics/btl646. 53

J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, et al. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001. 7

J.-P. Vert and K. Bleakley. Fast detection of multiple change-points shared by many signals using group LARS. *Advances in Neural Information Processing Systems*, 23: 2343–2351, 2010. III, 34, 35, 36, 175, 176

P. M. Visscher, M. A. Brown, M. I. McCarthy, and J. Yang. Five years of GWAS discovery. *The American Journal of Human Genetics*, 90(1):7–24, 2012. 6

N. Wagle, C. Emery, and e. a. Berger, MF. Dissecting Therapeutic Resistance to RAF Inhibition in Melanoma by Tumor Genomic Profiling. *Journal of Clinical Oncology*, 29(22):3085–3096, 2011. doi: 10.1200/JCO.2010.33.2312. 3

D. G. Wang, J.-B. Fan, C.-J. Siao, A. Berno, P. Young, R. Sapolsky, G. Ghandour, N. Perkins, E. Winchester, J. Spencer, et al. Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome. *Science*, 280 (5366):1077–1082, 1998. 5

J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963. 106, 113

J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, C. G. A. R. Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013. 3

H. Willenbrock and J. Fridlyand. A comparison study: applying segmentation to array-CGH data for downstream analyses. *Bioinformatics*, 21(22):4084–91, Nov 2005. doi: 10.1093/bioinformatics/bti677. 46, 47, 48, 49, 54, 55

C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001. 74, 75

T. Yang. Simple binary segmentation frameworks for identifying variation in DNA copy number. *BMC bioinformatics*, 13(1):277, 2012. 27

A. Yates, W. Akanni, M. R. Amode, D. Barrell, K. Billis, D. Carvalho-Silva, C. Cummins, P. Clapham, S. Fitzgerald, L. Gil, C. G. Girón, L. Gordon, T. Hourlier, S. E. Hunt, S. H. Janacek, N. Johnson, T. Juettemann, S. Keenan, I. Lavidas, F. J. Martin, T. Maurel, W. McLaren, D. N. Murphy, R. Nag, M. Nuhn, A. Parker, M. Patricio, M. Pignatelli, M. Rahtz, H. S. Riat, D. Sheppard, K. Taylor, A. Thormann, A. Vullo, S. P. Wilder, A. Zadissa, E. Birney, J. Harrow, M. Muffato, E. Perry, M. Ruffier, G. Spudich, S. J. Trevanion, F. Cunningham, B. L. Aken, D. R. Zerbino, and P. Flicek. Ensembl 2016. *Nucleic Acids Research*, 44(D1):D710–D716, 2016. doi: 10.1093/nar/gkv1157. URL http://nar.oxfordjournals.org/content/44/D1/D710.abstract. 155

H. Zhang and L. Zhao. On the inclusion relation of reproducing kernel hilbert spaces. *CoRR*, 2011. 79, 177

J. Zhang, J. Baran, A. Cros, J. M. Guberman, S. Haider, J. Hsu, Y. Liang, E. Rivkin, J. Wang, B. Whitty, et al. International cancer genome consortium data portal—a one-stop shop for cancer genomics data. *Database*, 2011:bar026, 2011. 3

N. Zhang. DNA copy number profiling in normal and tumor genomes. In J. Feng, W. Fu, and F. Sun, editors, *Frontiers in Computational and Systems Biology*, pages 259–281. Springer-Verlag, 2010. 21

N. R. Zhang and D. O. Siegmund. A modified Bayes Information Criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63 (1):22–32, 2007. 42, 46

université
PARIS-SACLAY

ÉCOLE DOCTORALE
Structure et dynamique
des systèmes vivants (SDSV)

Titre : Développement de méthodes statistiques pour l'analyse du nombre de copies d'ADN en cancérologie

Mots clefs : Nombre de copies d'ADN, segmentation, cancérologie, dictionary learning, hétérogénéité, grande dimension.

Résumé : Les données génomiques issues d'expériences de puces à ADN ou de séquençage ont deux caractétistiques principales: leur grande dimension (le nombre de marqueurs dépassant de plusieurs ordres de grandeurs le nombre d'observations), et leur forte structuration (notamment via les dépendances entre marqueurs). La prise en compte de cette structuration est un enjeu clé pour le développement de méthodes performantes en grande dimension.

Cette thèse est axée sur les données présentant une forte structure le long du génome. C'est le cas des données de nombres de copies d'ADN, mais aussi des données de génotypes. La thèse couvre à la fois le developpement de méthodes statistiques, l'implémentation logicielle, et l'application des méthodes développées à des jeux de données réelles. Nous avons, en particulier, étudié des méthodes de segmentation, et de dictionary learning. Toutes les implémentations logiciel de ces méthodes sont librement disponibles sous forme de packages R.

Title : Development of statistical methods for DNA copy number analysis in cancerology

Keywords : DNA copy number, segmentation, cancerology, dictionary learning, heterogeneity, high dimension.

Abstract : Genomic data from DNA microarray or sequencing technologies have two major characteristics: their high dimension (number of markers larger than the number of observations), and their strong structuration (dependence between markers). Taking into account this structuration, it is a challenging issue for the development of efficient methods.

This work is focused on the data with a strong spatial structuration, namely DNA copy number data in tumor samples. We developed statistical models, software implementations and we applied these developments to real data. We explored in particular segmentation models and dictionary learning methods. All the software Implementations of these methods are freely available as R packages.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France