ORSAY
N° d'ordre : 8071

UNIVERSITÉ DE PARIS-SUD
U.F.R SCIENTIFIQUE D'ORSAY

THÈSE

présentée pour obtenir

Le GRADE de DOCTEUR EN SCIENCES
DE l'UNIVERSITÉ PARIS XI ORSAY

Spécialité : Mathématiques

par

Franck Picard

le 16 Novembre 2005

Sujet : **PROCESS SEGMENTATION/CLUSTERING. APPLICATION TO THE ANALYSIS OF CGH MICROARRAY DATA.**

M. Antoniadis Anestis	Rapporteur
M. Daudin Jean-Jacques	Directeur
M. Gascuel Olivier	Rapporteur
M. Lavielle Marc	Examinateur
Mme. Mangin Brigitte	Examinatrice
M. Massart Pascal	Président

- Remerciements -

Un exercice de remerciement n'est jamais facile. Peut-être parce qu'il comporte toujours les trois éléments suivants : merci + à qui + pourquoi, et qu'il est difficile dans ce cadre de remercier plus de cinq personnes sans que les autres abandonnent la lecture. Cet exercice est aussi délicat car il convient de diversifier les mercis: un « merci » tout court, un « merci beaucoup », « merci du fond du coeur », jusqu'au « merci pour tout » qui dispense de justification. Les mercis qui vont suivre traduisent ma profonde reconnaissance envers ceux qui m'ont accompagné pendant ces trois ans, et qui m'ont permis d'aboutir ce travail. Ils sont d'ordre professionnel ou personnel, ou les deux. Je pense particulièrement à l'équipe avec laquelle j'ai travaillé. Je pense également à ma famille et à mes lamis, ils sauront pourquoi.

★

Je remercie tout d'abord Anestis Antoniadis et Olivier Gascuel d'avoir accepté d'être les rapporteurs de ma thèse, et Marc Lavielle, Brigitte Mangin et Pascal Massart d'avoir accepté de faire partie de mon jury.

Ensuite je tiens à remercier Jean-Jacques Daudin et Stéphane Robin qui m'ont encadré pendant cette thèse. Leur confiance et leur patience m'ont permis de travailler sereinement pendant ces trois ans, malgré mes inquiétudes régulières. D'autres personnes m'ont guidé dans mon travail, et j'aimerais citer Avner Bar-Hen, Gilles Celeux, Camille Duby, Marc Lavielle et Christian Vaisse.

Je remercie également les membres de l'équipe Statistique et Génome de l'INA P-G, tout particulièrement Emilie, Tristan, Marie-Laure et Julie, pour leurs encouragements, leur enthousiasme, et pour le reste. J'aimerais également exprimer ma reconnaissance envers tous les membres du Département OMIP de l'INA P-G, avec qui j'ai eu le plaisir de travailler, dans une ambiance aussi dynamique que stimulante.

Un grand « merci pour tout » à mes parents et à ma soeur, qui m'ont soutenu pendant ces trois ans, à mes lamis, avec une mention spéciale pour Amandine [1], Julie [2] et Julien [3], pour $\mathcal{M}$ & $\mathcal{B}$ de Mézières et pour mes amis d'outre-Atlantique, Cécile, Sean, et Val.

Enfin, je remercie le lecteur qui je l'espère aura autant de plaisir à lire mon travail que j'en ai eu à l'aboutir.

★

[1] rendez-vous au Canon d'Italie, disons vers 21h30.

[2] rendez-vous dimanche prochain.

[3] rendez-vous à Bangkok, San Francisco ou Milan, c'est toi qui vois.

Contents

Introduction

Many technologies provide signals which are modelled by non stationary time-series. Among different approaches that exist for signal processing, segmentation methods have focused much attention. Their purpose is to detect abrupt changes that occur in some characteristics of the signal. This detection can be done *on-line* or *off-line*. In this work we are interested in the *off-line* detection problem which is also called the multiple change-point problem. In this setting, a signal $\{y_1, \ldots, y_n\}$ is observed, and modelled by a random process $\{Y_1, \ldots, Y_n\}$ whose probability distribution $f(\cdot)$ depends on a parameter θ. It is assumed that parameter θ is affected by $K-1$ abrupt changes at unknown instants called breakpoints, and noted $t_1 < \ldots < t_{K-1}$. These breakpoints define a partition of the data into K intervals or segments $I_1, \ldots, I_K$ of length n_k such that parameter θ is constant within an interval, and different from one interval to another. A segmentation model can be defined as follows:

$$\forall t \in I_k, \;\; Y_t \sim f(\theta_k).$$

In the multiple change-point context, the only objective is to partition the data into locally stationary time-series. However in practical situations, the characteristics of the signal may not only depend on intervals $I_1, \ldots, I_k$. An example is provided in the Gaussian case in Figure 1. When dealing with segmentation methods, the objective is to partition the data into segments for which the mean of the process is constant and equals μ_k in I_k. In the segmentation/clustering context, we suppose that the mean of the process belongs to the finite set $\{m_1, \ldots, m_P\}$. More generally, parameters $\{\theta_k\}_k$ may be constraint to take a limited number of values $\{\theta_1, \ldots, \theta_P\}$ with $P \leq K$. The fact the signal shows the same characteristics on different segments may indicate that there exists a secondary structure of the data, which is the belonging of segments to different clusters. In this context, the set of segments characterized by parameter θ_p may be interpreted using a knowledge which depends on the application field. Thus the objective of segmentation/clustering is to partition the data into K segments and to cluster the K segments into P clusters.

Segmentation/clustering problems are traditionally studied using hidden Markov models. In this thesis, I propose to develop an alternative statistical model combining segmentation models and mixture models. In this context, the density of data points within segments is supposed to be a mixture density. We note π_p the *prior* probability of belonging to cluster p for a segment and Y^k the set of data points within segment k, $Y^k = \{Y_t, \; t \in I_k\}$. If variables Y_t are assumed to be

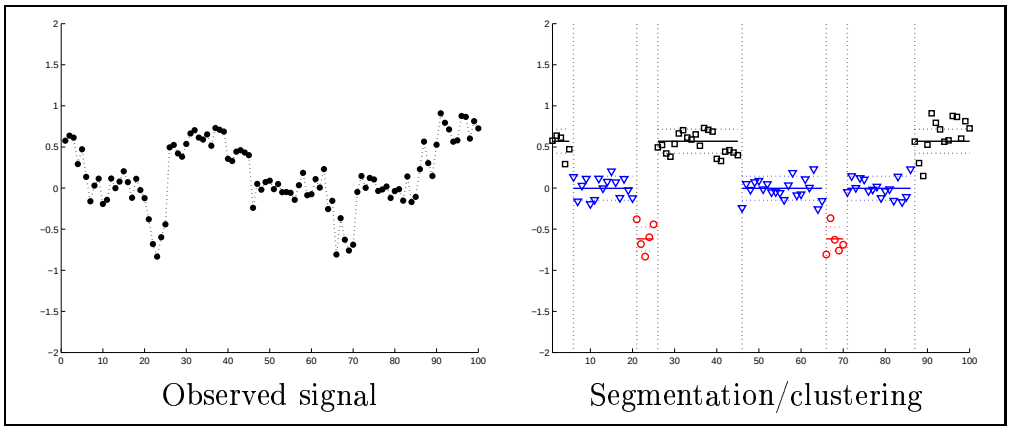


Figure 1: Illustration of segmentation/clustering.

independent, it follows that the density of vector Y^k can be written such that:

$$f(y^k;\psi) = \sum_{p=1}^{P} \pi_p \prod_{t\in I_k} f(y_t;\theta_p).$$

Thus the segmentation/clustering model can be expressed as a mixture model for which statistical units are vectors of different sizes.

I propose to construct a segmentation/clustering model in three major steps which are: (1) the determination of the parameters affected by changes and according to which the data should be clustered, (2) the choice of an estimation strategy to estimate the parameters of the model, (3) the selection of the number of clusters P and of the number of segments K.

We are focused on the development of a segmentation/clustering model in the Gaussian case where the mean and variance of segments are supposed to depend on clusters such that:

$$Y^k | Y^k \in C_p \sim \mathcal{N}\left(\mathbb{1}_{n_k} m_p, s_p^2 I_{n_k}\right).$$

We also present an extension of our model to the case where the data are discrete and modelled by Markov chains. In both cases the model is characterized by two sets of parameters: the set of breakpoint coordinates $T = \{t_1, \ldots, t_{K-1}\}$ which are discrete parameters, and the set of mixture parameters. We estimate these parameters by maximum likelihood, and we construct a hybrid algorithm for this purpose, which is based on dynamic programming and on the EM algorithm. Then we address the question of the selection of the number of clusters P and of the number of segments K. Selecting the dimension of a model has lead to the development of many statistical criteria largely based on penalized likelihoods. However our problem is original since both P and K should be selected. In this work, I propose a model selection heuristic for this choice.

Application to the analysis of genomic data

Genomics and related fields constitute a vast source of data to analyze. Recent advances in technology have allowed biologists to quantify molecular phenomena on a genomic scale, such as gene expression. Microarray technology is the most widely used technique for this purpose. It has shown its power to study the expression of thousands of genes, and has been adapted to explore other biological questions. Among these questions, gene-dosage effect has recently focused much attention since altering DNA copy number is one of the many ways that gene expression and function may be modified. For example many defects in human development are due to gains and losses of chromosomes and chromosomal segments, and DNA dosage alterations that occur in somatic cells are frequent contributors to cancer. Over the past several years array comparative genomic hybridization (array CGH) has demonstrated its value for analyzing DNA copy number variations.

While many statistical approaches have been explored for the analysis of gene-expression microarray data, the analysis of CGH microarray is an emerging field. The particularity of these data is that gene copy-numbers present a spatial coherence on the genome. To this extent, segmentation methods have been used to analyze this type of data, in order to detect genomic regions which share the same gene copy numbers on average. Nevertheless, another question which is asked is to cluster the detected regions into a finite number of clusters with biological interpretation (deleted or amplified regions for instance). This is why we apply our segmentation/clustering model to the analysis of array CGH data.

Organization

In a first Part, I present the biological context of my work, with a detailed presentation of the data under study, and I present biological problems which are currently studied using array CGH. In Part II, I propose a first method to analyse array CGH data. This method is based on segmentation methods, and has been published in Picard *et al.* (2005). Moreover, this method has been cited recently in Lai *et al.* (2005) who show its efficiency on simulated and real data sets. In Part III I develop a new statistical model for segmentation/clustering problems in the Gaussian case. This method is implemented in Part IV, and its performance is compared with hidden Markov Models which constitute the most widely used models to assess segmentation/clustering problems. This part also presents the application of our method to real CGH data. The last part of my work is devoted to an extension of the segmentation/clustering model to discrete variables, with an application to the analysis of DNA sequences.

Part I

Biological context

Chapter 1

Definition of array-based Comparative Genomic Hybridization

Since chromosomes have been demonstrated to be the physical carrier of genetic information, the study of their structure, function and evolution has become central in human genetics. This is the purpose of cytogenetics, which emerged in 1956 with the determination of the correct number of chromosomes in humans. The correct identification of each chromosome enabled the visualization and localization of chromosomal defects, that could be linked to human diseases. One classical example is the visualization of trisomies by karyotype (Figure 1.1). In 50 years, considerable efforts have been made to detect small chromosomal aberrations and the resolution of the varying techniques has evolved from several megabases with chromosome banding, to 50 kilobases with array-based comparative genomic hybridization. This increase in the resolution has given the molecular basis of known syndromes, such as Prader-Willy and Angelman syndromes as well as mental retardation, and has provided molecular portraits of numerous cancer diseases. The diagnosis of tumors has now shifted from histological analysis to molecular characterization. More than a medical-oriented approach to human cytogenetics, variations in gene copy numbers have become central to the study of genome dynamics and human evolution.

1.1 Historical perspective of human cytogenetics techniques

Chromosome banding

Human chromosomes are classically represented as they appear in metaphase during cell cycle, with 2 chromatids joined by the centromere and ended by telomeres. In 1960 Casperson *et al.* (1968) developed a staining protocol that produced highly reproducible patterns of dark and light bands along the length of each chromosome. These bands became barcodes that allowed the unique identification of each chromosome. Chromosome banding has been extensively used and a band-

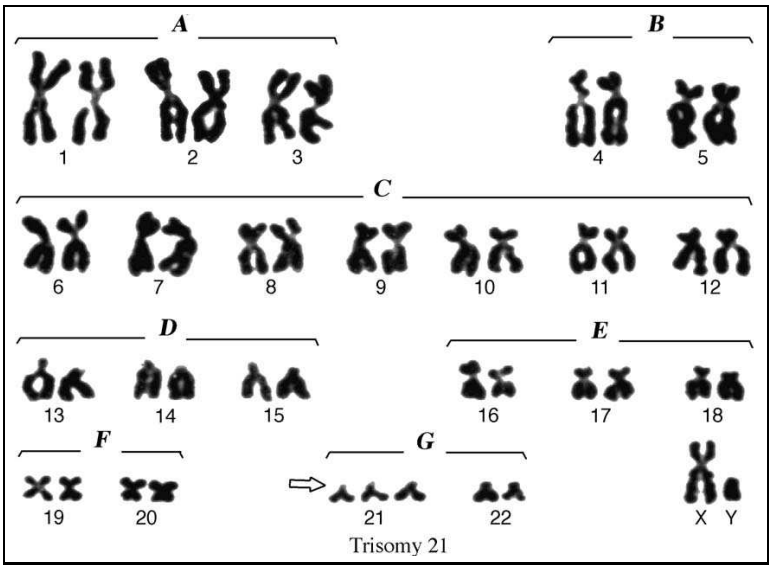


Figure 1.1: Original karyotype of a trisomy 21 (Down Syndrome) after "solid" Giemsa staining. Since it is impossible to recognize all individual chromosomes, they are subdivided into several groups (A-G and sex chromosomes) based on their total length and location of the centromere. From Smeets *et al.* (2004).

naming convention was introduced in the seventies. The banding pattern enabled the detection of various structural aberrations such as translocations, inversions, deletions and duplications, with a resolution of 500 bands (approximatively 50 genes per band). This was improved by the development of high-resolution banding (Yunis (1976)), which allowed the precise characterization of already known chromosomal aberrations, but also the detection of unnoticed subtle aberrations such as microdeletions or amplifications. Thousands of chromosomal abnormalities have been associated with inherited or *de novo* disorders, generating many clues for their molecular basis.

Fluoresence In Situ Hybridization

Despite the increase in the resolution of banding techniques, no aberration was found at the cytogenetic level for numerous patients showing clear clinical signs of syndromes. A new technique called FISH (Fluorescence In Situ Hybridization) allowed researchers to fill the gap between chromosome banding and sequence-level information. This technique is based on the molecular re-association of two complementary DNA molecules. A DNA molecule is composed of two complementary strands. Each strand can bind with its template molecule, but not with templates whose sequences are very different from its own. Hybridization techniques take advantage of this property of DNA molecules. In FISH experiments, a probe is a perfectly known and mapped sequence (a cloned piece of the genome), which is hybridized to chromosomes of a patient (see Figure 1.2). This technique allows the chromosomal and nuclear location of the probe to be seen through the microscope. As a consequence of the Human Genome Project, more and more

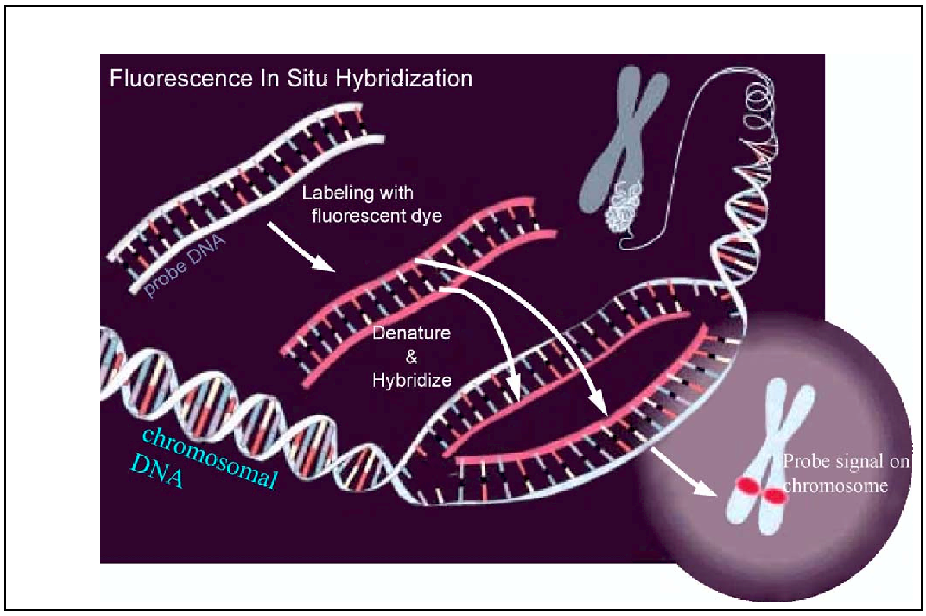


Figure 1.2: Fluorescence In Situ Hybridization (FISH). From Smeets *et al.* (2004).

probes from cloned and mapped sequences of the human genome, such as BACs (Bacterial Artificial Chromosomes) became available for diagnosis purposes. In less than 15 years, the sensitivity of FISH has improved 10,000 fold, and today, the detection of chromosomal abnormalities that involve sequences of 10kb is feasible. Nevertheless, the use of FISH requires knowledge of the probe to be studied which hampers a blind search of chromosomal aberrations.

Chromosome Comparative Genomic Hybridization

Comparative Genomic Hybridization has allowed the analysis of DNA copy number imbalances at a genomic scale in a single experiment. Two samples of genomic DNA (referred to as the sample DNA and the test DNA) are differentially labelled with distinct fluorescent dyes and competitively hybridized to a target DNA which is a normal chromosome (Kallioniemi *et al.* (1992)). Subsequently, the ratio of the intensities of the two fluorochromes is computed and its changes indicate either gain or loss of sequences in the sample DNA compared with the test DNA. Chromosome CGH is different from FISH since DNA targets come from the genomic DNA of a normal patient and from a patient to be studied, both target DNAs being hybridized on a template chromosome. A blind search of chromosomal aberrations is then feasible.

Chromosome CGH has been a powerful tool to study gene copy number imbalances in tumor tissues as it was the first technique that allowed the mapping of gene copy number imbalances at a genomic scale in a single experiment. In comparison, Southern analysis, PCR, or fluorescence *in situ* hybridization (FISH) only examine one specific chromosomal region or gene. Although chromosome

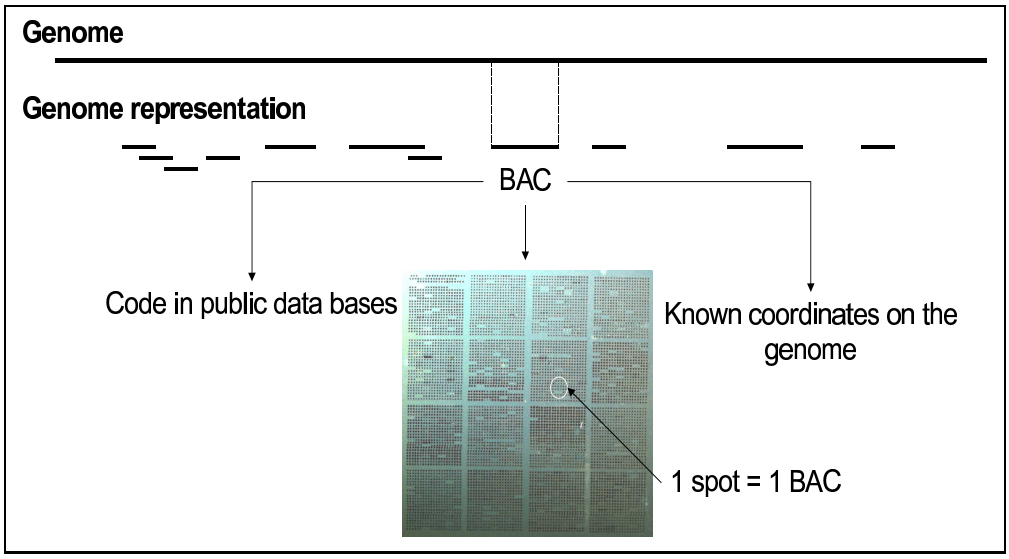


Figure 1.3: Schematic representation of array CGH conception.

CGH has become a standard method for cytogenetic studies, technical limitations restrict its usefulness as a comprehensive screening tool: the condensed and supercoiled state of the target DNA in the chromosomes limits the resolution to 10Mb for loss and 2Mb for amplification (Beheshti *et al.* (2002)). The resolution of Comparative Genomic Hybridizations has been greatly improved using microarray technology (Solinas-Toldo *et al.* (1997)).

1.2 Application of microarray technology to comparative genomic hybridization

The difference between chromosome CGH and array-based CGH lies in the support which is used for hybridization. For chromosome CGH, this support is a chromosome, whereas in CGH array experiments, the support is a slide. Since more and more DNA clones have been mapped and sequenced, they are spotted on a slide (Figure 1.3). In parallel, genomic DNA is extracted from biological samples, amplified and labelled with fluorescent dyes, called Cy3 and Cy5 (Figure 1.4). This mixture of targets, is hybridized on the chip, and DNA sequences can bind their complementary template. Since probes are uniquely localized on the slide, the quantification of the fluorescence signals on the chip will define a measurement of the abundance of thousands of genomic sequences in a cell in a given condition.

Microarray technology is well-known and widely used to study gene expression profiles. CGH microarrays use reference DNA that do not present any alteration, allowing an "absolute" quantification of genomic imbalances for the sample DNA. The application of microarray technology to CGH has improved the resolution from megabases to 100kb. Pinkel *et al.* (1998) further refined this technique and have shown that CGH microarrays can detect chromosomal aberrations of 40kb.

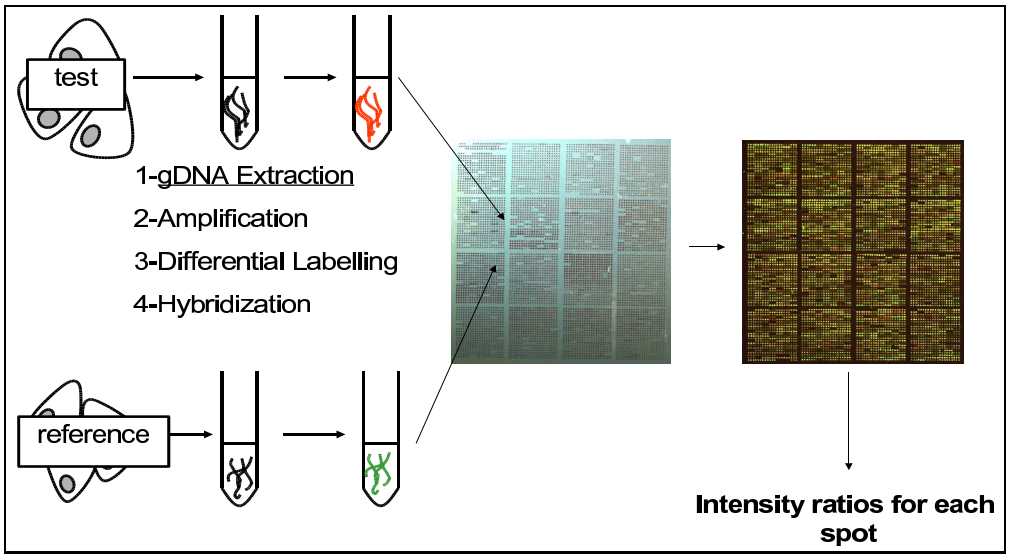


Figure 1.4: Schematic representation of array CGH experimental protocole.

The application of the microarray technology to comparative genomic hybridization has allowed three main advances in molecular cytogenetics:

- automatization of large parts of the experimental procedures,
- realization of genome-wide scans for the loss or gain of chromosomal material without looking at the subject chromosome,
- upgrade of the number of probes, with new-generation arrays consisting in approximatively 32,000 BACs (Ishkanian *et al.* (2004)).

1.3 Performance of array CGH

The microarrays we use are described in Snijders *et al.* (2001) and consist of 2460 human BACs and P1 clones in triplicate, representing approximately 7,500 spots on the arrays. Each single BAC is mapped on the genome with at least one STS (Site Tag Sequence) and all clones on the array were identified by FISH confirming 93.4% as single copy numbers. The clones offer a coverage of the 22 autosomal chromosomes and of the 2 sex chromosomes, with an average of one clone every 1.4Mb. The resolution is then defined either by the distance between targets or by the length of the cloned DNA segments.

Pinkel *et al.* (1998) studied the sensitivity of the technique and showed that fluorescence ratios were proportional to copy numbers. This was achieved by comparing cell populations containing 1 to 5 copies of the X chromosome with normal female DNA.

Although the relationship between the number of X chromosomes and the ratio of the intensities is linear, the slope differs from the theoretical expected value of 0.5, due to non specific hybridizations. Snijders *et al.* (2001) also describe

this underestimation of gene copy number imbalances, obtaining a log_2 fluorescence ratio of 0.49 ± 0.05 compared to the ideal value of 0.58 for a 3/2 ratio for trisomic chromosomal region, and a log_2 ratio of 0.72 ± 0.08 for the X chromosome in a male/female comparison, compared with the expectation of 1.0. This underestimation problem could be explained using four arguments:

- if deletions concern only part of the BACs, the resulting signals will show less dramatic differences than expected in the case of a complete BAC deletion,
- the presence of repetitive sequences depends on the individual BAC. Snijders *et al.* (2001) show that sequence characteristics of individual clones have a measurable effect on X chromosome ratios, but not on autosomal chromosome ratios,
- the efficiency to block the probes' repetitive sequences with Cot1 DNA may not be 100%,
- this underestimation could also reflect the presence of admixed normal DNA. In the case of tumor DNA extractions, tissues are composed of heterogeneous cell types resulting in a mix of different types of DNA.

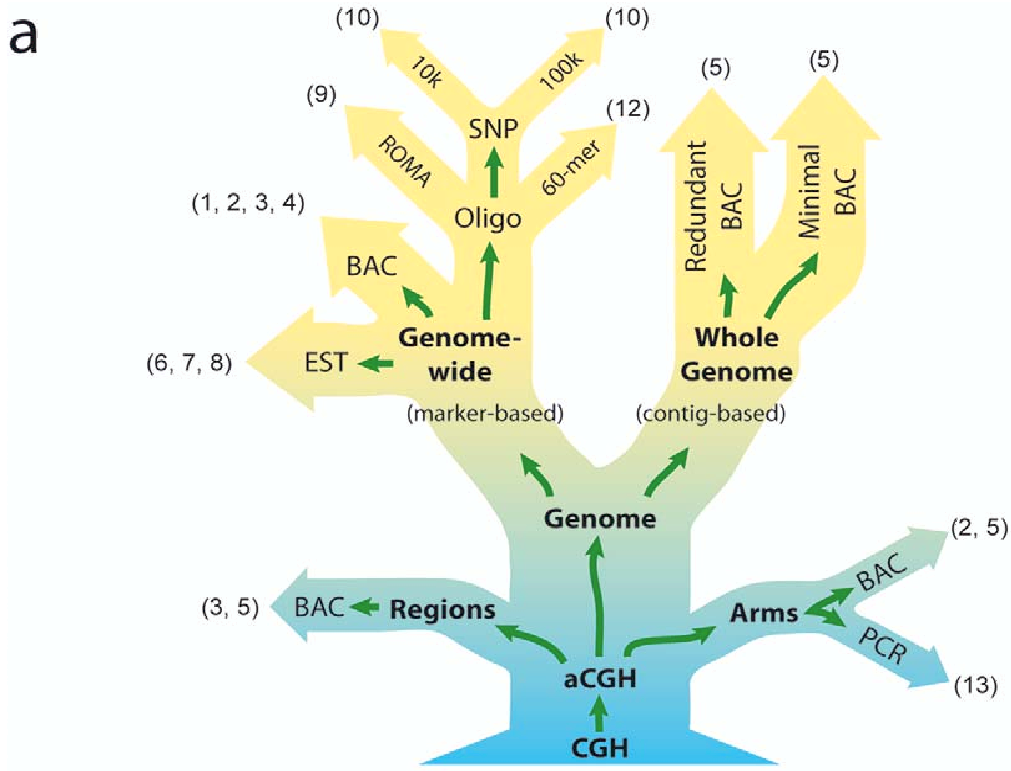
Together these points suggest that even though the Comparative Genomic Hybridization method aims at studying discrete phenomena such as gene deletions/amplifications, providing a quantitative answer in terms of presence/absence via microarray technology is not straightforward.

1.4 Diversity of CGH microarrays

Different genomic microarrays have been constructed, each being differentiated by the type of reference sequences used as target : cDNAs, oligonucleotides, and BACs. Figure 1.5 gives an overview of all possible techniques (from Davies *et al.* (2005)). Historically, Solinas-Toldo *et al.* (1997) were the first to use microarray technology based on cDNA arrays, with approximatively 3000 target clones throughout the genome. One advantage of this technique is the easy use of the same platform for gene expression measurements. The link between genomic data and expression data is then facilitated (see Pollack *et al.* (1999)). Nervertheless, this technique offers a low signal-to-noise ratio, due to the small size of cDNA clones compared with large insert clones.

Another strategy consists in the use of oligonucleoties, *ie* small sequences of 25 to 80 nucleotides. Affymetrix proposes several platforms, such as p501 arrays and Mapping 10K arrays, which contain 8473 and 11555 target probes respectively (see Davies *et al.* (2005) for a complete review). The advantage of oligonucleotide arrays is the easy link that can be made with SNPs and LOH data (Loss Of Heterozygocity), which could improve the understanding of complex events that may be found in cancer genomes. Nevertheless, the small size of oligonucleotides favors non specific hybridization to multiple genomic loci that increase the noise in the data and reduces the 30kb theoretical resolution.

The last insert clones that are used are Bacterial Artificial Chromosomes (BACs) which are large-scale inserts of hundreds of kilobases. This method has



b

Probe Type	Manufacturer	Website
(1) BAC*	Sanger	www.sanger.ac.uk/Projects/Microarrays/
(2) BAC*	UCSF	cc.ucsf.edu/microarray
(3) BAC*	DKFZ	www.dkfz-heidelberg.de/kompl_genome
(4) BAC	AFCRI	acgh.afcri.upenn.edu
(5) BAC	BCCA	www.bccrc.ca/cg/ArrayCGH_Group.html
(6) EST	BD Biosciences	www.bdbiosciences.com
(7) EST	Bioscience Corp.	www.superarray.com
(8) EST	UHN	www.microarray.ca
(9) oligo	NimbleGen Systems	www.nimblegen.com/
(10) oligo	Affymetrix	www.affymetrix.com
(11) oligo	Bioscience Corp.	www.superarray.com
(12) oligo	Agilent Technologies	we.home.agilent.com/USeng/home.html
(13) PCR*	UU	puffer.genpat.uu.se/chrom_22_array/chrom22.php

Figure 1.5: (a) Evolution of CGH arrays technologies. (b) Examples of current array platforms, BAC= bacterial artificial chromosome, EST = expressed sequence tag, USCF = university of California San Francisco, DKFZ = Deutsches Krebsforschungszentrum, AFCRI = Abramsom Family Cancer Research Institute, BCCRC = British Columbia Cancer Research Center, UHN = University Health Network, UU = Uppsala Universitet. From Davies *et al.* (2004).

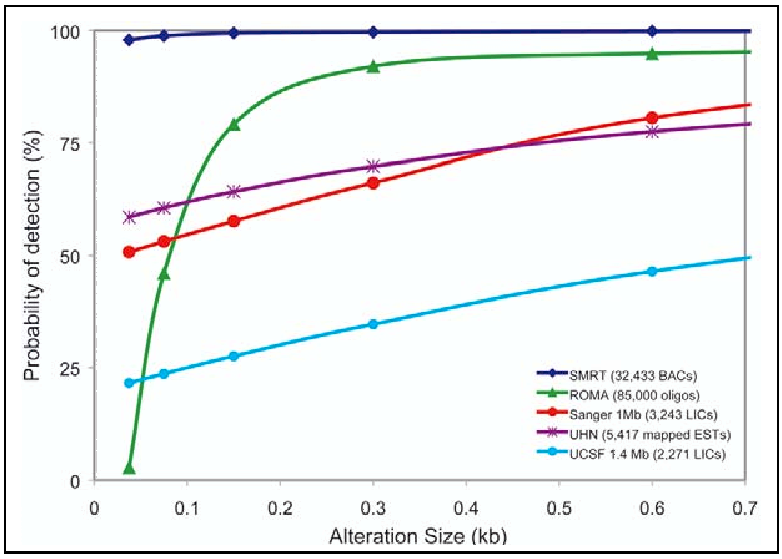


Figure 1.6: Power of different array CGH platform to detect a chromosomal alteration of a given size. From Davies *et al.* (2004).

been described previously, with BAC arrays introduced by Pinkel *et al.* (1998), with an average of 1 BAC every megabase on the genome. Recently Ishkanian *et al.* (2004) published the first SubMegabase Resolution Tiling set (SMRT arrays) that continuously covers the human genome. These new arrays offer the possibility to detect chromosomal aberrations of 40 to 80kb. Davies *et al.* (2005) compares the power of different CGH arrays to detect an alteration of varying size. Figure 1.6 shows that the probability of detecting a small aberration decreases with its size, but more interestingly, this probability increases if the size of the clones spotted on the array increases. To this extent, large clones such as BACs are more powerful in the detection of microalterations, while oligoarrays are not likely to detect aberrations smaller than 300kb. This figure shows that the SMRT arrays present the best performance, being robust to the size of the defect to be detected.

Microarray technology has offered wide possibilities for the diversification of comparative genomic hybridization techniques. While oligonucleotide and cDNA arrays offer the possibility to link other types of data to genomic alteration data, BAC and SMRT arrays constitute the most promising technology for the investigation of chromosomal alterations throughout the genome. Since SMRT arrays are recent (2004) compared with BAC arrays (1998), little is known about the statistical analysis of such data. This is why the following work will be focused on the analysis of BAC arrays exclusively.

Chapter 2

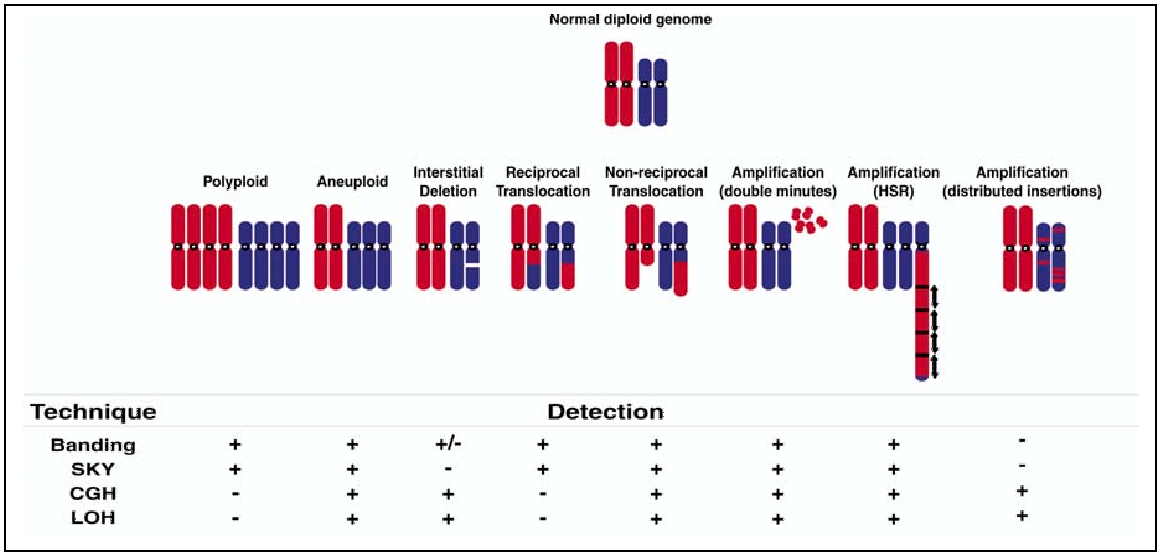
Applications of genomic microarrays in human genetics

2.1 Impact of molecular cytogenetics on human cancers

The first hypothesis that cancer was linked to chromosomal aberration drawn by Boveri in 1914 has now been demonstrated by cytogenetics. The nature of chromosomal abnormalities can concern the number of genes or the structure of chromosomes (Figure 2.1 from Albertson and Pinkel (2003)). They can be equilibrated (without quantitative abnormalities) or disequilibrated (gain or loss of genomic material). In most cases one cell carries an initial acquired genetic defect that is transmitted to the offspring cells, which may acquire new genomic defects. Since many cells may be concerned by tumor genesis, there exists a selection process that will choose favorable mutations, based on growth speed, or drug resistance for instance. Then the resulting tumor will show a diversity of chromosomal defects which are the result of a selection process. In solid tumors for instance, these alterations include altered ploidy, gain or loss of individual chromosomes or portions of chromosomes, and structural rearrangements (Albertson *et al.* (2003)).

Cytogenetics has helped in the discovery of many of those defects, and extensive catalogues are now publicly available (Mitelman *et al.* (2003), Huret *et al.* (2003)). There exists an important variability in the degree to which tumor genomes are aberrant at the chromosomal level. Some rearrangements are specific to some pathologies, but there often exists a pattern signature: a set of abnormalities which have no biological effect when isolated, but which contribute to a diagnosis when associated.

CGH arrays have greatly improved the understanding of tumor genesis and progression. Regional arrays have been used to investigate specific genomic hotspots, like the chromosome 20 arrays from Pinkel *et al.* (1998). Recently CGH arrays using overlapping BACs representative of the 1p, 3p and 5p arms of human chromosomes have been developed, for regions which are frequently altered in a variety of cancers. The different CGH arrays have now provided a molecular portrait of many cancer diseases. The reader can be referred to Albertson *et al.* (2003),



Technique	Detection							
Banding	+	+	+/-	+	+	+	+	-
SKY	+	+	-	+	+	+	+	-
CGH	-	+	+	-	+	+	+	+
LOH	-	+	+	-	+	+	+	+

Figure 2.1: Diversity of genomic alterations and detectability of chromosomal aberrations by different cytogenetic techniques. From Albertson and Pinkel (2003).

Davies *et al.* (2005), Bernheim *et al.* (2004) for a complete review of the progression in the field of cancer genomics.

2.2 New insights into human genetic variation

The understanding of the evolution and adaptation of an organism is mainly based on the study of gene diversity between and within species. Genetic variation between genomes can be introduced by mutation, that is a change in the DNA sequence of a gene within an organism, or by recombination that produces different combinations of alleles as a result of the physical exchange of DNA between chromosomes in eukaryotes. There also exist other sources of variation, such as small insertions and deletions, and a variable number of repetitive sequences.

Genetic variation can be studied with a variety of techniques, the simplest being the measure of an observed phenotype in a given population, and molecular studies are required to understand the fundamental processes that lead to differences among phenotypes. Considerable efforts have been made for the understanding of global regulation of gene expression, mainly with the development of microarray technology. Nevertheless, even if variability in phenotypes can be directly linked to differences in gene regulation, the understanding of genetic variation is crucial since a change in gene expression may result from coding sequence variability rather than difference in gene regulation.

As technical limitations have hampered any exhaustive study of large scale copy number variations throughout the genome (Armour *et al.* (2002)), their prevalence and contribution to human genetic variation have long remained unknown, or underestimated. Nevertheless, recent studies suggest that many of the

genetic differences between humans and other primates for instance are the result of large duplications and deletions Locke *et al.* (2003), indicating the importance of large-scale copy number variation in the dynamics of the genome of closely related species. The systematic search for large scale copy number variations, *ie* variation in the number of sequences of several kilobases in phenotypically normal humans, is now feasible, thanks to the use of microarray CGH and its variants.

Sebat *et al.* (2004) studied 20 unrelated individuals from different geographic backgrounds. Using a variant of microarray CGH, called ROMA (Representational Oligonucleotide Microarray Analysis), they found an average of 11 copy number polymorphisms between 2 individuals, with median length of 222kb, with half being recurrent in multiple individuals. These polymorphisms are widely spread throughout the genome, with some being clustered near putative hot-spots of genomic variation. Their location near other types of chromosomal rearrangements may reflect regions of instability on the genome. An additional study conducted by Iafrate *et al.* (2004) explored the genome of 55 individuals, and found similar results, with an average large-scale copy number variation of 12 per individual, not limited to intergenic or intronic regions. Overall, the authors have described more than 200 large scale copy number variations in human genomes, 24 of which are present in more than 10% of the individuals.

Variation in the dosage of individual genes can lead to different phenotypes and diseases. The biological impact of those copy number variations range from non selective to embryonically lethal if they affect development genes for instance. Sebat *et al.* (2004) observed copy number variation in genes involved in neurodevelopment, breast cancer, leukemia, food intake and body weight regulation. Iafrate *et al.* (2004) show that 142 of 255 polymorphic clones overlap with known coding regions and that 67 clones encompass one or more genes. 14 large copy number variations were found near loci associated with human syndromes or with cancer. Since the individuals under study were phenotypically normal, the presence of variation close to such susceptibility regions could be a source of chromosomal rearrangement that could influence the expression of specific genes.

Iafrate *et al.* (2004) as well as Sebat *et al.* (2004) indicate that the restricted number of individuals studied or the limited resolution of the technique used constitute a limitation to their study and results. Nevertheless, the authors suggest that the impact of large scale copy number variation has been underestimated, and van Ommen (2004) gives a rough estimate of the emergence of random large segment copy number polymorphism: 1:8 for deletions and 1:50 for duplications. The author concludes as follows: "Given the frequency of the emergence of random segmental duplications and deletions, they are therefore likely to contribute substantially to why we are all different."

Chapter 3

Presentation of array CGH data

Even though microarray technology was developed for the study of gene expression, it has now been extended to the study of diverse molecular biology issues, such as protein binding, interference RNAs, chromatine structure, and chromosomal aberrations for instance. Since the technology is similar, we will use a dedicated terminology. A *probe* will denote a biological "entity" or object, that is perfectly referenced, and which is spotted on a slide. In the case of array CGH, a probe is a cloned piece of the genome, whose sequence and location on the genome are known. It can be a BAC (Bacterial Artificial Chromosome), or a cDNA (complementary DNA) for instance. A *target* is the complementary "entity" of the probe, *ie* it is of the same nature, but has been extracted from a biological sample. One major difference is that the identification of one specific *target* can only be done through the hybridization *probe/target*, which is specific (in theory). In array CGH experiments, targets are obtained through the digestion of genomic DNA by restriction enzymes for instance.

The use of the same technology leads to some patterns in the analysis of microarray data in general. In this chapter, we briefly establish the issues that are raised by array CGH data analysis, like for all microarray-generated data, and we will specify the points on which our study focuses.

3.1 On the use of microarray technology

3.1.1 Image Acquisition

After biological experiments and hybridizations are performed, the fluorescence intensities are measured with a scanner. This image acquisition and data collection step can be divided into four parts (Leung and Cavalieri (2003)). The first step is image acquisition by scanners, independently for the two conditions present on the slide. The second step consists in spot recognition or gridding. Automatic procedures are used to localize the spots on the image, but a manual adjustment is often needed for the recognition of low quality spots that are flagged and often eliminated. Then the image is segmented to differentiate the foreground pixels in a spot grid from the background pixels. After the spots have been segmented, the pixel intensities within the foreground and background masks are averaged

separately to give the foreground and background intensities. After the image processing is done, the raw intensity data are extracted from the slide, independently for the test and the reference, and the data for each gene are typically reported as intensity ratios that measure the relative abundance of the targets in the test condition compared to the reference condition.

3.1.2 Experimental design

Once biological experiments are done and images are acquired, the researcher has At his disposal the measurements of relative amounts of thousands of targets simultaneously. The aim is then to extract biological significance from the data, in order to validate a hypothesis. The need for statistics became striking soon after the appearance of microarray technology, since the abundance of the data required rigorous procedures for analysis. It is important to notice that the intervention of statistical concepts occurs long before the analysis of the data *stricto sensu*. Looking for an appropriate method to analyze the data, when no experimental design has been planned, or no normalization procedure has been applied, is unrealistic.

Dedicated experimental designs have been developped for expression profile microarrays (Yang and Speed (2002), Kerr and Churchill (2001)). Since CGH microarray experiments are relatively new, no dedicated experimental design has yet been developped. Nevertheless, the discovery of more and more polymorphic Copy Number Variations in humans (as explained in Chapter 2) could be a potential issue. Since it has been shown that the frequency of silent deletions is not negligible in humans, the definition of a "normal" genome will have to be set in order to correctly quantify chromosomal aberrations in human diseases.

3.2 The variability of microarray data and the need for normalization

Even if microarray technology provides new potential for the analysis of thousands of targets, several problems arise in the execution of a microarray experiment that can make two independent experiments on the same biological material differ completely, because of the high variability of microarray data. Even if some variability can be controlled using appropriate experimental designs and procedures, other sources of errors cannot be controlled, but still need to be corrected. The most famous of these sources of variability is the intensity-dependent dye bias for cDNA microarray experiments.

3.2.1 The Loess normalization procedure for gene expression experiments

To perform a comparison between two conditions labelled with Cy3 and Cy5, respectively, one needs to state that differential labelling will not corrupt the log-

ratio values M. Yet, it is well-known that a dye effect exists that can have two different causes:

- optical: the higher the mean intensity of the gene is, the more the green label prevails over the red one when the slide is scanned.

- biological: some specific targets are systematically badly labeled by Cy3 or Cy5. For instance, Cy3 can be preferentially incorporated into some sequences, relative to Cy5.

In expression profile experiments, the dye effect is clearly Intensity-dependent. To correct this, classical statistical procedures assume that the dye effect depends on the gene only through its mean intensity A. This assumption allows a convenient graphical observation of the dye effect, the M-A plot, proposed by Yang *et al.* (2002), along with a more robust estimation of the effect. In figure 3.1 (left) we observe the differential effect of the two dyes: M values increase with A values, confirming that the Cy5 signal prevails for high mean expression genes. Moreover, it is clear that the shape of the data cloud is neither constant nor linear, meaning that a constant or linear modelling will not adequately correct the dye effect. In this case, one needs to perform non linear normalization methods.

The Loess procedure (Cleveland (1979)) was the first non linear method proposed to correct the dye effect (Yang *et al.* (2002)). The Loess is a robust locally weighted regression based on the following model:

$$M = c(A) + E$$

where c is an unknown function and E is a symmetric centered random variable with constant variance. The aim of the Loess procedure is to locally approximate c with a polynomial function of order d, and to estimate the polynomial parameters by weighted least square minimization from the neighboring points (A_i, M_i). In figure 3.1 (left) the Loess estimation of the data cloud trend appears in grey. As for systematic biases, once the trend is estimated it is substracted from the log-ratio to obtain a centered data cloud.

3.2.2 A Loess normalization procedure for array CGH data?

Since the Loess procedure performs well for gene expression microarrays, a natural question is its application to array CGH data. The labelling procedure of the DNA targets being similar to the labelling of cDNAs for gene expression experiments, the intensity dependent dye bias also exists in CGH experiments. Nevertheless, Loess normalization can be performed under some hypotheses that are not valid for array CGH experiments. These hypotheses are:

- Most of the genes that are used to estimate the artifact contribution to signal are supposed to be unaltered,

- The artifacts that are corrected are not confounded with a biological effect,

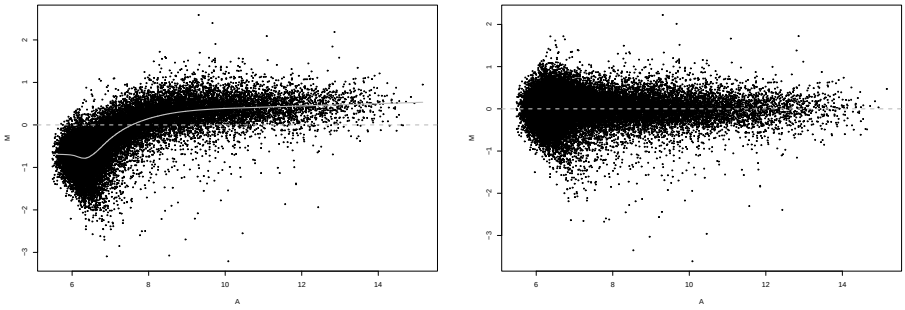


Figure 3.1: **Left:** M-A graph on raw data for gene expression experiment. The gray line is the loess estimation of function c, the dotted line represents the abscissa axis **Right:** M-A graph after Loess normalization.

Figure 3.2 shows a typical example of MA plots for array CGH data. More than an intensity-dependent dye bias, it appears that the data cloud is structured according to log-ratio values, and this structure can be interpreted in terms of sequence copy numbers. Amplified DNA sequences will show a positive log-ratio, whereas deleted targets will show a negative log-ratio. The location of such amplified and deleted sequences on the genome will be the purpose of the next section (the MA plot representation does not consider the physical order of the targets).

Nevertheless, the first hypothesis for loess normalization requires that targets used for estimation have a constant log-ratio with respect to the biological problem, and therefore only reflect bias effects (Ball *et al.* (2003)). In array CGH experiments, this hypothesis is clearly not respected, since differences in log-ratio values reflect biological information in terms of gene copy numbers. To this extent, none of the hypotheses are true for array CGH experiments, since:

- A cancer genome may present a high degree of variability, leading to an important proportion of the genome being altered by chromosomal aberrations,

- Log-ratio values are centered around mean log ratios for each biological class (deleted, normal, amplified). This biological information should not be corrected by a global loess procedure.

Finally, the departure from crucial hypotheses hampers the application classical normalization Procedures to array CGH data, and no dedicated method has yet been proposed. In the following, we will propose some perspectives for this problem, but in this work, we are focused instead on the identification and localization of altered chromosomal regions, which constitutes the specificity of array CGH data analysis.

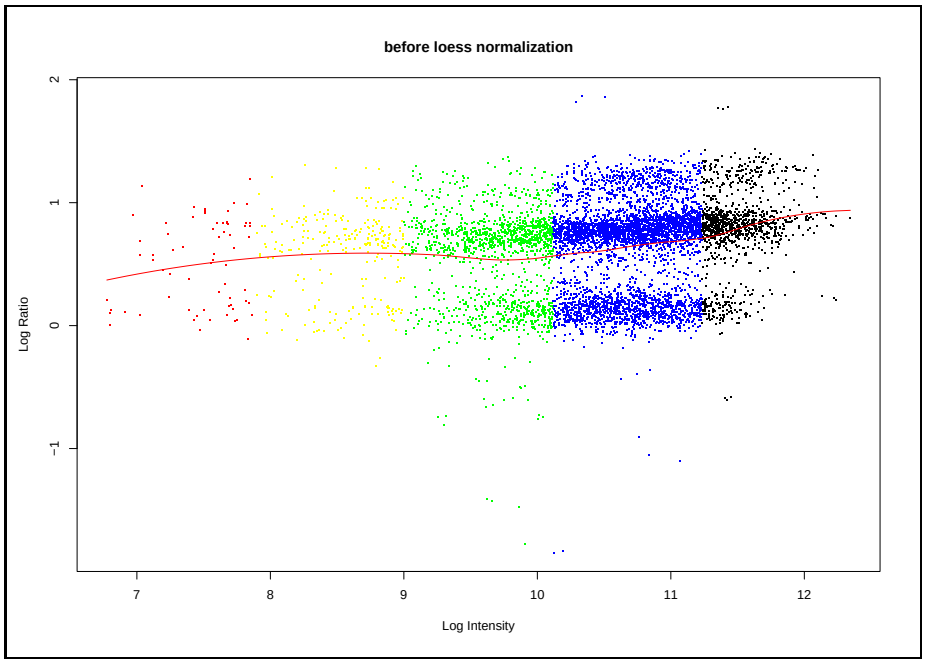


Figure 3.2: Example of MA-plot for array CGH. Example provided by P. Hupé, Institut Curie (Paris).

3.3 Specificity of array CGH data

When analyzing array CGH data, we use fluorescence log-ratios that can be ordered according to the physical location of each BAC on the genome. Such a ratio will be denoted y_t for the log-ratio of the BAC located at position x_t on the genome. The representation of CGH data will be called CGH profiles which are drawn for each individual chromosome.

Let us focus on Figure 3.3 and Figure 3.4 which present a theoretical and real CGH profile. In theory, the underlying biological process that is studied is discrete (counting of relative copy numbers of DNA sequences). Nevertheless, Figure 3.4 shows that the resulting signal is rather continuous. This is due to the quantification process, which is based on fluorescence measurements, and also to the nature of the genomes under study, since the possible values for chromosomal copy numbers in the test sample may vary considerably, especially in the case of clinical tumor samples that present mixtures of tissues of different natures.

Each profile can be viewed as a succession of 'segments' that represent homogeneous regions in the genome whose BACs share the same relative copy number on average. Array CGH data are normalized with a median set to $\log_2(\text{ratio}) = 0$ for regions of no change, segments with positive means represent duplicated regions in the test sample genome, and segments with negative means represent deleted regions. A rough manual annotation can be used to delimit the different regions, but experimental variability makes statistical procedures essential for a reliable analysis. The objective of the statistical study is then to determine how

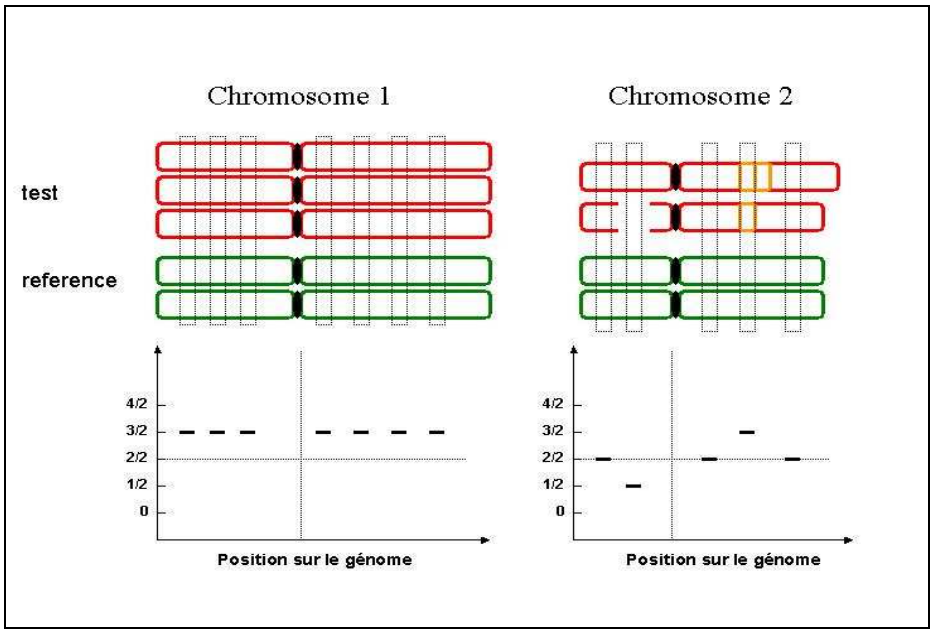


Figure 3.3: Principle of a CGH experiment. Two genomes are compared through representative sequences (dot-boxes). Microarray CGH technology aims at counting the relative copy numbers of each representative sequence. An amplification would show a ratio of 3:2 whereas a deletion would show a ratio of 1:2. Note that these numbers are theoretical numbers.

many chromosomal aberrations there are in a CGH profile, and to localize them on the genome. Segmentation methods are natural for this purpose, in order to determine chromosomal segments on the genome which share the same relative copy number on average. In a first step, the following statistical study will consist in the development of an appropriate segmentation method for the analysis of array CGH data.

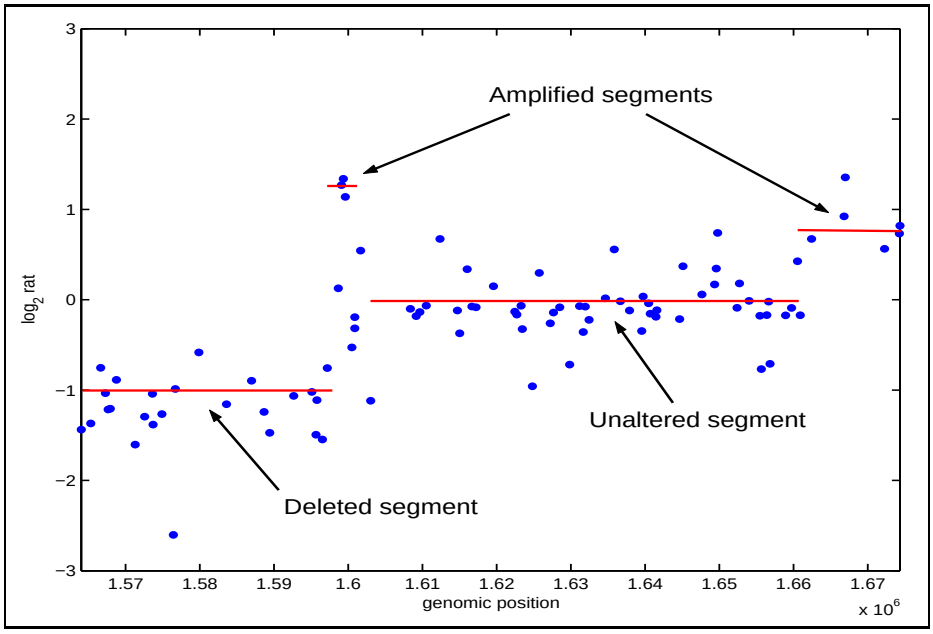


Figure 3.4: Example of a CGH profile. Data are described in Snijders *et al.* (2001). Dots on the graph represent the $\log_2$ ratio of intensities for each BAC according to their physical position on the genome.

Part II

Introduction to segmentation methods for the analysis of array CGH data

Chapter 4

Process segmentation

Many application fields in Statistics provide signals that are modelled through time series which are not stationary. Since the interpretation of such signals is complex, one aim is often to isolate zones in which the signal can be considered as stationary. In this context, the signal can be studied with parametric models for which the parameters are supposed to be affected by abrupt changes at unknown instants. The purpose of the statistical study is then to detect changes in these parameters. Quality control or monitoring has been one of the earliest applications of change detection. In this context, a production process is observed and must be controlled; the quick identification of disorders may be crucial for safety or quality control reasons. Dedicated statistical methods are based on the observation of sequential data, for which the detection of the change has to be done with the past observations as the only available information. The reader is referred to Basseville and Nikiforov (1993) for a complete review of *on-line* detection of abrupt changes, which does not constitute the purpose of our work. We are focused instead on the case where the analyst studies one global signal. In this case the change detection is done *off-line*, and the problem shifts to the global segmentation of the process.

The multiple change-point problem

In the global segmentation context we aim at delimiting segments for which the characteristics of the signal are homogeneous within segments and different from one segment to another. We note $\{y_t\}_{t=1,\dots,n}$ the observed data which are modelled by a random process $\{Y_t\}_{t=1,\dots,n}$ that is supposed to be drawn from a probability distribution $f(\cdot)$ that depends on a parameter θ. Then we assume that this parameter is affected by $K-1$ abrupt changes at unknown instants noted $t_1 < \dots < t_{K-1}$, with the convention $t_0 = 1$ and $t_K = n$. The model is formulated as follows:

$$\forall t \in I_k, \quad Y_t \sim f(\theta_k),$$

with $I_k = \{t \in]t_{k-1}, t_k]\}$ being the interval of size n_k for which the parameter θ is constant and equals θ_k. Many parameters can be affected by abrupt changes, the simplest ones being the mean and the covariance of the process, but changes can also affect the spectral distribution, or transition probabilities of Markov chains for instance.

From a statistical point of view, the problem of global segmentation gives rise to three main issues: (1) the determination of the parameter(s) affected by the change(s), (2) the estimation of the breakpoint instants, and the estimation of the parameters within segments, (3) the determination of the number of segments. The problem of determining which characteristics of the signal are affected by the changes may require a precise knowledge of the phenomenon under study. In the following, we will restrict the study to the case of changes in the mean only or in the mean and the variance of an independent Gaussian process. This model is detailed in section 4.1.

Estimating the breakpoint coordinates

Once the model has been specified, the problem is to estimate the location of the breakpoints and the parameters within segments. We will focus on two classical methods for this purpose: the maximum likelihood method and the least-squares method. For this estimation step, the number of segments has to be fixed. In the global segmentation setting, the estimation of the breakpoints can be viewed as a partitioning problem, where the purpose is to find the best partition of the data into K segments. Since the number of possible partitions of the data into K segments is C_{n-1}^{K-1}, the exploration of all possible partitions would be of order $\mathcal{O}(n^K)$. This computational problem explains why many segmentation methods only consider the detection of one change, compared to the multiple change-point problem. In section 4.2 we will explain how dynamic programming provides a solution to this problem of partitioning, and how the CART algorithm proposed by Breiman *et al.* (1984) can be used for the detection of multiple changes in the mean for large samples.

Model selection

Once an estimation procedure is available for a fixed number of segments, the question of choosing this number remains. In practice this number is unknown and should be estimated. This problem can be viewed as a model selection issue. To date the number of segments is estimated with a penalized criterion:

$$crit(K) = J_K - \beta_n pen(K). \tag{4.1}$$

The first term J_K measures the quality of fit of the model to the data. It can be the log-likelihood at its maximum noted $\log \widehat{\mathcal{L}}_K$, or minus the sum of squares of the model for instance. The second term is an increasing function of the number of segments, and is used to penalize the selection of an overly high-dimensional model. The term β_n is a positive constant. This criterion establishes a trade-off between a good quality of fit and a reasonable number of segments. The definition of an appropriate penalty function and constant has focused much attention. In Section 4.3 we detail existing methods for model selection procedures in the multiple change-point context.

The multiple change-point problem in the Bayesian setting

The last section will be devoted to a different approach which has been used to study multiple change-point problems, the Bayesian approach. In this context, the number of breakpoints and their location are viewed as random variables. The objective is to estimate their *posterior* distribution with MCMC methods. In this section we will compare two parametrizations which have been proposed by Green (1995) and Lavielle and Lebarbier (2001). Our objective is to explain the main differences between the two approaches and to draw analogies with the frequentist setting, when possible.

4.1 Detection of changes in the mean of a Gaussian process

In this section we consider that the data are independent and drawn from a Gaussian distribution, such as

$$\forall t \in \{1, \ldots, n\}, \;\; Y_t \sim \mathcal{N}(\mu(t), \sigma(t)^2).$$

Then we assume that the mean and the variance of the process are affected by $K-1$ abrupt changes at unknown instants noted $t_1 < \ldots < t_{K-1}$. This model will be denoted $\mathcal{M}_1$, in contrast to model $\mathcal{M}_2$ where the only parameter affected by the changes is the mean, with a constant variance σ^2. Then we have:

$$\forall t \in I_k \;\; Y_t \sim \begin{cases} \mathcal{N}(\mu_k, \sigma_k^2) & \text{model } \mathcal{M}_1, \\ \\ \mathcal{N}(\mu_k, \sigma^2) & \text{model } \mathcal{M}_2, \end{cases}$$

Since the data are independent, the log-likelihood of the model can be written as a sum of local log-likelihoods calculated on each individual segment, that is:

$$\log \mathcal{L}_K = \sum_{k=1}^{K} \sum_{t=t_{k-1}+1}^{t_k} \log \left\{ \frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(y_t - \mu_k)^2}{2\sigma^2} \right\}.$$

This additivity property will be central for the downstream estimation procedures that are based on maximum likelihood.

4.2 Estimation procedures when the number of segments is fixed

4.2.1 The maximum likelihood method

If the breakpoints are known, the estimators of the mean and the variance are the classical maximum likelihood estimators:

$$\begin{aligned}
\hat{\mu}_k &= \frac{1}{n_k} \sum_{t=t_{k-1}+1}^{t_k} y_t, \\
\hat{\sigma}_k^2 &= \frac{1}{n_k} \sum_{t=t_{k-1}+1}^{t_k} (y_t - \hat{\mu}_k)^2 \text{ for } \mathcal{M}_1, \\
\hat{\sigma}^2 &= \frac{1}{n} \sum_{k} \sum_{t=t_{k-1}+1}^{t_k} (y_t - \hat{\mu}_k)^2 \text{ for } \mathcal{M}_2.
\end{aligned}$$

For a model with K segments the log-likelihood at its maximum is:

$$\begin{aligned}
\log \widehat{\mathcal{L}}_K &= -\frac{n}{2} \log 2\pi - \frac{1}{2} \sum_k n_k \log \hat{\sigma}_k^2 \text{ for } \mathcal{M}_1, \\
\log \widehat{\mathcal{L}}_K &= -\frac{n}{2} \log 2\pi - \frac{n}{2} \log \hat{\sigma}^2 \text{ for } \mathcal{M}_2.
\end{aligned}$$

Nevertheless, the position of the breakpoints is unknown and should be estimated. This problem can be formulated as a partitioning problem whose aim is to find the best partition of the grid $\{1, \ldots, n\}$ into K segments. If we note $\mathcal{P}_K$ the set of all possible partitions of the grid $\{1, \ldots, n\}$ into K segments, the breakpoints are estimated as follows:

$$\hat{T}_K = \{\hat{t}_1, \ldots, \hat{t}_{K-1}\} = \underset{T_K \in \mathcal{P}_K}{\text{Argmax}} \left\{ \log \widehat{\mathcal{L}}_K(T_K) \right\}.$$

Dynamic programming is an efficient recursive approach that can be used to reduce the computational time of the exhaustive search.

4.2.2 Dynamic programming and the shortest path problem to estimate the breakpoint instants

Dynamic programming has been introduced by Bellman and Dreyfus (1962) and Auger and Lawrence (1989) were the first to use it in the context of global segmentation. It is a recursive approach based on the Bellman optimality principle (Bellman and Dreyfus (1962)). Let's consider model $\mathcal{M}_2$ with a constant variance. The quantity to be optimized is then:

$$J_K = \sum_{k=1}^{K} \sum_{t=t_{k-1}+1}^{t_k} (y_t - \hat{\mu}_k)^2.$$

The mean squares criterion is Broken down into a sum of minimum mean squares criteria. This break-down allows us to draw an analogy to the shortest path problem. Criterion J_K can be seen as the total length of a path connecting point 1 to point n. The problem is then to find the shortest path connecting point 1 to point n with $K-1$ steps, the steps being the breakpoint instants $t_1, \ldots, t_{K-1}$.

Denoting $J_k(i,j)$ the cost (length) of the path connecting point i and j in k steps, the algorithm is as follows:

$$\begin{aligned} \forall 0 \leq i \leq j, \;\; J_1(i,j) &= \sum_{t=i+1}^{j} (y_t - \bar{Y}_{ij})^2, \\ \forall 1 \leq k \leq K-1, \;\; J_{k+1}(1,j) &= \min_{1 \leq h \leq j} \left\{ J_k(1,h) + J_1(h+1,j) \right\}. \end{aligned}$$

In this context, the Bellman optimality principle is formulated as follows: "subpaths of optimal paths are themselves optimal". This global minimization property is crucial since it ensures the optimized criterion to be at its global maximum (compared with other estimation algorithms such as the EM algorithm that only ensures a local maximum). Moreover, this algorithm reduces the computational burden of the exhaustive search from $\mathcal{O}(n^K)$ to $\mathcal{O}(n^2)$ for a given K. This approach has been used by many authors and the reader is referred to Auger and Lawrence (1989), Braun *et al.* (2000) and Hawkins (2001) for instance.

4.2.3 A CART-based approach for the multiple change-point problem

Even if dynamic programming considerably reduces the computational time of the exhaustive search, it cannot be used for overly large samples. If the data to be partitioned are DNA sequences for instance, the storage of a cost matrix that is $n \times n$ with $n \sim 10^9$ is difficult. For this reason, Gey and Lebarbier (2002) recently proposed combining dynamic programming with a CART-based approach for the estimation of the breakpoints, when the size of the data is large. The role of the CART-based method for segmentation is to restrict the collection of visited partitions $\mathcal{P}_K$ to the relevant ones. This leads to a fast algorithm of order $\mathcal{O}(n \log n)$.

The CART algorithm is computed in two steps (Breiman *et al.* (1984)). The first one is called the growing procedure and consists in the recursive construction of a collection of partitions using data-dependent dyadic splitting. The computational schema of the first step is as follows:

- Compute the change-point $\hat{t}_c$ such as $\hat{t}_c = \underset{j}{\text{Argmin}} \left\{ J_1(1,j) + J_1(j+1,n) \right\}$.

  The objective of this step is to find the first best partition of $\{1, \ldots, n\}$ into 2 segments.

- Apply the same procedure on the new defined segments, and so on until the number of points within each resulting segment is smaller than a given threshold.

Other sequential methods have been proposed for the change-point estimation problem, see Ghorbanzdeh (1995), Picard (1985) and Chong (2001) for instance.

Nevertheless, those methods aim at finding the relevant breakpoints directly, leading to sequential tests that require the definition of many tuning parameters. The use of a CART-based method is different. The first step (growing procedure) provides a collection of segmentations and the only parameter to be tuned is the minimum size for a segment to be split. In a second step (the pruning step), a relevant segmentation is chosen with a model selection procedure.

Once this segmentation has been chosen, it appears that some breakpoints can be irrelevant. This is due to the sequential nature of the CART algorithm that does not guarantee the finding of the global optimum. In order to circumvent this difficulty, Gey and Lebarbier (2002) propose combining the CART algorithm with a partial exhaustive search. The general idea is to consider that the breakpoints that have been proposed by CART (at the end of the growing and pruning procedures) constitute candidates that can be removed if they correspond to false alarms. This is done by dynamic programming, which performs a partial exhaustive search on the proposed breakpoints to free the results from the hierarchic nature of the CART candidates. This leads to a hybrid algorithm that has been shown to be efficient (see Gey and Lebarbier (2002)).

4.2.4 Statistical properties of the breakpoint estimators

Once the position of the breakpoints has been estimated, a classical question is the statistical properties of the resulting estimators. Nevertheless, since the breakpoint parameters are discrete, the likelihood is not continuous with respect to those parameters. This particularity hampers the use of classical techniques to show their consistency for instance. Many articles have considered this problem, see Yao and Au (1989), Siegmund (1988), Lavielle (1999), Braun *et al.* (2000) for instance. Yao and Au (1989) have shown that in the case of a jump in the mean of an independent Gaussian process, the breakpoint estimators were consistent, and Braun *et al.* (2000) later show the consistency in the case of processes whose variance depends on the mean. Lavielle and Moulines (2000), Lavielle (1999) further extended those results to the case of time series and dependent processes, showing that the rate of convergence of $\hat{t}_k$ does not depend on the covariance structure of the process. In the case of a jump in the mean Yao and Au (1989) provide a theorem concerning the limiting distribution of the breakpoint estimators.

As for the confidence set of the change-point estimators, many strategies have been formulated for the single change-point problem. Siegmund (1988) and Worsley (1986) propose methods based on the likelihood ratio statistic, and Cobb (1978) provides an approximation of the conditional distribution of the maximum likelihood estimator of the change-point given the adjacent observations. In the multiple change-point context, current approaches use tests based on a change in the parameter of the distribution (see Avery and Henderson (1999) for a nonparametric approach in the case of Bernoulli sequence, Venter and Steel (1996) for maximum likelihood approaches in the Gaussian case). Those approaches focus on the change in the parameter with which the data are modelled, and not on the existence of a change-point t_k.

An interesting question would be to assess a simultaneous confidence region

of the breakpoint estimators $\hat{t}_1, \ldots, \hat{t}_{K-1}$. To our knowledge no confidence set has yet been proposed for the sequence of the change-point estimators in the case of multiple breakpoints.

4.3 Model selection procedures to estimate the number of segments

Once the model has been specified and the location of the breakpoints can be estimated with an appropriate method, the problem is to determine the number of segments into which the data should be partitioned. In practice this number is unknown and can be estimated with a penalized criterion defined in Equation 4.

To date, two approaches have been considered to define the penalty term. The first one considers that there exists a true number of breakpoints $K^\star$ that should be estimated, and a true underlying model from which the data have been generated. In this context, Yao and Au (1989) showed that the Bayesian Information Criterion (BIC) provides a consistent estimator of the number of breakpoints. This criterion uses $J_K = \log \widehat{\mathcal{L}}_K$ and $pen(K) = 2K$ for the number of parameters to be estimated (K means, 1 variance and $K-1$ breakpoints), and $\beta_n = 0.5 \times \log(n)$ for the penalty constant. This result is extended to the case of a dependent process, and Lavielle (1999) shows that if constant β_n goes to 0 at an appropriate rate depending on the covariance structure of the process, the estimated number of change points converges to the true number.

Since practical use of penalized criteria is done in a non asymptotic context, another approach for model selection has been provided by Birgé and Massart (2001). This model selection procedure has been applied to process segmentation by Lebarbier (2005) and Lavielle (2005), who propose two strategies that lead to different penalty functions and constants.

4.3.1 Motivation of model selection

In the context of model selection, we have n independent random variables $\{Y_t\}_{t=1,\ldots,n}$ whose distribution s is unknown and has to be recovered. In the case of process segmentation, this function s is recovered using a collection of piecewise constant functions. For this purpose, Lebarbier (2005) defines model $\mathcal{S}_m$ that is the subset of piecewise constant functions on partition $m = \{I_k\}_{k=1,\ldots,K_m}$ of dimension K_m:

$$\mathcal{S}_m = \left\{ u = \sum_{k=1}^{K_m} u_k \mathbb{1}_{I_k}, (u_k)_{k=1,\ldots,K_m} \in \mathbb{R}^{K_m} \right\}.$$

Classical estimation procedures consider that distribution s belongs to $\mathcal{S}_m$. Nevertheless, since s is unknown, it is unlikely that it belongs to any model. The approach developed by Birgé and Massart (2001) considers that model $\mathcal{S}_m$ only constitutes an approximation of s. Since s is unknown, it is approximated by $\bar{s}_m$ that belongs to model $\mathcal{S}_m$. Nevertheless, $\bar{s}_m$ itself is unknown and is estimated

by $\hat{s}_m$. Then the quality of an estimator $\hat{s}_m$ is assessed with a quadratic risk, $\mathbb{E}\|s - \hat{s}_m\|^2$, and the chosen estimator should minimize this risk. The quadratic risk of $\hat{s}_m$ can be broken down such that:

$$\mathbb{E}\|s - \hat{s}_m\|^2 = \mathbb{E}\|s - \bar{s}_m\|^2 + \mathbb{E}\|\bar{s}_m - \hat{s}_m\|^2.$$

The first term $\mathbb{E}\|s - \bar{s}_m\|^2$ measures the distance of the unknown s to the approximator $\bar{s}_m$ in $\mathcal{S}_m$. This is a bias term that is small if the approximation is good. The second term $\mathbb{E}\|\bar{s}_m - \hat{s}_m\|^2$ measures the quality of the estimation of $\bar{s}_m$ by $\hat{s}_m$. This quantity should be small to prevent estimation errors. The purpose of model selection is then to establish a trade-off between a model that is close to the unknown distribution and which provides a good approximation of the unknown distribution, but that is not too big to prevent from estimation errors. This is called the bias/variance trade-off.

An ideal estimator of s, noted $\hat{s}_m$ could be defined as the estimator that achieves the best bias/variance trade-off. The objective of model selection is then to construct a criterion that will be used to select a partition $\hat{m}$ which behaves as well as the best estimator, up to some constant. This criterion is composed of two terms, a first term that quantifies the closeness of model $\mathcal{S}_m$ to the data, that increases with the dimension of the model, and a penalty term to control the estimation errors.

In the context of process segmentation, a model is selected through its dimension, *ie* we aim at selecting $\hat{m}$ the partition of dimension K_m. This is achieved with a penalty function defined by Lebarbier (2005), such that:

$$\beta_n \times pen(K) = \frac{K_m}{n}\sigma^2 \times \left\{ c_1 \log\left(\frac{n}{K_m}\right) + c_2 \right\}, \tag{4.2}$$

with c_1, c_2 two positive constants to be calibrated and σ^2 to be estimated. This function increases with the dimension of the model K_m, and the $\log(n/K_m)$ term reflects the richness of collection of partitions, since there exists $C_{n-1}^{K_m-1}$ possible partitions of the grid $\{1, \ldots, n\}$ into K_m segments.

The performance of this penalty function has been assessed by simulation studies, and compared to other penalized criteria, such as the Mallows C_p criterion, and the Bayesian Information Criterion (BIC) in a non asymptotic context. The main difference between those criteria is that criteria constructed on asymptotic considerations do not consider the complexity of the different models. Let us recall that the construction of BIC in the context of process segmentation considers that the number of parameters to be estimated is K_m means, $K_m - 1$ breakpoints and 1 variance, whereas this new penalty considers that there exists $C_{n-1}^{K_m-1}$ possible partitions when K_m is fixed. This leads to a penalization that is more stringent, and to the selection of a lower number of segments. Note that the construction of a penalty function is based on different objectives that will explain its behavior. For instance, the use of BIC to select the number of segment is motivated by the finding of the true number and of the true breakpoint coordinates. On the other hand, the penalty given by Lebarbier (2005) aims at minimizing a quadratic risk,

and will tend to ignore some irrelevant breakpoints corresponding to small jumps in the mean.

To complete the introduction of model selection for segmentation process, the reader is referred to (Lebarbier 2005) for further information concerning penalty 4.3.1, the calibration of constant c_1, c_2 and the estimation of σ^2. Model selection theory has been applied to a wide range of statistical problems. See Birgé and Massart (2001) for a general presentation of model selection theory, Castellan (2000) for the application of model selection to the estimation of histograms, and Gey and Nedelec (2002) for model selection for CART Regression Trees.

4.3.2 An adaptive method to estimate the number of segments

In contrast to Lebarbier (2005) who aims at finding a universal penalty for selecting the number of segments, Lavielle (2005) has developed an adaptive method that is heuristically based. The motivation of such method is that the penalties defined for the BIC criterion or by Lebarbier (2005) are adapted to a very particular context. In the first one, the objective is to recover the true configuration, and the second one aims at minimizing a very specific criterion (the quadratic risk of the estimator), but none of these methods holds in the non-Gaussian case or for dependent variables for instance. The aim of Lavielle (2005) is to propose a method that can be used in many different situations, with very few hypotheses.

First of all let us notice that when the number of segments is small regarding the size of the data, penalty 4.3.1 is linear in the number of segments, and Lavielle (2005) suggests using a penalty in the form:

$$pen(K) = 2K.$$

The new objective is to estimate β adaptively to the data. This estimation is done considering the behavior of the quality of fit criterion that is used. If this criterion is the least-squares criterion noted J_K, it will decrease as the number of segments increases, and the method consists in the determination of the number of segments for which the criterion ceases to decrease significantly. The proposition considers the slope between points (K_i, J_{K_i}) and $(K_{i+1}, J_{K_{i+1}})$. Looking where J_K ceases to decrease significantly means looking for a break in the slope of this curve. An illustration is provided in Figure 4.1.

This method is heuristically based and requires the tuning of a parameter to assess the "significance" of the slope break. Nevertheless, it appears to be very flexible and has been shown to be efficient in many situations. Simulation results comparing this adaptive method to the penalty defined by Lebarbier (2005) show that it is more robust to the addition of noise (Picard *et al.* (2005)).

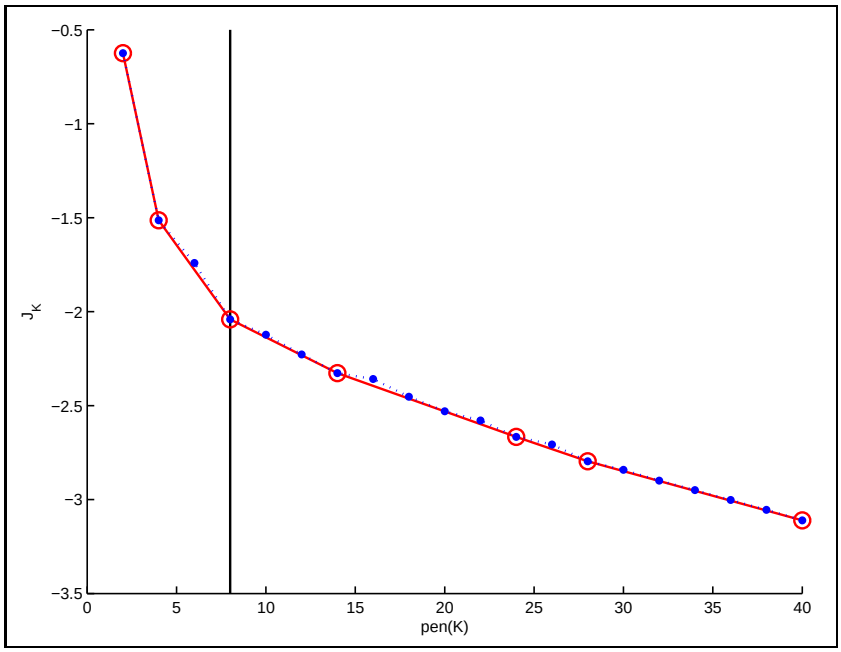


Figure 4.1: Illustration of the model selection procedure proposed by Lavielle (2005). Circles represent the convex hull of contrast J_K. The vertical line indicates the number of segments for which the contrast ceases to decrease significantly.

4.4 Bayesian formulation of the multiple change-point problem

In order to complete this review on segmentation methods, we present another modelling strategy that has been considered for this problem, in the Bayesian framework. See Green (1995), Carlin (1992), Barry and Hartigan (1993), Avery and Henderson (1999), Lavielle and Lebarbier (2001) for instance. Previous sections were dedicated to strategies whose objective is to provide the best segmentation on the data, based on a specific criterion. The objective is different in the Bayesian setting, where the number of segments as well as their position is random. As a consequence, their *posterior* distribution will be used to choose the most appropriate number of segments, and will provide local information regarding the position of the breakpoints.

The model can be specified as follows. Let $\{Y_t\}$ be a real process such that

$$\forall t \in \{1, \ldots, n\}, \;\; Y_t = s(t) + \varepsilon_t,$$

where ε_t is a sequence of zero-mean random variables. The function s to be recovered is supposed piecewise constant. With the conventional notations:

$$\forall t \in I_k, \;\; s(t) = \mu_k.$$

4.4.1 The multiple change-point problem and the reversible jump algorithm

Two approaches have been considered to model the sequence of change-points. Green (1995) specifies the *prior* model as follows. Suppose that the number of segments K is drawn from a Poisson distribution $\mathcal{P}(\lambda)$. Given K, the breakpoint positions $t_1, \ldots, t_{K-1}$ are distributed as the even-numbered order statistics from $2K-1$ points uniformly distributed on $[1, n]$, and the means μ_k are independently drawn from the gamma density $\Gamma(\alpha, \beta)$.

A Monte Carlo Markov Chain algorithm is required to calculate the *posterior* probabilities of both breakpoint instants and means. Nevertheless, those probabilities depend on the number of segments which may vary. Many authors have solved this problem by fixing K at 1. The development of the reversible jump MCMC sampler has allowed this limitation to be circumvented, and the multiple change-point problem was one of its first applications. The reader is referred to Green (1995) for further details on the application of the Reversible Jump algorithm to the multiple change-point problem.

4.4.2 A reparametrization of the multiple change-point problem

Instead of a parametrization that considers the breakpoint instants $\{t_k\}_k$, Lavielle (1998) and Lavielle and Lebarbier (2001) propose introducing a sequence of constant size $\{R_t\}$, such that:

$$R_t = \begin{cases} 1 & \text{if there exists } k \text{ such that } t = t_k. \\ 0 & \text{otherwise.} \end{cases}$$

The variables are supposed to be independent with *prior* Bernoulli distribution $\mathcal{B}(\lambda)$. Let us concentrate on the differences between the model specified by Green (1995) compared to this formulation.

In the formulation proposed by Lavielle and Lebarbier (2001) the variable of interest is the *presence* of a breakpoint, which is supposed independent from the presence of a breakpoint at close instants. In the framework defined by Green (1995) however the sequence of breakpoint instants $\{t_k\}$ indicates the *position* of the breakpoints, and positions are not independent from each other. As a consequence the *posterior* distribution of the $\{t_k\}$ in the reversible jump context will directly quantify the uncertainty regarding the breakpoints location. With the reparametrization of the model, this information will be provided by the quantity $\Pr\{\sum_{t=t_a}^{t_b} R_t = k | y; \theta\}$, which is the probability of having exactly k change-points between instants t_a and t_b.

Another difference lies in the distribution of the number of segments. In the first case, this number is assumed to follow a Poisson distribution, and the distribution of the breakpoint instants only depends on its current value. In the formulation proposed by Lavielle and Lebarbier (2001), the *prior* distribution of

the sequence $\{R_t\}$ defines the *prior* distribution of the number of segments. Since $K_R = \sum_{t=1}^{n-1} R_t + 1$, and $R_t \sim \mathcal{B}(\lambda)$, it follows that $K_R \sim \mathcal{B}(n-1, \lambda)$. The choice of the number of segments will depend on the choice of λ. More than a strict impact of parameter λ on the distribution on the number of segments, the Bernoulli *prior* on R_t specifies the distribution of the distance between two breakpoint instants since

$$\Pr\{R_{t+1} = 0, \ldots, R_{t+\ell-1} = 0, R_{t+\ell} = 1 | R_t = 1\} = \lambda(1-\lambda)^{\ell-1}.$$

In this formulation, the *prior* distribution has a double impact: it specifies the distribution of the number of segments, as well as the distribution of the length of the segments, which implicitly becomes geometric.

4.4.3 Recovering the Maximum A Posteriori estimator of the breakpoints sequence

The main advantage of the formulation proposed by Lavielle and Lebarbier (2001) lies in the computational approach that can be used to recover the *posterior* distribution of the sequence $\{R_t\}$. The authors emphasize the hierarchy of the model, that is:

$$p(R, \mu | y; \theta) = p(R | y; \theta) \times p(\mu | y, R; \theta),$$

with θ the set of hyperparameters. The first term $p(R|y;\theta)$ is used to recover the sequence of the breakpoint instants, and once this distribution is known, the signal is reconstructed with a Gibbs sampler to calculate $p(\mu|y, R; \theta)$, the hyperparameters of the model being estimated with a stochastic approximation of the EM algorithm, SAEM (Delyon *et al.* (1999)). Since the size of sequence $\{R_t\}$ is fixed, a Hastings-Metropolis algorithm can be used to sample sequences of 0 and 1 of size n. This parametrization prevents the use of a reversible jump algorithm, which is known to converge slowly.

Moreover Lavielle and Lebarbier (2001) show that the posterior distribution of R is in the form

$$p(R|y;\theta) = C(y;\theta) \exp\{-U_\theta(y, R)\},$$

where

$$U_\theta(y, R) = \phi \sum_{k=1}^{K_R} \sum_{t_{k-1}+1}^{t_k} (y_t - \bar{y}_k)^2 + \gamma K_R,$$

and where (ϕ, γ) depend on the hyperparameters of the model. The Maximum A Posteriori (MAP) estimator of R that minimizes $U_\theta(y, R)$ is then a penalized least-squares estimator. An analogy can be drawn with the breakpoint estimators defined in the frequentist context:

$$\{\hat{t}_1, \ldots, \hat{t}_{K-1}\} = \underset{t_1,\ldots,t_{K-1}}{\text{Argmin}} \left\{ \frac{1}{n} \sum_{k=1}^{K} \sum_{t=t_{k-1}+1}^{t_k} (y_t - \hat{\mu}_k)^2 - 2\beta K \right\}.$$

The recovery of the MAP estimator of the breakpoint sequence can face local maxima that should be avoided. To do so, Lavielle and Lebarbier (2001) propose

a modification of the Hastings-Metropolis algorithm, with the introduction of a temperature parameter T such that:

$$p_T(R|y;\theta) = C_T(y;\theta)\exp\{-\frac{U_\theta(y,R)}{T}\}.$$

The interest in this temperature parameter is that when T tends to 0, $p_T(\cdot|y;\theta)$ converges to the uniform distribution on the set of global maxima of $p(\cdot|y;\theta)$. Simulated annealing algorithms consist in using a sequence of temperatures $T^{(i)}$ that decrease at each iteration. Nevertheless, the use of this sequence would require a very large number of iterations. In practice, Lavielle and Lebarbier (2001) suggest running the Hastings-Metropolis algorithm at a fixed low temperature. The problem is to choose this temperature parameter.

4.5 Conclusion

In this chapter, we presented a brief review of existing statistical methods concerning the multiple change-point problem. Of course this review is not exhaustive, since the bibliography related to this subject is ample. Our scope was to present and explain the main tools that will be used in the following, such as dynamic programming and model selection, but also to present other existing methods, such as Bayesian methods that constitute an alternative modelling strategy. Once the statistical concepts related to process segmentation have been presented, our first work has been to apply them to real genomic data sets.

Chapter 5

Application of segmentation methods to CGH array data analysis

The reasons for using segmentation methods to the analysis of array CGH data has been previously motivated in Chapter 3, and Chapter 4 was dedicated to the review of statistical methods for process segmentation. Nevertheless, some other approaches have been proposed in the specific context of array CGH data analysis. They mainly concern the estimation method for the breakpoints that can be based on many different algorithms. We will briefly present the main statistical approaches based on segmentation methods that are currently used in the field of array CGH data analysis. We also make a clear distinction between visualization tools and toolboxes that have been proposed (Kim *et al.* (2005), Chen *et al.* (2005)) compared to statistical methodologies which are the purpose of our study. The notations introduced in Chapter 4 will be used in this section, in order to draw analogies between the proposed methods, and to compare them.

The second section of this chapter will be devoted to the presentation of the segmentation method we proposed in an article published in the journal BMC-Bioinformatics, Picard *et al.* (2005). A copy of this article can be found in the Publications section. The last section will be devoted to the comparison of segmentation methods, based on real data sets, in order to discuss the behavior of three existing methods, the method proposed by Lavielle and Lebarbier (2001), Hupe *et al.* (2004) and the method we propose.

5.1 Diversity of segmentation methods for array CGH data

5.1.1 A sequential procedure to segment array CGH profiles

Historically, Olshen and Venkatraman (2002) were the first to propose a statistical method for the detection of breakpoints in array CGH data. Their method is based

on a statistical test developed by Sen and Srivastava (1975), which is modified and called binary circular segmentation (CBS). Considering the partial sum $S_i = \sum_{t=1}^{i} Y_t$, $1 \leq i \leq n$, the likelihood ratio statistic for testing the null hypothesis of no change against the alternative that there exists exactly one change at unknown location i is $Z = \max_{1 \leq i < n} |Z_i|$ where Z_i equals:

$$Z_i = \frac{S_i/i - (S_n - S_i)/(n-i)}{\sqrt{1/i + 1/(n-i)}}.$$

Olshen and Venkatraman (2002) propose a modification of the test statistic that is based on a test which considers only a single change. The new statistic is $Z = \max_{1 \leq i < j \leq n} |Z_{ij}|$, such that:

$$Z_{ij} = \frac{(S_j - S_i)/(j-i) - (S_n - S_j + S_i)/(n-j+i)}{\sqrt{1/(j-i) + 1/(n-j+i)}}.$$

A Monte Carlo method is used to calculate the tail of the distribution under the null hypothesis. The procedure is applied recursively until all the changes have been identified, and a permutation approach is considered to relax the normality assumption under the null hypothesis. This method is sequential and close to *on-line* detection methods that do not provide a global optimal segmentation.

5.1.2 A smoothing method to estimate the breakpoints

This approach has been proposed by Hupe *et al.* (2004) and is based on Polzehl and Spokoiny (2000). This method consists in the local estimation of a smoothing function s for each position x_t. In this context the log-ratio Y_t is supposed to depend on the position of the BAC at X_t via an unknown function s, and a regression model is considered such that:

$$Y_t = s(X_t) + \varepsilon_t,$$

with ε_t being i.i.d. random error terms with distribution $\mathcal{N}(0, \sigma^2)$. The Adaptive Weights Smoothing procedure is iterative and finds around every x_t the maximal possible neighborhood in which the function s is constant: a weight $w_{t\ell}$ is assigned to every observation Y_t. This model leads to the following weighted maximum likelihood estimator :

$$\hat{s}(x_t) = \min_s \frac{1}{2\sigma^2} \sum_{\ell=1}^{n} w_{t\ell}(Y_\ell - s)^2,$$

with σ^2 assumed to be known. Briefly, weights are calculated and up-dated using kernel functions, one denoted f_ℓ that considers the proximity of the x_ℓs in the neighborhood, and a second one denoted f_s which penalizes for too many breakpoints.

The AWS procedure provides one estimate $\hat{s}(x_t)$ for each position. Since the data have been smoothed, this procedure requires an ad-hoc definition of a breakpoint. Hupe *et al.* (2004) define a breakpoint at position x_t if $\hat{s}(x_t) \notin$

$[\hat{s}(x_t) + \epsilon, \hat{s}(x_t) - \epsilon]$, with ϵ a tuning parameter fixed at 10^{-2}. Note that ϵ is not the only tuning parameter of the procedure, others being the maximal width of the neighborhoods that can be considered, and the weight given to the kernel that penalizes for an overly large number of breakpoints.

As for the choice of the number of breakpoints, the authors indicate that the kernel function f_s is not sufficient to prevent false positive breakpoints. This is why a filtering step is added to remove these undesirable breakpoints. The new function to be optimized is then :

$$\sum_{k=1}^{K} n_k \log \hat{\sigma}_k^2 + \lambda \sum_{k=1}^{K} f\left(\frac{|\hat{\mu}_k - \hat{\mu}_{k+1}|}{\hat{\sigma}^2}\right) \log(n),$$

with n_k the size of segment k, $\hat{\mu}_k$ and $\hat{\sigma}_k^2$ the empirical estimators of the mean and variance of segments k and $\hat{\sigma}^2$ the estimation of the global variance, $f(\cdot)$ the tricubic kernel function, and λ a penalty constant that has to be fixed. Note that there exists an ambiguity in the definition of this criterion, since the left part represents the log-likelihood of a segmentation model where the variance depends on each segment, whereas the initial definition of the model stipulates a constant variance.

This method seems to be efficient for array CGH data analysis, but requires the tuning of many parameters. It has the advantage of considering that BACs are not evenly spaced on the genome, and the neighborhood that is used to locally estimate function s considers that the distance $x_{t+1} - x_t$ is not constant. Other smoothing algorithms have been proposed (Eilers and Menezes (2005)), but their aim is mainly in the representation and visualization of the data.

5.1.3 Finding breakpoints with a genetic algorithm

This method has been proposed by Jong *et al.* (2003) who consider a segmentation model in the Gaussian framework with heterogeneous variances. The model is then model $\mathcal{M}_1$. The function to be optimized is the penalized log-likelihood :

$$\log \tilde{\mathcal{L}}_K = \sum_{k=1}^{K} n_k \log \hat{\sigma}_k^2 + \lambda K,$$

with λ set at 10. The point of the authors is not the correct choice of the penalty term, but rather the estimation of the breakpoints, which is done with a genetic algorithm.

The Genetic Local Search algorithm is based on a local update of the breakpoints. Random breakpoints are chosen and repeatedly updated in order to maximize the penalized log-likelihood function $\log \tilde{\mathcal{L}}_K$. At each iteration, the update rule chooses a breakpoint which is moved only if this move increases the scoring function. The iterative process terminates when no move increases the scoring function. In addition to this local search, a genetic rule is applied using the penalized likelihood as fitness function. Briefly, mutations randomly decide whether to add or to remove breakpoints: adding a breakpoint is done if a segment presents a variance that is too high, and places the breakpoint in the middle of that region. The removal of a breakpoint consists in the removing of the breakpoint that leads to the best fitting function.

5.2 An efficient segmentation method and model selection procedure for the analysis of array CGH data

In Chapter 4 we provided guidelines for the use of segmentation methods in statistical analysis. They were formulated as follows: (1) the determination of the parameters affected by the changes, (2) the procedure for estimating the breakpoints when the number of segments is fixed, (3) the estimation of the number of segments. It can be seen the from previous section that current approaches for array CGH data analysis have focused on point (2). Nevertheless, little information has been provided concerning the most appropriate modeling strategy for array CGH data (point (1)). Models $\mathcal{M}_1$ and $\mathcal{M}_2$ have been used by different authors, but none has specified the impact and the behavior of such models. In the following, we compare the two modeling strategies and show that model $\mathcal{M}_2$ seems more appropriate for array CGH data. Interestingly, dynamic programming has not focused much attention on the case of array CGH data segmentation. The only reference to this algorithm is Autio *et al.* (2003), who do not precisely explain its potential and do not assess its performance. This is why we propose to study the ability of the maximum likelihood method combined with dynamic programming to correctly locate the breakpoints with simulation studies. As for the estimation of the number of breakpoints (point (3)), current methods use ad-hoc procedures.

In February 2005 we proposed an efficient segmentation method and model selection procedure for the analysis of array CGH data. The purpose of this section is to present the main results of our work that has been published in the journal BMC Bioinformatics (Picard *et al.* (2005)). We have also developed a software program dedicated to segmentation methods for independent Gaussian processes. This software has been implemented with the MATLAB® software and can be used in a more general context than the analysis of array CGH data.

1- Determination of the parameter(s) affected by the changes

The determination of the parameter(s) affected by the changes requires some knowledge about the underlying phenomenon that is modelled. In the case of array CGH data, it is the $\log_2$ ratio of fluorescence intensities, which reflects the relative number of DNA copies throughout the genome. We have chosen to model this ratio with an independent Gaussian process. The problem is then to determine whether model $\mathcal{M}_1$ or $\mathcal{M}_2$ is more appropriate, *ie* if the variance of the process is homogeneous within segments, or constant between segments.

In this work, we show that a model with constant variance $\mathcal{M}_2$ is more appropriate for two reasons. The first argument is experimental, since it has been shown that the variability of the signal was the same for unaltered or altered clones, leading to a constant variability along the genome (Snijders *et al.* (2001)). The second argument is based on the behavior of both models. If we consider a model with heterogeneous variances, the addition of outliers within a segment will lead to an increase in the variance of the segments, whereas model $\mathcal{M}_2$ will tend to add a segment of small size in order to maintain a constant variance along the profile. The illustration of this behavior is provided in Figure 5.1. Outliers are a major concern in CGH array data analysis. If only one BAC is altered, whereas its neighbors are not, the conclusion could be that it is biologically relevant, or that it is a technical artefact, or that the altered BAC has been misannotated and is plotted at the wrong position. For these reasons we emphasize the use of a segmentation model with homogeneous variance for array CGH data.

2- Assessing the performances of dynamic programming

For the case where the number of segments is fixed, we studied the ability of dynamic programming to correctly locate the breakpoints given different amounts of noise. To do so, we considered two types of numerical simulations. In the first case (regular case), segments are of the same size and the jump in the mean d is constant and equals 1. In this case we show that the breakpoints are correctly located even if the amount of noise is large (frequency 1 if $d/\sigma = 10$, frequency 0.65 if $d/\sigma = 2$, and frequency 0.25 if $d/\sigma = 1$). Even if this probability decreases, the breakpoint is still located at positions close to the true ones. In the irregular case where segments are of different sizes and where jumps in the mean are heterogeneous, the tendency will be to ignore segments of small size that show a small jump in the mean. The reader is referred to Picard *et al.* (2005), pages 5 and 6 for more details.

3- Comparing penalization strategies to select the number of segments

Our scope is to compare different penalization strategies and to assess the most appropriate in the case of array CGH data segmentation. In Chapter 4 we presented two new penalized criteria that have been proposed by Lebarbier (2005) and Lavielle (2005). Our work constitutes the first comparison of the performance of those criteria, based on a simulation study. Briefly, we consider simulated data with a fixed number of segments and an increasing amount of noise. We show

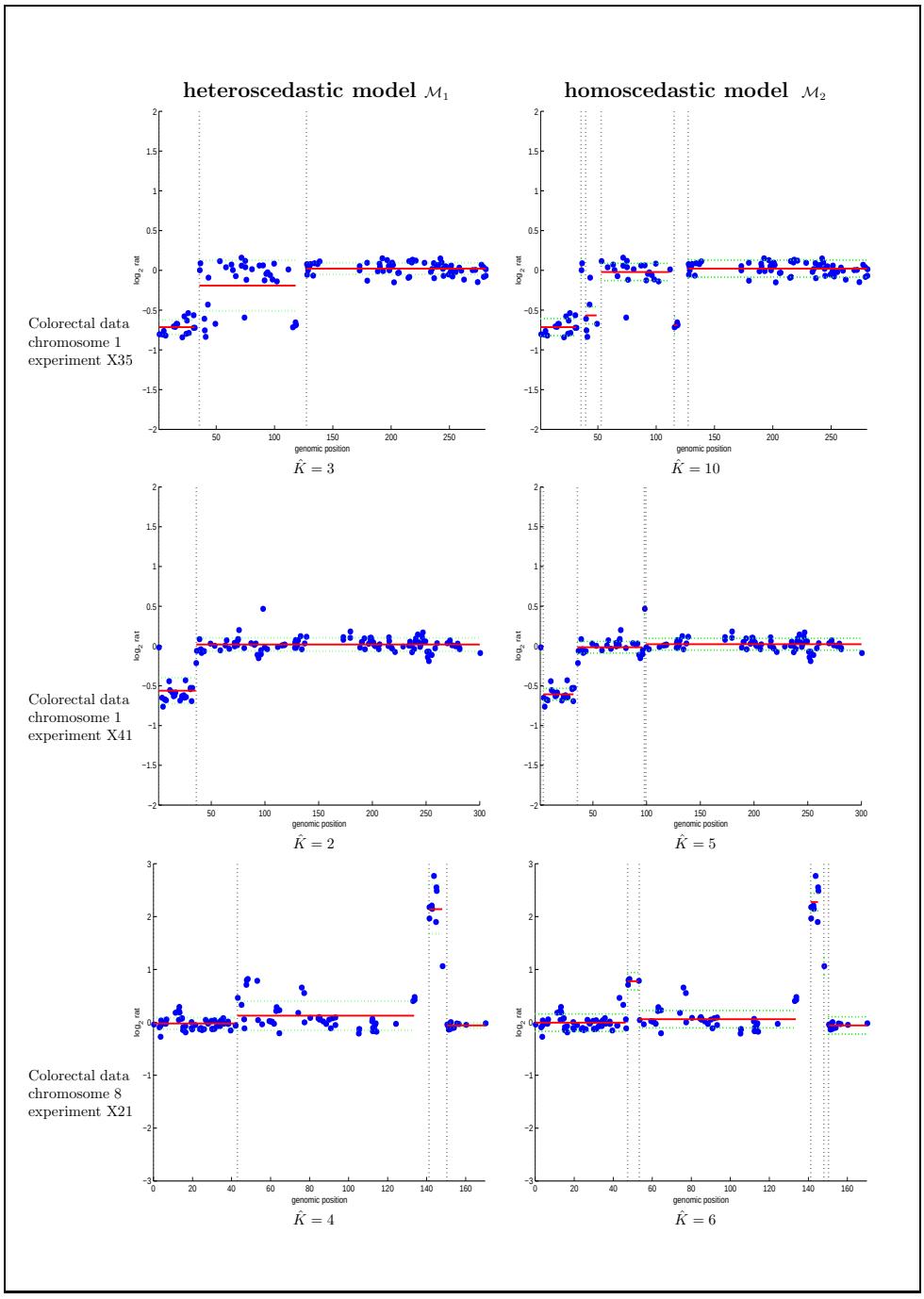


Figure 5.1: Comparison of segmentation results for model $\mathcal{M}_1$ and $\mathcal{M}_2$.

the adaptive behavior of the criterion defined by Lavielle (2005) which estimates a stable number of segments, whatever the noise. On the contrary, the criterion defined by Lebarbier (2005) tends to select a small number of segments when the amount of noise increases. This behavior is directly linked to the construction of the penalized criterion which tends to ignore small jumps in the mean in order to maintain a small quadratic risk. Finally we show the good performance of the adaptive criterion based on real data sets, and we recommend its use in the context of array CGH data analysis.

5.3 Comparison of segmentation methods

5.3.1 Comparison with Bayesian methods

To complete the application of segmentation methods to array CGH data, we choose to compare the resulting segmentations provided by the method we propose (based on dynamic programming and adaptive model selection) with the results provided by the Bayesian approach proposed by Lavielle and Lebarbier (2001). Since model $\mathcal{M}_2$ (changes in the mean and constant variance) has been shown to be more appropriate in the case of array CGH data analysis, we restrict our comparison to this model.

Let us briefly recall the analogies that exist between the two methods. In the Bayesian setting proposed by Lavielle and Lebarbier (2001), the aim is to recover the Maximum A Posteriori estimator of the change sequence $\{R_t\}$, whose *posterior* distribution is in the form:

$$p_T(R|y;\theta) = C_T(y;\theta)\exp\{-\frac{U_\theta(y,R)}{T}\},$$

where

$$U_\theta(y,R) = \phi\sum_{k=1}^{K_R}\sum_{t_{k-1}+1}^{t_k}(y_t-\bar{y}_k)^2 + \gamma K_R.$$

$U_\theta(y,R)$ is penalized contrast and the temperature parameter T aims at avoiding local maxima.

In Chapter 4 we compared different strategies to model the sequence of breakpoints. We discussed the reparametrization proposed by Lavielle and Lebarbier (2001), based on a sequence of constant size $\{R_t\}$ that indicates the *presence* of a breakpoint at coordinate t, compared to the *position* of a change as proposed in Green (1995). In this section, we want to discuss the combined effect of the reparametrization and the use of a Hastings-Metropolis algorithm at a fixed temperature.

Figure 5.2 shows the segmentation given by our method compared to the posterior probability of having one breakpoint at coordinate t on the genome. Results are similar in this example. Note that the posterior probabilities are close to 1 for all positions, since the jump detected is high compared to the variance of the

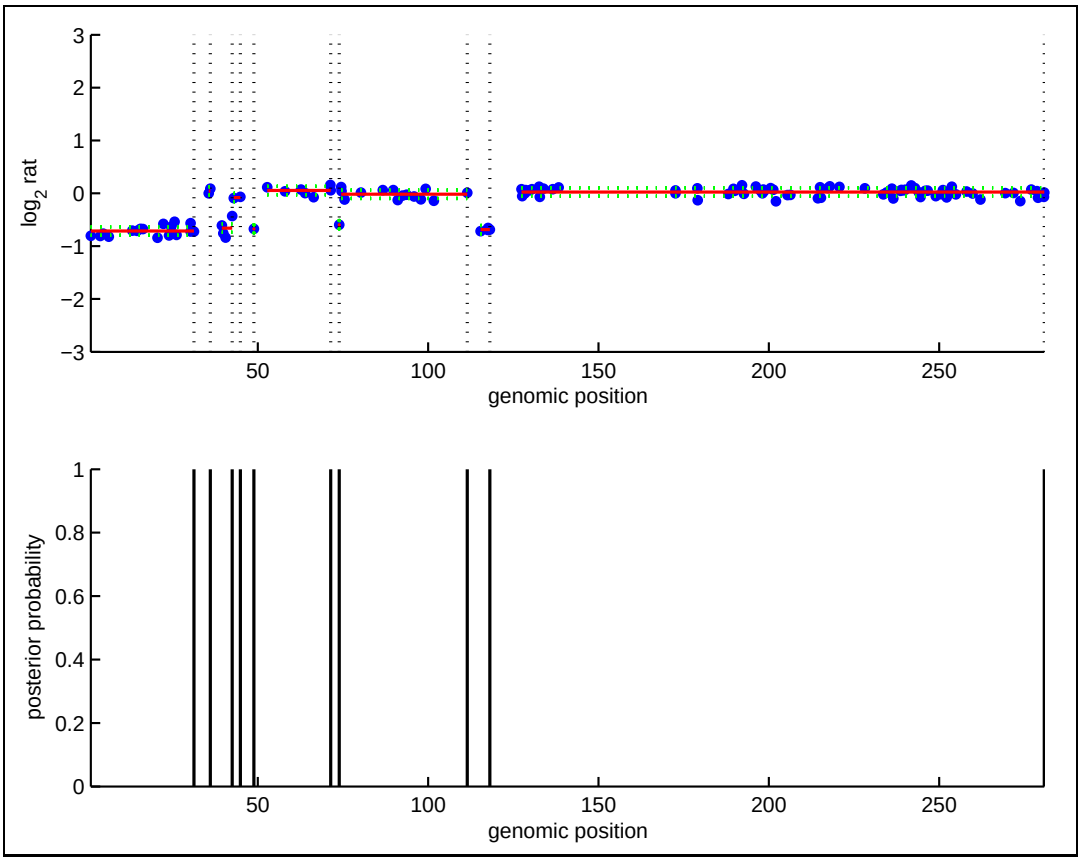


Figure 5.2: Comparison of segmentation results. Top: segmentation provided by dynamic programming with an adaptive model selection procedure. Bottom: posterior probability of the change-point sequence, according to the physical location on the genome. This posterior probability has been estimated with a temperature parameter fixed at $T = 0.5$. Data are described in Nakao *et al.* (2003), chromosome 1.

data. Lavielle and Lebarbier (2001) have shown that a jump of 0.4 in the mean with a variance of 0.1 was detected with posterior probability close to 1.

In the case of array CGH data segmentation, Figure 5.3 shows that the resulting segmentation is sensitive to the temperature parameter (to be compared with Figure 5.2), and the resulting segmentation presents fewer segments. Interestingly, the effect of a decrease in the temperature parameter is to remove some breakpoints, whereas it could have been to decrease the *posterior* probability of irrelevant changes. It appears that the tuning of the temperature parameter is important for the determination of the number of segments, rather than for the position of the breakpoints.

5.3.2 Comparison with smoothing methods

The last comparison that is presented concerns the segmentation method proposed by Hupe *et al.* (2004) and presented in Section 5.1. Segmentation of real CGH

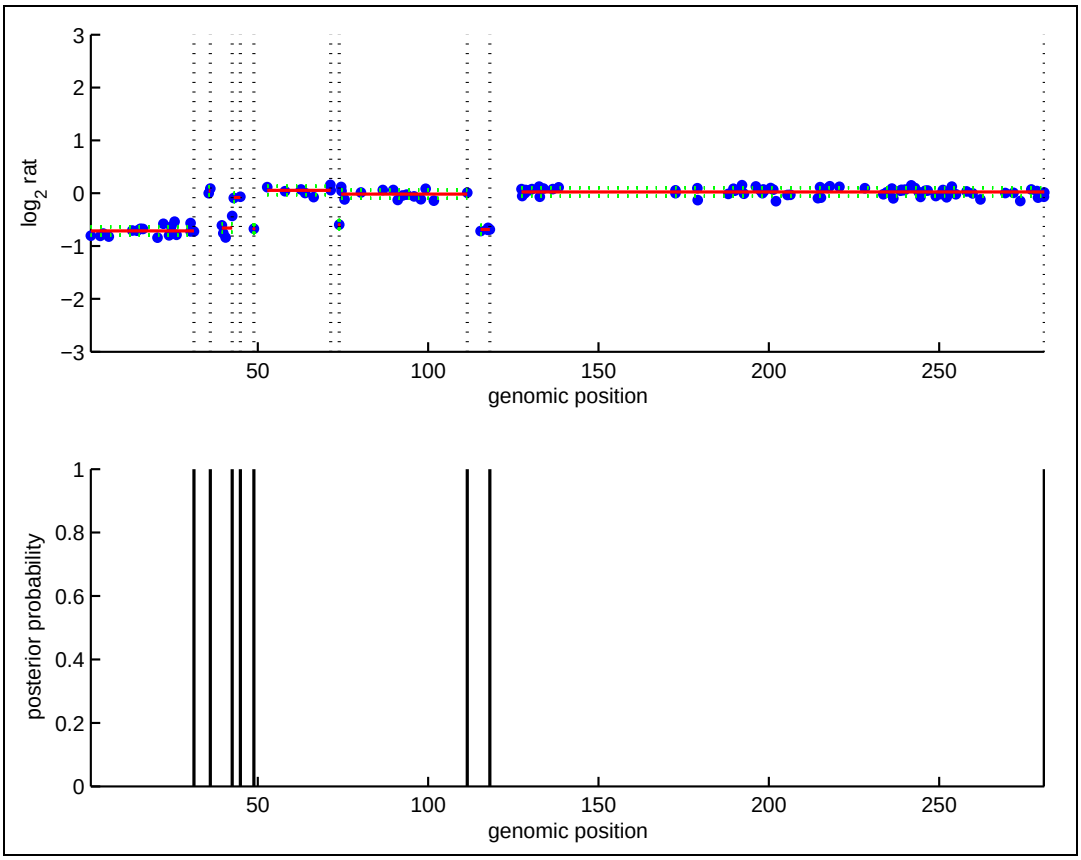


Figure 5.3: Comparison of segmentation results. Top: segmentation provided by dynamic programming with an adaptive model selection procedure. Bottom: posterior probability of the change-point sequence, according to the physical location on the genome. This posterior probability has been estimated with a temperature parameter fixed at $T = 0.1$. Data are described in Nakao *et al.* (2003), chromosome 1.

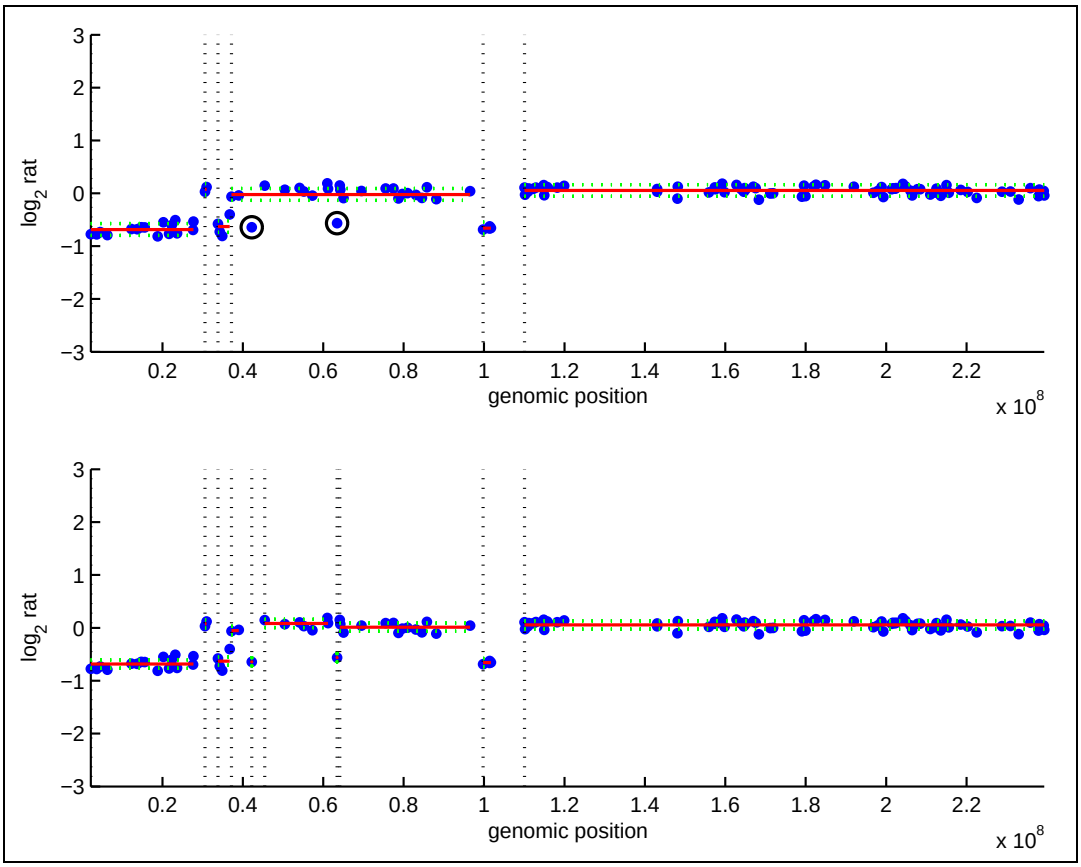


Figure 5.4: Comparison of segmentation results. Top: segmentation result provided by the smoothing method proposed by Hupe *et al.* (2004). Bottom: segmentation provided by dynamic programming with an adaptive model selection procedure. Circled dots represent outliers. Data are described in Nakao *et al.* (2003), chromosome 1, experiment X38.

data shows similar results for the number and the position of the breakpoints (Figures 5.4 and 5.5). Nevertheless, Hupe *et al.* (2004) propose an additional step in the analysis which consists in the identification of outliers, with a median absolute deviation criterion. The main difference between the two approaches lies in the definition of outliers, since model $\mathcal{M}_2$ can detect segments of size 1 that would be defined as outliers by Hupe *et al.* (2004). In the following, we will show that an automatic identification of outliers based on pure segmentation methods may not be appropriate in the case of array CGH data. Interestingly, the level of outliers in Figure 5.4 is close to the deleted segment at the beginning of the profile, indicating that "outliers" may represent small deleted regions. The segmentation/clustering model we propose in the following will address this problem.

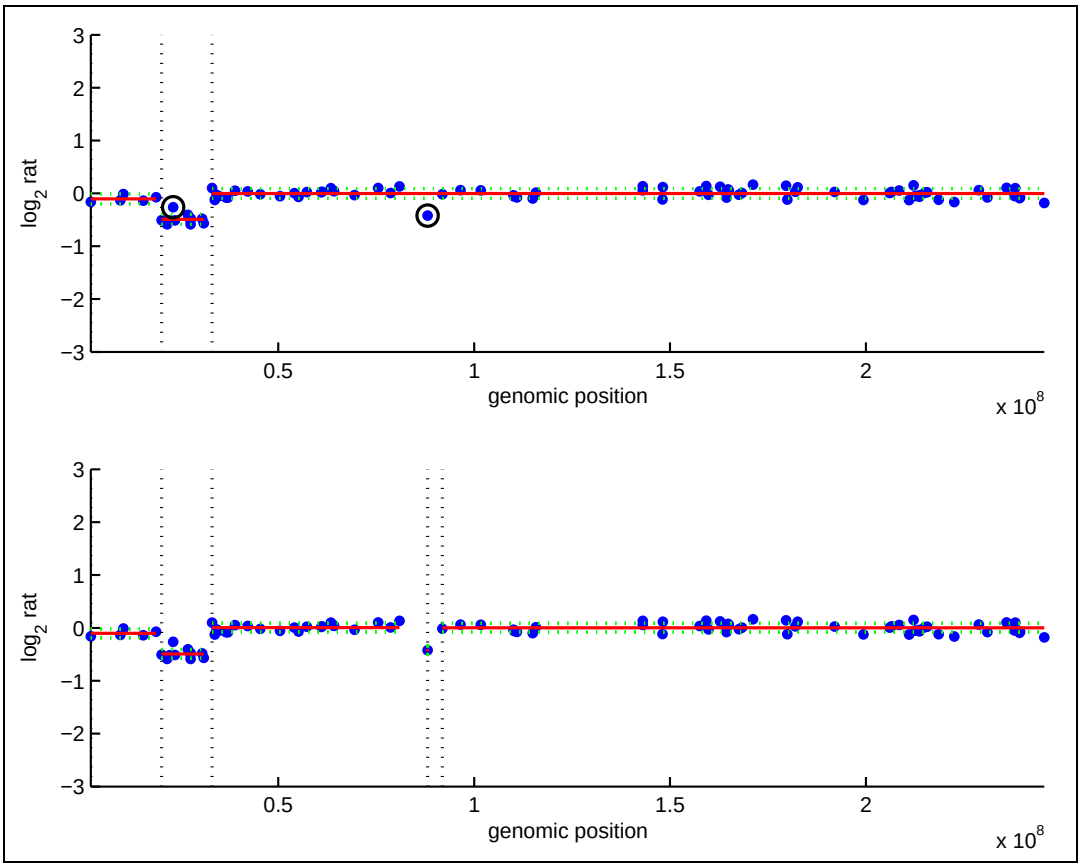


Figure 5.5: Comparison of segmentation results. Top: segmentation result provided by the smoothing method proposed by Hupe *et al.* (2004). Bottom: segmentation provided by dynamic programming with an adaptive model selection procedure. Circled dots represent outliers. Data are described in Nakao *et al.* (2003), chromosome 1, experiment X265.

5.4 Conclusion

In this chapter, we applied existing segmentation methods to the analysis of array CGH data. We focused on the comparison of different segmentation methods. We compared penalized criteria for choosing the number of segments in the profile (Section 5.2), and we compared different modelling strategies. In a recent paper Lai *et al.* (2005) compared 11 segmentation methods for the analysis of array CGH. This study is performed using simulated data for which the size of segments and the noise level vary. Using Receiver Operating Characteristic (ROC) curve, they show that the method we propose performs consistently well. In the following, we focus on the modification of our method in order to detect genomic regions on the genome affected by amplification/deletion but also to cluster these regions into a finite number of groups.

Part III

A new model for segmentation/clustering

Introduction

Motivations for a new statistical model

Part II was dedicated to a review of existing segmentation methods. Even if the application of such methods to array CGH data seems promising, we claim that they do not answer the specific question that is asked when analyzing such data. In array CGH data analysis the final objective is to determine which regions are altered by chromosomal aberrations and then to label each individual BAC with respect to their biological status. Clustering is the principal motivation of array CGH data analysis. A first idea could be to apply standard clustering methods such as mixture models, in order to cluster data points according to their $\log_2$ ratios. An example is provided in Figure 5.6. Nevertheless, array CGH data are characterized by their physical order on the genome and this information is not considered by the mixture. Consequently the clustering results lead to regions on the genome that can not be interpreted as homogeneous in terms of gene copy numbers (Figure 5.6, bottom).

It has been shown that segmentation methods are adapted to array CGH data since they provide a signal that can be interpreted as a succesion of homogeneous regions on the genome that share the same relative copy number on average. In one sense, segmentation methods could be viewed as space-ordered clustering methods. However they are based on the hypothesis that the signal should be homogeneous within segments and heterogeneous between segments. Consequently there is no constraint regarding the mean and variance of the process which can take any possible values. In the case of array CGH data, the problem is different. We know that the phenomenon under study is discrete by definition since gene copy numbers can only take a finite number of values. A first idea would be to constraint the mean of segments to be in the set $\{0, 1/2, 2/2, 3/2, ...\}$, reflecting possible gene copy number ratios between a diploid reference genome (2 copies) and a test genome. This model is called the "step-ratio" model, and is illustrated in Figure 5.7. Nevertheless, array CGH data are characterized by experimental and biological variability which hamper the use of such a simple model.

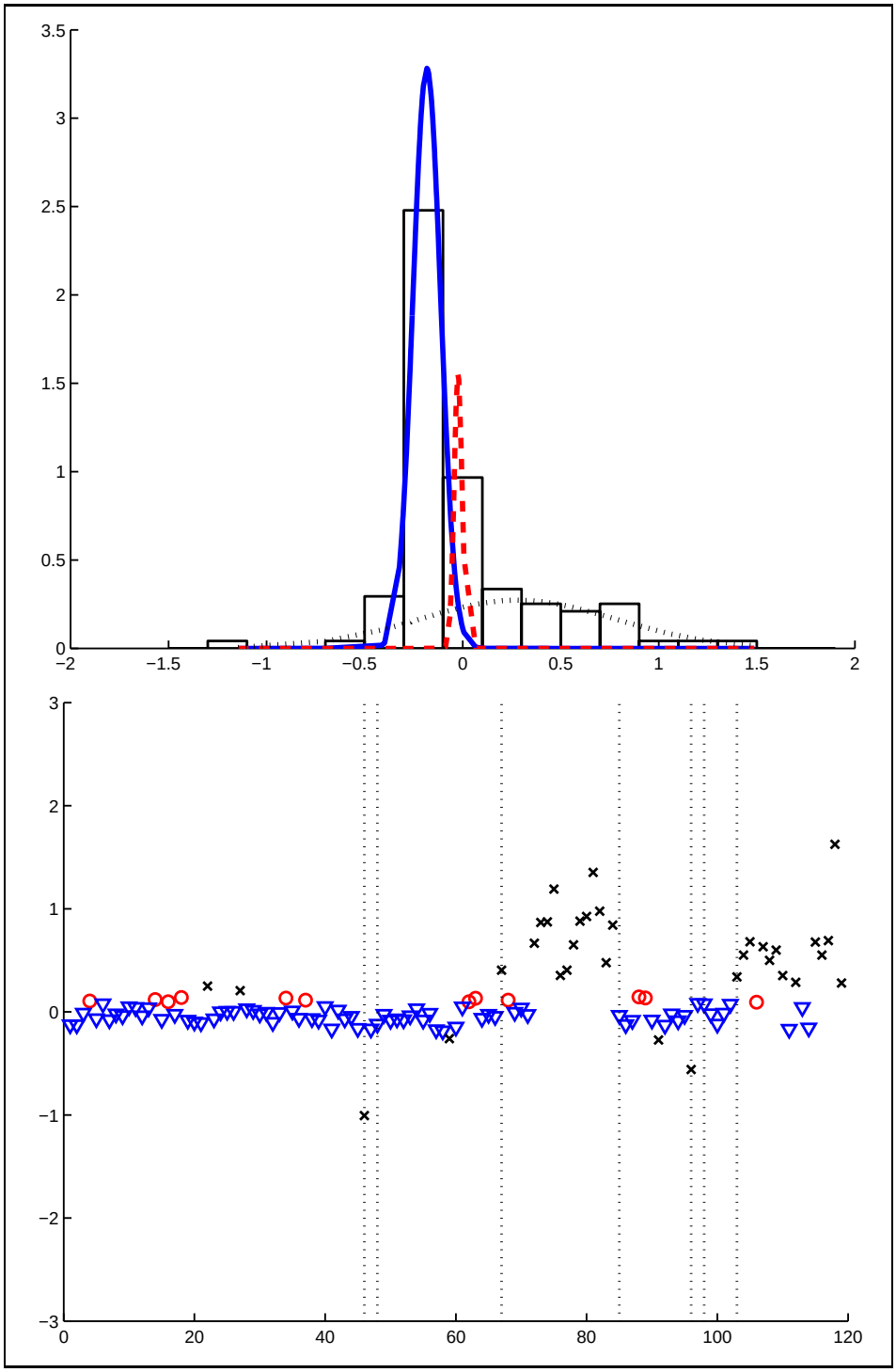


Figure 5.6: Application of a Gaussian mixture model to array CGH data. Top: estimated densities of a Gaussian mixture model with 3 clusters based on $\log_2$ ratios. Bottom: representation of the clustering result according to the physical order of data points on the genome. Data concern the breast cancer cell line Bt474, chromosome 1 described in Snijders *et al.* (2001).

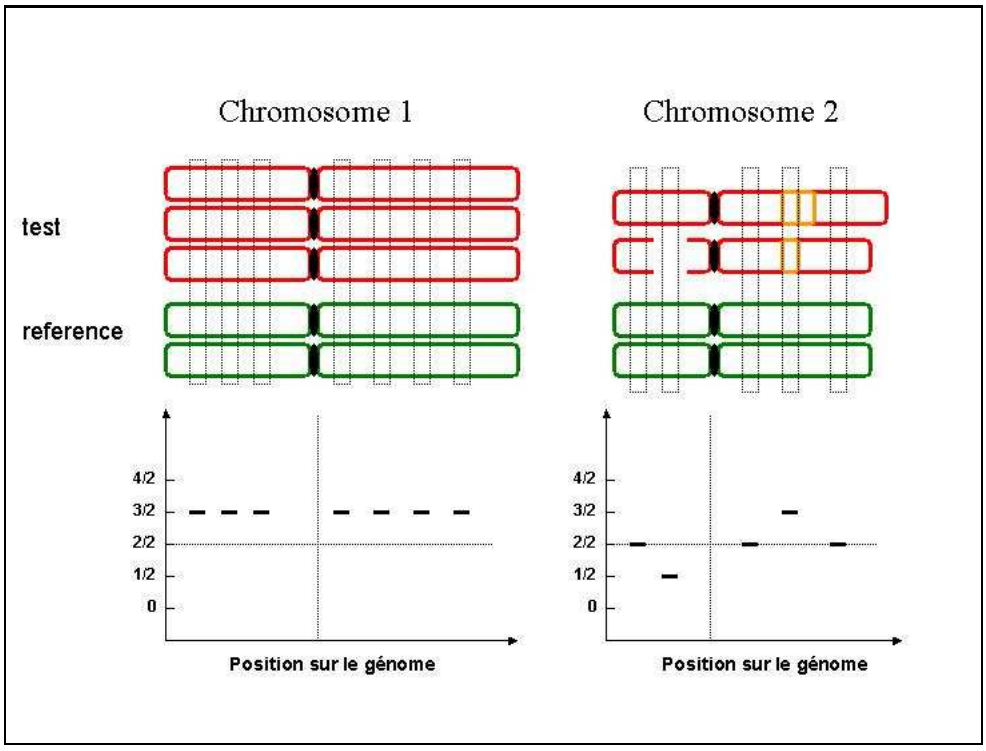


Figure 5.7: Principle of an array CGH experiment. Chromosome 1 shows a complete duplication, leading to constant copy number ratios of 3/2. Chromosome 2 shows one deletion (ratio 1/2) and one amplification (ratio 3/2), other regions being unaltered (ratio 2/2).

Biological variability

The most frequent phenomenon arising in the analysis of primary tumors is an imperfect dissection leading to normal cell contamination. Generally pathologists make sure that tumor samples do not contain more than 50 or even 30% of normal cells. The effect of normal cell contamination is a potential "dilution" of chromosomal defects, leading to ratios which do not necessarily reflect true copy numbers. Moreover even if the tumor sample is composed of altered cells, tumors are characterized by genomic instabilities and heterogeneities within a sample, since not all tumor cells may have acquired a given aberration. All these factors reduce the expected magnitude of copy number ratios and often make the estimation of a true copy number for a given clone impossible.

Technical variability

The second argument against the step-ratio model is that even if the underlying biological phenomenon is discrete and linear by definition (counting of DNA sequences), measurements are done using fluorescence measurements with continuous outputs. In addition to pure biological variability, the resulting measurements are corrupted by experimental noise which should be considered by any statistical model. Moreover, it appears that gene copy numbers can take any possible value. Amplifications of more than 10 fold are not rare in cancer genomics. Regarding this amplification, the technology could be affected by saturation effects which could modify the proportionality relationship that holds between gene copy numbers and fluorescence ratios (Snijders *et al.* (2001), see Chapter 1).

Introduction to a segmentation/clustering model

Considering all these arguments we propose a model that is more flexible compared with the simple step-ratio model. Since the final objective of array CGH data analysis is to cluster data points into a finite number of groups with biological interpretation, we propose to develop a new model that considers both structures: the organization of the data into segments, and the organization of segments into clusters of biological interest.

Segmentation methods aim at partitioning the data into a finite number of segments. Their objective is to facilitate the interpretation of the signal regarding its spatial coherence along the genome. This principle is illustrated in Figure 5.8, left. In the context of array CGH data, we claim that the model should consider a secondary structure of the signal. Our hypothesis is that the level of each segment belongs to a finite set $\{m_1, \ldots, m_P\}$, with P denoting a number of clusters to be selected and m_p the mean log-ratio for cluster p (to be estimated). This situation is illustrated in Figure 5.8, right. In this context, we are still interested in the detection of changes in the signal, but we also want to assign a label to segments in order to provide results with biological interpretation. For example, segments 2 and 3 should be clustered into the same group since their average signal is close. Similarly, segments 1 and 4 should be affected to the same group, even if they are not connex. The breakpoint that would have been detected between segments 2

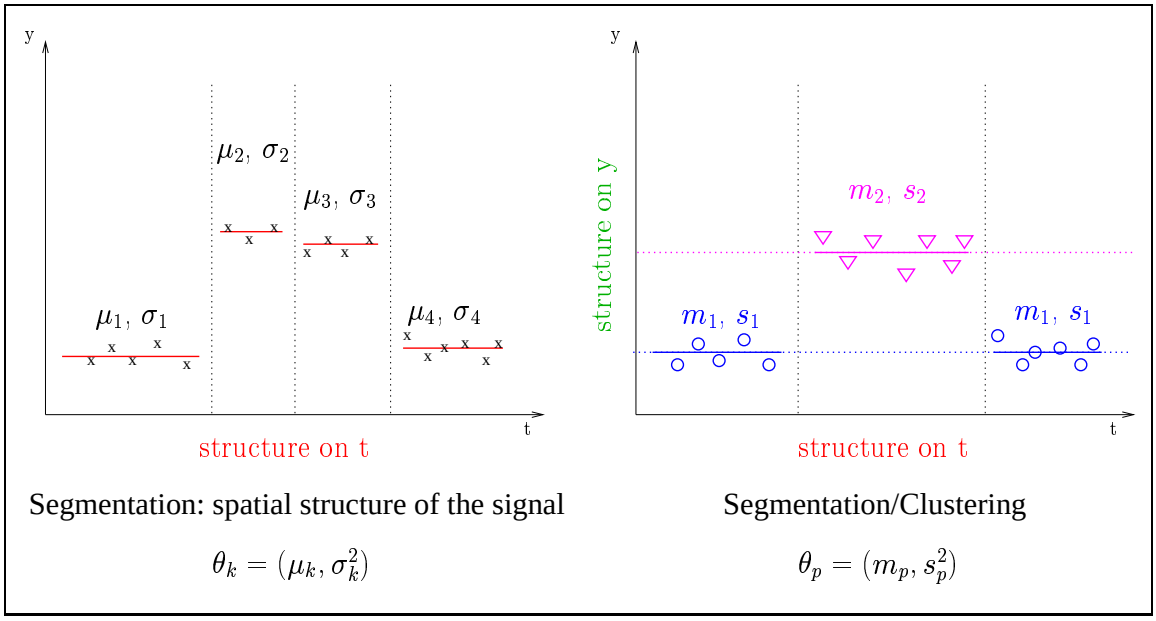


Figure 5.8: Principle of the segmentation/clustering model.

and 3 should be removed in order to cluster the segments into the same group. Similarly we think that the correct identification of segment 1 into cluster 1 could help for the detection of segment 4 since they belong to the same group. As a result the segmentation should change when considering the clustering objective of the model.

Many strategies exist for clustering. They can be based on heuristic procedures (k-means, hierarchical clustering for instance), or on a statistical model such as mixture models. We choose to adopt a model-based strategy for two reasons. The first one has been motivated previously: since the signal can take any possible values, we want to provide some uncertainty regarding the label given to each segment. This can be achieved with the use of *posterior* probabilities of membership calculated in the context of mixture models. The second reason is that the number of groups is unknown. A sample may contain different levels of amplifications or deletions and the analyst does not have any *prior* knowledge regarding this number. One objective of our methodology will be to estimate this number and model-based clustering provides theoretical tools for this purpose.

Mixture models themselves are not sufficient to model the nature of array CGH data, which are intrinsically structured into segments of homogeneous gene copy numbers. Nevertheless, they constitute the second basis of the construction of our model (the first one being segmentation models which have been described in Part II). This is why the first chapter of this part will be devoted to a review concerning model-based clustering.

The second chapter focuses on the definition of the segmentation/clustering model. As mentionned in Part II concerning the development of segmentation models, we can follow the same guidelines regarding the construction of our model which are:

1. determining the parameters affected by changes and according to which the data should be clustered,
2. choosing the estimation method and constructing an appropriate algorithm to estimate the parameters of the model which are the breakpoint instants and the mixture model parameters. This estimation is done for a fixed number of segments and clusters,
3. selecting the number of segments K, and the number of clusters P.

It appears that the model selection problem is unusual in our case, since K and P should be jointly selected. In Chapter 8 we will discuss possible strategies, based on existing methods for selecting K and P separately. Then we propose a heuristic for model selection.

While developing a new methodology for segmentation/clustering problems, it appears crucial to compare the new approach with existing methods. Hidden Markov models consitute the most widely used approach regarding this problem, and they have been applied to array CGH data analysis in Fridlyand *et al.* (2004). In order to be complete, we propose to discuss the formulation of our model, and to compare it with Hidden Markov models in Chapter 2. Our strategy will be also compared with a new method to analyse array CGH data proposed by Wang *et al.* (2005) and based on hierarchical clustering.

Chapter 6

Mixture models

The purpose of cluster analysis is to determine the inner structure of clustered data when no information other than the observed values is available. Interest in clustering has increased due to the emergence of new domains of application, such as astronomy, biology, physics and social sciences. Most clustering done in practice is based largely on heuristic or distance-based procedures, such as hierarchical agglomerative clustering or iterative relocation procedures. These methods present two major advantages: their construction is intuitive and the associated computational time is reasonable. Nevertheless their lack of statistical basis appears to be a limitation for their use, since classical questions in clustering such as the number of clusters, can hardly be theoretically handled by heuristic procedures.

Clustering methods based on probability models offer a principal alternative to heuristic-based algorithms. In this context the data are viewed as coming from a mixture of probability distributions, each representing a different cluster. In addition to clustering purposes, finite mixtures of distributions have been applied to a wide variety of statistical problems such as discriminant analysis, image analysis and survival analysis. To this extent finite mixture models have continued to receive increasing attention from both theoretical and practical points of view.

In a mixture model based approach to clustering the data are assumed to have arisen from a mixture of an initially specified number of populations in different proportions. Let us note $Y = \{Y_1, \ldots, Y_n\}$ a random sample of size n, where Y_t is a q-dimensional random vector with probability density function $f(y_t)$ on $\mathbb{R}^q$, and y_t its realization. We suppose that the density of Y_t can be written in the form

$$f(y_t) = \sum_{p=1}^{P} \pi_p f_p(y_t),$$

where $f_p(y_t)$ is a component density of the mixture, and π_p the weight of population p (with the constraints $0 \leq \pi_p \leq 1$ and $\sum_p \pi_p = 1$). In many applications the component densities are assumed to belong to some parametric family. In this case, they are specified as $f(y_t; \theta_p)$, where θ_p is the unknown vector of parameters of the postulated form for the p^{th} component of the mixture. Let $\psi = (\pi_1, \ldots, \pi_{P-1}, \theta_1, \ldots, \theta_P)$ denote the vector containing all the unknwon parameters of the mixture. Section 6.1 will be devoted to the formulation of mixture models in the parametric context.

Since we are interested in clustering it appears that one information is missing regarding the observed sample: the assignment of data points to the different clusters. A new random variable is introduced and noted Z_{tp} that equals 1 if data point y_t belongs to population p, and 0 otherwise. We suppose that variables $\{Z_1, \ldots, Z_n\}$ are independent (with $Z_t = \{Z_{t1}, \ldots, Z_{tP}\}$) and that the conditional density of Y_t given $\{Z_{tp} = 1\}$ is $f(y_t; \theta_p)$. Therefore variables Z_{tp} can be viewed as categorial variables that indicate the labelling of the data points. Thus Z_t is assumed to be distributed according to a multinomial distribution consisting of one draw on p categories with probabilities $\pi_1, \ldots, \pi_P$:

$$\{Z_{t1}, \ldots, Z_{tP}\} \sim \mathcal{M}(1; \pi_1, \ldots, \pi_P).$$

In terms of clustering, the p^{th} mixing proportion can be viewed as the prior probability that one data point belongs to population p. The posterior probability of Z_{tp} given the observed value of y_t will be central for clustering purposes:

$$\tau_{tp} = \Pr\{Z_{tp} = 1 | Y_t = y_t\} = \frac{\pi_p f(y_t; \theta_p)}{\sum_{\ell=1}^{P} \pi_\ell f(y_t; \theta_\ell)}.$$

In order to formalize the incomplete data structure of mixture models, let $X = (Y, Z)$ denote the complete data vector, whose only component being observed is Y. This reformulation clearly shows that mixture models can be viewed as a particular example of models with hidden structure such as hidden Markov models or models with censored data.

If the label of each data point was observed, the estimation of the mixture parameters would be straightforward since the parameters of each density component $f(y_t; \theta_p)$ could be estimated only via the data points from population p. Nevertheless the categorial variables are hidden, and the estimation can only be based on the observed data Y. The main reason for the important work on estimation methodology for mixtures is that explicit formulas for parameter estimates are not available in a closed form, leading to the need for iterative estimation procedures. Fitting mixture distributions can be handled by a wide variety of techniques, such as graphical methods, the method of moments, maximum likelihood and Bayesian approaches. It has only been since 30 years that considerable advances have been made in the fitting of mixture models, especially via the maximum likelihood method, thanks to the publication of Dempster *et al.* (1977) and to the introduction of the EM algorithm.

The purpose of the EM algorithm is the iterative computation of maximum likelihood estimators when observations can be viewed as incomplete data. The basic idea of the EM algorithm is to associate a complete data model to the incomplete structure that is observed in order to simplify the computation of maximum likelihood estimates. Similarly, a complete data likelihood is associated to the complete data model. The EM algorithm exploits the simpler MLE computation of the complete data likelihood to optimize the observed data likelihood. Section 6.2 is devoted to the general description of the EM algorithm and to its general properties. Despite a wide range of successful applications and the important work on its properties, the EM algorithm presents two intrinsic limitations: it appears to be slow to converge and as many iterative procedures, is sensitive to

the initialization step. This has lead to the development of modified versions of the EM algorithm, which will be detailed in section 6.2.

Once the mixture model has been specified and its parameters have been estimated, one central question remains: "How many clusters?". Mixture models present a main advantage compared with heuristic cluster algorithms in which there is no established method to determine the number of clusters. With the underlying probability model, the problem of choosing the number of components can be reformulated as a statistical model choice problem. Testing for the number of components in a mixture appears to be difficult since the classical likelihood ratio test does not hold for mixtures. On the contrary, criteria based on penalized likelihood, such as the Bayesian Information Criterion (BIC) have been successfully applied to mixture models. Nevertheless, it appears that those criteria do not consider the specific objective of mixture models in the clustering context. This has lead to the construction of classification-based criteria. These criteria will be discussed in Section 6.3.

6.1 Mixture models in the parametric context

6.1.1 Definition of the model

Let $Y = \{Y_1, \ldots, Y_n\}$ denote a random sample of size n where Y_t is a vector of $\mathbb{R}^q$, y_t its realization and $f(y_t)$ its density function. In the mixture model context the density of Y_t is supposed to be a mixture of P parametric densities such that:

$$f(y_t; \psi) = \sum_{p=1}^{P} \pi_p f(y_t; \theta_p), \tag{6.1}$$

with the constraint $\sum_{p=1}^{P} \pi_p = 1$, P being fixed. Coefficients π_p can be viewed as the weights of the p^{th} component of the mixture, which is characterized by parameter θ_p. $\psi = (\pi_1, \ldots, \pi_{P-1}, \theta_1, \ldots, \theta_P)$ denotes the vector of parameters of the model.

Mixture models are reformulated as an incomplete data problem since the assignment of the observed data is unknown. If we note $X_t = \{Y_t, Z_t\}$ the complete data vector whose only component being observed is Y_t, its density function is then:

$$g(x_t; \psi) = \prod_{p=1}^{P} [\pi_p f(y_t; \theta_p)]^{z_{tp}}. \tag{6.2}$$

6.1.2 Clustering via mixture models

When mixture models are used in the clustering context, the aim is to provide a partition of the data into P groups, with P being fixed. The populations' weights are interpreted as *prior* probabilities of belonging to a given population.

$\Pr\{Z_{tp} = 1\} = \pi_p$ represents the probability to assign one data point to population p when the only available information about the data are the weights of each group.

In the complete data specification the clustering procedure aims at recovering the associated label variables $z_1, \ldots, z_n$ having observed $y_1, \ldots, y_n$. After the mixture model has been fitted and its parameter ψ has been estimated, a probabilistic clustering of the observations is provided in terms of their *posterior* probabilities of component membership:

$$\hat{\tau}_{tp} = \Pr_{\hat{\psi}}\{Z_{tp} = 1 | Y_t = y_t\} = \frac{\hat{\pi}_p f(y_t; \hat{\theta}_p)}{\sum_{\ell=1}^{P} \hat{\pi}_\ell f(y_t; \hat{\theta}_\ell)}.$$

Probabilities $\hat{\tau}_{t1}, \ldots, \hat{\tau}_{tP}$ are the estimated probabilities that data point y_t belongs to the first, second, ..., P^{th} component of the mixture.

Instead of fuzzy classification results each data point can be assignated to a particular population with the maximum *a posteriori* rule (MAP):

$$\hat{z}_{tp} = \begin{cases} 1 & \text{if } p = \underset{\ell}{\text{Argmax}}\{\hat{\tau}_{t\ell}\}, \\ \\ 0 & \text{otherwise.} \end{cases}$$

6.2 Fitting mixture models via the EM algorithm

The estimation of the parameters of a mixture can be handled by a variety of techniques from graphical to Bayesian methods (see Titterington *et al.* (1985) for an exhaustive review of those methods). Nevertheless the maximum likelihood method has focused many attentions, mainly due to the existence of an associated statistical theory. Given a sample of n independent observations from a mixture defined in 6.1.1, the likelihood function is:

$$\mathcal{L}(y; \psi) = \prod_{t=1}^{n} \left\{ \sum_{p=1}^{P} \pi_p f(y_t; \theta_p) \right\}.$$

The particularity of mixture models is that the maximization of the likelihood defined above with respect to ψ is not straightforward and requires iterative procedures. The EM algorithm has become the method of choice for estimating the parameters of a mixture model, since its formulation leads to straighforward estimators.

6.2.1 General presentation of the EM algorithm

In the incomplete data formulation of mixture models let us note $\mathcal{X}$ the complete data sample space from which x arises, $\mathcal{Y}$ the observed sample space and $\mathcal{Z}$ the hidden sample space. It follows that $\mathcal{X} = \mathcal{Y} \times \mathcal{Z}$, and $x = (y, z)$. The density of the observed data X can be written:

$$g(x; \psi) = f(y; \psi) k(z | y; \psi),$$

where $f(y;\psi)$ is the density of the observed data and $k(z|y;\psi)$ is the conditional density of the missing observations given the data. This leads to the definition of different likelihoods: the observed/incomplete-data likelihood $\mathcal{L}(y;\psi)$ and the unobserved/complete-data likelihood $\mathcal{L}^c(x;\psi)$. These likelihoods are linked with the relationship:

$$\log \mathcal{L}^c(x;\psi) = \log \mathcal{L}(y;\psi) + \log k(z|y;\psi),$$

with

$$\log \mathcal{L}^c(x;\psi) = \sum_{t=1}^{n} \log g(x_t;\psi),$$

and

$$\log k(z|y;\psi) = \sum_{t=1}^{n}\sum_{p=1}^{P} z_{tp} \log \mathbb{E}\left\{Z_{tp}|Y_t = y_t\right\}.$$

Since the hidden variables are not observed, the EM machinery consists of the indirect optimization of the incomplete-data likelihood *via* the iterative optimization of the conditional expectation of the complete-data likelihood using the current fit for ψ. If we note $\psi^{(h)}$ the value of the parameter at iteration h, it follows that:

$$\log \mathcal{L}(y;\psi) = Q(\psi;\psi^{(h)}) - H(\psi;\psi^{(h)}), \tag{6.3}$$

with conventions:

$$\begin{aligned} Q(\psi;\psi^{(h)}) &= \mathbb{E}_{\psi^{(h)}}\left\{\log \mathcal{L}^c(X;\psi)|Y\right\}, \\ H(\psi;\psi^{(h)}) &= \mathbb{E}_{\psi^{(h)}}\left\{\log k(Z|Y;\psi)|Y\right\}, \end{aligned}$$

where $\mathbb{E}_{\psi^{(h)}}\{\cdot\}$ denotes the expectation operator, taking the current fit $\psi^{(h)}$ for ψ.

The EM algorithm consits of two steps:

- E-step: calculate $Q(\psi;\psi^{(h)})$,
- M-step: choose $\psi^{(h+1)} = \underset{\psi}{\text{Argmax}}\left\{Q(\psi;\psi^{(h)})\right\}$.

The E- and M- steps are repeated alternatively until the difference $|\psi^{(h+1)} - \psi^{(h)}|$ changes by an arbitrarily small amount. Note that another stopping rule could be the difference of log-likelihoods between two steps, $|\log \mathcal{L}(y;\psi^{(h+1)}) - \log \mathcal{L}(y;\psi^{(h)})|$. However if the log-likelihood is "flat" with respect to ψ this difference can be stable whereas parameter $\psi^{(h)}$ keeps changing.

The key property of the EM algorithm established by Dempster *et al.* (1977) is that the incomplete data log-likelihood increases after each iteration of the algorithm. The proof of this theorem is based on the definition of the M-step that ensures

$$Q(\psi;\psi^{(h+1)}) \geq Q(\psi;\psi^{(h)}),$$

while the application of the Jensen inequality gives

$$H(\psi;\psi^{(h+1)}) \leq H(\psi;\psi^{(h)}).$$

Put together and considering relation 6.2.1, these inequalities ensure the monotonicity of the likelihood sequence:

$$\log \mathcal{L}(y;\psi^{(h+1)}) \geq \log \mathcal{L}(y;\psi^{(h)}).$$

This inequality proves that the EM sequence of likelihoods must converge if the likelihood is bounded above.

6.2.2 Formulation of the EM algorithm for mixture models

When applied to the special case of mixture models the log-likelihoods are written in the form:

$$\begin{aligned}
\log \mathcal{L}(y;\psi) &= \sum_{t=1}^{n} \log f(y_t;\psi) = \sum_{t=1}^{n} \log \left\{ \sum_{p=1}^{P} \pi_p f(y_t;\theta_p) \right\} \\
\log \mathcal{L}^c(x;\psi) &= \sum_{t=1}^{n} \log g(x_t;\psi) = \sum_{t=1}^{n} \sum_{p=1}^{P} z_{tp} \log \left\{ \pi_p f(y_t;\theta_p) \right\}
\end{aligned}$$

Since the complete data log-likelihood is linear in the unobservable data z_{tp} the E-step only requires the computation of the conditional expectation of the missing information given the observed data y_t, using the current fit $\psi^{(h)}$ for ψ. It gives

$$Q(\psi;\psi^{(h)}) = \sum_{t=1}^{n} \sum_{p=1}^{P} \mathbb{E}_{\psi^{(h)}} \left\{ Z_{tp} | Y_t = y_t \right\} \log \left\{ \pi_p f(y_t;\theta_p) \right\},$$

with

$$\mathbb{E}_{\psi^{(h)}} \left\{ Z_{tp} | Y_t = y_t \right\} = \Pr_{\psi^{(h)}} \left\{ Z_{tp} = 1 | Y_t = y_t \right\} = \tau_{tp}^{(h)},$$

and

$$\tau_{tp}^{(h)} = \frac{\pi_p^{(h-1)} f(y_t;\theta_p^{(h-1)})}{\sum_{\ell=1}^{P} \pi_\ell^{(h-1)} f(y_t;\theta_\ell^{(h-1)})}.$$

Then

$$Q(\psi;\psi^{(h)}) = \sum_{t=1}^{n} \sum_{p=1}^{P} \tau_{tp}^{(h)} \log \left\{ \pi_p f(y_t;\theta_p) \right\}.$$

The M-step requires the global maximization of $Q(\psi;\psi^{(h)})$ with respect to ψ to give an updated estimate $\psi^{(h+1)}$.

For finite mixture models, the estimation of the mixing proportions is done via constrained maximization of the incomplete-data log-likelihood which gives:

$$\hat{\pi}_p^{(h+1)} = \frac{\sum_{t=1}^{n} \tau_{tp}^{(h)}}{n}.$$

This estimator has a natural interpretation: it summarizes the contribution of each data point y_t to the p^{th} component of the mixture via its *posterior* probability of membership. As for the updating of θ, it is obtained as an appropriate root of

$$\sum_{t=1}^{n}\sum_{p=1}^{P} \tau_{tp}^{(h)} \frac{\partial \log f(y_t;\theta_p)}{\partial \theta} = 0.$$

6.2.3 Information matrix using the EM algorithm

Once the parameters of the mixture have been estimated via maximum likelihood, a natural question is to assess the standard errors of the estimator $\hat{\psi}$. This can be done with the evaluation of the expected information matrix

$$\mathcal{I}(\psi) = \mathbb{E}_Y \left\{ \frac{-\partial^2}{\partial\psi\partial\psi^T} \log \mathcal{L}(Y;\psi) \right\},$$

with $\log \mathcal{L}(y;\psi)$ being the incomplete-data likelihood calculated on the available observations, and $\mathbb{E}_Y\{\cdot\}$ designating the expectation operator with respect to the random variable Y.

In practice this quantity is often estimated by the observed information matrix calculated at $\hat{\psi}$, $I(\hat{\psi}, y)$, with the relationship

$$\mathcal{I}(\psi) = \mathbb{E}_Y \{I(\psi;Y)\}.$$

Efron and Hinkley (1978) have provided a justification for this approximation. Since the data Y are considered as incomplete within the EM framework, $I(\psi;Y)$ will be denoted as the incomplete-data observed information matrix.

The use of the EM algorithm is often motivated by the analytic form of the observed-data likelihood, whose gradient or curvature matrices are difficult to derive analytically (which is typically the case for mixture models). As the estimation problem has been solved using the missing-data framework of EM, the derivation of the information matrix $I(\psi;y)$ can be simplified using the missing information principle introduced by Woodbury (1971).

Missing information principle

If we consider the formulation of mixtures as a missing-data problem, we define the complete-data observed information matrix based on the complete-data log-likelihood:

$$I^c(\psi;x) = \frac{-\partial^2}{\partial\psi\partial\psi^T} \log \mathcal{L}^c(x;\psi).$$

Since the incomplete data and the complete data likelihood are linked by definition:

$$\log \mathcal{L}(y;\psi) = \log \mathcal{L}^c(x;\psi) - \log k(z|y;\psi),$$

on differentiating both sides twice with respect to ψ, we have

$$I(\psi;y) = I^c(\psi;x) - I^m(\psi,z),$$

where

$$I^m(\psi, z) = \frac{-\partial^2}{\partial\psi\partial\psi^T} \log k(z|y;\psi)$$

is the missing-data observed information matrix. This term can be viewed as the "missing information", the consequence of having observed only y and not z.

Since the complete-data are not fully observed, we take the conditional expectation of both sides over Y that yields to:

$$I(\psi; y) = \mathbb{E}_{X|Y}\left\{I^c(\psi; X)\right\} - \mathbb{E}_{Z|Y}\left\{I^m(\psi, Z)\right\} \tag{6.4}$$

Then the problem is to formulate the conditional expectations of $I^c(\psi; x)$ and $I^m(\psi, z)$ in directly computable terms within the EM framework.

Extracting the observed information matrix in terms of the complete-data likelihood

Let us introduce the score notation such that:

$$\begin{aligned} S(y;\psi) &= \frac{\partial}{\partial\psi} \log \mathcal{L}(y;\psi), \\ S^c(x;\psi) &= \frac{\partial}{\partial\psi} \log \mathcal{L}^c(x;\psi). \end{aligned}$$

Louis (1982) gives a formulation of the missing information matrix, in the form:

$$\mathbb{E}_{Z|Y}\left\{I^m(\psi, Z)\right\} = \mathbb{E}_{X|Y}\left\{S^c(X;\psi)S^c(X;\psi)^T\right\} - S(y;\psi)S(y;\psi)^T,$$

meaning that the all the conditional expectations calculated in 6.4 can be computed in the EM algorithm only using the conditional expectation of the gradient and curvature of the complete-data likelihood.

Since $S(y;\psi) = 0$ for $\psi = \hat{\psi}$, Formula 6.4 is restated as:

$$I(\hat{\psi}; y) = \mathbb{E}_{X|Y}\left\{I^c(\psi; X)\right\}\Big|_{\psi=\hat{\psi}} - \mathbb{E}_{X|Y}\left\{S^c(X;\psi)S^c(X;\psi)^T\right\}\Big|_{\psi=\hat{\psi}}.$$

Hence the observed information matrix of the initial incomplete-data problem can be computed as the conditional moments of the gradient and curvature matrix of the complete-data likelihood introduced in the EM framework.

6.2.4 Convergence properties of the EM algorithm

It has been seen in previous sections that the EM algorithm generates a sequence $\left(\psi^{(h)}\right)_{h\geq 0}$ which increases the incomplete data log-likelihood at each iteration. The convergence of this EM-generated sequence has been studied by many authors, such as Dempster *et al.* (1977) and Wu (1983). Under some regularity conditions of the model, Wu (1983) shows the convergence of the sequence $\psi^{(h)}$ to a stationary point of the incomplete-data likelihood. The convergence of the EM algorithm to a local maximum of the incomplete data likelihood has also

been established by Wu (1983) under restrictive hypothesis, that have been released by Delyon *et al.* (1999). One important theorem is provided by Wu (1983):

Suppose that $Q(\psi, \Phi)$ is continuous in both ψ and Φ, then all the limit points of any instance $\{\psi^{(h)}\}$ of the EM algorithm are stationary points of $\mathcal{L}(\psi)$ and $\mathcal{L}(\psi^{(h)})$ converges monotonically to some value $\mathcal{L}^*$ for some stationary point ψ^*.

Moreover in many practical situations $\mathcal{L}^*$ will be a local maximum. In general if the likelihood has several stationary points the convergence of an EM sequence to a local/global maximum or to a saddle point will depend on the choice of the starting value $\psi^{(0)}$, unless the likelihood is unimodal.

6.2.5 Modified versions of the EM algorithm

Despite appealing features, the EM algorithm presents some well documented shortcomings: the resulting estimate $\hat{\psi}$ can strongly depend on the starting position $\psi^{(0)}$, the rate of convergence can be slow and it can provide a saddle point of the likelihood function rather than a local maximum. For these reasons several authors have proposed modified versions of the EM algorithm: deterministic improvements (Louis (1982), Meilijson (1989), Green (1990) for instance), and stochastic modifications (Broniatowski *et al.* (1983) Celeux and Dielbolt (1985) Wei and Tanner (1990), Delyon *et al.* (1999)).

Broniatowski *et al.* (1983) proposed a Stochastic EM algorithm (SEM) which provides an attractive alternative to EM. The motivation of the simulation step (S-step) is based on the Stochastic Imputation Principle, where the purpose of the S-step is to fill-in for the missing data z with a single draw from $k(z|y; \psi^{(h)})$. This imputation of z is based on all the available amount of information about ψ and provides a pseudo complete sample. More precisely the current *posterior* probabilities $\tau_{tp}^{(h)}$ are used in the S-step wherein a single draw from distribution $\mathcal{M}_P(1; \tau_{t1}^{(h)}, \ldots, \tau_{tP}^{(h)})$ is used to assign each observation to one of the component of the mixture. The deterministic M-Step and the stochastic S-Step generate a Markov Chain $\psi^{(h)}$ which converges to a stationary distribution under mild conditions. In pratice a number of iterations is required as a burn in period to allow $\psi^{(h)}$ to approach its stationary regime. In mixture models 100-200 iterations are often used for burn in.

This stochastic step can be viewed as a random perturbation of the sequence $\psi^{(h)}$ generated by EM. This perturbation prevents the algorithm from staying near an unstable fixed point of EM, and prevents stable fixed points corresponding to insignificant local maxima of the likelihood. The Stochastic EM algorithm provides an interesting alternative to the limitations of EM, concerning local maxima and starting values.

Other stochastic versions of the EM algorithm have been proposed, among them, the Stochastic Annealing EM algorithm (SAEM, Celeux and Dielbolt (1992)) which is a modification of SEM, the Monte Carlo EM (Wei and Tanner (1990)), which replaces analytic computation of the conditional expectation of the complete-data log-likelihood by a Monte Carlo approximation, and a stochastic approximation of EM (Delyon *et al.* (1999)). Nevertheless, empirical studies from Dias and

Wedel (2004) and Biernacki *et al.* (2003) suggest the practical use of SEM in the context of mixture models, for its simplicity of implementation compared with Monte Carlo-based improvements, for its quick rate of convergence, and for its property to avoid spurious local maximizers.

6.3 Choosing the number of clusters via model selection criteria

Choosing the number of clusters is often the first question that is asked by/to the analyst. Two approaches can be considered to answer this question. The first one can be to fix this number and to propose different classifications. Since every clustering method (heuristically or model-based) can be run for a fixed number of groups, this strategy can be applied to any method. Nevertheless, the question can be to score different classifications with different numbers of clusters. In the model-based context, the choice of the number of clusters can be formulated as a model selection problem, and it can be performed with a penalized criterion, such as:

$$\log \mathcal{L}_P(y; \hat{\psi}) - \beta pen(P),$$

with $\log \mathcal{L}_P(y; \hat{\psi})$ being the observed data log-likelihood for a mixture with P clusters, calculated at $\psi = \hat{\psi}$, β a positive constant and $pen(P)$ an increasing function with respect to the number of clusters.

6.3.1 Bayesian approaches for model selection

As previously described in the context of segmentation methods (4), the purpose of model selection is to select a candidate model m_i among a finite collection of models $\{m_1, \ldots, m_\ell\}$, in order to estimate function f from which the data $Y = \{Y_1, \ldots, Y_n\}$ are drawn. Each model is characterized by a density g_{m_i} whose parameters ψ_i are of dimension ν_i.

In the Bayesian context, ψ_i and m_i are viewed as random variables with *prior* distributions noted $\Pr\{m_i\}$ and $\Pr\{\psi_i|m_i\}$ for ψ_i when model m_i is fixed. This formulation is flexible since additional information can be modelled through *prior* distributions, and if no information is available a non-informative prior can be used. The Bayesian Information Criterion (BIC) developed by Schwartz (1978) aims at selecting the model which maximizes the posterior probability $\Pr\{m_i|Y\}$. Using the Bayes formula:

$$\Pr\{m_i|Y\} = \frac{\Pr\{Y|m_i\} \Pr\{m_i\}}{\Pr\{Y\}},$$

and considering the case where the prior distribution $\Pr\{m_i\}$ is non informative, the search for the best model only requires the computation of distribution $\Pr\{Y|m_i\}$ which is the integrated likelihood of the data for model m_i. This distribution can be approximated using the Laplace approximation method (see

Lebarbier and Mary-Huard (2004) for more details), which yields to the following penalized criterion:

$$BIC_i = -2\Pr\{Y|m_i\} \simeq -2\log g_{m_i}(Y, \hat{\psi}_i) + \nu_i \times \log(n),$$

where $\hat{\psi}_i$ is the maximum likelihood estimator of ψ_i. The BIC is used to assess a score to each model m_i and the selected model is such that:

$$\hat{m}_{BIC} = \underset{i}{\text{Argmax}}\, BIC_i.$$

Interestingly regularity conditions for BIC do not hold for mixture models, since the estimates of some mixing proportions can be on the boundary of the parameter space. Nevertheless there is considerable practical support for its use in this context (see Fraley and Raftery (1998) for instance). Other approaches have been considered for Bayesian model selection (see Kass and Raftery (1995) for a complete review on Bayes Factors for instance). Nevertheless the BIC has focused much attention, for its simplicity of implementation and for its statistical properties. Gassiat and Dacunha-Castelle (1997) have shown that the use of BIC leads to a consistent estimator of the number of clusters.

6.3.2 Strategy-oriented criteria

Other criteria have been defined for the special case of mixture models. They can be based on Bayesian methods, on the entropy function of the mixture, or on information theory. The reader is referred to McLachlan and Peel (2000) for a complete review on the construction of those criteria. Empirical comparisons of those criteria have been extensively used to determine the "best" criterion. As noted by Biernacki *et al.* (2000), the use of the BIC can lead to an overestimation of the number of clusters regardless the clusters separation. Moreover estimating the "true" numbers of clusters, which is the objective of the BIC, is not necessarily suitable in a practical context. For these reasons, Biernacki *et al.* (2000) propose a new criterion, the Integrated Classification Criterion (ICL) that considers the clustering objective of mixture models. In this paragraph we present the main steps of the construction of ICL.

In a mixture model context, the integrated likelihood is noted $f(y|m_P)$ for a model m with P clusters. It is calculated such that:

$$f(y|m_P) = \int_{\Psi_P} f(y|m_P, \psi) h(\psi|m_P) d\psi,$$

with

$$f(y|m_P, \psi) = \prod_{t=1}^{n} f(y_t|m_P, \psi),$$

Ψ_P being the parameter space of model m_P, and $h(\psi|m_P)$ a non-informative prior distribution on ψ. Instead of considering the incomplete-data integrated

likelihood for which the BIC approximation is not valid, the authors suggest to use the complete-data integrated likelihood or integrated classification likelihood:

$$f(y, z|m_P) = \int_{\Psi_P} f(y, z|m_P, \psi)h(\psi|m_P)d\psi,$$

with

$$f(y, z|m_P, \psi) = \prod_{t=1}^{n}\prod_{p=1}^{P}\{\pi_p f(y_t; \theta_p)\}^{z_{tp}}.$$

Then the idea is to isolate the contribution of the missing data z by conditioning on z, and it follows that:

$$f(y, z|m_P) = f(y|z, m_P)f(z|m_P),$$

provided that $h(\psi|m_P) = h(\theta|m_P)h(\pi|m_P)$.

The authors emphasize that the BIC approximation is valid for the term $f(y|z, m_P)$, such that:

$$\log f(y|z, m_P) \simeq \max_{\theta} \log f(y|z, m_P, \theta) - \frac{\lambda_P}{2}\log(n),$$

where λ_P is the number of free components in θ. Note that the parameter θ which maximizes $\log f(y|z, m_P, \theta)$ is not the maximum likelihood estimator. Nevertheless, the authors propose to use the maximum likelihood estimator as an approximation.

As for term $f(z|m_P)$ it can be directly calculated using a Dirichlet prior $\mathcal{D}(\delta, \ldots, \delta)$ on proportion parameters. It follows that:

$$f(z|m_P) = \int \pi_1^{n_1} \ldots, \pi_P^{n_P} \frac{\Gamma(P\delta)}{\Gamma(\delta)^P} \mathbb{1}_{\sum_p \pi_p = 1} d\pi,$$

with n_p being the number of data points belonging to cluster p. Then parameter δ is fixed at $1/2$ which corresponds to the Jeffreys non-informative distribution for proportion parameters.

The last steps of the construction of ICL consists in replacing the missing data z which are unknown by the recovered label variables $\tilde{z}$ using a MAP rule. Then an approximation of $f(z|m_P)$ is given when n is large. It follows that:

$$ICL(m_P) = \max_{\psi} \log f(y, \tilde{z}|m_P, \psi) - \frac{\nu_P}{2}\log(n),$$

with ν_P the number of free parameters for model m_P. Therefore the ICL criterion is an "*à la BIC*" approximation of the completed log-likelihood or classification log-likelihood. Since this criterion considers the classification results to score each model it has been shown to lead to a more sensible partitioning of the data, compared with BIC.

The performance of ICL have been tested based on real and simulated data sets. Compared with BIC, ICL tends to select a lower number of clusters which provides good clustering results in real situations, compared with BIC which tends to select a too overly high number of clusters. When the data are simulated, ICL tends to select a lower number of clusters if the groups are not well separated, contrary to BIC which finds the true number of classes. From a theoretical point of view, no result has yet been demonstrated for the properties of ICL.

Chapter 7

A new model for segmentation/clustering problems

As mentionned in the introduction of this part, the construction of our segmentation/clustering model will follow two major steps.

1 - Determination of the parameters affected by changes

The determination of the parameters affected by changes strongly depends on the phenomenon under study. It is a modelling issue rather than a statistical issue. In this part we focus on the case where the data are assumed to be drawn from Gaussian distributions. Nevertheless the segmentation/clustering model could be applied to other types of data. Part V will be devoted to an extension of our model in the case of DNA sequences with the segmentation/clustering model applied to Markov chains.

2 - Estimation strategy

Once the model has been specified, it is crucial to develop an estimation strategy. In our case we choose the maximum likelihood method. The segmentation/clustering model is a "fusion" of a segmentation model and a mixture model, and the likelihood is the quantity that links both models. Moreover, existing algorithms have been proposed to optimize the likelihood function in both cases: dynamic programming for the breakpoints and the EM algorithm for the mixture parameters.

In this Chapter, we propose a hybrid algorithm to estimate the parameters of our model that combines a dynamic programming algorithm and an EM algorithm. We focus on the theoretical developments of such algorithm, and on its convergence properties. The implementation of the algorithm will be developed in Part IV.

As for the problem of model selection which consists in the selection of the number of clusters and the number of segments, it will be studied in the next chapter. Here we focus on the definition of the segmentation/clustering model,

and on its properties when the number of clusters and segments are fixed.

In this Chapter we also propose to compare our model with other models dedicated to segmentation/clustering problems. We focus on two different models for this comparison. A first comparison is made with hidden Markov models which constitute the most widely used strategy to assess segmentation/clustering problems. This methodology has already been applied to array CGH data by Fridlyand *et al.* (2004). Then we study another model which has been developed by Wang *et al.* (2005), and which is called CLAC for "Cluster Along Chromosomes". Modelling strategies will be compared and discussed, and the comparison of their performance will be done using simulation studies in the next Part.

7.1 Definition of a new model

Let $Y = \{Y_1 \ldots, Y_n\}$ be an independent Gaussian process whose mean and variance are affected by $K-1$ abrupt changes at unknown coordinates $T = \{t_1, \ldots, t_{K-1}\}$ with convention $t_0 = 1$ and $t_K = n$. This defines a partition of the data into K vectors of length n_k such that

$$Y = \{Y^1, \ldots, Y^K\} \text{ with } Y^k = \{Y_t, t \in I_k\} \text{ and } I_k = \{t, t \in]t_{k-1}, t_k]\}.$$

Classical segmentation models would consider constant means and variances between two changes, leading to the model:

$$\forall t \in I_k, \;\; Y_t \sim \mathcal{N}(\mu_k, \sigma_k^2).$$

In our case we assume that the mean and variance of Y can only take a limited number of values with:

$$\mu_k \in \{m_1, \ldots, m_P\}, \;\; \sigma_k^2 \in \{s_1^2, \ldots, s_P^2\}.$$

This means that there exists a secondary structure of the process into P clusters in addition to the spatial organization of the data . We assume that the *partitioned* data $\{Y^1, \ldots, Y^K\}$ are structured into P populations with weights π_p (with the constraint $\sum_p \pi_p = 1$). The data Y are viewed as being incomplete since their belonging to the different populations is unknown. Then we introduce a sequence of hidden independent random variables, $Z^k = \{Z_1^k, \ldots, Z_P^k\}$ such that $Z_p^k = 1$ with probability π_p if vector Y^k is drawn from the p^{th} component of the mixture. Thus Z^k is distributed according to a multinomial distribution consisting of one draw on P categories with probabilities $\pi_1, \ldots, \pi_P$, that is

$$Z^k \sim \mathcal{M}(1, \pi_1, \ldots, \pi_P).$$

Mixing proportions $\pi_1, \ldots, \pi_P$ represent the *prior* probability for *vector* Y^k to belong to the p^{th} component, while the *posterior* probability that the vector belongs to the p^{th} component with y^k having been observed is:

$$\tau_p^k = \Pr\left\{Z_p^k = 1 | Y^k = y^k\right\}.$$

Remark that contrary to classical mixture models where indicator variables provide information about the labelling of individual data points (which would be Y_t in our case), our model focuses on the belonging of vectors Y^k to different populations. Moreover, the label variables being independent, it means that a small change in the parmeters of the observed sequence does not systematically lead to a change in the label of the segment.

The complete-data vector is then $X = (Y, Z)$, with Y being the observed-data vector and Z the hidden component indicator vector. We focus on the case where the data are supposed to be drawn from a mixture of Gaussian densities, with parameters m_p, s_p^2. Given the indicator vectors Z^k, the Y^k are supposed to be independent with Gaussian probability distribution:

$$Y^k | Z_p^k = 1 \sim \mathcal{N}(m_p \mathbb{1}_{n_k}, s_p^2 I_{n_k}).$$

More than a conditional independence of segments, we assume that data are conditionally independent *within* a segment:

$$f(y^k; \theta_p) = \prod_{t \in I_k} f(y_t; \theta_p).$$

The parameters of this model are: the breakpoint coordinates $T = \{t_1, \ldots, t_{K-1}\}$ and the parameters of the mixture model $\psi = \{\pi_1, \ldots, \pi_{P-1}, \theta_1, \ldots, \theta_P\}$. The density of vector Y^k can be written as:

$$f(y^k; \psi) = \sum_{p=1}^{P} \pi_p f(y^k; \theta_p).$$

Let us focus on the specific signification of each parameter:

- T, the set of breakpoint parameters: indicates the spatial structure of the data,
- π, the proportions of *segments* belonging to each group. These parameters concern segments which constitute the statistical units to be clustered.
- θ, the level and variability of the signal for each group. These parameters provide information regarding the *individual data points*.

Defining statistical units of the model is not straightforward in this formulation. We observe n data points that are segmented into K segments. The segments being defined, they are clustered into P groups. The statistical units of the mixture model are *segments* which constitute random individuals. Nevertheless, we are also interested in the behavior of individual data points within segments which provides information regarding the parameters of each group. Therefore the complete-data vector should be restated as:

$$X^k = (Y^k, Z^k) = (Y_{t_{k-1}+1} \ldots, Y_{t_k}, Z^k).$$

7.2 Estimating model parameters via maximum likelihood when P and K are fixed

Once the model has been defined, the next step is to estimate the parameters for a fixed number of segments and clusters. We choose the maximum likelihood method since the likelihood of the model is the central link between the segmentation model and the mixture model which are combined.

The likelihood of this model is a function of two sets of parameters: the set of breakpoint coordinates T, and the set of mixture parameters ψ. It also depends on both the number of segments and the number of clusters that are fixed in this section. Since we assume the independence of the data between segments, the log-likelihood is written as:

$$\log \mathcal{L}_{KP}(T, \psi) = \sum_{k=1}^{K} \log f(y^k; \psi) = \sum_{k=1}^{K} \log \left\{ \sum_{p=1}^{P} \pi_p f(y^k; \theta_p) \right\},$$

and we aim at determining:

$$(\hat{T}, \hat{\psi}) = \underset{T,\psi}{\text{Argmax}} \left\{ \log \mathcal{L}_{KP}(T, \psi) \right\}.$$

In Chapters 4 and 6 we reviewed existing algorithms for the estimation of the breakpoint instants in the segmentation context, and for the estimation of mixture model parameters with the EM algorithm. The principle of our method is to build a hybrid algorithm based on dynamic programming and EM. Our strategy is iterative: when the breakpoint coordinates T are known, the EM algorithm is used to estimate the mixture parameters ψ. Once ψ has been estimated, the breakpoint coordinates are computed using dynamic programming.

7.2.1 Estimating mixture model parameters when breakpoint coordinates are known

When breakpoint coordinates are known, we have a partition of the data into K vectors $\{Y^1, \ldots, Y^K\}$. The purpose is to maximize the log-likelihood of the model according to ψ, and the maximum likelihood estimator of this parameter is

$$\hat{\psi}(T) = \underset{\psi}{\text{Argmax}} \left\{ \log \mathcal{L}_{KP}(T, \psi) \right\}.$$

Since the breakpoints are fixed in this step, the mixture parameters depend on T. In the following, this mention will be omitted. Nevertheless, the impact of the breakpoint instants T on parameter estimate $\hat{\psi}(T)$ will be studied later.

The optimization of this log-likelihood can be handled using the EM algorithm in the complete data framework. The introduction of hidden label variables Z^k leads to the definition of a complete-data log-likelihood such that:

$$\log \mathcal{L}^c_{KP}(T, \psi) = \sum_{k=1}^{K} \sum_{p=1}^{P} z_p^k \log \left\{ \pi_p f(y^k; \theta_p) \right\}.$$

The principle of the EM algorithm is to iteratively maximize the conditional expectation of the complete-data likelihood in order to maximize the incomplete-data likelihood, as shown in Chapter 6.

E-step

In the Expectation step, the conditional expectation of the complete-data log-likelihood is computed, given the observed data y, and using the current fit $\psi^{(\ell)}$ for ψ at the ℓ^{th} iteration. This quantity can be written as

$$Q_{KP}(\psi|\psi^{(\ell)};T) = \mathbb{E}_{\psi^{(\ell)}}\left\{\log \mathcal{L}^c_{KP}(\psi;T)|y\right\} = \sum_{k=1}^{K}\sum_{p=1}^{P} \tau_p^{k(\ell)} \log\left\{\pi_p f(y^k;\theta_p)\right\},$$

with

$$\tau_p^{k(\ell)} = \frac{\pi_p^{(\ell)} f(y^k;\theta_p^{(\ell)})}{\sum_{q=1}^{P} \pi_q^{(\ell)} f(y^k;\theta_q^{(\ell)})}.$$

M-step

The M-step at the $(\ell+1)^{th}$ iteration requires the global maximization of $Q_{KP}(\psi|\psi^{(\ell)};T)$ with respect to ψ to give the updated estimate $\psi^{(\ell+1)}$. In the Gaussian case, the updated estimators are :

$$\left\{\begin{array}{lcl} \hat{m}_p^{(\ell+1)} & = & \sum_k \tau_p^{k(\ell)} \sum_{t\in I_k} y_t / \sum_k n_k \tau_p^{k(\ell)}, \\ \hat{s}_p^{2(\ell+1)} & = & \sum_k \tau_p^{k(\ell)} \sum_{t\in I_k} (y_t - \hat{m}_p^{(\ell+1)})^2 / \sum_k n_k \tau_p^{k(\ell)}, \\ \hat{\pi}_p^{(\ell+1)} & = & \sum_k \tau_p^{k(\ell)} / K. \end{array}\right. \tag{7.1}$$

7.2.2 Estimating breakpoint coordinates when the mixture parameters are known

When the number of segments K, the number of clusters P and the parameters of the mixture are known, the objective is to find the best K-dimensional partition of the data according to the log-likelihood $\log \mathcal{L}_{KP}(T,\psi)$, with ψ being fixed. We aim at finding:

$$\hat{T}(\psi) = \underset{T}{\text{Argmax}} \left\{\log \mathcal{L}_{KP}(T,\psi)\right\}.$$

Since the incomplete-data log-likelihood is still additive according to the number of segments a dynamic programming approach can be used to compute the breakpoint instants, as it is the case for classical segmentation models (Chapter 4). The main difference lies in the estimation of the parameters of segments. In classical segmentation models, the estimation of the breakpoint instants and of the segments parameters is done simultaneously. In the segmentation/clustering context, the estimation of ψ is postponed and is done with the EM algorithm.

Let $C_{k+1,P}(i,j;\psi)$ be the log-likelihood obtained by the best partition of the data $Y^{ij} = \{Y_{i+1},...,Y_j\}$ into $k+1$ segments when the mixture parameters ψ are

known. The algorithm starts as follows: for $k = 0$ and for $(i, j) \in [1, n]^2$, with $i < j$, calculate:

$$C_{1,P}(i, j; \psi) = \log \left\{ \sum_{p=1}^{P} \pi_p f(y^{ij}; \theta_p) \right\} = \log \left\{ \sum_{p=1}^{P} \pi_p \prod_{t=i+1}^{j} f(y_t; \theta_p) \right\}.$$

$C_{1,P}(i, j; \psi)$ represents the local log-likelihood for segment Y^{ij}. Then the algorithm is run as follows:

$$\forall k \in [1, K_{max}] \quad C_{k+1,P}(1, j; \psi) = \max_h \{C_{k,P}(1, h; \psi) + C_{1,P}(h + 1, j; \psi)\}$$

7.2.3 Monotonicity property of the hybrid algorithm

Proposition

For a fixed number of clusters and segments the hybrid algorithm generates a sequence $\left(T^{(h)}, \psi^{(h)}\right)_{h \geq 0}$ *that increases the incomplete-data log-likelihood such that:*

$$\log \mathcal{L}_{KP}\left(T^{(h+1)}, \psi^{(h+1)}\right) \geq \log \mathcal{L}_{KP}\left(T^{(h)}, \psi^{(h)}\right).$$

Proof

The proof is based on the properties of dynamic programming (Chapter 4) and of the EM algorithm (Chapter 6). These two convergence properties can be used since both algorithms are linked through the log-likelihood they alternatively optimize: the incomplete-data log-likelihood of the mixture of segments.

Thanks to the Bellman optimality principle (Bellman and Dreyfus (1962) and Chapter 5), dynamic programming globally optimizes the likelihood with respect to T. At iteration (h) we have:

$$\log \mathcal{L}_{KP}\left(T^{(h+1)}; \psi^{(h)}\right) \geq \log \mathcal{L}_{KP}\left(T^{(h)}, \psi^{(h)}\right).$$

On the other hand, the key monotonicity property of the EM algorithm is the increase of the incomplete-data log-likelihood at each step (Dempster *et al.* (1977)):

$$\log \mathcal{L}_{KP}\left(T^{(h)}, \psi^{(h+1)}\right) \geq \log \mathcal{L}_{KP}\left(T^{(h)}, \psi^{(h)}\right).$$

Put together, our algorithm generates a sequence $\left(T^{(h)}, \psi^{(h)}\right)_{h \geq 0}$ that increases the incomplete-data log-likelihood such that:

$$\log \mathcal{L}_{KP}\left(T^{(h+1)}, \psi^{(h+1)}\right) \geq \log \mathcal{L}_{KP}\left(T^{(h)}, \psi^{(h)}\right).$$

A schematic representation of the hybrid algorithm is given in Figure 7.1. Note that two levels of indices are required for this algorithm: h describes iterations between the EM algorithm and dynamic programming, whereas ℓ is used for the iterations within the EM algorithm.

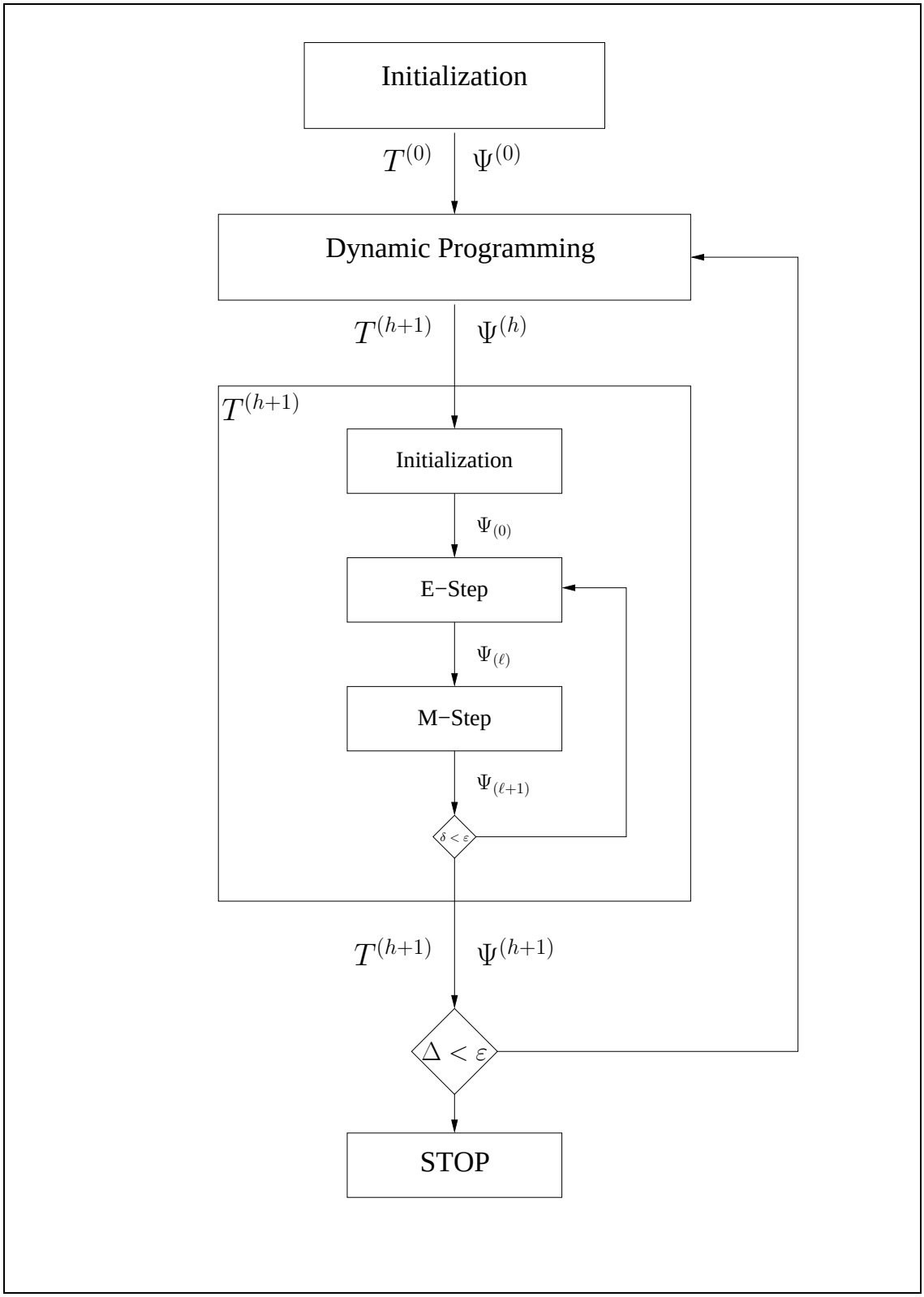


Figure 7.1: Diagram of the hybrid algorithm.

7.3 Assessing variance estimates

The theoretical derivation of the information matrix for mixture model parameters has been explained in Chapter 6. In this section, we aim to provide the estimation of the information matrix for the mixture parameters of the segmentation/clustering model when the number of clusters and the number of segments are fixed. Briefly, let us recall some important notations. The objective is to calculate the observed information matrix:

$$I(\psi;Y) = -\frac{\partial^2}{\partial\psi\partial\psi^T}\log\mathcal{L}_{KP}(Y;\psi,T)$$

In the following, we will use the score function, noted S, such that:

$$\begin{cases} S(\psi;Y) = \frac{\partial}{\partial\psi}\log\mathcal{L}_{KP}(Y;\psi,T), \\ S^c(\psi;X) = \frac{\partial}{\partial\psi}\log\mathcal{L}^c_{KP}(X;\psi,T). \end{cases}$$

The differentiation of the incomplete-data log-likelihood being difficult, we use the complete-data likelihood, with the relationship given by Louis (1982):

$$\begin{aligned} I(\hat{\psi};Y) &= \mathbb{E}_{X|Y}\left\{I^c(\psi;X)\right\}\big|_{\psi=\hat{\psi}} - \mathbb{E}_{X|Y}\left\{I^m(\psi;X)\right\}\big|_{\psi=\hat{\psi}} \\ &= \mathbb{E}_{X|Y}\left\{I^c(\psi;X)\right\}\big|_{\psi=\hat{\psi}} - \mathbb{C}ov_{X|Y}\left\{S^c(\psi;X)\right\}\big|_{\psi=\hat{\psi}}, \end{aligned} \tag{7.2}$$

with $I^m(\psi;X)$ being the missing data information matrix. Since the complete calculation of this information matrix is long, it is fully detailed in Appendix section 7.7. In this section, we focus on the particular structure of the information matrix in the segmentation/clustering context. For this purpose, we use the complete data information matrix as an example. Similar comments can be made on the missing information matrix.

In the Appendix section 7.7 we show that the complete-data information matrix $I^c(\psi;X)$ is block diagonal with terms:

$$\mathbb{E}_{X|Y}\left\{I^c(\pi;X)\right\}\big|_{\pi=\hat{\pi}} = K \times \begin{bmatrix} \frac{1}{\hat{\pi}_1}+\frac{1}{\hat{\pi}_P} & \dots & \frac{1}{\hat{\pi}_P} \\ \vdots & \ddots & \vdots \\ \frac{1}{\hat{\pi}_P} & \dots & \frac{1}{\hat{\pi}_{P-1}}+\frac{1}{\hat{\pi}_P} \end{bmatrix},$$

$$\mathbb{E}_{X|Y}\left\{I^c(m;X)\right\}\big|_{\theta=\hat{\theta}} = \begin{bmatrix} \frac{n_p}{\hat{s}_1^2} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{n_P}{\hat{s}_P^2} \end{bmatrix},$$

$$\mathbb{E}_{X|Y}\left\{I^c(s^2;X)\right\}\big|_{\theta=\hat{\theta}} = \begin{bmatrix} \frac{n_p}{2\hat{s}_1^4} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{n_P}{2\hat{s}_P^4} \end{bmatrix},$$

and $n_p = \sum_k n_k \tau_p^k$ represents the number of data points within every segment in cluster p. The information matrix concerning proportion parameters can be written as:

$$\mathbb{E}_{X|Y}\left\{I^c(\pi;X)\right\}\big|_{\pi=\hat{\pi}} = K \times \mathbb{E}_{X|Y}\left\{i^c(\pi;X)\right\}\big|_{\pi=\hat{\pi}},$$

where $i^c(\pi;X)$ would represent the information provided by one statistical unit. Consequently, the number of data points has no influence on the precision of the proportion estimators which only depend on the number of segments. On the contrary, the precision of the external parameters (mean and variance) will increase with the number of data points, since

$$\mathbb{E}_{X|Y}\left\{I^c(m;X)\right\}\big|_{\theta=\hat{\theta}} \simeq n \times \mathbb{E}_{X|Y}\left\{i^c(m;X)\right\}\big|_{\theta=\hat{\theta}}.$$

This particular behavior of the parameters' estimators is linked to the definition of the statistical units of the model: hidden variables $\{Z^1,\ldots,Z^K\}$ only concern K *segments*, whereas observed variables $\{Y_1,\ldots,Y_n\}$ concern n data points.

7.4 Behavior of the model when K and P are fixed

As previously described the objective of the segmentation/clustering problem is to define segments and to cluster them into a finite number of groups. In a first step, we focus on the case where K and P are fixed. In this case the model can be viewed as a mixture model of *segments*. The objective is then to cluster vectors $\{Y^1,\ldots,Y^K\}$ into P groups. The originality of this model is that the "data" which are considered by the mixture are segments of different sizes, these sizes being defined by the breakpoint parameters T when the number of segments is fixed, since the length of segment k is $n_k = t_k - t_{k-1}$. As segments show different sizes, this heterogeneity is likely to have consequences on the behavior of the model.

7.4.1 Impact of segments' size on *posterior* probabilities

As noted previously, $f(y^k;\theta_p)$ is the density of a Gaussian vector of length n_k. In the case of a homoscedastic model its expression is:

$$f(y^k;\theta_p) = \exp\left\{-\frac{n_k}{2}\left(\log(2\pi s^2) + \frac{1}{s^2}\left[d_{kp}^2 + w_k^2\right]\right)\right\},$$

with:

$$\begin{cases} d_{kp}^2 &= (\bar{y}_k - m_p)^2 \\ w_k^2 &= (\bar{y_k^2} - \bar{y}_k^2) \end{cases} \qquad \begin{cases} \bar{y}_k &= \frac{1}{n_k}\sum_{t\in I_k} y_t, \\ \bar{y_k^2} &= \frac{1}{n_k}\sum_{t\in I_k} y_t^2. \end{cases}$$

d_{kp}^2 represents the distance of the average of vector Y^k to the mean of cluster p, whereas w_k^2 quantifies the intra-vector variability. The *posterior* probability of belonging to population p is:

$$\tau_p^k = \frac{\pi_p \exp\left\{-\frac{n_k}{2s^2} d_{kp}^2\right\}}{\sum_{\ell=1}^{P} \pi_\ell \exp\left\{-\frac{n_k}{2s^2} d_{k\ell}^2\right\}}.$$

Proposition

Noting p_0 the cluster such that:

$$d_{k0}^2 = \inf_p \left\{ d_{kp}^2 \right\},$$

then:

$$\begin{array}{rcl|rcl} \lim\limits_{n_k \to \infty} \tau_{p_0}^k & = & 1 & \lim\limits_{n_k \to \infty} \tau_p^k & = & 0 \\ \lim\limits_{n_k \to 0} \tau_{p_0}^k & = & \pi_{p_0} & \lim\limits_{n_k \to 0} \tau_p^k & = & \pi_p. \end{array}$$

Proof

For the proof, we detail the calculus of the *posterior* probabilites such that:

$$\begin{aligned} \tau_{p_0}^k &= \frac{1}{1 + \sum_{\ell \neq p_0} \frac{\pi_\ell}{\pi_{p_0}} \exp\left\{ -\frac{n_k}{2s^2} (d_{k\ell}^2 - d_{k0}^2) \right\}} \\ \tau_p^k &= \frac{\pi_p \exp\left\{ -\frac{n_k}{2s^2} (d_{kp}^2 - d_{k0}^2 \right\}}{\pi_{p_0} + \sum_{\ell \neq p_0} \pi_\ell \exp\left\{ -\frac{n_k}{2s^2} (d_{k\ell}^2 - d_{k0}^2) \right\}} \end{aligned}$$

Together, these equations show that:

$$\begin{array}{rcl|rcl} \lim\limits_{n_k \to \infty} \tau_{p_0}^k & = & 1 & \lim\limits_{n_k \to \infty} \tau_p^k & = & 0 \\ \lim\limits_{n_k \to 0} \tau_{p_0}^k & = & \pi_{p_0} & \lim\limits_{n_k \to 0} \tau_p^k & = & \pi_p \end{array}$$

This means that vector Y^k will be assigned (with probability one) to the closest population as its length increases. On the contrary, if the length of the vector is small, the observation of its realization y^k will provide little information regarding its membership, and the *posterior* probability will tend to the *prior* probability of membership to one group.

7.4.2 Impact of segments' size on mixture parameters estimates

The formulation of the mixture parameter estimates is given in Equation (7.1). Let us focus on the estimation of the groups' weight:

$$\hat{\pi}_p = \frac{\sum_k \hat{\tau}_p^k}{K}. \tag{7.3}$$

Proportion π_p will be interpreted as the proportion of *segments* which belong to cluster p. Since we have shown that τ_p^k tends to one as the size of segment k increases, the estimator of π_p will be also sensitive to the size of the segments. Equation (7.3) indicates that segments with important size will highly contribute to the estimation of the groups proportions, and may create a group of their own.

As for the estimation of internal parameters it can be seen from Equation (7.1), that vectors of important size will influence the estimation of the mean and variance of each cluster, since

$$\hat{m}_p = \frac{\sum_k n_k \tau_p^k \bar{y}_k}{\sum_k n_k \tau_p^k}.$$

This means that the mean m_p of the p^{th} component of the mixture will be representative of the longest vectors Y^k which are the closest to population p.

7.5 Comparison with other methods for segmentation/clustering problems

Now that we defined our model for segmentation/clustering problems, our purpose is to compare it with existing models which deal with similar issues. Hidden Markov models constitute the most widely used method to assess segmentation/clustering questions. This is why we propose to explore the differences that exist between our model and HMMs, from a modelling point of view.

Hidden Markov models have been applied to array CGH data analysis by Fridlyand *et al.* (2004). However another strategy has been proposed to analyse these data. This method is called CLAC for Clustering Along Chromosomes (Wang *et al.* (2005)). We also compare our model to this new method whose objectives are similar.

7.5.1 Comparison with hidden Markov models

In this section, we aim at comparing the different modelling strategies between HMMs and the segmentation/clustering model we propose. Note that we do not provide an extensive review of hidden Markov models. The reader is referred to Rabiner (1989) and Ephraim (2002) for this purpose.

Brief presentation of hidden Markov models

In order to draw analogies between HMMs and the segmentation/clustering model, we will use similar terminologies. When using a hidden Markov model we consider a sequence of hidden variables $Z_1^n = \{Z_1, \ldots, Z_n\}$ taking values in a state space $\mathcal{Z} = \{1, \ldots, P\}$, where P is the number of clusters. This sequence indicates the labelling of each individual data point to P clusters. The emission of the observed sequence $y_1^n = \{y_1, \ldots, y_n\}$ is modelled through the associated hidden states, meaning that the emission of Y_t is modelled conditionally to Z_t. In the Gaussian case, it follows that

$$Y_t | Z_t = p \sim \mathcal{N}(m_p, s_p^2).$$

If variables Z_ts were independent, we would be in the case of a mixture model which does not consider any spatial dependency for clustering. In order to introduce this spatial dependency among hidden variables, the sequence Z_1^n is supposed

to be drawn from a Markov chain with transition matrix $\phi = \{\phi(p,\ell)\}_{P\times P}$ such that:

$$\phi(p,\ell) = \Pr\{Z_t = \ell | Z_{t-1} = p\}.$$

In the following, we will note $\pi = \{\pi_p\}_P$ the vector representing the initial distributions, *i.e.* the probability that the initial state is p :

$$\pi_p = \Pr\{Z_1 = p\}.$$

Reconstruction of the hidden sequence

When the parameters of the model are known, we aim at reconstructing the hidden sequence Z_1^n which is associated to the observed data. This sequence is used to give a label to each individual data point Y_t. This reconstruction is done using the Viterbi or the forward-backward algorithm. The Viterbi algorithm aims at finding the most probable hidden sequence $\tilde{z}_1^n$ given the observed sequence and parameters (ϕ, π) such that:

$$\tilde{z}_1^n = \underset{z_1^n}{\text{Argmax}} \left\{ \Pr_{(\phi,\pi)} \{Z_1^n = z_1^n | Y_1^n = y_1^n\} \right\},$$

whereas the forward-backward algorithm aims at calculating at every position t the probability of each couple of states $\{Z_t = p, Z_{t+1} = \ell\}$ and of each state $\{Z_t = p\}$ conditionally to the observed sequence:

$$\Pr_{(\phi,\pi)} \{Z_t = p, Z_{t+1} = \ell | Y_1^n = y_1^n\} \text{ and } \Pr_{(\phi,\pi)} \{Z_t = p | Y_1^n = y_1^n\}, (p,\ell) \in \mathcal{Z}^2$$

Note that both methods use a Maximum A Posteriori (MAP) rule to reconstruct the hidden states.

Status of the hidden sequence

Since the segmentation/clustering model and HMMs can be considered as models with hidden structure, we propose to compare the signification of both hidden variables. While applied to array CGH data analysis, both hidden structures aim at describing the non-observed gene copy-number which can be associated to each clone. In the case of HMMs, this sequence is supposed to follow a Markovian distribution, meaning that the gene copy-number at one point depends on the gene copy-number of neighboring clones. This Markovian property allows the sequence of hidden variables to show homogeneous labels such that:

$$\Pr\{Z_{t_{k-1}+2}^{t_k} = (p,\ldots,p), Z_{t_k+1} \neq p | Z_{t_{k-1}+1} = p, Z_{t_{k-1}} \neq p\} = \phi(p,p)^{n_k}(1-\phi(p,p)),$$

with $n_k = t_k - t_{k-1}$ being the length of segment k for which the label is p. In order to draw analogies with the segmentation/clustering model, we will call breakpoint instant, the position t_k for which the label changes. This allows us to define I_k, the k^{th} segment for which the label variables $\tilde{z}_t$ show the same label:

$$I_k =]t_{k-1}, t_k] \ / \ \forall t \in I_k, \ \ \tilde{z}_t = \tilde{z}^k, \tag{7.4}$$

with $\tilde{z}^k$ the label of segment k. One property of hidden Markov models is that the length of segments follows a geometric distribution with mean $1/(1-\phi(p,p))$. This means that modelling the spatial dependency with a Markovian property implicitly models the distribution of the length of segments.

In the case of the segmentation/clustering model we propose, the label variables concern segments, and not data points. Segments are interpreted as genomic regions which share the same gene-copy number. In our case, the hidden variables are supposed to be independent, meaning that we do not model the probability to pass from one biological state to another. Nevertheless, there exists a spatial dependency among gene copy numbers, since one point whose neighbors are deleted may be deleted as well. Hidden Markov models use the Markovian property to model this biological hypothesis, whereas our model uses the segmented nature of the observed sequence to do so. This means that in the case of segmentation/clustering, the spatial dependency is modelled through the observed sequence, with the breakpoints which are parameters to estimate.

One major difference beween both modelling strategies lies in the variables which are affected by changes. Indeed, the segmentation/clustering model supposes that the *observed* data are affected by changes (through their distribution parameters), whereas HMMs model changes that affect the *hidden* sequence. Therefore changes and segments have different significations in both models. In the segmentation/clustering context, a change in the parameters of the distribution of the observed sequence is not necessarily linked to a change in the label of the segment, since there exists no constraint for Z^k to be different from Z^{k+1}. Moreover breakpoint instants do not have the same status, since they are parameters which are optimized by maximum-likelihood in the case of our model, whereas they are recovered in the case of HMMs.

7.5.2 Comparison with the CLAC approach

A different modelling strategy has been considered by Wang *et al.* (2005), who propose a new method for calling gains or losses in array CGH data. This method is called $CLAC$ for Clustering Along Chromosomes. In the following, we propose to present briefly this method and to compare it with our model.

The purpose of CLAC is to build a hierarchical cluster tree along each chromosome arm, such that gain/loss regions are separated into different branches. Since the order of the genes is fixed along the chromosome, the order of the leaves of the tree is fixed as well. Consequently, only adjacent clusters are joined together when the tree is generated from the bottom-up.

In order to measure the similarity between neighboring clones, the authors introduce a statistic called relative difference and defined such that:

$$rd(y_t, y_{t+1}) = \frac{|y_t - y_{t+1}|}{|y_t| + |y_{t+1}| + |y_{t+1} + y_t|}.$$

The distance between two contiguous clusters $C_i = \{t_1, \ldots, t_k\}$ and $C_j = \{\ell_1, \ldots, \ell_k\}$ is:

$$rd(C_i, C_j) = rd(y_{t_k}, y_{\ell_1}),$$

where $\ell_1 = t_k + 1$ since C_i and C_j are neighboring clusters.

Once the tree has been constructed the question is to select interesting clusters. To do so, the authors propose to study three characteristics of each cluster which are: the relative difference of the node in the tree, the number of clones in the subtree and the mean value of the leaves of the subtree. Interesting clusters are chosen using cut-off values from these criteria.

A major difference between the CLAC approach and the segmentation/clustering model is that the first one is not based on a statistical model. This allows the authors to define a very simple clustering procedure, based on the classical hierarchical clustering method. This is why modelling strategies are difficult to compare in this case. Nevertheless, it is clear that this approach only considers neighboring points for clustering, whereas there may be non-contiguous points belonging to the same cluster. On the contrary, the segmentation/clustering method considers every data points to estimate the parameters of each group. The other major difference is that the CLAC approach does not consider any breakpoint in the observed data. The spatial information is used to cluster neighboring points only.

One major criticism that can be made to the CLAC approach is its lack of statistical basis for model selection. This procedure is done using empirical characteristics which are supposed to define homogeneous clusters. Therefore the CLAC method is very dependent on the type of data under study, which is not the case for HMMs and for our model. In the next chapter, we will construct a model selection heuristic to select the number of clusters and segments. This procedure is based on the behavior of the likelihood of the model, and can be applied to other types of data.

7.6 Conclusion

In this chapter we constructed a new model for segmentation/clustering purposes. Compared with other modelling approaches this model is new since it combines a segmentation model and a mixture model. This combination is thought to handle the specificity of the data, which is a spatial dependency and a structure into groups. We proposed a hybrid algorithm to estimate the parameters of the model. The practical implementation of this algorithm will be studied in the next part. It will require an appropriate initialization step, as every iterative algorithm, and we will propose a method to stabilize the method when it faces local maxima.

The next step is the selection of the number of segments and clusters, which is the purpose of the next chapter.

7.7 Appendix

In this section, our objective is to calculate the empirical Fisher information matrix with the relationship provided by Louis (1982):

$$\begin{aligned} I(\hat{\psi};Y) &= \mathbb{E}_{X|Y}\left\{I^c(\psi;X)\right\}\big|_{\psi=\hat{\psi}} - \mathbb{E}_{X|Y}\left\{I^m(\psi;X)\right\}\big|_{\psi=\hat{\psi}} \\ &= \mathbb{E}_{X|Y}\left\{I^c(\psi;X)\right\}\big|_{\psi=\hat{\psi}} - \mathbb{C}ov_{X|Y}\left\{S^c(\psi;X)\right\}\big|_{\psi=\hat{\psi}}, \end{aligned} \tag{7.5}$$

where $I^c(\psi;X)$ denotes the complete-data information matrix and $I^m(\psi;X)$ the missing data information matrix.

7.7.1 Complete-data information matrix for mixture parameters

First of all, we show that the complete data information matrix is bloc-diagonal. In the complete-data framework, we can use the following formula $g(Y,Z;\psi) = h(Z;\pi) \times f(Y|Z;\theta)$, with h being the marginal density of the hidden variables. Then we have:

$$\begin{aligned} S^c(\psi;X) &= \frac{\partial}{\partial\psi}\log g(Y,Z;\psi) \\ &= \frac{\partial}{\partial\psi}\log h(Z;\pi) + \frac{\partial}{\partial\psi}\log f(Y|Z;\theta). \end{aligned}$$

Since

$$\begin{aligned} \frac{\partial}{\partial\theta}\log h(Z;\pi) &= 0, \\ \frac{\partial}{\partial\pi}\log f(Y|Z;\theta) &= 0, \end{aligned}$$

It is clear that:

$$I^c(\psi;X) = \begin{bmatrix} I^c(\pi;X) & 0 \\ 0 & I^c(\theta;X) \end{bmatrix}.$$

In order to derive the calculus of the complete-data information matrix, we need to caculate the complete-data score $S^c(\psi;X)$ such that:

$$\begin{aligned} \frac{\partial}{\partial\pi_p}\mathcal{L}^c_{KP}(X;\psi,T) &= \sum_k \frac{Z_{kp}}{\pi_p} - \frac{Z_{kP}}{\pi_P}, \\ \frac{\partial}{\partial m_p}\mathcal{L}^c_{KP}(X;\psi,T) &= \sum_k Z_{kp}\frac{\sum_{t\in I_k}(y_t - m_p)}{s_p^2}, \\ \frac{\partial}{\partial s_p^2}\mathcal{L}^c_{KP}(X;\psi,T) &= \sum_k \frac{Z_{kp}}{2}\left(\frac{-n_k}{s_p^2} + \frac{\sum_{t\in I_k}(y_t - m_p)^2}{s_p^4}\right). \end{aligned} \tag{7.6}$$

Let us focus on the proportion parameters in a first step. Differentiating twice the complete-data log-likelihood gives:

$$\mathbb{E}_{X|Y}\left\{I^c(\pi;X)\right\} = \sum_k \begin{bmatrix} \frac{\tau_1^k}{\pi_1^2}+\frac{\tau_P^k}{\pi_P^2} & \cdots & \frac{\tau_P^k}{\pi_P^2} \\ \vdots & \ddots & \vdots \\ \frac{\tau_P^k}{\pi_P^2} & \cdots & \frac{\tau_{P-1}^k}{\pi_{P-1}^2}+\frac{\tau_P^k}{\pi_P^2} \end{bmatrix}.$$

Since

$$\hat{\pi}_p = \frac{1}{K}\sum_k \tau_p^k,$$

we finally obtain:

$$\mathbb{E}_{X|Y}\left\{I^c(\pi;X)\right\}\big|_{\pi=\hat{\pi}} = K \times \begin{bmatrix} \frac{1}{\hat{\pi}_1}+\frac{1}{\hat{\pi}_P} & \cdots & \frac{1}{\hat{\pi}_P} \\ \vdots & \ddots & \vdots \\ \frac{1}{\hat{\pi}_P} & \cdots & \frac{1}{\hat{\pi}_{P-1}}+\frac{1}{\hat{\pi}_P} \end{bmatrix}.$$

For the external parameters $\theta = (m, s^2)$, we can write:

$$I^c(\theta;X) = \begin{bmatrix} I^c(m;X) & I^c(m,s^2;X) \\ I^c(m,s^2;X) & I^c(s^2;X) \end{bmatrix},$$

For the mean parameters, $I^c(m;X)$ is diagonal since

$$\frac{\partial^2}{\partial m_p \partial m_\ell}\log \mathcal{L}_{KP}^c(X;\psi,T) = 0.$$

It follows that:

$$\mathbb{E}_{X|Y}\left\{I^c(m;X)\right\}\big|_{\theta=\hat{\theta}} = \mathrm{diag}\left[\frac{\sum_k n_k\tau_{k1}}{\hat{s}_1^2},\ldots,\frac{\sum_k n_k\tau_P^k}{\hat{s}_P^2}\right].$$

Then we have:

$$\begin{aligned} \mathbb{E}_{X|Y}\left\{I^c(m,s^2;X)\right\}\big|_{\theta=\hat{\theta}} &= 0, \\ \mathbb{E}_{X|Y}\left\{I^c(s^2;X)\right\}\big|_{\theta=\hat{\theta}} &= \frac{1}{2}\mathrm{diag}\left[\frac{\sum_k n_k\tau_{k1}}{\hat{s}_1^4},\ldots,\frac{\sum_k n_k\tau_P^k}{\hat{s}_P^4}\right]. \end{aligned}$$

If we note $n_p = \sum_k n_k\tau_p^k$ the number of data points within cluster p, the complete-data information matrix has the following expression:

$$\mathbb{E}_{X|Y}\left\{I^c(\theta;X)\right\}\big|_{\theta=\hat{\theta}} = \mathrm{diag}\left[\frac{n_1}{\hat{s}_1^2},\ldots,\frac{n_P}{\hat{s}_P^2};\frac{n_1}{2\hat{s}_1^4},\ldots,\frac{n_P}{2\hat{s}_P^4}\right].$$

7.7.2 Missing-data information matrix

In this section, our aim is to calculate:

$$\mathbb{E}_{X|Y}\left\{I^m(\psi;X)\right\}\big|_{\psi=\hat{\psi}} = \mathbb{C}ov_{X|Y}\left\{S^c(\psi;X)\right\}\big|_{\psi=\hat{\psi}}.$$

For this purpose we need to caculate the complete-data score $S^c(\psi;X)$ as in Equation (7.7). In the following, the calculus is decomposed such as:

$$I^m(\psi;X) = \begin{bmatrix} I^m(\pi;X) & I^m(\pi,m;X) & I^m(\pi,s^2;X) \\ & I^m(m;X) & I^m(m,s^2;X) \\ & & I^m(s^2;X) \end{bmatrix}.$$

Missing-data information matrix for proportions parameters:

The calculus of $I^m(\pi;X)$ can be done separately for diagonal and non-diagonal terms.

$$\begin{aligned} \text{diag}\ \mathbb{E}_{X|Y}\left\{I^m(\pi;X)\right\} &= \mathbb{V}\left\{\sum_k \frac{Z_p^k}{\pi_p} - \frac{Z_P^k}{\pi_P}|Y\right\}, \\ \text{triu}\ \mathbb{E}_{X|Y}\left\{I^m(\pi;X)\right\} &= \mathbb{C}ov\left\{\sum_k \frac{Z_p^k}{\pi_p} - \frac{Z_P^k}{\pi_P}; \sum_k \frac{Z_\ell^k}{\pi_\ell} - \frac{Z_P^k}{\pi_P}|Y\right\}. \end{aligned}$$

$$\begin{aligned} \text{diag}\ \mathbb{E}_{X|Y}\left\{I^m(\pi;X)\right\} &= \sum_k \frac{\tau_p^k(1-\tau_p^k)}{\pi_p^2} + \frac{\tau_P^k(1-\tau_P^k)}{\pi_P^2} + 2\frac{\tau_p^k\tau_P^k}{\pi_p\pi_P}, \\ \text{triu}\ \mathbb{E}_{X|Y}\left\{I^m(\pi;X)\right\} &= \sum_k \frac{\tau_P^k(1-\tau_P^k)}{\pi_P^2} - \frac{\tau_p^k\tau_\ell^k}{\pi_p\pi_\ell} + \frac{\tau_p^k\tau_P^k}{\pi_p\pi_P} + \frac{\tau_\ell^k\tau_P^k}{\pi_\ell\pi_P}. \end{aligned}$$

Missing-data information matrix for means parameters

In order to simplify notations for the calculus of $I^m(m;X)$, we use notation $d_{kp} = (\bar{y}_k - m_p)$. The missing-data information matrix for the mean parameters is then:

$$\begin{aligned} \text{diag}\ \mathbb{E}_{X|Y}\left\{I^m(m;X)\right\} &= \mathbb{V}\left\{\sum_k \frac{Z_p^k}{s_p^2}\sum_{t\in I_k}(y_t - m_p)|Y\right\}, \\ \text{triu}\ \mathbb{E}_{X|Y}\left\{I^m(m;X)\right\} &= \mathbb{C}ov\left\{\sum_k \frac{Z_p^k}{s_p^2}\sum_{t\in I_k}(y_t - m_p); \sum_k \frac{Z_\ell^k}{s_\ell^2}\sum_{t\in I_k}(y_t - m_\ell)|Y\right\}. \end{aligned}$$

$$\begin{aligned} \text{diag}\ \mathbb{E}_{X|Y}\left\{I^m(m;X)\right\} &= \sum_k \tau_p^k(1-\tau_p^k) \times n_k^2 \times \frac{d_{kp}^2}{s_p^4}, \\ \text{triu}\ \mathbb{E}_{X|Y}\left\{I^m(m;X)\right\} &= -\sum_k \tau_p^k\tau_\ell^k \times n_k^2 \times \frac{d_{kp}d_{k\ell}}{s_p^2 s_\ell^2}. \end{aligned}$$

Missing-data information matrix for variances parameters

$$\begin{aligned}
\text{diag } \mathbb{E}_{X|Y}\left\{I^m(s^2;X)\right\} &= \mathbb{V}\left\{\sum_k \frac{Z_p^k}{2}\left(\frac{-n_k}{s_p^2}+\frac{\sum_{t\in I_k}(y_t-m_p)^2}{s_p^4}\right)|Y\right\},\\
\text{triu } \mathbb{E}_{X|Y}\left\{I^m(s^2;X)\right\} &= \mathbb{C}ov\left\{\sum_k \frac{Z_p^k}{2}\left(\frac{-n_k}{s_p^2}+\frac{\sum_{t\in I_k}(y_t-m_p)^2}{s_p^4}\right);\right.\\
&\qquad \left.\sum_k \frac{Z_\ell^k}{2}\left(\frac{-n_k}{s_\ell^2}+\frac{\sum_{t\in I_k}(y_t-m_\ell)^2}{s_\ell^4}\right)|Y\right\}.
\end{aligned}$$

$$\begin{aligned}
\text{diag } \mathbb{E}_{X|Y}\left\{I^m(s^2;X)\right\} &= \sum_k \frac{1}{4}\tau_p^k(1-\tau_p^k)\left(\frac{-n_k}{s_p^2}+\frac{\sum_{t\in I_k}(y_t-m_p)^2}{s_p^4}\right)^2,\\
\text{triu } \mathbb{E}_{X|Y}\left\{I^m(s^2;X)\right\} &= -\sum_k \frac{1}{4}\tau_p^k\tau_\ell^k\left(\frac{-n_k}{s_p^2}+\frac{\sum_{t\in I_k}(y_t-m_p)^2}{s_p^4}\right)\left(\frac{-n_k}{s_\ell^2}+\frac{\sum_{t\in I_k}(y_t-m_\ell)^2}{s_\ell^4}\right)
\end{aligned}$$

Missing-data information matrix for means and proportions parameters

$$\begin{aligned}
\text{diag } \mathbb{E}_{X|Y}\left\{I^m(\pi,m;X)\right\} &= \mathbb{C}ov\left\{\sum_k \frac{Z_p^k}{\pi_p}-\frac{Z_p^k}{\pi_P};\sum_k \frac{Z_p^k}{s_p^2}\sum_{t\in I_k}(y_t-m_p)|Y\right\},\\
\text{triu } \mathbb{E}_{X|Y}\left\{I^m(\pi,m;X)\right\} &= \mathbb{C}ov\left\{\sum_k \frac{Z_\ell^k}{\pi_\ell}-\frac{Z_P^k}{\pi_P};\sum_k \frac{Z_{kp}}{s_p^2}\sum_{t\in I_k}(y_t-m_p)|Y\right\}.
\end{aligned}$$

$$\begin{aligned}
\text{diag } \mathbb{E}_{X|Y}\left\{I^m(\pi,m;X)\right\} &= \sum_k\left[\frac{\tau_p^k(1-\tau_p^k)}{\pi_p}+\frac{\tau_p^k\tau_P^k}{\pi_P}\right]\left[\frac{\sum_{t\in I_k}(y_t-m_p)}{s_p^2}\right],\\
\text{triu } \mathbb{E}_{X|Y}\left\{I^m(\pi,m;X)\right\} &= \sum_k\left[\frac{\tau_P^k\tau_p^k}{\pi_P}-\frac{\tau_p^k\tau_\ell^k}{\pi_\ell}\right]\left[\frac{\sum_{t\in I_k}(y_t-m_p)}{s_p^2}\right].
\end{aligned}$$

Missing-data information matrix for variances and proportions parameters

$$\begin{aligned}
\text{diag } \mathbb{E}_{X|Y}\left\{I^m(\pi,s^2;X)\right\} &= \mathbb{C}ov\left\{\sum_k \frac{Z_p^k}{\pi_p}-\frac{Z_P^k}{\pi_P};\sum_k \frac{Z_p^k}{2}\left(\frac{-n_k}{s_p^2}+\frac{\sum_{t\in I_k}(y_t-m_p)^2}{s_p^4}\right)|Y\right\},\\
\text{triu } \mathbb{E}_{X|Y}\left\{I^m(\pi,s^2;X)\right\} &= \mathbb{C}ov\left\{\sum_k \frac{Z_\ell^k}{\pi_\ell}-\frac{Z_P^k}{\pi_P};\sum_k \frac{Z_p^k}{2}\left(\frac{-n_k}{s_p^2}+\frac{\sum_{t\in I_k}(y_t-m_p)^2}{s_p^4}\right)|Y\right\}.
\end{aligned}$$

$$\begin{aligned}
\text{diag } \mathbb{E}_{X|Y}\left\{I^m(\pi, s^2; X)\right\} &= \frac{1}{2}\sum_k \left(\frac{\tau_p^k(1-\tau_p^k)}{\pi_p} + \frac{\tau_P^k\tau_p^k}{\pi_P}\right)\left(\frac{-n_k}{s_p^2} + \frac{\sum_{t\in I_k}(y_t - m_p)^2}{s_p^4}\right), \\
\text{triu } \mathbb{E}_{X|Y}\left\{I^m(\pi, s^2; X)\right\} &= \frac{1}{2}\sum_k \left(\frac{\tau_P^k\tau_p^k}{\pi_P} - \frac{\tau_p^k\tau_\ell^k}{\pi_\ell}\right)\left(\frac{-n_k}{s_p^2} + \frac{\sum_{t\in I_k}(y_t - m_p)^2}{s_p^4}\right).
\end{aligned}$$

Missing-data information matrix for means and variances parameters

$$\begin{aligned}
\text{diag } \mathbb{E}_{X|Y}\left\{I^m(m, s^2; X)\right\} &= \mathbb{C}ov\left\{\sum_k \frac{Z_p^k}{s_p^2}\sum_{t\in I_k}(y_t - m_p)\ ; \right. \\
&\qquad \left. \sum_k \frac{Z_p^k}{2}\left(\frac{-n_k}{s_p^2} + \frac{\sum_{t\in I_k}(y_t - m_p)^2}{s_p^4}\right)|Y\right\}, \\
\text{triu } \mathbb{E}_{X|Y}\left\{I^m(\pi, s^2; X)\right\} &= \mathbb{C}ov\left\{\sum_k \frac{Z_p^k}{s_p^2}\sum_{t\in I_k}(y_t - m_p)\ ; \right. \\
&\qquad \left. \sum_k \frac{Z_\ell^k}{2}\left(\frac{-n_k}{s_\ell^2} + \frac{\sum_{t\in I_k}(y_t - m_\ell)^2}{s_\ell^4}\right)|Y\right\}.
\end{aligned}$$

$$\begin{aligned}
\text{diag } \mathbb{E}_{X|Y}\left\{I^m(m, s^2; X)\right\} &= \sum_k \frac{1}{2}\tau_p^k(1-\tau_p^k)\left[\frac{\sum_{t\in I_k}(y_t - m_p)}{s_p^2}\right]\left[\frac{-n_k}{s_p^2} + \frac{\sum_{t\in I_k}(y_t - m_p)^2}{s_p^4}\right], \\
\text{triu } \mathbb{E}_{X|Y}\left\{I^m(\pi, s^2; X)\right\} &= -\sum_k \frac{1}{2}\tau_p^k\tau_\ell^k\left[\frac{\sum_{t\in I_k}(y_t - m_p)}{s_p^2}\right]\left[\frac{-n_k}{s_\ell^2} + \frac{\sum_{t\in I_k}(y_t - m_\ell)^2}{s_\ell^4}\right].
\end{aligned}$$

7.7.3 Practical calculation of the information matrix

Despite a long calculus, all these terms can be easily calculated within the EM framework. It requires the calculus of the missing information matrix $\mathbb{E}_{X|Y}\left\{I^m(\psi; X)\right\}$ at $\psi = \hat{\psi}$, and the computation of

$$I(\hat{\psi}; Y) = \mathbb{E}_{X|Y}\left\{I^c(\psi; X)\right\}\big|_{\psi=\hat{\psi}} - \mathbb{E}_{X|Y}\left\{I^m(\psi; X)\right\}\big|_{\psi=\hat{\psi}}.$$

Unfortunatelly there is no simple form of the missing information matrix even for $\psi = \hat{\psi}$.

Chapter 8

Model selection

In practice neither the number of segments nor the number of clusters are known, and they should be estimated. Nevertheless, the joint estimation of the number of segments and groups is new and no method has yet been proposed. This is why we proceed in two steps.

Different methods have been proposed in the context of segmentation methods to choose the number of segments (see Chapter 4), and in the context of mixture models to choose the number of clusters (see Chapter 6). These methods are largely based on penalized likelihood criteria and our primary objective is to determine if they can be applied to our case. This is why we propose in a first step to study model selection strategies in two situations: when the number of clusters is fixed, and when the number of segments is fixed. Then the second step will be to develop a heuristic to select the number of clusters first and then to deduce the number of segments.

8.1 Selection of K when P is fixed

Since model selection criteria are mainly based on the likelihood of the model, we aim at studying the behavior of the likelihood of the segmentation/clustering model when P is fixed. In a first step we show that this likelihood is not necessarily increasing, since models with increasing numbers of segments are not nested.

8.1.1 Non-nested models

Proposition

Denoting $\mathcal{M}(K,P)$ the set of all segmentation/clustering models with K segments and P clusters, $\mathcal{M}(K,P)$ and $\mathcal{M}(K+1,P)$ are not nested such that:

$$\mathcal{M}(K,P) \not\subset \mathcal{M}(K+1,P).$$

Proof

Let us note $\mathcal{T}_K$ the set of possible breakpoints and Ψ_P the set of mixture parameters:

$$\begin{aligned}\mathcal{T}_K &= \{1 = t_0 < t_1 < t_2 < \ldots < t_{K-1} < t_K = n,\ \ t_k \in \{2, \ldots, n-1\}\} \\ \Psi_P &= \{\pi_1, \ldots, \pi_P; m_1, \ldots, m_P; s_1, \ldots, s_P \ \mid \\ &\qquad 0 \leq \pi_p \leq 1, \sum_{p=1}^{P} \pi_p = 1; m_p \in \mathbb{R}, s_p \in \mathbb{R}\}.\end{aligned}$$

The fact that $\mathcal{M}(K, P)$ and $\mathcal{M}(K+1, P)$ are not nested is due to the discrete nature of breakpoints. Since segments of null size are not allowed in the model, it follows that:

$$\mathcal{T}_K \not\subset \mathcal{T}_{K+1},$$

meaning that:

$$\mathcal{M}(K, P) \not\subset \mathcal{M}(K+1, P).$$

Since the models are not nested it follows that the log-likelihood does not necessarily increase when the number of segments increases. In the following, we provide an illustrated example to interpret this particular behavior.

8.1.2 A likelihood that can decrease

We study a sequence $y = \{y_1, \ldots, y_n\}$ with $P = 2$ clusters and $K = 4$ segments, simulated with parameters defined in Equation (8.1). This sequence is shown in Figure 8.1, top.

$$\left\{\begin{array}{rcl} m &=& \begin{bmatrix} 0 & 5 \end{bmatrix}, \\ s^2 &=& \begin{bmatrix} 1 & 1 \end{bmatrix}, \\ \pi &=& \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}, \\ n_k &=& 20, \\ T &=& \begin{Bmatrix} 20 & 40 & 60 & 80 \end{Bmatrix}. \end{array}\right. \tag{8.1}$$

In practice we noticed that the addition of new segments could result in a decrease in the log-likelihood of the model. In order to illustrate this particular behavior, the log-likelihood $\log \mathcal{L}_{KP}(\hat{T}; \hat{\psi})$ is calculated in the case of example 1, for a fixed number of clusters ($P = 2$) and for an increasing number of segments ($K = 2, \ldots, 20$). This curve is represented in Figure 8.1 (bottom), and shows a decreasing log-likelihood when the number of segments is greater than 4.

When considering a partition with K segments, suppose that segment Y^ℓ belongs to cluster p_0. This means that:

$$f(Y^\ell; \psi_P) = \sum_{p=1}^{P} \pi_p f(Y^\ell; \theta_p) \simeq \pi_{p_0} f(Y^\ell; \theta_{p_0}).$$

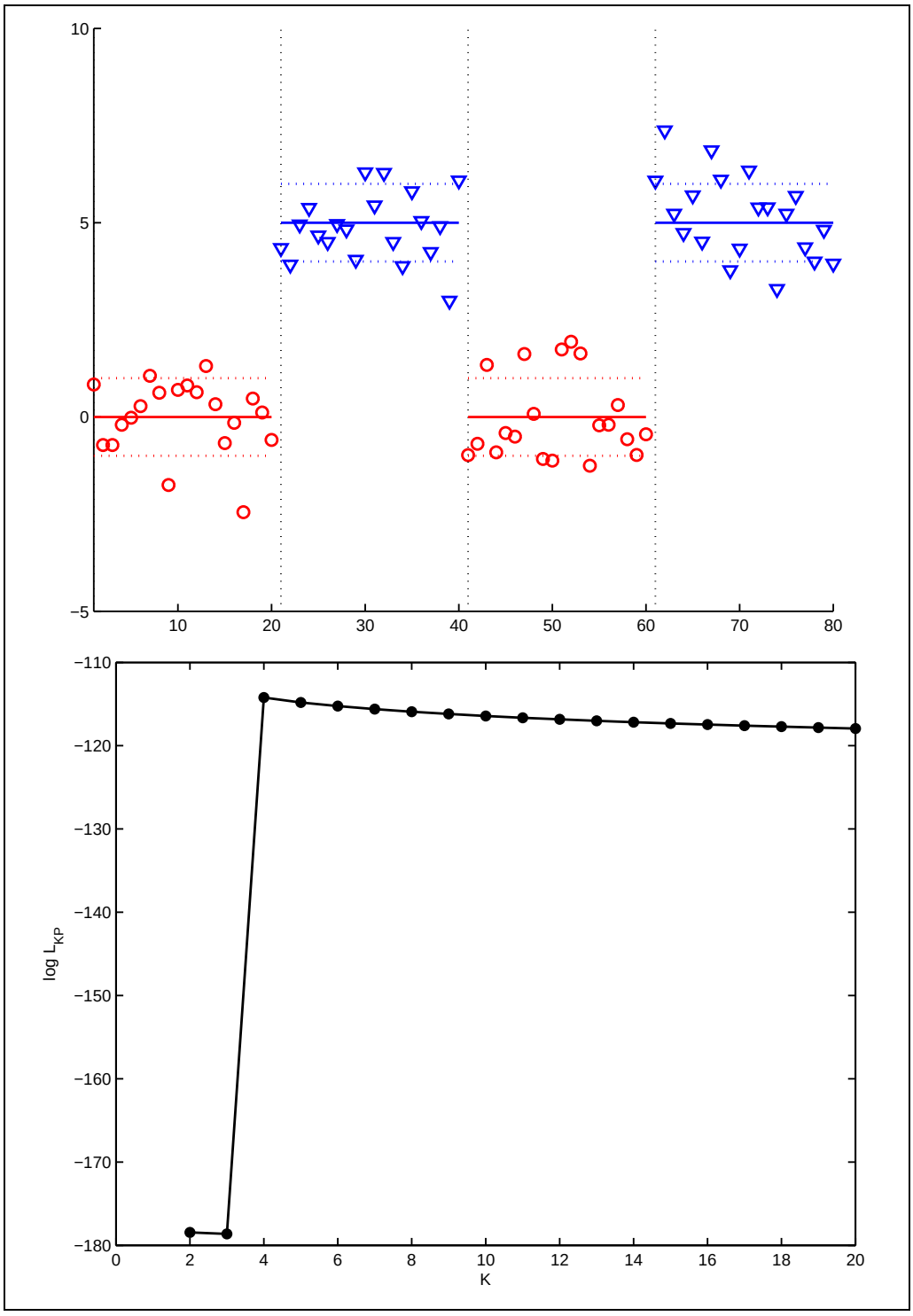


Figure 8.1: Example of simulation with 2 clusters and 4 segments (top). Behavior of the incomplete-data log-likelihood according to the number of segments for P=2 clusters (bottom).

The log-likelihood of this configuration can be approximated such that:

$$\log \mathcal{L}_{KP}(\hat{T}_K; \hat{\psi}_P) \simeq \sum_{k \neq \ell}^{K-1} \log \left\{ \sum_{p=1}^{P} \hat{\pi}_p f(Y^k; \hat{\theta}_p) \right\} + \log \left\{ \hat{\pi}_{p_0} f(Y^\ell; \hat{\theta}_{p_0}) \right\}$$

This approximation seems realistic in the case of well separated clusters, like in the simulated example. Then we consider the case where a new segment is added and where this addition only concerns the cut of segment Y^ℓ into (Y^{ℓ_1}, Y^{ℓ_2}). Since they remain conditionally independent we have:

$$\forall p, \quad f(Y^\ell; \theta_p) = f(Y^{\ell_1}; \theta_p) \times f(Y^{\ell_2}; \theta_p).$$

If Y^ℓ belongs to p_0 and if the creation of new segments does not affect the labelling of Y^{ℓ_1} and Y^{ℓ_2}, the log-likelihood of this new configuration is approximated such that:

$$\begin{aligned} \log \mathcal{L}_{K+1,P}(\hat{T}_{K+1}; \hat{\psi}_P) \simeq & \sum_{k \neq \ell_1, \ell_2}^{K-1} \log \left\{ \sum_{p=1}^{P} \hat{\pi}_p f(Y^k; \hat{\theta}_p) \right\} \\ & + \log \left\{ \hat{\pi}_{p_0} f(Y^{\ell_1}; \hat{\theta}_{p_0}) \right\} + \log \left\{ \hat{\pi}_{p_0} f(Y^{\ell_2}; \hat{\theta}_{p_0}) \right\}. \end{aligned}$$

If we consider that partitions $\hat{T}_K$ and $\hat{T}_{K+1}$ are nested [1] it follows that the log-likelihood can decrease since:

$$\log \mathcal{L}_{K+1,P}(\hat{T}_{K+1}; \hat{\psi}_P) - \log \mathcal{L}_{KP}(\hat{T}_K; \hat{\psi}_P) \simeq \log(\hat{\pi}_{p_0}) \leq 0.$$

The last step would be to show that there exists a number of segments $\tilde{K}_P$ for which $\log \mathcal{L}_{KP}(\hat{T}, \hat{\psi})$ decreases. The existence of this number can be shown using some approximations as above, but it appears that the form of the likelihood hampers every theoretical calculus. In the following, we will suppose that this number exists.

In order to understand the particular behavior of the likelihood, we represent the segmentation/clustering results when the data are partitioned into more than 4 segments (Figure 8.2). Since the two clusters are well separated, the true configuration is recovered for $K = 4$. Then we want to know what the result is when a new segment is added. Since segments of null size are not allowed in the model, the addition of new segments leads to segments of minimal size (1 data point) without any change in clustering results. As shown in Figure 8.2 the creation of segments of size 1 can affect either cluster 1 or 2. This decrease in the log-likelihood could be interpreted as follows. Even if a new segment is added this addition does not necessarily lead to an increase in the quality of fit of the mixture model to the data. This means that when the number of clusters is fixed, there exists a number of segments for which the quality of fit of the model is maximal.

[1] Nested partitions means that $T_{K+1} = T_K \cup \{t_{K+1}\}$. This does not mean that models are nested.

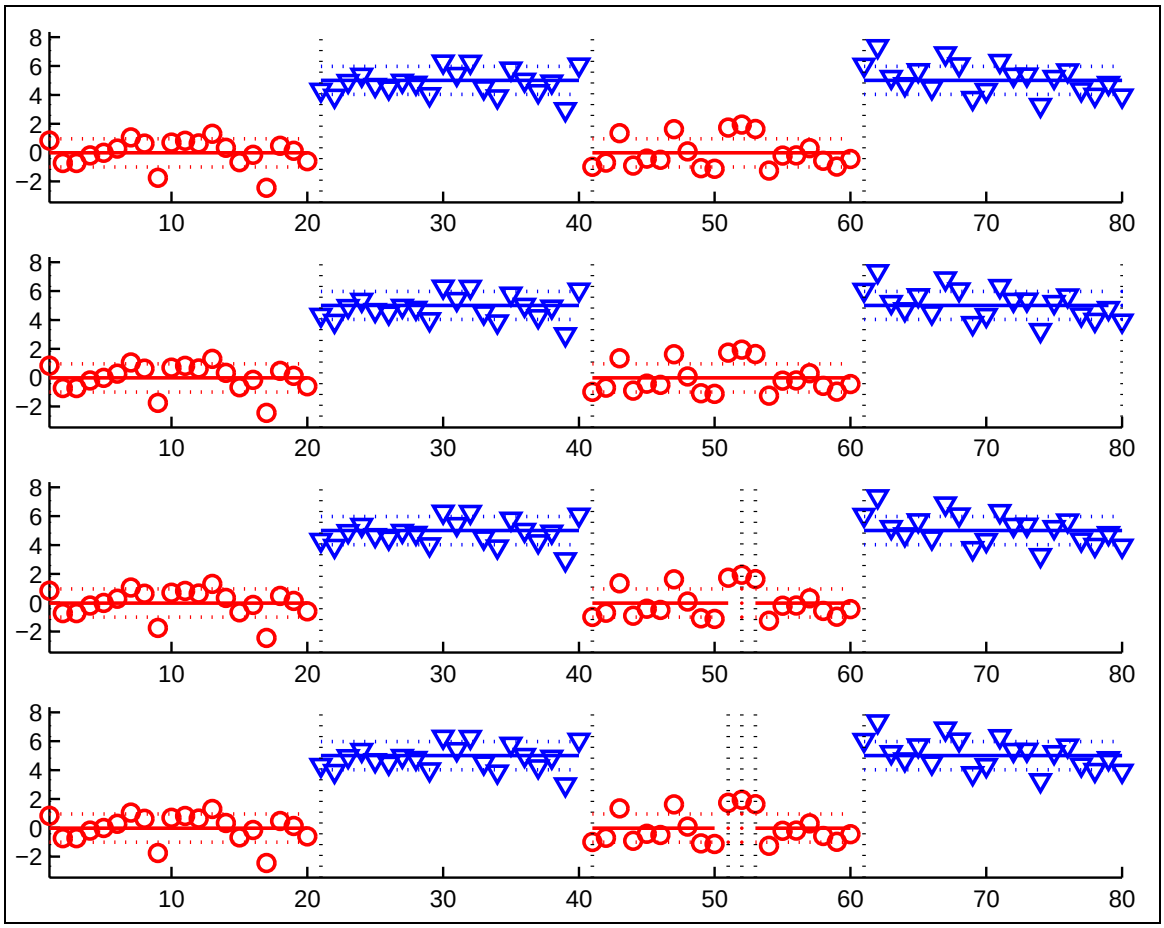


Figure 8.2: Segmentation/classification results when the number of segments increases (K=4,5,6,7). The number of clusters is $P = 2$. Simulation 1.

8.1.3 Model selection

Once the behavior of the model has been specified when the number of segments increases, a natural question would be to select the number of segments when the number of clusters is fixed.

In example 1 (Figure 8.1), it is clear that clusters are well separated. This is why the introduction of more than 4 segments does not lead to an interesting segmentation/classification result. Consequently the log-likelihood is strongly decreasing and an intuitive selection method would consist in the selection of the number of segments for which the log-likelihood decreases. Nevertheless a question would be to study the behavior of the model when clusters are not well separated. Figure 8.3 (top) illustrates a simulation which is similar to the first one, with clusters that are less separated, and Figure 8.3 (bottom) shows the log-likelihood for $P = 2$.

We notice that the log-likelihood is still decreasing, but for a number of segments ($K = 8$) which is more important than the true one ($K = 4$). We can distinguish 3 steps: in the first step ($K = 2, \ldots, 4$), the addition of new segments leads to an important increase in the fit of the model, in the second step ($K = 5, \ldots, 8$) this increase is less important, and in the last step ($K = 9, \ldots, K_{max}$), the log-likelihood decreases. If we study the segmentation/clustering results (Figure 8.4), the true configuration is not recovered for $K = 4$, since a breakpoint is at the wrong coordinate ($\hat{t}_3 = 65$ whereas $t_3 = 61$). However this error seems "reasonable" since the variance of each group is high regarding the means difference. If the number of segments is 8, new segments are added and affected to the closest group in terms of mean. The "variance" effect could explain the fact that some points may appear close to the first group with zero mean. Nevertheless, if clusters were less separated, selecting the number of segments that makes the log-likelihood decrease would lead to an overly segmented profile whose interpretation would be difficult.

Therefore the natural decrease that occurs when the number of segments increases is not sufficient to select the number of segments. This means that the log-likelihood should be penalized in addition to the observed penalization.

An empirical method

We propose a first method to select the number of segments when the number of clusters is fixed, which is derived from procedure proposed by Lavielle (2005) and discussed in Chapter 4. The purpose of this method is to determine if there exists a number of segments for which the log-likelihood ceases to increase significantly between the minimum number of segments $K = P$ and the number of segments for which the log-likelihood decreases $K = \tilde{K}_P$. This is why we propose to isolate the part of the curve for which the likelihood increases.

Denoting $J_K = -\log \mathcal{L}_{KP}(\hat{T}, \hat{\psi})$, with P being fixed, we calculate the normal-

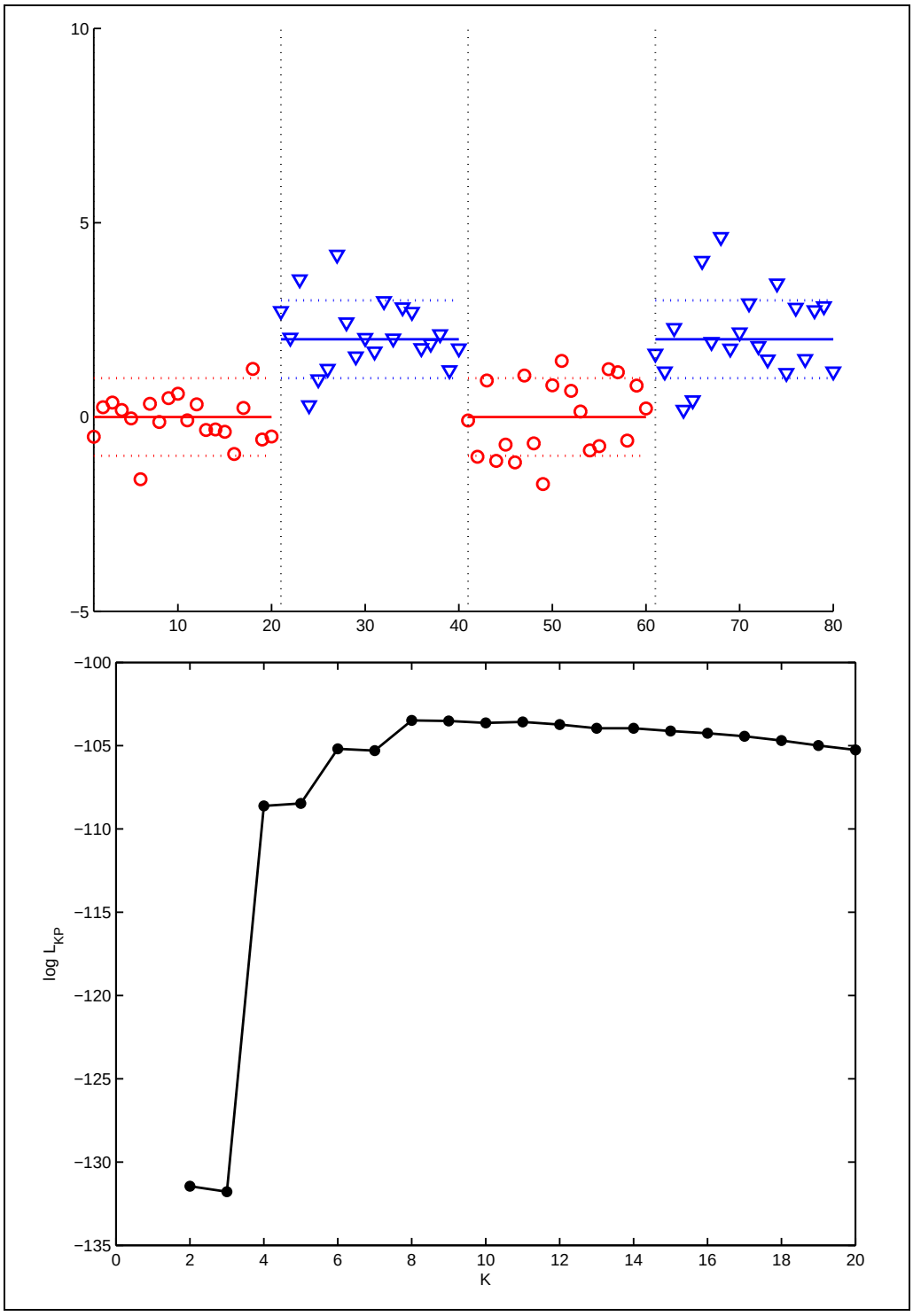


Figure 8.3: Simulation 2 with 2 clusters and 4 segments (top). Behavior of the incomplete-data log-likelihood according to the number of segments for P=2 clusters (bottom).

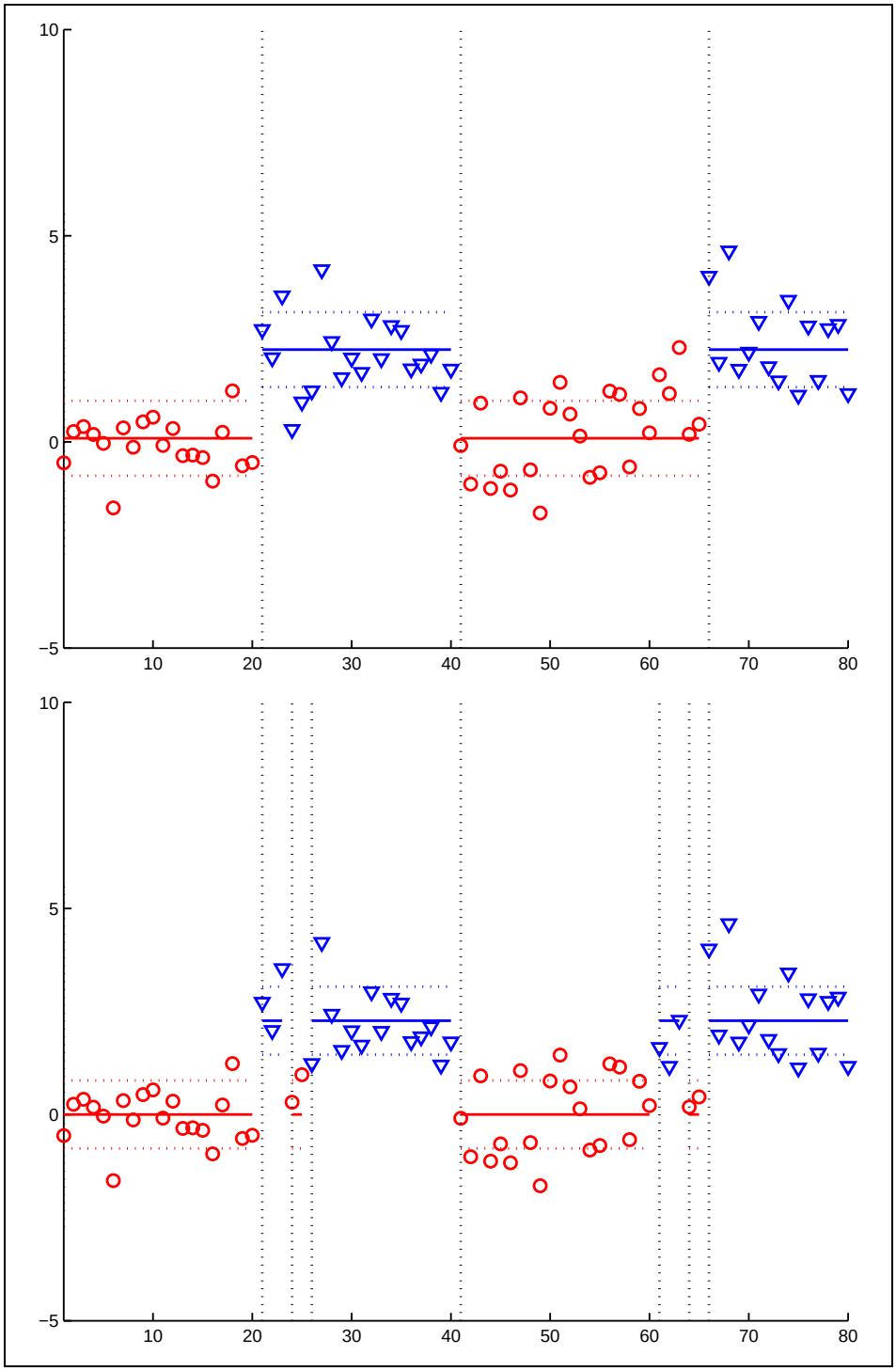


Figure 8.4: Segmentation/clustering results for simulation 2, with $K = 4$ (top) and $K = 8$ (bottom).

ized contrast, such as:

$$\forall K \in \{P, \dots, \tilde{K}_P\} \quad \tilde{J}_K = \frac{J_{\tilde{K}_P} - J_K}{J_{\tilde{K}_P} - J_P}(\tilde{K}_P - P) + P,$$

such that $\tilde{J}_{\tilde{K}_P} = P$ and $\tilde{J}_P = \tilde{K}_P$. Then we propose to calculate the empirical second derivative of $\tilde{J}_K$, such that:

$$\forall K \in \{P+1, \dots, \tilde{K}_P - 1\}, \quad D_K = \tilde{J}_{K+1} - 2\tilde{J}_K + \tilde{J}_{K-1}.$$

Then select the number of clusters, such that:

$$\hat{K}_P = \max_K \left\{ K \in \{P+1, \dots, \tilde{K}_P - 1\} \mid D_K \geq s \right\},$$

Of course this method requires $\tilde{K}_P \geq P + 2$. If this condition is not true, we choose $\hat{K} = \tilde{K}_P$.

One major problem with this method is that $\tilde{K}_P$ may be close to P meaning that in some configurations, the second derivative may be calculated on few points. For instance in example 2, we have $\tilde{K}_P = 8$ for $P = 2$. Since this could lead to some instabilities in the result of the procedure we propose a second strategy.

A second method to choose K

Another possibility could be to consider a penalty term without theoretical justification. We will consider the following criterion to select the number of segments:

$$\hat{K}_P = \operatorname*{Argmax}_K \{\log \mathcal{L}_{KP}(\hat{T}, \hat{\psi}) - \frac{1}{2}\log(n) \times K\}.$$

This criterion could be interpreted as a *pseudo*-BIC criterion, which penalizes the addition of new segments as if they were continuous parameters. Applied to example 2, this method would select 4 segments as shown in Figure 8.5. Since these criteria are empirically motivated, their performance should be addressed using simulation studies. This is the purpose of the next part.

8.1.4 No application of existing methods for model selection

We proposed two model selection methods to select the number of segments. These methods are empirically motivated and we want to explain why existing methods can not be applied to our case.

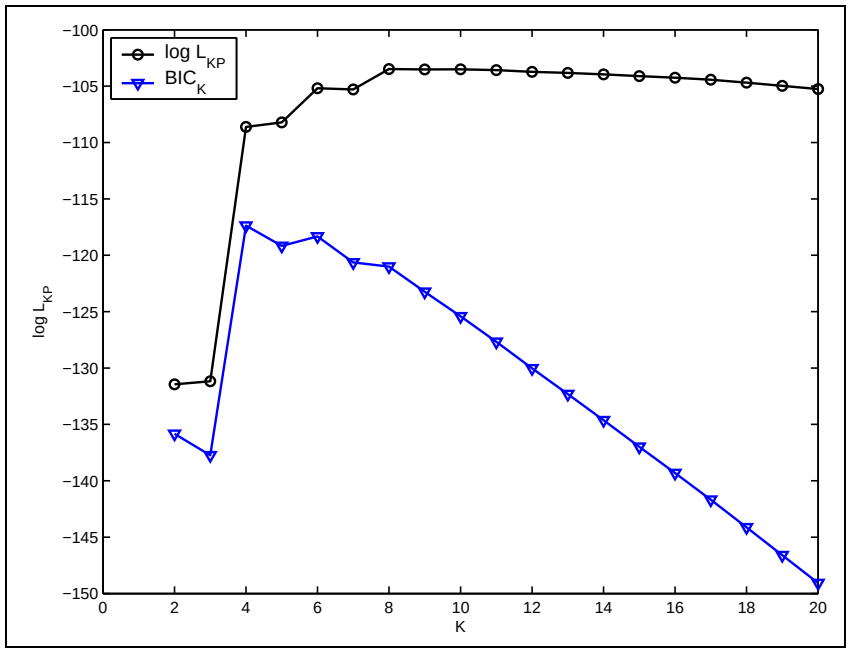


Figure 8.5: Selecting the number of segments with $K/2\log(n)$ as a penalty when $P = 2$, example 2.

Bayesian strategy

Bayesian techniques have been successfuly applied to select the number of segments in the segmentation context. Then the question is: why can not we derive a Bayesian criterion in our case? The difficulty lies in the fact that breakpoints are discrete parameters.

Since these parameters are discrete, the likelihood of a segmentation model is not continuous with respect to these parameters. Thus the application of Bayesian techniques requires the breakpoints to be fixed. Consequently the BIC focuses on the number of continuous parameters of the model. When applied to segmentation models the resulting penalty term is $(K + 1)\log(n)$ where $K + 1$ does not mean $K + 1$ breakpoints but K means and 1 variance, which are the continuous parameters of the model.

If we wanted to use similar techniques to penalize the likelihood of the segmentation/clustering model, the breakpoints would have to be fixed as well. When the breakpoints are fixed the number of continous parameters is P means P variances and $P - 1$ mixing proportions. This means that the number of continuous parameters is independent of the number of discrete parameters. Consequently, when the number of segments increases, the penalty term would be $3P - 1$ which does not penalize the addition of new segments.

Birgé-Massart strategy

Another stragegy which could have been used is the one developed by Birgé and Massart (2001). This method has been applied to the case of segmentation models (Lebarbier (2005)). Contrary to Bayesian techniques, this method exploits the fact that breakpoint coordinates are discrete parameters, and considers the number of possible segmentations which is $\mathcal{C}_{n-1}^{K-1}$ for a model with K segments. Nevertheless, the technique proposed by Birgé and Massart (2001) to derive a penalty term seems difficult to apply to our case, first because of the unusual behavior of the likelihood with respect to the number of segments, and also because this technique has never been developed for models with hidden structure.

8.2 Selection of P when K is fixed

Now that we have proposed selection methods for K, the next question is to select P when K is fixed. Let us recall that when the number of segments is fixed at K, the purpose of the segmentation/clustering model is to cluster segments into an increasing number of groups. In order to illustrate the behavior of the likelihood in this case, we propose to consider example 2 previously defined. In Table 8.1 are shown the resulting mixture model estimators when the number of segments is fixed at $K = 6$, for an increasing number of clusters ($P = 1, \ldots, K$). We also provide some segmentation/clustering results in Figure 8.6.

8.2.1 Nested models

Since the number of segments is the number of statistical units of the mixture model, it is clear that the model is constraint when the number of clusters increases. Consequently, when P increases whereas the number of statistical units is constant, means' estimators can be equal for different groups, and proportions' estimates can be small. As a result, the maximum *a posteriori* rule which is used to cluster segments will create empty clusters, as shown in Figure 8.6. Moreover this figure shows that the resulting segmentation $\hat{T}_K$ depends on the number of clusters and should be noted $\hat{T}_K(\hat{\psi}_P)$.

Proposition

Denoting $\mathcal{M}(K, P)$ the set of segmentation/clustering model with K segments and P clusters, $\mathcal{M}(K, P)$ and $\mathcal{M}(K, P+1)$ are nested such as:

$$\mathcal{M}(K, P) \subset \mathcal{M}(K, P+1).$$

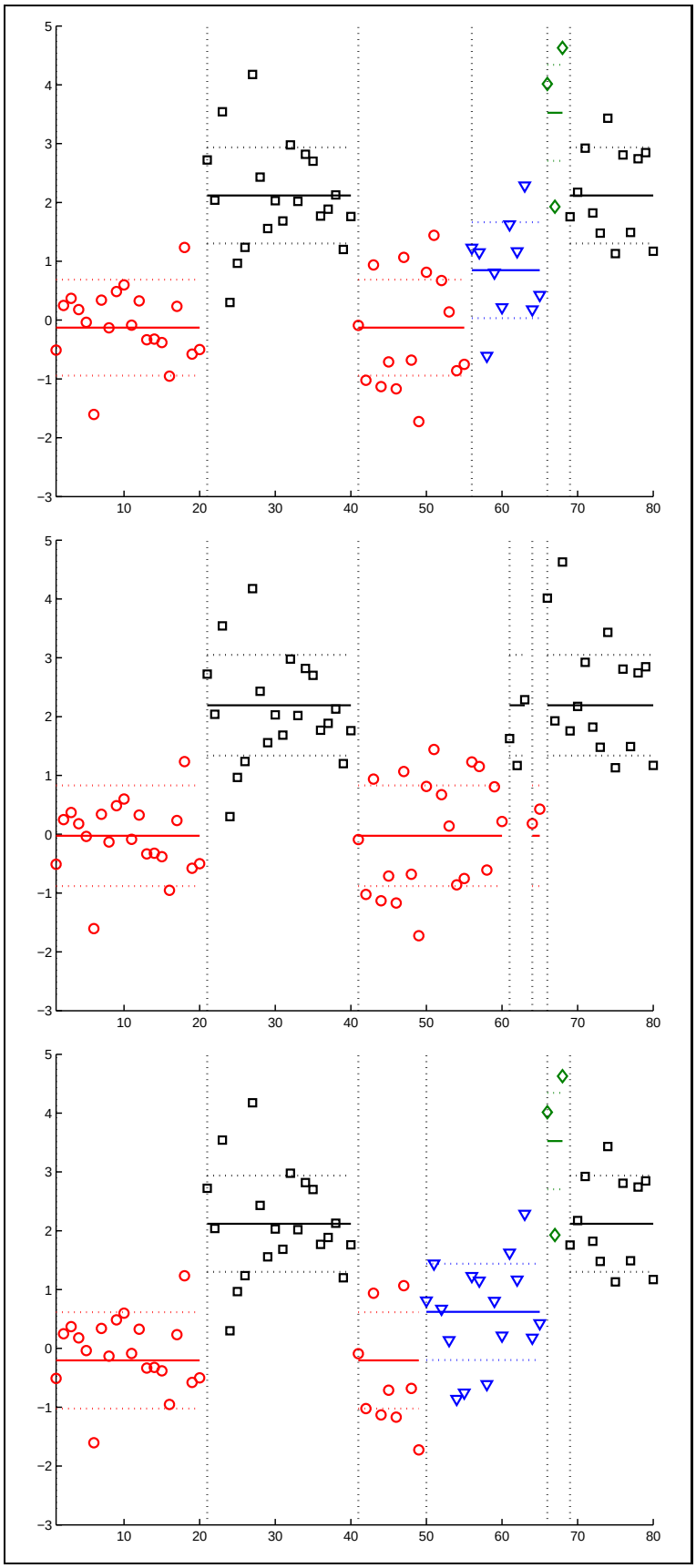


Figure 8.6: Segmentation/clustering results when the number of clusters increases ($P = 4, 5, 6$).

P	$\hat{m}_1$	$\hat{m}_2$	$\hat{m}_3$	$\hat{m}_4$	$\hat{m}_5$	$\hat{m}_6$
1	1.0284					
2	-0.0258	2.1933				
3	-0.0258	-0.0258	2.1933			
4	-0.1281	0.8485	2.1193	3.5242		
5	-0.0258	-0.0258	-0.0258	2.1933	2.1933	
6	-0.2083	-0.2039	0.6206	2.1194	2.1194	3.5242
	$\hat{\pi}_1$	$\hat{\pi}_2$	$\hat{\pi}_3$	$\hat{\pi}_4$	$\hat{\pi}_5$	$\hat{\pi}_6$
1	1.0000					
2	0.4992	0.5008				
3	0.4931	0.0061	0.5008			
4	0.3336	0.1664	0.3373	0.1627		
5	0.3444	0.1487	0.0061	0.2969	0.2039	
6	0.0884	0.2449	0.1667	0.1241	0.2133	0.1626

Table 8.1: Estimated means and proportions for a segmentation/clustering model with $K = 6$ segments (fixed) for an increasing number of clusters

Proof

Let us note $\mathcal{T}_K$ the set of possible breakpoints and Ψ_P the set of mixture parameters:

$$\begin{aligned}\mathcal{T}_K &= \{1 = t_0 < t_1 < t_2 < \ldots < t_{K-1} < t_K = n, \;\; t_k \in \{2, \ldots, n-1\}\} \\ \Psi_P &= \{\pi_1, \ldots, \pi_P; m_1, \ldots, m_P; s_1, \ldots, s_P \;\; | \\ &\qquad 0 \leq \pi_p \leq 1, \sum_{p=1}^{P} \pi_p = 1; m_p \in \mathbb{R}, s_p \in \mathbb{R}\}.\end{aligned}$$

If partition $T_K \in \mathcal{T}_K$ was fixed and did not depend on the number of clusters, it is clear that mixture models would be nested since parameter π_{P+1} can be set to 0. This implies that

$$\mathcal{M}(T_K, P) \subset \mathcal{M}(T_K, P+1).$$

However the dimension of the set Ψ_P does not depend on T_K meaning that:

$$\forall T_K \in \mathcal{T}_K, \;\; \Psi_P \subset \Psi_{P+1}.$$

It follows that

$$\mathcal{M}(K, P) \subset \mathcal{M}(K, P+1),$$

whatever the breakpoints.

Lemma

For a segmentation/clustering model with K segments and P clusters, the model log-likelihood $\log \mathcal{L}_{K,P}(\hat{T}, \hat{\psi})$ is an increasing function with respect to P :

$$\log \mathcal{L}_{K,P}(\hat{T}, \hat{\psi}) \leq \log \mathcal{L}_{K,P+1}(\hat{T}, \hat{\psi})$$

Proof

The proof of this lemma is straighforward regarding the preceeding proposition.

8.2.2 Model selection

Contrary to the previous section, the behavior of the model's log-likelihood is "conventional" with respect to the number of clusters since the likelihood increases with the complexity of the model. The sequence of increasing log-likelihoods is represented in Figure 8.7 for example 2 with K fixed at 6 and with $P = 1, \ldots, K$. It can be seen that this log-likelihood is constant when $P \geq 2$. In order to select the number of clusters when the number of segments is fixed, we could use a penalized version of the log-likelihood and select the number of clusters such that:

$$\hat{P}_K = \underset{P}{\text{Argmax}} \left\{ \log \mathcal{L}_{K,P}(\hat{T}, \hat{\psi}) - \beta_K pen(P) \right\}.$$

The next step would be to find $\beta_K pen(P)$ analytically. Nevertheless the selection of the number of clusters for a given number of segments is of little interest from a practical point of view. In the context of array CGH data analysis for instance, fixing the number of segments at some value appears contradictory since the objective of these studies is to determine how many chromosomal aberrations there are in a CGH profile. To this extent, we choose to focus on the next section, which deals with the joint selection of the number of segments and clusters.

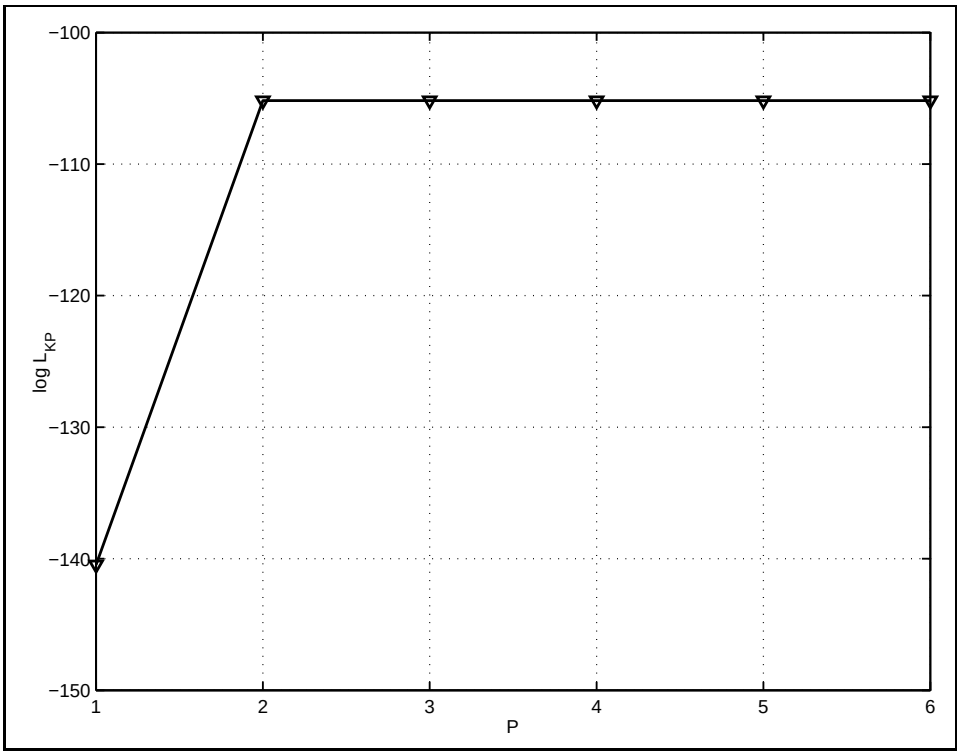


Figure 8.7: Representation of the log-likelihood of the model according to the number of clusters, for a fixed number of segments $K = 6$.

8.3 A heuristic to select K and P

We proposed an algorithm to estimate the parameters of the model when the number of segments and the number of clusters are fixed. In practice these numbers are not known and should be estimated, as it is the case in the segmentation context (Chapter 4) or in the context of model-based clustering (Chapter 6). The originality of our problem is that these numbers should be jointly estimated in order to select a model with $\hat{K}$ segments and $\hat{P}$ clusters. Existing methods have been proposed in the context of segmentation or for model-based clustering, based on a penalization of the model's log-likelihood. In our case, the log-likelihood depends on both K and P. It is a surface that can be represented like in Figure 8.8. An ideal result would be to adapt existing results to our problem, and to derive a penalty to select K and P such that:

$$(\hat{K}, \hat{P}) = \underset{KP}{\text{Argmax}} \left\{ \log \mathcal{L}_{KP}(\hat{T}, \hat{\psi}) - \beta \times pen(K, P) \right\}.$$

Nevertheless, previous discussions on model selection justify the difficulty to derive such a criterion theoretically. Moreover, we showed that the complexity of the model only depends on the number of continuous parameters of the mixture, and not of the number of discrete parameters. This why we choose to adopt a sequential strategy selecting the number of clusters first. Moreover the construction of a heuristic for model selection should consider the final objective of the method. In array CGH data analysis for instance, the primary goal of the method

is to cluster chromosomal regions into a finite number of clusters. Consequently choosing the number of clusters first seems reasonable from a practical point of view.

8.3.1 Selecting the number of clusters

In a first step, we focus on the estimation of the number of clusters. In Section 8.2 we discussed the fact that selecting the number of clusters for a fixed number of segments had little interest in practice. This is why we need to select the number of clusters whatever the number of segments. The following proposition aims at constructing a sequence of increasing log-likelihoods which can be used for this purpose.

Proposition

Hypothesis (H):

$$\forall P \in \{1, \ldots, P_{max}\}, \;\; \exists \tilde{K}_P, \; \textit{such that} : \tilde{K}_P = \underset{K}{\text{Argmax}} \left\{ \log \mathcal{L}_{KP}(\hat{T}; \hat{\psi}) \right\}.$$

For a set of segmentation/clustering models with P clusters, $P \in \{1, \ldots, P_{max}\}$ and K segments, $K \in \{P, \ldots, n\}$, under hypothesis (H) there exists a sequence of increasing log-likelihoods noted $\log \widetilde{\mathcal{L}}_P$ such that $\log \widetilde{\mathcal{L}}_1 \ldots \leq \log \widetilde{\mathcal{L}}_P \leq \ldots \log \widetilde{\mathcal{L}}_{P_{max}}$ with

$$\log \widetilde{\mathcal{L}}_P = \max_K \left\{ \log \mathcal{L}_{KP}(\hat{T}_K; \hat{\psi}_P) \right\}.$$

Proof

The proof of this proposition uses the results that have been shown is Sections 8.1 and 8.2. We note $\mathcal{M}(K, P)$ the set of all segmentation/clustering models with K segments and P clusters. In section 8.2 we showed that

$$\mathcal{M}(K, P) \subset \mathcal{M}(K, P+1),$$

and that

$$\forall K \in \{P, \ldots, n\} \quad \log \mathcal{L}_{KP}(\hat{T}, \hat{\psi}) \leq \log \mathcal{L}_{K,P+1}(\hat{T}, \hat{\psi}). \tag{8.2}$$

Using Hypothesis (H) we suppose that there exists a sequence $\{\tilde{K}_1, \ldots, \tilde{K}_{P_{max}}\}$ such that:

$$\tilde{K}_P = \underset{K}{\text{Argmax}} \left\{ \log \mathcal{L}_{KP}(\hat{T}; \hat{\psi}) \right\}.$$

Considering Equation (8.2) and the number of segments $\tilde{K}_P$ which maximizes the log-likelihood of a model with P clusters it follows that:

$$\log \mathcal{L}_{\tilde{K}_P,P}(\hat{T}, \hat{\psi}) \leq \log \mathcal{L}_{\tilde{K}_P,P+1}(\hat{T}, \hat{\psi}). \tag{8.3}$$

Since

$$\tilde{K}_{P+1} = \underset{K}{\text{Argmax}} \left\{ \log \mathcal{L}_{K,P+1}(\hat{T}; \hat{\psi}) \right\},$$

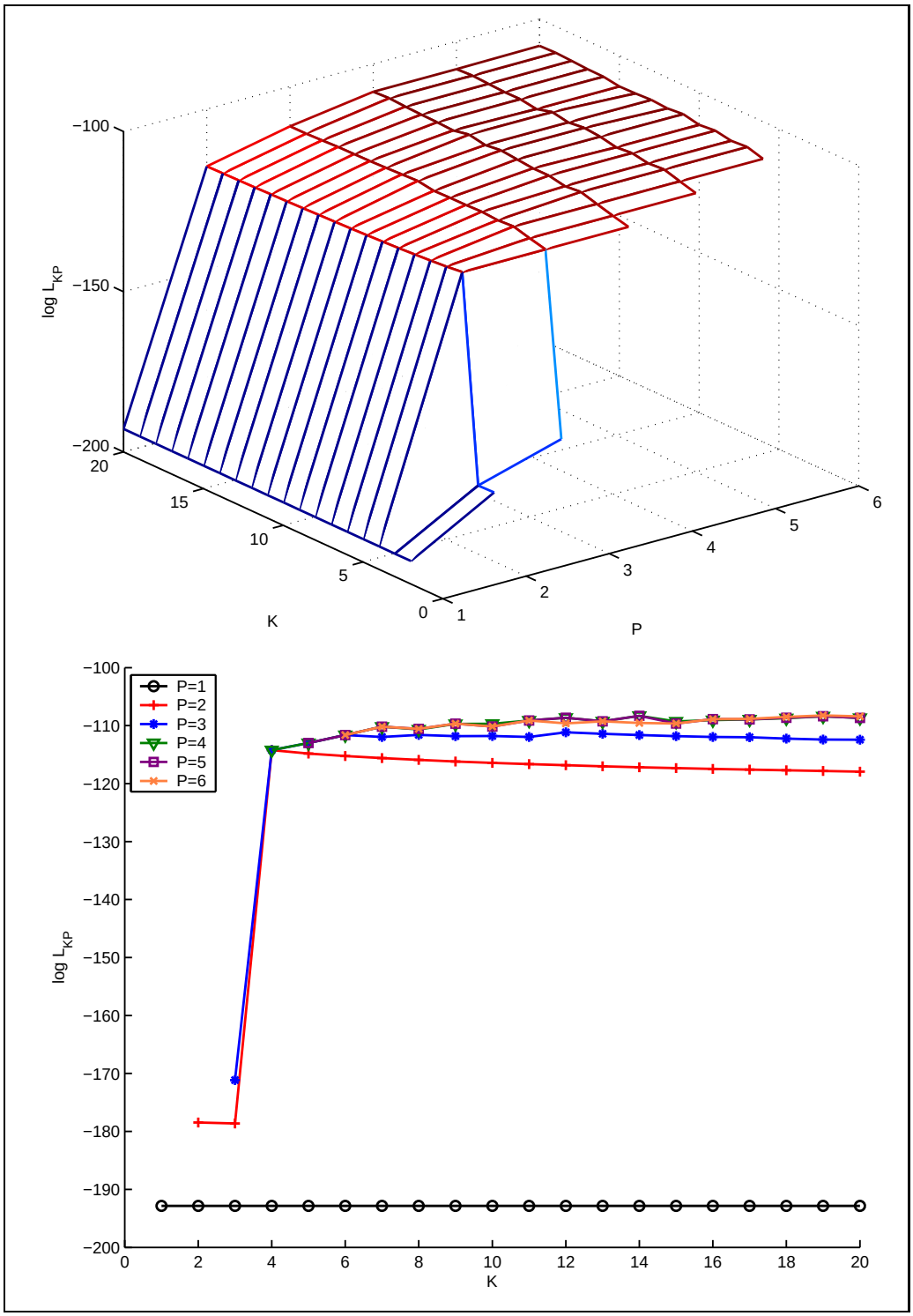


Figure 8.8: Two possible representations of the log-likelihood according to the number of clusters and to the number of segments.

we also have

$$\log \mathcal{L}_{\tilde{K}_P,P+1}(\hat{T},\hat{\psi}) \leq \log \mathcal{L}_{\tilde{K}_{P+1},P+1}(\hat{T},\hat{\psi}) \tag{8.4}$$

If we note $\log \widetilde{\mathcal{L}}_P$ the maximal log-likelihood for a segmentation/clustering model with P clusters:

$$\begin{aligned} \log \widetilde{\mathcal{L}}_P &= \max_K \left\{ \log \mathcal{L}_{KP}(\hat{T};\hat{\psi}) \right\}, \\ &= \log \mathcal{L}_{\tilde{K}_P,P}(\hat{T},\hat{\psi}), \end{aligned}$$

from Equations (8.3) and (8.4) we have:

$$\log \widetilde{\mathcal{L}}_1 \ldots \leq \log \widetilde{\mathcal{L}}_P \leq \ldots \log \widetilde{\mathcal{L}}_{P_{max}}.$$

A first strategy to select the number of clusters

An illustration of this sequence of increasing log-likelihoods is provided in Figure 8.9. $\log \widetilde{\mathcal{L}}_P$ can be interpreted as the maximal quality of fit that can be reached by a segmentation/clustering model with P clusters. Consequently the sequence of increasing log-likelihoods can be viewed as the target to be penalized in order to select the number of clusters whatever the number of segments.

An intuitive way to penalize this sequence of increasing log-likelihoods would be to use a *à la BIC* penalty, such as:

$$\hat{P} = \underset{P}{\text{Argmax}} \left\{ \log \widetilde{\mathcal{L}}_P - \frac{\nu_P}{2} \log(n) \right\}$$

with ν_P being the number of independent parameters of the mixture. This criterion is represented in Figure 8.10. In Chapter 7 we discussed the fact that the BIC approximation was not valid for classical mixture models, but since this penalty provides good results in practice, we choose to adapt it to our case. We also discussed the construction of ICL in the case of mixture models. Since this criterion has been shown to be adapted when the objective of the study is to cluster the data, we propose to discuss the adaptation of ICL to our case.

Adapting an ICL criterion to the case of segmentation/clustering

Since the Integrated Classification Likelihood has been shown to be efficient to select the number of clusters in the context of mixture models, we propose to adapt this criterion the the case of segmentation/clustering. The purpose of the ICL criterion is to calculate the *posterior* probability for each model given the complete data, $f(y,z|m_{PK})$, which is the complete-data integrated likelihood of model m_{PK} with P clusters and K segments. One particularity of our model is that the likelihood is not differentiable with respect to the breakpoint parameters. This is why we fix the breakpoints at T and we note $m_{PK}(T)$ this particular model. In this context, the complete-data integrated likelihood is

$$f(y,z \mid m_{PK}(T)) = \int_{\Psi_P} f(y,z \mid m_{PK}(T),\psi)\, h(\psi \mid m_{PK}(T))\, d\psi,$$

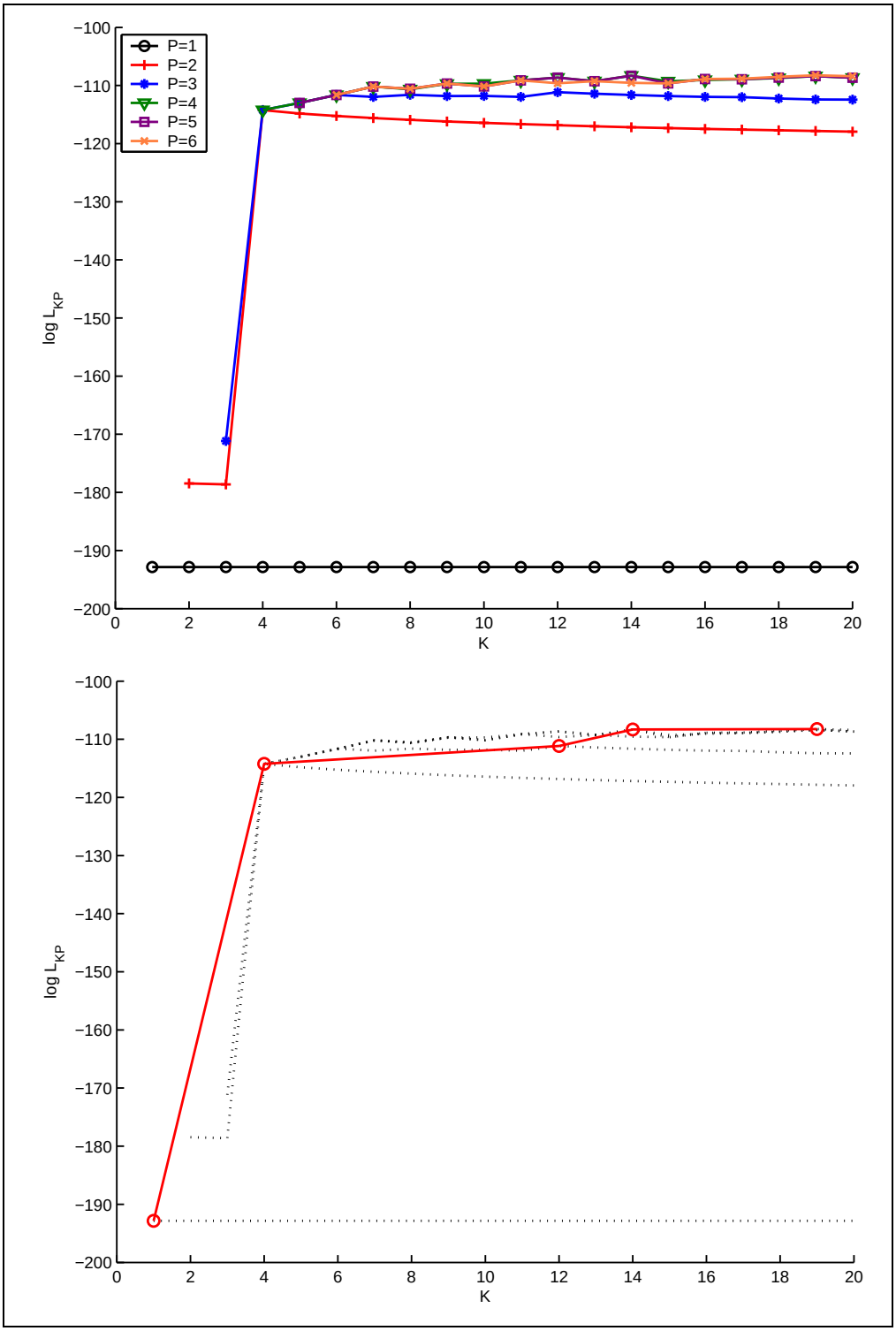


Figure 8.9: Representation of the sequence of increasing log-likelihoods $\left\{\log \widetilde{\mathcal{L}}_P\right\}$. Top: representation of the log-likelihoods according to the number of clusters and segments. Bottom: circles are used to illustrate the sequence of increasing log-likelihoods.

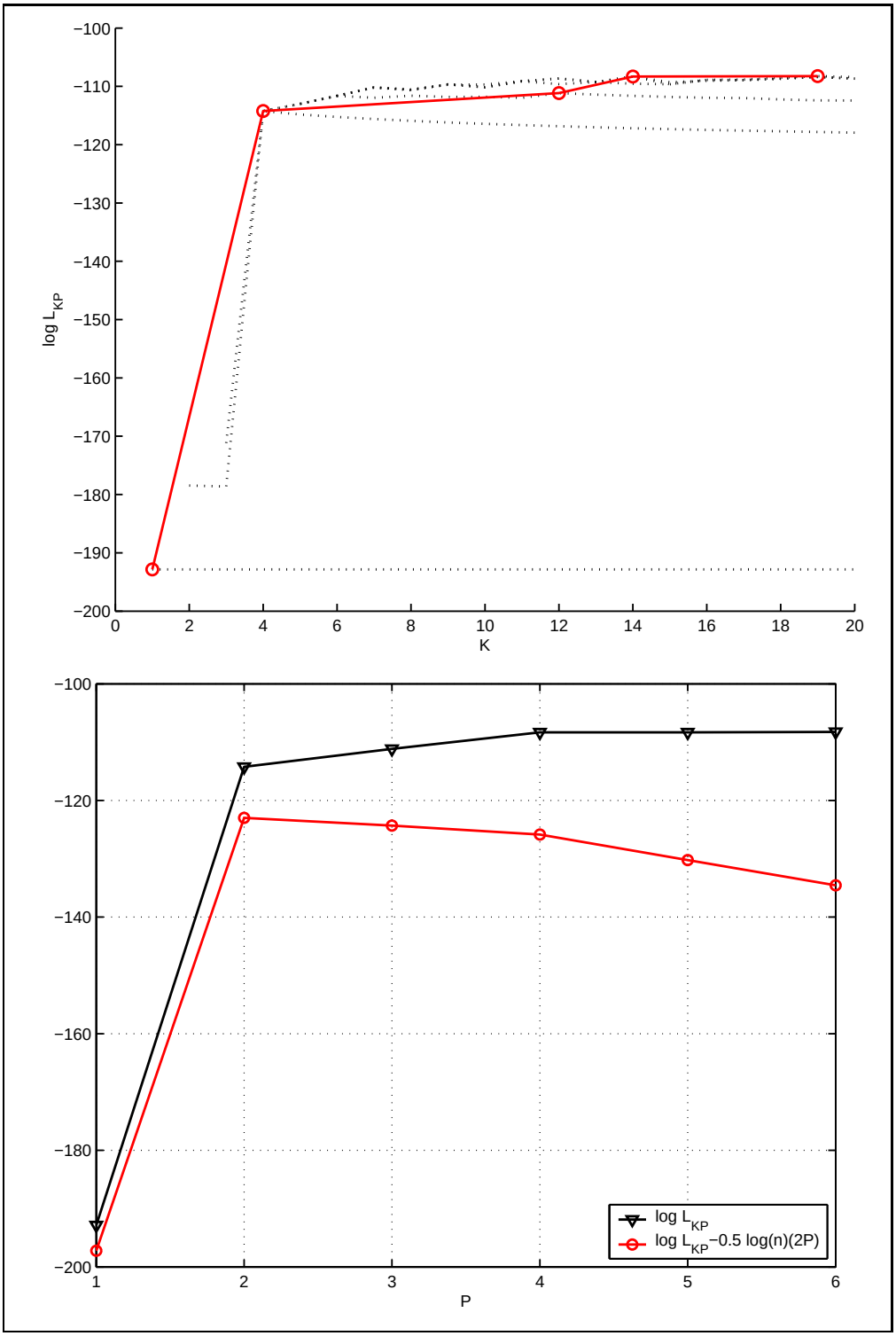


Figure 8.10: Top: Representation of the log-likelihoods according to the number of segments for varying numbers of clusters, and of the maximum log-likelihoods for a given number of cluster. Presentation of the maximum log-likelihoods according to the number of segments, penalized with a *à la* BIC penalization.

with

$$f(y, z \mid m_{PK}(T), \psi) = \prod_{k=1}^{K} \prod_{p=1}^{P} \left\{ \pi_p f(y^k; \theta_p) \right\}^{z_p^k}.$$

Ψ_P denotes the parameter space of model $m_{PK}(T)$, $\psi = (\theta, \pi)$ denotes the mixture parameters and $h(\psi|m_{PK}(T))$ a non-informative prior distribution on ψ for the same model. Since we can make the assumption that

$$h(\psi \mid m_{PK}(T)) = h(\theta \mid m_{PK}(T)) h(\pi \mid m_{PK}(T)),$$

we can apply the lemma proposed by Biernacki *et al.* (2003) to isolate the contribution of the missing data, such that:

$$f(y, z \mid m_{PK}(T)) = f(y \mid z, m_{PK}(T)) f(z \mid m_{PK}(T)).$$

When the breakpoint coordinates are fixed at T the BIC approximation is valid for $f(y \mid z, m_{PK}(T))$ which is approximated such that:

$$\log f(y \mid z, m_{PK}(T)) \simeq \max_{\theta} \log f(y \mid z, m_{PK}(T), \theta) - \frac{\lambda_P}{2} \log(n),$$

with λ_P the number of free parameters in θ.

As for the missing information, a direct calculus can be derived using a Dirichlet prior distribution for mixing proportions, noted $\mathcal{D}(\delta, \ldots, \delta)$. Parameter δ is set to $1/2$ to have a Jeffreys non informative distribution for mixing proportions. Then we introduce notation

$$k_p = \text{card}\{k,\ 1 \leq k \leq K \mid Z_p^k = 1\}\ (1 \leq p \leq P),$$

which corresponds to the number of segments belonging to cluster p. The missing-information integrated likelihood can be calculated such that:

$$\begin{aligned} f(z \mid m_{PK}(T)) &= \int \pi_1^{k_1} \ldots \pi_P^{k_P} \frac{\Gamma(P/2)}{\Gamma(1/2)^P} \mathbb{1}_{\sum_p \pi_p = 1} d\pi \\ &= \frac{\Gamma(P/2)}{\Gamma(1/2)^P} \frac{\Gamma(k_1 + 1/2) \ldots \Gamma(k_P + 1/2)}{\Gamma(K + P/2)}. \end{aligned}$$

A first remark is that the missing integrated likelihood does not depend on n, the number of data points. While calculating $f(z \mid m_{PK}(T))$, it appears that the objects to be clustered are segments, and not data points. This is why we introduced notation k_p instead of n_p which represents the number of data points within a cluster for the traditional ICL criterion. In the case of mixture models, the expression of $f(z \mid m_P)$ is simplified using an approximation of the Gamma function with the Stirling formula when $\tilde{n}_p$ is large. This leads to a simplification such that:

$$f(\tilde{z} \mid m_P) \simeq \sum_{p=1}^{P} \tilde{n}_p \log \frac{\tilde{n}_p}{n} - \frac{P-1}{2} \log(n).$$

Combining this missing-data integrated likelihood with the BIC approximation of $f(y|z, m_P)$ leads to the penalization of the complete-data likelihood with a term

$\nu_P \log(n)/2$, with ν_P the total number of free parameters in the mixture (these calculus have been detailed in the previous chapter on mixture models).

Nevertheless, it appears that the simplification of the missing-data integrated likelihood is not possible in our case, since $f(\tilde{z} \mid m_{PK}(T))$ does not depend on n, but only on K. It follows that term $\frac{P-1}{2}\log(n)$ does not appear in the penalization of the complete-data integrated likelihood, and this leads to a criterion which is not efficient in practice (it over-estimates the number of clusters systematically). This is why we choose to focus on a second alternative strategy to select the number of clusters.

A second strategy to select the number of clusters

A second strategy to select the number of clusters could be to apply the adaptive method proposed by Lavielle (1999). Applied to our problem, this method aims at finding the number of clusters for which the log-likelihood ceases to increase significantly. It is based on the calculus of the empirical second-derivative of the log-likelihood. Thus we propose a second strategy to select the number of clusters. Denoting $J_P = -\log \tilde{\mathcal{L}}_P$, the first step consists the calculus of $\widetilde{J}_P$ such that:

$$\widetilde{J}_P = \frac{J_{P_{max}} - J_P}{J_{P_{max}} - J_1} \times (P_{max} - 1) + 1.$$

This normalization step ensures that $\widetilde{J}_1 = P_{max}$ and that $\widetilde{J}_{P_{max}} = 1$. Then in a second step, calculate:

$$\forall P \in \{2, \ldots, P_{max} - 1\}, \quad D_P = \tilde{J}_{P-1} - 2\tilde{J}_P + \tilde{J}_{P+1}.$$

Then select the number of clusters, such that:

$$\hat{P} = \max_P \left\{ P \in \{2, \ldots, P_{max} - 1\} \mid D_P \geq s \right\},$$

with s a threshold to be determined in practice.

Adaptive strategies have been shown to be efficient in the segmentation context (Picard *et al.* (2005)) since they tend to ignore segments if their size is small of if the jump in the mean between two segments is small. If we transpose this behavior to the case of mixture models, we hope that the adaptive strategy will tend to ignore clusters if their distance is small in terms of parameters.

8.3.2 The problem of the null case

In any model selection procedure, it is crucial to determine what is the behavior of the criterion when there is nothing to detect. In our case, this means when there is no group and no segment. In previous sections we showed that the incomplete-data likelihood can decrease when the addition of new segments does not lead to an interesting clustering result. Consequently we could think that the log-likelihood is always decreasing in the null case. Interestingly this is not true.

In Figure 8.11 is showed the incomplete-data log-likelihood calculated for a sample of 100 data points with no group and no segment. It can be seen that this likelihood shows a similar behavior compared with other cases, meaning that it can show a maximum when P is fixed. In the case where there exists groups and segments, we showed that the likelihood was decreasing when two consecutive segments belong to the same cluster. In the null case, since the minimum size of segments is one point, the hybrid algorithm can lead to configurations where 2 consecutive points consitute 2 segments belonging to distinc clusters. This is why the likelihood increases even in the null case.

Since the likelihood is constant when $P = 1$, considering the sequence of maximum likelihoods $\log \widetilde{\mathcal{L}}_P$ will necessarily lead to an increasing sequence with an important jump between $\log \widetilde{\mathcal{L}}_1$ and $\log \widetilde{\mathcal{L}}_2$ as shown in Figure 8.11. The problem is that this situation can not be distinguished from the case where there exists clusters.

Since we need to find a way to select 1 cluster when there is no group, we need to find a way to circumvent this pitfall. To do so, we consider that when there is no segment there is no group. To this extent, we propose to check that there is no segment in the data first. In this case, the number of clusters will be 1, and otherwise, the heuristic to choose the number of clusters is applied. This step can easily be done with the segmentation method we developped in Chapter 5. Moreover, if this preliminary segmentation provides no segment, the hybrid algorithm does not need to be run, leading to a gain in computational time.

8.3.3 Selecting the number of segments

Once the number of clusters $\hat{P}$ has been selected, the objective is to select the number of segments $\hat{K}_{\hat{P}}$. In Section 8.1, we discussed two possibilities for this choice. When P has been selected we propose to apply one of those methods to select K. One is based on the choice of the number of segments for which the log-likelihood of the model ceases to increase significantly, and the other one is based on a BIC penalty. Once more, the adaptive strategy depends on a threshold that should be set in practice. This will be the purpose of the next part.

8.4 Interpretation and conclusion

In this chapter we discussed different model selection strategies to select the number of clusters and the number of segments in the context of segmentation/clustering. We proposed to adapt existing methods to the sequential selection of P and K in our case. Unfortunatelly we can not derive theoretical criteria for model selection as our model presents an unusual structure. This is why we propose a heuristic method for model selection.

The heuristic we propose for the selection of the number of clusters and segments can be interpreted as follows. Since the objective of our model is to cluster segments into a finite number of groups, we choose to select the number of groups

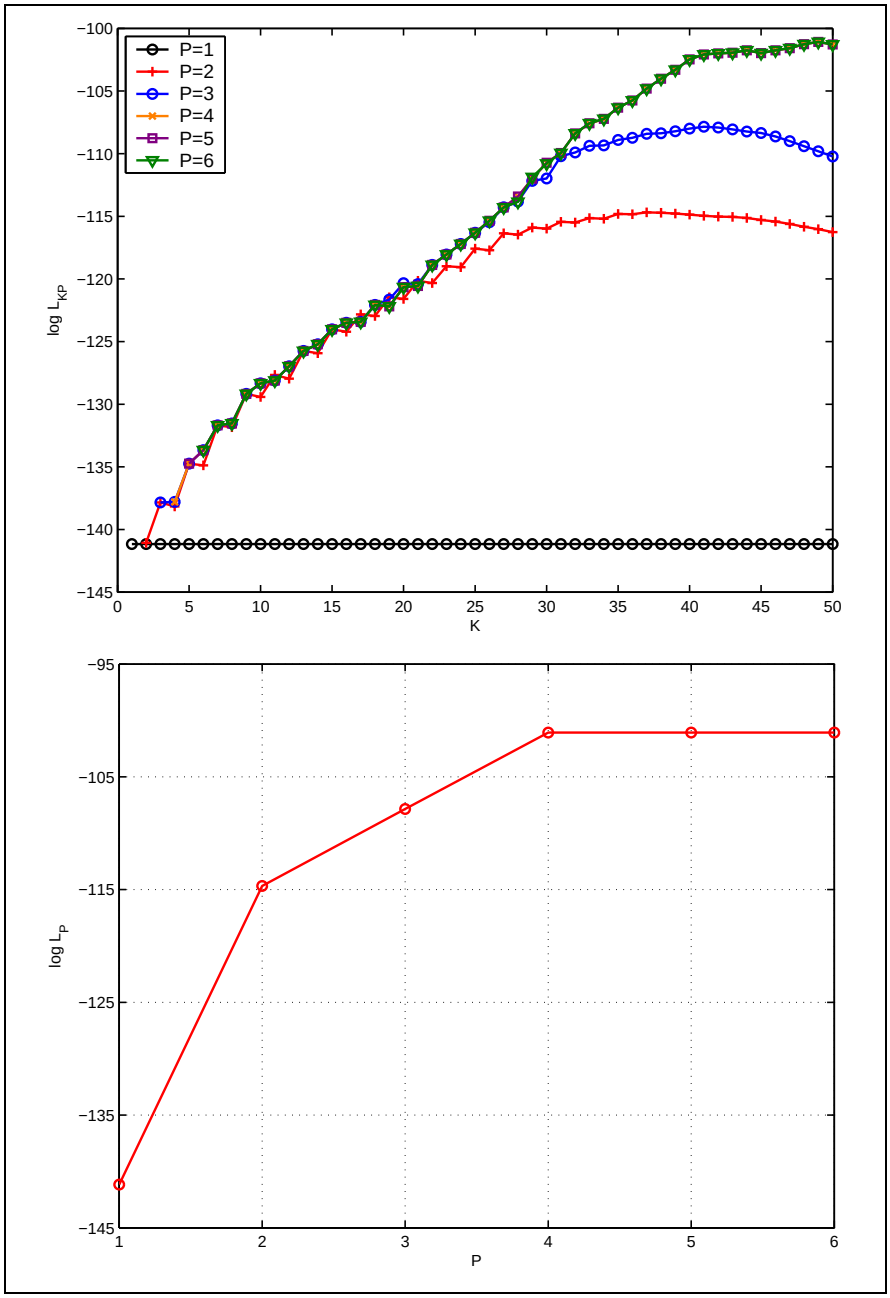


Figure 8.11: Top: Representation of the log-likelihoods according to the number of segments for varying numbers of clusters when there is no cluster and no segment. Bottom sequence of log-likelihoods $\log \widetilde{\mathcal{L}}_P$.

first and then to select the number of segments for a given number of groups. The advantage of the sequential procedure is that the first selection step appears to be "classical" since we penalize increasing log-likelihoods in order to select a model which is parcimonious regarding the number of parameters to be estimated. As for the second step, we propose two different methods which are empirically motivated.

As a consequence this method depends on some parameters that should be tuned in practice. This shows the difficulty to develop an automatic method when no theoretical result is available. In the following, a crucial question will be to choose among different methods, and to assess the performance of our heuristic. This is why the behavior of our procedure will be studied with simulation studies.

Part IV

Implementation of the clustering/segmentation method

Introduction

The previous part was dedicated to the definition of a new segmentation/clustering model. Since our objective is to apply this method to real data, the purpose of this part is to implement the method in order to provide a software program that can be used by biologists to analyze their data. We proceed in 4 steps.

Implementing the hybrid algorithm

We proposed a hybrid algorithm to estimate the parameters of the model when the number of clusters and segments are fixed. Since this algorithm is based on the EM algorithm, it faces usual pitfalls which are: the problem of the initialization step, and the problem of local maxima. In the first chapter we propose to explore different strategies to assess these problems. The problem of initialization is double in our case, since both breakpoints and mixture model parameters should be initialized. In the following, we propose to compare different initialization strategies, based on segmentation methods and on modified versions of the EM algorithm. We also propose a new method to initialize the EM algorithm based on a hierarchical clustering step. This method is used in the context of our algorithm, but it could be used in the more general setting of mixture models. The choice of initialization strategies is done using real CGH data sets in order to assess the performance of each method on average.

As we will see the hybrid algorithm faces many local maxima whatever the initialization step. This is why we propose a re-estimation step in order to avoid these local maxima. This method is based on the finding of parameter candidates which can be used to improve the likelihood of the model.

Model selection

Once the hybrid algorithm has been implemented, our objective is to assess the performance of the model selection heuristic we proposed. In the previous part we discussed different strategies that could be used. The purpose of the second chapter will be to choose among them. We propose to do so using a simulated data set which will be used in chapters 10 and 11. The principle of this simulation study is to consider factors of variation which can have an impact on the performance of the procedure. We choose to study the influence of two factors which are the separability of the mixture and the size of segments. We show that our model selection procedure is adaptive in the sense that it leads to the selection of parcimonious models when the separability of the mixture is low.

Assessing performance

In the previous part we compared our model with hidden Markov models which are widely used for segmentation/clustering problems. In Chapter 11 we propose to compare the performance of both methods. This is done using the simulated data set, in order to assess the ability of both method to correctly cluster the data, and to correctly locate the breakpoints. To do so, we propose to compare quality criteria which are the empirical error rate, the specificity and sensitivity of both methods. In this chapter we show that both methods are efficient, with a slight advantage for the segmentation/clustering model we propose.

Analysis of real data sets

The last chapter of this part will be dedicated to the application of our method to real CGH data sets. In a first step we propose to assess whether the segmentation/clustering model should consider homogeneous or heterogeneous variances, and we show that a homoscedastic model is suitable for real data sets. Unfortunatelly there does not exist public data sets for which the biological status of the clones have been confirmed by other methods. This is why we can not assess the ability of our method to find biologically relevant events. In chapter 12, we propose to compare the results of existing methods for array CGH data analysis, which are segmentation methods, and HMMs. We propose guidelines to interpret CGH results and give some perspectives regarding the analysis of such data.

Chapter 9

Initialization strategies for the hybrid algorithm

9.1 Initialization strategies, who is first?

The hybrid algorithm requires an appropriate initialization step. This step consists in the initialization of both breakpoint parameters $T^{(0)}$ and of mixture model parameters $\psi^{(0)}$ for a fixed number of segments and groups. Two strategies can be considered:

- Strategy 1 : initialize the breakpoint coordinates first based on a segmentation model, and deduce the parameters of the mixture model.
- Strategy 2 : initialize the mixture parameters first based on a mixture model on individual data points, and deduce the breakpoint coordinates.

However, it appears that Strategy 2 is not well adapted to real CGH data. Figure 9.1 illustrates the result of a mixture model based on individual data points for a real data set (the number of clusters is fixed and equals 3). The application of a classical mixture model leads to the creation of one cluster with high variance. Consequently putative deleted and amplified segments are clustered within the same group. The result of the downstream segmentation procedure is illustrated in Figure 9.2 (top). On the contrary, an initialization based on strategy 1 directly provides segments which are clustered in different groups which are highly separable (Figure 9.2, bottom). These examples show that Strategy 1 helps to recover the clustered structure of segments. This is why we choose this strategy.

9.2 Initializing breakpoint coordinates

The proposal of breakpoint coordinates $T^{(0)}$ can be done using standard segmentation techniques as described in Part II. In this chapter we consider a mixture model with heterogeneous variances. Therefore the natural model that is considered for the initialization of the breakpoints is a segmentation model in the Gaussian framework, where the parameters that are affected by the changes are the

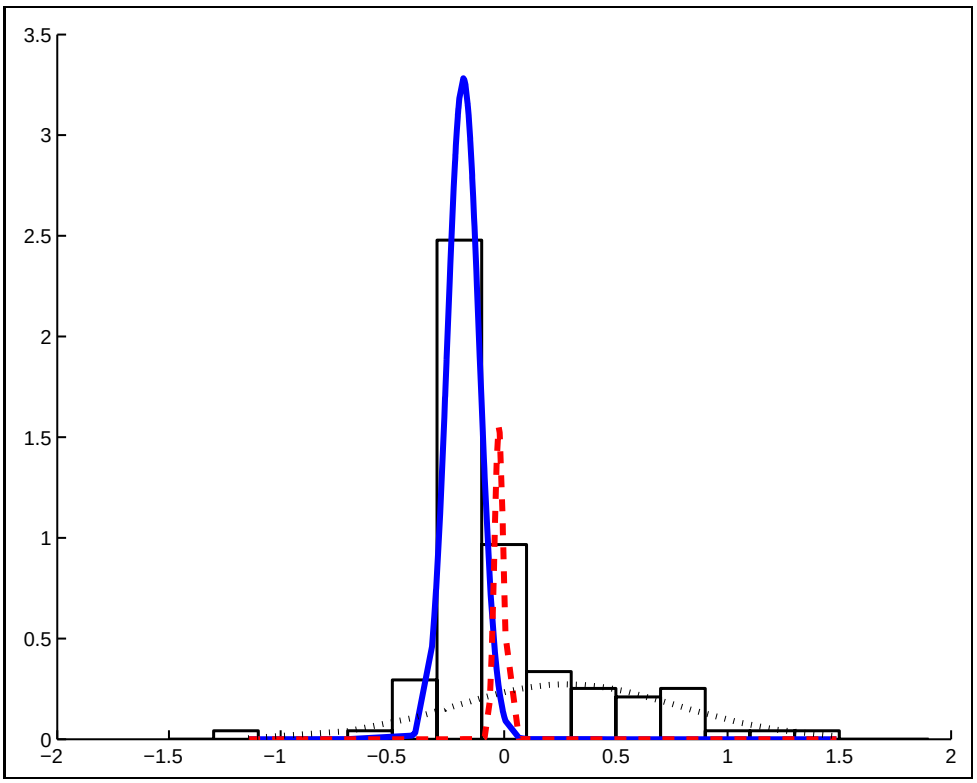


Figure 9.1: histogram of $\log_2$ ratios for Bt474 chromosome 1, and estimated densities of a mixture model with $P = 3$ groups. The estimated parameters are $\hat{m} = \{-0.18, -0.02, 0.28\}$, $\hat{s} = \{0.07, 0.02, 0.50\}$ and $\hat{\pi} = \{0.58, 0.08, 0.34\}$.

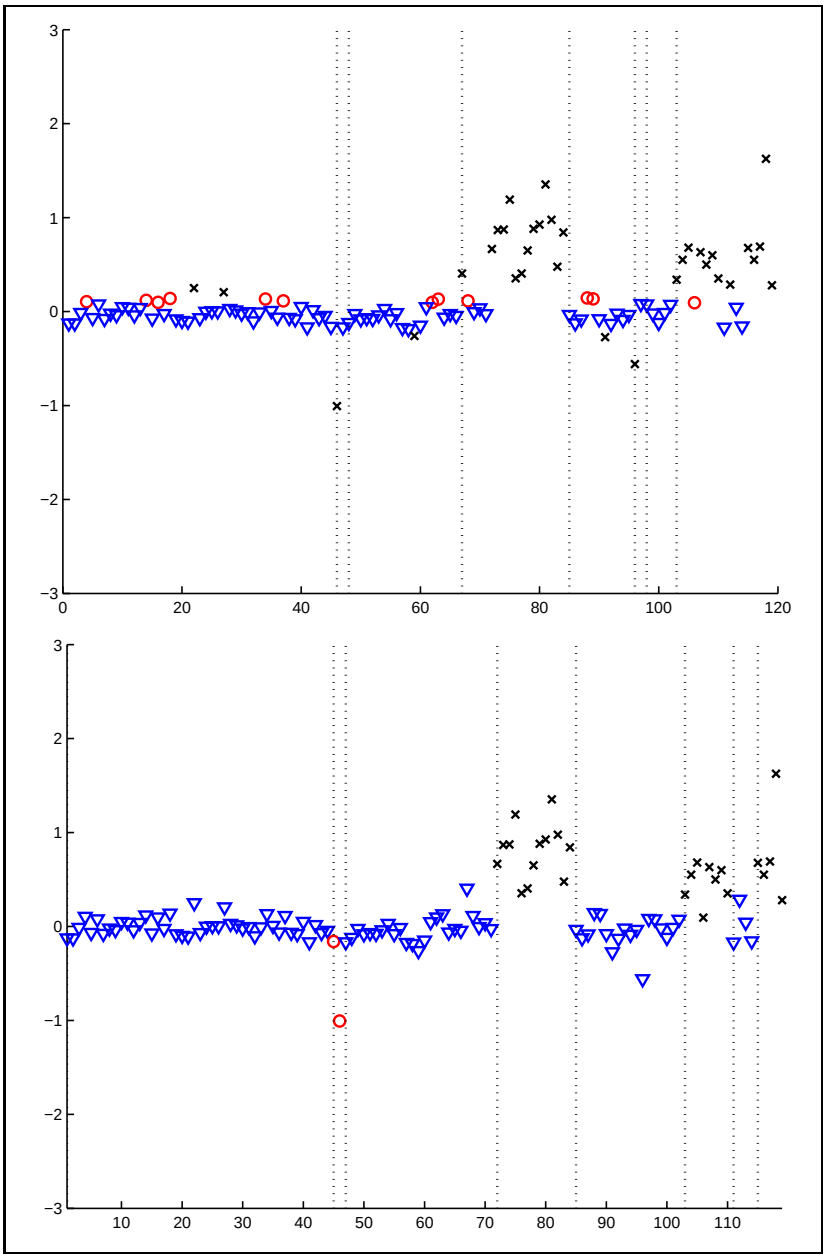


Figure 9.2: Comparison of two initialization strategies for data Bt474, chromosome 1, for $P = 3$ and $K = 8$. In the first case (top) the mixture model parameters have been estimated first, and then a segmentation was used to estimate the breakpoint coordinates. The estimated parameters of the mixture are $\hat{m} = \{-0.18, -0.02, 0.28\}$, $\hat{s} = \{0.07, 0.02, 0.50\}$ and $\hat{\pi} = \{0.58, 0.08, 0.34\}$. In the second case (bottom), breakpoint coordinates have been estimated first using a segmentation method, and the resulting segments have been clustered into three groups using hierarchical clustering. The estimated parameters of the mixture are $\hat{m} = \{-0.72, -0.16, 0.55\}$ and $\hat{s} = \{0.42, 0.12, 0.43\}$ and $\hat{\pi} = \{0.12, 0.50, 0.38\}$.

mean and the variance (model $\mathcal{M}_1$). Nevertheless Picard *et al.* (2005) have shown that a model with homogeneous variance (model $\mathcal{M}_2$) was more appropriate for array CGH data. This is why we choose to try both preliminary segmentations.

9.3 Initializing mixture model parameters

The proposal of a mixture model parameter candidate $\psi^{(0)}$ is done once the breakpoints have been initialized. We propose three strategies for this step.

9.3.1 Hierarchical clustering

The first strategy is to cluster the previously defined segments using a hierarchical clustering method. Agglomerative hierarchical clustering is a stepwise procedure in which pairs of clusters are successively merged. In any hierarchical clustering, one needs to define an objective criterion to optimize, a distance between groups, and a rule to merge clusters within the same group. The most popular methods are based on geometric considerations, where the objective criterion to optimize can be the within-class variability for instance (Ward (1963)). Nevertheless our objective is to provide a good candidate for mixture model parameters. This is why we choose to develop a hierarchical clustering algorithm in the context of model-based clustering, using the classification likelihood as a criterion.

Let us present the classical algorithm of a hierachical clustering when having n objects to cluster into P groups.

- *Initialization:*
  - $\forall (k,\ell) \in \{1,\dots,n\}^2$ *compute* $d_{ind}(k,\ell)$ *the distance between individuals* k *and* ℓ.
  - *Set* $h = n$ *the initial number of clusters.*
- *Repeat:*
  - *merge clusters* C_i *and* C_j *if* $d_{clust}(i,j) = \min_{k\ell} \{d_{clust}(k,\ell)\}$,
  - $h = h - 1$,
  - $\forall (k,\ell) \in \{1,\dots,h\}^2$ *update distances* $d(k,\ell)$.
- *Stopping rule:* $h = P$.

In our context, the data to be clustered are K segments $\{Y^1,\dots,Y^K\}$ which have been defined by the first segmentation step. We note $T^{(0)}$ the initial breakpoint coordinates. Let us consider a Gaussian mixture model with P clusters such that:

$$Y^k \in C_p \sim \mathcal{N}(\mu_p, \sigma_p^2), \text{ with } \theta_p = (\mu_p, \sigma_p^2).$$

We define the classification log-likelihood of this model:

$$CL_P(Y;\theta) = \sum_{p=1}^{P} \sum_{Y^k \in C_p} \log f(y^k;\theta_p). \tag{9.1}$$

At iteration (h) of the hierarchical clustering, the segments are clustered into h groups denoted $C_1^{(h)}, \ldots, C_h^{(h)}$, with parameters $\theta_1, \ldots, \theta_h$. Replacing θ_p by its maximum likelihood estimator, the classification log-likelihood at this step is:

$$\widehat{CL}_h(Y; \hat{\theta}) = -\sum_{p=1}^{h} n_p \log \hat{\sigma}_p^2,$$

with n_p being the number of segments in cluster p. At iteration $(h+1)$, suppose that clusters $C_i^{(h)}$ and $C_j^{(h)}$ are merged, noting σ_{ij}^2 the variance of cluster $\{C_i^{(h)} \cup C_j^{(h)}\}$. The classification log-likelihood at its maximum is:

$$\widehat{CL}_{h-1}(Y; \hat{\theta} | C_i^{(h)} \cup C_j^{(h)}) = -\sum_{p=1}^{h-2} n_p \log \hat{\sigma}_p^2 - (n_i + n_j) \log \hat{\sigma}_{ij}^2.$$

It follows that:

$$\widehat{CL}_{h-1}(C_i^{(h)} \cup C_j^{(h)}) - \widehat{CL}_h(C_i^{(h)}, C_j^{(h)}) = -(n_i + n_j) \log \hat{\sigma}_{ij}^2 + n_i \log \hat{\sigma}_i^2 + n_j \log \hat{\sigma}_j^2.$$

Our objective being the maximization of the classification log-likelihood at each step, we define the distance between two clusters, such that

$$d(i,j) = (n_i + n_j) \log \hat{\sigma}_{ij}^2 - n_i \log \hat{\sigma}_i^2 - n_j \log \hat{\sigma}_j^2.$$

Two clusters will be merged if their distance is the smallest among all possible pairwise distances.

Proposition

Let us denote C_i, C_j two clusters such that $\#C_i = n_i$, $\#C_j = n_j$ and $n_i, n_j \geq 2$. Consider $d(i,j)$ the distance between clusters such that:

$$d(i,j) = (n_i + n_j) \log \hat{\sigma}_{ij}^2 - n_i \log \hat{\sigma}_i^2 - n_j \log \hat{\sigma}_j^2.$$

Then the classification log-likelihood defined in Equation (9.1) is locally maximized at each iteration of the hiearchical clustering algorithm.

Proof

The proof of this proposition is straighforward since we chose the distance such that:

$$\widehat{CL}_{h-1}(C_i^{(h)} \cup C_j^{(h)}) - \widehat{CL}_h(C_i^{(h)}, C_j^{(h)}) = -d(i,j).$$

To this extent, merging two clusters with minimal distance $d(i,j)$ ensures that $\widehat{CL}_{h-1}(C_i^{(h)} \cup C_j^{(h)})$ is maximal at iteration $(h+1)$.

Comments

- First of all, it is important to notice that the optimization is local, meaning that this algorithm does not provide an optimal solution.

- The algorithm is repeated until a fixed number of groups, and the resulting mean, variance and proportion of each group constitute the parameters $\psi^{(0)}$ used to initialize the hybrid algorithm. This strategy is thought to provide a reasonnable candidate regarding the downstream mixture model.

- This step requires the ability to calculate a variance at the first step (when each segment is in its own group). In our case, we constraint our segmentation procedure to provide segments with a minimum size of 2.

9.3.2 Stochastic strategies

Stochastic version of EM

A second strategy that can be considered to initialize the mixture model parameters is motivated by the discussion concerning the EM algorithm given in Part III. There is a strong link between the initialization step of the EM algorithm and its tendency to converge to local maxima. This is why we propose to use the stochastic version of the EM algorithm (SEM) to initialize the mixture model parameters $\psi^{(0)}$ once the breakpoint coordinates $T^{(0)}$ have been proposed. This algorithm is known to avoid spurious maximizers, compared with the hierarchical clustering step which locally maximizes the likelihood.

Short EM

The last strategy which is considered has been suggested by Biernacki *et al.* (2003). Instead of running the stochastic version of the EM algorithm which may require many iterations and may be slow to converge, the authors suggest to use short runs of the EM algorithm with random starts, and to choose among the best proposed candidates. By short runs of the EM algorithm the authors mean that the algorithm is stopped before its convergence. This strategy can be viewed as a modified version of the SEM algorithm for which the most important iterations are the first ones.

9.4 Choice of an initialization strategy based on real data sets

We have proposed six initialization strategies for the hybrid algorithm: two for the breakpoint coordinates, and three for the mixture models parameters. The question is to choose among them.

Definition of quality criteria to choose among strageties

Five different criteria are defined for this purpose. The purpose of the initialization step is to provide good candidates $(T^{(0)}, \psi^{(0)})$ to start the hybrid al-

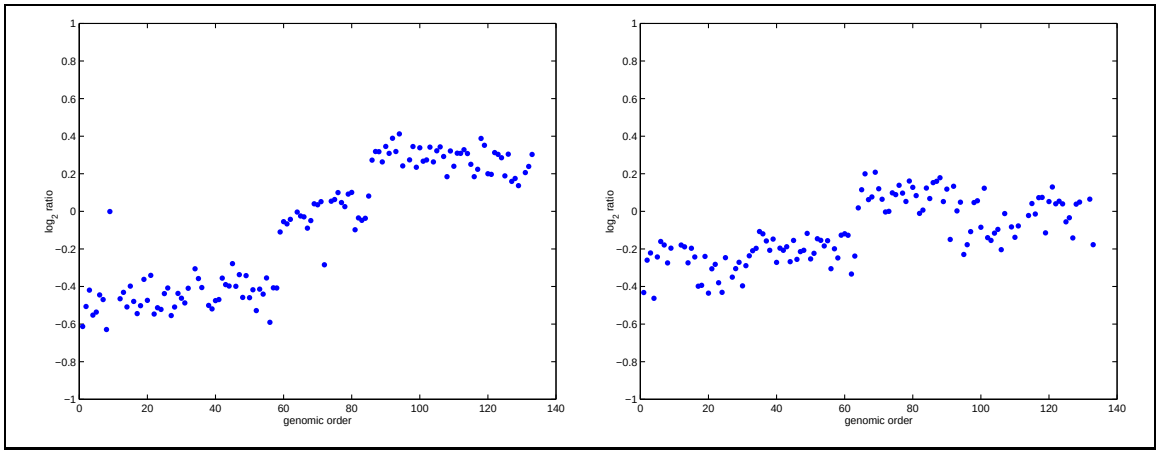


Figure 9.3: Example of CGH profile for real data sets described in Nakao *et al.* (2004). These profiles correspond to chromosome 8 for two different patients.

gorithm. The first natural criterion that is used to score the different strategies is the incomplete-data log-likelihood of the model (`Linc0`) calculated at $T = T^{(0)}$ and $\psi = \psi^{(0)}$. The second criterion is the incomplete-data log-likelihood of the model (`Linc`) after convergence of the hybrid algorithm. The best initialization method should provide the best likelihood. Other criteria are used to assess the stability of the algorithm for a given initialization method as shown in Table 9.4.

`Linc0`	best candidate
`Linc`	best fit (final)
`time`	computational time
`empty_clust`	create empty clusters
`conv_max`	converge in more that 5000 iterations

Table 9.1: Criteria used to assess the best initialization strategy.

Presentation of the data set

The data we use to choose among different initialization strategies have been described in Nakao *et al.* (2004). We have CGH profiles for 125 patients, and for each patient we consider chromosomes 1, 8 and 20. Finally we have 375 profiles, each being of size 100 points. Figure 9.4 gives an example of such profiles. The hybrid algorithm is run for each profile, with different numbers of groups ($P = 1, \ldots, 6$) and segments ($K = P, \ldots, 20$).

Analyzing the results with a linear model

We have 5 different criteria, each being the result of the optimization procedure based on the 6 initialization strategies. Let us consider the incomplete-data log-likelihood (`Linc`) for instance. For each K and P the hybrid algorithm is run

using 6 different initialization strategies. In order to determine if one strategy systematically provides the best fit, we rank these log-likelihoods for each P and K. Then the purpose is to determine which initialization strategy affects this rank. To do so we use a linear model as follows. We note α_i the effect of the initialization strategy for the breakpoints ($i = 1, 2$) and β_j for the mixture ($j = 1, 3$). Denoting R_{ij} the rank of the incomplete-data log-likelihood for a mixture model which has been initializee with strategy i, j the model is written as follows:

$$R_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + E_{ijk} \text{ with } E_{ijk} \sim \mathcal{N}(0, \sigma^2)^1.$$

Table 9.4 gives the results of the linear model performed on the ranks of the final log-likelihood. The highest rank indicates that an initialization method provides the best fit of the model to the data after convergence of the algorithm. We note that the first factor of variation is the method of initialization of the breakpoints (F-value=8316.47), meaning that the initialization of the segmentation is crucial to reach the best fit of the model to the data. When adjusted to other effects the results suggest that the best initialization procedure for the breakpoints is model $\mathcal{M}_1$ which considers heterogeneous means and variances. This result is in accordance with the downstream mixture model which considers heterogeneous variances.

The method of initialization for the mixture parameters is the second factor of variation in terms of F-value (3652.61), and testing LSMEANS to compare the adjusted effects indicates that the strategy based on short random EM (`rEM`) provides the best log-likelihood. Results are similar if the criterion is the initial log-likelihood (`Linc0`), meaning that `rEM` provides the best mixture model candidate on average.

Other effects have been added to check that the performance of one initialization procedure did not depend on the complexity of the CGH profiles under study (Table 9.4). Over 125 patients, some may have deletions or amplifications on one chromosome, but not on the others (3 chromosomes total), leading to more or less complex CGH profiles. Since SEM is known to avoid local maximizers, its performance could be better in more complex situations, compared with the hierarchical method. Nevertheless the results suggest that these effects have only a limited impact on the performances of the initialization procedures (small F-values).

Stability and computational time

In addition to the performance of each initialization procedure, we check their stability *i.e.* the tendency of each method to generate empty clusters. Table 9.4 shows that 80% of the empty clusters were created by the `rEM` strategy, meaning that this initialization method leads to a very unstable algorithm, whereas the hierarchical method is very stable. This tendency can be explained by the random starts of the short EM algorithms, compared with the reasonable candidate proposed by the hierarchical clustering strategy. As for the tendency to converge

[1]Note that the Gaussian and the independence assumptions for residuals are not valid in this case. Nevertheless, we are focused on the estimation of the average effects. This is why we consider that these hypothesis are not crucial in this context.

in more than 5000 iterations (criterion `conv_max`), the impact of the different strategies is similar between initialization methods (data not shown).

The last criterion that has been studied is the computational time required by each initialization method. A linear model is considered, with the rank of the cpu-time as the variable to be explained and the different methods of initialization as factors of variation. Table 9.4 clearly shows that the method of initialization of the mixture parameters is the greatest factor of variation. This can be explained be the fact that estimating mixture model paramaters requires iterative methods, and the Hierarchical method is the fastest.

Results suggest that there is no best initialization method according to the different criteria we have studied. The method based on `rEM` provides best candidates, but often leads to the creation of empty clusters. Since our ojective is to automatically apply our algorithm to real data sets, it is of crucial importance to propose a stable estimation algorithm. Combined with the fact that the hierarchical method is faster, we decide to use this initialization method for the mixture parameters with a segmentation based on model $\mathcal{M}_1$ for the initialization of the breakpoints.

Source	DF	Mean Square	F Value	Pr > F
mixt	2	5294.71823	3652.61	<.0001
break	1	12055.30286	8316.47	<.0001
mixt*break	2	1124.57859	775.80	<.0001
patient(chromosome)	374	1.16670	0.80	0.9977
mixt*patien(chromos)	748	5.38304	3.71	<.0001
break*patien(chromo)	374	60.84263	41.97	<.0001
mixt*brea*pati(chro)	748	2.65889	1.83	<.0001

mixt	rLinc LSMEAN
CAH	3.12452951
SEM	3.01368774
rEM	3.36784091

break	rLinc LSMEAN
M1	3.31719347
M2	3.02017864

Table 9.2: Results of the linear model on the ranks of the incomplete-data likelihood. Main effects concern different strategies of initialization (for breakpoints and for mixture parameters). The comparisons of LSMEANS are all significantly different from zero (tests not shown)

Percent	CAH	SEM	rEM	Total
M1	3.68	5.50	40.04	49.21
M2	4.39	6.20	40.19	50.79
Total	8.07	11.70	80.24	100.00

Table 9.3: Cross table to study the number of times that an initialization procedure creates empty clusters.

Source	DF	Mean Square	F Value	Pr > F
mixt	2	20435.07855	6661.91	<.0001
break	1	188.63001	61.49	<.0001
mixt*break	2	151.17714	49.28	<.0001
patient(chromosome)	374	70.08549	22.85	<.0001
mixt*patien(chromos)	748	42.36919	13.81	<.0001
break*patien(chromo)	374	24.13899	7.87	<.0001
mixt*brea*pati(chro)	748	4.36250	1.42	<.0001

mixt	time LSMEAN
CAH	0.98443656
SEM	1.37154979
rEM	1.66166537

break	time LSMEAN
M1	1.35779388
M2	1.32064060

Table 9.4: Result of the linear model to study the impact of initialization procedures on the computational time of the algorithm.

9.5 Avoiding local maxima

Despite an appropriate choice of starting values for the hybrid algorithm it appears that the resulting log-likelihood can present a chaotic behavior when the number of segments increases, as presented in Figure 9.4 (dot lines). Considering the two examples, it is clear that the hybrid algorithm faces many local maxima. A first reason for this would be that the initialization strategy is inappropriate, despite the previous study. For instance, SEM could be thought to prevent those instabilities. Nevertheless, this unstable behavior is observed whatever the initialization method. Consequently, we need to find a method in order to stabilize the estimation results.

Our hypothesis is that the log-likelihood should be smooth when the number of segments increases. Let us consider the case for which the algorithm faces a local maximum when the number of clusters is P and the number of segments is K. Our aim is to find a way to increase the log-likelihood $\log \mathcal{L}_{KP}(\hat{T}, \hat{\psi})$ using new starting values. Then we consider neighboring configurations with say

K_i and K_{i+1} segments ($K_i \leq K \leq K_{i+1}$) for which the likelihoods are greater than $\log \mathcal{L}_{KP}(\hat{T}, \hat{\psi})$. Corresponding mixture parameters estimators $\hat{\psi}(\hat{T}_{K_i})$ and $\hat{\psi}(\hat{T}_{K_{i+1}})$ could be used to initialize the hybrid algorithm for K segments. If the new likelihood is greater, then we keep it as well as the resulting parameters $\hat{T}_K$ and $\hat{\psi}(\hat{T}_K)$, and it is not changed otherwise. Then the problem is to determine the neighboring configurations that will be used to re-initialize the hybrid algorithm. Our strategy is to consider the "best" neighboring log-likelihoods to provide new starting values. Consequently, these configurations are determined calculating the convex hull of the log-likelihood. In the following we present the re-estimation procedure denoting L_{KP} for $\log \widehat{\mathcal{L}}_{KP}$.

- Repeat
  - Find $\{(L^{(h)}_{K_iP}, K^{(h)}_i), i \geq 1\}$ the convex hull of the set $\{(L^{(h)}_{KP}, K), K \geq P\}$.
  - $\forall K \in]K^{(h)}_i, K^{(h)}_{i+1}[$, calculate

$$\begin{cases} L^{[i]}_{KP} & \text{using} \quad \psi^{(0)} = \hat{\psi}(\hat{T}_{K^{(h)}_i}), \\ L^{[i+1]}_{KP} & \text{using} \quad \psi^{(0)} = \hat{\psi}(\hat{T}_{K^{(h)}_{i+1}}). \end{cases}$$

  - Update the log-likelihood such that:

$$L^{(h+1)}_{KP} = \max \left\{ L^{(h)}_{KP}, L^{[i]}_{KP}, L^{[i+1]}_{KP} \right\}.$$

  - Update parameters $\psi^{(h)}$ and $T^{(h)}_K$ consequently.
- Stopping rule : when $L^{(h+1)}_{KP} = L^{(h)}_{KP}$.

Figure 9.4 shows two examples of the re-estimation results for real data sets (described in Nakao *et al.* (2004)) and for $P = 2$ and $P = 3$ clusters. The log-likelihood before and after re-estimation are represented according to the number of segments (dot and plain lines respectively). It can be seen that the result is spectacular since the log-likelihood can be reconstructed completely. This stabilization step appears crucial to have better parameter estimates, but also for the downstream model selection heuristic which is based on the geometrical behavior of the likelihood. Nevertheless, one major draw-back is that the associated computational time is increased due to the re-estimation procedure.

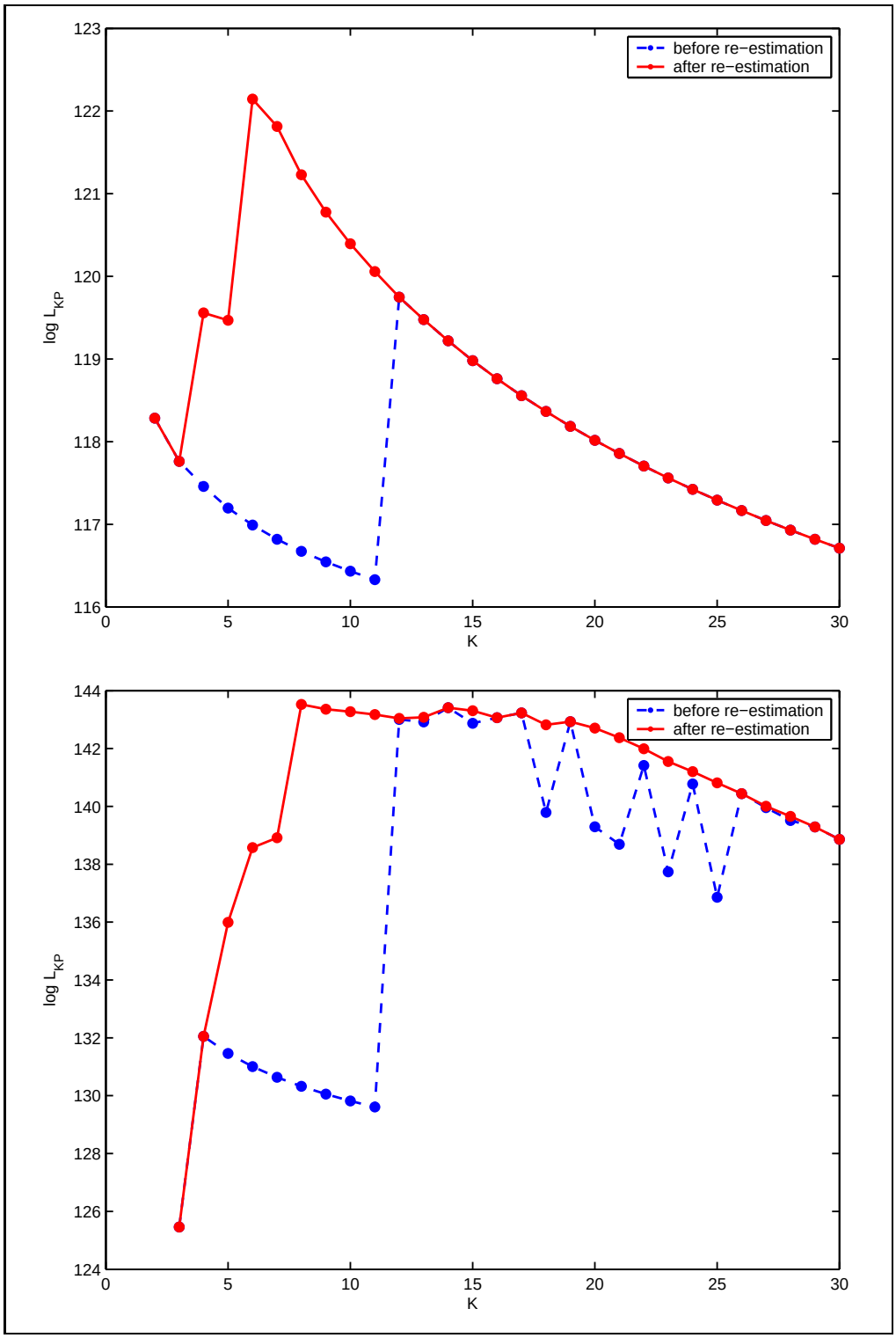


Figure 9.4: Example of re-estimation and reconstruction of the log-likelihood. Data are described in Nakao *et al.* (2004) and correspond to experiment X411 chromosome 1 for $P = 2$ (top) and $P = 3$ clusters (bottom). Log-likelihoods are represented in dot lines before re-estimation and plain lines after re-estimation.

Chapter 10

Behavior of the model selection heuristic

Now that the hybrid algorithm has been implemented when the number of segments and the number of clusters are fixed, the next step consists in the selection of P and K. In Chapter 8 we proposed a heuristic to select the number of segments and the number of clusters. Nevertheless the heuristic we propose is empirically motivated and its performance should be addressed. To do so we propose to study the performance of our method based on a simulation study.

10.1 Design and objectives of the simulation study

In order to study our selection procedure, we list the possible factors of variation that can have an impact on its performance. These factors can be listed as follows:

- the size of the data set.
- The number of segments, and the number of groups, which reflect the complexity of the configurations.
- The size of segments: segments of large size are easier to detect.
- The "detectability" of breakpoints, which depends on the normalized mean difference between two segments, and on the size of segments.
- The "separability" of the mixture, which is defined as the closeness of two groups in terms of parameters.

It is clear that many factors of variation are linked when the size of the data is fixed. For instance, the "separability" of the mixture is linked to the number of clusters, and the "detectability" of breakpoints is linked to the number of segments. Moreover the detectability of the breakpoints is also linked to the jump in the mean between two segments. Therefore it is linked to the separability of the mixture. Since all factors can not be crossed, we choose to fix some of them.

A typical CGH profile for one chromosome is constituted of 100 data points approximatively, and we choose to fix the size of our simulations at 100 points.

Then we decide to fix the number of clusters at 3 and the number of segments at 5. We focus on two major factors of variation, which are the "separability" of the mixture and the "detectability" of breakpoints.

Separability of the mixture model

In the previous part, we proposed two methods to select the number of clusters with penalized criteria. One is based on a *à la* BIC penalization of the log-likelihood and the other one is based on an adaptive strategy. Selecting the number of clusters depends on the separability of the groups, meaning that if two clusters are close with respect to their parameters, a parcimonious method would choose one cluster instead of two. In the simulation study, we propose to fix the parameters of two clusters, and to vary the parameters of the third one. The distance between clusters p and q is calculated such that:

$$d_{pq} = \frac{|m_p - m_q|}{\sqrt{s_p^2 + s_q^2}}.$$

In the simulation study, we propose to fix the means of clusters 1 and 2 and to decrease the mean of the third one in order to decrease the distance between clusters. The variances of the three groups are different but constant. Parameters are chosen as shown in Table 10.1. An illustration of two configurations is given in Figure 10.1 when $d = 2$ (top) and $d = 0$ (bottom).

	cluster 1	cluster 2	cluster 3
m	0	-5	varying
s^2	1	2	3
π	2/5	1/5	2/5

m_3	4	3	2	1	0
d	2	1.5	1	0.5	0

Table 10.1: Varying distances between clusters for the simulation study.

Detectability of breakpoints

As for the selection of the number of segments, we also proposed two methods. One is based on a *à la* BIC penalization and the other one is based on an adaptive strategy. As it is the case for the selection of the number of clusters, some segments may not be detected if their size is small. This is why we choose to fix the size of two segments at 20 points, and to vary the size of the other segments as detailed in Table 10.2. The interest lies in cluster 3, which is composed of two segments, one being of constant size ($n_5 = 20$) and the other being of increasing size ($n_2 = 2, \ldots, 20$). An illustration of two configurations is given in Figure 10.1 when $n_2 = 20$ (left) and $n_2 = 2$ (right).

Objectives of the simulation study

The design of the simulation study is a factorial design, with factors being d the distance between clusters 1 and 3 (5 levels) and the size of segments (5 levels).

Size	n_1	n_2	n_3	n_4	n_5
cluster label	1	3	1	2	3
	30	2	28	20	20
	28	5	27	20	20
	25	10	25	20	20
	23	15	22	20	20
	20	20	20	20	20

Table 10.2: Varying sizes of segments for the simulation study.

When the distance between clusters 1 and 3 is small (Figure 10.1 (bottom)), the question will be to determine if our selection procedure tends to select 2 clusters rather than 3. For the selection of the number of segments, the question will be to determine if the selection procedure will tend to ignore segments of small size, as shown in the segmentation context (Chapter 5).

Illustrations

Four examples of simulations are provided in Figure 10.1. The first case (top left) is supposed to be the "easy" configuration, where clusters are well separated and where segments have large size. In this configuration, we hope that the selection procedure will select the correct number of clusters and segments. A second situation is illustrated (top right), where clusters are well separated, but one cluster has a segment of small size. In this situation, the question will be to assess the ability of the selection procedure to detect segments of small size when the separability of the mixture is high. The two other situations are illustrated in Figure 10.1, bottom. In these cases, clusters 1 and 3 have the same mean, but different variances. Since clusters are not well separated, the question will be the selection of the number of clusters.

Each configuration has been simulated 100 times and the estimation algorithm has been run using the initialization method chosen in the previous chapter as well as the re-estimation procedure. Note that in this chapter we are focused on the estimation of the *number* of clusters and segments. This means that even if the "correct" numbers are estimated, it does not imply that the breakpoints have been correctly located neither that the mixture parameters have been correctly estimated.

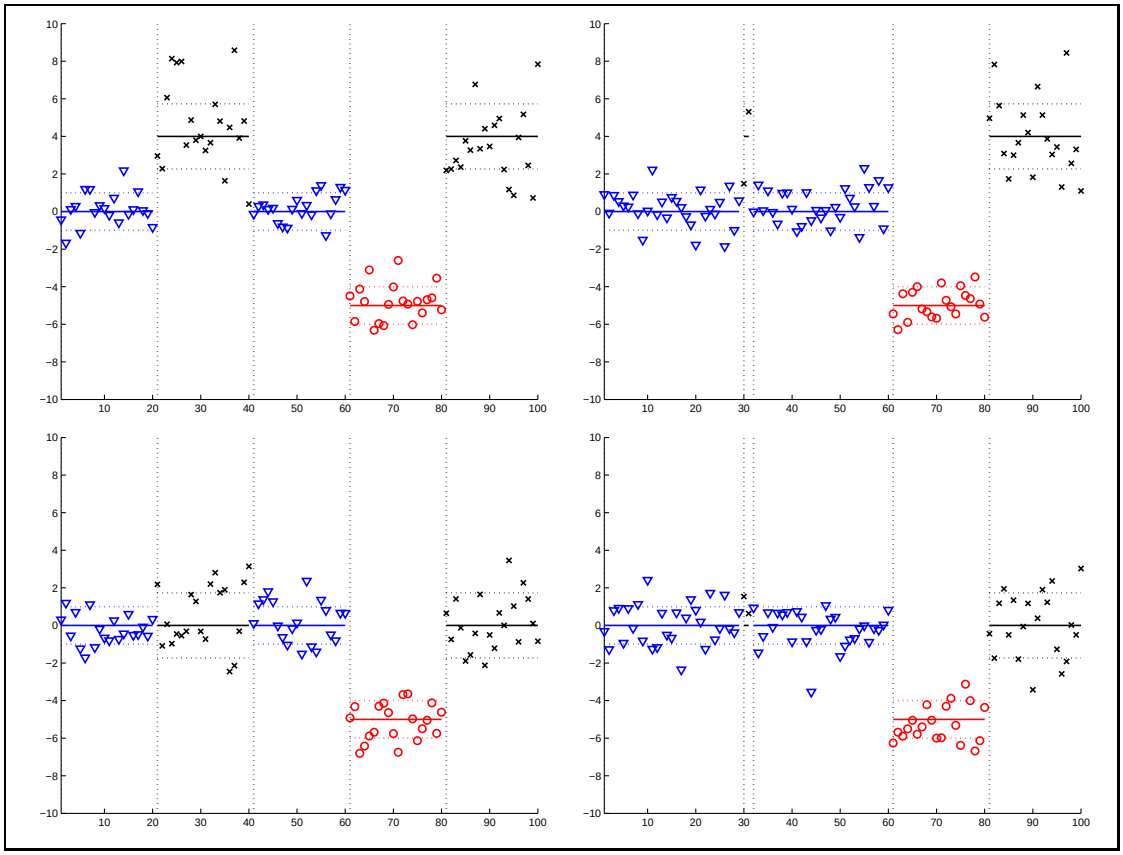


Figure 10.1: Four examples of simulations, with $n_2 = 20$ (left) and $n_2 = 2$ (right), $d = 2$ (top) and $d = 0$ (bottom). Cluster 1 is represented with circles, cluster 2 with triangles and cluster3 with crosses.

10.2 Selecting the number of clusters

In Chapter 8 we proposed to select the number of clusters first, using a BIC penalty or an adaptive method. Let us recall the principle of these methods. In a first step we construct the sequence of increasing likelihoods: $\log \widetilde{\mathcal{L}}_1 \ldots \leq \log \widetilde{\mathcal{L}}_P \leq \ldots \log \widetilde{\mathcal{L}}_{P_{max}}$, such that:

$$\begin{aligned} \log \widetilde{\mathcal{L}}_P &= \max_K \left\{ \log \mathcal{L}_{KP}(\hat{T}; \hat{\psi}) \right\} \\ \tilde{K}_P &= \operatorname*{Argmax}_K \left\{ \log \mathcal{L}_{KP}(\hat{T}; \hat{\psi}) \right\} \end{aligned}$$

$\log \widetilde{\mathcal{L}}_P$ represents the maximal fit that a segmentation/clustering model can reach when the number of clusters is P. This quantity is used to select the number of clusters whatever the number of segments. Then we propose to penalize these likelihoods as follows.

A BIC penalty

The first strategy is to use the traditional BIC criterion applied to $\log \widetilde{\mathcal{L}}_P$ such that:

$$\hat{P} = \operatorname*{Argmax}_P \left\{ \log \widetilde{\mathcal{L}}_P - \frac{\nu_P}{2} \log(n) \right\},$$

with ν_P the number of independent parameters of a mixutre with P groups. In our case $\nu_P = 3P - 1$ since we consider a mixture model with heterogeneous variances.

Adaptive method

Using the adaptive method, $\hat{P}$ is estimated like in Lavielle (2005) such that:

- calculate

$$\widetilde{J}_P = \frac{J_{P_{max}} - J_P}{J_{P_{max}} - J_1} \times (P_{max} - 1) + 1, \quad \text{with} \quad J_P = -\log \widetilde{\mathcal{L}}_P,$$

- calculate the empirical second derivative of $\widetilde{J}_P$ such that:

$$\forall P \in \{2, \ldots, P_{max} - 1\}, \quad D_P = \tilde{J}_{P-1} - 2\tilde{J}_P + \tilde{J}_{P+1},$$

- then select the number of clusters, such that:

$$\hat{P} = \max_P \{P \in \{2, \ldots, P_{max} - 1\} \mid D_P \geq s\}.$$

This method requires the determination of parameter s. In Figure 10.2 is plotted the second derivative D_P according to the number of clusters for a configuration where $d = 1$. If threshold $s = 0.75$ the estimated number of cluster is $\hat{P} = 2$, whereas it is $\hat{P} = 4$ if $s = 0.5$. Consequently selecting the number of clusters with the adaptive strategy appears to be sensitive to this threshold. The simulation

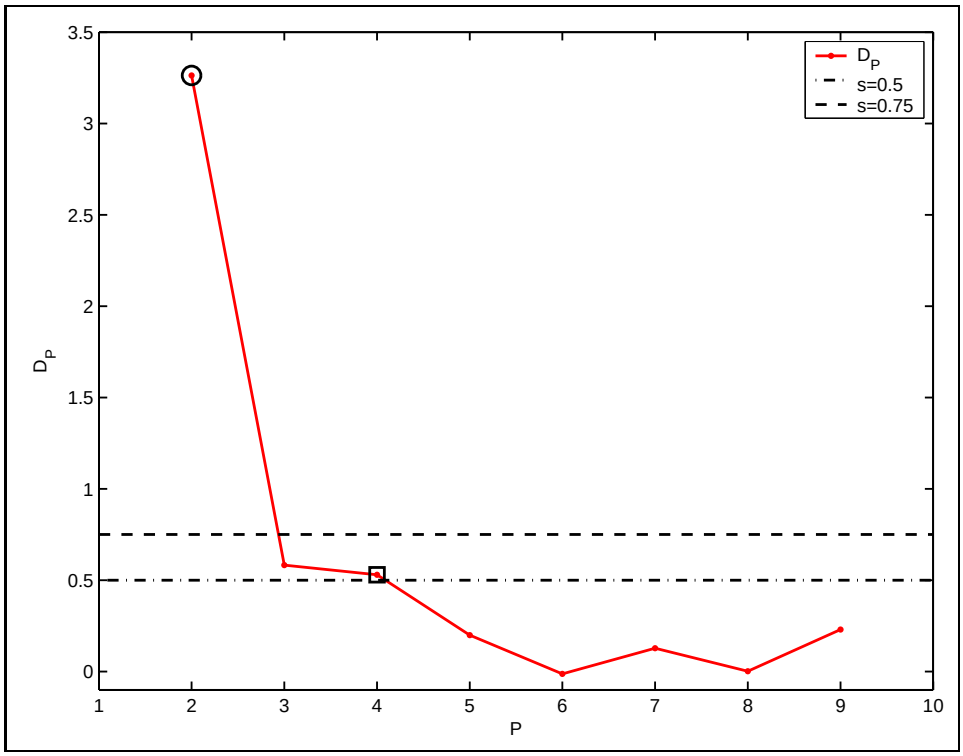


Figure 10.2: Representation of the second derivative of the normalized contrast $\widetilde{J}_P$ according to the number of clusters and sensitivity to threshold s. If $s = 0.5$, $\hat{P} = 4$ (square) and if $s = 0.75$, $\hat{P} = 2$ (circle).

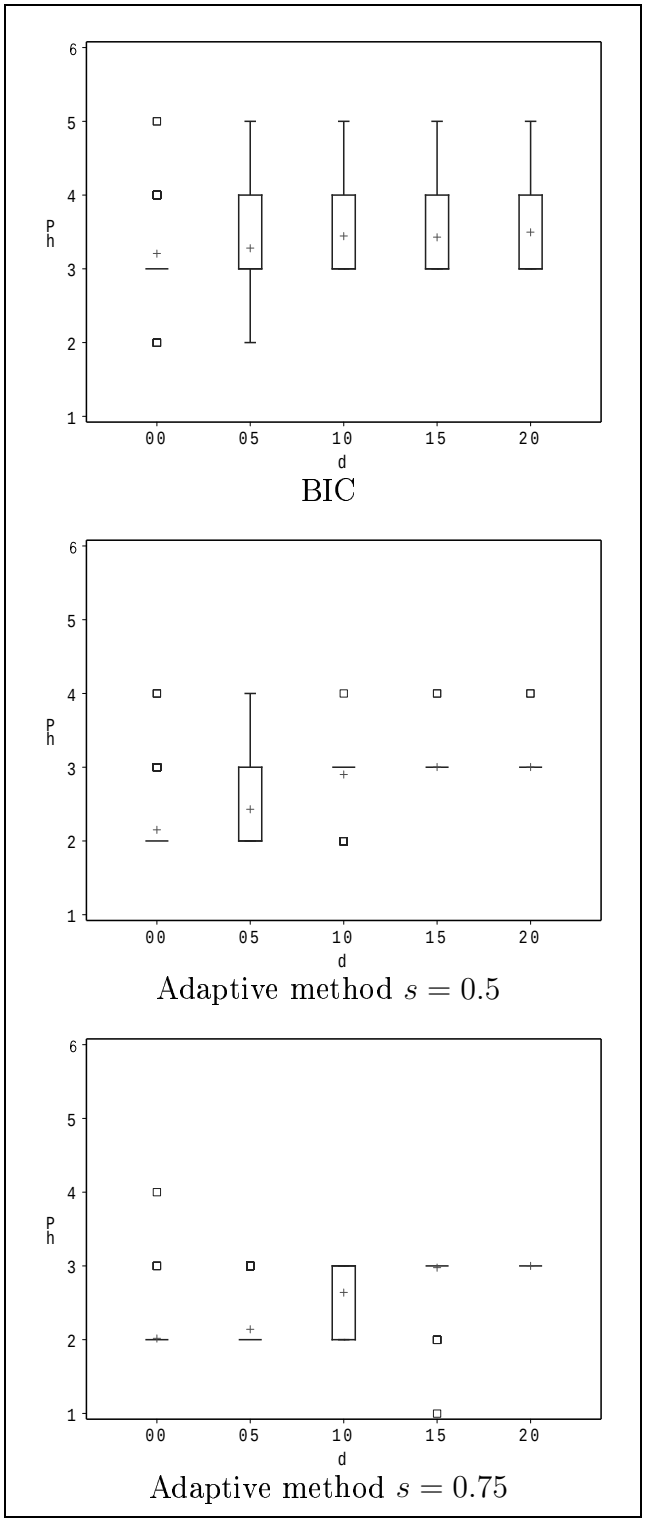


Figure 10.3: Estimated number of clusters according to the distance between clusters.

study will be used to choose this parameter.

Results analysis

Figure 10.3 presents the estimated number of clusters with the two methods, according to the distance between clusters 1 and 3. A first result is that the BIC penalty systematically leads to the selection of an overly high number of clusters, even in easy configurations. This is why we focus on the adaptive method.

It can be seen that the adaptive method tends to select a lower number of groups as the distance between clusters 1 and 3 decreases. As a result, if we consider the examples presented in Figure 10.1 (bottom), the adaptive method will "prefer" to select 2 clusters rather than 3, which leads to a partition which is more parcimonious. Moreover, this behavior seems to be stable according to the size of segment 2 (data not shown). This result seems reasonable, since the ability to detect a cluster is linked to the "separability" of clusters rather than to the size of segments.

As for parameter s, its values change the results when the separability of the mixture is intermediate. A higher value will tend to select 2 clusters rather than 3 when $d = 0.5$ and $d = 1$. Consequently, the adaptive method is sensitive to threshold s. The choice of this parameter should be done based on the objective of our method. Preventing false positives would lead to the choice of $s = 0.75$ in order to prevent the addition of two many clusters, but this could lead to a decrease in the power of the method to detect clusters. Nevertheless when the distance bewteen clusters is lower than one, it means that the jump in the mean is close to the variance. We recall that in array CGH data analysis, we are interested in the detection of jumps in the mean of the signal that reflect changes in gene copy-numbers. Consequently it appears reasonable to neglect small jumps and then to prevent false positive errors. This is why we choose $s = 0.75$.

10.3 Selecting the number of segments

Now that the number of clusters has been selected with the adaptive method, we discuss the choice of the selection of the number of segments. In Chapter 7 we proposed two methods for this choice.

When P is fixed we note $J_K = -\log \mathcal{L}_{KP}(\hat{T}, \hat{\psi})$, and we note $\tilde{K}_P$ the number of segments for which the log-likelihood decreases. Then we calculate:

$$\tilde{J}_K = \frac{J_{\tilde{K}_P} - J_K}{J_{\tilde{K}_P} - J_P}(\tilde{K}_P - P) + P,$$

such that $\tilde{J}_{\tilde{K}_P} = P$ and $\tilde{J}_P = \tilde{K}_P$. Then we calculate the empirical second derivative of $\tilde{J}_K$, such that:

$$\forall K \in \{P+1, \ldots, \tilde{K}_P - 1\}, \;\; D_K = \tilde{J}_{K+1} - 2\tilde{J}_K + \tilde{J}_{K-1}.$$

The number of segments is chosen such that:

$$\hat{K} = \max_K \left\{ K \in \{P+1, \ldots, \tilde{K}_P - 1\} \mid D_K \geq s \right\},$$

with conditions:

1. if $\tilde{K}_P = P$ then $\hat{K} = P$,
2. if $\tilde{K}_P = P + 1$ then $\hat{K} = P + 1$,
3. if $D_K \leq s$ then $\hat{K} = \tilde{K}_P$.

Condition 3 means that if there is no break in the slope of the curve between $K = P$ and $K = \tilde{K}_P$ we consider that the interesting number of segments is the one for which the log-likelihood decreases. Figure 10.4 illustrates the influence of threshold s on this procedure. If s is small, the procedure will select $\hat{K} = 10$ segments, whereas if $s = 0.75$ the procedure will select $\hat{K} = \tilde{K}_P$ segments. Setting s to a higher value will tend to ignore small breaks in the curve between $K = P$ and $K = \tilde{K}_P$. Consequently, the selection procedure will be more conservative if the threshold is small, *i.e.* it will tend to select a lower number of segments.

The second method is based on a BIC penalty such that:

$$\hat{K}_{\hat{P}} = \operatorname*{Argmax}_K \left\{ \log \mathcal{L}_{K,\hat{P}}(\hat{T}, \hat{\psi}) - \frac{1}{2} \log(n) \times K \right\}.$$

Results analysis

Figures 10.5 and 10.8 present the estimated number of segments according to the distance between clusters and to the size of segments respectively. The first result is that the adaptive method produces outliers, meaning that for some configurations, the estimated number of segments can explose whatever the threshold s. On the contrary, the BIC penalty seems to be very stable. We recall that boxplots were constructed on 500 points (there are 5 different sizes of segments for each d).

It appears that the unstable behavior of the adaptive method is linked to the estimation of the number of clusters. To explain why, we consider an example where the number of clusters is over-estimated. In figure 10.7 is presented the log-likelihood for a configuration with $d = 0$ and $n_2 = 2$. In this case the estimation of the number of clusters gives 4 clusters whereas 2 clusters would have been suitable. Consequently the log-likelihood is increasing until $\tilde{K}_P = 19$, and the adaptive method selects $\hat{K} = 10$ with a threshold $s = 0.5$. This example has already been used to show the effect of threshold s on the estimation of $\hat{K}$. The fact that $\tilde{K}_P$ is high can be explained by the estimated number of clusters. Indeed when $P = 4$ there exists more possibilities to segment data with alternate labels, as shown in Figure 10.6. On the contrary the BIC penalty seems to be robust to the bad estimation of P. The penalized likelihood is presented in Figure 10.7. This stability is due to the fact that the BIC penalty is less sensitive

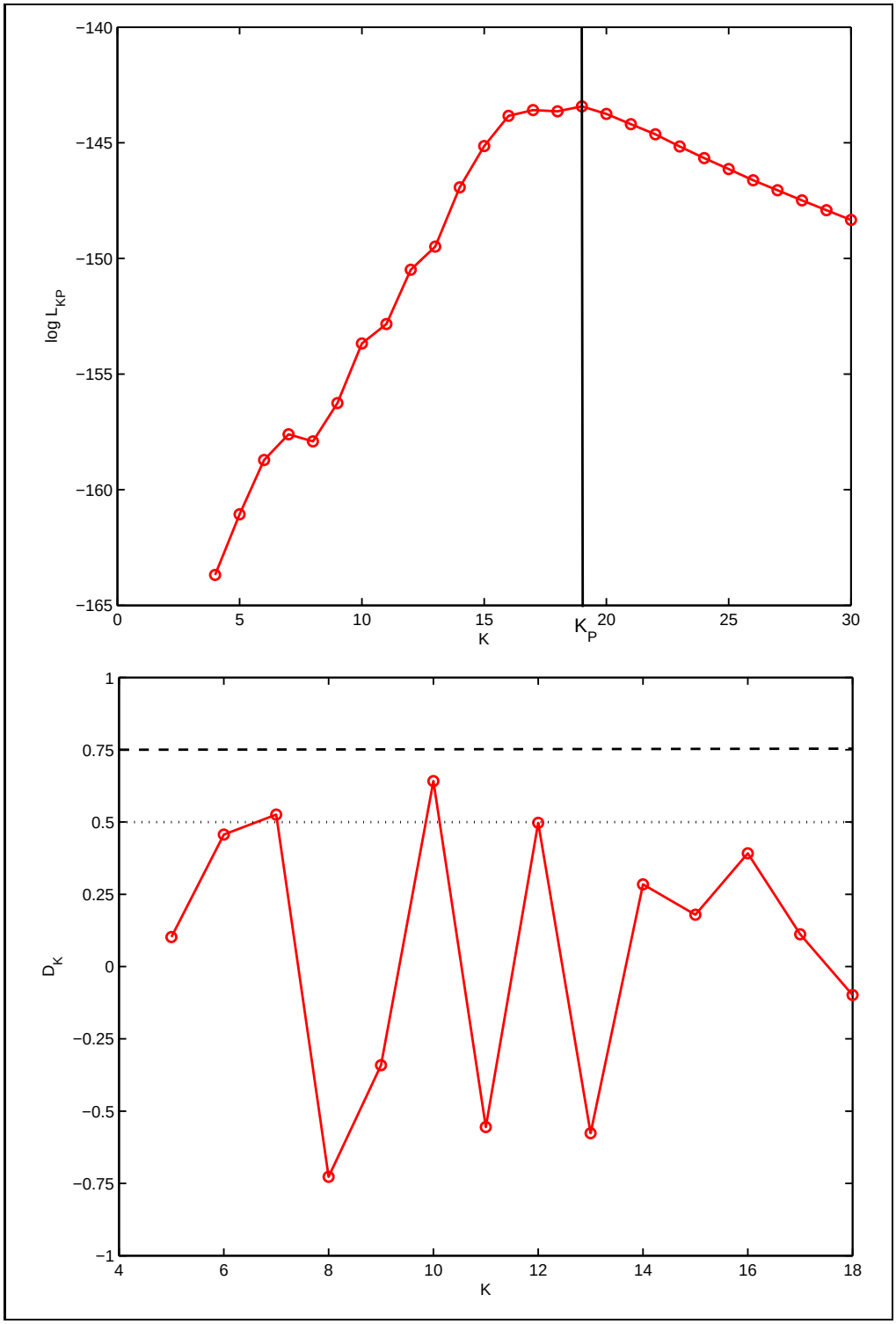


Figure 10.4: Illustration of the effect of threshold s on the selection of K. Top: log-likelihood when $\hat{P} = 4$ for a configuration with $d = 0$ and $n_2 = 2$. Bottom: empirical second derivative D_K.

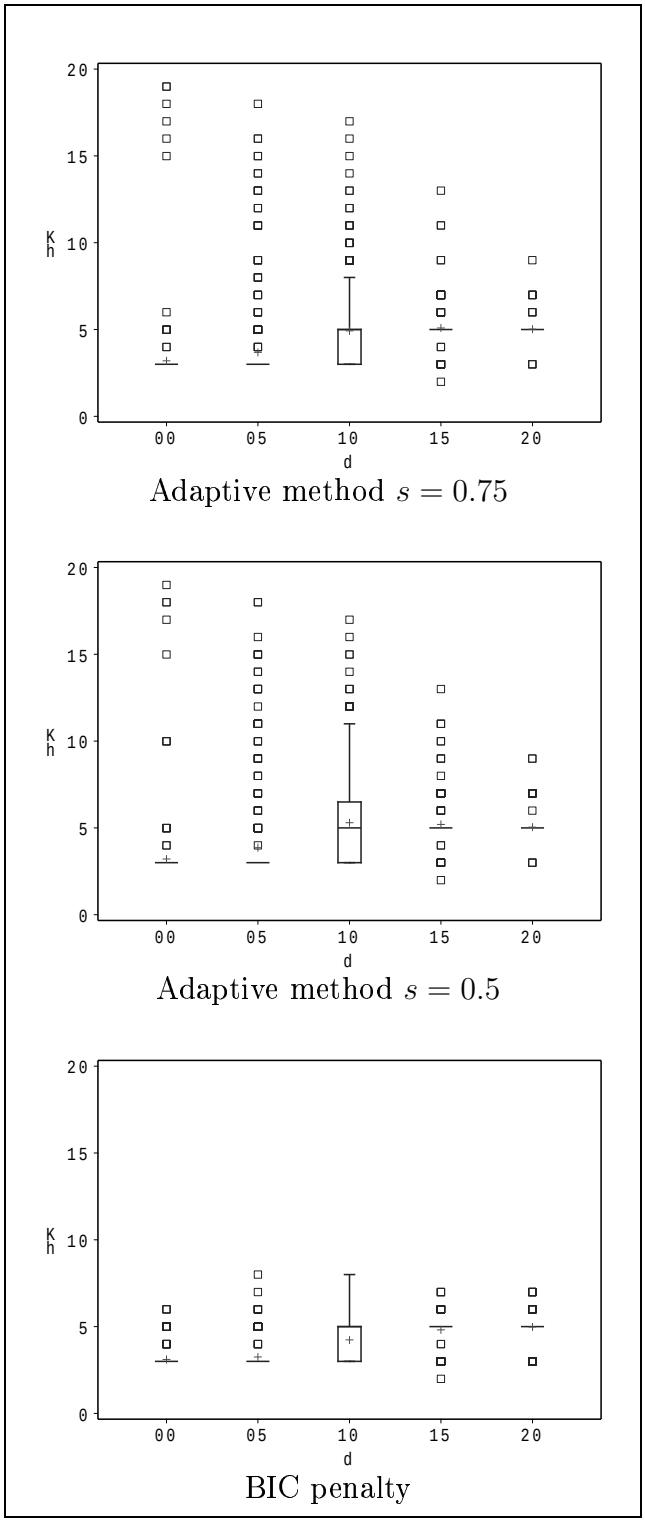


Figure 10.5: Estimated number of segments according to the distance between clusters.

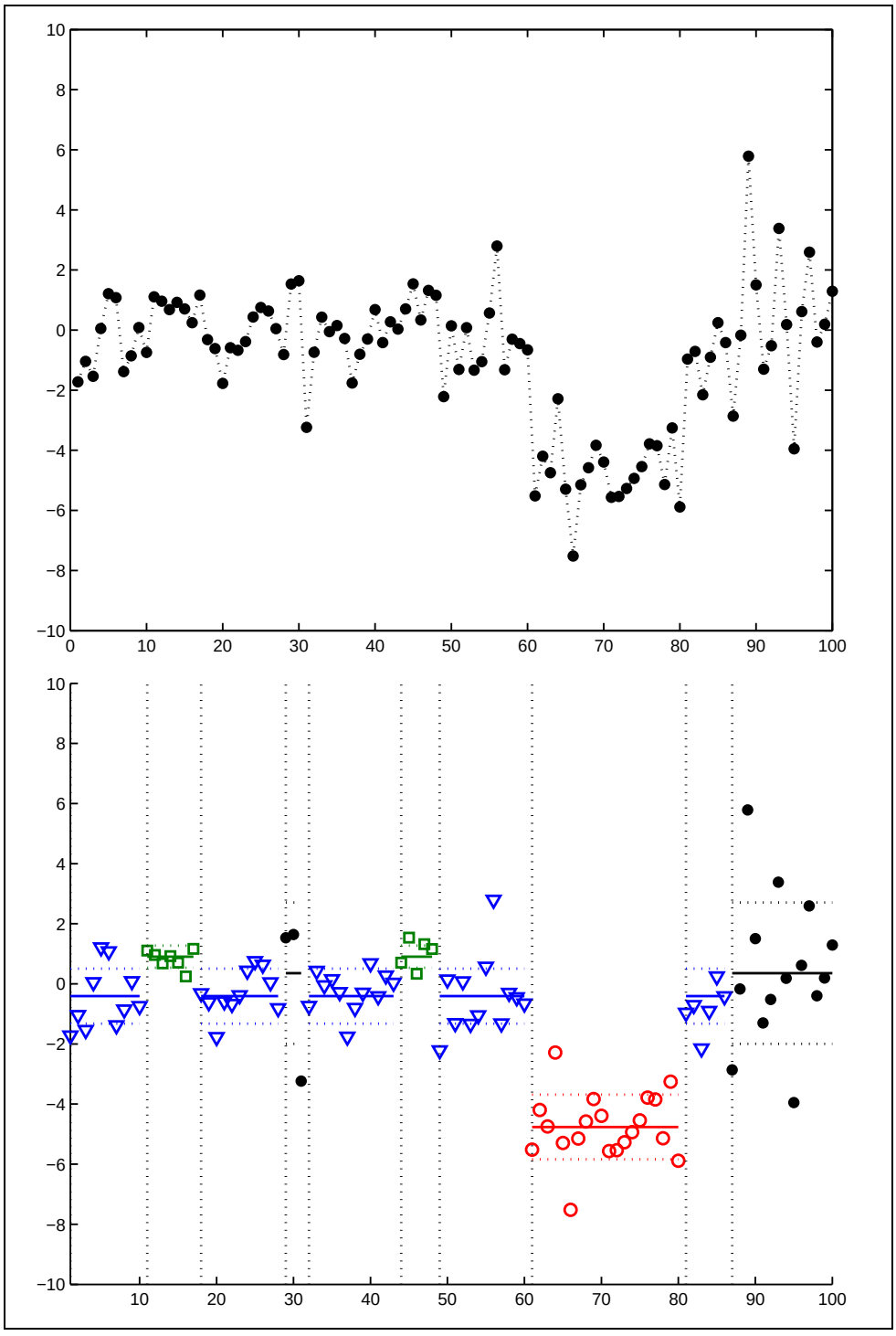


Figure 10.6: Result of the selection procedure when P is overestimated. Bottom: $\hat{P} = 4$ and $\hat{K} = 10$ with the adaptive method ($s = 0.5$).

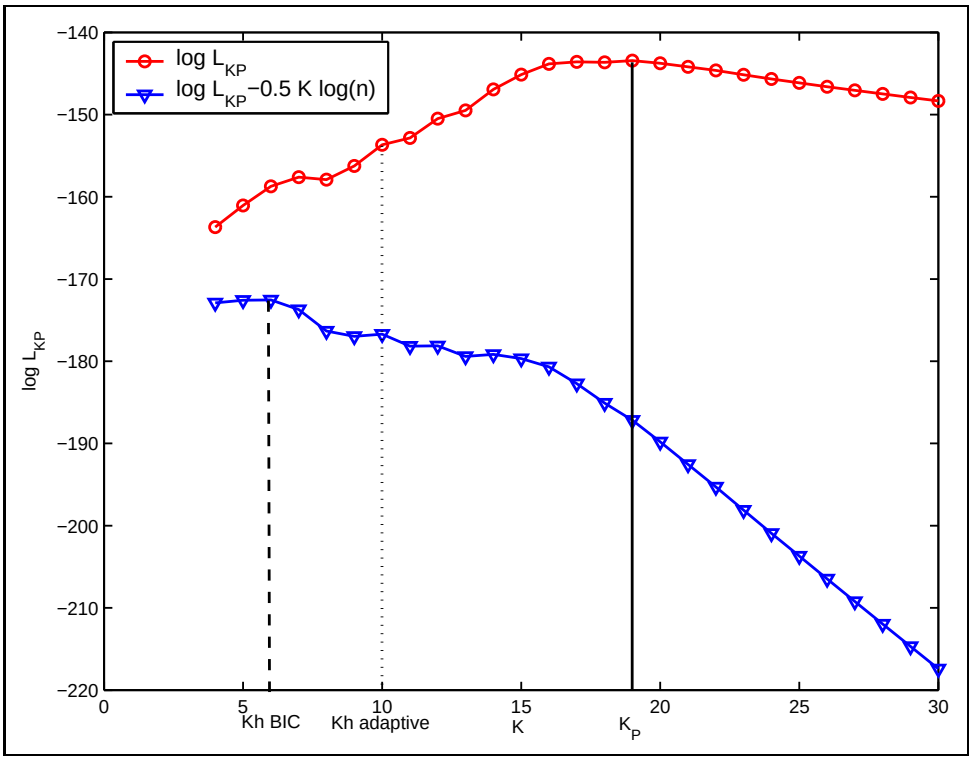


Figure 10.7: Penalizing the log-likelihood with a BIC penalty to select the number of segments, and comparison with the adaptive method.

to the geometrical behavior of the likelihood. Interestingly, the BIC penalty does not over-estimate the number of segments, as it is the case in segmentation models.

Since the estimated number of segments depends on the estimated number of clusters, we consider 2-way frequency tables for $\hat{P}$ and $\hat{K}$ in Table 10.3. These tables are calculated according to the distance between clusters. From Table 10.3 we have:

- when clusters are not separated ($d = 0.0$) both methods select exclusively $\hat{P} = 2$ and $\hat{K} = 3$,
- for intermediate configurations ($d = 0.5, 1$) two configurations are selected ($\hat{P} = 2, \hat{K} = 3$) and ($\hat{P} = 3, \hat{K} = 5$)
- when clusters are well separated ($d = 1.5, 2$), both methods select almost systematically $\hat{P} = 3$ and $\hat{K} = 5$.

From this table, we can see that the BIC penalty is more conservative compared with the adaptive method. When the distance between clusters is intermediate ($d = 0.5, 1$) the estimated number of segments is higher than 5 in 10% and 30% of cases for the adaptive method, whereas the BIC penalty gives 2% and 10% of outliers. This is why we choose the BIC penalty in the following.

Interestingly, the main factor affecting the estimated number of segments seems to be the distance between clusters, *i.e.* the relative jump in the mean between two segments, and not the size of segments (Figure 10.8). This means that even if segments are of small size, they are detected by the method, as well as segments of big size. This could be interpreted as an advantage of the segmentation/clustering compared with a pure segmentation model. With our method, even if a segment is of small size, the fact that it belongs to a cluster being composed of other segments helps in its recovering. This particular behavior will be studied in the next chapter.

10.4 Conclusion

In Chapter 8 we proposed a heuristic to select the number of clusters and the number of segments. Since this method is empirically motivated, it was crucial to assess its performance on simulated data, to determine the average behavior of the method. Interestingly our model selection heuristic appears to be "doubly adaptive". Indeed, since the selection of $\hat{K}_{\hat{P}}$ strongly depends on the choice of $\hat{P}$, a parcimonious method to select P leads to a parcimonious method to select $K_{\hat{P}}$, as shown in Table 10.3. When the number of clusters is underestimated the resulting number of segments is underestimated as well. This illustrates the interest to have an adaptive method to select the number of clusters, as discussed before. As for the choice of the selection method for the number of segments, the BIC penalty seems more appropriate since it is more robust to a bad estimation of the number of clusters.

---------- d=0.0 ----------

Kh	hatP			
Percent	2	3	4	Total
3	96.80	0.00	0.00	96.80
4	0.40	0.00	0.00	0.40
5	1.20	0.00	0.00	1.20
5+	0.00	1.40	0.20	1.60
Total	98.40	1.40	0.20	100.00

---------- d=0.5 ----------

Kh	hatP		
Percent	2	3	Total
3	85.80	0.00	85.80
4	0.00	0.20	0.20
5	0.00	3.60	3.60
5+	0.00	10.40	10.40
Total	85.80	14.20	100.00

---------- d=1.0 ----------

Kh	hatP		
Percent	2	3	Total
3	36.00	0.20	36.20
4	0.00	0.80	0.80
5	0.00	33.40	33.40
5+	0.00	29.60	29.60
Total	36.00	64.00	100.00

---------- d=1.5 ----------

Kh	hatP			
Percent	1	2	3	Total
3-	0.20	0.00	0.00	0.20
3	0.00	1.80	1.20	3.00
4	0.00	0.00	0.40	0.40
5	0.00	0.00	85.00	85.00
5+	0.00	0.00	11.40	11.40
Total	0.20	1.80	98.00	100.00

---------- d=2.0 ----------

Kh	hatP	
Percent	3	Total
3	0.60	0.60
5	96.40	96.40
5+	3.00	3.00
Total	100.00	100.00

Adaptive method $s = 0.5$

---------- d=0.0 ----------

Kh	hatP			
Percent	2	3	4	Total
3	93.00	0.80	0.00	93.80
4	1.20	0.40	0.00	1.60
5	3.80	0.20	0.00	4.00
5+	0.40	0.00	0.20	0.60
Total	98.40	1.40	0.20	100.00

---------- d=0.5 ----------

Kh	hatP		
Percent	2	3	Total
3	85.20	2.40	87.60
4	0.40	0.60	1.00
5	0.20	9.40	9.60
5+	0.00	1.80	1.80
Total	85.80	14.20	100.00

---------- d=1.0 ----------

Kh	hatP		
Percent	2	3	Total
3	36.00	8.00	44.00
4	0.00	1.20	1.20
5	0.00	45.00	45.00
5+	0.00	9.80	9.80
Total	36.00	64.00	100.00

---------- d=1.5 ----------

Kh	hatP			
Percent	1	2	3	Total
3-	0.20	0.00	0.00	0.20
3	0.00	1.80	9.40	11.20
4	0.00	0.00	0.40	0.40
5	0.00	0.00	83.80	83.80
5+	0.00	0.00	4.40	4.40
Total	0.20	1.80	98.00	100.00

---------- d=2.0 ----------

Kh	hatP	
Percent	3	Total
3	3.20	3.20
5	93.60	93.60
5+	3.20	3.20
Total	100.00	100.00

BIC penalty

Table 10.3: Two-way frequency tables for $\hat{P}$ and $\hat{K}$ according to the distance between clusters.

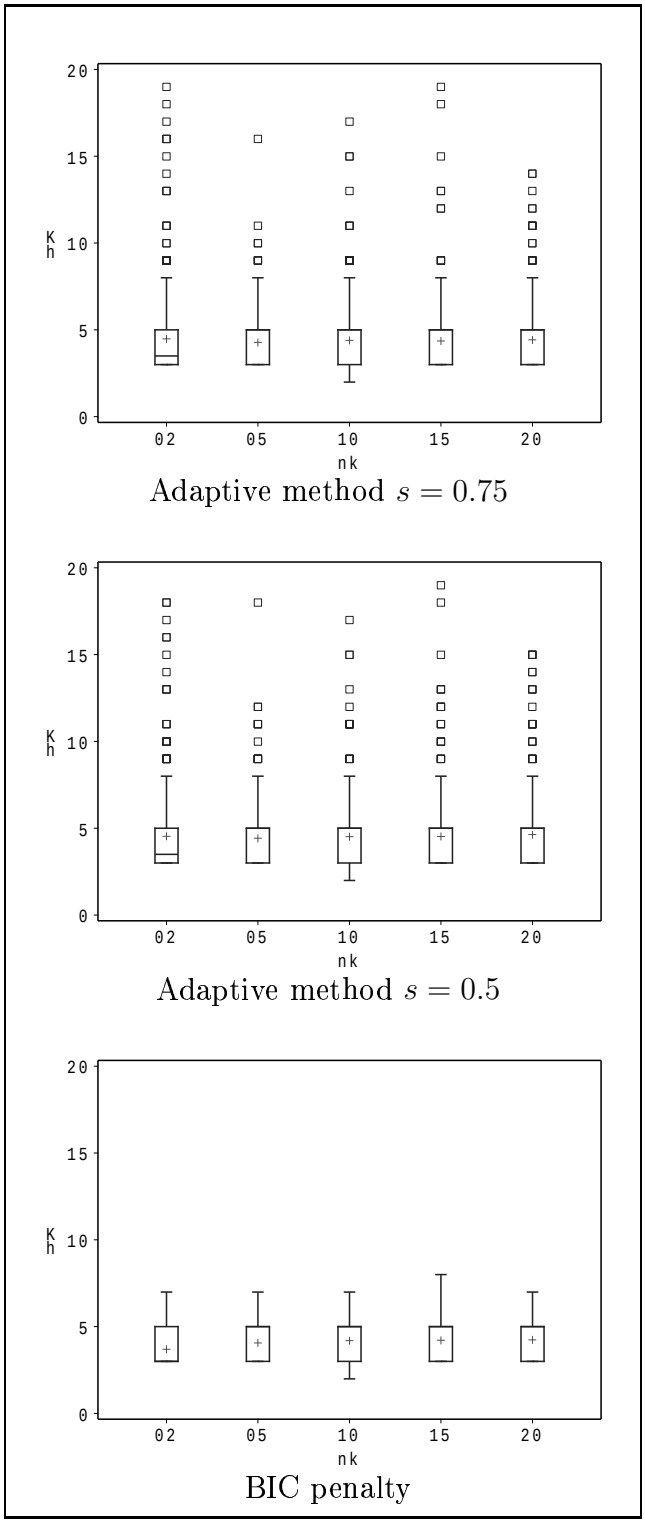


Figure 10.8: Estimated number of segments according to the size of segment 2.

Chapter 11

Performance

In this chapter we propose to compare the performance of our method with the performance of hidden Markov models. In chapter 7 we explained the differences between both methods, from a modelling point of view. In this chapter we propose to compare the methods using simulated data. For this purpose we use the simulated data which have been described in the previous chapter.

Let us briefly recall the construction of the simulated data set. One simulation is constituted on 100 data points which are segments into 5 segments, the segments being clustered into 3 groups. The design of the simulation study is a factorial design with two factors of variations: the distance between cluster 1 and 3 noted d, and the size of segment 2. Five levels are considered for each factor, as shown in Table 11.1. A representation of 4 situations is given in Figure 11.1.

Since the objective of the segmentation/clustering model is to cluster data points into a finite number of groups, we will use criteria such as the empirical error rate, and the specificity/sensitivity to compare our model with HMMs in terms of clustering. The second objective of these methods being to provide a segmentation of the data, we will also assess the ability of both method to correctly locate the breakpoints. A comparison will also be made with pure segmentation methods.

	cluster 1	cluster 2	cluster 3
m	0	-5	varying
s^2	1	2	3
π	2/5	1/5	2/5

m_3	4	3	2	1	0
d	2	1.5	1	0.5	0

Size	n_1	n_2	n_3	n_4	n_5
cluster label	1	3	1	2	3
	30	2	28	20	20
	28	5	27	20	20
	25	10	25	20	20
	23	15	22	20	20
	20	20	20	20	20

Table 11.1: Factors of variation for the simulation study.

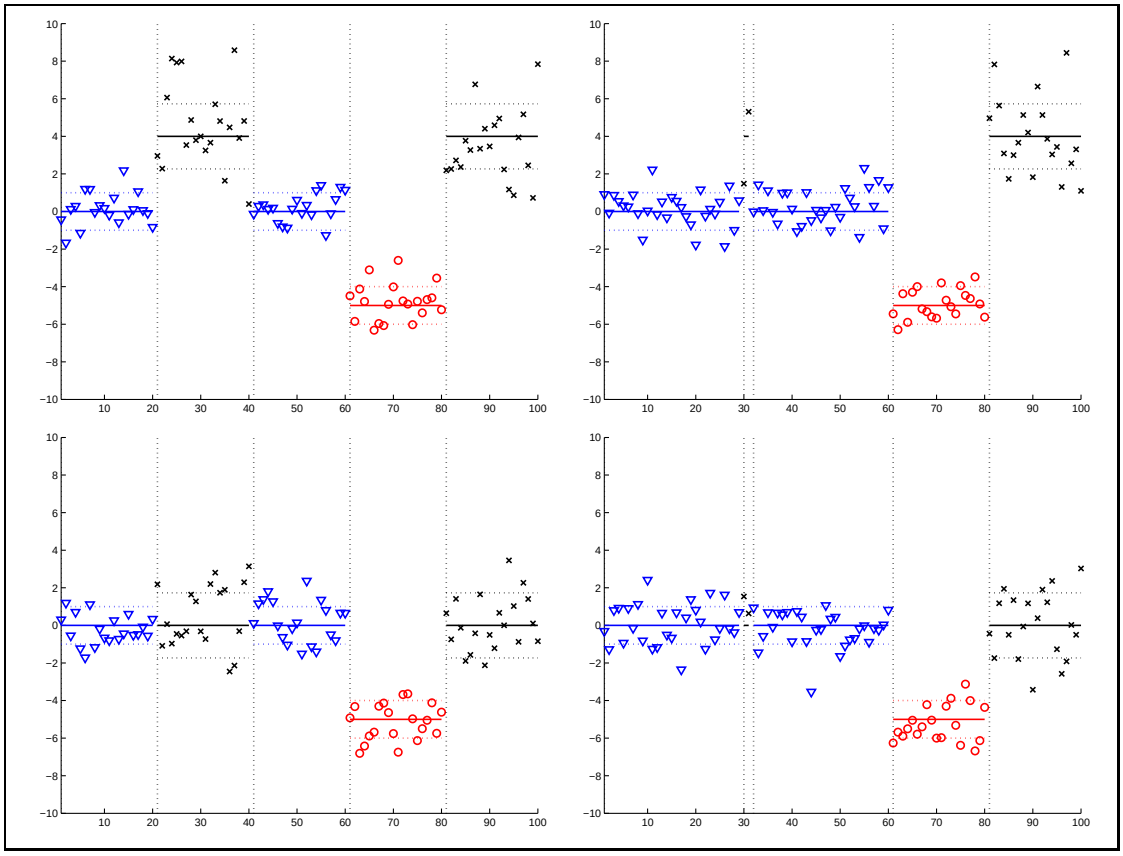


Figure 11.1: Four examples of simulations, with $n_2 = 20$ (left) and $n_2 = 2$ (right), $d = 2$ (top) and $d = 0$ (bottom).

11.1 Clustering results

Running HMMs

The Gaussian hidden Markov model is run using a freely available toolbox created by Kevin Murphy [1], University of British Columbia, Vancouver. When the number of hidden states is fixed, the EM algorithm is used to estimate the parameters of the HMM, using 10 random starts. Since the number of hidden states of the HMM is linked to the number of clusters in the segmentation/clustering model we choose to select the number of hidden states using the same adaptive procedure we use for the selection of the number of clusters. The adaptive method will tend to select two hidden states rather than 3 when the distance between clusters is small. Once this number has been estimated, the hidden sequence is recovered using the Viterbi algorithm which provides the recovered sequence of hidden variables, noted $\tilde{z}_1^n$ in the following.

In order to compare the performance of our method with HMMs, we choose to define some quality criteria to compare. Let us recall that we know the true label of the data, which are noted z_t for the label of data point y_t.

11.1.1 Quality criteria

The aim of both methods is to cluster data points into a finite number of groups. In the HMM context, this clustering is done via the reconstruction of the hidden sequence of variables, previously noted $\tilde{z}_1^n$. In the segmentation/clustering context, the recovered label variables are obtained with the MAP rule, and noted $\hat{z}^k$, $k = 1, \ldots, K$. Since $\hat{z}^k$ indicates the label of segment k, we introduce a secondary sequence of label variables noted $\hat{z}_1^n$ which indicates the label of data points within segments such that:

$$\forall t \in I_k, \quad \hat{z}_t = \hat{z}^k.$$

Then we define the following quality criteria to compare:

- the empirical error rate noted EER, and defined such that:

$$\begin{aligned} EER_{HMM} &= \frac{1}{n}\sum_{t=1}^{n} \mathbb{1}\{\tilde{z}_t \neq z_t\}, \\ EER_{seg/clust} &= \frac{1}{n}\sum_{t=1}^{n} \mathbb{1}\{\hat{z}_t \neq z_t\}. \end{aligned}$$

- the sensitivity which is defined as the proportion of true positives detected compared with the total number of positives. Using terminology: TP =true positives, FN =false negatives, it follows that:

$$Sensitivity = \frac{TP}{TP + FN}.$$

[1] `http://www.cs.ubc.ca/ murphyk/Software/HMM/hmm.html`

- the specificity which is defined as the proportion of true negatives detected compared with the total number of negatives. Using terminology: TN =true negatives, FP =false positives, it follows that:

$$Specificity = \frac{TN}{TN + FP}.$$

Since cluster 2 is well separated from the other clusters, we will focus on the specificity and sensitivity for clusters 1 and 3 only. The following table gives the definitions required to calculate these criteria for cluster 1 for instance.

$$\begin{array}{c|l} TP_1 & \sum_{t=1}^{n} \mathbb{1}\{\tilde{z}_t = 1, z_t = 1\} \\ TN_1 & \sum_{t=1}^{n} \mathbb{1}\{\tilde{z}_t \neq 1, z_t \neq 1\} \\ FP_1 & \sum_{t=1}^{n} \mathbb{1}\{\tilde{z}_t = 1, z_t \neq 1\} \\ FN_1 & \sum_{t=1}^{n} \mathbb{1}\{\tilde{z}_t \neq 1, z_t = 1\} \end{array}$$

11.1.2 Results

Figure 11.2 represents the empirical error rate calculated for the segmentation/clustering model and for HMMs, according to the distance between clusters. It can be seen that the segmentation/clustering method has a slight advantage since its average and median EER are lower. As expected, the EER decreases as the distance between clusters 1 and 3 increases and it is close to zero in easy configurations. It can be seen that when the distance between clusters is null ($d = 0$) the HMM can lead to a high empirical error rate. Figure 11.3 gives an example of such situation. Let us recall that when $d = 0$ it means that the difference of means is null between clusters 1 and 3, but their variance is different $s_1^2 = 1, s_3^2 = 3$. This is why three hidden states are selected, whereas 2 clusters are selected for the segmentation/clustering in this case (data not shown). Nevertheless, the selection of 3 hidden states does not lead to the recovering of segment 2 which is not detected and affected to cluster 1. On the contrary, the segmentation/clustering method selects 2 clusters and 3 segments, leading to more conservative results.

In this study we choose to consider the sensitivity and specificity for each individual clusters. However it is clear that false positives and false negatives are linked in this case. When a point is affected to cluster 1 whereas it belongs to cluster 3, it is considered as a false negative for cluster 1 and as a false positive for cluster 3. This is why the specificity associated to cluster 1 shows a symetrical behavior compared with the sensitivity associated to cluster 3. From Figure 11.4 it is clear that when the distance is small, points will be clustered to group 1 leading to an increase in false positives for group 1 and a deacrease in the associated specificity. It is the contrary for group 3 which presents many false negatives when d is small and the sensitivity for this group increases with d. The results suggest that both specificity and sensisitivity are greater for the segmentation/clustering model.

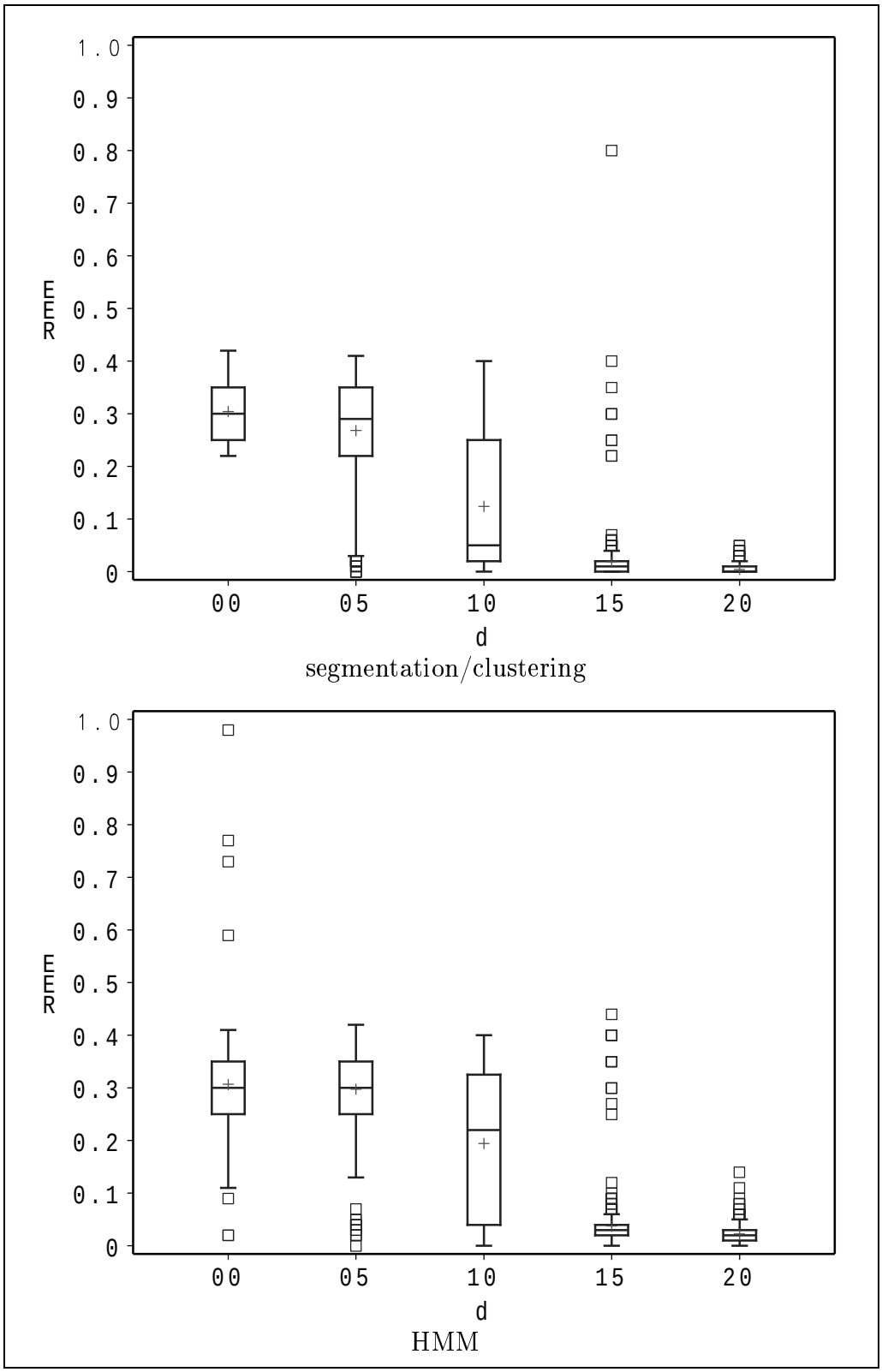


Figure 11.2: Comparison of empirical error rates between segmentation/clustering and HMMs.

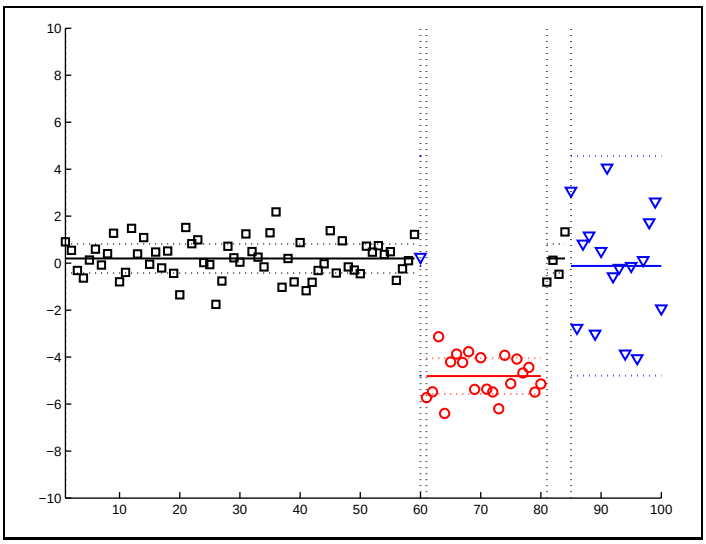


Figure 11.3: Example of simulation for which HMMs give a high empirical error rate.

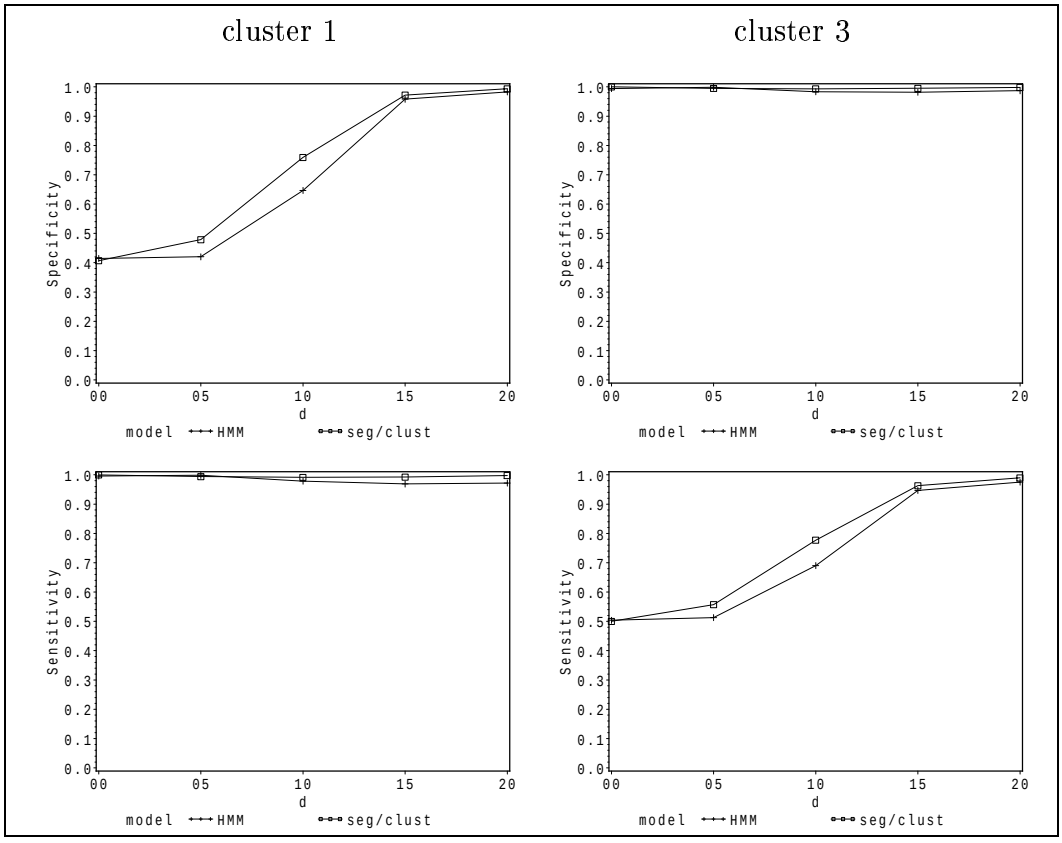


Figure 11.4: Specificity/Sensitivity of HMMs and segmentation/clustering for clusters 1 and 3, according to the distance between clusters 1 and 3.

11.2 Segmentation results

11.2.1 Quality criteria for segmentation results

Since the objective of HMMs is also to provide a segmentation of the data, we compare both methods with respect to the number of estimated segments, and to their ability to correctly locate the breakpoints. As discussed in Chapter 7 we define segments in the case of HMMs as a region for which the recovered label variables $\tilde{z}_t$ are homogeneous, and we call breakpoint the position $\tilde{t}_k$ for which the label changes. This allows us to define a segment I_k such that:

$$\forall t \in I_k \ , \ \tilde{z}_t = \tilde{z}^k.$$

In order to assess the ability of the methods to locate the breakpoints, we use a secondary sequence $\{r_t\}_{1,n}$ such that:

$$r_t = \begin{cases} 1 & \text{if there exists } k \text{ such that } t = t_k. \\ 0 & \text{otherwise.} \end{cases}$$

This new sequence allows us to define the specificity and sensitivity of the method regarding the detection of breakpoints. A false positive will be defined as a position for which the method detects a breakpoint whereas there is no breakpoint. The following table gives the definition of such criteria. Consequently, a

$$\begin{array}{c|l} TP & \sum_{t=1}^{n} \mathbb{1}\{\tilde{r}_t = 1, r_t = 1\} \\ TN & \sum_{t=1}^{n} \mathbb{1}\{\tilde{r}_t \neq 1, r_t \neq 1\} \\ FP & \sum_{t=1}^{n} \mathbb{1}\{\tilde{r}_t = 1, r_t \neq 1\} \\ FN & \sum_{t=1}^{n} \mathbb{1}\{\tilde{r}_t \neq 1, r_t = 1\} \end{array}$$

segmentation method with a high specificity does not add false positive breakpoints. On the contrary a method with a low sensitivity will tend to ignore some breakpoints and will be conservative.

11.2.2 Results for segmentation

First of all, Figure 11.5 presents the recovered number of segments compared with segmentation/clustering. A first result is that these numbers are comparable but the average number of segments in the case of HMMs is greater in easy configurations ($d = 1.5, 2$). This illustrates the fact that in the case of HMMs, the number of segments is not selected but recovered once the number of hidden states have been estimated. On the contrary the segmentation/clustering model provides a way to control the number of segments to be put in the profile, whereas in the case of HMMs, this number only depends on the choice of the number of hidden states.

Since the size of the data set is fixed, adding segments leads to an increase in the number of false positives, and then to a decrease in the specificity of the method. Figure 11.6 shows the evolution of the specificity/sensitivity of both methods with respect to the size of segment 2 and the distance between clusters.

It can be seen that even if the number of segments is "over-estimated" in the case of HMMs, the associated decrease in the specificity is negligible. Consequently both methods are very specific, meaning that they do not create breakpoints when there is no breakpoint to detect.

From Figure 11.6 we can see that the sensitivity is constant according to the size of segment 2 and that it increases with the distance between clusters. Let us recall that the sensitivity is low when the number of false negatives is high. This behavior can be linked to the associated number of segments. Indeed, when the distance between clusters is small, the associated number of segments is 3 rather than 5 (the true number). The behavior can be interpreted as a tendency to ignore segments when the jump in the mean between segments is small regarding the variance. Consequently, ignoring segments leads to an increase in false negatives and then to a decrease in the sensitivity. Moreover the results suggest that the segmentation/clustering model is more sensitive compared with HMM for the positioning of breakpoints.

The last comparison we make is between segmentation/clustering and pure segmentation methods. Since segmentation/clustering aims at finding breakpoints as well as clustering the data, our hypothesis is that the mixture model can help in the recovering of breakpoints. Let us consider the examples shown in Figure 11.1. Since segments 2 and 5 belong to the same cluster, our question is to assess if a segment can help to recover segments being in the same group. To answer this question we consider a segmentation model and we compare the sensitivity and specificity of segmentation/clustering and segmentation for the breakpoints. We consider a segmentation model with heterogeneous variances, since the mixture model which has been used to simulate the data considers heterogeneous variances. The number of segments has been estimated using the adaptive method.

Figure 11.6 shows the sensitivity for breakpoint positioning, has it was done to compare HMM and segmentation/clustering. The same conclusions can be drawn, meaning that the specificity is high for both methods (not shown). Since the number of segments is not over-estimated when d is high, there is no decrease in the specificity of the segmentation method, as it was the case of HMMs. As for the sensitivity, it increases with the distance between clusters, since the number of false negative decreases. In the case of segmentation, the sensitivity decreases when the size of segment 2 decreases, meaning that the method ignores segments with small size. Interestingly the effect of the size of segment 2 is lower for segmentation/classification. This behavior illustrates the ability of segmentation/clustering to recover segments of small size if they belong to a cluster with other segments. Consequently, the mixture model in the segmentation/clustering method helps to recover some breakpoints.

11.3 Conclusion

In this chapter we proposed to compare the performance of the segmentation/clustering model with hidden Markov models based on simulation studies. We proposed to

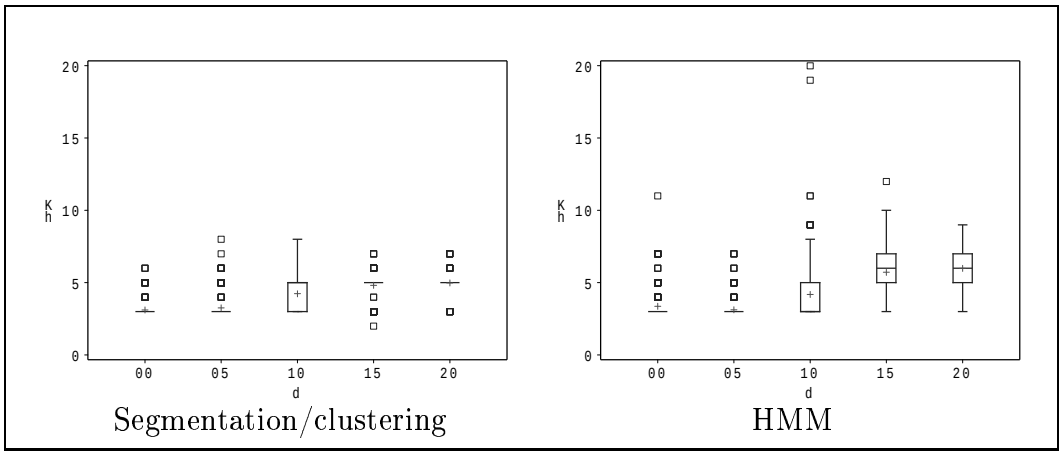


Figure 11.5: Number of "segments" in the case of HMM compared with the selected number of segments in the case of segmentation/clustering.

do so by comparing the ability to cluster the data into a finite number of groups, and the ability of both methods to correctly locate breakpoints. This comparison has been done using quality criteria for clustering and for segmentation, and we showed that the performance of the segmentation/clustering model were better with respect to every quality criteria compared with hidden Markov models. Nevertheless, we should recall that the data have been simulated using a segmentation/clustering model. To this extent, the fact that our model shows better performance on these data may be due to the simulation model which favours the segmentation/clustering model. This remark can also be made when comparing the segmentation/clustering model with the segmentation model, since the clustered nature of the simulated data favours the segmentation/clustering model which considers an additional information.

Nevertheless, the fact that our model shows (slightly) better performance compared with two different methods indicates that the segmentation/clustering model combines the advantages of both methods. Indeed the segmentation/clustering model is as sensitive and specific as HMMs in the clustering context, and as sensitive and specific as segmentation methods in the segmentation context. To this extent the method we propose is efficient for both clustering and segmentation.

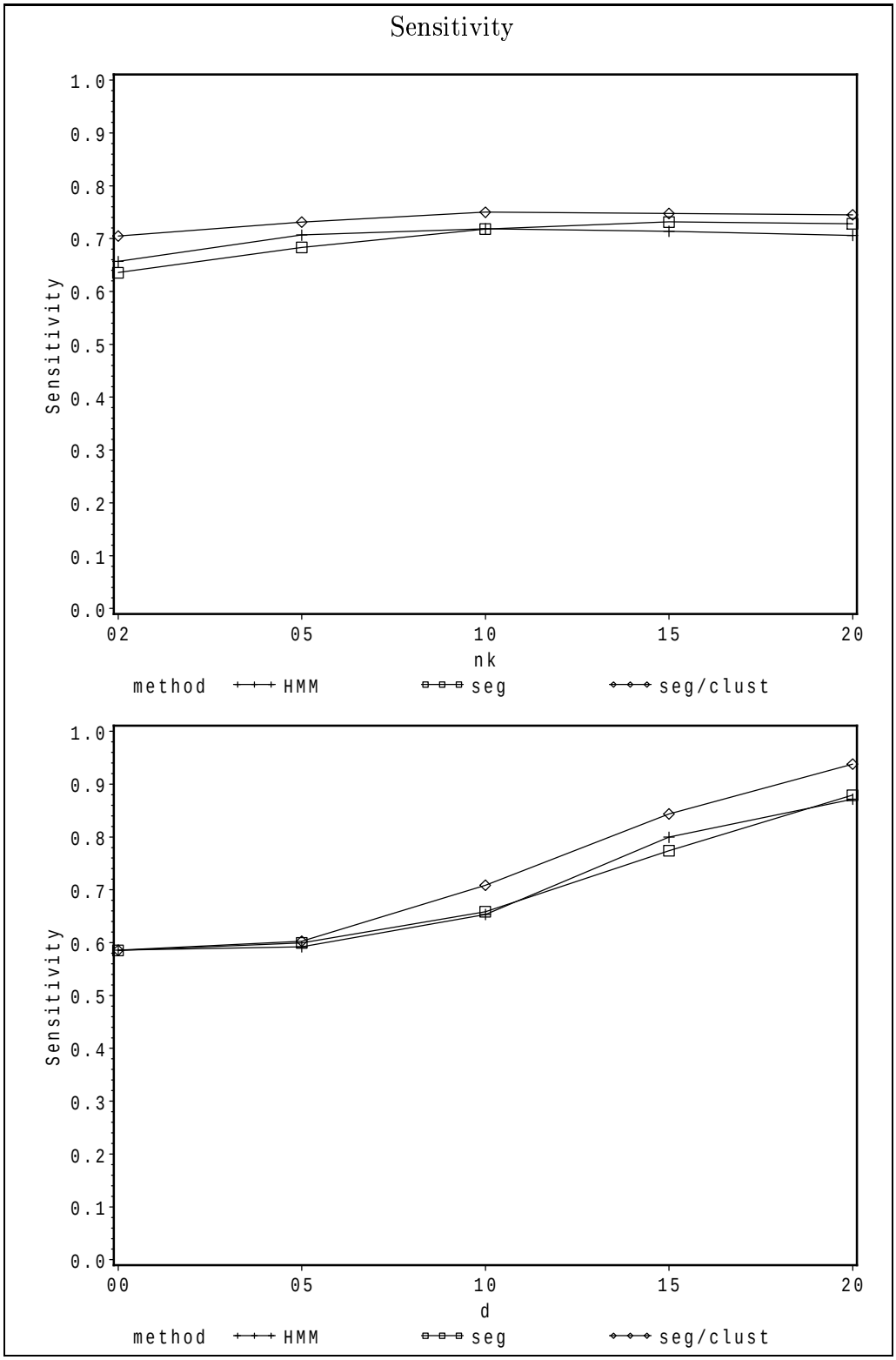


Figure 11.6: Sensitivity for breakpoint positioning between segmentation/clustering, HMM and segmentation. Top: according to the size of segment 2, Bottom: according to the distance between clusters.

Chapter 12

Analysis of CGH array data

Now that our method for segmentation/clustering has been extensively studied and compared with other methods based on simulation studies, the last chapter of this part is devoted to its application on real array CGH data. For this purpose we use publicly available data sets, which have been described in Nakao *et al.* (2004) and Snijders *et al.* (2001).

12.1 Homogeneous or heterogeneous variances ?

In the segmentation context we discussed the choice of modelling that should be done to determine which parameters of the model were affected by changes. Two choices are possible: a segmentation model with a homogeneous variance or with heterogeneous variances. Based on real data sets, we emphasize the choice for a model with homogeneous variance (see Picard *et al.* (2005) for a complete discussion). More than a pure choice of modelling this question can be biologically interpreted, since a model with homogeneous variance means that the variability presented by gene copy numbers does not depend on the position of the gene on the genome.

In the case of a segmentation/clustering model, mixture model parameters are used to characterize clusters which are supposed to have a biological interpretation. A comparison can be made between the two models such that:

segmentation	segmentation/clustering
σ_k^2	s_p^2
variability on segment k	variability in cluster p

The problem of determining if parameter s_p^2 is constant could be interpreted as follows: is the variability of gene copy-numbers specific to one biological group? We propose to assess this question using a statistical criterion.

We consider the data set described in Nakao *et al.* (2004), which has been already used in Chapter 9 to choose an initialization strategy for the hybrid algorithm. This data set consists in 125 CGH profiles corresponding to 125 patients affected by colorectal cancer. Among these profiles, we focus on chromosomes

Source	DF	Type III SS	Mean Square	F Value	Pr > F
var	1	2034.355465	2034.355465	15421.9	<.0001
patient(chromosome)	374	0.435337	0.001164	0.01	1.0000
var*patient(chromos)	374	3694.354877	9.877954	74.88	<.0001

var	rLpen LSMEAN	t Value	Pr > |t|
heterogeneous	1.33759033	-124.18	<.0001
homogeneous	1.66086557		

Table 12.1: Determining the best modelling strategy using a linear model.

1, 8 and 20. Then the hybrid algorithm is run for $P = 1, \ldots, 6$ clusters and $K = P, \ldots, 20$ segments in the case of a mixture model with homogeneous and heterogeneous variances. The resulting optimized log-likelihoods are used to score each modelling strategy. However, it is clear that when the mixture considers heterogeneous variances, the log-likelihood is greater since the number of parameters is higher than in the case of and homogeneous variance ($3P - 1$ parameters vs. $2P$). Since we aim at assessing the quality of fit of a model whatever its number of parameters, we choose to penalize each log-likelihood according to the number of parameters of the corresponding model. We compare two criteria, which are:

$$\begin{aligned} C_{KP}(s_p^2) &= \log \mathcal{L}_{KP}(\hat{T}, \hat{\psi}) - \frac{1}{2}\log(n) \times (3P - 1), \\ C_{KP}(s^2) &= \log \mathcal{L}_{KP}(\hat{T}, \hat{\psi}) - \frac{1}{2}\log(n) \times (2P). \end{aligned}$$

Then our purpose is to assess if one choice of modelling systematically leads to the best penalized criterion. For this purpose the penalized criteria are ranked for each CGH profile and for each K and P and we use a linear model to determine the average rank given by one method. This strategy has already been used in Chapter 9, and the results are shown in Table 12.1. It can be seen that the mean rank of the penalized criterion is greater when the variance is homogeneous compared with a heteroscedastic model. This indicates that the segmentation/clustering model with a homogeneous variance fits better to real CGH data when adjusted to the number of parameters. Even if this result has a clear modelling interpretation, it appears difficult to discuss its biological relevance since we do not have experimental results for validation. Indeed we could have thought that amplified regions would have shown a greater variability in gene copy numbers compared with non-affected regions. On the contrary, our results suggest that the variability of gene copy numbers seem to be constant whatever the biological status of genomic regions.

12.2 Application to real data sets

In this section we propose to discuss the results of the segmentation/clustering model on real data sets. Unfortunatelly there is no data set for which the biological status of the clones has been determined and validated with a different technique than array CGH. Since the true gene copy number is unknwon, we can

not assess the performance of our method regarding its ability to detect true biological events. This is why we choose to compare the results of our method with other methods such as segmentation methods and HMMs.

12.2.1 Segmentation/clustering vs. segmentation

A first comparison is made between segmentation results and segmentation/clustering. In Figure 12.1 are shown two CGH profiles described in Nakao *et al.* (2004). This first situation is used to illustrate the case where the separability of the clusters is high. In this case, the jump in the mean between clusters is high regarding the variability of the signal. Consequently no breakpoint is removed/added between segmentation and segmentation/clustering results. The interest in the clustering model is that it provides labels to segments. In this case, the interpretation of the biological status of genomic regions is straightforward in terms of deletion and amplification.

The problem of outliers

In Figure 12.2 we show how the segmentation/clustering model can change the segmentation results. In the first example (X38, top) we can see that considering the clustered structure of segments leads to the addition of breakpoints. Let us consider the point located at $t = 33$ for instance. This point is not detected in the segmentation context. Nevertheless, since its value is close to the mean of the deleted cluster ($\hat{m} \simeq -0.7$), a new breakpoint is added and this point is declared as a deleted clone. This leads to the segmentation of the profile into $K = 10$ segments, whereas the segmentation model only considers $K = 6$ segments. In the segmentation context, we discussed the difficulty to identify outliers in the context of array CGH data analysis. Indeed the point located at $t = 33$ and affected to the "deleted" cluster may be interpreted as a real deleted clone, or as an outlier. Nevertheless, the definition of outliers appears to be ambiguous in this case. A first possibility would be that this clone is deleted whereas its close neighbors are not. In this case, the segmentation/clustering model results can be directly interpreted. Another case could be that this clone has been misannotated, meaning that its coordinate on the genome is wrong. In this case, its "true" position could be between $t = 1$ and $t = 17$ for instance. Nevertheless, the analyst is dependent on the informations which are stored in public data bases. To this extent we can not choose between three situations which are: the clone is a real deleted clone at the correct coordinate, the clone is a real deleted clone at the wrong coordinate, the clone is a false positive and the value of the signal is due to technical artefacts. Nevertheless, we can consider that segments of size 1 should be carefully interpreted.

The problem of amplified regions

The second example (X480, Figure 12.2 bottom) illustrates a situation where the variance of the signal is high regarding the jump of means between clusters. The segmentation/clustering method selects $P = 3$ clusters. Once more, considering the clustered structure of segments leads to the addition of many segments compared with pure segmentation methods ($K = 15$ vs $K = 3$). When looking at the results, two clusters have an easy interpretation. The first one (triangles) has a null log-ratio on average, and the third one (+) presents a mean log-ratio close to 0.8, which corresponds to an amplification. As for the second cluster, its interpretation appears difficult. Its mean log-ratio is lower than $\log_2(3/2)$, but different from 0. Nakao *et al.* (2004) propose to make a distinction between gained and amplified regions. To do so, they propose some thresholds such that if $0.225 \leq log_2ratio < 0.9$ the region is considered as gaines, and amplified if $log_2ratio \geq 0.9$. Using these thresholds, we could interpret cluster 2 as the cluster of gained regions, and cluster 3 as the cluster of amplified regions.

Many groups, many segments

One last example that is provided is illustrated in Figure 12.3. In this case, 5 clusters are selected with 13 segments (11 segments for segmentation). This example presents the advantages of the segmentation/clustering method which are the addition of breakpoints when data points are close to one cluster in terms of mean (at position $t = 12$) the labelling of data points which are isolated but which clearly belong to one cluster ($t = 46, 68$), and the distinction between deletion and three types of amplification. This example shows that our method performs well even if the number of clusters is high.

12.2.2 Comparison with hidden Markov models

In the previous part we proposed to compare the performance of our model with HMMs using simulated data. We propose here to discuss the results provided by both methods on real data sets. A first comment is that when the data can be easily interpreted, both methods give similar results. This is why we focus on cases where results are different.

Viterbi vs. Forward/Backward

When using HMMs for segmentation, two methods can be used to reconstruct the sequence of hidden states. The first one is to use the forward/backward algorithm to calculate the posterior probability of each state given the observed data, and the second one is the Viterbi algorithm, which computes the most probable path for the hidden states. Consequently, the forward/backward algorithm gives local results, whereas the Viterbi algorithm provides a global solution. While applied to array CGH data, Fridlyand *et al.* (2004) suggest to use the forward/backward algorithm. In Figure 12.4 is shown the segmentation/clustering results provided by our method and by HMMs with the forward/backward and the Viterbi algorithm. The number of clusters/hidden states is 5 in both cases,

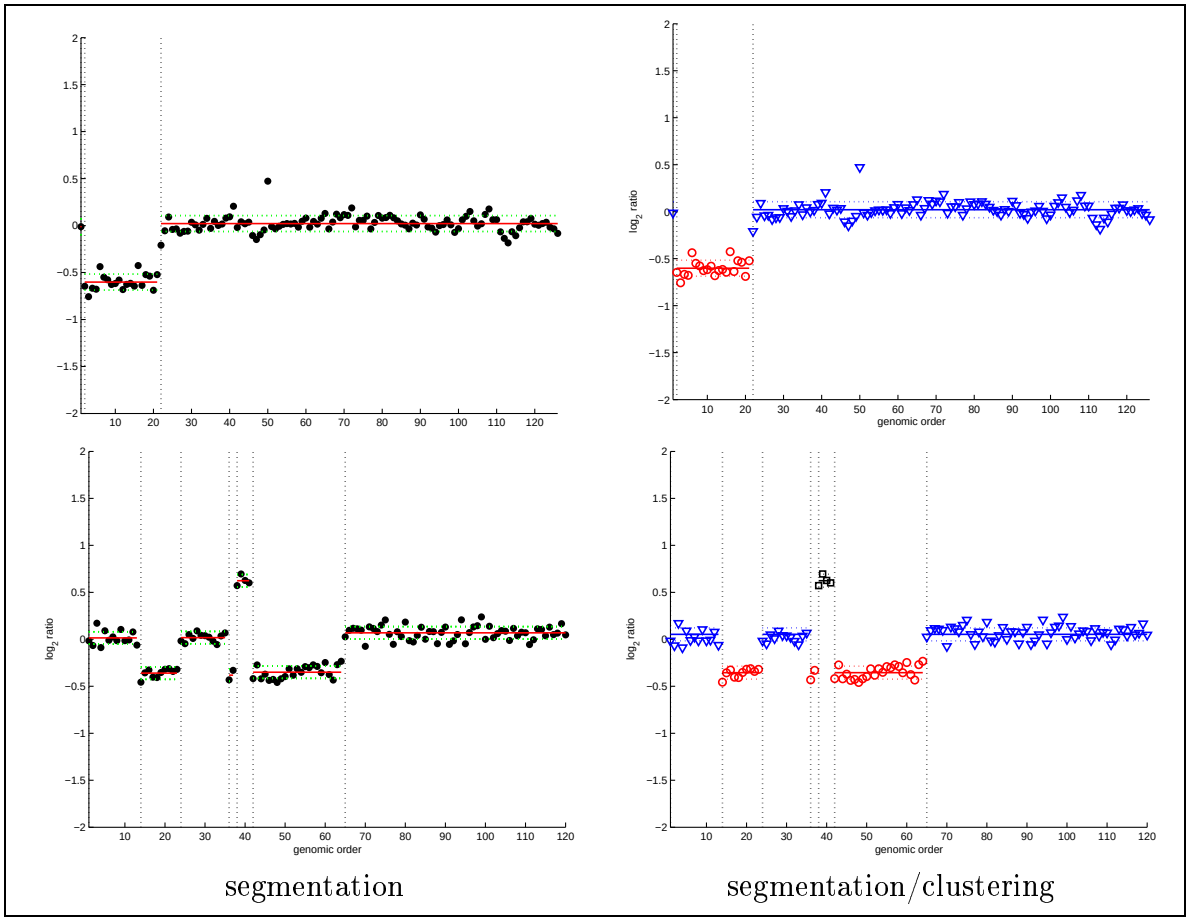


Figure 12.1: Segmentation vs segmentation/clustering for data set Nakao, chromosome 1 experiments X411 and X524.

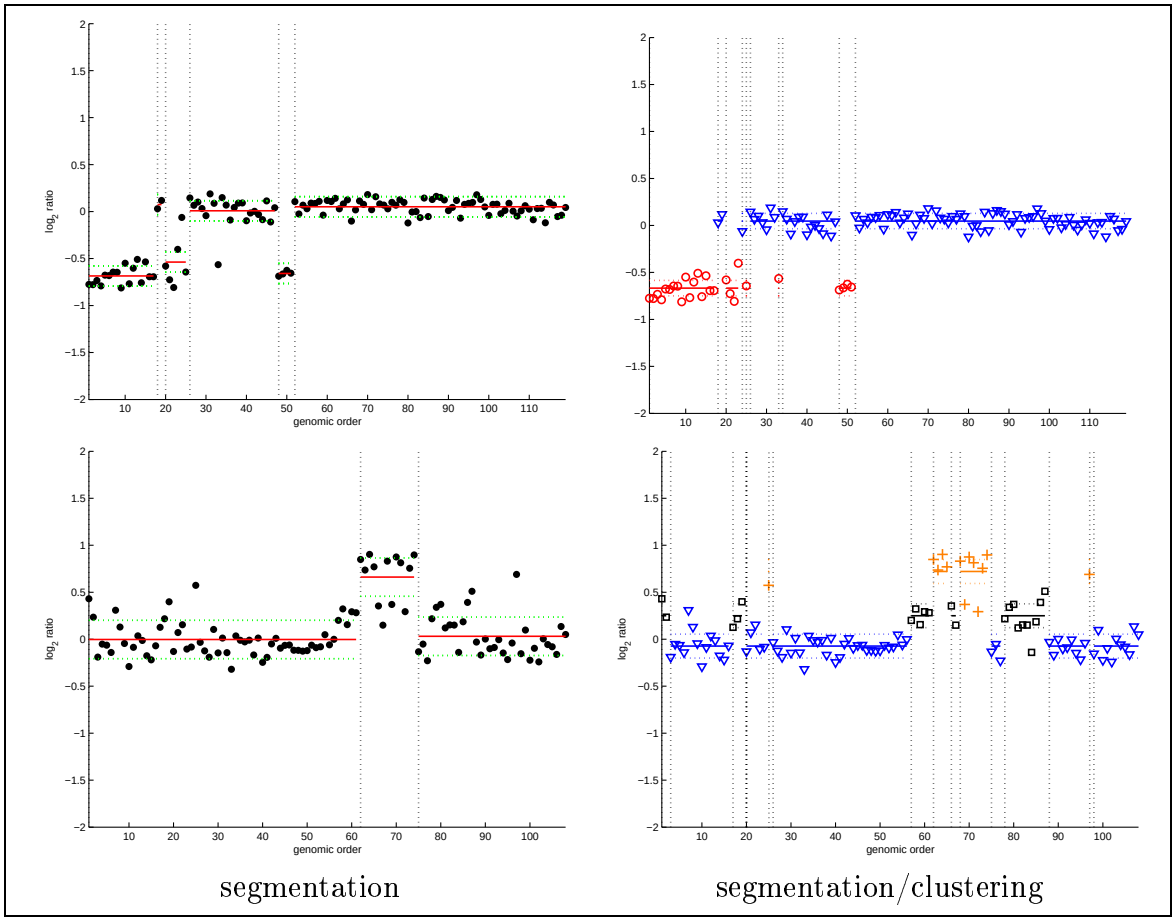


Figure 12.2: Comparison segmentation vs segmentation/clustering for data set Nakao, chromosome 1 experiments X38, X480.

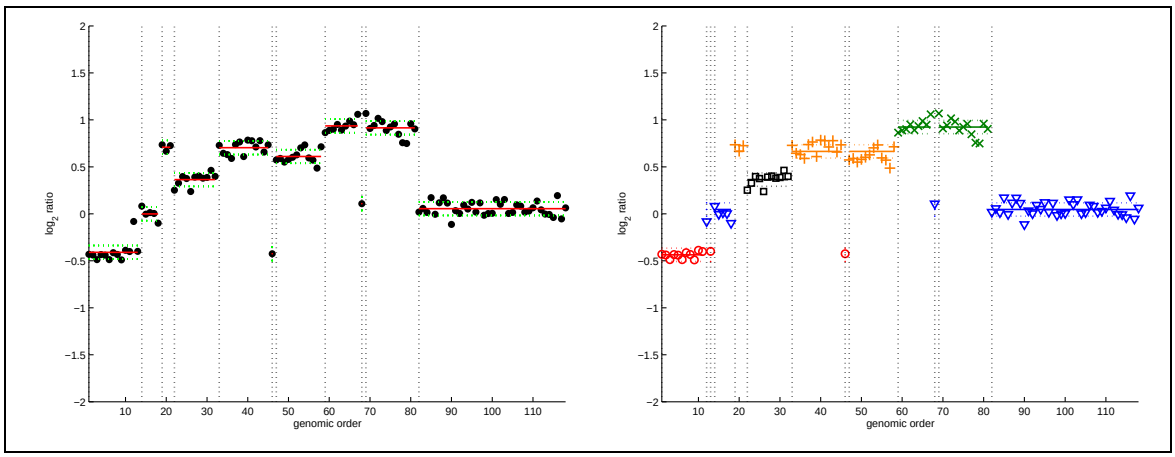


Figure 12.3: Segmentation vs segmentation/clustering for data set Nakao, chromosome 8 experiment X54.

and the results are globally similar. Nevertheless, it can be seen that the forward/backward and the Viterbi algorithm do not provide the same clustering results. In the case of the Viterbi algorithm, single points ($t = 46, 57, 68$) are clustered into the "normal" group (triangles) whereas they are affected to distinct clusters with forward/backward. This illustrates the difference between a local and a global strategy to recover the hidden sequence. In practice the local strategy is prefered to cluster data points into groups (see Fridlyand *et al.* (2004)).

As already mentioned in simulation studies, for an identical number of hidden states, the number of segments is higher for HMMs compared with our model. In Figure 12.4 we can see that breakpoints are added at positions $t = 58$ and $t = 79$, and the label of the associated regions are changed. Nevertheless, since we do not have any criterion to determine which method is right, we can not conclude on those differences. However, one remark could be that HMMs may be more powerful than our model, leading to more detailed profiles. On the contrary our method seems to be more conservative. This is illustrated in the next example.

Type I errors vs. type II errors

The last examples that are considered to compare HMMs and our model is presented in Figures 12.5 and 12.6. In Figure 12.5 it can be seen that the number of clusters is similar, but the number of segments is higher when using HMMs. One characteristics of HMMs is that many small jumps in the signal are associated with a change in the label of the region. Consequently HMMs appear to focus on label changes rather than on the spatial structure of the signal. On the contrary our method considers that local variability within segments is not necessarily associated to a change in the label. This behavior is striking in Figure 12.5 but is also illustrated in Figure 12.6 where the number of hidden states is 4 for HMMs and 2 in our case. This ability to detect local changes in the label of individual data points indicates that HMMs show a high statistical power, whereas our method is more conservative.

From a biological point of view, it would be crucial to determine if changes in gene copy numbers affect isolated clones, or if the process of deletion/amplification concerns chromosomal regions. Moreover it should be recalled that increasing the power of a method may lead to an increase in false positives, which may not be suitable. False positive errors can be viewed as the "curse" of microarrays studies. Indeed when studying gene copy-numbers for thousands of clones, each positive result should be verified with another technique, which may be expensive. In microarray studies technical artifacts constitute one major source of false positives. Among those is the well known dye-effect which can be corrected by statistical methods in the case of expression profile microarrays. Since this normalization procedure has never been adapted to the case of array CGH, a question could be to determine if the local variations detected by HMMs are due to technical artifacts or not.

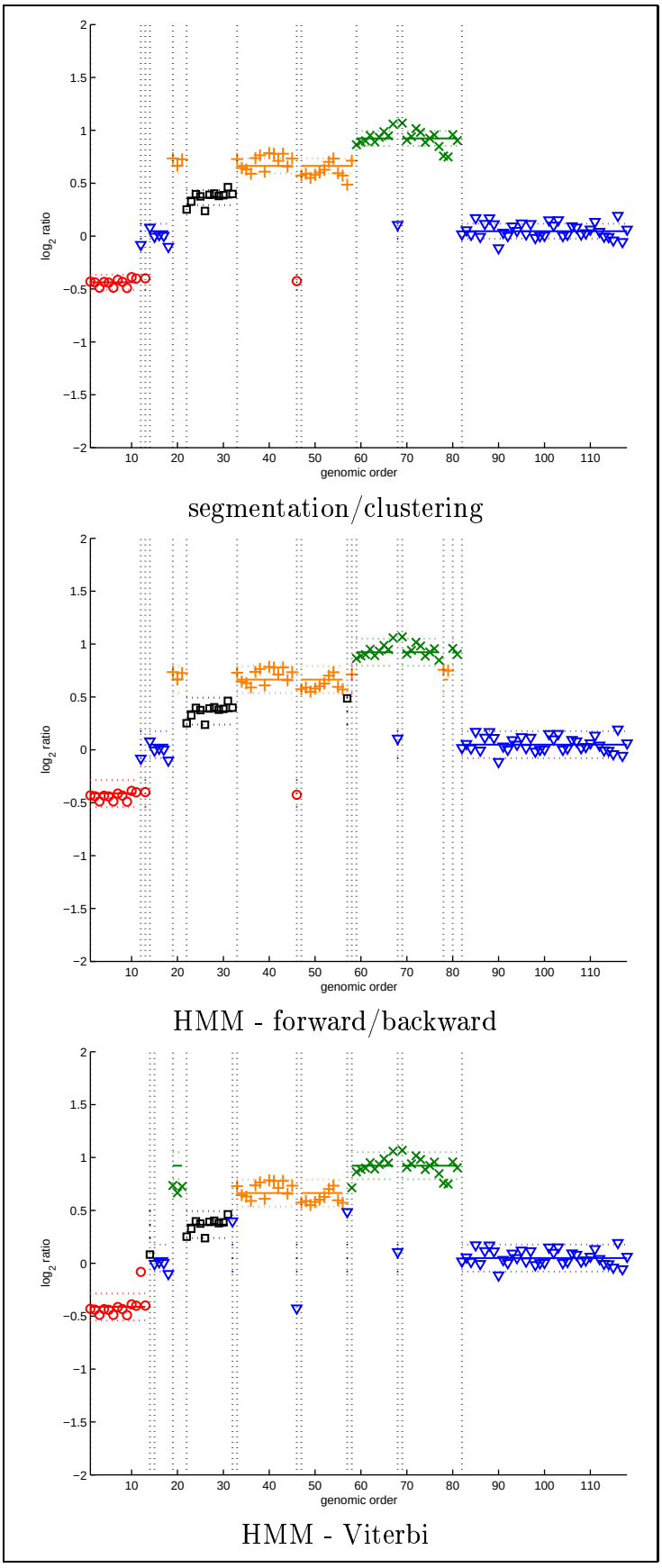


Figure 12.4: Comparison between segmentation/clustering and HMMs-1 (Nakao data set chromosome 8 experiment X54).

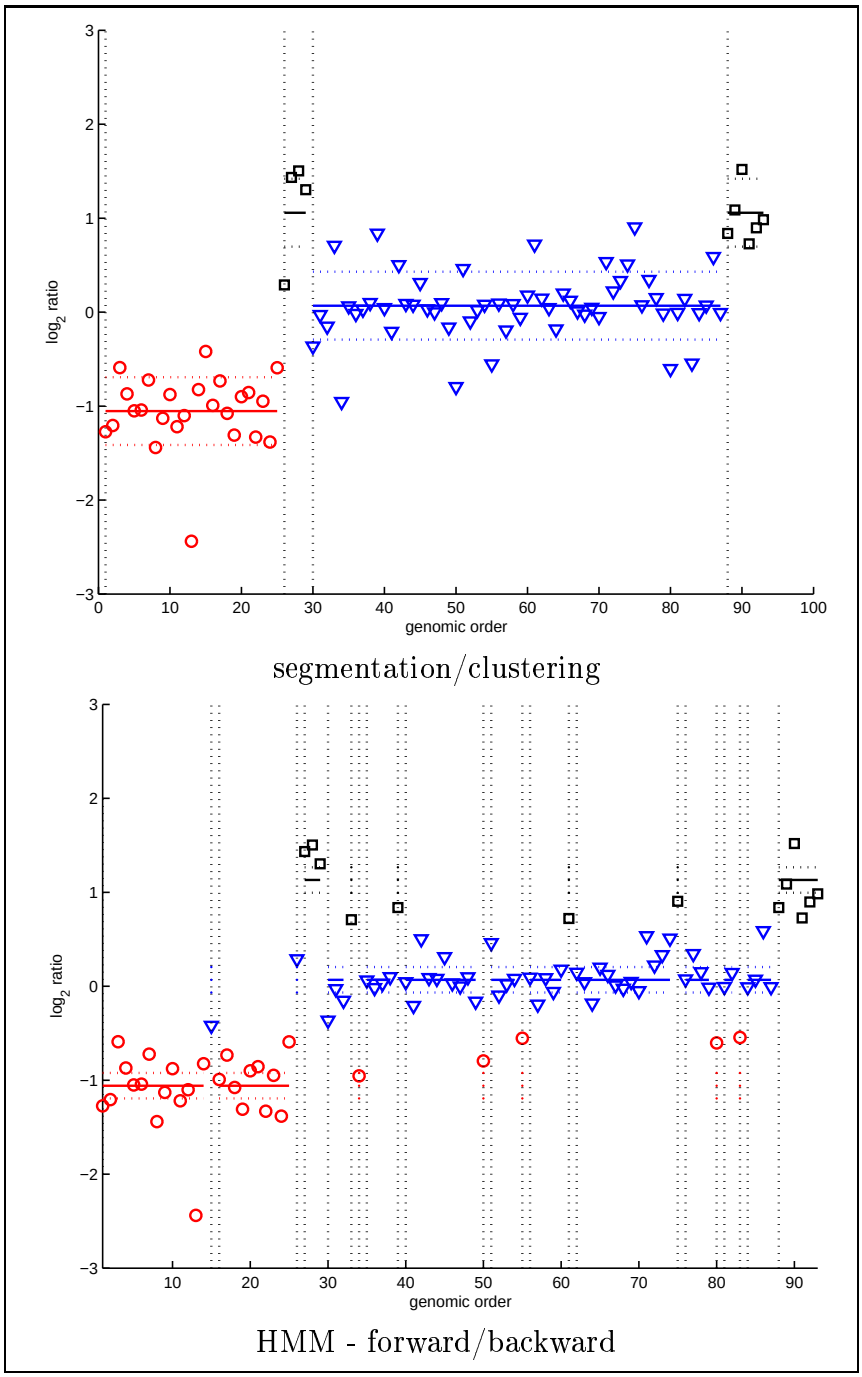


Figure 12.5: Comparison between segmentation/clustering and HMMs-2. Data set described in Snijders *et al.* (2001), Bt474 cell lines, chromosome 9.

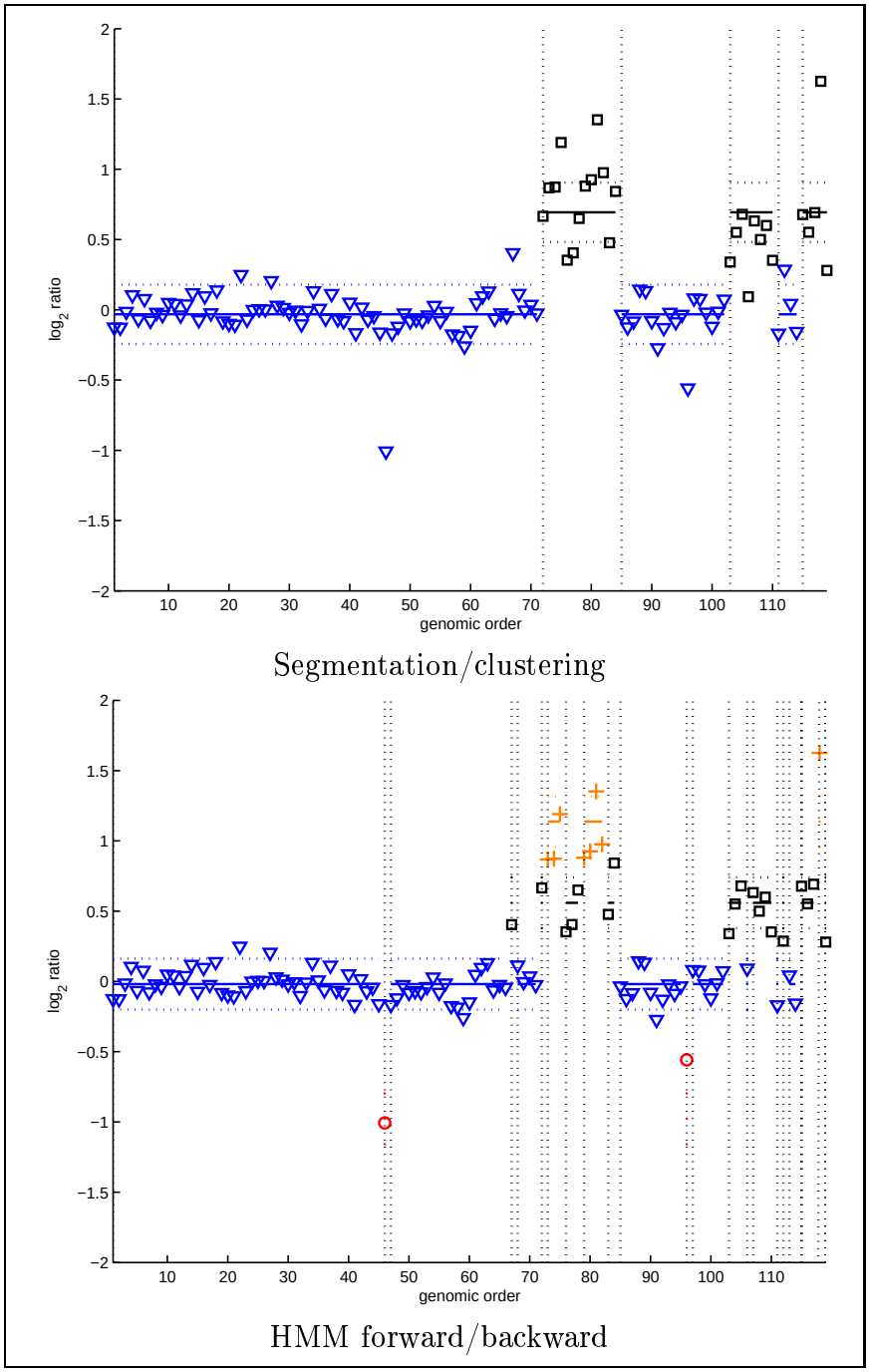


Figure 12.6: Comparison between segmentation/clustering and HMMs-3. Data set described in Snijders *et al.* (2001), Bt474 cell lines, chromosome 1.

12.3 Future prospects for array CGH data analysis

In this last section we would like to provide guidelines for future developments concerning array CGH data analysis. As previously mentioned finding chromosomal alterations in CGH profiles strongly depends on the quality of the data, since small changes in the signal may be due to technical variability and may not reflect biological events. To this extent it appears crucial to develop normalization procedures for array CGH data.

In Part I we discussed the specificity of array CGH data, and we explained why normalization procedures could not be directly applied to these data. Let us recall that gene copy numbers are measured through fluorescence intensities between two conditions. One condition is labelled in green and the other in red. A dot on a CGH profile corresponds to the log-ratio $M = \log_2(R/G)$. A common problem in microarray experiments is that the log-ratio often depends on the mean intensity noted $A = \log_2(R \times G)$. Then the Loess method is used to correct this intensity-dependent dye bias (see Part I for further details). Nevertheless it appears that this normalization step can not be performed for array CGH data for two major reasons:

- an important proportion of the genome is altered by chromosomal aberrations,
- log-ratio values are centered around mean log ratios for each biological class (deleted, normal, amplified).

Since our segmentation/clustering procedure provides labels for each chromosomal region, we can isolate the regions of the genome which show amplifications or deletions, and we can estimate the mean log-ratio for each biological group. In Figure 12.7 is shown a MA plot with data points being labelled according to the segmentation/clustering results.

As a perspective of the analysis of array CGH data, it would be interesting to perform a loess normalization on each group detected by the segmentation/clustering model. A question will be to assess whether this normalization step changes the segmentation/results or not. An idea could be to run iteratively the normalization step and the analysis step in order to provide results which should be less corrupted by technical artifacts.

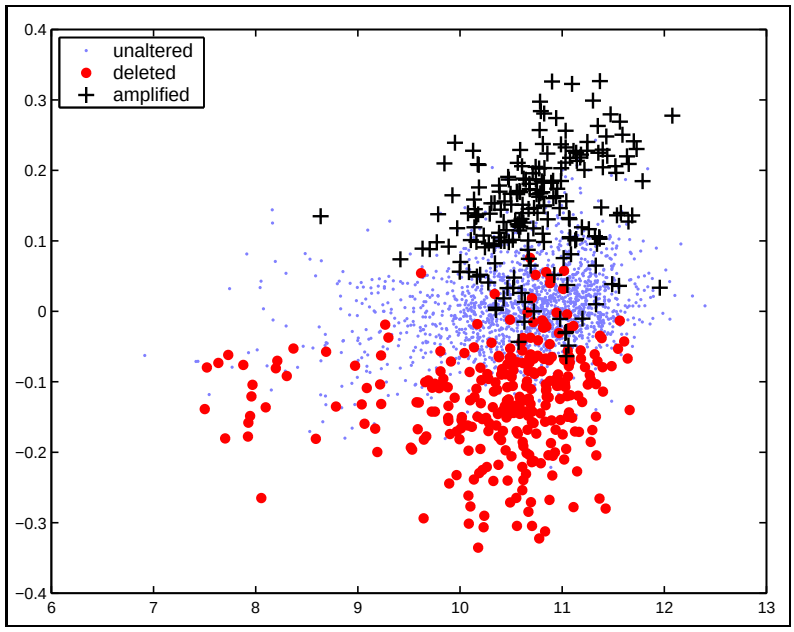


Figure 12.7: MA-plot after segmentation/clustering.

Conclusion

In this part, we proposed to implement the segmentation/clustering method and to assess its performance on simulated and real data sets, and to compare it with existing methods whose objectives are similar. The originality of our method is that it requires the selection of both the number of clusters and segments. We proposed a heuristic for this new double-selection problem, and we showed that this procedure was efficient. An interesting perspective would be to develop theoretical criteria for this problem, and to assess the statistical properties of the estimators we proposed.

From a modelling point of view, an interesting development would be to consider a segmentation/clustering model for which the variability is independent of the group but depends on segments. In other words, we could develop a model such that:

$$\forall t \in I_k, \quad Y_t | Z_p^k = 1 \sim \mathcal{N}(m_p \mathbb{1}_{n_k}, \sigma_k^2 I_{n_k}).$$

the advantage of such model would be to consider that the level of segments depends on the clusters whereas the variability depends on the position of the segments. This model could be more flexible compared with the segmentation/clustering model we propose.

While compared with HMMs, our method shows slightly better performance on simulated data. However when applied to real CGH data, it appears that our method is more conservative, and less sensitive to small changes in the signal that may be due to technical artifact. Consequently the segmentation/clustering model appears to be an interesting alternative to HMMs.

From a practical point of view, we are currently developing a software program for the analysis of array CGH data. Even if our method is applied to this particular data, we claim that it can be applied in a the more general setting of signal processing. In the following, we propose an extension of our model to discrete variables, with an application to the analysis of DNA sequences.

Part V

Segmentation/clustering for the analysis of biological sequences

Introduction

In Parts III and IV we proposed a new statistical model for segmentation/clustering problems in the Gaussian case. These developments where oriented towards the analysis of array CGH data. In this last part we propose another application of our segmentation/clustering model devoted to the analysis of a different type of biological data which are biological sequences, and especially DNA sequences.

Nature of biological sequences

The basics of molecular biology has been summarized in a concept called the Central Dogma of Molecular Biology (Figure 12.8). This dogma aims at describing existing relationships between biological macromolecules, which are DNA, RNA and proteins. DNA [1] molecules are the basic carriers of genetic information and are found in all living cells. A DNA sequence is made up of a string of bases which are Adenosine (A), Thymine (T), Cytosine (C), and Guanine (G). These bases are attached to a sugar-phosphate backbone. The succession of these letters constitutes the complete genetic information defining the structure and function of an organism. Proteins can be viewed as effectors of the genetic information contained in DNA coding sequences. They are made up of a string of 20 different amino acids which is formed using the genetic code to convert the information contained in the 4 letter alphabet of the DNA sequence into a new alphabet of 20 amino acids. This *translation* procedure requires an intermediate step in eukariotyc cells called *transcription*. During transcription a DNA segment is read and transcribed into a single stranded molecule of RNA[2] whose chemical composition is similar (the 4 letter alphabet remains with the replacement of Thymine molecules by Uracyle molecules). RNAs which contain information to be translated into proteins are called messenger RNAs (mRNAs), but other types of RNAs are created during transcription, such as ribosomal RNAs (rRNAs) and transfert RNAs (tRNAs). The last step consists in the translation of mRNAs into proteins. A shematic representation of these relationships is given in Figure 12.8. Even if they are characterized by different functions, biological macromolecules share one commun feature: they are constituted of oriented sequences of letters (in different alphabets) whose succession constitutes a biological information, and whose variations generate biological diversity. Consequently the local composition of macromolecules (in base pairs for nucleic acids or in amino acids for proteins) consitutes an information itself.

[1] DNA: Desoxyribo Nucleic Acid
[2] DNA: Ribo Nucleic Acid

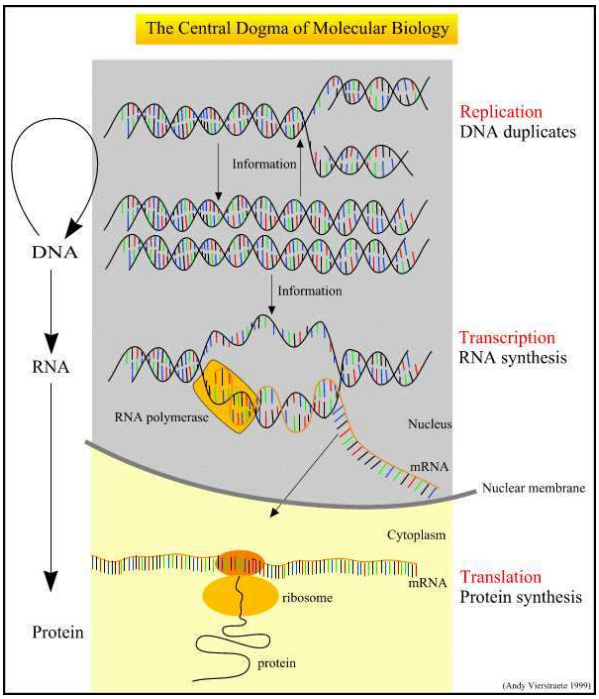


Figure 12.8: The central dogma of molecular biology

Considerable effort has been made in the collection and in the dissemination of DNA sequence informations, through initiatives such as the Human Genome Project [3]. The explosion of sequence-based informations is illustrated by the sequencing of the genome of more than 800 organisms, which represents more than 3.5 million genetic sequences deposited in international repositories (Butte and Atul (2002)). The aim of this first phase of the genomic area consisted in the elucidation of the exact sequence of the nucleotides in DNA molecules, which has allowed the search for functional sequences diluted all along the genomes.

Statistical analysis of biological sequences

One efficient way for identifying and screening for structure in biological sequences is to use statistical techniques. Of particular interest are techniques which capture the evolution of the sequence composition along the molecule. Variations in the base-pair composition may constitute a biological signal, such as a richness is Guanine and Cytosine nucleotides in DNA sequences, which can be linked to banding patterns in mammalian chromosomes (Ikemura *et al.* (1990)), or the succession of three letters (or codons in DNA sequences) which can indicate the termination of transcription (STOP codons). To this extent, segmenting DNA sequences into regions which are compositionally different from the rest of the sequence appears of primary interest.

The problem of segmenting DNA sequences is far from new and has focused

[3] `http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml`

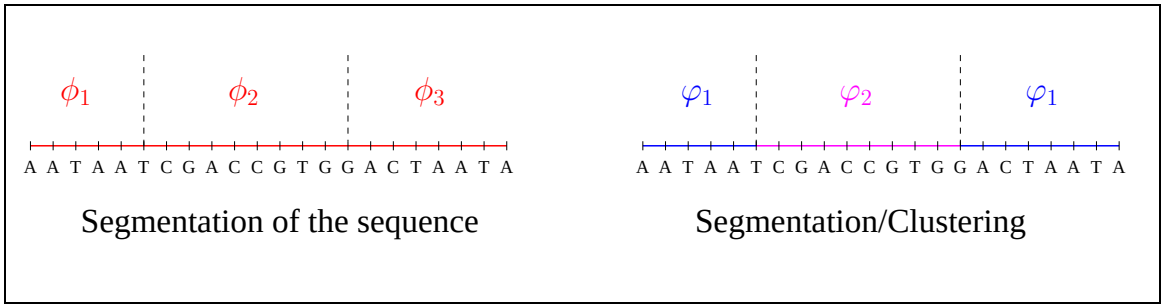


Figure 12.9: Principle of the segmentation/clustering model for DNA sequences.

much attention in the last four decades. Many statistical techniques have been considered, such as Hidden Markov Models, Scan statistics, Bayesian techniques and change-point methods. A complete review of these methods can be found in Braun and Muller (1998). In this work we consider the DNA segmentation problem in the framework of the multiple change-point problem for categorial data. In this context the data are assumed to be drawn from binomial or multinomial distributions (Braun and Muller (1998), Braun *et al.* (2000)), and Lebarbier (2002) recently developed the case of multiple changes in Markov chains which will be presented in Chapter 13. Figure 12.9 (left) presents an illustration of this model, where the data are assumed to be drawn from a Markov chain whose transition matrix is supposed to be affected by abrupt changes at unknown coordinates. Modelling DNA sequences by Markov chains is thought to describe the dependency that exists between nucleotides within a sequence. A complete discussion on DNA sequence modelling can be found in Nicolas (2003), Robin *et al.* (2003), and Schbath (2000).

A Segmentation/clustering problem

In addition to the hypothesis that DNA segments should be of homogeneous composition within segments and of heterogeneous composition between segments, another hypothesis can be made. Functional patterns are likely to be repeated along the DNA molecules in terms of function and composition. For instance there exists an alternance between G+C rich regions and G+C poor regions, and also an alternance between coding and non-coding regions. Since base-pair composition can be linked to biological function, it is likely that segments are structured into a finite number of clusters which could be interpreted as clusters of functional interest. This recalls the formulation of the segmentation/clustering problem which has been studied in the Gaussian case previous parts.

In Chapter 13 we propose to apply our method for the analysis of DNA sequences, with the development of a segmentation/clustering model for Markov Chains. Since the methodology has been extensively detailed in the Gaussian case, we choose to give the main steps of the construction of the model only. This part is devoted to the application of our method to real DNA sequences. One main advantage when working on DNA sequences is that some biological information is already available through genome annotations, which provide the positioning of functional elements along the sequence. These annotations will be used to assess the performance of our method. In this Chapter, we will also compare our method to Hidden Markov Models which constitute a method of choice for DNA sequence analysis (Muri (1997), Nicolas (2003)).

Chapter 13

Application of the segmentation/clustering model to Markov chains

13.1 Multiple changes in Markov chains

The construction of the segmentation/clustering model is linked to the underlying segmentation model. In this section we aim at presenting the main results developed in Lebarbier (2002) which will be used to construct a segmentation/clustering model for categorial data.

13.1.1 Presentation of the model

Let $\{Y_1, \ldots, Y_n\}$ denote a sequence of dependent categorial variables with Y_t taking values in a finite set of integers $\mathcal{Y} = \{1, \ldots, r\}$, with $r \geq 2$. The dependency between variables is modelled through a Markov chain of order m, m being fixed. In the multiple change-point context, we suppose that there exists a sequence of breakpoints $T = \{t_1, \ldots, t_{K-1}\}$ which defines a partition of the data into K intervals noted $I_1, \ldots, I_K$ in which the transition matrix and the initial distribution of the Markov chain are such that:

$$\forall t \in I_k \qquad \Pr\{Y_t = b | Y_{t-1} = a_1, \ldots, Y_{t-m} = a_m\} = \phi^k(a_1, \ldots, a_m; b),$$

for $(a_1, \ldots, a_m, b) \in \mathcal{Y}^{m+1}$ and with the condition

$$\sum_{b \in \mathcal{Y}} \Pr\{Y_t = b | Y_{t-1} = a_1, \ldots, Y_{t-m} = a_m\} = 1. \tag{13.1}$$

As for initial distributions, we suppose that:

$$\Pr\{Y_{t_{k-1}+1} = a_i\} = \alpha^k(a_i).$$

Lebarbier (2002) emphasizes the fact that initial distributions constitute nuisance parameters, and this is why they are not considered in the following, using a partial likelihood, which will be abusively called likelihood for simplicity. Denoting

$\mathcal{M}_{(i)\times(j)}([0,1])$ the set of matrices of dimension $i \times j$ taking values in $[0,1]$, we note

$$\Phi = \{\phi^1, \phi^2, \ldots, \phi^k, \ \ \phi^i \in \mathcal{M}_{(r^m)\times(r)}([0,1]) \ \text{ verifying condition } \ (13.1)\},$$

the set of transition matrices of the model, the likelihood of the model is:

$$\mathcal{L}_K(Y; T, \Phi) = \prod_{k=1}^{K} \prod_{a_1,\ldots,a_m,b \in \mathcal{Y}^{m+1}} \phi^k(a_1, \ldots, a_m; b)^{N^k(a_1,\ldots,a_m,b)},$$

where $N^k(a_1, \ldots, a_m, b)$ is the counting of the word $\{a_1 \ldots a_m b\}$ in sequence Y^k defined by

$$N^k(a_1, \ldots, a_m, b) \ = \ \sum_{t=t_{k-1}+1}^{t_k - m} \mathbb{1}\{Y_t = a_1, \ldots, Y_{t+m-1} = a_m, Y_{t+m} = b\}.$$

13.1.2 Estimation

Parameters T and Φ are jointly estimated by maximum likelihood, and the log-likelihood is optimized using a dynamic programming algorithm. Denoting $\ell(i,j)$ the local log-likelihood calculated on segment Y^{ij} with i, j as starting and ending points, we have:

$$\ell(i,j) = \sum_{a_1,\ldots,a_m,b \in \mathcal{Y}^{m+1}} N^{ij}(a_1, \ldots, a_m, b) \log \widehat{\phi}^{ij}(a_1, \ldots, a_m; b), \qquad (13.2)$$

with

$$\begin{aligned}
N^{ij}(a_1, \ldots, a_m, b) &= \sum_{t=i+1}^{j} \mathbb{1}\{Y_t = a_1, \ldots, Y_{t+m-1} = a_m, Y_{t+m} = b\}, \\
N^{ij}(a_1, \ldots, a_m, +) &= \sum_{b \in \mathcal{Y}} N^{ij}(a_1, \ldots, a_m, b), \\
\widehat{\phi}^{ij}(a_1, \ldots, a_m; b) &= \frac{N^{ij}(a_1, \ldots, a_m, b)}{N^{ij}(a_1, \ldots, a_m, +)}.
\end{aligned}$$

$\widehat{\phi}^{ij}(a_1, \ldots, a_m; b)$ is the transition probability on segment Y^{ij}, and $N^{ij}(a_1, \ldots, a_m, +)$ is the counting of word $a_1 \ldots a_m$ of size m in segment Y^{ij}.

Then the dynamic programming algorithm is used to estimate the breakpoints as shown in Chapter 4. Unfortunately dynamic programming can not be run if the size of the sample is overly high since it requires the storage of a cost matrix which is $n \times n$ dimensional. In DNA sequence analysis, the size of the data set is huge, with $n \geq 10,000$ in most cases. For instance the genome of Bacteriophage lambda, which is a "small" genome is constituted of $48,502$ base-pairs. In the Gaussian case Gey and Lebarbier (2002) proposed to adapt the CART algorithm to detect jumps in the mean for large samples, and Lebarbier (2002) proposes to adapt this algorithm to the case of categorial variables.

13.2 Segmentation/Clustering in the case of Markov Chains

In the context of segmentation/clustering, we suppose that there exists a secondary structure of the data, which is the belonging of segments into a finite number of clusters. Then the segmentation/clustering model which has been studied in the Gaussian case can be adapted to the case of categorial variables as follows. Let us note $\{Z^1, \ldots, Z^K\}$, with $Z^k = \{Z_1^k, \ldots, Z_P^k\}$, a sequence of independent categorial variables indicating the belonging of segments $\{Y^1, \ldots, Y^K\}$ to P possible clusters. Then we model the dependency between letters within segment Y^k with a Markov chain of order m such that:

$$\forall t \in I_k \ \Pr\{Y_t = b | Y_{t-1} = a_1, \ldots, Y_{t-m} = a_m, Z_p^k = 1\} = \phi_p(a_1, \ldots, a_m; b),$$

with $\phi_p(a_1, \ldots, a_m; b)$ being the transition probability characterizing cluster p.

Then we define *prior* and *posterior* probabilities of membership of segment k to cluster p such that:

$$\begin{aligned} \pi_p &= \Pr\{Z_p^k = 1\}, \\ \tau_p^k &= \frac{\pi_p \Pr\{Y^k | Z_p^k = 1\}}{\sum_\ell \pi_\ell \Pr\{Y^k | Z_\ell^k = 1\}}, \end{aligned}$$

with notation:

$$\Pr\{Y^k | Z_p^k = 1\} = \prod_{a_1, \ldots, a_m, b \in \mathcal{Y}^{m+1}} \phi_p(a_1, \ldots, a_m; b)^{N^k(a_1, \ldots, a_m, b)}.$$

Segment Y^k will be affected to a cluster with the Maximum A Posteriori rule (MAP). The parameters of this model are the breakpoint coordinates T and the parameters of the mixture noted ψ such that:

$$\psi = \{\pi_1, \ldots, \pi_p; \phi_1, \ldots, \phi_P\}.$$

13.2.1 Running the hybrid algorithm in the case of Markov Chains

This algorithm is run for a fixed number of clusters and segments. Here we show how mixture model parameters are estimated when breakpoints are fixed, and how the breakpoints are estimated when the mixture parameters are fixed. The structure of the hybrid algorithm is similar to the one presented in Chapter 7.

Estimating mixture model parameters

In the incomplete-data framework we define the incomplete and complete-data likelihoods of the model such that:

$$\log \mathcal{L}_{KP}(Y; T, \psi) = \sum_{k=1}^{K} \log \left\{ \sum_{p=1}^{P} \pi_p \prod_{a_1, \ldots, a_m, b \in \mathcal{Y}^{m+1}} \phi_p(a_1, \ldots, a_m; b)^{N^k(a_1, \ldots, a_m, b)} \right\}$$

$$\log \mathcal{L}^c_{KP}(Y,Z;T,\psi) = \sum_{k=1}^{K}\sum_{p=1}^{P} Z_p^k \left\{ \log \pi_p + \sum_{a_1,\ldots,a_m,b\in\mathcal{Y}^{m+1}} N^k(a_1,\ldots,a_m,b)\log \phi_p(a_1,\ldots,a_m;b) \right\}$$

Then an EM algorithm can be run to estimate mixture parameters when the breakpoints are fixed. The resulting estimators are:

$$\begin{aligned} \hat{\pi}_p^{(h+1)} &= \frac{\sum_{k=1}^K \tau_p^{k(h)}}{K}, \\ \widehat{\phi}_p^{(h+1)}(a_1,\ldots,a_m;b) &= \frac{\sum_{k=1}^K \tau_p^{k(h)} N^k(a_1,\ldots,a_m;b)}{\sum_{k=1}^K \tau_p^{k(h)} N^k(a_1,\ldots,a_m,+)}. \end{aligned}$$

Note that $\sum_{k=1}^K \tau_p^{k(h)} N^k(a_1,\ldots,a_m;b)$ represents the weighted counting of word $a_1\ldots a_m b$ in cluster p.

Estimating breakpoints

As for the estimation of the breakpoint coordinates, the dynamic programming method still holds using

$$\ell(i,j) = \log\left\{ \sum_{p=1}^{P} \pi_p \prod_{a_1,\ldots,a_m,b\in\mathcal{Y}^{m+1}} \log \phi_p(a_1,\ldots,a_m;b)^{N^{ij}(a_1,\ldots,a_m,b)} \right\} \quad (13.3)$$

as a local log-likelihood when the parameters of the mixture are fixed. Nevertheless, the size of the data being overly high, we propose to use a preliminary segmentation based on the CART algorithm, in order to reduce the computational load of the method.

13.2.2 Initializing the hybrid algorithm

Need for a preliminary segmentation

Since the size of the data is large when analyzing DNA sequences, we propose to use the growing step of the CART algorithm to provide a preliminary segmentation that will be used by the downstream hybrid algorithm. The principle of this method is to reduce the size of the initial data set, using a segmentation method which provides K_{cart} segments with $K_{cart} << n$. Consequently the role of the CART algorithm is to restrict the collection of visited partitions, as shown in Figure 13.1.

The computational schema of this step is as follows:

- Compute the change-point $\hat{t}_c$ such as $\hat{t}_c = \underset{j}{\text{Argmax}}\,\{\ell(1,j)+\ell(j+1,n)\}$. with $\ell(1,j)$ and $\ell(j+1,n)$ defined as in Equation (13.2). The objective of this step is to find the first best partition of $\{1,\ldots,n\}$ into 2 segments.

Organism	gene size (kb)
Yeast	1.4
Nematods	2.7
Drosophila	3
A. Thaliana	2.1
Human	28

Table 13.1: Average gene size for different organisms.

- Apply the same procedure on the new defined segments, and so on until the number of points within each resulting segment is smaller than a given threshold $lmin$.

At the end of the growing step of the CART algorithm the data have been preliminary segmented, and the hybrid algorithm is run on $\{Y_{cart}^1, \ldots, Y_{cart}^{K_{cart}}\}$. Note that no pruning step is necessary in this context since irrelevant breakpoints will be removed by the downstream dynamic programming during the hybrid algorithm. Consequently CART gives the finest segmentation that can be reached by segmentation/clustering.

One major draw-back of this method is that the segments found by the CART method will not be split by the hybrid algorithm. This is why parameter $lmin$ is important since it determines the minimum size of segments to be split during the CART algorithm. This parameter should be fixed at a low value ($lmin = 500, 1000$ for instance) in order to obtain a reasonnably fine segmentation to start the hybrid algorithm. Interestingly this parameter can be set using some prior biological knowledge. If we are interested in the screening of genes for instance, their average length could be used to set this parameter. An example is given in Table [1] 13.1 which presents the average size of genes for different organisms.

Initializing breakpoint parameters

Once the preliminary segmentation has been provided by the CART algorithm, breakpoints still need to be initialized for a given number of segments. To do so we use the segmentation method developed be Lebarbier (2002) to provide candidates for $T^{(0)}$.

Initializing mixture model parameters

In the previous part (Chapter 9), we chose to use a hierarchical clustering method to initialize mixture parameters in the Gaussian case. This method is based on the local optimization of a classification likelihood at each step. The same method can be applied to the case of Markov chains to provide parameter candidates $\pi^{(0)}$ and $\phi^{(0)}$. In the following, we will note $N^k(w)$ the counting of word $w \in \mathcal{Y}^{m+1}$ in segment Y^k, and $N_+^k(w)$ the summation of this counting with

[1] from `http://www.genoscope.cns.fr/externe/Francais/Sequencage/`

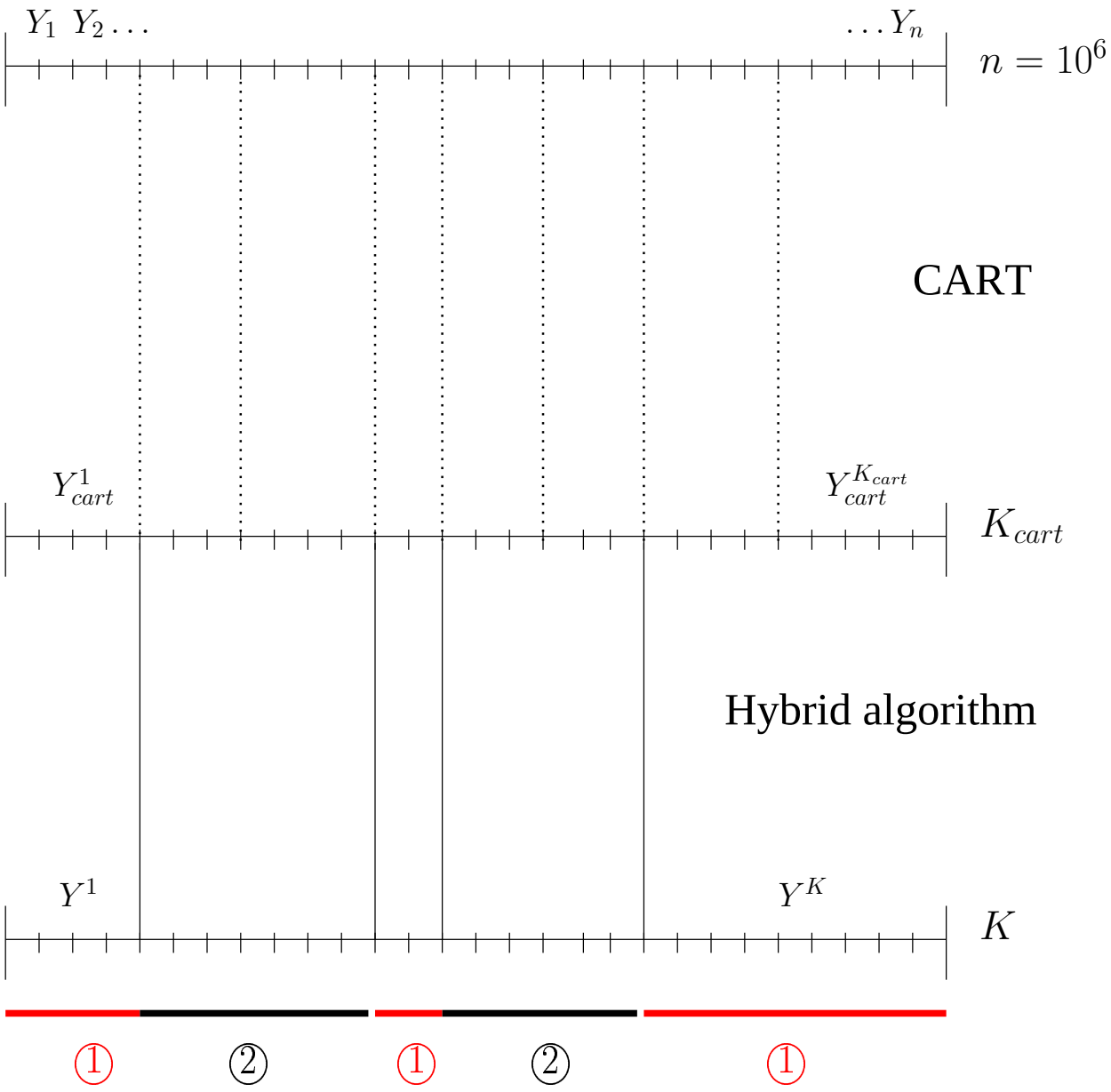


Figure 13.1: Principle of the presegmentation using CART.

respect to the last letter of word w. Then we define the following distance to compare clusters i and j:

$$\begin{aligned} d(C_i, C_j) &= \sum_{w \in \mathcal{Y}^{m+1}} N^i(w) \log \left[\frac{N^i(w)}{N^i_+(w)} \right] + N^j(w) \log \left[\frac{N^j(w)}{N^j_+(w)} \right] \\ &\quad - \sum_{w \in \mathcal{Y}^{m+1}} [N^i(w) + N^j(w)] \times \log \left[\frac{N^i(w) + N^j(w)}{N^i_+(w) + N^j_+(w)} \right]. \end{aligned}$$

13.2.3 Model Selection

Once the parameters have been estimated, the last step is the selection of P and K. We propose to apply the same model selection procedure defined in the Gaussian case. The log-likelihood of the model still shows the same behavior. It can decrease when P is fixed and K increases, whereas it increases with P, as shown in Figure 13.2. In this case the size of the data set is large meaning that the number of segments for which the log-likelihood decreases can be large.

For the estimation of P we propose to use the sequence of increasing log-likelihoods $\left\{ \log \widetilde{\mathcal{L}}_P \right\}_P$ which is shown is Figure 13.2 (bottom). The choice of P is done using an adaptive strategy.

The the purpose is to choose K when P has been chosen. In the Gaussian case (Part IV) we proposed to use a penalty such that:

$$\hat{K}_{\hat{P}} = \underset{K}{\text{Argmax}} \left\{ \log \mathcal{L}_{KP}(\hat{T}, \hat{\psi}) - \frac{K}{2} \log(n) \right\}, \tag{13.4}$$

with n the size of the data set. This choice was done in comparison with an adaptive method which has been shown to be unstable when the size of the data set is small and when the number of segments is small regarding the number of clusters. However, in the context of DNA sequences, the size of the data set is large and the behavior of the likelihood is smooth according to the number of segments. Moreover, if n is large, penalty 13.4 can not be used, since it is overly high compared with the log-likelihood to penalize. This is why we propose to use the adaptive strategy to select the number of segments defined in Chapter 8.

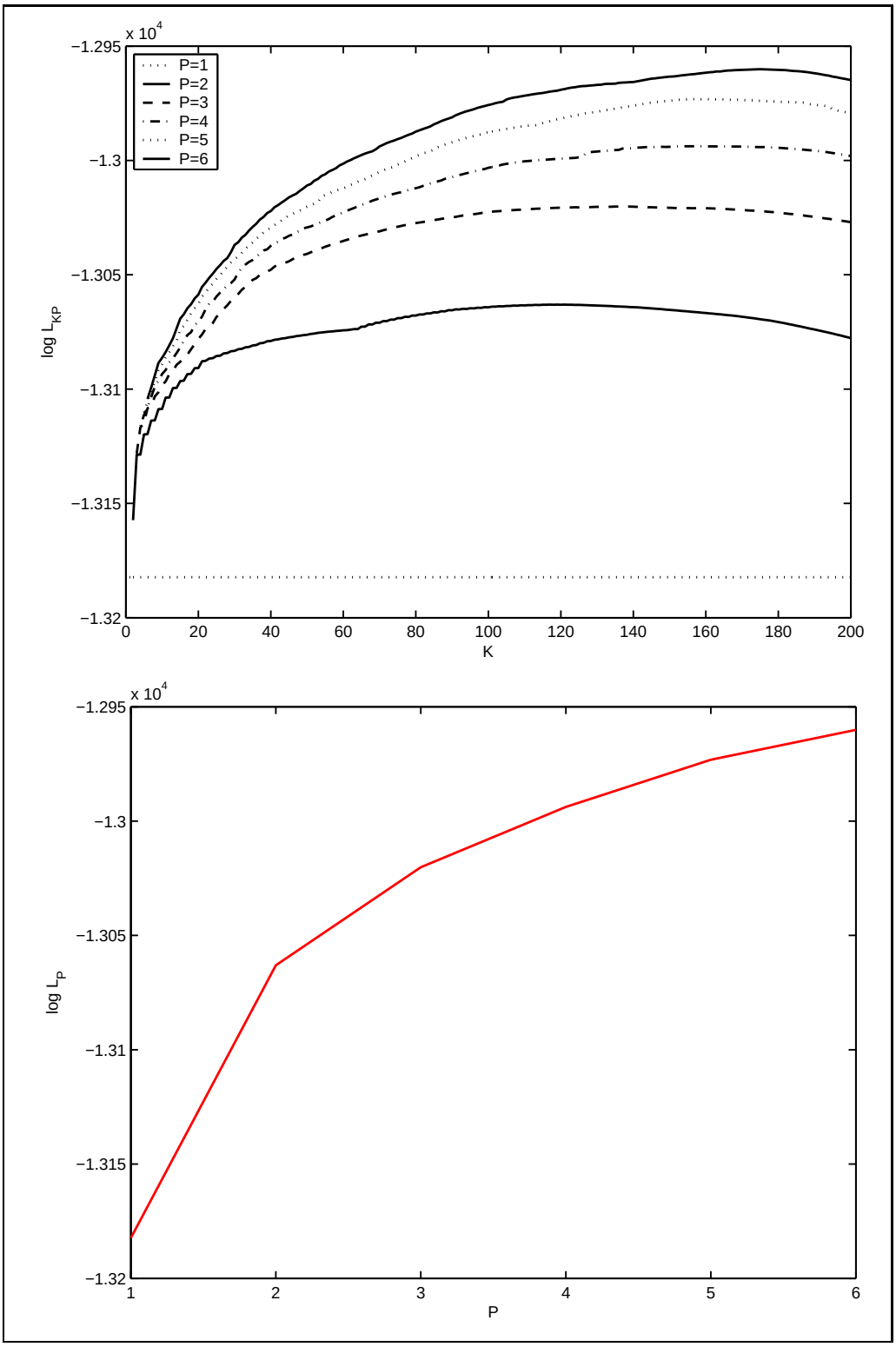


Figure 13.2: Incomplete-data log-likelihood for categorial data and associated sequence of increasing log-likelihoods $\{\log \widetilde{\mathcal{L}}_P\}$.

13.3 Analyzing the genome of Bacteriophage lambda

In this section we propose to apply our segmentation/clustering method to the analysis of Bacteriophage *lambda*. The complete genome of this virus is composed of $48,502$ base-pairs and the sequence is publicly available (NCBI web site, GeneBank, accession number NC_001416) as well as its annotation (Tables [2] 13.6 and 13.7). We propose to compare our results with HMMs (Muri (1997)) and segmentation (Lebarbier (2002)).

We consider model $M0$ where variables $\{Y_1, \ldots, Y_n\}$ are supposed to be independent. Table 13.2 gives the estimation of the proportion of each letter when the number of groups is $P = 2$ and $P = 3$. Estimation results are compared with HMMs. Table 13.3 provides the position of the breakpoints, and these positions are compared with those provided by the segmentation model. Figure 13.3 illustrates the posterior probabilities of membership to clusters for $P = 2$ and $P = 3$.

A first result is that the estimation of the frequency of letters gives the same results compared with HMMs, and posterior probabilities delimitate the same genomic regions (data not shown for HMMs). According to Table 13.2, clusters are characterized by a strong composition in G for cluster 1 and a richness in A and T for cluster 2. When adding a third group, the first region seems to be conserved and concerns cluster 1 exclusively (Figure 13.3). The second half of the genome concerns clusters 2 and 3. These results suggest that the heterogeneity which is observed in the genome of lambda is strongly linked to the existence of this first region which shows a richness in G. On the contrary, clusters 2 and 3 are characterized by a richness in A and T respectively.

Interestingly the breakpoints estimated by the segmentation/clustering are different from breakpoints estimated by the segmentation model (Table 13.3). Even if some position are common between both methods ($t = 22546, 46528$), most of them are different. As it has been shown in the Gaussian context, considering the clustered nature on segments changes the position of the breakpoints. Consequently, the segmentation/clustering model delimitates regions which are characterized by a global richness in some letters, whereas the segmentation model only considers local composition. If we compare the delimited regions to the annotation of the genome (Tables 13.6 and 13.7) it appears that the segmentation/clustering model and HMMs do not delimitate regions of particular biological function.

[2] from `http://www.ncbi.nlm.nih.gov/genomes/`

	A	G	C	T
1	0.2467	**0.2977**	0.2476	0.2080
2	**0.2697**	0.1969	0.2073	**0.3261**

	A	G	C	T
1	0.2464	0.2982	0.2475	0.2078
2	0.2697	0.1983	0.2083	0.3235

	A	G	C	T
1	0.2297	**0.3160**	0.2455	0.2000
2	**0.2840**	0.2584	0.2336	0.2240
3	0.2655	0.1972	0.2069	**0.3304**

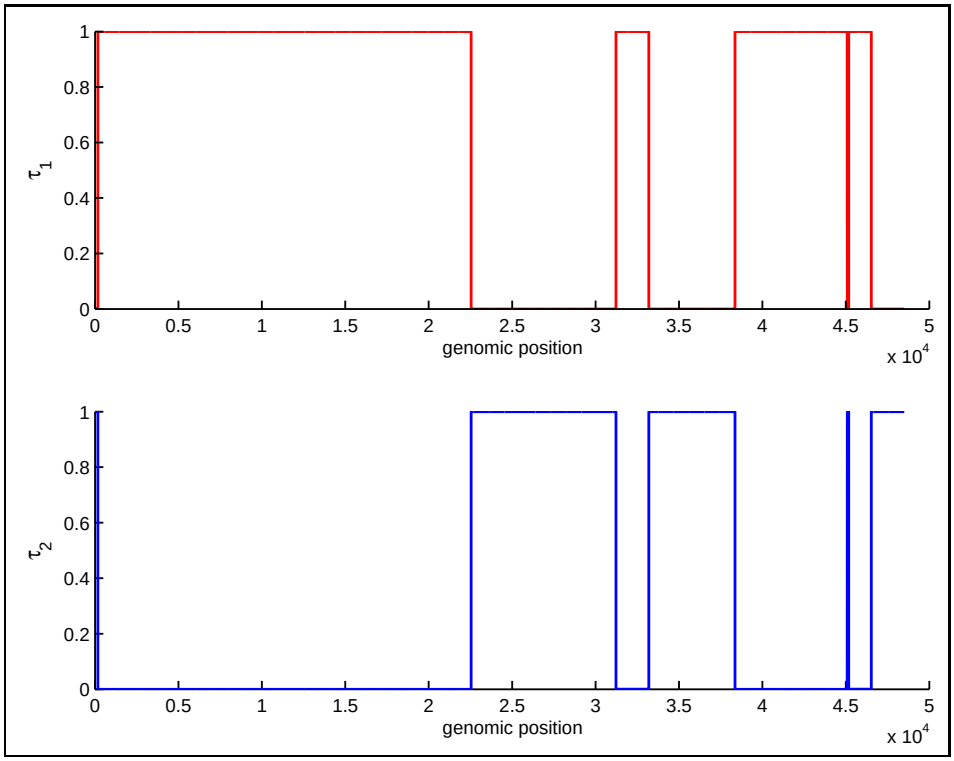
	A	G	C	T
1	0.2290	0.3167	0.2542	0.1998
2	0.2825	0.2593	0.2340	0.2241
3	0.2660	0.1968	0.2070	0.3299

Segmentation/clustering HMM

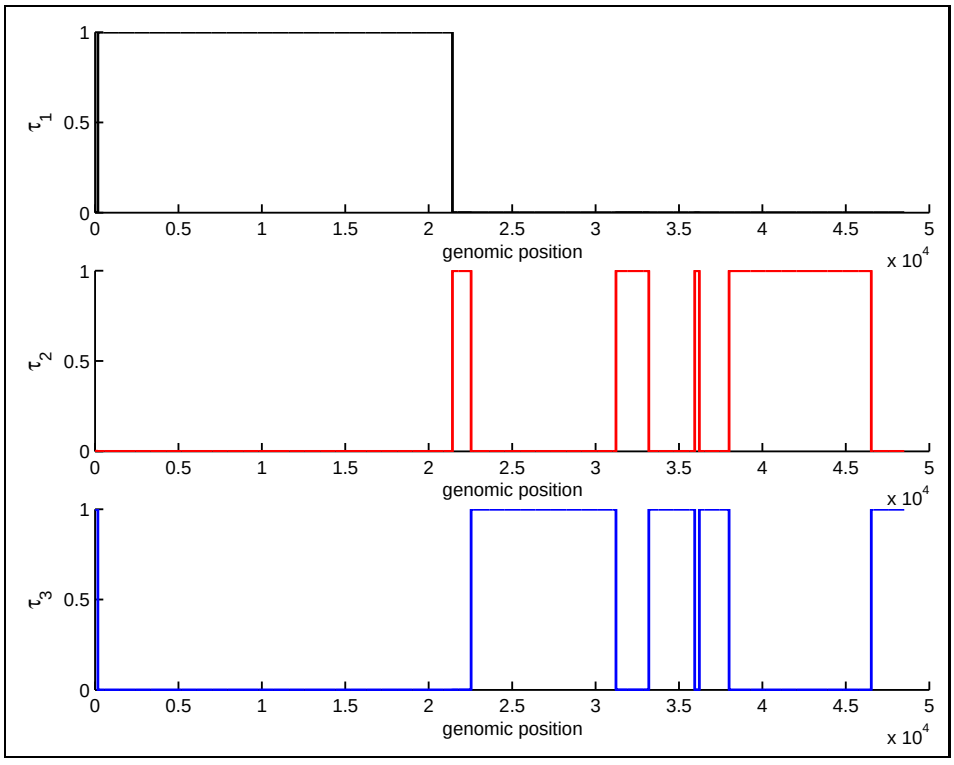
Table 13.2: Estimation results for Lambda, with $M0$, $P = 2$ and $P = 3$ clusters.

$P = 2$	$P = 3$	seg.
176	176	×
×	×	20010
×	×	20919
×	21425	×
22546	22546	22546
×	×	24117
×	×	27829
31224	31224	×
×	×	33082
33194	33194	×
×	35940	×
×	36219	×
×	38004	×
×	×	38082
38361	×	×
45085	×	×
45174	×	×
46528	46528	46528
48502	48502	48502
$\hat{K} = 9$	$\hat{K} = 10$	$\hat{K} = 9$

Table 13.3: Breakpoint positions for segmentation/clustering and segmentation (Lambda), model $M0$.



Posterior probabilities when $P = 2$



Posterior probabilities when $P = 3$

Figure 13.3: Clustering results for Lambda, $M0$. Posterior probabilities are plotted according to the position on the genome.

	A	G	C	T
1	0.2615	**0.3044**	0.2260	0.2082
2	**0.3035**	0.2349	0.1918	0.2699

	A	G	C	T
1	0.2476	0.2306	0.2063	**0.3155**
2	0.2615	**0.3041**	0.2260	0.2084
3	**0.3118**	0.2355	0.1896	0.2632

Segmentation/clustering

	A	G	C	T
1	0.1829	0.2197	0.2118	0.3854
2	0.5263	0.2685	0.1602	0.0448

	A	G	C	T
1	0.0480	0.4754	0.3699	0.1065
2	0.2409	0.0064	0.0442	0.7083
3	0.9469	0.0350	0.0166	0.0013

HMM

Table 13.4: Estimation results for B. Subtilis, with $M0$, $P = 2$ and $P = 3$ clusters.

13.4 Analyzing the genome of Bacillus Subtilis

In this section we apply our method to the analysis of B. subtilis [3]. We choose to restrict our study to the first 200,000 bases of the genome. An annotation of this part is provided in Tables [4] 13.8, 13.9, 13.10 and 13.11.

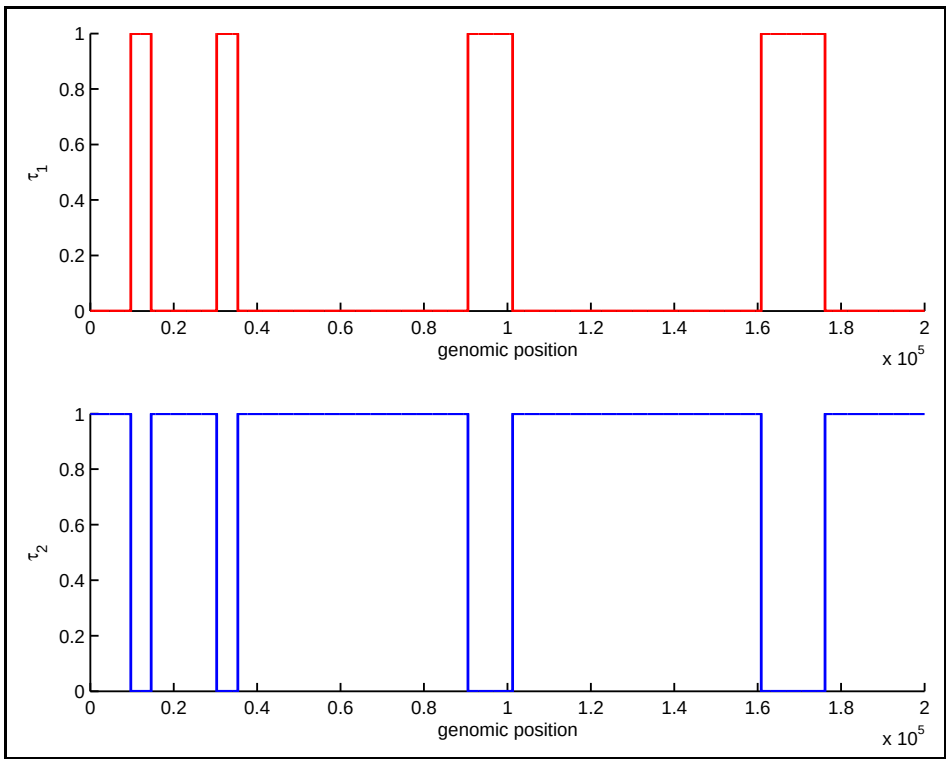
Interestingly it appears that HMMs do not detect any homogeneous region when models are $M0$ and $M1$. When considering $P = 2$ hidden states, Muri (1997) shows that HMMs delimitate 14397 regions whose average size is $4bp$ and $7bp$ respectively. In the previous example, we showed that our results were close to HMMs and different from segmentation. Nevertheless, in this example, it is the contrary: our results are very different from HMMs, and close to segmentation results.

In Table 13.4 is provided the estimation of the frequency of letters in each cluster. When $P = 2$, clusters are characterized by a strong composition in G and A, and the addition of a third group results in the creation of a cluster with a strong composition in T. When compared with HMMs, it can be seen that the estimation results are very different.

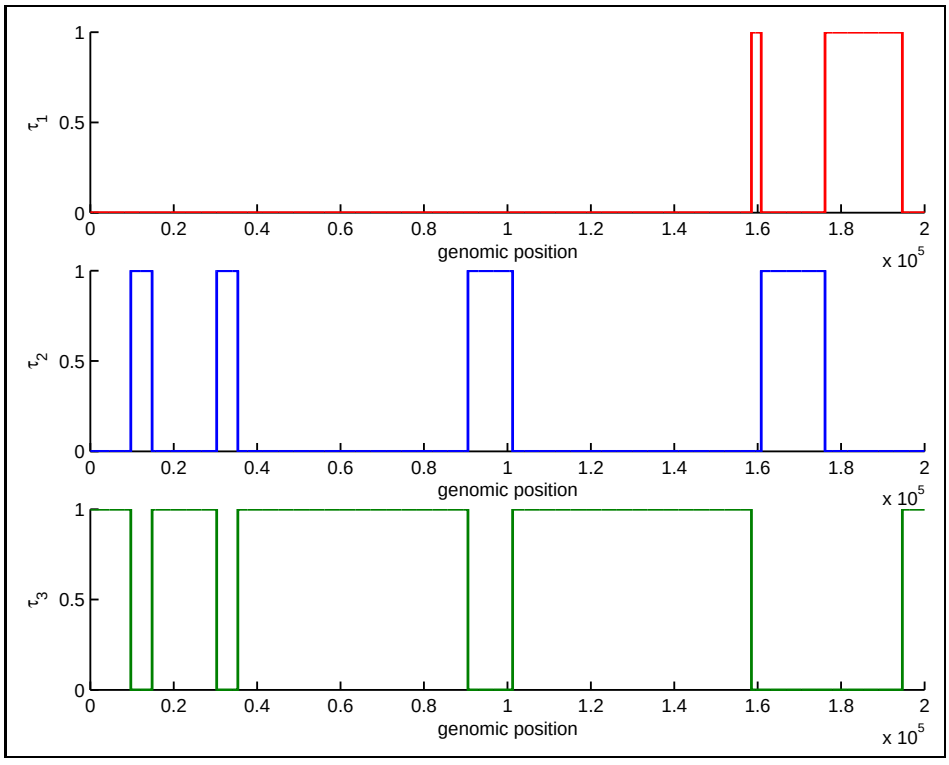
When studying the position of the breakpoints (Table 13.5 and Figure 13.4) we can see that the majority of breakpoints are conserved between segmentation and segmentation/clustering. When $P = 2$ some breakpoints are removed $t = 14833, 158486, 194694$, and when comparing the regions with the annotation, it can be seen that our method delimitates coding (cluster 2) and non-coding regions (cluster 1). When a third cluster is added, coding and non-coding regions remain, with the distinction of coding regions with a "positive" sens of transcription (cluster 3, strand +), and coding regions with a "negative" sens of transcription (cluster 1, strand -). Consequently the addition of a cluster leads to the detection of regions which are biologically different.

[3] Sequence available at `http://www.ncbi.nlm.nih.gov/` , accession number Z99104

[4] from `http://genolist.pasteur.fr/SubtiList/`



Posterior probabilities when $P = 2$



Posterior probabilities when $P = 3$

Figure 13.4: Clustering results for B. Subtilis, $M0$. Posterior probabilities are plotted according to the position on the genome.

$P = 2$	$P = 3$	seg.
9712	9712	9712
14607	$\times$	$\times$
$\times$	14833	14833
30294	30294	30294
35365	35365	35365
90537	90537	90537
101233	101233	101233
$\times$	158486	158486
160859	160859	160859
176132	176132	176132
$\times$	194694	194694
200000	200000	200000
$\hat{K} = 9$	$\hat{K} = 11$	$\hat{K} = 11$

Table 13.5: Breakpoint positions for segmentation/clustering and segmentation (B. subtilis), model $M0$.

13.5 Conclusion

In this last part we proposed an extension of our segmentation/clustering model to the case of categorial variables, with an application to DNA segmentation. The construction of this model is similar to the Gaussian case, but its implementation requires the use of a preliminary segmentation using CART since the size of the data is large. When applied to the analysis of genomes, we showed that our method detects regions with homogeneous composition, and that it provides a tradeoff between pure segmentation models and HMMs. Indeed, when HMMs fail to identify genomic regions, the underlying segmentation model helps in the recovering of regions with particular composition. In this chapter we studied two examples of sequences, and the analysis of other sequences with Markov models of higher order will have to be explored.

13.6 Annexes

begin	end	Strand	Synonym Product
191	736	+	DNA packaging protein
711	2636	+	DNA packaging protein
2633	2839	+	head-tail joining protein
2836	4437	+	capsid component
4418	5737	+	capsid component
5132	5737	+	capsid assembly
5747	6079	+	head-DNA stabilization
6135	7160	+	capsid component
7202	7600	+	DNA packaging
7612	7965	+	head-tail joining
7977	8555	+	tail component
8552	8947	+	tail component
8955	9695	+	tail component
9711	10133	+	tail component
10115	10549	+	tail component
10542	13103	+	tail component
13100	13429	+	tail component
13429	14127	+	tail component
14276	14875	+	tail component
14773	15444	+	tail component
15505	18903	+	tail:host specificity
18965	19585	+	outer host membrane
19650	20855	+	Tail fiber protein
20147	20767	-	Hypothetical protein
21029	21973	+	Tail fiber
21973	22557	+	Putative fiber assembly protein
22686	23918	-	ea47
24509	25399	-	ea31
25396	26973	-	ea59
27812	28882	-	integration protein
28860	29078	-	Excisionase
29118	29285	-	Hypothetical protein
29374	29655	-	ea8.5
29847	30395	-	ea22
30839	31024	-	Hypothetical protein
31005	31196	-	Hypothetical protein

Table 13.6: Bacteriophage lambda - annotation 1.

begin	end	Strand	Synonym Product
31169	31351	-	Hypothetical protein
31348	32028	-	exonuclease
32025	32810	-	bet
32816	33232	-	host-nuclease inhib. protein
33187	33330	-	host-killing
33299	33463	-	antitermination
33536	33904	-	Putative s-s DNA binding protein
34087	34287	-	restriction alleviation
34271	34357	-	Hypothetical protein
34482	35036	+	Superinfection exclusion protein B
35037	35438	-	early gene regulator
35825	36259	-	exclusion
36275	37114	-	exclusion
37227	37940	-	repressor
38041	38241	+	antirepressor
38360	38653	+	antitermination
38686	39585	+	DNA replication
39582	40283	+	DNA replication
40280	40570	+	ren exclusion protein
40644	41084	+	Nin
41081	41953	+	Nin protein
41950	42123	+	Nin protein
42090	42272	+	Nin
42269	42439	+	Nin
42429	43043	+	Nin
43040	43246	+	Nin
43224	43889	+	Nin protein
43886	44509	+	late gene regulator
44621	44815	+	Hypothetical protein
45186	45509	+	cell lysis protein
45493	45969	+	cell lysis protein
45966	46427	+	cell lysis protein
46459	46752	-	Bor protein precursor
47042	47575	-	putative envelope protein
47738	47944	+	Hypothetical protein

Table 13.7: Bacteriophage lambda - annotation 2.

begin	end	Strand	Synonym Product
410	1750	+	replication initiation protein
1939	3075	+	DNA polymerase III (beta subunit)
3206	3421	+	hypothetical protein
3437	4549	+	DNA repair and genetic recombination protein F
4567	4725	+	hypothetical protein
4866	6782	+	DNA gyrase (subunit B)
6993	9458	+	DNA gyrase (subunit A)
14845	15792	-	hypothetical protein
15913	17379	+	inosine-monophosphate dehydrogenase
17532	18863	+	D-alanyl-D-alanine carboxypeptidase
19060	19944	+	hypothetical protein
19966	20556	+	hypothetical protein
20878	22155	+	seryl-tRNA synthetase
22494	23147	-	deoxyadenosine/deoxycytidine kinase
23144	23767	-	deoxyguanosine kinase
23866	25149	-	hypothetical protein
25219	25764	-	hypothetical protein
25850	26335	+	hypothetical protein
26812	28503	+	DNA polymerase III (gamma and tau subunits)
28527	28850	+	hypothetical protein
28865	29461	+	DNA repair and genetic recombination protein R
29479	29703	+	hypothetical protein
29770	30033	+	inhibition of the pro-sigma-K processing machinery
35529	35723	+	hypothetical protein
35843	36457	+	hydrolysis of 5-bromo 4-chloroindolyl phosphate
36476	37636	+	hypothetical protein
37718	39160	+	hypothetical protein
39157	39795	+	thymidylate kinase
39869	40198	+	hypothetical protein
40211	40651	+	hypothetical protein
40663	41652	+	DNA polymerase III (delta' subunit)
41655	42482	+	hypothetical protein
42497	42856	+	hypothetical protein
42915	43658	+	hypothetical protein
43645	43944	+	hypothetical protein
43919	44797	+	hypothetical protein
44846	45136	-	transcriptional regulator
45631	47625	+	methionyl-tRNA synthetase
47704	48471	+	hypothetical protein
48627	49940	+	hypothetical protein
50085	50645	+	ribonuclease M5
50638	51516	+	dimethyladenosine transferase
51678	52550	+	hypothetical protein
52761	53021	+	hypothetical protein
53181	53366	+	small acid-soluble spore protein

Table 13.8: B. Subtilis - annotation 1.

begin	end	Strand	Synonym Product
53514	54383	+	4-diphosphocytidyl-2-C-methyl-D-erythritol kinase
54439	55296	+	transcriptional regulator
55293	55670	+	hypothetical protein
55864	56157	+	required for spore cortex synthesis
56350	57720	+	UDP-N-acetylglucosamine pyrophosphorylase
57743	58696	+	phosphoribosylpyrophosphate synthetase
58781	59395	+	general stress protein
59502	60068	+	peptidyl-tRNA hydrolase
60128	60358	+	hypothetical protein
60428	63961	+	transcription-repair coupling factor
64097	64633	+	transcriptional regulator
64815	66413	+	hypothetical protein
66403	67872	+	hypothetical protein
67875	68135	+	hypothetical protein
68214	68516	+	hypothetical protein
68513	69148	+	hypothetical protein
69166	69543	+	cell-division initiation protein
69624	70010	+	hypothetical protein
70536	73019	+	serine phosphatase
73104	73841	+	hypothetical protein
73807	74823	+	hypothetical protein
74927	76345	+	hypothetical protein
76342	76884	+	hypoxanthine-guanine phosphoribosyltransferase
76982	78895	+	cell-division protein and general stress protein
79090	79791	+	hypothetical protein
79877	80752	+	hypothetical protein
80799	81692	+	hypothetical protein
81768	82694	+	cysteine synthetase A
82861	84273	+	para-aminobenzoate synthase (subunit A)
84287	84871	+	anthranilate synthase (subunit II)
84871	85752	+	aminodeoxychorismate lyase
85734	86591	+	dihydropteroate synthase
86584	86946	+	dihydroneopterin aldolase
86943	87446	+	7,8-dihydro-6-hydroxymethylpterin pyrophosphokinase
87398	87607	+	hypothetical protein
87631	88632	+	hypothetical protein
88724	90223	+	lysyl-tRNA synthetase
101446	101910	+	transcriptional regulator
101924	102481	+	modulation of CtsR repression
102481	103572	+	modulation of CtsR repression
103569	106001	+	class III stress response-related ATPase
106093	107469	+	DNA repair protein homolog
107473	108555	+	hypothetical protein
108671	109771	+	hypothetical protein
109786	110484	+	hypothetical protein
110477	110953	+	hypothetical protein
111044	112495	+	glutamyl-tRNA synthetase

Table 13.9: B. Subtilis - annotation 2.

begin	end	Strand	Synonym Product
112797	113450	+	serine acetyltransferase
113447	114847	+	cysteinyl-tRNA synthetase
114851	115282	+	hypothetical protein
115266	116015	+	hypothetical protein
116022	116534	+	hypothetical protein
116597	117253	+	RNA polymerase sigma-30 factor (sigma-H)
117346	117495	+	ribosomal protein L33
117529	117708	+	preprotein translocase subunit
117887	118420	+	transcription antitermination factor
118588	119013	+	ribosomal protein L11 (BL11)
119107	119805	+	ribosomal protein L1 (BL1)
120057	120557	+	ribosomal protein L10 (BL5)
120604	120975	+	ribosomal protein L12 (BL9)
121065	121670	+	hypothetical protein
121916	125497	+	RNA polymerase (beta subunit)
125559	129158	+	RNA polymerase (beta' subunit)
129339	129587	+	hypothetical protein
129701	130117	+	ribosomal protein S12 (BS12)
130159	130629	+	ribosomal protein S7 (BS7)
130683	132761	+	elongation factor G
132881	134071	+	elongation factor Tu
134170	135126	+	hypothetical protein
135362	135670	+	ribosomal protein S10 (BS13)
135710	136339	+	ribosomal protein L3 (BL3)
136367	136990	+	ribosomal protein L4
136990	137277	+	ribosomal protein L23
137309	138142	+	ribosomal protein L2 (BL2)
138200	138478	+	ribosomal protein S19 (BS19)
138495	138836	+	ribosomal protein L22 (BL17)
138840	139496	+	ribosomal protein S3 (BS3)
139498	139932	+	ribosomal protein L16
139922	140122	+	ribosomal protein L29
140145	140408	+	ribosomal protein S17 (BS16)
140449	140817	+	ribosomal protein L14
140855	141166	+	ribosomal protein L24 (BL23)
141193	141732	+	ribosomal protein L5 (BL6)
141755	141940	+	ribosomal protein S14
141972	142370	+	ribosomal protein S8 (BS8)
142400	142939	+	ribosomal protein L6 (BL8)
142972	143334	+	ribosomal protein L18
143359	143859	+	ribosomal protein S5
143873	144052	+	ribosomal protein L30 (BL27)
144083	144523	+	ribosomal protein L15
144525	145820	+	preprotein translocase subunit

Table 13.10: B. Subtilis - annotation 3.

begin	end	Strand	Synonym Product
145875	146528	+	adenylate kinase
146525	147271	+	methionine aminopeptidase
147583	147801	+	initiation factor IF-I
147835	147948	+	ribosomal protein L36
147971	148336	+	ribosomal protein S13
148357	148752	+	ribosomal protein S11 (BS11)
148929	149873	+	RNA polymerase (alpha subunit)
149951	150313	+	ribosomal protein L17 (BL15)
150441	151286	+	hypothetical protein
151301	152131	+	hypothetical protein
152128	152925	+	hypothetical protein
152935	153678	+	pseudouridylate synthase I
153841	154278	+	ribosomal protein L13
154299	154691	+	ribosomal protein S9
155155	155922	+	hypothetical protein
156108	156551	+	hypothetical protein
156611	157324	+	N-acetylmuramoyl-L-alanine amidase
157420	158478	+	hypothetical protein
158514	159071	-	germination response to L-alanine
159181	159777	+	activation of the KinB signaling pathway to sporulation
159778	160542	-	hypothetical protein
177082	178518	+	hypothetical protein
178733	179584	+	hypothetical protein
179594	180163	-	hypothetical protein
180168	181352	-	integral membrane protein
181345	182349	-	integral membrane protein
182368	183321	-	iron-binding protein
183412	185001	-	hypothetical protein
185192	186436	-	hypothetical protein
186450	188378	-	hypothetical protein
188406	189731	-	hypothetical protein
189788	191155	-	hypothetical protein
191181	192032	-	hypothetical protein
192049	192963	-	hypothetical protein
193134	193553	-	hypothetical protein
193566	194021	-	hypothetical protein
194838	195401	+	RNA polymerase ECF
195415	196041	+	hypothetical protein
196202	197023	+	hypothetical protein
197016	198467	+	hypothetical protein
198486	199832	+	hypothetical protein

Table 13.11: B. Subtilis - annotation 4.

Part VI

Publications

BMC Bioinformatics

BioMed Central

Research article

Open Access

A statistical approach for array CGH data analysis

Franck Picard*[1], Stephane Robin*[1], Marc Lavielle[2], Christian Vaisse[3] and Jean-Jacques Daudin[1]

Address: [1]Institut National Agronomique Paris-Grignon, UMR INAPG/ENGREF/INRA MIA 518, Paris, France, [2]Université Paris Sud, Equipe Probabilités, Statistique et Modélisation, Orsay, France and [3]University of California San Francisco, Diabetes Center, San Francisco, USA

Email: Franck Picard* - picard@inapg.fr; Stephane Robin* - robin@inapg.fr; Marc Lavielle - lavielle@math.u-psud.fr; Christian Vaisse - vaisse@medicine.ucsf.edu; Jean-Jacques Daudin - daudin@inapg.fr

* Corresponding authors

Published: 11 February 2005

BMC Bioinformatics 2005, **6**:27 doi:10.1186/1471-2105-6-27

Received: 18 August 2004
Accepted: 11 February 2005

This article is available from: http://www.biomedcentral.com/1471-2105/6/27

Abstract

Background: Microarray-CGH experiments are used to detect and map chromosomal imbalances, by hybridizing targets of genomic DNA from a test and a reference sample to sequences immobilized on a slide. These probes are genomic DNA sequences (BACs) that are mapped on the genome. The signal has a spatial coherence that can be handled by specific statistical tools. Segmentation methods seem to be a natural framework for this purpose. A CGH profile can be viewed as a succession of segments that represent homogeneous regions in the genome whose BACs share the same relative copy number on average. We model a CGH profile by a random Gaussian process whose distribution parameters are affected by abrupt changes at unknown coordinates. Two major problems arise : to determine which parameters are affected by the abrupt changes (the mean and the variance, or the mean only), and the selection of the number of segments in the profile.

Results: We demonstrate that existing methods for estimating the number of segments are not well adapted in the case of array CGH data, and we propose an adaptive criterion that detects previously mapped chromosomal aberrations. The performances of this method are discussed based on simulations and publicly available data sets. Then we discuss the choice of modeling for array CGH data and show that the model with a homogeneous variance is adapted to this context.

Conclusions: Array CGH data analysis is an emerging field that needs appropriate statistical tools. Process segmentation and model selection provide a theoretical framework that allows precise biological interpretations. Adaptive methods for model selection give promising results concerning the estimation of the number of altered regions on the genome.

Background

Chromosomal aberrations often occur in solid tumors: tumor suppressor genes may be inactivated by physical deletion, and oncogenes activated via duplication in the genome. Gene dosage effect has become particularly important in the understanding of human solid tumor genesis and progression, and has also been associated with other diseases such as mental retardation [1,2]. Chromosomal aberrations can be studied using many different techniques, such as Comparative Genomic Hybridization (CGH), Fluorescence in Situ Hybridization (FISH), and Representational Difference Analysis (RDA).

Although chromosome CGH has become a standard method for cytogenetic studies, technical limitations restrict its usefulness as a comprehensive screening tool [3]. Recently, the resolution of Comparative Genomic Hybridizations has been greatly improved using microarray technology [4,5].

The purpose of array-based Comparative Genomic Hybridization (array CGH) is to detect and map chromosomal aberrations, on a genomic scale, in a single experiment. Since chromosomal copy numbers can not be measured directly, two samples of genomic DNA (referred to as the reference and test DNAs) are differentially labelled with fluorescent dyes and competitively hybridized to known mapped sequences (referred to as BACs) that are immobilized on a slide. Subsequently, the ratio of the intensities of the two fluorochromes is computed and a CGH profile is constituted for each chromosome when the $\log_2$ of fluorescence ratios are ranked and plotted according to the physical position of their corresponding BACs on the genome [6]. Different methods and packages have been proposed for the visualization of array CGH data [7,8].

Each profile can be viewed as a succession of "segments" that represent homogeneous regions in the genome whose BACs share the same relative copy number on average. Array CGH data are normalized with a median set to $\log_2(\text{ratio}) = 0$ for regions of no change, segments with positive means represent duplicated regions in the test sample genome, and segments with negative means represent deleted regions. Even if the underlying biological process is discrete (counting of relative copy numbers of DNA sequences), the signal under study is viewed as being continuous, because the quantification is based on fluorescence measurements, and because the possible values for chromosomal copy numbers in the test sample may vary considerably, especially in the case of clinical tumor samples that present mixtures of tissues of different natures.

Two main statistical approches have been considered for the analysis of array CGH data. The first has focused many attentions, and is based on segmentation methods where the purpose is to locate segments of biological interest [7,9-11]. A second approach is based on Hidden Markov Models (aCGH R-package [12]), where the purpose is to cluster individual data points into a finite number of hidden groups. Our approach can be put into the first category. Segmentation methods seem to be a natural framework to handle the spatial coherence of the data on the genome that is specific to array CGH. In this context the signal provided by array CGH data is supposed to be a realization of a Gaussian process whose parameters are affected by an unknown number of abrupt changes at unknown locations on the genome. Two models can be considered, according to the characteristics of the signal that is affected by the changes: it can be either the mean of the signal [7,10,11] or the mean and the variance [9]. Since the choice of modeling is crucial in any interpretation of a segmented CGH profile, we provide guidelines for this choice in the discussion. Two major issues arise in break-points detection studies: the localization of the segments on the genome, and the estimation of the number of segments. The first point has lead to the definition of many algorithms and packages: segmentation algorithms [9,10] and smoothing algorithms [11] where the break-points are defined with a *posterior* empirical criterion. These methods are defined by a criterion to optimize and an algorithm of optimization. Different criteria have been proposed: the likelihood criterion [9,11], the least-squares criterion [7], partial sums [10], and algorithms of optimization are based on genetic algorithms [9], dynamic programing [7], binary segmentation (DNAcopy R-package [10]) and adaptive weigths smoothing (GLAD R-package [11]). Since many criteria and algorithms have been proposed, one important question is the resulting statistical properties of the break-point estimators they provide. Note that smoothing techniques do not provide estimators of the break-point coordinates, since the primary goal of the underlying model is to smooth the data, and break-points are not parameters of the model (in this case, they are defined after the optimization of the criterion [11]). Here we consider the likelihood criterion and we use dynamic programming that provides a global optimum solution, contrary to genetic algorithms [9], in a reasonable computational time.

As for the estimation of the number of segments, the existing articles have not defined any statistical criterion adapted to the case of process segmentation. This problem is theoretically complex, and has lead to *ad hoc* procedures [9-11]. Since the purpose of array CGH experiments is to discover biological events, the estimation of the number of segments remains central. This problem can be handled in the more general context of model selection. In the discussion we explain why classical criteria based on penalized likelihoods are not valid for break-points detection. Criteria such as the Akaike Information Criterion (AIC) and the Bayes Information Criterion (BIC) lead to an overestimation of the number of segments. For this reason, an arbitrary penalty constant can be chosen in order to select a lower number of segments in the profile [9]. We propose a new procedure to estimate the number of segments, choosing the penalty constant adaptively to the data. We explain the construction of such penalty, and its performances are compared to other criteria in the Results Section, based on simulation studies and on publicly available data sets. Put together, we propose a methodology that considers a simple modeling, a fast and effective

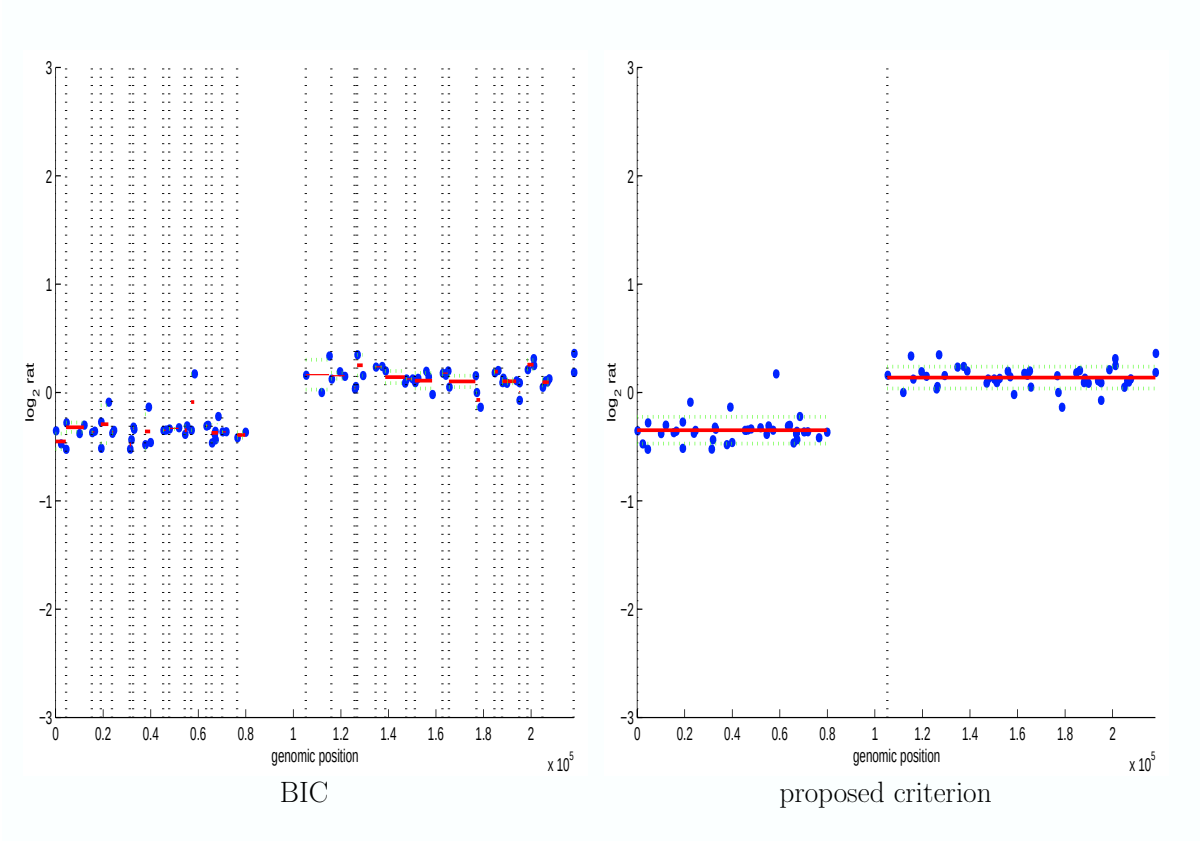


Figure 1
Results of the segmentation procedure when using the Bayesian Information Criterion (BIC) and the proposed criterion. Data shown corresponds to Coriell cell lines GM03563, chromosome 3. Red lines represent the estimated mean of each segments, and green lines, the estimated mean plus one standard deviation.

algorithm of optimization and that takes advantages of the statistical properties of the maximum likelihood. Our procedure has been implemented on MATLAB Software and is freely available http://www.inapg.fr/ens_rech/mathinfo/recherche/mathematique/outil.html.

Results

Comparison of model selection criteria

To show the importance of the choice of the model selection criterion on simple data, we use the results of a single experiment performed on fibroblast cell lines (see the Materials Section), with one known chromosomal aberration. Figure 1 shows the resulting segmentations when using the Bayesian Information Criterion, and our criterion. BIC leads to an oversegmented profile that is not interpretable in terms of relative copy numbers. Our procedure estimates the correct number of segments $K = 2$. This example shows the practical consequences of the use of theoretically unappropriated criteria. This point constitutes the main purpose of the discussion (see the Discussion Section).

Numerical simulations are performed to study the sensitivity of different criteria to varying amounts of noise. The simulation design is described in the Methods Section. We compare four different criteria: the Bayesian Information Criterion, two previously described criteria [9,13], and the criterion we propose, in their ability to estimate the correct number of segments. Two configurations were tested, for a true number of segments $K^* = 5$. In the first situation, the segments are regularly spaced with a jump of the mean of 1 (Figure 3), whereas in the second case, the segments

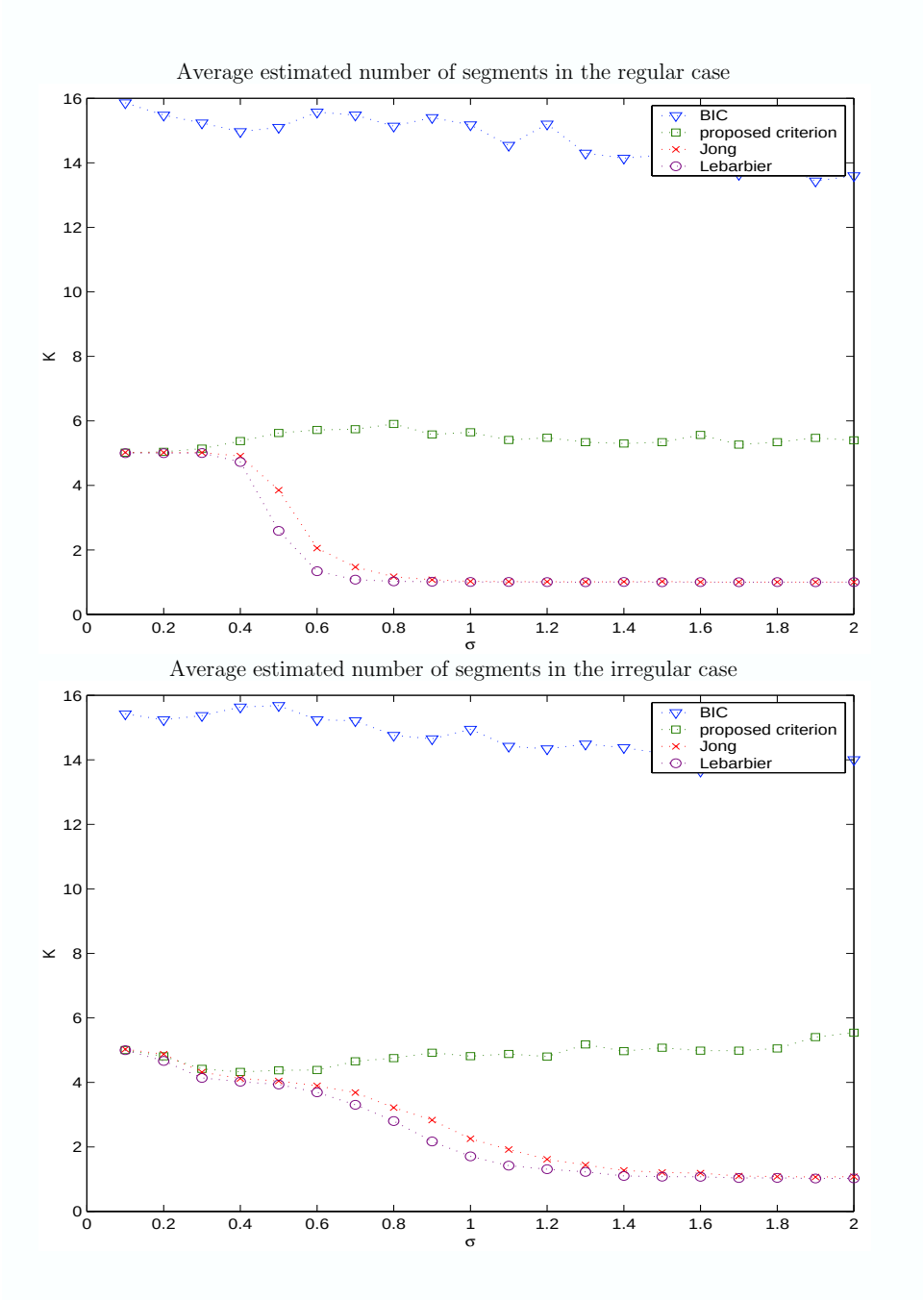


Figure 2
Estimated number of segments for 4 different penalized criteria in the regular case (top) and the irregular case (bottom). Top : Results of the simulations for 5 regularly spaced segments with n = 100 data points. The graph represents the average estimated number of segments for each criterion according to the standard deviation of the noise (ω). Bottom: Results of the simulations for 5 unregularly spaced segments with n = 100 data points. The adaptive criterion is robust to the additional noise since it maintains an estimate close to 5 segments whatever the noise and the configuration.

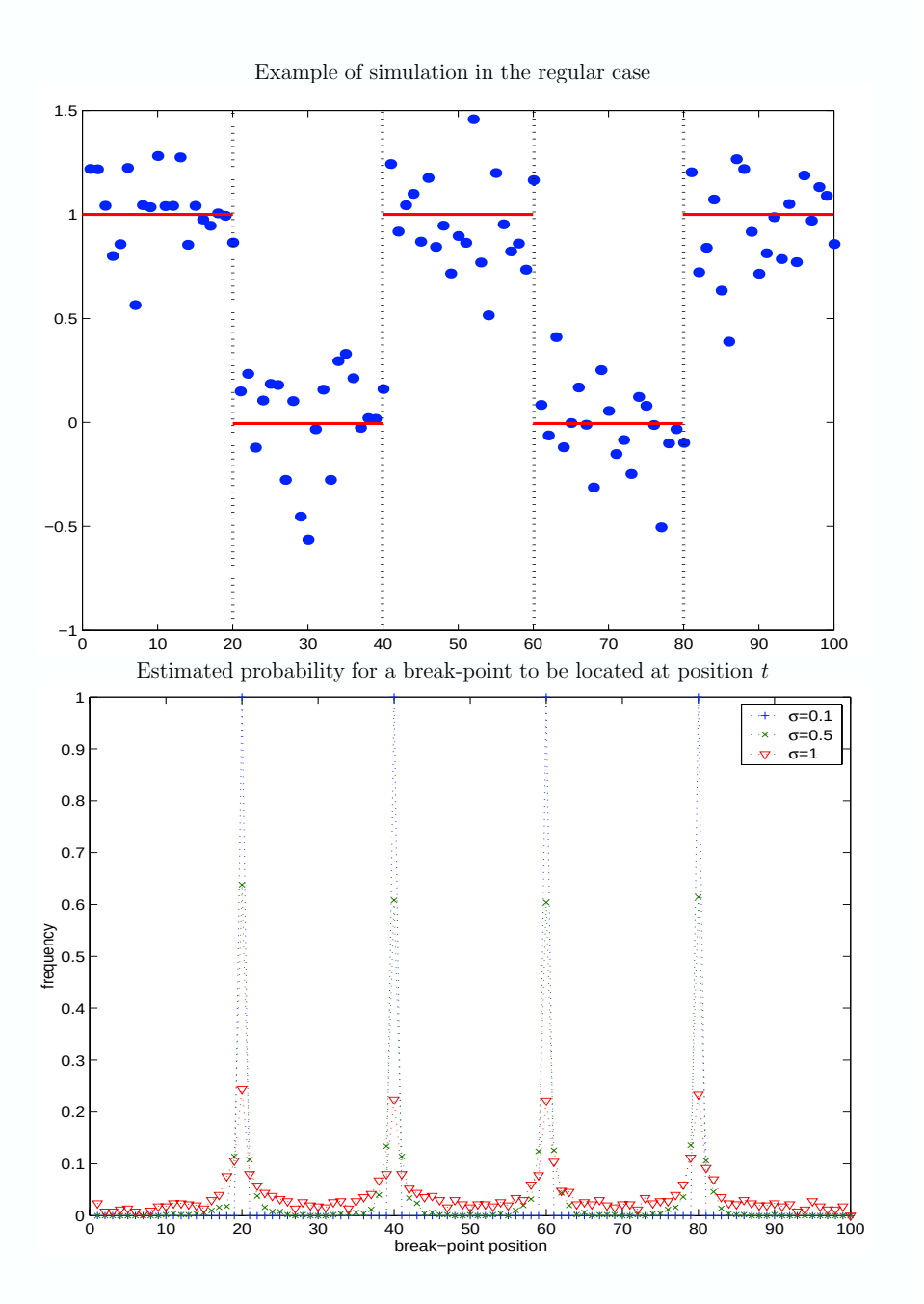


Figure 3
Example of a simulation in the regular case, and result of the dynamic programming algorithm for the estimation of the break-point coordinates. Top: Example of simulation for 100 data points and 5 segments in the regular case. The true break-points are designated by vertical lines, and the red lines correspond to the mean of each segment. The difference of means d is constant and equals 1. Bottom: Estimated frequency for a break-point to be located at coordinate t for t = 1 to 100. Different levels of noise are considered with ω = 0.1, ω = 0.5, ω = 1.

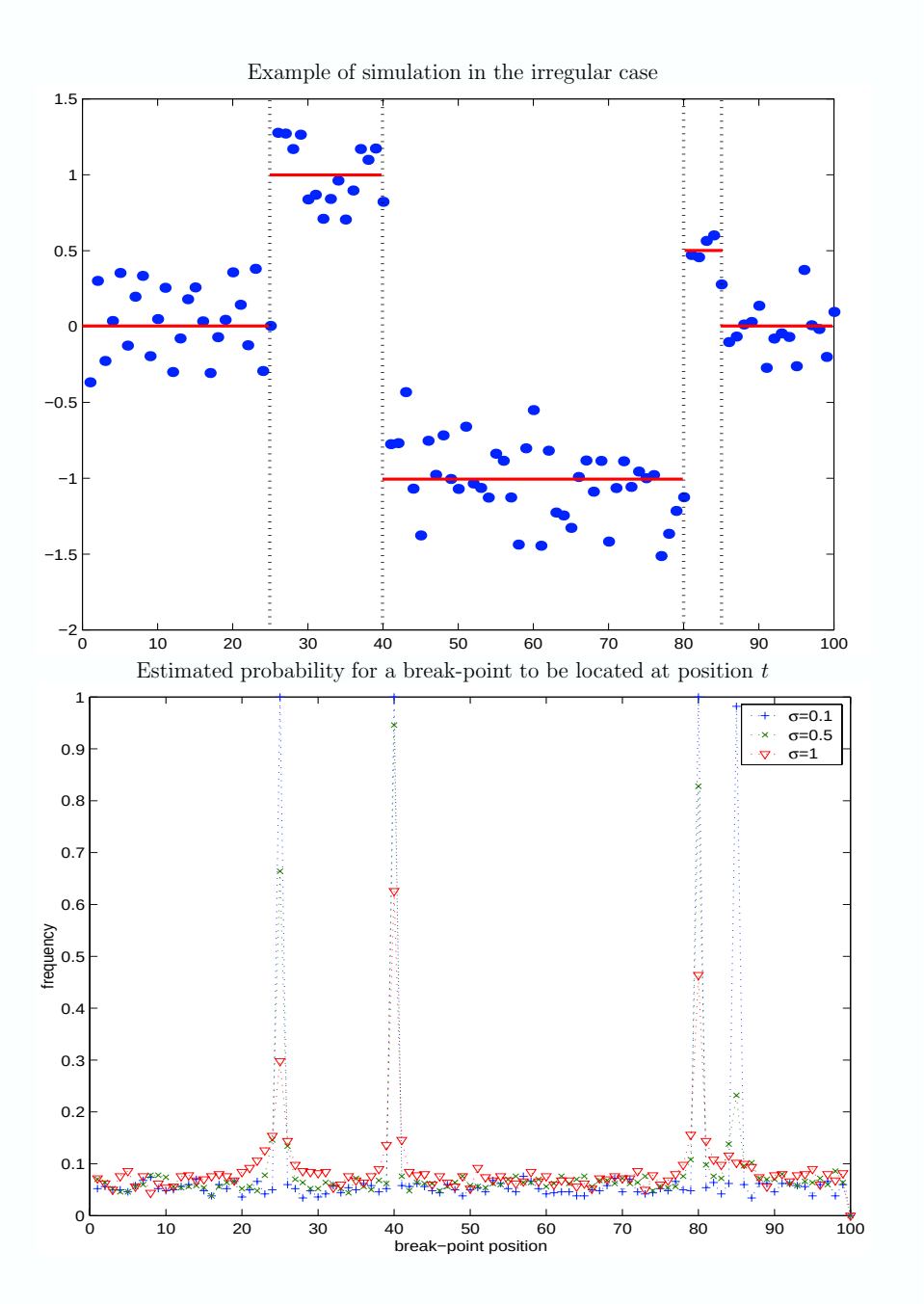


Figure 4
Example of a simulation in the irregular case, and result of the dynamic programming algorithm for the estimation of the break-point coordinates. Top: Example of simulation for 100 data points and 5 segments in the irregular case. The true break-points are designated by vertical lines, and the red lines correspond to the mean of each segment. The difference of means varies between $d = 2$ to $d = 0.5$. Bottom: Estimated probability for a break-point to be located at coordinate t for $t = 1$ to 100. Different levels of noise are considered with $\omega = 0.1$, $\omega = 0.5$, $\omega = 1$.

are not regularly spaced and the differences of means vary between $d = 2$ and $d = 0.5$ (Figure 4). The first result is that BIC overestimates the number of segments, whatever the noise and the configuration (Figure 2). On the contrary, previously described criteria [9,13] tend to underestimate the number of segments when the noise increases, whatever the configuration. These results suggest that those two criteria "prefer" to detect no break-point as the noise increases, leading to possible false negative results.

The behavior of the criterion we propose is different. It seems to be more robust to the noise, as it will give a number of segments that is close to the true number. In particular, the irregular configuration presents a segment of small size (5 points at $t = 80$) that could be interesting to detect in the case of array CGH profile (a putative gained region for instance). Since the previously described criteria [9,13] tend to underestimate the number of segments, this particular region would not be detected. On the contrary, the adaptive criterion will be able to detect it, even if the noise is important, since it selects a constant number of segments close to the true number whatever the noise. These simulation examples perfectly illustrate the capacity of an adaptive criterion to find a reasonable number of segments even in configurations where the profile is not very separated.

We also compare the performance of our criterion and of the arbitrary criterion [9] on breast cancer cell lines. Figure 5 shows the resulting segmentations on chromosomes 9 and 10 of the Bt474 cell line (see the Materials Section for further description). As previously mentioned, the arbitrary criterion [9] selects a lower number of segments compared to the adaptive criterion, and we note that interesting regions are not detected (a putative outlier on chromosome 9 at 1.58 Mb and a putative deleted region on chromosome 10 at 1.76 Mb). Since the aim of array CGH experiments is to discover unknown chromosomal aberrations, the use of an adaptive criterion seems more appropriate in this context since it allows the identification of regions that seem biologically relevent.

The second simulation-based result concerns the ability of dynamic programming to locate the break-points at the correct coordinate, given different amounts of noise (Figures 3 and 4). In the regular configuration (Figure 3), simulation results show that dynamic programming perfectly localizes the break-points when the variability of the noise ω^2 is low regarding the jump d of the mean. If $d/\omega = 10$ the estimated probability to localize the break-points at the correct coordinate is 1, and this probability deacreases with the noise (probability close to 0.65 for $d/\omega = 2$ and 0.25 for $d/\omega = 1$). The effect of additional noise is to widden the zone of estimation, but the estimated break-points remain close to the true break-points. If the true break-point is located at t^*, the estimated break-point stays in the interval $t^* \pm 3$. In the irregular configuration, additional noise has similar effects on the break-point's positioning, but the probability to correctly estimate a break-point depends on the jump of the mean between two segments. In the irregular case, Figure 4, at position $t = 40$ the difference of mean is $d = 2$, and the probability to locate the break-point at the true coordinate is higher than 0.65 for any additional noise. On the contrary, at position $t = 85$ where the different of mean equals $d = 0.5$ the probability to correctly locate the break-point decreases dramatically with the noise (probability 1 for $\omega = 0.1$ and probability 0.25 for $\omega = 0.5$). This means that dynamic programming is sensitive to small segments that present little differences in the mean regarding the noise. Nevertheless, the example on the real data set presented in Figure 5 shows that using an adaptive criterion with dynamic programming allows for the identification of small regions of putative biological interest as mentioned above. Put together, these simulation results show that the adaptive method selects the good number of segments even in the presence of important noise, and that when this number is selected, dynamic programming is able to correctly localize the break-point. In addition to its ability to locate precisely the break-points, it is important to notice that dynamic programming provides a global optimum of the likelihood that is required for any model selection procedure to select the number of segments, compared to genetic algorithms [9].

Segmentation models in the Gaussian framework

The CGH profile is supposed to be a Gaussian signal. In a segmentation framework, two types of changes can be considered: changes in the mean and the variance of the signal, or changes in the mean only. Let us define model Q_1 where each segment has a specific mean and variance [9], and model Q_2, where the variance is common between segments [7].

Since both models can be used, it is important to explore their behavior in order to know which model is the best adapted to the special case of array CGH data. We use clinical data obtained from primary dissected tumors of colorectal cancers (see the Materials Section for further details). Figure 6 presents the results of segmentations for three experiments obtained with the two models Q_1 and Q_2 when our criterion is used to estimate the number of segments. The main result of this comparison is that the number of segments is higher using model Q_2 compared to model Q_1. This behavior of model Q_2 could be interpreted as a trend to divide large segments into smaller parts, in order to maintain the variance homogeneous between segments. This leads to a more segmented profile, maybe more precise, but that may be

BMC Bioinformatics 2005, **6**:27 http://www.biomedcentral.com/1471-2105/6/27

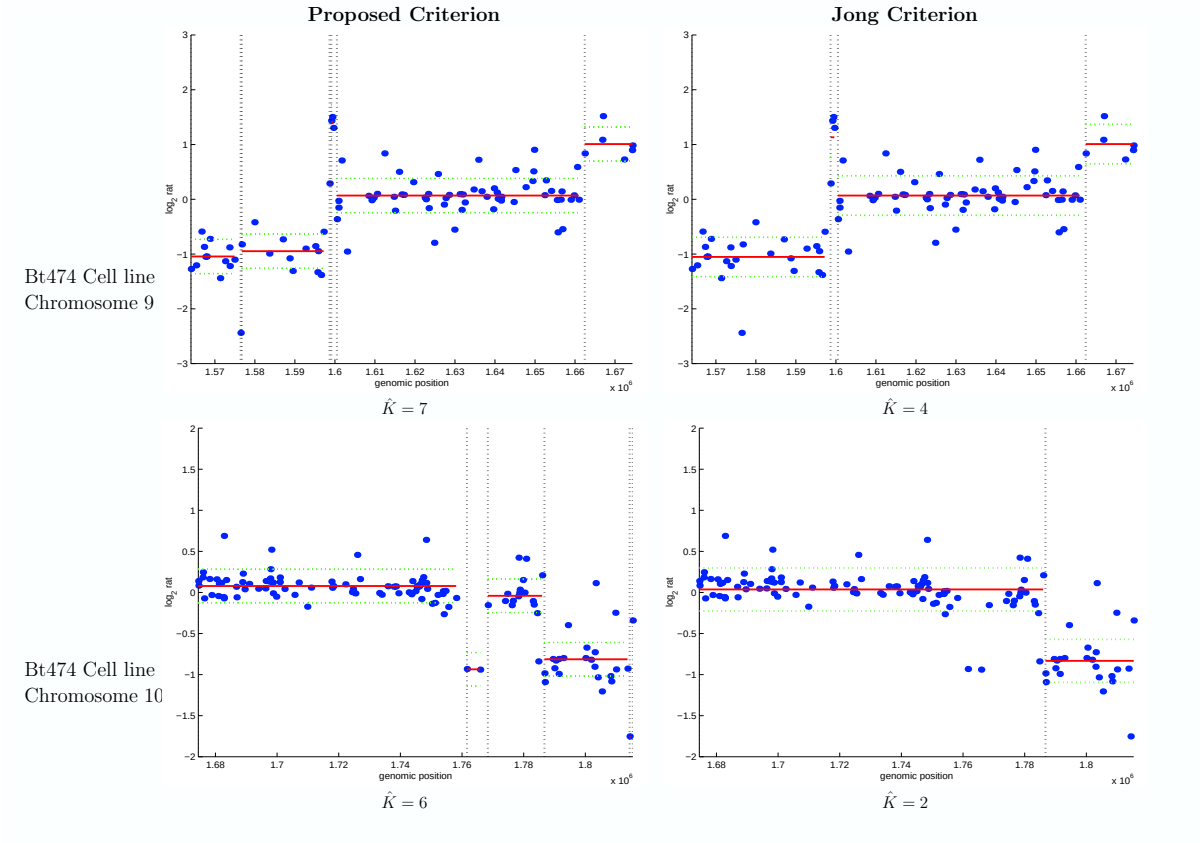


Figure 5

Comparison of segmentation results based on Breast Cancer Cell lines using the adaptive criterion and Jong criterion. Results of the segmentation procedure for Breast cancer cell lines Bt474, chromosomes 9 and 10. Fluoresence $\log_2$-ratios are plotted according to their location on the genome in megabases. Left profiles are segmented using the adaptive criterion and right profiles using Jong's criterion. The adaptive method detects a break-point at 1.58 MB on chromosome 9 that seems to be an outlier, and detects a putative deleted region on chromosome 10 at 1.76 MB.

more difficult to interpret in terms of relative copy numbers. Nevertheless, as model Q_2 allows the exploration of segments with one observation, it will be more efficient for the identification of outliers, as shown in Figure 6 (experiment X411, model Q_2, point at 100 Mb).

Discussion

The definition of an appropriate penalized criterion has been an issue for previous works using segmentation methods for array CGH data analysis [8,9,11]. In this section, we explain the specificity of model selection in the case of process segmentation, in order to give further justification to the inefficiency of classical criteria to select the number of segments, as shown in the Results Section.

Estimating the number of segments via penalized likelihood

When the number of segments is known, the maximization of the log-likelihood P_K gives the best segmentation with K segments (see the Methods Section). In real situations this number is unknown, and one has to choose among many possible segmentations. The maximum of the log-likelihood $\hat{P}_K$ can be viewed as a quality measurement of the fit to the data of the model with K segments, and will be maximal when each data point is in its own segment. Therefore selecting the number of segments only based on the likelihood criterion would lead to overfitting. Furthermore, the number of parameters to estimate is proportional to the number of segments, and a

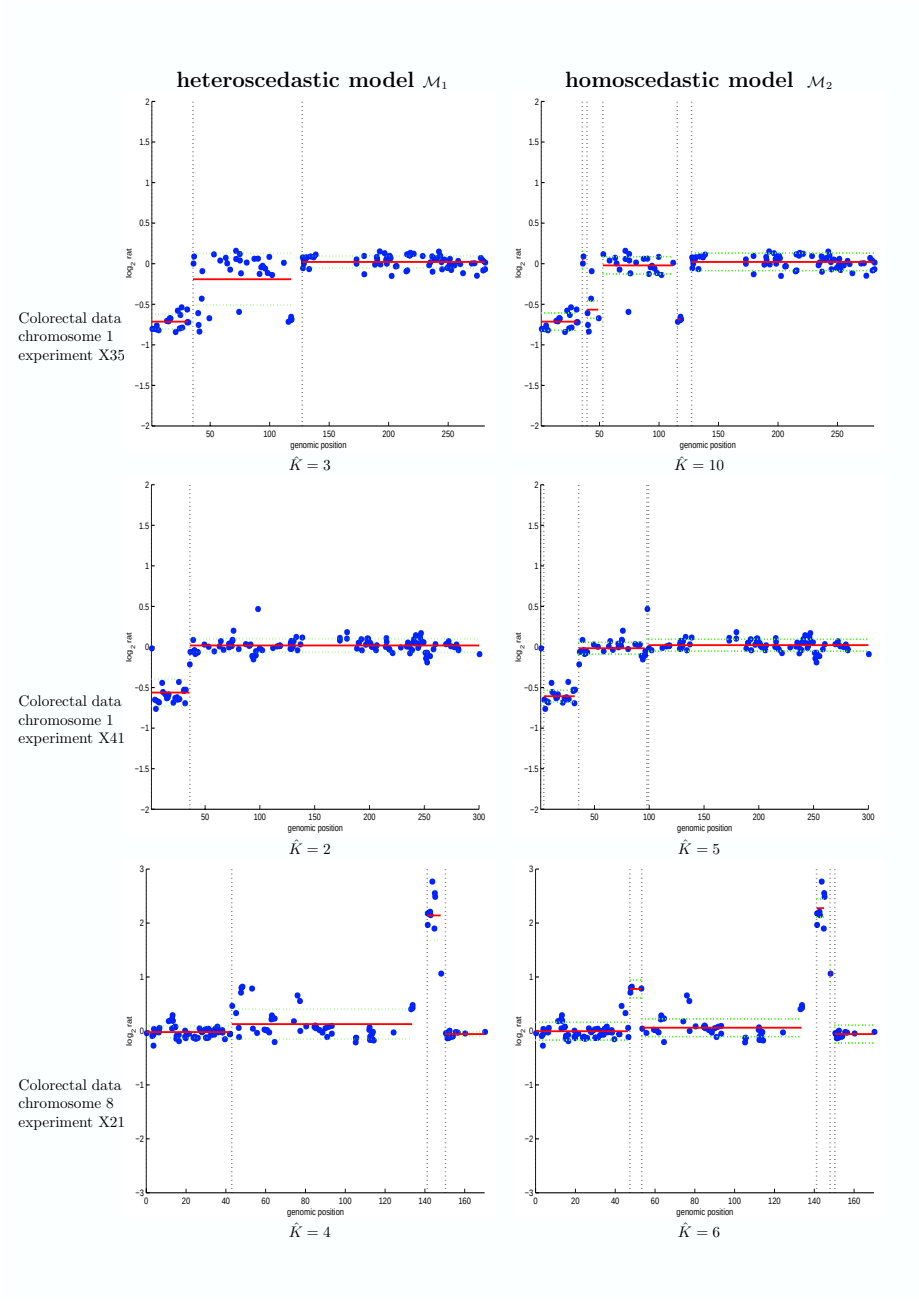


Figure 6

Comparison of segmentation results based on colorectal cancer data, using model Q_1 and Q_2. Results of the segmentation procedure for colorectal cancer data, chromosome 1 and chromosome 8. Fluoresence $\log_2$-ratios are plotted according to their location on the genome in megabases. Left profiles are segmented using model Q_1, and right profiles using model Q_2. Our criterion is used to estimate the number of segments.

BMC Bioinformatics 2005, **6**:27 http://www.biomedcentral.com/1471-2105/6/27

Table 1: Constants and penalty funtions for different penalized criteria, in a heteroscedastic model with *K* segments.

criterion	η	$pen(K)$
AIC	1	$2K$
BIG	$\frac{1}{2}\log(n)$	$2K$
Jong (2003)	10/3	$3K - 1$
Lebarbier (2003)	adaptive	$2K\left(c_1 + c_2 \log\left(\frac{n}{K}\right)\right)$
Lavielle (2003)	adaptive	$2K$

too large number of segments would lead to a large estimation error. A penalized version of the likelihood is used as a trade-off between a good adjustement and a reasonable number of parameters to estimate. It is noted

$$\text{Þ}_K = \hat{\text{P}}_K - \eta pen(K),$$

where $pen(K)$ is a penalty function that increases with the number of segments, and η is a constant of penalization. The estimated number of segments is such as :

$$\hat{K} = Argmax_K\left(\text{Þ}_K\right).$$

It is crucial to notice that the criterion which is penalized should provide the best partition of *K*-dimensional, *ie* for a fixed K the criterion has to be globally maximized to ensure convergence of the break-point estimators to the true break-points [14]. This optimum is provided by dynamic programming, but not by other algorithms [9,10].

Choice of the penalty function and constant

Classical penalized likelihoods use the number of independent continuous parameters to be estimated as a penalty function. Even though those criteria are widely used in the context of model selection, theoretical considerations suggest that they are not appropriate in the context of an exhaustive search for abrupt changes.

Let us focus on the penalty function in a first step. Table 1 provides a summary of different penalties. For classical information criteria, such as the Akaike Information Criterion and the Bayes Information Criterion, the penalty function equals to $2K$ (K means and K variances) for a heteroscedastic model with K segments. Penalized criteria have already been used in the context of array CGH data analysis to estimate the number of segments [9]. In addition to the $2K$ parameters, they implicitly consider that the break-points are also continuous parameters, leading to a new penalty function $pen(K) = 3K - 1$, which considers $K - 1$ break-points. Nevertheless, the characteristic of break-point detection models lies in the mixture of continuous parameters and discrete parameters that can not be counted as continuous parameters, since the number of possible configurations for K segments is finite and equals G_{n-1}^{K-1} (with n the total number of points) [13].

This leads to the definition of a new penalty function adapted to the special context of the exhaustive search of abrupt changes. This function (table 1) is proportional to the number of continuous parameters, but is also proportional to a new term in $\log\left(\frac{n}{K}\right)$ that takes the complexity of the visited configurations into account. It is written $pen(K) = 2K(c_1 + c_2 \log\left(\frac{n}{K}\right))$, where c_1 and c_2 are constant coefficients that have to be calibrated using numerical simulations. Since AIC and BIC and the criterion proposed in [9] do not consider the complexity of the visited models, they select a too high number of segments. The second term of the penalty is the penalty constant η. This term is constant in the case of AIC and BIC ($\eta = 1$, $\eta = \frac{1}{2}\log(n)$, respectively), and contributes to the oversegmentation as mentioned above. This can lead to an empirical choice for the constant, in order to obtain expected results based on *a priori* knowledge. For this reason, an arbitrary penalty constant can be chosen for the procedure to select a reasonable number of segments ($\eta = 10/3$ in [9]). Instead of an arbitrary choice for this constant, η can be adaptively chosen to the data [13,14]. Furthermore, when the number of segments is small with respect to the number of data points (which is the case in CGH data analysis), the log-term can be considered as a constant [14]. The author rather suggests to use the penalty function $pen(K) = 2K$ and to define an automatic procedure to choose the constant of penalization η adaptively.

We explain the estimation procedure for the penalty constant in the Methods Section.

The power of adaptive methods for model selection lies in the definition of a penalty that is not universal (such as in the case of AIC and BIC). This means that the dimension of the model is estimated adaptively to the data. The efficiency of such method has been shown on simulated data as well as on experimental results (Results Section), and adaptive model selection criteria seem to be very appropriate for array CGH data analysis.

Choice of modelling for array CGH data

Since the choice of modeling affects the resulting segmentation, it is crucial to provide guidelines for their use. This can be done with the interpretation of the statistical models in terms of their biological meaning. The difference between model Q_1 and Q_2 concerns the modeling of the variance: model Q_1 assumes that the variability of the signal is organized along the chromosome, whereas model Q_2 specifies that the variance is constant. Since it has been shown that the vast majority of clones all had the same response to copy number changes in the aneuploid cell lines [6], the use of model Q_2 would be justified regarding this experimental argument.

Outliers seem to be a major concern in microarray CGH data analysis. For instance, if only one BAC is altered whereas its neighbors are not, the conclusion could be either that it is biologically relevant, or that the signal is due to technical artefacts. Replications are crucial in this situation, as well as secondary validations. An other possibility could be that the BAC is misannotated: if the ratio is plotted at the wrong coordinate on the genome, it will appear as an outlier, when it is not. The importance of outlier identification is another argument in favor of model Q_2, that can detect changes for one data point, whereas with model Q_1 outliers would belong to segments with higher variance.

It has to be noted that classical models used in segmentation methods assume the independence of the data. This may be a reasonnable assumption for BAC arrays whose genome representation is approximately 1 BAC every 1.4 Mb [6]. Nevertheless, a new generation of arrays now provides a tiling resolution of the genome [15]. The overlapping of successive BACs could lead to statistical correlations that will require developments of new segmentation models for correlated processes.

Conclusions

Microarray CGH currently constitutes the most powerful method to detect gain or loss of genetic material on a genomic scale. To date, applications have been mainly restricted to cancer research, but the emerging potentialities of this technique have also been applied to the study of congenital and acquired diseases. As expression profile experiments require careful statistical analysis before any biological expertise, CGH microarray experiments will require specific statistical tools to handle experimental variability, and to consider the specificity of the the studied biological phenomena. We introduced a statistical method for the analysis of CGH microarray data that models the abrupt changes in the relative copy number ratio between a test DNA and a reference DNA. We discuss the effects of different modelings that can be used in segmentation methods, and suggest the use of a model that considers the homogeneity of the signal variability based on experimental arguments and regarding the specificity of array CGH data.

The main theoretical issue of array CGH data analysis lies in the estimation of the number of segments that requires the definition of appropriate penalty function and constant. We define a new procedure that estimates the number of segments adaptively to the data. This method selects the number of segments with high accuracy compared to previously mapped aberrations, and seems to be more efficient compared to others proposed to date. The use of dynamic programming remains central to localizing the break-points, and the simulation results show that when the good number of segments are selected, the algorithm localizes the break-points very close to the truth. Assessing the number of segments in a model is theoretically complex, and requires the definition of a precise model of inference. To that extent, microarray CGH analysis not only requires computational approaches, but also a careful statistical methodology.

Methods

Materials

We briefly present the data we used in this article. The first data we use in the Results Section consist of a single experiment on fibroblast cell lines (Coriell Cell lines) whose chromosomal aberrations have been previously mapped. Those defaults concern partial or whole chromosome aneuploidy. This data have been previously used by other authors [10]. The second group of data used in the Results section is described in [6]. A test genome of Bt474 cell lines is compared to a normal reference male genome. The last data set used is described in [16] and consists of 125 primary colorectal tumors that were surgically dissected and frozen. The arrays used for these analysis are BAC arrays described in [6].

Models and Likelihoods

In this section, we define the models Q_1 and Q_2. Let us consider a CGH profile, and note y_t, the $\log_2$-ratio of the

intensities for the t^{th} BAC on the genome. Precisely γ_t represents the average signal obtained from the replicated spots on the slide. BACs are the basic units in our model, and are ordered according to their physical position. We suppose that the γ_t are the realizations of independent random variables $\{Y_t\}_{t = 1...n}$, with Gaussian distributions $\mathcal{R}\left(\sigma_t, \omega_t^2\right)$. We assume that K - 1 changes affect the parameters of the distribution of the Ys, at unknown coordinates $(t_0, t_1, t_2,...,t_{K-1}, t_K)$ with convention $t_0 = 1$ and $t_K = n$, and that the parameters of the Ys distributions are constant between two changes:

$$\mathcal{Q}_1: \quad \forall t \in]t_{k-1}, t_k], \quad Y_t = \sigma_k + \kappa_t, \quad \kappa_t \sim \mathcal{R}\left(0, \omega_k^2\right),$$

$$\mathcal{Q}_2: \quad \forall t \in]t_{k-1}, t_k], \quad Y_t = \sigma_k + \kappa_t, \quad \kappa_t \sim \mathcal{R}\left(0, \omega^2\right).$$

where σ_k is the mean of the k^{th} segment. Model $\mathcal{Q}_1$ specifies that the variance is segment-specific (ω_k^2), whereas $\mathcal{Q}_2$ considers that the variance is common between segments (ω^2). Since BACs are supposed to be independent, the log-likelihood can be decomposed into a sum of "local" likelihoods, calculated on each segments: $\mathcal{P}_K = \sum_{k=1}^{K} \Lambda_k$, with

$$\mathcal{Q}_1 \quad : \quad \Lambda_k = -\frac{1}{2} \sum_{t=t_{k-1}+1}^{t_k} \left\{ \log(2\phi \times \omega_k^2) + \left[\frac{\gamma_t - \sigma_k}{\omega_k} \right]^2 \right\}$$

$$\mathcal{Q}_2 \quad : \quad \Lambda_k = -\frac{1}{2} \sum_{t=t_{k-1}+1}^{t_k} \left\{ \log(2\phi \times \omega^2) + \left[\frac{\gamma_t - \sigma_k}{\omega} \right]^2 \right\}.$$

Estimation of the segment's mean and variance

Given the number of segments K and the segments' coordinates $(t_0, t_1, t_2,...,t_{K-1}, t_K)$, we estimate the mean and the variance for each segment using maximum likelihood :

$$\hat{\sigma}_k = \frac{1}{t_k - t_{k-1}} \sum_{t=t_{k-1}+1}^{t_k} \gamma_t, \quad \hat{\omega}_k^2 = \frac{1}{t_k - t_{k-1}} \sum_{t=t_{k-1}+1}^{t_k} [\gamma_t - \hat{\sigma}_k]^2.$$

If the variance of the segments is homogeneous, its estimator is given by:

$$\hat{\omega}^2 = \frac{1}{n} \sum_{k=1}^{K} \sum_{t=t_{k-1}+1}^{t_k} \left[\gamma_t - \hat{\sigma}_k\right]^2.$$

Notice that when the segment coordinates are known, the estimation of the mean and variance for each segment is straightforward. Then, the key problem is to estimate K and $(t_0, t_1, t_2,...,t_{K-1}, t_K)$. We will proceed in two steps: in the first step, we will consider that the number of segments is known, and the problem will be to estimate the t_ks, that is, to find the best partition of a set of n individuals into K segments. In the second step, we will estimate the number of segments, using a penalized version of the likelihood.

A segmentation algorithm when the number of segments is known

When the number of segments K is known, the problem is to find the best partition of $\{1,...,n\}$ into K segments, according to the likelihood, where n is the size of the sample. An exhaustive search becomes impossible for large K since the number of partitions of a set with n elements into K segments is $\mathcal{G}_{n-1}^{K-1}$. To reduce the computational load, we use a dynamic programming approach (programs are coded in MATLAB language and are available upon request). Let $\hat{\mathcal{P}}_{k+1}(i, j)$ be the maximum log-likelihood obtained by the best partition of the data $\{Y(i), Y(i + 1),...,Y(j)\}$ into $k + 1$ segments, with k break-points, and let note $\hat{J}_{k+1}(i, j) = -2\hat{\mathcal{P}}_{k+1}(i, j)$. The algorithm is as follows:

$$k = 0, \quad \forall 0 \le i < j \le n \quad J_1(i, j) = \sum_{t=i+1}^{j} \left\{ \log(2\phi \times \omega_1^2) + \left[\frac{\gamma_t - \sigma_1}{\omega_1} \right]^2 \right\}$$

$$\forall k \in [1, K_{max}] \quad J_{k+1}(1, j) = \min_h \left\{ J_k(1, h) + J_1(h + 1, j) \right\}$$

Dynamic programming takes advantage of the additivity of the log-likelihood described above, considering that a partition of the data into $k + 1$ segments is a union of a partition into k segments and a set containing 1 segment. This approach presents two main advantages: it provides an exact solution for the global optimum of the likelihood [17], and reduces the computational load from $\mathcal{O}(n^K)$ to $\mathcal{O}(n^2)$ for a given K (the algorithm only requires the storage of an upper $n \times n$ triangular matrix). At the end of the procedure, the quantities $\hat{J}_1(1, n), ..., \hat{J}_{K_{max}}(1, n)$ are stored and will be used in the next step. Notice that this problem of partitioning is analogous to the search for the shortest path to travel from one point to another, where $\hat{J}_{k+1}(1, n)$ represents the total length of a $(k + 1)$-step-path connecting the point with coordinate 1 to the point with coordinate n.

An adaptive method to estimate the penalty constant

The purpose of this section is to explain an adaptive method to estimate the number of segments. Further theoretical developments can be found in [14]. If we consider that the likelihood $\hat{\mathcal{P}}_K$ measures the adjustment of a model with K segments to the data, we aim at selecting the

dimension for which $\hat{P}_K$ ceases to increase significantly. For this purpose, let us define a decreasing sequence (η) such as $\eta_0 = \leftarrow$ and

$$\forall i \geq 1 \quad \eta_i = \frac{\hat{P}_{K_{i+1}} - \hat{P}_{K_i}}{2K_{i+1} - 2K_i}.$$

If we represent the curve ($pen(K)$, $\hat{P}_K$), the sequence of η_i represents the slopes between points ($pen(K_{i+1})$, $\hat{P}_{K_{i+1}}$) and ($pen(K_i)$, $\hat{P}_{K_i}$), where the subset $\{(pen(K_i), \hat{P}_{K_i}), i \varnothing 1\})$ is the convex hull of the set $\{(pen(K), \hat{P}_K)\}$.

Since we aim at selecting the dimension for which $\hat{P}_K$ ceases to increase significantly, we look for breaks in the slope of the curve. We define l_i, the variation of the slope, that exactly corresponds to the length of the interval $]\eta_i, \eta_{i-1}]$: $l_i = \eta_{i-1} - \eta_i$. The length of these intervals is directly related to the second derivative of the likelihood. The automatic procedure to estimate the number of segments is then to calculate the second derivative (finite difference) of the likelihood:

$$\forall K \in \{1,...,K_{max}\} \; D_K = P_{K-1} - 2P_K + P_{K+1},$$

and we select the highest number of segments K such that the second derivative is lower than a given threshold :

$$\hat{K} = \max\{K \in \{1,...,K_{max}\} \mid D_K < s \times n\}$$

Other procedures have been developed to automatically locate the break in the slope of the likelihood. Nevertheless, the criterion we use can be interpreted geometrically and is easy to implement. The choice of the constant s is arbitrary. According to our experience, a threshold $s = -0.5$ seems appropriate for our purpose. A criticism that can be made to this procedure is its dependency on the threshold which is chosen. Nevertheless, it is important to point out that despite this thresholding the procedure remains adaptive, since the penalty constant is estimated according to the data.

Simulation studies

We performe numerical simulations to assess the sensitivity of our procedure to the addition of noise. In the first case, we simulate 100 points with $K^* = 5$ segments. In the first case Figure 3, the segments are regularly spaced and the difference of the means between two segments is $d = 1$. In the second case (Figure 4) the segments are irregularly spaced and the difference of the means varies between $d = 2$ and $d = 0.5$. The standard deviation of the Gaussian errors varies from $\omega = 0.1$ to $\omega = 2$. Each configuration is simulated 500 times, and we calculate the average selected number of segments over 500 simulations. In order to assess the performance of the dynamic programming algorithm, we calculate the empirical probability over 500 simulations for a break-point to be located at coordinate t (for $t = 1$ to 100).

Authors' contributions

FP developed the statistical models and the programs dedicated to array CGH data analysis, ML developped the adaptive selection of the number of segments. SR, CV and JJD supervised the study.

Acknowledgements

The authors want to thank Prs D. Pinkel and D. G. Albertson, and Dr E. Lebarbier for helpful discussion and comments, and L. Spector for editing the manuscript. CV is supported by grant NIH RO1 DK60540.

References

1. Albertson D, Collins C, McCormick F, Gray J: **Chromosome aberrations in solid tumors.** *Nature Genetics* 2003, **34:**369-376.
2. Albertson D, Pinkel D: **Genomic Microarrays in Human Genetic Disease and Cancer.** *Human Molecular Genetics* 2003, **12:**145-152.
3. Beheshti B, Park P, Braude I, Squire J: *Molecular Cytogenetics: Protocols and Applications* Humana Press; 2002.
4. Solinas-Toldo S, Lampel S, Stilgenbauer S, Nickolenko J, Benner A, Dohner H, Cremer T, Lichter P: **Matrix-based Comparative Genomic Hybridization: Biochips to Screen for Genomic Imbalances.** *Genes, Chromosomes and Cancer* 1997, **20:**399-407.
5. Pinkel D, Segraves R, Sudar D, Clark S, Poole I, Kowbel D, Collins C, Kuo W, Chen C, Zhai Y, Dairkee S, Ljung B, Gray J: **High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays.** *Nature Genetics* 1998, **20:**207-211.
6. Snijders AM, Nowak N, Segraves R, Blakwood S, Brown N, Conroy J, Hamilton G, Hindle AK, Huey B, Kimura K, Law S, Myambo K, Palmer J, Ylstra B, Yue JP, Gray JW, Jain A, Pinkel D, Albertson DG: **Assembly of microarrays for genome-wide measurement of DNA copy number.** *Nature Genetics* 2001, **29:**263-264.
7. Autio R, Hautaniemi S, Kauraniemi P, Yli-Harja O, Astola J, Wolf M, Kallioniemi A: **CGH-plotter: MATLAB toolbox for CGH-data analysis.** *Bioinformatics* 2003, **13:**1714-1715.
8. Eilers P, Menezes R: **Quantile smoothing of array CGH data.** *Bioinformatics* 2004 in press.
9. Jong K, Marchiori E, van der Vaart A, Ylstra B, Weiss M, Meijer G: *Applications of Evolutionary Computing: EvoWorkshops 2003: Proceedings, Springer-Verlag Heidelberg, chap. chromosomal breakpoint detection in human cancer* 2003, **2611:**54-65.
10. Olshen A, Venkatraman E, Lucito R, Wigler M: **Circular Binary segmentation for the analysis of array-based DNA copy number data.** *Biostatistics* 2004, **5(4):**557-572.
11. Hupe P, Stransky N, Thiery J, Radvanyi F, Barillot E: **Analysis of array CGH data: from signal ratio to gain and loss of DNA regions.** *Bioinformatics* 2004, **20(18):**3413-3422.
12. Fridlyand J, Snijders A, Pinkel D, Albertson D, Jain A: **Hidden Markov Models approach to the analysis of array CGH data.** *Journal of Multivariate Analysis* 2004, **90:**132-1533.
13. Lebarbier E: **Detecting Multiple Change-Points in the Mean of Gaussian Process by Model Selection.** *(to appear in) Signal Processing* 2005.
14. Lavielle M: **Using penalized contrasts for the change-point problem.** *(to appear in) Signal Processing* 2005.
15. Ishkanian A, Malloff C, Watson S, deLeeuw R, Chi B, Coe B, Snijders A, Albertson D, Pinkel D, Marra M, Ling V, MacAulay C, Lam W: **A tiling resolution DNA microarray with complete coverage of the human genome.** *Nature Genetics* 2004, **36(3):**299-303.
16. Nakao K, Mehta K, Fridlyand J, Moore DH, Jain AJ, Lafuente A, Wiencke J, Terdiman J, Waldman F: **High-resolution analysis of DNA copy number alterations in colorectal cancer by array-**

based comparative genomic hybridization. *Carcinogenesis* 2004, **25(8):**1345-1357.
17. Auger I, Lawrence C: **Algorithms for the optimal identification of segments neighborhoods.** *Bull Math Biol* 1989, **51:**39-54.

A Segmentation-Clustering problem for the analysis of array CGH data ⋆

F. Picard[1], S. Robin[1], E. Lebarbier[1], and J-J. Daudin[1]

Institut National Agronomique Paris-Grignon
UMR INA P-G/ENGREF/INRA MIA 518
16 rue Claude Bernard,75231 Paris cedex 05.
(e-mail: picard@inapg.fr)

Abstract Microarray-CGH experiments are used to detect and map chromosomal imbalances, by hybridizing targets of genomic DNA from a test and a reference sample to sequences immobilized on a slide. A CGH profile can be viewed as a succession of segments that represent homogeneous regions in the genome whose representative sequences (or BACs) share the same relative copy number on average. Segmentation methods constitute a natural framework for the analysis, but they do not assess a biological status to the detected segments. We propose a new model for this segmentation-clustering problem, combining a segmentation model with a mixture model. We present an hybrid algorithm to estimate the parameters of the model by maximum likelihood. This algorithm is based on dynamic programming and on the EM algorithm. We also propose to adaptively estimate the number of segments when the number of clusters is fixed. An example of our procedure is presented, based on publicly available data sets.
Keywords: Segmentation methods, Mixture Models, Dynamic Programming, EM algorithm, Model Selection.

Introduction

Chromosomal aberrations often occur in solid tumors: tumor suppressor genes may be inactivated by physical deletion, and oncogenes activated via duplication in the genome. The purpose of array-based Comparative Genomic Hybridization (array CGH) is to detect and map chromosomal aberrations, on a genomic scale, in a single experiment. Since chromosomal copy numbers can not be measured directly, two samples of genomic DNA (referred as the reference and the test DNA) are differentially labelled with fluorescent dyes and competitively hybridized to known mapped sequences (referred as BACs) that are immobilized on a slide. Subsequently, the ratio of the intensities of the two fluorochromes is computed and a CGH profile is constituted for each chromosome when the $\log_2$ of fluorescence ratios are ranked and plotted according to the physical position of their corresponding BACs on the genome.

⋆ This article has been published in the proceedings of the Conference "International Symposium on Applied Stochastic Models and Data Analysis", Brest 2005.

Each profile can be viewed as a succession of 'segments' that represent homogeneous regions in the genome whose BACs share the same relative copy number on average. Array CGH data are normalized with a median set to $\log_2(\text{ratio}) = 0$ for regions of no change, segments with positive means represent duplicated regions in the test sample genome, and segments with negative means represent deleted regions. It has to be noted that even if the underlying biological process is discrete (counting of relative copy numbers of DNA sequences), the signal under study is viewed as being continuous, because the quantification is based on fluorescence measurements, and because the possible values for chromosomal copy numbers in the test sample may vary considerably, especially in the case of clinical tumor samples that present mixtures of tissues of different natures.

Segmentation methods seem to be a natural framework to handle the spatial coherence on the genome that is a specificity of array CGH data [Autio *et al.*, 2003,Jong *et al.*, 2003]. These methods provide a partition of the data into segments, each segment being characterized by its mean and variance μ_k and σ_k^2 in the Gaussian case. Nevertheless, even if the data are instrinsically segmented, they are also structured into clusters which have a biological interpretation: we can define a group of deleted segments, a group of unaltered segments, and many groups of amplified segments for instance. This refinement means that the mean and variance of each segment should be restricted to a finite set such that $\mu_k \in \{m_1, \ldots, m_P\}$ and $\sigma_k^2 \in \{s_1^2, \ldots, s_P^2\}$ if the segments are structured into P clusters.

We propose to handle this segmentation-clustering problem combining a segmentation model and a mixture model to assign a biological status to segments. Section 1 is devoted to the precise definition of such model. In Section 2 we propose an hybrid algorithm combining dynamic programming and the EM algorithm to alternatively estimate the break-point coordinates and the parameters of the mixture. The convergence properties of this algorithm are presented.

Once the parameters of the model have been estimated, a key issue is the estimation of the number of segments and of the number of clusters. We propose to estimate the number of segments when the number of groups is fixed, using a penalized version of the likelihood. We propose to apply the procedure defined by [Lavielle, 2005], that has been successfully applied to array CGH data [Picard *et al.*, 2005]. An example of our method is provided in Section 3, using publicly available data sets.

1 A new model for the segmentation-clustering problem

Let y_t represent the $\log_2$ ratio of the t^{th} BAC on the genome and $y = \{y_1 \ldots, y_n\}$ the entire CGH profile constituted by n data points. We suppose that y is the realization of a Gaussian process Y whose mean and variance are

affected by $K+1$ abrupt changes at unknown coordinates $T = \{t_0, t_1, \ldots, t_K\}$ with the convention $t_0 = 1$ and $t_K = n$. This defines a partition of the data into K segments of length n_k. We write Y as $\{Y^1, \ldots, Y^K\}$, where $Y^k = \{Y_t, t \in I_k\}$, with $I_k = \{t, t \in]t_{k-1}, t_k]\}$. We suppose that the mean and the variance of the process are constant between two break-points and they are noted μ_k and σ_k^2.

More than classical segmentation models, we assume that the mean and variance of the segment Y^k can only take a limited number of values with $\mu_k \in \{m_1, \ldots, m_P\}$, and $\sigma_k^2 \in \{s_1^2, \ldots, s_P^2\}$. In addition to the spatial organization of the data, via the partition T, there exists a secondary structure of the process into P clusters, and we adopt a mixture model approach to handle this problem.

We assume that the partitionned data $\{Y^1, \ldots, Y^K\}$ are structured into P clusters with weights π_p ($\sum_p \pi_p = 1$). We introduce a sequence of independent hidden random variables, $Z^k = \{Z_1^k, \ldots, Z_P^k\}$ such that Z^k is distributed according to a multinomial distribution consisting of one draw on P categories with probabilities $\pi_1, \ldots, \pi_P$. The mixing proportions $\pi_1, \ldots, \pi_P$ then represent the *prior* probability for segment Y^k to belong to the p^{th} component, while the *posterior* probability of membership to the p^{th} component with y^k having been observed is: $\tau_p^k = \Pr\left\{Z_p^k = 1 | Y^k = y^k\right\}$. Contrary to classical mixture models, where the indicator variables provide informations about the labelling of individual data points (which would be Y_t in our case), our model focuses on the belonging of the segments Y^k to different clusters.

We focus on the case where the data are supposed to be drawn from a mixture of Gaussian densities, with parameters $\theta_p = (m_p, s_p^2)$. If we suppose the indepence of individual data points Y_t within a segment, the model can be formulated as follows:

$$Y^k | Z_p^k = 1 \sim \mathcal{N}(m_p \mathbb{1}_{n_k}, s_p^2 I_{n_k}).$$

We note $\psi = \{\pi_1, \ldots, \pi_{P-1}, \theta_1, \ldots, \theta_P\}$ the vector of unknown independent parameters of the mixture, and the log-likelihood of the model is:

$$\log \mathcal{L}_{KP}(T, \psi) = \sum_{k=1}^{K} \log \left\{ \sum_{p=1}^{P} \pi_p f(y^k; \theta_p) \right\}.$$

$f(y^k; \theta_p)$ represents the conditional density of a vector of size n_k. Our purpose is to optimize this likelihood to estimate the parameters of the model using an hybrid algorithm.

2 An hybrid algorithm combining the EM algorithm and Dynamic Programming

The principle of our algorithm is simple: when the break-point coordinates T are known, the EM algorithm is used to estimate the mixture parameters

ψ, and once ψ has been estimated, the break-point coordinates are computed using dynamic programming. This algorithm requires the *prior* knowledge of both the number of segments K and the number of populations P. The choice for these components of the model will be discussed in a later section.

2.1 Estimating the break-point coordinates when the mixture parameters are known

When the number of segments K and the parameters of the mixture are known, the problem is to find the best K-dimensional partition of the data according to the log-likelihood $\log \mathcal{L}_{KP}(T, \psi)$. Since the number of of partitions of a set with n elements into K segments is $\mathcal{C}_{n-1}^{K-1}$, and because of the additivity in K of the log-likelihood, we use a dynamic programming approach to reduce the computational load from $\mathcal{O}(n^K)$ to $\mathcal{O}(n^2)$, as suggested by [Auger and Lawrence, 1989].

Let $\hat{C}_{k+1,P}(i, j; \psi)$ be the maximum log-likelihood obtained by the best partition of the data $Y^{ij} = \{Y_i, Y_{i+1}, ..., Y_j\}$ into $k+1$ segments, when the mixture parameters ψ are known. The algorithm starts as follows: for $k = 0$ and for $(i, j) \in [1, n]^2$, with $i < j$, calculate:

$$\hat{C}_{1,P}(i, j; \psi) = \log \left\{ \sum_{p=1}^{P} \pi_p f(y^{ij}; \theta_p) \right\} = \log \left\{ \sum_{p=1}^{P} \pi_p \prod_{t=i+1}^{j} f(y_t; \theta_p) \right\}.$$

$\hat{C}_1(i, j; \psi)$ represents the local log-likelihood for segment Y^{ij}. Then the algorithm is run as follows:

$$\forall k \in [1, K_{max}] \quad \hat{C}_{k+1,P}(1, j; \psi) = \max_h \left\{ \hat{C}_{k,P}(1, h; \psi) + \hat{C}_{1,P}(h+1, j; \psi) \right\}$$

Dynamic programming considers that a partition of the data into $k+1$ segments is a union of a partition into k segments and a set containing 1 segment. More than a reduction in the computational load, this approach provides an exact solution for the global optimum of the likelihood, that will be central for downstream model selection procedures.

2.2 Estimate the mixture model parameters when the break-point coordinates are known

When the break-point coordinates are known, we dispose of a partition of the data into K segments $\{Y^1, \ldots, Y^K\}$. This partition defines the statistical units of a mixture model whose parameters have to be estimated. The purpose is then to maximize the log-likelihood of the model $\log \mathcal{L}_{KP}(T, \psi)$ according to ψ. As it is the case in classical mixture models, the direct optimization of the likelihood is impossible, but can be handled using the EM

algorithm in the complete-data framework [Dempster *et al.*, 1977]. Let us define the complete-data log-likelihood:

$$\log \mathcal{L}^c_{KP}(T, \psi) = \sum_{k=1}^{K} \sum_{p=1}^{P} z_p^k \log \left\{ \pi_p f(y^k; \theta_p) \right\}.$$

The EM algorithm is as follows:

- *E*-**step**: compute the conditional expectation of the complete-data log-likelihood, given the observed data Y, using the current fit $\psi^{(h)}$ for ψ.

$$Q_{KP}(\psi|\psi^{(h)}; T) = \sum_{k=1}^{K} \sum_{p=1}^{P} \tau_p^{k(h)} \log \left\{ \pi_p f(y^k; \theta_p) \right\},$$

with

$$\tau_p^{k(h+1)} = \frac{\pi_p^{(h)} f(y^k; \theta_p^{(h)})}{\sum_{\ell=1}^{P} \pi_\ell^{(h)} f(y^k; \theta_\ell^{(h)})}.$$

- *M*-**step**: The M-step on the $(h+1)^{th}$ iteration requires the global maximization of $Q_{KP}(\psi|\psi^{(h)}; T)$ with respect to ψ to give the updated estimate $\psi^{(h+1)}$:

$$\psi^{(h+1)} = \underset{\psi}{\text{Argmax}} \left\{ Q_{KP}(\psi|\psi^{(h)}; T) \right\}.$$

2.3 Convergence properties of the hybrid algorithm

The proof of the convergence of our algorithm is based on the properties of both dynamic programming and EM. It can be seen that both algorithms are linked through the likelihood they alternatively optimize: the incomplete-data likelihood of the mixture of segments.

Dynamic programming globally optimizes the likelihood with respect to T. At iteration (ℓ) we have:

$$\log \mathcal{L}_{KP} \left(T^{(\ell+1)}; \psi^{(\ell)} \right) \geq \log \mathcal{L}_{KP} \left(T^{(\ell)}, \psi^{(\ell)} \right).$$

On the other hand, the key convergence property of the EM algorithm is the increase of the incomplete-data log-likelihood at each step [Dempster *et al.*, 1977]:

$$\log \mathcal{L}_{KP} \left(T^{(\ell)}, \psi^{(\ell+1)} \right) \geq \log \mathcal{L}_{KP} \left(T^{(\ell)}, \psi^{(\ell)} \right).$$

Put together, our algorithm generates a sequence $\left(T^{(\ell)}, \psi^{(\ell)} \right)_{\ell \geq 0}$ that increases the incomplete-data log-likelihood such as:

$$\log \mathcal{L}_{KP} \left(T^{(\ell+1)}, \psi^{(\ell+1)} \right) \geq \log \mathcal{L}_{KP} \left(T^{(\ell)}, \psi^{(\ell)} \right).$$

3 Estimating the number of segments K when the number of clusters P is fixed.

Once the parameters of the model have been estimated (for a fixed K and a fixed P), the next question is the estimation of the number of segments and of the number of clusters. Since the principal objective of biologists is rather the detection of biological events on the genome rather than the clustering of those events into groups, we choose to focus on the estimation of the number of segments when the number of groups is fixed.

The maximum of the log-likelihood $\log \hat{\mathcal{L}}_{KP} = \log \mathcal{L}_{KP}(\hat{T}, \hat{\psi})$ can be viewed as a quality measurement of the fit to the data of the model with K segments. In classical segmentation models, this quantity is maximal when the number of segments equals the number of data points. Nevertheless, as our model also considers the clustered nature of segments, it appears that the quality of fit of the model is not always increasing with the number of segments, as shown in Figure 1. For $P = 2$ the incomplete-data log-likelihood is decreasing for a number of segments $K \geq 12$ for instance. This behavior of the model can be interpreted as follows: since the segmentation-clustering model is under the constraint $P \leq K$, the addition of new segments can lead to contiguous segments affected to the same cluster. This configuration leads to an increase in the number of parameters (one additional break-point) without any gain for the fit of the mixture model. These considerations imply that there will be a number of segments above which the addition of a new segment will not increase the log-likelihood.

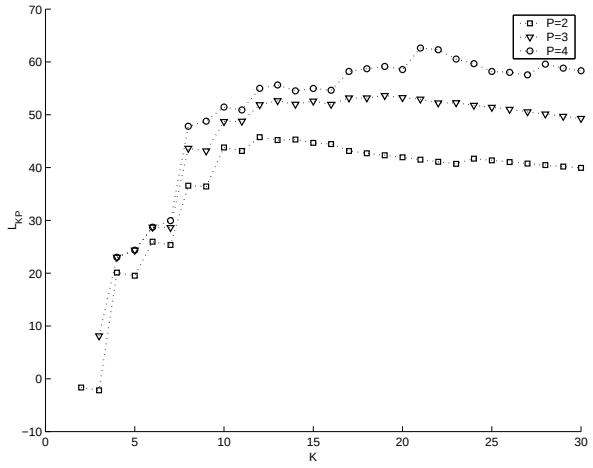


Figure1. Evolution of the incomplete-data log-likelihood $\log \hat{\mathcal{L}}_{KP}$ with the number of segments K for different number of clusters ($P = 2, 3, 4$).

A penalized version of the likelihood is used as a trade-off between a good adjustement and a reasonnable number of break-points. The estimated number of segments is such as:

$$\hat{K}_P = \underset{K}{Argmax} \left(\hat{\mathcal{L}}_{KP} - \beta_P pen(K) \right),$$

with $pen(K)$ a penalty function that increases with the number of segments, and β_P a penalty constant. The definition of an appropriate penalty function and constant has lead to theoretical developments in the context of break-point detection models. Recently, [Lavielle, 2005] proposed to use an adaptive procedure to estimate the penalty constant, that has been successfully applied to array CGH data [Picard *et al.*, 2005]. The principle of this procedure is to find the number of segments for which the log-likelihood ceases to increase significantly. It is geometrically linked to the finding of the number of segments for which the second derivative of the log-likelihood function is maximal (see [Lavielle, 2005] for further details). A result of our procedure is shown in Figure 2. For a number of clusters $P = 3$, the adpative procedure estimates a number of segments $\hat{K}_3 = 10$. This leads to a profile which presents three types of segments that can be interpreted in terms of biological groups, as shown in Figure 2.

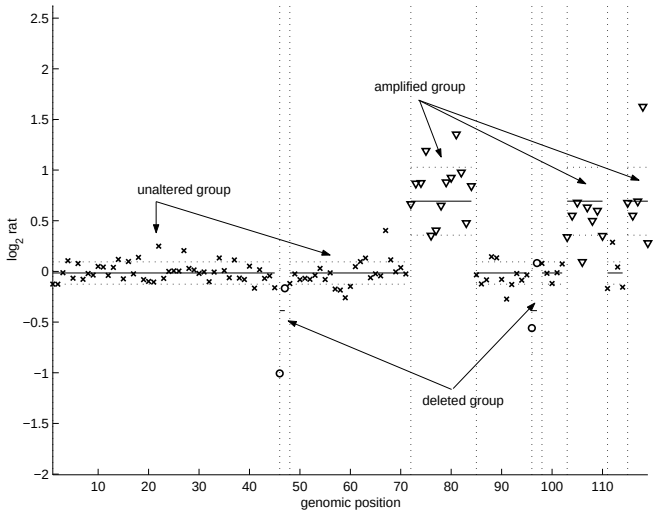


Figure2. Result of the segmentation-clustering procedure for a fixed number of clusters $P = 3$ and an estimated number of segments $\hat{K}_3 = 10$. These data concern chromosome 1 of breast cancer cell lines Bt474.

4 Discussion

Microarray CGH currently constitutes the most powerful method to detect gain or loss of genetic material on a genomic scale. We introduced a statistical methodology for the analysis of CGH microarray data, that combines segmentation methods and clustering techniques. It terms of modeling, the discovery of homogeneous regions clustered into groups could have been handled using Hidden Markov Models, as in [Fridlyand *et al.*, 2004]. In those models, the segmented structure of the data is recovered using the *posterior* probability of membership of individual data points into a fixed number of hidden groups, whereas our method focuses on the labelling of segments to hidden groups. Moreover, a property of Hidden Markov Models is that the distance between two 'break-points' is dependent on the probability distribution of the hidden sequence: the within-class sojourn time is geometrically distributed. Our approach is free from those constraints, since break-point coordinates are 'real' parameters of the model that are not randomly distributed.

The definition of this new model leads to unusual statistical considerations: it appears that the statistical units of the mixture model (when the segmentation is known) are segments of different size. Since the partition of the data is random, the individuals of the mixture model themselves are random. This explains the difficulty of the joint estimation of K the number of segments, and P the number of clusters, since classical model selection procedures are based on a compromize between a reasonnable number of parameters to estimate given a fixed number of statistical units. To these extents, this problem of model selection for two components remains an open question.

References

[Auger and Lawrence, 1989]I.E. Auger and C.E. Lawrence. Algorithms for the optimal identification of segments neighborhoods. *Bull. Math. Biol.*, 51:39–54, 1989.

[Autio *et al.*, 2003]R. Autio, S. Hautaniemi, P. Kauraniemi, O. Yli-Harja, J. Astola, M. Wolf, and A. Kallioniemi. CGH-plotter: MATLAB toolbox for cgh-data analysis. *Bioinformatics*, 13:1714–1715, 2003.

[Dempster *et al.*, 1977]A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.

[Fridlyand *et al.*, 2004]J. Fridlyand, A. Snijders, D. Pinkel, D.G. Albertson, and A.N. Jain. Hidden markov models approach to the analysis of array CGH data. *Journal of Multivariate Analysis*, 90(1):132–1533, 2004.

[Jong *et al.*, 2003]K. Jong, E. Marchiori, A. van der Vaart, B. Ylstra, M. Weiss, and G. Meijer. *Applications of Evolutionary Computing: EvoWorkshops 2003: Proceedings*, volume 2611, chapter chromosomal breakpoint detection in human cancer, pages 54–65. Springer-Verlag Heidelberg, 2003.

[Lavielle, 2005]M. Lavielle. Using penalized contrasts for the change-point problem. *(to appear in) Signal Processing*, 2005.

[Picard *et al.*, 2005]F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J-J. Daudin. A statistical approach for CGH microarray data analysis. *BMC Bioinformatics*, 6:27, 2005.

List of Figures

List of Tables

References

Albertson, D.G., C. Collins, F. McCormick, and J. Gray (2003). Chromosome aberrations in solid tumors. *Nature Genetics* **34**, 369–376.

Albertson, D.G. and Dan Pinkel (2003). Genomic microarrays in human genetic disease and cancer. *Human Molecular Genetics* **12**, 145–152.

Armour, J.A.L., D.E. Barton, D.J. Cockburn, and G.R. Taylor (2002). The detection of large deletions or duplications in genomic DNA. *Human Mutation* **20**, 325–337.

Auger, I.E. and C.E. Lawrence (1989). Algorithms for the optimal identification of segments neighborhoods. *Bull. Math. Biol.* **51**, 39–54.

Autio, R., S. Hautaniemi, P. Kauraniemi, O. Yli-Harja, J. Astola, M. Wolf, and A. Kallioniemi (2003). CGH-plotter: MATLAB toolbox for CGH-data analysis. *Bioinformatics* **13**, 1714–1715.

Avery, P.J. and D.A. Henderson (1999). Detecting a changed segment is DNA sequences. *Appl.Statis. 48*(4), 489–503.

Ball, C.A., Y. Chen, S. Panavally, G. Sherlock, T. Speed, P.T. Spellman, and Y.H. Yang (2003). *Section7: An introduction to microarray bioinformatics.* In DNA Microarrays: A Molecular Cloning Manual. Cold Spring Harbor Press.

Barry, D.B. and J.A. Hartigan (1993). A Bayesian analysis for change-point problems. *JASA 88*(421), 309–319.

Basseville, N. and I. Nikiforov (1993). *Detection of abrupt changes. Theory and application.* Prentice Hall Information and system sciences series.

Beheshti, B., P.C. Park, I. Braude, and J.A. Squire (2002). *Molecular Cytogenetics: Protocols and Applications.* Humana Press.

Bellman, R.E. and S.E. Dreyfus (1962). *Applied dynamic programming.* Princeton University Press.

Bernheim, A., J.L. Huret, M. Guillaud-Bataille, O. Brison, and J. Couturier (2004). de la cytogénétique à la cytogénomique des tumeurs. *Bull Cancer 91*(1), 29–43.

Biernacki, C., G. Celeux, and G. Govaert (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence 22*(7), 719–725.

Biernacki, C., G. Celeux, and G. Govaert (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics and data analysis* **41**, 561–575.

Birgé, L. and P. Massart (2001). Gaussian model selection. *J. European Math. Soc.* **3**, 203–268.

Braun, J. V., R.K. Braun, and H.G Muller (2000). Multiple change-point fitting via quasilikelihood, with application to DNA sequence segmentation. *Biometrika* **87**, 301–314.

Braun, J. V. and H.G Muller (1998). Statistical methods for DNA sequence segmentation. *Statistical Science 13*(2), 142–162.

Breiman, L., J.H. Friedman, R.A. Olshen, and C.J. Stone (1984). *Classification and Regression Trees*. Chapman & Hall.

Broniatowski, M., G. Celeux, and J. Dielbolt (1983). Reconnaissance de mélanges de densités par un algorithme d'apprentissage probabiliste. *Data analysis and informatics* **3**, 359–374.

Butte and Atul (2002). The use and analysis of microarray data. *Nature Reviews* **1**, 951–960.

Carlin, B.P. (1992). Hierarchical Bayesian analysis of change-point problems. *Appl. Statist. 41*(2), 389–405.

Casperson, T., Farber S., Foley GE., and et al. (1968). Chemical differenciation along metaphase chromosomes. *Exp. Cell Res* **49**, 219–222.

Castellan, G. (2000). Histogram selection with an Akaike type criterion. *C. R. Acad. Sci., Paris, Sér. I, Math. 330*(8), 729–732.

Celeux, G. and J. Dielbolt (1985). The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistic Quaterly* **2**, 73–82.

Celeux, G. and J. Dielbolt (1992). A stochastic approximation type EM algorithm for the mixture problem. *Stichastics and Stochastic reports* **41**, 119–134.

Chen, W., F. Erdogan, HH. Ropers, S. Lenzner, and R. Ullman (2005). CGH-PRO – a comprehensive data analysis tool for array CGH. *BMC Bioinformatics 6*(1), 85.

Chong, T.T-L (2001). Estimating the locations and number of change-points by the sample splitting method. *Statist. Papers 42*(1), 53–79.

Cleveland, W.S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association* **74**, 829–836.

Cobb, G.W. (1978). The problem of the Nile. Conditional solution to a change-point problem. *Biometrika 65*(2), 243–251.

Davies, J.J., I. M. wilson, and W.L. Lam (2005). Array CGH technologies and their applications to cancer genomics. *Chromosome research* **13**, 237–248.

Delyon, B., M. Lavielle, and E. Moulines (1999). Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics 27*(1), 94–28.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B* **39**, 1–38.

Dias, J.G. and M. Wedel (2004). An empirical comparison of EM, SEM and MCMC performances for problematic Gaussian mixture likelihoods. *Statistics and Computing* **14**, 323–332.

Efron, B. and D.V. Hinkley (1978). Assessing the accuracy of the maximum likelihood estimator: observed versus expected Fisher information. *Biometrika 65*(3), 457–487.

Eilers, PH. and RX.de Menezes (2005). Quantile smoothing of array CGH data. *Bioinformatics 21*(7), 1146–1153.

Ephraim, Y. (2002). Hidden Markov processes. *IEEE Transactions on Information theory 48*(6), 1518–1569.

Fraley, C. and A.E. Raftery (1998). How many clusters ? *The Computer Journal 41*(8), 578–587.

Fridlyand, J., A. Snijders, D. Pinkel, D.G. Albertson, and A.N. Jain (2004). Hidden Markov models approach to the analysis of array CGH data. *Journal of Multivariate Analysis 90*(1), 132–1533.

Gassiat, E. and D. Dacunha-Castelle (1997). Estimation of the number of components in a mixture. *Bernoulli* **3**, 279–299.

Gey, S. and E. Lebarbier (2002). A CART based algorithm for detection of multiple change-points in the mean of large samples. Technical Report 10, Université Paris Sud.

Gey, S. and E. Nedelec (2002). Risk Bounds for CART Regression Trees. *MSRI Proceedings on Nonlinear Estimation and Classification*.

Ghorbanzdeh, D. (1995). Un test de détection de rupture de la moyenne dans un modèle gaussien. *Rev. Statist. Appl. 43*(2), 67–76.

Green, P.J. (1990). On the use of the EM algorithm for penalized likelihood estimation. *JRSS-B* **52**, 443–452.

Green, P.J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika 82*(4), 711–732.

Hawkins, D.M. (2001). Fitting multiple change-point models to data. *Computational Statistics and data analysis* **37**, 323–341.

Hupe, P., N. Stransky, JP. Thiery, F. Radvanyi, and E. Barillot (2004). Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics 20*(18), 3413–3422.

Huret, J., P. Dessen, and A. Bernheim (2003). Atlas of genetics and cytogenetics in oncology and haematology. http://www.infobiogen.fr/services/chromcancer/.

Iafrate, A.J., L. Feuk, M.N. Rivera, M. L. Listewnick, P. K. Donahoe, Y. Qi, S.W. Scherer, and C. Lee (2004). Detection of large-scale variation in the human genome. http://www.nature.com/naturegenetics.

Ikemura, T., K. Wada, and S. Aota (1990). Giant G+C % mosaic structures of the human genome found by rearrangement of GenBank human DNA sequences according to genetic positions. *Genomics* **8**, 207–216.

Ishkanian, A.S., C.A. Malloff, S.K. Watson, R.J. deLeeuw, B. Chi, B.P. Coe, A. Snijders, D.G. Albertson, D. Pinkel, M.A. Marra, V. Ling, C. MacAulay, and W.L. Lam (2004). A tiling resolution DNA microarray with complete coverage of the human genome. *Nature Genetics 36*(3), 299–303.

Jong, K., E. Marchiori, A. van der Vaart, B. Ylstra, M. Weiss, and G. Meijer (2003). *Applications of Evolutionary Computing: EvoWorkshops 2003: Proceedings*, Volume 2611, Chapter chromosomal breakpoint detection in human cancer, pp. 54–65. Springer-Verlag Heidelberg.

Kallioniemi, A., O.P. Kallioniemi, D. Sudar, D. Rutovitz, J.W. Gray, F. Waldman, and D. Pinkel (1992). Comparative genomic hybridization for molecular cytogenetics analysis of solid tumors. *Science* **258**, 818–821.

Kass, E. and A.E. Raftery (1995). Bayes factors. *Journal of the American statistical Association 90*(430), 773–795.

Kerr, M.K. and G. Churchill (2001). Experimental design for gene expression microarrays. *Biostatistics* **2**, 183–201.

Kim, SY., SW. Nam, SH. Lee, WS. Park, NJ. Yoo, and YJ. Chung (2005). ArrayCyGHt: a web application for analysis and visualization of array-CGH data. *Bioinformatics 21*(10), 2554–2555.

Lai, W.R., M.D. Johnson, R. Kucherlapati, and P. J. Park (2005). Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics 0*(0), 1–8.

Lavielle, M. (1998). Optimal segmentation of random processes. *IEEE Transactions on signal processing 46*(5), 1365–1373.

Lavielle, M. (1999). Detection of multiple changes in a sequence of dependent variables. *Stoch. Proc. and Appl.* **83**, 79–102.

Lavielle, M. (2005). Using penalized contrasts for the change-point problem. *Signal Processing 85*(8), 1501–1510.

Lavielle, M. and E. Lebarbier (2001). An application of MCMC methods for the multiple change-points problem. *Signal Processing* **81**, 39–53.

Lavielle, M. and E. Moulines (2000). Least squares estimation of an unknown number of shifts in a time series. *Journal of Time series analysis 21*(1), 33–59.

Lebarbier, E. (2002). *Quelques approches pour la détection de ruptures à horizon fini*. Ph.D. thesis, Université Paris XI Orsay.

Lebarbier, E. (2005). Detecting multiple change-points in the mean of Gaussian process by model selection. *Signal Processing* **85**, 717–736.

Lebarbier, E. and T. Mary-Huard (2004). Le critère BIC: fondements théoriques et interprétation. Technical Report 5315, INRIA.

Leung, Y.F. and D. Cavalieri (2003). Fundamentals of cDNA microarray data analysis. *Trends in Genetics 19*(11), 649–659.

Locke, D.P., R. Segraves, L. Carbone, N. Archidiacono, D.G. Albertson, D. Pinkel, and E.E. Eichler (2003). Large scale variation among human and great ape genomes determined by array comparative genomic hybridization. *Genome Research* **13**, 347–357.

Louis, T. A. (1982). Finding the observed information matrix when using the EM-algorithm. *JRSS-B 44* (2), 226–233.

McLachlan, G. and D. Peel (2000). *Finite Mixture Models*. Wiley Inter-Science.

Meilijson, I. (1989). A fast improvement to the EM algorithm in its own terms. *JRSS-B 51* (1), 127–138.

Mitelman, F., B. Johansson, and F. Mertens (2003). Database of chromosome aberrations in cancer. http://www.cgap.nci.nih.gov/Chromosomes/Mitelman.

Muri, F. (1997). *Comparaison d'algorithmes d'identification de chaînes de Markov cachées et application à la détection de régions homogènes dans les séquences d'ADN*. Ph.D. thesis, Université René Descartes Paris V.

Nakao, K., K.R. Mehta, J. Fridlyand, D. H. Moore, A.J.Jain, A. Lafuente, J.W. Wiencke, J.P. Terdiman, and F.M. Waldman (2004). High-resolution analysis of DNA copy number alterations in colorectal cancer by array-based comparative genomic hybridization. *Carcinogenesis 25* (8), 1345–1357.

Nicolas, P. (2003). *Mise au point et utilisation de modèles de chaînes de Markov cachées pour l'étude des séquences d'ADN*. Ph.D. thesis, Université d'Evry.

Olshen, A. B. and E. S. Venkatraman (2002). Change-point analysis of array-based comparative genomic hybridization data. Technical report, Memorial Sloan-Kettering Cancer Center, http://www.mskcc.org/mskcc/html/14044.cfm.

Picard, D. (1985). Testing and estimating change-points in time series. *J. Applied Prob.* **17**, 841–867.

Picard, F., S. Robin, M. Lavielle, C. Vaisse, and J-J. Daudin (2005). A statistical approach for CGH microarray data analysis. *BMC Bioinformatics* **6**, 27.

Pinkel, D., R. Segraves, D. Sudar, S. Clark, I. Poole, D. Kowbel, C. Collins, W. Kuo, C. Chen, Y. Zhai, S.H. Dairkee, B. Ljung, and J.W. Gray (1998). High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nature Genetics* **20**, 207–211.

Pollack, J.R., C. M. Perou, A.A. Alizadeh, M.B. Eisen, A. Pergamenschikov, C. F. Williams, S.S. Jeffreys, D. Botstein, and P. O. Brown (1999). Genome-wide analysis of DNA copy-number changes using cDNA microarrays. *Nature Genetics* **23**, 41–46.

Polzehl, J. and S. Spokoiny (2000). Adaptive weights smoothing with applications to image restoration. *J. Roy. Statistical Society, Series B 62* (2), 335–354.

Rabiner, L.R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*(2), 257–286.

Robin, S., S. Schbath, and F. Rodolphe (2003). *ADN, mots et modèles*. Belin.

Schbath, S. (2000). An overview on the distribution of word counts in Markov chains. *Journal of Computational Biology 7*(1), 193–201.

Schwartz, G. (1978). Estimating the dimension of a model. *Annals of Statistics* **6**, 461–464.

Sebat, J., B. Lakshmi, J. Troge, J. Alexander, J. Young, P. Lundin, S. Maner, H. Massa, M. Walker, M. Chi, N. Navin, R. Lucito, J. Healy, J. Hicks, K. Ye, A. Reiner, T.C. Gilliam, B. Trask, N. Patterson, A. Zetterberg, and M. Wigler (2004). Large scale copy number polymorphism in the human genome. *Science* **305**, 525–528.

Sen, A. and M.S. Srivastava (1975). On test for detecting a change in mean. *Annals of Statistics* **3**, 98–108.

Siegmund, D. (1988). Confidence sets in change-point problems. *International Statistical Review 56*(1), 31–48.

Snijders, A. M., N. Nowak, R. Segraves, S. Blakwood, N. Brown, J. Conroy, G. Hamilton, A. K. Hindle, B. Huey, K. Kimura, S. Law, K. Myambo, J. Palmer, B. Ylstra, J. P. Yue, J. W. Gray, A.N. Jain, D. Pinkel, and D. G. Albertson (2001). Assembly of microarrays for genome-wide measurement of DNA copy number. *Nature Genetics* **29**, 263–264.

Solinas-Toldo, S., S. Lampel, S. Stilgenbauer, J. Nickolenko, A. Benner, H. Dohner, T. Cremer, and P. Lichter (1997). Matrix-based comparative genomic hybridization: Biochips to screen for genomic imbalances. *Genes, Chromosomes and Cancer* **20**, 399–407.

Titterington, D.M., A.F.M. Smith, and U.E. Makov (1985). *Statistical analysis of finite mixture distributions*. Wiley.

van Ommen, GJ. B. (2004). Frequency of new copy number variation in humans. http://www.nature.com/naturegenetics.

Venter, J.H. and S.J. Steel (1996). Finding multiple abrupt change-points. *Computational Statistic and data analysis* **22**, 481–504.

Wang, P., Y. Kim, J. Pollack, B. Narasimhan, and R. Tibshirani (2005). A method for calling gains and losses in array CGH data. *Biostatistics 6*(1), 45–58.

Ward, J.H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association* **58**, 234–244.

Wei, G.C.G. and M.A. Tanner (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithm. *JASA* **82**, 528–550.

Woodbury, M.A. (1971). Discussion of paper by Hartley and Hocking. *Biometrics* **27**, 808–817.

Worsley, K.J. (1986). Confidence regions and tests for a change-point in a sequence of exponential family random variables. *Biometrika 73*(1), 91–104.

Wu, C.F.J. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics 11*(1), 95–103.

Yang, Y.H., S. Dudoit, P. Luu, and T. Speed (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acid Research 30*(4), e15.

Yang, Y.H. and T. Speed (2002). Design issues for cDNA microarray experiments. *Nature reviews* **3**, 579–588.

Yao, Y.C. and S.T. Au (1989). Least square estimation of a step function. *Sankhya* **3**, 370–381.

Yunis, JJ. (1976). High resolution of human chromosomes. *Science* **191**, 1268–1270.

Résumé :

Dans cette thèse nous proposons un nouveau modèle statistique pour l'analyse des problèmes de segmentation/classification dont l'objectif est de partitionner des données en zones homogènes, et de regrouper ces zones en un nombre fini de classes. Les problèmes de segmentation/classification sont traditionnellement étudiés à l'aide des modèles de chaînes de Markov cachées. Nous proposons un modèle alternatif qui combine un modèle de segmentation et un modèle de mélange.

Nous construisons notre modèle dans le cas gaussien et nous proposons une généralisation à des variables discrètes dépendantes. Les paramètres de ce modèle sont estimés par maximum de vraisemblance à l'aide d'un algorithme hybride fondé sur la programmation dynamique et sur l'algorithme EM. Nous abordons un nouveau problème de sélection de modèle qui est la sélection simultanée du nombre de groupes et du nombre de segments et proposons une heuristique pour ce choix.

Notre modèle est appliqué à l'analyse de données issues d'une nouvelle technologie, les microarrays CGH (Comparative Genomic Hybridization). Cette technique permet de compter le nombre de milliers de gènes le long du génome en une seule expérience. L'application de notre méthode à ces données permet de localiser des zones délétées ou amplifiées le long des chromosomes. Nous proposons également une application à l'analyse des séquences d'ADN pour l'identification de régions homogènes en terme de composition en nucléotides.

Mots clés: DÉTECTION DE RUPTURES – MODÈLES DE MÉLANGE – SÉLECTION DE MODÈLES – PROGRAMMATION DYNAMIQUE – ALGORITHME EM – MICROARRAY CGH – SÉQUENCES D'ADN

Classification AMS: 62P10, 62O7, 6299, 62H30.

Abstract :

This thesis is devoted to the development of a new statistical model for segmentation/clustering problems. The objective is to partition the data into homogeneous regions and to cluster these regions into a finite number of groups. Segmentation/clustering problems are traditionally studied with hidden Markov models. We propose an alternative model which combines segmentation models and mixture models.

We construct our model in the Gaussian case and we propose a generalization to discrete dependent variables. The parameters of the model are estimated by maximum likelihood with a hybrid algorithm based on dynamic programming and on the EM algorithm. We study a new model selection problem which is the simultaneous selection of the number of clusters and of the number of segments. We propose a heuristic for this choice.

Our model is applied to the analysis of CGH microarray data (Comparative Genomic Hybridization). This technique is used to measure the number of thousands of genes on the genome in one experiment. Our method allows us to localize deleted or amplified regions along chromosomes. We also propose an application to the analysis of DNA sequences for the identification of homogeneous regions in terms of nucleotide composition.

Keywords: MULTIPLE CHANGE-POINTS – MIXTURE MODELS – MODEL SELECTION – DYNAMIC PROGRAMMING – EM ALGORITHM – MICROARRAY CGH – DNA SEQUENCES

AMS Classification: 62P10, 62O7, 6299, 62H30.