# Multiscale Copy Number Alteration Analysis Using Wavelets


Maharani Ahsani Ummi

Department of Statistics

University of Leeds

Submitted in accordance with the requirements for the degree of

*Doctor of Philosophy*

15th September 2023

*To my mother, father, and husband for their endless and unconditional support...*

# Acknowledgements

I would like to express my deep gratitude to my supervisors, Dr. Arief Gusnanto and Dr. Stuart Barber who have been excellent mentors. It would have been impossible to finish this project without their constant advice, patience and encouragement. They always helped me not to give up my research with kind advice and supports. I would like to thank them for giving me the amazing opportunity to do research and study at the University of Leeds. I am also thankful to the School of Mathematics PhD Scholarship at the University of Leeds for sponsoring my living costs and tuition fees.

Finally, I am profoundly grateful to my husband, Fathiro, whose unconditional love, support and courage helped me withstand hardship and make the best decisions in my life. Also, I am more than grateful to my parents who encourage me in my daily life as well as all my lovely friends who help me and support me these years.

# Abstract

The need for multiscale modelling comes from the fact that it is rare for measured data to contain contributions at a single scale. For example, a typical signal from an experimental process may contain contributions from a variety of sources, such as noise and faults. These features usually occur with different localisation and at different locations in time and frequency. It is also inevitable for copy number DNA sequencing. Identifying Copy Number Alteration (CNA) from a sample cell faces difficulties due to errors, different sizes of reads being recorded, infiltration from normal cells, and different sizes of test and normal genomes. Thus, the representation of the measurements in terms of multiscale offers efficient feature extraction or noise removal from a typical process signal.

One of the powerful tools used to extract the multiscale characteristics of the observed data is wavelets. Wavelets are mathematical expansions that are able to transform data from the time domain into different layers of frequency levels. In this thesis, wavelets are used, first, to segment the CNA data into regions of equal copy number and secondly, to extract useful information from the original data for a better prediction of tumour subtypes. For the first purpose, an approach called TGUHm method is presented which applies the tail-greedy unbalanced Haar (TGUH) wavelet transform (Fryzlewicz, 2018) to perform segmentation of CNA data. The 'unbalanced' characteristic of the TGUH approach gives the advantage that the data length does not have to be a power of two as in the traditional discrete Haar wavelet method. An additional benefit is it can address the problem

that commonly arises in Haar wavelet estimation where the estimator is more likely to detect jumps at dyadic locations which might not be the actual locations of the jumps/drops in the true underlying CNA pattern.

For the next step, the TGUHm method is applied to the existing data-driven wavelet-Fisz methodology to deal with the heteroscedastic noise problem that we often find in CNA data. In practice, real CNA data deviate from homoscedastic noise assumption and indicate some dependencies of the variance on the mean value. The proposed method performs variance stabilisation to bring the problem into a homoscedastic model before applying a denoising procedure. The use of the unbalanced Haar wavelet also makes it possible to estimate short segments better than the balanced Haar wavelet-based segmentation methods. Moreover, our simulation study indicates that the proposed methodology has substantial advantages in estimating both short and long-altered segments in copy number data with heteroscedastic error variance.

For the second purpose, a wavelet-based classification framework was proposed which employs non-decimated Haar wavelet transform to extract localised differences and means of the original data into several scales. The wavelet transformation decomposes the original data into detail (localised difference) and scaling (localised means) coefficients into different resolution levels. This would bring an advantage to discover hidden features or information which are difficult to find from original data only. Each resolution level corresponds to a different length of wavelet basis and by considering which levels are most useful in a model, the length of the region that is responsible for the prediction could be identified.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation and Background

Changes in genomic copy number are commonly found in cancer patients and become hallmarks of its progression (Hanahan & Weinberg, 2011). These changes are called copy number alteration (CNA) and are created as a result of genomic events that cause discrete gains or losses of copy number at some genomic regions. Identifying the type (gains/losses) and location of the alterations in the genome becomes an important step toward improved diagnosis and treatment of cancer (Dancey *et al.*, 2012).

Sequencing technologies such as next-generation sequencing (NGS) enable us to obtain copy number ratio data between tumour and normal cells along the genomes and make the estimation of CNA possible (Wagle *et al.*, 2012). But these copy number data can be very noisy due to biological variation and random experimental error. An important step called segmentation is needed here to deal with the noise in the analysis of CNA data. The goal of this segmentation step is to remove the random error and recover the unknown function that represents the true CNA pattern from the observed measurements, so that subsequent downstream analyses can be carried out based on this information. Furthermore, this underlying pattern can be considered to be a piecewise constant function and, therefore, it can be said that the identification of the CNA pattern is, in principle, a change-point detection and estimation problem.

Many change-point detection methods have been developed to locate change-points in copy number data where the noise level is constant across the genome. These includes, circular binary segmentation (CBS), proposed by Olshen *et al.* (2004), which is a copy number segmentation method developed based on a statistical test to detect significant breakpoints in the data. An approach based on a least-squares segmentation algorithm named CopyNumber method was introduced by Nilsen *et al.* (2012) to perform copy number segmentation by combining least squares principles and a penalization scheme. Some of wavelet-based change-point detection methods have also been developed such as HaarSeg (Ben-Yaacov & Eldar (2008)) which applies the non-decimated discrete wavelet transform (NDWT) and wavelet thresholding to CNA data. In a more recent approach, Li *et al.* (2016) proposed FDRSeg method, a segmentation method which controls the false discovery rate of the whole segmentation.

To deal with this problem, in this thesis, a multiscale analysis using wavelets was conducted. The need for multiscale modeling comes from the necessity to represent the observed data into some different layer of frequency level since it is rare for measured data to contain contributions only from a single scale. For example, a typical signal from an experimental process may contain contributions from a variety of sources, such as sensor noise, disturbances, and faults. These features usually occur with different localizations and at different locations in time and frequency.

One of the powerful methods used to extract the multiscale characteristics of the observed data is wavelet method. The key information that wavelets extract is the 'detail' in the observed data at different scales and different locations. This representation provides tools to estimate the true signal hidden in the data. This reason motivates to conduct research concerning the investigation of multiscale models of copy number data using wavelets.

## 1.2 Objectives

There are many problems related to copy number alteration analysis. The focus of this thesis is outlined and summarised into three main problems as follows.

1. Identify the most suitable wavelets for CNA data analysis. When choosing a suitable wavelet, it is essential to understand the characteristics of the DNA copy number profile and the basic properties of wavelets. The DNA copy number profile of a tumour is a piecewise constant that reflects the relative abundance of chromosomal segments. Due to this, it is natural to employ the Haar wavelets which belong to a family of square-shaped functions that are able to produce the piecewise constant approximation. But recently, there are several choices of Haar wavelet-based methods that can be considered. Thus, an investigation to identify the most appropriate Haar wavelet-based method for CNA data analysis is needed.

2. Develop a Haar wavelet-based segmentation method. One of the main challenges in the analysis of CNA data is segmentation. Knowing the exact location of abrupt changes in DNA profiles serves several biological needs such as identifying possibly damaged genes involved in a particular type of cancer. Many segmentation methods have been developed to produce a clear piecewise approximation of CNA profiles, but most of them are only sensitive to long altered segments. Meanwhile, in the context of low-coverage, short segments also potentially bring key oncogenes or tumour-suppressor genes of interest. Therefore, a method that performs well in estimating both long and short segments is generally needed.

3. Prediction of tumour subtypes using wavelets. Lung cancer is one of the major causes of cancer mortality in the world (Siegel *et al.*, 2012). The most common lung cancer that contributes to this is non–small cell lung cancer (NSCLC) which can be further divided into lung adenocarcinoma (LA) and lung squamous cell carcinoma (LS). These two subtypes are often classified together as NSCLC even though they have different biological signatures (Herbst *et al.*, 2008). Hence, it is essential to investigate statistical models to distinguish these two subgroups clinically. By utilizing the wavelet transform, the aim here is to decompose the CNA segmented data into several scales and investigate the hidden information which is not easy to identify only by the original data.

## 1.3 Outline of Thesis

This thesis is structured as follows. Chapter 2 begins with an introduction to CNA and wavelet analysis. Descriptions of the CNA data and the CNA identification as a change point detection problem are provided. The wavelet analysis, including a brief theoretical review of wavelets and the wavelet denoising principle, is also described in Chapter 2. A simple wavelet denoising example using four types of wavelets is presented, which illustrates why the Haar wavelet is used as the main wavelet to perform multiscale analysis of CNA data.

In Chapter 3, a comparison study of three kinds of Haar wavelet-based segmentation methods is presented. Three wavelet methods considered in this chapter are the basic Haar wavelet, HaarSeg, and tail-greedy unbalanced Haar methods. The first two methods utilise the standard 'balanced' Haar wavelet, while the third one uses the 'unbalanced' Haar wavelet. The flexibility of the unbalanced Haar wavelet to adjust its breakpoint to follow the pattern of observed data makes the TGUH method able to provide more clean segmentation results compared to the balanced Haar wavelet-based methods. A comparative simulation study presented in Chapter 3 also suggests that the TGUH method is the most preferable.

Even though the TGUH method has a strong ability to provide clean segmentation compared to the 'balanced' Haar wavelet-based methods, its copy number segmentation results for NGS data have a tendency to estimate spikes (very short altered segments of only one or two data points). This is due to the characteristic of NGS data that it often contains many extreme observations (outliers). The wavelet thresholding used in the standard TGUH method is unable to threshold/remove the detail coefficients corresponding to these outliers as they are likely translated into large coarse-scale coefficients by the TGUH transform. This causes the final estimator to contain spurious change points as spikes (very short altered segments of only one or two data points). To address this problem, in Chapter 4, an extended TGUH method named TGUHm method is introduced. In our TGUHm method, an additional procedure called unconnected thresholding is added to the connected thresholding used in TGUH (Fryzlewicz, 2018) for pruning the spikes and controlling the minimum altered segment size.

Another characteristic of CNA data from NGS technology is that the variance exhibits some association with the mean. In practice, it often observed that in the NGS data, the higher the copy number ratio, the higher the random variation. This brings a disadvantage to the TGUHm method, as it is designed to deal with the homoscedastic noise problem. The TGUHm method would produce many spurious change points in a high copy number ratio region. To address this problem, Chapter 5 presents a new wavelet approach named data-driven TGUH-Fisz (DDTF) that extends the data-driven wavelet-Fisz methodology (Fryzlewicz, 2008) to TGUHm wavelets denoising for handling non-negative data with heteroscedastic noise whose variance is a non-decreasing function of the mean. This method performs variance stabilisation before the denoising/thresholding step so that it allows us to translate the signal into a set of unbalanced Haar wavelet coefficients that are approximately Gaussian, and then the standard wavelet denoising/thresholding technique can be applied to those coefficients.

In Chapters 4 and 5, two unbalanced wavelet-based segmentation methods have been introduced, the TGUHm and DDTF methods. Those methods can be used to separate noise from the CNA data, resulting in chromosomes splitting into regions of equal copy number. The resulting CNA estimates can then be processed into a cancer subtype classification procedure. Given those segmented lines, Chapter 6 utilises the wavelet transform to gain a more detailed summary for each location in the genome as part of different scales in viewing the data. The use of the segmented line itself is already known to be useful for investigating the important gains and losses along the genome that contribute to cancer subtype classification, but the wavelet-transformed data is expected to be more informative due to its ability to decompose data into several scales. Furthermore, determining which resolution scales are the most informative can open up the opportunity for improved interpretation. For the analysis in Chapter 6, the DDTF method is considered to perform the segmentation, and the non-decimated Haar wavelet transform is used to extract the localised information of those segmented CNA data.

The final summary and conclusions of the thesis are given in Chapter 7. The final conclusion is included with some suggestions for further work. The thesis is concluded with an Appendix and References.

# Chapter 2

# Introduction to Copy Number Alteration and Wavelet Analysis

## 2.1 Introduction

Copy Number Alterations (CNAs) play a crucial role in various biological processes and are frequently associated with complex diseases, including cancer. Detecting CNAs and understanding their locations within the genome are essential for uncovering underlying genetic mechanisms and potential therapeutic targets.

Change point analysis, on the other hand, is a statistical technique used to identify points in a dataset where a significant shift or change occurs. In the context of genomics, change point analysis is employed to pinpoint regions within a genome that exhibit abrupt changes in characteristics such as gene expression, copy number, or other biological properties. Since CNA can be defined as a genetic variation or anomaly that involves changes in the number of copies of a particular segment of DNA within an individual's genome, it is natural to consider the identification of CNA from the noisy raw CNA data as the change-point detection problem.

The relationship between CNAs and wavelet analysis lies in the application of wavelet methods to detect change points in genomic data affected by CNAs. Wavelet analysis is a mathematical tool that allows the analysis of signals at multiple scales and resolutions. Particularly, by performing suitable wavelet denoising, it has been proven to be effective in capturing abrupt changes in

complex and noisy datasets (Ben-Yaacov & Eldar, 2008; Hsu *et al.*, 2005), making it particularly suited for detecting the CNAs.

This chapter delves into two significant subjects: copy number alteration (CNA) and wavelet analysis. This chapter starts by briefly reviewing CNA in Section 2.2, the subsequent explanation of the CNA detection as the change-points analysis can be found in Section 2.3. The discussion also extends to the dataset utilized for CNAs in this thesis, detailed in Section 2.4. Following this, the introduction of wavelet is outlined in Section 2.5, which encompasses both theoretical understanding and the application of wavelet denoising.

## 2.2 Copy Number Alteration

Copy number alterations refer to gains (duplications) and losses (deletions) of large segments, with a size from a few kilobases up to whole chromosomes (Wu *et al.*, 2014). A kilobase (kb) refers to a unit of measurement in molecular biology equal to a thousand base pairs of DNA or RNA where a base pair (bp) is a fundamental unit of double-stranded nucleic acids. In a normal human body, each cell has two copies of every chromosome except sex chromosomes (chromosomes X and Y) which vary between males and females. Copy number alterations mark the change in the number of copies of genomic DNA from the normal two copies. For example, if some genomic regions exhibit gains (duplications) of genetic code, this means that the copies of chromosome (copy number) in those regions are larger than two. On the other hand, if some genomic regions exhibit losses (deletions), this means that the copy number related to those regions is less than two.

CNAs are commonly found in cancer patients and become hallmarks of their progression (Hanahan & Weinberg, 2011). Many cancers, such as breast cancer, lung cancer, and prostate cancer, are a consequence of CNA. Even though the specifics of genome alteration may vary drastically among different tumour types, Hanahan & Weinberg (2011) have shown that instability of the genome results in the vast majority of human cancer cells. Furthermore, previous CNA studies have shown that distinct patterns of DNA copy number alteration are associated with different cancer subtypes (Bergamaschi *et al.*, 2006; Pei *et al.*, 2001; Sy *et al.*, 2004; Yakut *et al.*, 2006) so that identifying the accurate locations of gains and

losses of DNA copy number is very important for prediction of cancer subtype and knowing the correct subtype is critical for considering the right treatment for the patient (Dancey *et al.*, 2012).

### 2.2.1 Next-Generation Sequencing for CNA

Several technologies are available to identify CNAs. One of the most recent technologies is called Next-Generation Sequencing (NGS). The NGS platforms such as Illumina, Roche 454, and ION Torrent PGM, allow to sequence a large number of DNA fragments at a reasonable cost and speed. Another advantage of using sequencing instead of array technology is that it avoids the typical saturation or background noise commonly observed in hybridization techniques such as array technologies (Gusnanto *et al.*, 2012).

NGS machines extract a large number of short DNA fragments (reads) from a biological sample. The copy number of any genomic region can be estimated by counting the number of reads aligned to a particular region (Magi *et al.* (2011)). This procedure is done for a set of non-overlapping fixed-width genomic regions (windows). The data-based optimal window size can be obtained based on Akaike's information criterion and cross-validated log-likelihood (Gusnanto *et al.* (2014)).

The selection of optimal window size is a crucial step to make sure the patterns of genomic features are informative enough for further analysis. If a much smaller window size is used, this will cause many genomic regions with zero read count and make the overall analysis non-informative. On the other hand, using a much bigger window size will 'smooth out' some patterns of alteration. Then, by comparing the number of reads in each window and chromosome between cancer and normal cells, the copy number ratio can be estimated. The schematic representation of the NGS copy number data is shown in Figure 2.1.

## 2.3 Detecting CNA as a Change-Point Problem

Let $u_i$ be the observed number of reads from a tumour sample or genome at $i$-th window, where $i = 1, \ldots, n$ and $n$ is the total number of windows in a genome.

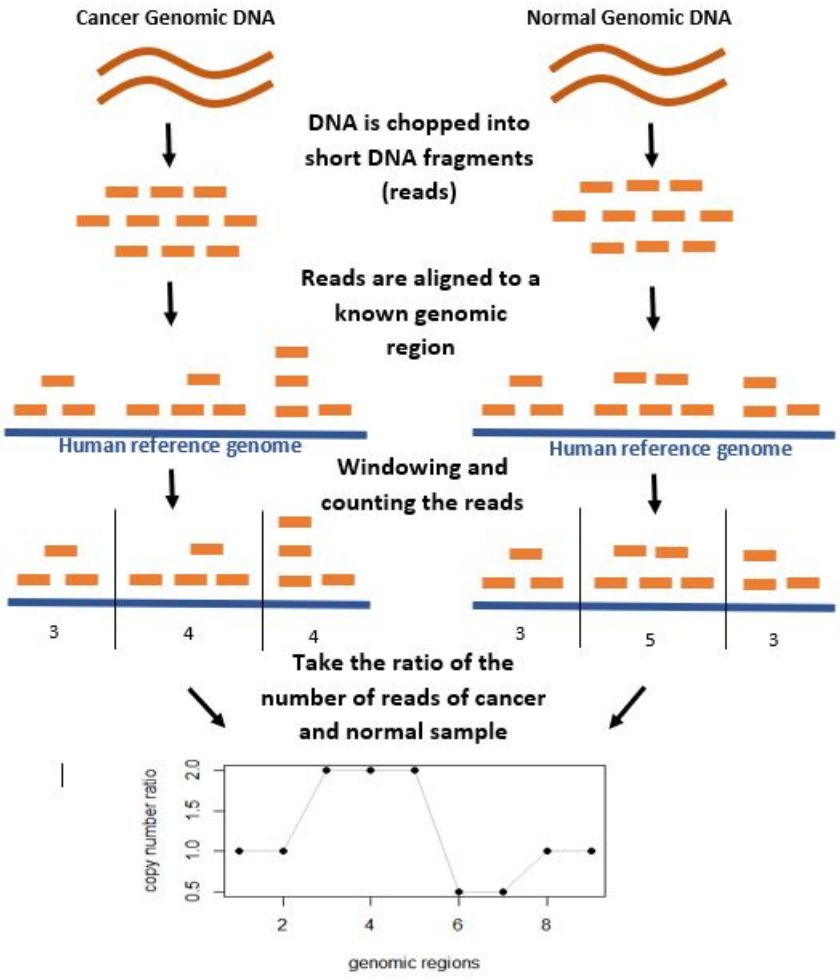


Figure 2.1: Schematic representation of the extraction of NGS DNA copy number data.

Let $v_i$ be the observed number of reads in a normal sample at $i$-th window. To identify the CNA in window $i$, the ratio between the tumour and the normal genomes in each window $i$ can be taken as,

$$r_i = \frac{u_i}{v_i}. \tag{2.1}$$

Ideally, as shown in Figure 2.1, the ratio in Equation 2.1 can then be plotted against the windows $i$ to identify regions of duplication or deletion.

In a normal human body, the number of chromosome copies in a cell or the ploidy is two. If there exists CNA within a cell, the ploidy can take any value

within a set $\{0, 1, 2, 3, \ldots\}$. Hence, ideally, the copy number ratios $r_i$ take any value in $\{0, 0.5, 1, 1.5, 2, \ldots\}$. The copy number alterations are indicated if the ratio of the experimental and normal samples $r_i$ deviates from one.

In real practice, the value of $r_i$ may deviate from its ideal values due to several factors such as experimental error and the different sizes of the tumour and normal cells. This measurement error can be modelled by relating the observed copy number changes $r_i$ to the true signal $f_i$ as follows,

$$r_i = f_i + \epsilon_i, \tag{2.2}$$

where $\epsilon_i$ is the noise contained in the measured data. Our aim is to estimate $f_i$ using the observations $r_i$.

Since the ideal values for the ratio $r_i$ lie in $\{0, 0.5, 1, 1.5, 2, \ldots\}$, the function $f_i$ in equation (2.2) can be considered as a one-dimensional, piecewise-constant signal with change-points whose number $N$ and locations $\nu_1, ..., \nu_N$ are unknown. Therefore, the identification of the CNA pattern, in principle, can be mentioned as a change-point detection and estimation problem.

## 2.4 Description of Non-Small Cell Lung Cancer Dataset

Throughout this thesis, to motivate and test the proposed methods, the copy number dataset used is from 76 patients with Non-Small Cell Lung Cancer (NSCLC) (Belvedere *et al.*, 2012). The dataset consists of two groups: squamous carcinoma (38 patients) and adenocarcinoma (38 patients). The detailed description of sample preparation, DNA extraction, and library preparation is described in Wood *et al.* (2010). DNA sequences were aligned to the human genome (USCS hg19) using the Burrows–Wheeler Alignment (BWA) tool suite version 0.5.9-r16 Li & Durbin (2009).

Only short sequences ('reads') with the BWA mapping score greater than 37 are used. Using 'depth of coverage' the reads are counted and mapped to fixed non-overlapping genomic regions ('windows'), estimated to be 150 kb (Gusnanto *et al.*, 2014). The total number of windows along the genome is 20,652. To avoid

problems occurring in further analysis, the sex chromosomes, the mitochondria chromosome and the centromere regions as missing data are removed. At the end of this removal procedure, the number of genomic windows becomes 17,931.

For easier comparison between CNA profiles, a normalisation using the CNAnorm package (Gusnanto *et al.*, 2012) is performed. Normalisation is a crucial step in CNA analysis to correct the variations caused by factors other than the copy number. For example, correcting for GC-content and tumour sample contamination. Both corrections mentioned are needed because the GC-contents can affect the staining intensity and subsequent analysis and the contamination of normal cells when the sample is taken from the cancerous tumour or the tumour sample contamination potentially leading to inaccurate conclusions.

As described in the previous section, normal genomic regions in the tumour cell, $r_i = 1$. Thus the CNAnorm proposed by Gusnanto *et al.* (2012) will shrink the copy number ratio $r_i$ towards ratio 1 when there exists contamination of normal cells with the tumour cells. It also aligns the CNA data so that the most common genomic regions are centred at ratio 1. Figure 2.2 shows an example of the CNA ratio of each lung cancer subtype, squamous carcinoma and adenocarcinoma, after the data preparation procedures mentioned above.

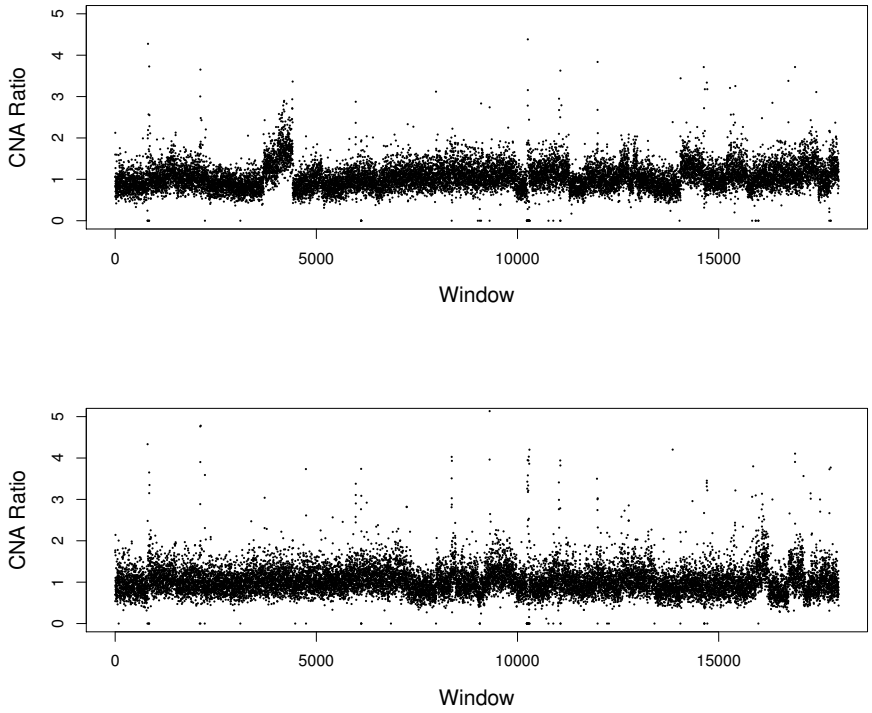


Figure 2.2: The CNA ratio for each window along the genome for a patient with squamous carcinoma (top) and adenocarcinoma (bottom) type lung cancer after removal and normalisation procedure. The y-axis denotes the CNA ratio and x-axis denotes the indication of window or $i$ in equation 2.2. The window size used is 150 kb, which means that for window equal to 1 denote the CNA ratio between 1–150.000 bp, window equal 2 between 150.001–300.000 bp of genome, and so on.

## 2.5 Wavelet Analysis

In this section, some mathematical background and terminology which is required to understand the wavelet application on the CNA data are explained. For more rigorous and general mathematical coverage of wavelets may be consulted to Daubechies (1992), Härdle *et al.* (2012) or Vidakovic (2009).

Since the discovery of wavelets in the early 1980s, wavelets have gained huge attention both from the mathematical and applied sciences points of view. The term wavelets is commonly referred to as a set of basis functions with special characteristics, the oscillation and the compact support. The oscillation refers to the wavelet 'goes up and down' or mathematically can be expressed by the condition that $\int_{-\infty}^{\infty} \psi(x)dx = 0$ where $\psi$ is wavelet function or mother wavelet. The compact support characteristic does not mean that all the wavelets have compact support but they must decay to zero rapidly.

Given the prevalence of wavelets and wavelet-like quantities across various disciplines, describing them becomes challenging due to the multitude of starting points. Therefore, in this section, a popular starting point of wavelet: the Haar wavelet is used to begin with.

### 2.5.1 Continuous Haar Wavelets

In the early 1980s, Morlet *et al.* (1982) introduced key theoretical results and laid the groundwork for continuous wavelet decompositions of $\mathbb{L}_2$ functions. Once one has a mother wavelet $\psi(x) \in \mathbb{L}_2$, a family of functions $\psi_{a,b}$, where $a \in \mathbb{R} \setminus \{0\}$ and $b \in \mathbb{R}$, can be defined as translations and re-scales (or dilations) of the function $\psi$:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right). \tag{2.3}$$

Here, the parameters $a$, and $b$ are called dilation and translation parameters, respectively. These parameters vary continuously over $\mathbb{R} \times \mathbb{R}$.

The mother wavelet $\psi$ is assumed to satisfy the admissibility condition,

$$C_\psi = \int_{\mathbb{R}} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty, \tag{2.4}$$

where $\Psi(\omega)$ is the Fourier transformation of $\psi(x)$. The admissibility condition (2.4), implies $0 = \Psi(0) = \int \psi(x)dx$. This property of the function $\psi$ motivates the name wavelet.

For any $\mathbb{L}_2$ function $f(x)$, the continuous wavelet transform (CWT) is defined as,

$$CWT_f(a,b) = \langle f, \psi_{a,b} \rangle = \int f(x)\overline{\psi_{a,b}(x)}dx. \tag{2.5}$$

In its essence, the Haar mother wavelet is expressed as follows:

$$\psi_{a,b}^{Haar} = \sqrt{a}\left[\mathbb{1}\left(b \leq x \leq \frac{a}{2} + b\right) - \mathbb{1}\left(\frac{a}{2} + b \leq x \leq a + b\right)\right], \tag{2.6}$$

where $a \in \mathbb{R}^+$, $b \in \mathbb{R}$, and $\mathbb{1}$ is an indicator function. Then the continuous Haar wavelet transform can be formulated as:

$$CWT_f(a,b) = \langle f, \psi_{a,b}^{Haar} \rangle = 2\sqrt{a} \times \left[F\left(\frac{a}{2} + b\right) - \frac{F(b) + F(a+b)}{2}\right]. \tag{2.7}$$

### 2.5.2 Discrete Haar Wavelets

Discrete wavelet transformations (DWT) are applied to discrete data sets and produce discrete outputs. From equation (2.5), the continuous wavelet transform is a convolution of the data sequence with a scaled, $a$, and translated, $b$, version of the mother wavelet. This means that the wavelet transform is calculated by continuously shifting a scalable function over a signal and calculating the correlation between the two. As a result, for practical application, the continuous wavelet transform will generate an infinite number of wavelets.

To simplify the continuous wavelet transform, it is possible to choose discrete values for the scaling parameter $a$ and translation parameter $b$ while ensuring that the transformation remains invertible. This means that the original function can be uniquely recovered by applying the inverse transformation, even with the discrete selections of $a$ and $b$. For some real parameters $a > 1$ and $b > 0$, one can discretise them by setting $a = 2^{-j}$ and $b = 2^{-j}k$, where $j, k \in \mathbb{Z}$. Therefore, one once has wavelet function $\psi$, one can generate the function

$$\psi_{j,k} = 2^{j/2}\psi(2^j x - k) \tag{2.8}$$

for integers $j$ and $k$ which can provide an orthogonal basis for suitable choices of $\psi$. It turns out that such wavelets can form an orthonormal set (containing orthogonal and unit length vectors). Moreover, such a set of wavelets can form bases for various spaces of functions. Hence, any function $f(x)$ can be decomposed into the following series,

$$f(x) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{j,k}\psi_{j,k}(x), \tag{2.9}$$

where, due to the orthogonality of the wavelets, the coefficients of the expansion can be found by,

$$d_{j,k} = \int_{-\infty}^{\infty} f(x)\psi_{j,k}(x)dx = \langle f, \psi_{j,k} \rangle, \tag{2.10}$$

for integers $j$ and $k$. This means that any signal can be reconstructed by taking a sum of the weighted orthogonal wavelet basis functions.

The simplest wavelet, Haar wavelet (or Haar mother wavelet), is a wavelet basis function which mathematically can be defined by

$$\psi(x) = \begin{cases} 1 & \text{for } x \in [0, 1/2), \\ -1 & \text{for } x \in [1/2, 1), \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

But this function is not enough to cover the whole real line. The addition function called father wavelet is needed to account for this problem. The Haar father (or scaling) wavelet function is defined by

$$\phi(x) = \begin{cases} 1 & \text{for } x \in [0, 1), \\ 0 & \text{otherwise.} \end{cases} \tag{2.12}$$

With the addition of the father wavelet, the constant function on $[0, 1)$ for example, or any constant function, can be represented easily as a multiple of the father wavelet. Similar to the function in (2.13), the translation of the father wavelet can be defined as the following function,

$$\phi_{j_0,k} = 2^{j_0/2}\phi(2^{j_0}x - k). \tag{2.13}$$

By using both mother and father wavelets, any $f(x)$ can be written in terms of integer translates of the father wavelets, $\phi_{j_0,k}(x)$, which represent the 'average'

or 'overall' level of function (large-scale behaviour), and of mother wavelets, $\psi_{j,k}(x)$, which represent discontinuities or sharp features (small-scale behaviour) accumulating information at a set of scales $j$ ranging from $j_0$ to infinity,

$$f(x) = \sum_{k=\mathbb{Z}} c_{j_0,k}\phi_{j_0,k}(x) + \sum_{j=j_0}^{\infty}\sum_{k=\mathbb{Z}} d_{j,k}\psi_{j,k}(x). \tag{2.14}$$

The numbers $\{c_{j_0,k}\}$ and $\{d_{j,k}\}$ are called the smooth (scaling) and detail (wavelet) coefficients of function $f$, respectively.

### 2.5.3 Multiresolution Analysis

The multiresolution analysis framework is often used to define DWT. A multiscale analysis is started with a scaling function $\phi$ and the nested sequence of close subspaces $V_n$, $n \in \mathbb{Z}$ in $L_2(\mathbb{R})$ that form a ladder:

$$\ldots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots \subset L_2(\mathbb{R}) \tag{2.15}$$

such that

1. $\{\phi(x-k)\}_k$ is an orthonormal basis for $V_0$
2. $f(x) \in V_j \iff f(2x) \in V_{j+1}$
3. $\bigcap_j V_j = \{0\}$, $\overline{\bigcup_j V_j} = L_2(\mathbb{R})$.

The second condition above is called interscale linkage. This condition means that if $f(x)$ is a member of $V_j$, then $f(2x)$ should belong to $V_{j+1}$. Moreover, if a function $f(x)$ is shifted along the line, for example, by an integral amount $k$, to form $f(x-k)$, then the function $f(x)$'s level of resolution does not change, or if $f(x)$ is a member of $V_0$, then so is $f(x-k)$.

Due to the conditions above, the scaling function $\phi(x)$ is an element of $V_0$ and $\{\phi(x-k)\}_k$ is an orthonormal basis for $V_0$. Also, due to the interscale linkage, the function $\phi(x) \in V_0$ can be represented as the linear combination of functions from $V_1$,

$$\phi(x) = \sqrt{2}\sum_k h_k\phi(2x-k), \tag{2.16}$$

where $\{h_k\}_k \in l_2$ and $x \in R$. Once one has the scaling function $\phi$, one can use it to define the wavelet function $\psi$.

Now, let define the wavelet function $\psi$ such the way that $\{\psi(x-k\}_k$ is an orthonormal basis for the space $W_0$ where $W_0$ is defined as the orthogonal complement of $V_0$ in $V_1$ ($V_1 = V_0 \oplus W_0$). By defining

$$W_j = \left\{ f \in L_2(R) \quad | \quad f(x) = 2^{j/2} \sum_k d_k \psi(2^j x - k) \right\},$$

$W_j$ can considered as the orthogonal complement of $V_j$ in $V_{j+1}$. Then,

$$V_{j+1} = V_j \oplus W_j = \ldots = V_0 \oplus \left( \bigoplus_{i=0}^{j} W_i \right).$$

Recall that $\bigcap_j V_j = \{0\}$ and $\overline{\bigcup_j V_j} = L_2(\mathbb{R})$, this implies that

$$L_2(\mathbb{R}) = V_0 \oplus \left( \bigoplus_{i=0}^{\infty} W_i \right) = V_{j_0} \oplus \left( \bigoplus_{i=j_0}^{\infty} W_i \right), \ \forall j_0.$$

In Daubechies (1992), precise procedures to obtain $\psi$ once $\phi$ is described (in Section 5.1). One possibility (Theorem 5.1.1 (Daubechies, 1992)) of the wavelet function is to represent it as

$$\psi(x) = \sqrt{2} \sum_k h_{1-k} (-1)^k \phi(2x - k). \tag{2.17}$$

The coefficients in (2.17) has its own notation:

$$g_k = (-1)^k h_{1-k}. \tag{2.18}$$

This coefficient is important to express how the wavelet is constructed from the finer-scale father wavelet coefficients. By using this notation, the wavelet function $\psi$ can be rewritten as

$$\psi(x) = \sqrt{2} \sum_k g_k \phi(2x - k). \tag{2.19}$$

for some $k \in \mathbb{Z}$.

### 2.5.4 Discrete Haar Wavelet Transform

To perform DWT, Mallat (1989) exploited the pyramidal structure of the multiresolution analysis to construct a DWT algorithm for discrete data. DWT produces a vector of wavelet coefficients of the input vector at dyadic scales and locations.

As the interest of this thesis is the Haar wavelet, here, the explanation is started by describing the Haar DWT algorithm. For input data with length $2^J$, $\mathbf{r} = (r_1, r_2, \ldots, r_n)$, for any integer $J \geq 0$, the Haar DWT transform works by pairing up adjacent input values and computing their difference and sum. This process is repeated iteratively, creating a multi-resolution analysis of the input data. Given an input vector $\mathbf{r}$, the Haar DWT is performed as follows:

1. Let $c_{J,i} = r_i$.
2. For each $j = J-1, J-2, \ldots, 0$, recursively form

$$c_{j,k} = c_{j+1,2k-1} + c_{j+1,2k}, \tag{2.20}$$

$$d_{j,k} = c_{j+1,2k-1} - c_{j+1,2k}, \tag{2.21}$$

   for $k = 1, 2, \ldots, 2^J$.

The $c_{j,k}$ are called the smooth (or scaling) coefficients and the $d_{j,k}$ are the detail (or wavelet) coefficients. The resulting coefficients then can be simply inverted to recover the original vector $\mathbf{r}$ by the inverse Haar DWT as follows,

1. For each $j = 0, 1, \ldots, J-1$, recursively form

$$c_{j+1,2k-1} = \frac{c_{j,k} + d_{j,k}}{2}, \tag{2.22}$$

$$c_{j+1,2k} = \frac{c_{j,k} - d_{j,k}}{2}, \tag{2.23}$$

   for $k = 1, 2, \ldots, 2^j$.

2. Set $r_i = c_{J,i}$.

The above Haar DWT algorithm is very effective and efficient but there it can not preserve the energy of the input. The energy here refers to the norm which is defined by $\|\mathbf{r}\|^2 = \sum_{i=1}^{n} r_i^2$. By the algorithm above, the norm, or energy of the

output sequence is much larger than that of the input. To address this energy problem, or to make the output energy is same as the input, a multiplier $\alpha$ is introduced to the formulae in (2.20) and (2.21) as follows

$$c_{j,k} = \alpha(c_{j+1,2k-1} + c_{j+1,2k}), \tag{2.24}$$

$$d_{j,k} = \alpha(c_{j+1,2k-1} - c_{j+1,2k}), \tag{2.25}$$

for $j = 0, 1, \ldots, J-1$ and $k = 1, 2, \ldots, 2^j$. For clarity, $J = 1$ is considered so that the procedure of the Haar DWT only needs one iteration hence the parameter $j$ can be ignored. Then, the (2.24) and (2.25) are equal to

$$c_k = \alpha(r_{2k-1} + r_{2k}) \tag{2.26}$$

$$d_k = \alpha(r_{2k-1} - r_{2k}). \tag{2.27}$$

Thus, the input $(r_{2k-1}, r_{2k})$ is transformed into the output $(d_k, c_k)$ and the (squared) norm of the output is

$$d_k^2 + c_k^2 = \alpha^2(r_{2k-1}^2 - 2r_{2k-1}r_{2k} + r_{2k}^2) + \alpha^2(r_{2k-1}^2 + 2r_{2k-1}r_{2k} + r_{2k}^2) \tag{2.28}$$

$$= 2\alpha^2(r_{2k-1}^2 + r_{2k}^2), \tag{2.29}$$

where $r_{2k-1}^2 + r_{2k}^2$ is the (squared) norm of the input. Hence, to make the norm of the output equal to the norm of the input, the multiplier $\alpha$ should be set to equal $2^{-1/2}$ ($\alpha = 2^{-1/2}$). With this normalisation, the equation (2.26) and (2.27) can be written as,

$$c_k = h_0 r_{2k-1} + h_1 r_{2k} \tag{2.30}$$

$$d_k = g_0 r_{2k-1} + g_1 r_{2k}. \tag{2.31}$$

where $h_0 = h_1 = 2^{-1/2}$, $g_0 = 2^{-1/2}$, and $g_1 = -2^{-1/2}$, or in more general form:

$$c_{j,k} = \sum_{l=-\infty}^{\infty} h_l c_{j+1,2k-l}, \tag{2.32}$$

$$d_{j,k} = \sum_{l=-\infty}^{\infty} g_l c_{j+1,2k-l}, , \tag{2.33}$$

where

$$h_l = \begin{cases} 2^{-1/2} & \text{for } l = 0, \\ 2^{-1/2} & \text{for } l = 1, \\ 0 & \text{otherwise,} \end{cases} \tag{2.34}$$

$$g_l = \begin{cases} 2^{-1/2} & \text{for } l = 0, \\ -2^{-1/2} & \text{for } l = 1, \\ 0 & \text{otherwise,} \end{cases} \tag{2.35}$$

and for $j = J$, the $c^J$ is equal to the input $r_i$ ($c_{J,i} = r_i$).

### 2.5.5 Matrix representation of discrete wavelet transform

Any discrete finite wavelet transform can be represented as a matrix. In the matrix representation, for given vector $\mathbf{r} = (r_1, \ldots, r_n)$, the DWT of $\mathbf{r}$ is

$$\mathbf{d} = \mathbf{W}\mathbf{r}, \tag{2.36}$$

where $\mathbf{d}$ is an $n \times 1$ vector comprising both discrete scaling coefficient $c_{0,k}$ and discrete wavelet coefficients $d_{j,k}$, and $\mathbf{W}$ is the $n \times n$ orthogonal matrix whose elements are defined by the wavelet basis generated by the dilation and translation of the wavelet function and father wavelet (scaling function). For example, for a vector input $\mathbf{r} = (r_1, r_2, \ldots, r_8)$, which produces the output Haar wavelet coefficient vector $\mathbf{d} = (c_{0,1}, d_{0,1}, d_{1,1}, d_{1,2}, d_{2,1}, d_{2,2}, d_{2,3}, d_{2,4})$, the following matrix multiplication matrix $\mathbf{W}$ gives the connection between $\mathbf{r}$ and the wavelet coefficients $\mathbf{d}$,

$$W = \begin{bmatrix} \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 1/2 & 1/2 & -1/2 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & -1/2 & -1/2 \\ \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 & -\sqrt{2}/4 & -\sqrt{2}/4 & -\sqrt{2}/4 & -\sqrt{2}/4 \end{bmatrix}. \tag{2.37}$$

It is easy to see that the three 'wavelet vectors' at different scales that are 'stored' within the matrix, for example, $(1/\sqrt{2}, -1/\sqrt{2})$ in rows two through five,

$(1/2, 1/2, -1/2, -1/2)$ in rows six and seven, and $(1, 1, 1, 1, -1, -1, -1, -1)/2\sqrt{2}$ in the last row.

The matrix $\mathbf{W}$ is orthogonal so that the inverse DWT (IDWT) is simply given by

$$\mathbf{r} = \mathbf{W}'\mathbf{d}, \tag{2.38}$$

where $\mathbf{W}'$ denotes the transpose of $\mathbf{W}$.

### 2.5.6 Wavelets Denoising

One of the most common statistical applications of wavelets is in nonparametric function estimation, also known as 'signal denoising'. Let $\mathbf{W}$ be the particular wavelet matrix associated with the orthonormal wavelet basis chosen. In the matrix representation, the wavelet transformed model of (2.2) can be written as

$$d^r = d + e, \tag{2.39}$$

where $d^r = \mathbf{W}r$, $d = \mathbf{W}f$, and $e = \mathbf{W}\epsilon$.

Because of the sparseness of the wavelet transformation, only a few large wavelet coefficients, $d^r$, contain information about the true signal, $f$, while small $d^r$ are related to noise. Due to this characteristic of the wavelet transform, Donoho & Johnstone (1994) proposed that the extraction of the informative wavelet coefficients can be done by thresholding, setting to zero the wavelet coefficients whose absolute value is below a certain threshold level and keeping those that are larger. Under this scheme, the thresholded wavelet coefficients can be obtained by

$$\hat{d} = \delta^H(d^r, \lambda) = \begin{cases} 0 & \text{if } \mid d^r \mid \leq \lambda \\ d^r & \text{if } \mid d^r \mid > \lambda, \end{cases} \tag{2.40}$$

and

$$\hat{d} = \delta^S(d^r, \lambda) = \begin{cases} 0 & \text{if } \mid d^r \mid \leq \lambda \\ \mid d^r \mid -\lambda & \text{if } \mid d^r \mid > \lambda. \end{cases} \tag{2.41}$$

where $\lambda$ is the chosen threshold. The thresholding technique in (2.40) and (2.41) are usually referred to as hard and soft thresholding, respectively. Roughly, hard

thresholding is a 'keep' or 'kill' rule, whereas soft thresholding is a 'shrink' or 'kill' rule. For the threshold $\lambda$, A commonly used and powerful threshold estimator called the universal threshold proposed by Donoho & Johnstone (1994) is used, which is defined by

$$\lambda = \hat{\sigma}\sqrt{2\log n}. \tag{2.42}$$

Here, $\hat{\sigma}$ is the estimate of the noise standard deviation $\sigma$ which is computed based on the median absolute deviation (MAD) of the sequence $\{|r_{i+1} - r_i|/\sqrt{2}|_{i=1}^{n-1}\}$; these values are the finest-scale balanced Haar wavelet coefficients of the sequence $r_i$.

Once one obtains the thresholded wavelet coefficients, then the IDWT can be used for reconstructing the response function. The resulting estimate can be written as

$$\hat{r} = \mathbf{W}'\hat{d}. \tag{2.43}$$

This three-step selective reconstruction estimation procedure can be summarized by

$$r \xrightarrow{DWT} d^r \xrightarrow{thresholding} \hat{d} \xrightarrow{IDWT} \hat{r}. \tag{2.44}$$

### Empirical Bayes for Wavelet Shrinkage

Besides the hard and soft thresholding, Bayesian wavelet methods have always been very popular for wavelet shrinkage. Wavelet representations are inherently sparse, and this sparsity can be considered a form of prior knowledge. Consequently, a set of wavelet coefficients will consist of a portion of specific coefficients that are exactly zero and the ones that are uncertain to us.

The standard procedure of a Bayesian wavelet shrinkage method is outlined below. Initially, a prior distribution is defined for the 'true' wavelet coefficients, $d_{j,k}$, which aims to capture the inherent sparsity present in wavelet representations. Subsequently, Bayes' theorem is employed to compute the posterior distribution of the wavelet coefficients (on $d^r_{j,k}$), taking into account a certain, usually assumed, known distribution of the noise wavelet coefficients, $\epsilon_{j,k}$. The ultimate goal is to calculate a posterior distribution for the unknown function by applying the inverse discrete wavelet transform (DWT) to the posterior distribution of the wavelet coefficients.

Empirical Bayes for wavelet estimation involves using the data itself to estimate the prior distribution of wavelet coefficients. The wavelet coefficients represent the coefficients obtained by decomposing the data using wavelet transform, and the goal is to estimate the underlying true wavelet coefficients.

The Empirical Bayes approach for wavelet estimation proceeds, first, by estimating the empirical prior distribution. Assume that the wavelet coefficients $d_{j,k}$ are sampled from a prior distribution. For clarity, now drop the $j, k$ indices as they add nothing to the current exposition. Johnstone & Silverman 2005b, 2004, 2005c suggest the prior for the wavelet coefficient, $d$, as

$$f_{prior}(d) = \omega\tau(d) + (1-\omega)\delta_0(d). \tag{2.45}$$

Under this model, each $d$ is zero with probability $(1-\omega)$, while, with probability $\omega$, $d$ is drawn from a symmetric heavy-tailed density $\tau$.

The key aspect of the empirical Bayes approach is the choice of mixing weight $\omega$. Let $g$ be the density function obtained by the convolution between the heavy-tailed density $\tau$ with the normal density $\phi$. The marginal density of the observed wavelet coefficients $d^r$ is given by

$$\omega g(d^r) + (1+\omega)\phi(d^r). \tag{2.46}$$

In this stage, the $g$, $\phi$, and $d^r$ are known, but the $\omega$ is not. Johnstone & Silverman (2004) then suggest to degine the marginal maximum likelihood estimator $\hat{\omega}$ of $\omega$ to maximize the log-likelihood

$$(\omega_j) = \sum_k \log\{\omega_j g(d^r_{j,k}) + (1-\omega_j)\phi(d^r_{j,k})\}. \tag{2.47}$$

Next, the estimated mixing weights are reintroduced into the prior model, and a Bayes procedure is employed to obtain the posterior distribution. Similarly, other parameters in the prior distribution can be estimated using a similar Maximum Marginal Likelihood (MML) approach. As for the noise variance $\sigma$, it is computed conventionally using the Median Absolute Deviation (MAD) of the finest-scale wavelet coefficients. Alternatively, if the noise is believed to be correlated across levels, the computation can be performed on each level.

### 2.5.7 The Examples of Wavelet Estimation

As an illustration of wavelet estimation, here, four types of popular wavelet basis are considered and shown in Figure 2.3 to perform wavelet denoising. Given the simulated noisy signal in the panel ($a$) of Figure 2.4, our objective is to remove the noise and get as close as possible to revealing the true structure (see panel (b) of Figure 2.4). In this example, the test function used is block function from Donoho & Johnstone (1994) as it very similar to the pattern of CNA data which can be seen as a piecewise constant function.

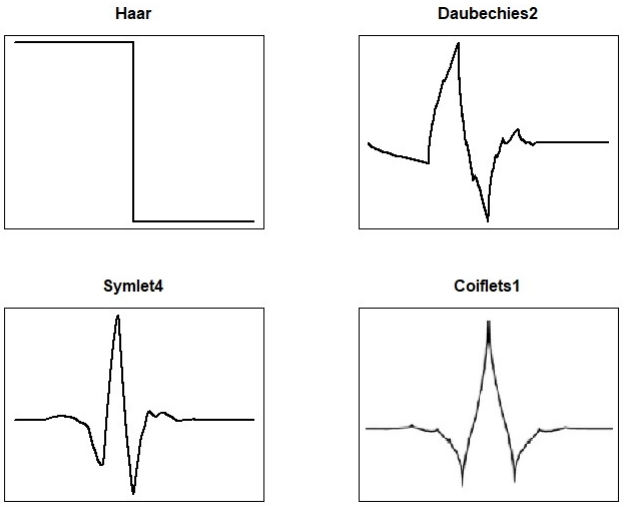


Figure 2.3: Example of few popular wavelets: Haar, Daubechies2, Symlet4, and Coiflet1. The number which follows the wavelet name represents the number of vanishing moments.

From Figure 2.4, the result of wavelet estimation using the Haar wavelet is the best in the sense that it yields piecewise constant estimates. This motivates us to employ Haar wavelet as the main basis to employ multiscale analysis on CNA data.

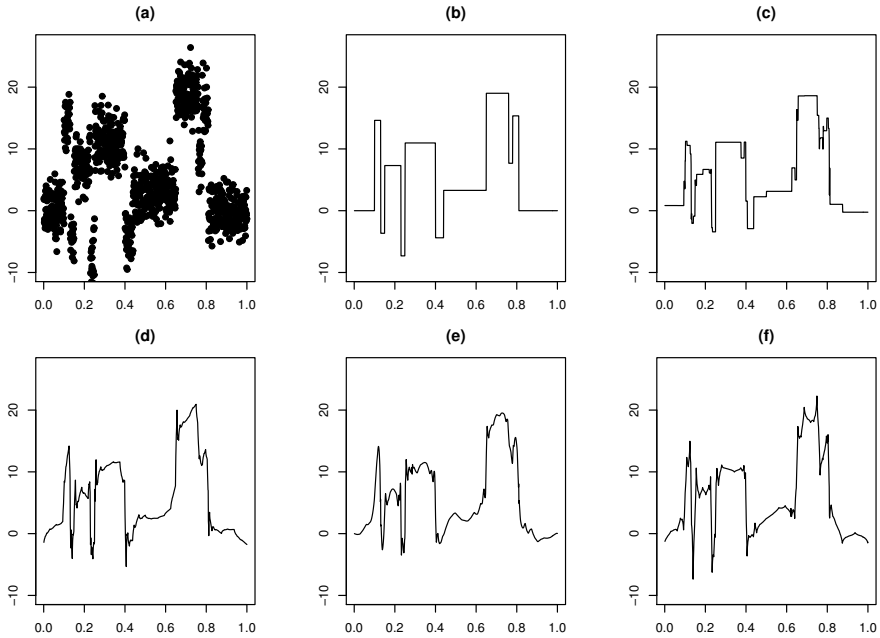


Figure 2.4: The examples of wavelet estimation. (a) Simulated noisy signal. (b) True function. (c) Reconstruction by Haar wavelet. (d) Reconstruction by Daubechies2 wavelet. (e) Reconstruction by Symlet4 wavelet. (f) Reconstruction by Coiflet1 wavelet.

# Chapter 3

# Wavelet Change Point Analysis

## 3.1 Introduction

Three Haar wavelet-based methods are compared in this chapter. The first method is the basic Haar wavelet denoising method using universal thresholding (Donoho & Johnstone, 1994) with two kinds of thresholding techniques, hard and soft thresholding. The second method is the HaarSeg method (Ben-Yaacov & Eldar, 2008), which is a copy number segmentation method based on nondecimated Haar wavelet transform. The last one is the tail-greedy unbalanced Haar (TGUH) method (Fryzlewicz, 2018) which applies the unbalanced Haar wavelet transform for signal denoising. The works in this chapter has been presented and submitted to the proceedings of Sriwijaya International Conference on Basic and Applied Sciences 2021 (Ummi *et al.*, 2023 planning to be published).

## 3.2 Non-Decimated Haar Wavelet

The decimated (basic) Haar wavelet is not translation-invariant. In other words, at any given scale, it only provides information about the input vector at certain (dyadic) locations. In contrast, the nondecimated Haar wavelet transform (NDWT) is over-complete (using more than $n$ coefficients to describe $n$ data points) and does contain a coefficient at each scale for each location. It achieves this by retaining both the odd and even decimations at each scale. This translation

invariance property makes the NDWT well suited for the task of data analysis (Starck *et al.*, 2004).

Suppose that the following CNA sequence $\mathbf{r} = (r_1, r_2, ..., r_n)$.The equations described in (2.33) and (2.32) can be rewritten more succinctly as

$$d_{j,k} = \mathcal{D}_0 \mathcal{G} r \text{ and } c_{j,k} = \mathcal{D}_0 \mathcal{H} r, \tag{3.1}$$

where

$$(\mathcal{D}_0 r)_l = r_{2l}, \tag{3.2}$$

and $\mathcal{G}$ and $\mathcal{H}$ denote the regular filtering operation, as in (2.33) and (2.32). The dyadic decimation step used in the discrete Haar wavelet transform, $\mathcal{D}_0$, essentially picked every even element from a vector $\mathbf{r}$, which only extract the information between $(r_1, r_2)$, $(r_3, r_4)$, and so on. But if the values $r_2$, $r_3$ have quite different values, we might miss something. In the NDWT transform, to obtain the wavelet coefficients not only for the even elements but also the odd ones, the father and mother wavelet coefficients for both even and odd elements at each level scale are calculated. The odd dyadic decimation operator $\mathcal{D}_1$ can be defined by

$$(\mathcal{D}_1 r)_l = r_{2l+1}. \tag{3.3}$$

More precisely, both $\mathcal{D}_0 \mathcal{G} r$ and $\mathcal{D}_1 \mathcal{G} r$ can be applied. The length of Each of these sequences is $n/2$, hence in total, the number of wavelet coefficients (both decimations) at the finest scale is $2 \times n/2 = n$. Then for the next level wavelet coefficients, both $\mathcal{D}_0 \mathcal{G}$ and $\mathcal{D}_1 \mathcal{G}$ are applied to both of $\mathcal{D}_0 \mathcal{G} r$ and $\mathcal{D}_1 \mathcal{G} r$. The number of the wavelet coefficients for each of these is $n/4$ at scale $J - 2$. Hence the total number of coefficients is $n$. For the next steps, these procedures are repeated until reach the coarsest scale.

The number of coefficients produced by non-decimated wavelet transform is $Jn$, since there are $J$-scale, or sometimes written as $n \log_2 n$. The computational effort required to calculate non-decimated wavelet transform is also $\mathcal{O}(n \log_2 n)$. This is not fast as the discrete wavelet transform, which is $\mathcal{O}(n)$, but the non-decimated algorithm could be considered efficient (with a reasonable constant factor).

In this chapter, the HaarSeg method from Ben-Yaacov & Eldar (2008) which is a segmentation method based on the nondecimated Haar wavelet decomposition

and thresholding that is particularly designed for the copy number segmentation problem is considered. The HaarSeg method proposed by Ben-Yaacov & Eldar (2008) includes four main steps. First, the nondecimated discrete wavelet transform is applied to the observed data $r_i$. Then, the second step is to denoise the data using the false discovery rate (FDR) thresholding procedure (Abramovich & Benjamini, 1995), where FDR is defined as the proportion of false positives out of all positives. After obtaining the denoised coefficients, a list of significant breakpoints in the data is created by setting a minimum distance between breakpoints to avoid the same breakpoint being detected at several levels. Finally, the segmentation result is reconstructed from the list of those significant breakpoints.

## 3.3 Tail-Greedy Unbalanced Haar Wavelet

Girardi & Sweldens (1997) introduce the unbalanced Haar wavelet basis where the difference between unbalanced and 'balanced' or traditional Haar wavelet is that the discontinuities in the basis functions do not necessarily occur in the middle of their support. The main idea of the unbalance Haar wavelet transformation (UHWT) is to concentrate as little of the variability in the data as possible at fine scales. The purely greedy approach to perform UHWT was done in the heuristic procedure and outlined in Fryzlewicz (2007) which then was improved using 'tail-greedy' approach in Fryzlewicz (2018). The UHWT does not have any fixed structure like the basic Haar wavelet. Hence the shape or breakpoint location of the unbalanced wavelet basis used in the TGUH transformation can be adjusted following the data. The Tail-Greedy Unbalanced Haar (TGUH) decomposition has been proven to be a powerful tool to estimate the locations of multiple change-points in data. Consider the problem of recovering a piecewise constant signal $f_i$ from its noisy measurements (observed copy number ratio) $r_i$ as modelled in equation (2.2) where $i = 1, \ldots, n$ and $n$ is the total number of windows in the genome, the TGUH approach proposed consists of three main steps: (i) Forward TGUH transform, (ii) Thresholding and (iii) Inverse TGUH transform.

### 3.3.1 Step 1: TGUH Transformation

The TGUH transformation is a bottom-up method that chooses adjacent pairs of data that are believed to have the least variability in each iteration, in an attempt to concentrate as little as possible variability or "power" in the data at the "finer" or lower levels of resolution (Fryzlewicz, 2018). This is done starting from the finest scale by recursively merging some neighbouring regions that have the smallest power or "differences". The merge here means calculating the "differences" between two consecutive regions and treating those two regions as one single region for the next scale.

The algorithm starts by first initiating the variables. Define the parameter $j$ to describe the scale of the transform. After each TGUH transformation procedure described below, the scale $j$ will increase by one. At the "finest" scale $j = 1$, the regions merged are between some neighbouring regions that are all individual points $\{i\}$, $i = 1, \ldots, n$. While at the "coarsest" scale $j = J$, there is only a single merge between regions $\{1, \ldots, b\}$ and $\{b+1, \ldots, n\}$ for a certain $b \in \{1, \ldots, n-1\}$.

Let $c_{s,e}$ be the smooth (local rescaled average) coefficients of copy number ratio data $r_i$ given by

$$c_{s,e} = \frac{1}{\sqrt{e-s+1}} \sum_{i=s}^{e} r_i. \tag{3.4}$$

The two subscripts in $c_{s,e}$ denote the start ($s$) and end ($e$) index of the region of the data used to compute $c_{s,e}$. For $j = 1$, the smooth coefficients are assigned to be the input data itself, $\mathbf{c} = (c_{1,1}, c_{2,2}, \ldots, c_{i,i}) = (r_1, r_2, \ldots, r_i)$.

For each iteration $\lceil \rho \alpha_k \rceil$ pairs are merged, where the parameter $\rho$ controls the proportion of pairs to merge in each iteration and $\alpha_j$ is the number of regions remaining, after the $j$-th iteration. In the application in this thesis, $\rho$ is set to be equal to 0.01 (Fryzlewicz, 2018).

To be more precise, the algorithm proceeds as follows:

1. At the $j$-th iteration, for each adjacent pair of local rescaled average coefficients, construct a 'detail' filter $(l_{s,b}, -r_{b+1,e})$ with $l_{s,b}^2 + r_{b+1,e}^2 = 1$ and $d_{s,b,e} = l_{s,b}c_{s,b} - r_{b+1,e}c_{b+1,e}$ should be zero if $(r_s, \ldots, r_e)$ is a constant vector. Compute the detail coefficient defined by

$$d_{s,b,e} = l_{s,b}c_{s,b} - r_{b+1,e}c_{b+1,e} \tag{3.5}$$

for each adjacent pair of coefficients in $\mathbf{c}$ and sort the sequence $|d_{s,b,e}|$ in ascending order. Then, search for the $\lceil \rho\alpha_k \rceil$ pairs of smooth coefficients that have the smallest absolute value of the detail coefficient vector and save them as detail coefficients of scale $j$, $d^j_{s,b,e}$.

2. Merge the local rescaled average coefficients which correspond to the selected detail coefficients. Then, produce new local rescaled average coefficients which define the scaled average of those merged regions. Specifically, for a selected detail coefficient $d^j_{s,b,e}$ at iteration $j$, the regions $\{s, \ldots, b\}$ and $\{b+1, \ldots, e\}$ are merged into single region $\{s, \ldots, e\}$. Note that the new detail and smooth coefficients pair $(d^j_{s,b,e}, c_{s,e})$ is the result of rotation of the pair $(c_{s,b}, c_{b+1,e})$ as following

$$\begin{pmatrix} d^j_{s,b,e} \\ c_{s,e} \end{pmatrix} = \begin{bmatrix} l_{s,b} & -r_{b+1,e} \\ r_{b+1,e} & l_{s,b} \end{bmatrix} \begin{pmatrix} c_{s,b} \\ c_{b+1,e} \end{pmatrix} =: \Lambda_{s,b,e} \begin{pmatrix} c_{s,b} \\ c_{b+1,e} \end{pmatrix}. \tag{3.6}$$

3. Set $j = j + 1$ and go back to step 1, unless only one detail coefficient was extracted in step 2, in which case the algorithm terminates.

### 3.3.2 Step 2: Thresholding

In a wavelet context, thresholding is commonly used to remove noise from data by shrinking/deleting some wavelet coefficients that fall below a specified threshold. In the TGUH decomposition, by construction, the bulk of the activity of the data will be concentrated in coarse-scale (large $k$) detail coefficients and fine-scale (small $k$) coefficients will be small and contain mostly noise. Therefore, by removing those coefficients which are smaller than some threshold, most of the noise can be removed.

The thresholding technique that is used by Fryzlewicz (2018) is called "connected thresholding". This thresholding preserves the 'unary-binary' structure of the detail coefficients and produces an estimate where the number of change-points is equal to the number of detail coefficients.

Let the children coefficients of detail coefficient $d^j_{s,b,e}$ be the set of finer-scale coefficients whose support is entirely inside $[s, e]$:

$$\mathcal{C}^j_{s,b,e} = \{d^{j'}_{s',b',e'} : [s', e'] \subseteq [s, e] \text{ for all } j' = 1, \ldots, j-1\}. \tag{3.7}$$

Connected thresholding, with threshold $\lambda > 0$, sets to zero all detail coefficients $d^j_{s,b,e}$ for which $|d^j_{s,b,e}| < \lambda$ and each of its children coefficients are also smaller in magnitude than $\lambda$. More formally, if $g^j_{s,b,e}$ and $d^j_{s,b,e}$ are the detail coefficients respectively of the true unknown signal $f_i$ and the observed data $r_i$ in Equation (2.2), the connected thresholding estimate of $g^j_{s,b,e}$ is given by

$$\hat{g}^j_{s,b,e} = d^j_{s,b,e} \mathbb{1}\{\exists d^{j'}_{s',b',e'} \in \mathcal{C}^j_{s,b,e} > \lambda\}, \tag{3.8}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. The default threshold $\lambda$ for the TGUH method is defined by

$$\lambda = \sigma(2(1 + 0.01)\log n)^{1/2}, \tag{3.9}$$

where $\sigma$ is estimated by computing the median absolute deviation (MAD) of the sequence $\{|r_{i+1} - r_i|/\sqrt{2}; i = 1, \ldots, n-1\}$ (Fryzlewicz, 2018). In other words, $g^j_{s,b,e}$ are estimated by zero if only if both $d^j_{s,b,e}$ and its all children coefficients fall below the threshold $\lambda$.

### 3.3.3 Step 3: Inverse TGUH Transform

Now, a set of survived detail coefficients, $\hat{g}^j_{s,b,e}$, was obtained. The final estimator can be obtained by taking inverse TGUH transform to those survived coefficients. The inverse TGUH transformation is performed by undoing the rotations specified in formula 3.6. Since $\Lambda_{s,b,e}$ is an orthonormal matrix, the operation can easily be undone as follows:

$$\begin{pmatrix} c_{s,b} \\ c_{b+1,e} \end{pmatrix} = \Lambda^T_{s,b,e} \begin{pmatrix} d_{s,b,e} \\ c_{s,e} \end{pmatrix} = \begin{bmatrix} l_{s,b} & r_{b+1,e} \\ -r_{b+1,e} & l_{s,b} \end{bmatrix} \begin{pmatrix} d_{s,b,e} \\ c_{s,e} \end{pmatrix}. \tag{3.10}$$

## 3.4 Simulation Study

A simulation experiment was conducted to compare the performance of the wavelet methods. Two kinds of the true function $f_i$ were considered and presented in Figure 3.1. The first type of true function is based on the block signal, a well-known synthetic signal taken from the work by Donoho & Johnstone (1994) with three additional short segments located at points $500, 600$, and $900$. The length of these short segments is set to be 6 points as in the copy number study the shortest

meaningful segment is about 1 Mb which is equivalent to 6-7 points in our data. The change in height of the first short segment was set to 1 and the others to 0.5 to see how each method deals with different heights of short segments. The second type of true function is based on the test function used in Fryzlewicz (2018) which aims to evaluate the performance in estimating segments with different lengths.

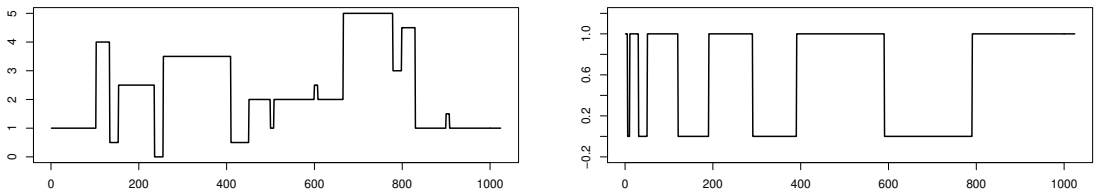


Figure 3.1: The first (left panel) and second (right panel) true functions.

The simulated CNA data $r_i$ were generated using the model $r_i = f_i + \epsilon_i$ where $\epsilon_i$ is a random error term independently distributed as $N(0, \sigma^2)$. The simulations were repeated for $\sigma = 0.1, \ldots, 0.5$ to obtain a controlled comparison of different levels of noise variance relative to the changes that are wished to be detected in CNA data, which are generally of magnitude 0.5 or 1. One thousand simulated data sets were generated for each value of $\sigma^2$ and all the segmentation methods explained in the previous sections were applied to each of them.

To evaluate the operating characteristic of each method, the problem of copy number segmentation can be viewed as a binary classification problem (Pierre-Jean *et al.*, 2015). A sequence $r = \{r_i\}_{i=1}^n$ is said to have a breakpoint at $j$ if $|r_{j+1} - r_j| > \theta$, where $1 \leq j < n$. The true positive (TP) is defined as an estimated breakpoint whose location is found inside a given tolerance parameter and closest to the true breakpoint location while the false positive (FP) is the remaining estimated breakpoints. For this simulation study, $\theta$ is set to $\theta = 0.1$ (Mermel *et al.*, 2011) and the tolerance parameter is equal to two, which means that an estimated breakpoint is classified as a true positive if it is closest and located within two points to the left/right to the true breakpoint. The illustration of these definitions is presented in Figure 3.2. Based on this definition, the average true positive rate (aTPR) and the average false positive rate (aFPR) were computed over

1000 replicates. Receiver operating characteristic (ROC) curve for each method over different noise levels was also presented to further evaluate the operating characteristics of each method.

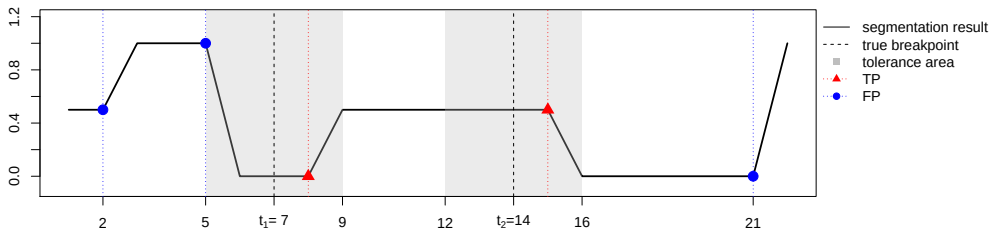


Figure 3.2: Illustration of false positive and true positive to build performance evaluation.

### 3.4.1 Results

Figure 3.3 indicates that the HaarSeg and TGUH methods are much better than the basic Haar wavelet methods in terms of average Mean Integrated Squared Error (aMISE) for both the first and second simulation settings. Figure 3.4 shows the average TPR (aTPR) results of simulation using the first and second test functions. These results indicate that the TGUH performs very well by showing excellent results in both simulation settings. Besides TGUH, HaarSeg comes second, showing a good result in terms of aTPR for the second test function, where it surpasses TGUH for the highest noise level ($\sigma = 0.5$).

Even though HaarSeg works well in terms of aMISE and estimating altered location, it comes with a high average false positive rate (aFPR) compare to TGUH as shown in Figure 3.5. Figure 3.5 also indicates that the aFPR of the basic Haar wavelet method using hard and soft thresholding is far above the TGUH method. This is due to the dyadic structure of the balanced Haar wavelet transform which causes a tendency to form spurious breakpoints at dyadic locations irrespective of the location of the true breakpoints in the underlying signal.

To investigate more the performance of each method in estimating the correct location of alterations, Figure 3.6 shows how many times (from 1000 simulated

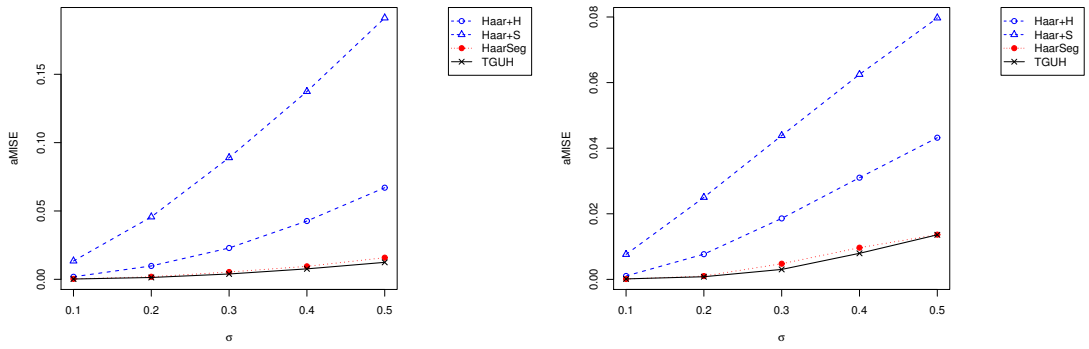


Figure 3.3: Average Mean Integrated Squared Error (aMISE) of simulation using first (left) and second (right) test function over 1000 replicates.

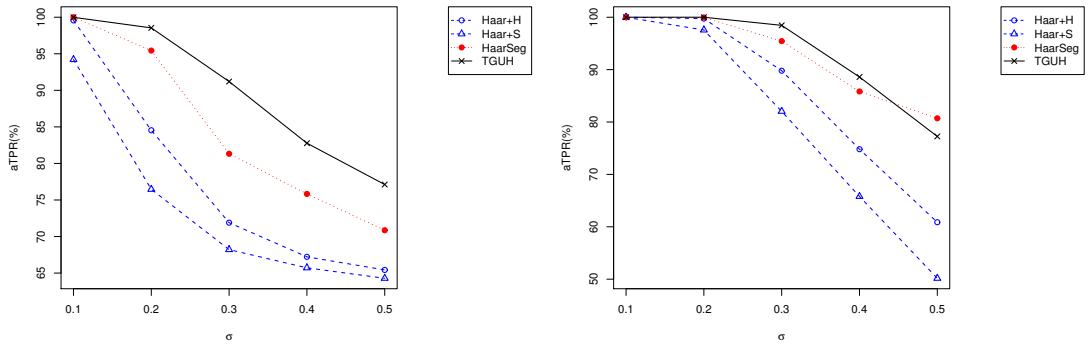


Figure 3.4: Average True Positive Rate (aTPR) of simulation using first (left) and second (right) test function over 1000 replicates.

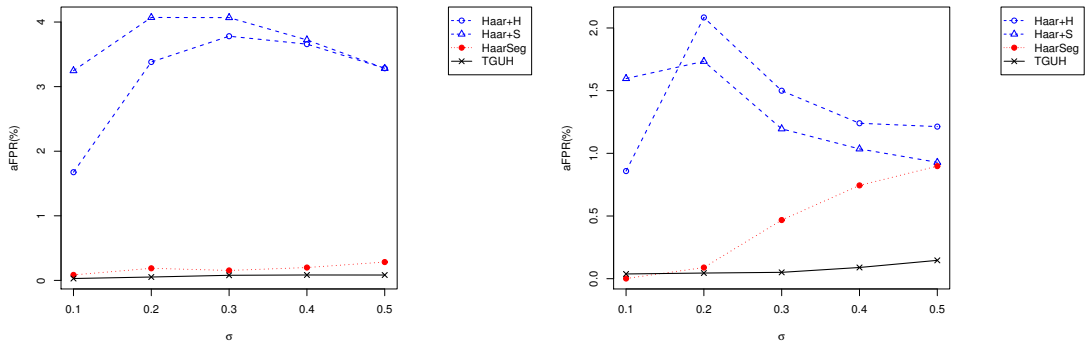


Figure 3.5: Average False Positive Rate (aFPR) of simulation using first (left) and second (right) test function over 1000 replicates.

datasets) each method detects a breakpoint at each location along the sequence. Here, only the results for $\sigma = 0.3$ are shown. The results of other noise levels are quite similar in terms of the pattern/rank but with different heights. Based on Figure 3.6, TGUH has the highest sensitivity in terms of detecting short segments while still showing a good performance in estimating long segments.

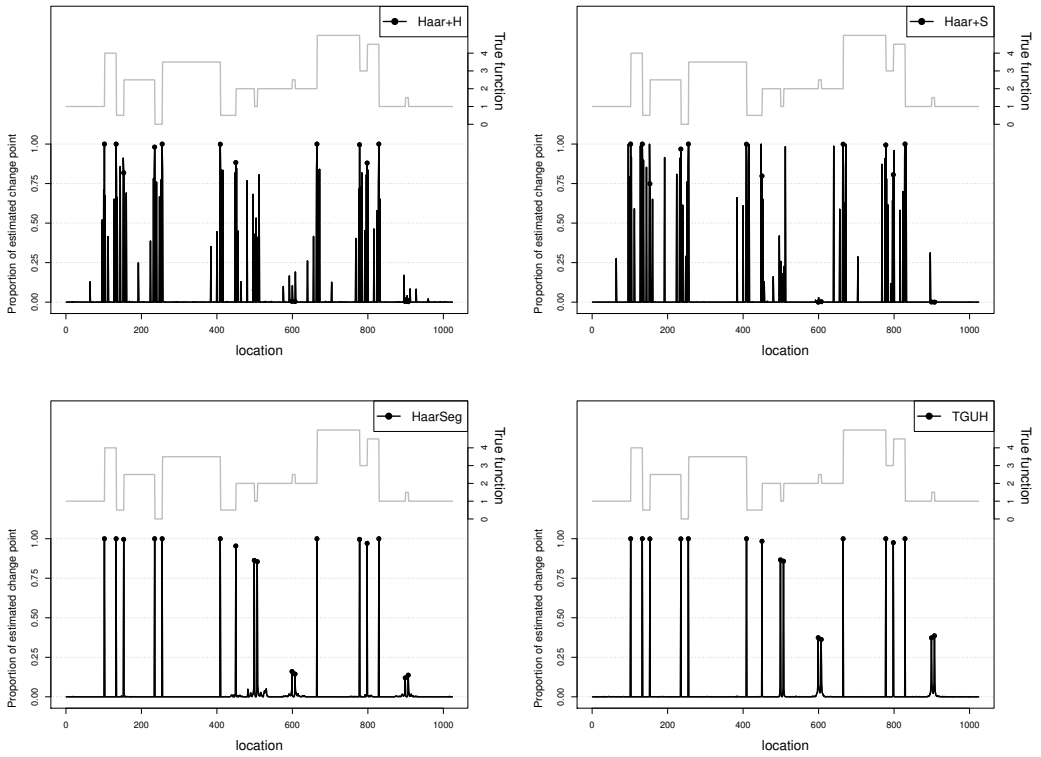


Figure 3.6: Plot of the proportion of replicates with an estimated breakpoint against location. Each value denotes the proportion of replicates where a breakpoint is found at the corresponding location out of 1000 simulated datasets contaminated by Gaussian noise with mean 0 and variance $\sigma^2 = 0.3^2$. The dots denote proportion of each of the methods producing breakpoints at the correct location. The grey solid line is the corresponding test function, repeated here from Figure 3.1 for a quick reference. The left and right vertical axis show the proportion of replicates with an estimated breakpoint and the corresponding test function's height, respectively.

A careful inspection shows that the basic Haar has peaks at the dyadic location near the true altered positions. More details explanation of this tendency is explained in Section 3.5. Even though HaarSeg tried to address this problem by setting a minimum distance between breakpoints, this artefact still remains and is reflected by small peaks near the true breakpoint location. Only TGUH

method can provide a clean segmentation result without showing any tendency to estimate spurious breakpoints at a particular location.

Figure 3.7 shows the AUC for all tested methods over different noise levels. Only for the second simulation with $\sigma = 0.5$, the AUC of TGUH below HaarSeg but it outperforms the others for the remaining.

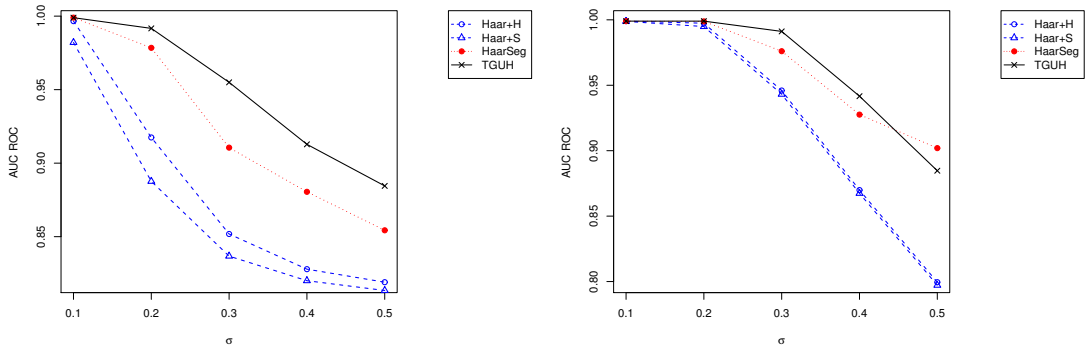


Figure 3.7: AUC of ROC of the methods correspond to the first (left) and second (right) type of simulated data over different noise level.

## 3.5 Dyadic Structure of Balanced Haar Wavelet Transform

Based on the simulation in Section 3.4, the balanced Haar wavelet method has a strong tendency for estimating change points at dyadic locations, and this brings a disadvantage to it as, in practice, the change points are not only located at dyadic locations. In this section, the focus is to discuss and seek to explain in more detail the reasons that cause this tendency in the balanced Haar wavelet method.

The most interesting property of the wavelet transform compared to the other signal processing transforms is *localization*. This localization feature makes wavelet transforms able to represent many functions "sparse" which results in many useful applications such as noise removal. Another characteristic of this localization feature is if a function $f(x)$ has a discontinuity or a jump (if $f(x)$ is a

piecewise constant function), it only affects the wavelet $\psi_{j,k}(x)$ that close to it. Only wavelet coefficients corresponding to wavelet $\psi_{j,k}(x)$ that overlaps the jump will be influenced.

This localization characteristic can also be seen in the balanced Haar wavelet transform. The balanced Haar wavelets can only shift at dyadic locations and do not themself overlap, hence, each point in the original data only corresponds to at most one wavelet at each scale. Due to its even dyadic decimation step which essentially picked every even element from an input vector, if the jump is located at for example 1/2 of the input vector, only the coarsest level wavelet coefficient will carry out this jump information. On the other hand, if it is not located at the dyadic location, then more wavelets will be affected.

Figure 3.8 shows the comparison of the Haar wavelets that are affected by the discontinuity in the vector $\{x_i\}_{i=1}^{16}$ which located between $x_8$ and $x_9$ (top panel) and $x_5$ and $x_6$ (bottom panel). The functions inside the graph illustrate the Haar wavelets used at each of the resolution levels to extract the information of the piecewise constant vector $\{x_i\}_{i=1}^{16}$ with only one discontinuity. When the discontinuity is located between $x_8$ and $x_9$ or exactly in the middle of the data, the coarsest scale (resolution level 0) wavelet is the only wavalet that carries out the information of this discontinuity as it is the only wavelet that overlaps the jump. While, when the discontinuity is located between $x_5$ and $x_6$, there are four wavelets overlap, which are denoted by red lines.

In terms of wavelet denoising, if noise is added to the vector $\{x_i\}_{i=1}^{16}$, the coefficients which correspond to those red color wavelets are likely to survive the thresholding. There is no big problem if the discontinuity is located at the dyadic location, but when the discontinuity is not located at the dyadic location, as shown in the bottom panel of Figure 3.8, there will be many spurious change points estimated. This is because not only the coefficient that corresponds to the wavelet whose breakpoint exactly aligns at the discontinuity survive the thresholding, but other coefficients in the coarser scales that correspond to the wavelet that overlaps the discontinuity also tend to survive the thresholding.

A simple simulation was carried out to illustrate the tendency of Haar wavelet estimation. A true function $\{x_i\}_{i=1}^{16}$ where $x_i = 0\mathbb{I}(1 \leq i \leq 5) + 3\mathbb{I}(6 \leq i \leq 16)$ was considered. Then, it is contaminated by Gaussian noise with mean 0 and variance

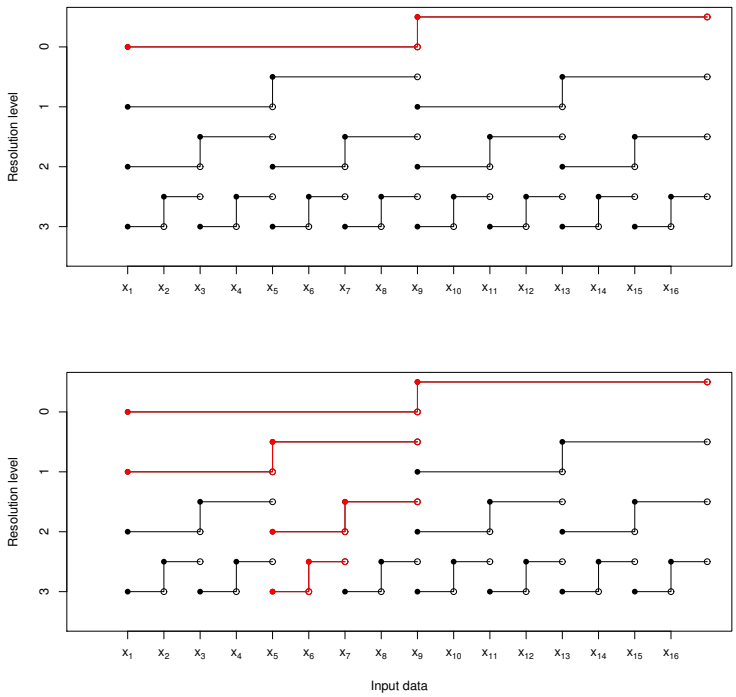


Figure 3.8: The illustration of Haar wavelet transform of an input data $\{x_i\}_{i=1}^{16}$ which contains a jump between $x_8$ and $x_9$ (top panel) and $x_5$ and $x_6$ (bottom panel). The vertical axis denotes the resolution level or scale and the horizontal axis denotes the input data. The functions inside the graph illustrate the Haar wavelets used at each of the resolution level to extract the information of the input vector $\{x_i\}$. The red color denotes Haar wavelets that are influenced by the jump.

$\sigma = 0.1$ to produce a noisy data $x_i^*$ (see top panel of Figure 3.9) and perform the Haar wavelet estimation to estimate $x_i$. The simulation was conducted 1000 times and the change-points estimated were counted at each location over 1000 trials (see bottom panel of Figure 3.9). Based on Figure 3.9, the Haar wavelet method obviously has a tendency to estimate change-point not only at the location where the jump is located at the true function but also at the dyadic location close to the true jump.

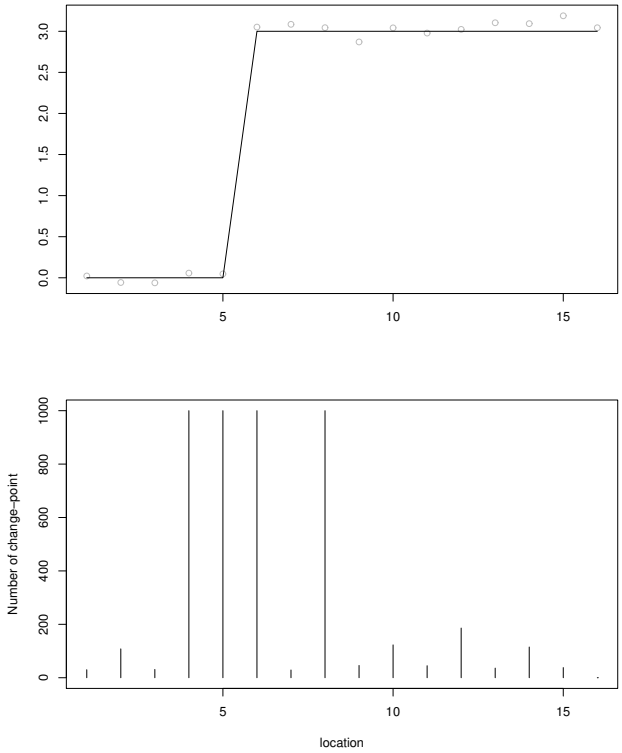


Figure 3.9: Top: The true function $x_i$ (black solid line) and the simulated data $x_i^*$ (gray dots). Bottom: Plot of the frequency of change-point estimation against location. Each value denotes how many times a change-point is found at the corresponding location over 1000 simulated datasets.

## 3.6 Application to Real Data

Figure 3.10 presents the results of segmentation based on the basic Haar, HaarSeg, and TGUH methods in chromosome 12 of lung adenocarcinoma patient LA57.

Each point in Figure 3.10 denotes the copy number ratio of LA57 patient data which corresponds to a genomic window (with size 150 kb). CNAnorm Gusnanto *et al.* (2012) was used to normalize the data and the missing regions were removed. From this example, the segmentation characteristic of each method can obviously be seen.

A visible difference between hard and soft thresholding used in the basic Haar wavelet method is clearly illustrated in Figure 3.10. The magnitude of the change or transition between segments of the hard thresholding is high compared to the soft one. This is due to the hard thresholding rule leaving large coefficients unchanged, while soft thresholding shrinks them towards zero, as defined in equation (2.40) and (2.41). This shrinkage will result in smaller jumps/drops within segments.

HaarSeg segmentation estimates fewer breakpoints than the basic Haar method but it tends to estimate a low magnitude of breakpoints near the high one as the one located around position 250 on chromosome 12. Among all the segmentation results, TGUH shows the cleanest segmentation as presented in the bottom right panel of Figure 3.10.

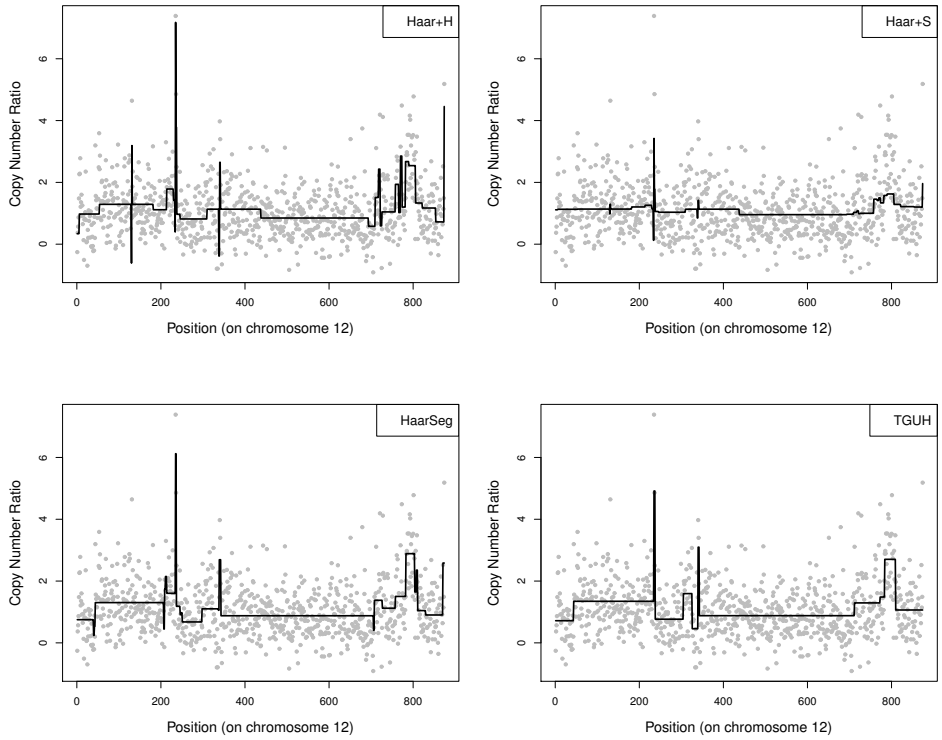


Figure 3.10: CNA estimate as a result of Haar wavelet-based segmentation of chromosome 12 from patient LA57. Top left: Haar+H segmentation. Top right: Haar+S segmentation. Bottom left: HaarSeg segmentation. Bottom right: TGUH segmentation.

## 3.7 Conclusion

In this chapter, a comparative simulation study was presented to evaluate the performance of several Haar wavelet-based methods for the segmentation of CNA data. The results suggest that the TGUH method has good operating characteristics to detect segments of different sizes and provide a clear segmentation result. The basic Haar wavelet method and HaarSeg method have a tendency to identify more spurious breakpoints due to the dyadic structure of the balanced Haar wavelet transformation which was described in Section 3.6. Only TGUH offers clean segmentation with high sensitivity but a low false positive rate.

In general, the appealing point of the wavelet approach is its ability to extract multiscale 'information' from the data and represent them as a series of coefficients. The key information that can be extracted here is the variation in the data at different scales and different locations. Moreover, the flexibility of the TGUH method to adjust the location of its discontinuity in the unbalanced Haar wavelets to follow the likely structure of the signal, resulting in more precise breakpoint location estimates than alternatives based on traditional balanced Haar wavelets. This advantage has made the TGUH method the most preferable alternative for CNA segmentation compared to the basic Haar and HaarSeg methods.

# Chapter 4

# Modified TGUH Method for Copy Number Segmentation

## 4.1 Introduction

The occurrence of extreme observations (outliers) of biological or technical origin is inevitable in NGS data. These outliers pose an additional challenge to the segmentation method's ability to provide a clear result. However, copy number segmentation of NGS data with the TGUH method which has been explained in Section 3.3 tends to estimate spikes (very short altered segments of only one or two data points) as a result of these outliers. Figure 4.1 shows the TGUH segmentation of the copy number ratio of patient TMA-93 which obviously illustrates spikes (due to extreme single points) that are often found in TGUH segmentation. Given that those spikes are single points, they rarely represent 'true' changes. Thus, an investigation of how these spikes occur is needed to improve the performance of the TGUH method.

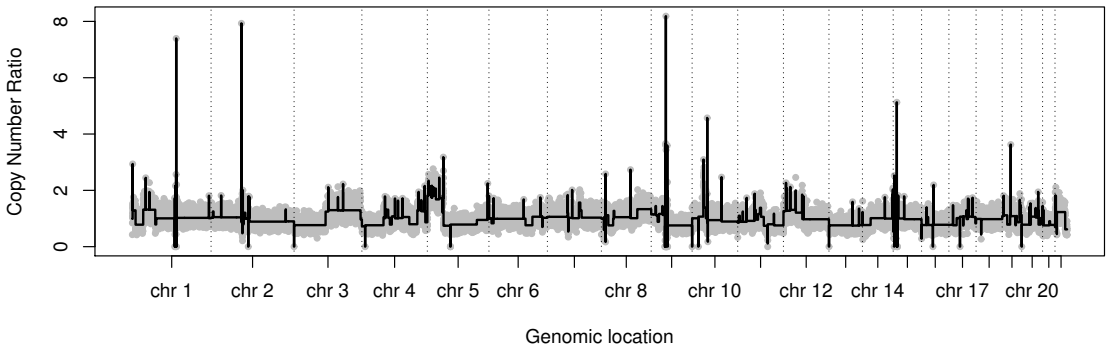


Figure 4.1: TGUH estimate as a result of segmentation of patient TMA-93.

This chapter particularly deals with the investigation of spurious change-points that are commonly found in the TGUH segmentation as single point spikes and also describes a modification to the TGUH method to reduce the occurrence of these spikes. The layout of this chapter is as follows: Section 4.2 describes how the spikes occur in the TGUH segmentation. This includes the visualisation of the TGUH transform and the explanation of how the outliers in the data can cause spurious change points to appear in the TGUH estimates. A modified TGUH method, named the TGUHm method, is proposed to deal with this problem in Section 4.3. A simple thresholding procedure, which is described in Section 4.3.2, is performed in the TGUHm method as an addition to the connected thresholding used in the original TGUH method to reduce the spikes. To assess the performance of the TGUHm method, some simulation studies are presented in Section 4.4. In Section 4.5, typical segmentation patterns for real NGS data are used in the simulations, and the segmentation results of the TGUHm method are compared with the original TGUH method and the other well-known segmentation methods. A paper based on the work in this chapter is currently under review (Ummi *et al.*, 2023 submitted).

## 4.2 Visualisation of TGUH Detail Coefficients

To identify the reason why spikes often occur in the TGUH segmentation, it is important to be able to illustrate or visualise the TGUH detail coefficient, $d_{s,b,e}$. Here, a simple example is shown to illustrate the relationship between detail coefficients with the resulting estimation.

Example 1. Let $x = (x_1, \ldots, x_n) = (1, 0, 3, 3, 2, 4, 3, 4, 5, 2)$. To perform TGUH segmentation as described in Section 3.3, the parameter $\rho$ is set to be 0.01 (Fryzlewicz, 2018). Then, for scale $j = 1$ or the first iteration, the smooth coefficients $c_{s,e}$ are set as

$$(c_{1,1}, c_{2,2}, \ldots, c_{10,10}) := (1, 0, 3, 3, 2, 4, 3, 4, 5, 2) \tag{4.1}$$

so that the filter coefficients are $(l_{s,b}, -r_{b+1,e}) = (1/\sqrt{2}, -1/\sqrt{2})$ since $s = b$ and $b + 1 = r$. For a quick reminder, $s$, $b$, and $e$ are related to the 'start', 'breakpoint', and 'end' of the unbalanced wavelet basis used to construct the detail coefficient $d_{s,b,e}$, respectively. The detail coefficient, $d_{s,b,e}$, is computed by $d_{s,b,e} = l_{s,b}c_{s,b} - r_{b+1,e}c_{b+1,e}$ for each adjacent pair of smooth coeffcients. For this example, the smallest detail coefficient in absolute order is $d^{1,1}_{3,3,4} = 0$. Therefore, $d^{1,1}_{3,3,4}$ is saved as a detail coefficient of scale one. Then, the pair of neighbours $(s_{3,3}, s_{4,4}) = (3, 3)$ is merged to be $s_{3,4} = (3 + 3)/\sqrt{2}$. At the end of the above pass through the data at scale $j = 1$, the input vector will therefore reduced to $(s_{1,1}, s_{2,2}, s_{3,4}, s_{5,5}, s_{6,6}, s_{7,7}, s_{8,8}, s_{9,9}, s_{10,10})$. Then, for the next scales, the same procedures are performed recursively. The iteration is continued until the input vector has been reduced to the single coefficient $s_{1,10}$. In this example will produce nine detail coefficients as following

$$\begin{aligned}(d^{1,1}_{3,3,4}, d^{2,2}_{1,1,2}, d^{3,3}_{6,6,7}, d^{4,4}_{6,7,8}, d^{5,5}_{3,4,5}, d^{6,6}_{6,8,9}, d^{7,7}_{3,5,6}, d^{8,8}_{3,6,10}, d^{9,9}_{1,2,10}) = \\ (0, 0.707, 0.707, -0.408, 0.816, -1.155, -1.745, 1.336, -3.478).\end{aligned} \tag{4.2}$$

The connected thresholding was performed to the detail coefficients in (4.2 to denoise or obtain an estimate of the 'true' underlying piecewise constant signal on which the noisy data $x$ is based. The detailed procedure of connected thresholding is explained in Section 3.3. Using formula in (3.9), the universal

threshol $\lambda = 2.2609$ is obtained. Therefore, by the thresholding, only $d^{9,9}_{1,2,10}$ survives and the remaining coefficients are set to zero.

Figure 4.2 shows the plot of detail coefficients of Example 1 before and after the thresholding. Throughout this thesis, the detail coefficients are depicted by plotting the detail coefficients $d_{s,b,e}$ against $b$. To be more precise, if coefficients $d_{s,b,e}$ survive the thresholding, this means that there is a change point at location $b$. The coefficients $d_{s,b,e}$ are plotted against index $b$ and the value of the coefficient is displayed by a vertical mark located along the imaginary line $y = 0$. The value or magnitude of the detail coefficients is displayed by a vertical mark located along the region that merges the line that corresponds to the coefficient. The position of the coefficients on the region merges line indicates the index $b$. The red and blue colours of the lines show the positive and negative signs of the coefficients, respectively. The black dashed lines indicate the detail coefficients whose value is equal to zero.

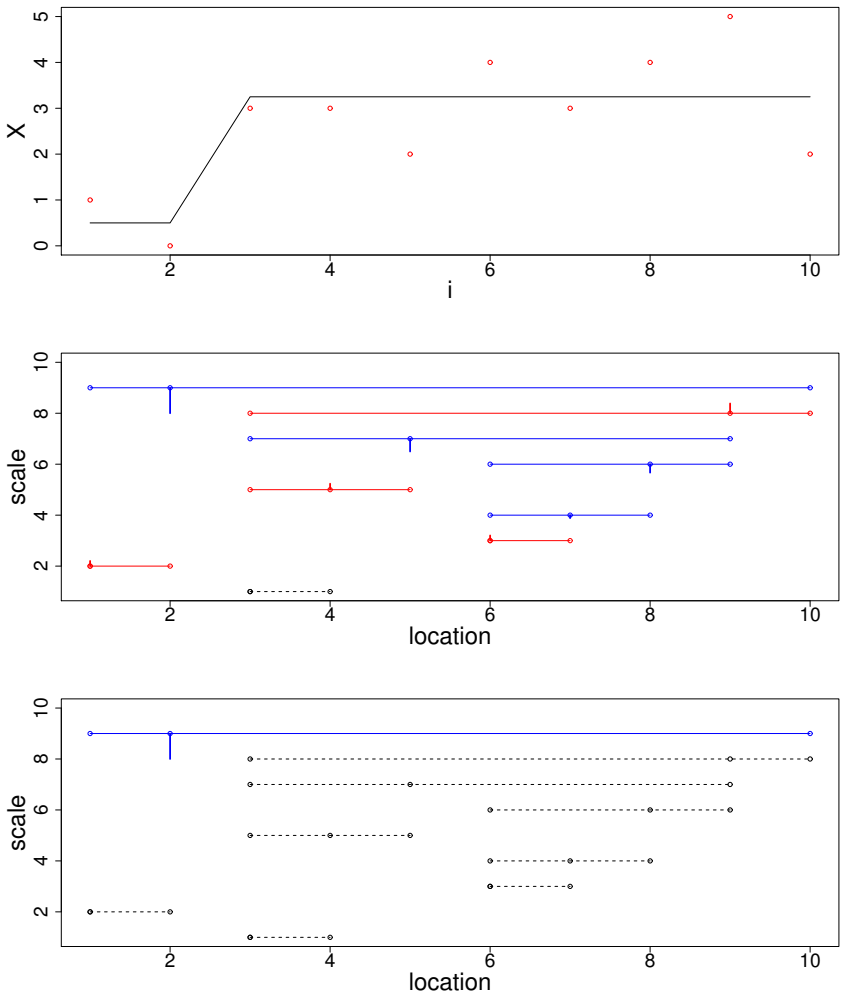


Figure 4.2: *Top:* Plot of the sequence $x_i$ and its resulting TGUH segmentation. Each of the red dots denotes the $x_i$ and the black solid line denotes $x_i$ after TGUH denoising. *Middle:* Plot of the detail coefficients of $x_i$ before the thresholding. *Bottom:* The detail coefficients of $x_i$ after thresholding. The value or magnitude of the detail coefficients is displayed by a vertical line located along the region that merges the line that corresponds to the coefficient. The position of the vertical line indicates the index $b$. The red and blue colours of the lines show the positive and negative signs of the coefficients, respectively. The black dashed lines indicate the detail coefficients whose value is equal to zero.

### 4.2.1 The Occurrence of Spikes in The Estimation

Spikes in the TGUH estimate are likely to occur when the detail coefficients $d^k_{s,b,e}$, with either $e-b$ or $b-s+1$ equals to one, survive the thresholding. Ideally, to control the occurrence of these 'spikes', the detail coefficients $d^k_{s,b,e}$ with either $e-b$ or $b-s+1$ less than a constant $m^*$ need to be set to zero. By setting $m^*=2$, the spikes can be reduced and the user will have direct control of the minimum length of segments. More formally, the connected thresholding estimate of $g^k_{s,b,e}$ in (3.7) can be rewritten as

$$\hat{g}^k_{s,b,e} = d^k_{s,b,e}\mathbb{1}\{\exists d^{k'}_{s',b',e'} \in \mathcal{C}^k_{s,b,e} > \lambda\}\mathbb{1}\{(b-s+1) > m^*\}\mathbb{1}\{(e-b) > m^*\}. \quad (4.3)$$

But due to the unary-binary structure of connected thresholding, there could be a case when a segment with a length less than $m^*$ could not be removed. This is due to the connected thresholding being unable to delete parent detail coefficients which have children coefficients whose magnitudes are above the threshold even though the parent coefficient corresponds to a segment with length less than $m^*$.

Let us define "wing" as the length of either $b-s+1$ or $e-b$ (the length of TGUH basis from the start point ($s$) to the breakpoint ($b$) or breakpoint ($b$) to the endpoint ($e$)). The length of the segments in the final estimator is determined by the length of the TGUH wavelet wing length used to produce the detail coefficients. Therefore, to remove all segments whose length is less than $m^*$, it must be ensured that all detail coefficients corresponding to the TGUH wavelet with wings length less than $m^*$ are to be removed. A simple example to illustrate this condition is presented in Figure 4.3.

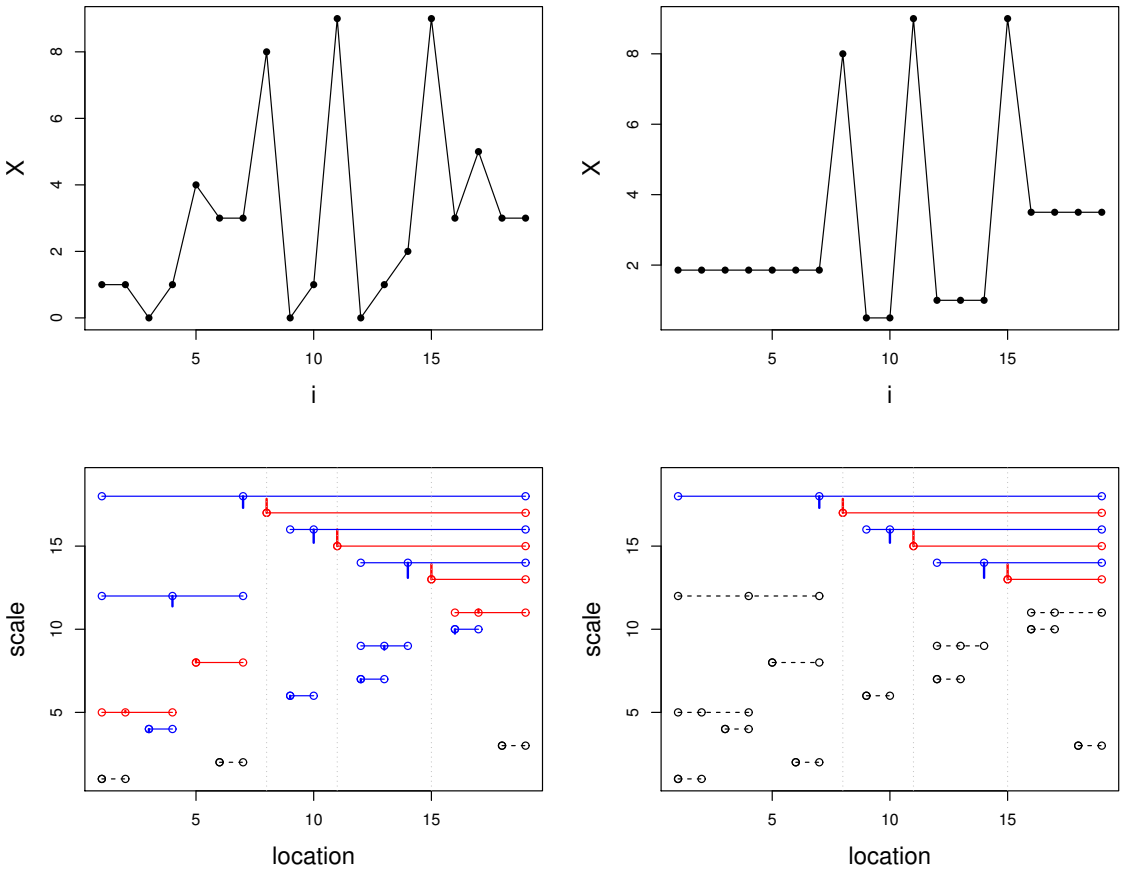


Figure 4.3: The detail coefficients of noise function $X_i$ before and after the thresholding. *Top left:* Plot of noise function $X_i$. *Top right:* Plot of TGUH denoising of $X_i$. *Bottom left:* The detail coefficients of $X_i$ before the thresholding. *Bottom right:* The detail coefficients of $X_i$ after the thresholding with $m^* = 1$. The grey dotted lines denote detail coefficients corresponding to the detail coefficient $d_{8,8,19}$, $d_{11,11,19}$, and $d_{15,15,19}$ which related to spikes in TGUH denoising of $X_i$ (top right panel). The value or magnitude of the detail coefficients is displayed by a vertical line located along the region that merges the line that corresponds to the coefficient. The position of the vertical line indicates the index $b$. The red and blue colours of the lines show the positive and negative signs of the coefficients, respectively. The black dashed lines indicate the detail coefficients whose value is equal to zero.

Figure 4.3 presents plots of a noised data $X_i = (X_1, X_2, ..., X_{19}) = (1, 1, 0, 1, 4, 3, 3, 8, 0, 1, 9, 0, 1, 2, 9, 3, 5, 3, 3)$ and its resulting TGUH segmentation together with their corresponding detail coefficients. The top right panel of Figure 4.3 shows that there are three spikes produced by the TGUH estimate. These spikes are related to detail coefficients $d_{8,8,19}$, $d_{11,11,19}$, and $d_{15,15,19}$. Among those three spikes, only the third spike corresponds to detail coefficient $d_{15,15,19}$ that can be removed by connected thresholding by increasing the value of $m^* > 1$. The detail coefficient $d_{15,15,19}$ does not have any children coefficients that survive the thresholding hence it can be removed without breaking the connected thresholding rule. On the other hand, both $d_{8,8,19}$ and $d_{11,11,19}$ have children coefficients whose value exceeds the threshold $\lambda$ and its unbalanced Haar wavelet wings length (either $e-b$ or $b-s+1$) are greater than one. Therefore these coefficients can not be set to zero as shown in Figure 4.4.

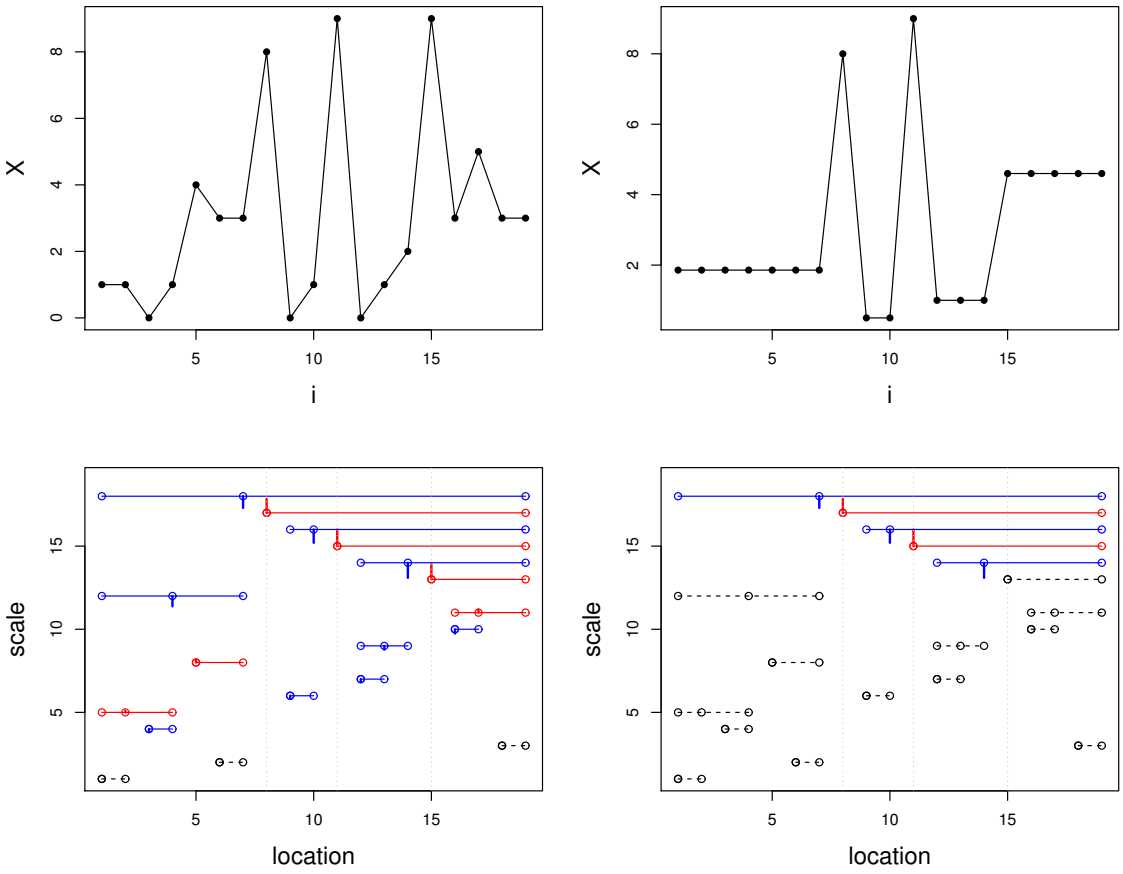


Figure 4.4: The detail coefficients of noise function $X_i$ before and after the thresholding with $m^* = 2$. *Top left:* Plot of noise function $X_i$. *Top right:* Plot of TGUH denoising of $X_i$. *Bottom left:* The detail coefficients of $X_i$ before the thresholding. *Bottom right:* The detail coefficients of $X_i$ after the thresholding with $m^* = 2$. The first, second, and third gray dotted lines denote detail coefficients corresponds to the detail coefficient $d_{8,8,19}$, $d_{11,11,19}$, and $d_{15,15,19}$, respectively, which related to spikes in the TGUH denoising of $X_i$ (top right panel). Here, only the detail coefficient $d_{15,15,19}$ can be 'killed' by setting $m^* = 2$. The value or magnitude of the detail coefficients is displayed by a vertical line located along the region that merges the line that corresponds to the coefficient. The position of the vertical line indicates the index $b$. The red and blue colours of the lines show the positive and negative signs of the coefficients, respectively. The black dashed lines indicate the detail coefficients whose value is equal to zero.

## 4.3 TGUHm method

In this section, the TGUHm method are described. The TGUHm method is a modified TGUH method with an additional procedure in the thresholding step for pruning the spikes commonly found in NGS data. This TGUHm method has shown the best performance compared to four other TGUH-based segmentation methods which are presented in Section 4.4.6.

Let $n$ be the number of windows/regions; we segment each chromosome separately, so $n$ is the number of windows in a chromosome and we do not need an index to denote chromosome. Alternatively, one can segment the whole genome simultaneously, in which case $n$ would denote the number of windows in the entire genome. Let $x_i$ denote the location of the $i$-th window in the chromosome/genome for $i = 1, 2, \ldots, n$, satisfying the condition $x_1 < x_2 < \cdots < x_n$. Let $N$ be the number of change-points in the data, with $0 \leqslant N \ll n$, and if $N > 0$, let $\eta_p$, $p = 1, \ldots, N$ be the locations of the change-points. For a sequence $\{r_i\}_{i=1,\ldots,n}$, a change-point is located at $\eta_p = x_i$ if $|r_{i+1} - r_i| > \theta$, where $0 < i \leq n$. The threshold $\theta$ directly affects the balance between sensitivity (the ability to detect true positives) and specificity (the ability to avoid false positives). A lower threshold may result in higher sensitivity, detecting more potentially relevant alterations, but it may also increase the likelihood of false positives. On the other hand, a higher threshold may increase specificity but could miss some genuine alterations. In this chapter, the height tolerance parameter $\theta$ is set to be equal to 0.1 as suggested in Mermel *et al.* (2011) to give the balance between sensitivity and specificity. As an illustration, $\eta_2 = x_{100}$ means that the second change-point in the data is located at $x_{100}$ and $|r_{101} - r_{100}| > \theta$. In a simulation study, $N$ and the $\eta_p$'s are known, but in practice for real data they are unknown.

In the context of NGS, let $r_i$ denote the ratio between the number of reads in the tumour and normal sample in the $i$-th window corresponding to location $x_i$ (Gusnanto *et al.*, 2012). There is no requirement for the $r_i$ to be normalised. In the normalisation of CNA data from clinical samples, segmentation may be involved at the start and the end of normalisation (see Gusnanto *et al.* (Gusnanto *et al.*, 2012)).

The observed $r_i$ are given by a true (unknown) signal $f_i$ obscured by additive random error. This model can be expressed as

$$r_i = f_i + \epsilon_i, \tag{4.4}$$

where $\epsilon_i$ represents measurement noise and $f$ is a one-dimensional, piecewise-constant signal with change-points at unknown locations $\eta_1, \ldots, \eta_N$. In this section, the error term $\epsilon_i$ is assumed as Gaussian noise with mean zero and variance $\sigma^2$. Thus the problem is how to estimate the true function $f$ from noisy data $r_i$.

The standard TGUH approach proposed by (Fryzlewicz, 2018) consists of three main steps: (i) Forward TGUH transform, (ii) Thresholding and (iii) Inverse TGUH transform. This subsection describes a method called TGUHm that mainly follows these steps but with some modifications in steps (ii) and (iii) to adapt to the characteristics of CNA data from NGS, particularly to address the 'spikes' that commonly occurs in the TGUH estimates. The TGUHm method can be outlined in the following steps.

1. Apply the (standard) TGUH transformation to the data to obtain TGUH detail coefficients. The coefficients are assigned into a unary-binary tree (i.e., one in which each 'parent' coefficient has one or two 'child' coefficients). Please see Section 3.3.1 for more detailed explanation.

2. Threshold or delete those detail coefficients whose values are less than a specified threshold. Two-stage thresholding is performed here to firstly remove smaller coefficients that are believed to represent noise $\epsilon$ rather than the true signal $f$. This is part of the standard TGUH method. Additionally, in the second stage, some coefficients that correspond to 'spikes' are removed. These spikes are occured due to extreme outliers in copy number ratios.

3. Reconstruct the segmentation result by returning the sample mean of the observed data within each segment between consecutive estimated change-points. This step is different to that in the standard TGUH method.

### 4.3.1 Step 1: TGUH Transformation

As explained in section 3.3.1, The TGUH wavelet transform is a bottom-up method that utilises unbalanced wavelets to translate the sequence $r_i$ into a set of different type coefficients that form an unary-binary tree structure (Fryzlewicz, 2018). Therefore, in the first step of the TGUHm method, the TGUH transform is applied to the data $r_i$ to obtain a sparse representation of the data $r_i$ in terms of a set of piecewise-constant basis functions. At the end of the TGUH decomposition of $r_i$, we have a set of detail coefficients $d^{j,k}_{s,b,e}$ and smooth coefficients $c^{j,k}_{s,b,e}$. Please see Section 3.3.1 for more detailed explanation of these coefficients.

### 4.3.2 Step 2: Thresholding

The 'tail-greediness' of the TGUH method induces the bulk of the variance of the data will be concentrated as a few large detail coefficients at coarse-scale (large $k$). Meanwhile, at the fine-scale (small $k$), the detail coefficients will be small and contain mostly noise. Therefore, by removing those coefficients that are smaller than some threshold, most of the noise can be removed. But in some cases where there is a frequent occurrence of outliers, as is often found in NGS data, basic wavelet thresholding is unable to threshold/remove the detail coefficients corresponding to these outliers as they are likely translated into large coarse-scale coefficients by the TGUH transform. This causes the final estimator to contain spurious change-points as spikes (very short altered segments of only one or two data points). In the TGUHm method, therefore, an additional procedure is added to the connected thresholding used in TGUH (Fryzlewicz, 2018) for pruning these spikes.

In more detail, the thresholding procedure in the TGUHm method proceeds in the following two stages.

1. *Connected thresholding*. Perform connected thresholding to detail coefficients $d^{k}_{s,b,e}$. This thresholding is used by Fryzlewicz (2018) which preserves the 'unary-binary' structure of the detail coefficients and produces an estimate where the number of change-points is equal to the number of detail coefficients.

Let the children coefficients of detail coefficient $d^k_{s,b,e}$ be the set of finer-scale coefficients whose support is entirely inside $[s, e]$:

$$\mathcal{C}^k_{s,b,e} = \{d^{k'}_{s',b',e'} : [s', e'] \subseteq [s, e] \text{ for all } k' = 1, \ldots, k-1\}.$$

Connected thresholding, with threshold $\lambda > 0$, sets to zero all detail coefficients $d^k_{s,b,e}$ for which $|d^k_{s,b,e}| < \lambda$ and each of its children coefficients are also smaller in magnitude than $\lambda$. More formally, if $g^k_{s,b,e}$ and $d^k_{s,b,e}$ are the detail coefficients respectively of the true unknown signal $f$ and the observed data $y$ in Equation (4.4), the connected thresholding estimate of $g^k_{s,b,e}$ is given by

$$\hat{g}^k_{s,b,e} = d^k_{s,b,e} \mathbb{1}\{\exists d^{k'}_{s',b',e'} \in \mathcal{C}^k_{s,b,e} > \lambda\}, \tag{4.5}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function.

2. *Unconnected thresholding*. An additional 'unconnected' form of thresholding is proposed after the above step. This thresholding does not preserve the 'unary-binary tree' structure of detail coefficients. This reduces the tendency of connected thresholding to leave 'spikes' in the estimated segmentation. 'Spikes' are likely to occur when the detail coefficients $d^k_{s,b,e}$, with either $e - b$ or $b - s + 1$ equals to one, survive the thresholding. To control the occurrence of 'spikes', the detail coefficients $d^k_{s,b,e}$ with either $e-b$ or $b-s+1$ less than a constant $m^*$ are set to zero. By setting $m^* = 2$, the spikes can be reduced and the user will have direct control over the minimum length of segments. The final estimator $\tilde{g}^k_{s,b,e}$ of $g^k_{s,b,e}$ is given by

$$\tilde{g}^k_{s,b,e} = \hat{g}^k_{s,b,e} \mathbb{1}\{(b - s + 1) > m^*\} \mathbb{1}\{(e - b) > m^*\}. \tag{4.6}$$

We will see later that the additional unconnected thresholding with $m^* = 2$ gives us better estimates compared to using connected thresholding only.

### 4.3.3 Step 3: Signal Reconstruction

Unlike the original TGUH method of Fryzlewicz (2018), the reconstruction procedure is not conducted by performing the inverse TGUH transform. This is due to the additional unconnected thresholding used in the previous step. If the inverse

TGUH transform is applied directly to the unconnected thresholding results, there may occur a situation where the estimated signal for a segment is not equal to the sample mean of the data in that segment. Since each breakpoint $b$ in the middle of surviving wavelet/detail coefficients $\tilde{g}^k_{s,b,e}$ denote the locations of change-points, we therefore, estimate the piecewise constant signal $f_i$ between two consecutive change-points by the sample mean of all copy number ratio data $r_i$ in that interval.

More formally, let $\mathbb{b} = \{b_l\}$ denote the collection of $b \in \tilde{g}^k_{s,b,e}$ in ascending order where $l = 1, .., N$ and $N$ is the number of estimated change-points. Define $\eta_p = \{0, b_1, b_2, \ldots, b_N, n\}$; $n$ is the length of the copy number ratio data $r_i$. So that, the final estimator $\hat{f}$ of the true function $f$ in Equation (4.4) is defined by

$$\hat{f}_t = \frac{1}{\eta_{p+1} - \eta_p} \sum_{k=\eta_p}^{\eta_{p+1}} r_k \tag{4.7}$$

for $t \in [\eta_p, \eta_{p+1}]$, $p = 1, \ldots, N + 1$.

## 4.4 Simulation Study

To evaluate the performance of the methods, a comparative simulation study was conducted by considering four kinds of test functions which are explained in detail as follows.

1. The first true function is shown in panel A of Figure 4.5 and this pattern is based on some patterns of different segment lengths including both long segments and short segments commonly observed in real data and the aberrations in height/depth varies between 0–4.

2. The second function only includes short segments with various heights which is shown in panel B of Figure 4.5. The aim of simulation using this test function is to assess the ability of the method in estimating short segments.

3. The third test function, which is presented in panel C of Figure 4.5, is an extreme case where there is only a single altered segment with a very short (6 point) width.

4. The fourth type test function is generated by adapting genomic profiles generation scheme proposed by Fridlyand (2004) to obtain more realistic DNA copy number truth with known truth. One thousand simulated copy number patterns (test functions) are generated based on the Circular Binary Segmentation (CBS) fit of a normalised 38 samples lung adenocarsinoma (LA) tumour dataset. We randomly sampled copy number levels from the empirical distribution of segment mean values, where mean values were binned into the intervals less than 0.25 (0 copies), between 0.25, and 0.75 (one copy), between 0.75 and 1.25 (2 copies), between 1.25 and 1.75 (three copies), between 1.75 and 2.25 (four copies), between 2.25 and 2.75 (five copies), between 2.75 and 3.25 (six copies). The length of normal segments (two copies) were assigned by randomly sampling the segment length from the empirical length distribution of copy number levels belong into the $[0.75, 1.25]$ bin. Similarly, the lengths were asigned to the altered segments by sampling from the length distribution for segments with levels outside that bin, without distinguishing among length distributions with different copy numbers. So that we could record the "true breakpoint". Since one of the interests is to know the ability of the method to estimate short segments, for each of the generated data, four short segments were assigned with a length of six data points and a height set to 0.5.

In this simulation study, the aberrations are distinguished into two types:(i) short segments and (ii) long segments, for the purpose of evaluating the ability of the method in estimating both of those types of abberations. The aberrations with length between 6–10 data points are referred as short segments while long segments comprise more than 10 data points. This is based on the window size used in our data (150kb), in which a 1 Mb segment is represented by only 6-7 windows or data points. The height of the short segments is set to 0.5 to represent the typical smallest change that might expected in real data.

One thousand replicates were generated for each of the first, second and third true functions. For each of the true functions considered, two kinds of noise models were used to contaminate those data. The first noise model is i.i.d. Gaussian noise $N(0, \sigma^2)$ and the second is a heavier-tailed noise model that reflects extreme

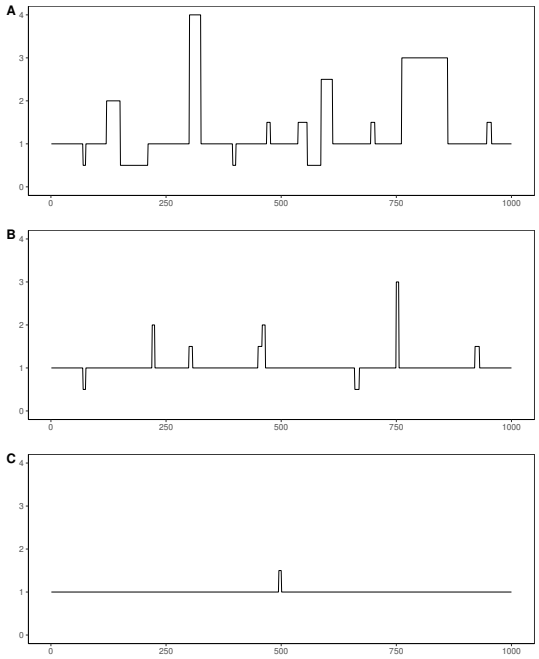


Figure 4.5: The true patterns of copy number alterations, denoted $f$, in simulated examples. **(A)** First true function. The irregular pattern of segment length is based on common patterns observed in real data. **(B)** Second true function, which aims to characterise the proposed method's performance in a case where the underlying true pattern only contains short altered segments. **(C)** Third true function. An extreme case where there is only a short altered segment in the middle of long segment.

observations that often occur in the NGS copy number ratio data. A classical way to generate the second noise model is to use "contaminated normals", where the error distribution is a mixture of two normal distributions (Tukey, 1960). With probability $1 - \alpha$ the error was drawn from a distribution $N(0, \sigma^2)$, and with probability $\alpha$ from $N(0, d^2\sigma^2)$, with $d = 3$ and $\alpha = 0.05$ (Nilsen *et al.*, 2012). The simulations were repeated for $\sigma = 0.1, \ldots, 0.5$ for both noises to obtain a controlled comparison of different levels of noise variance relative to the changes that are wished to be detected in CNA data, which are generally of magnitude 0.5 or 1 (Gusnanto *et al.*, 2012).

As the data generation framework used provides copy number profiles with known truth, the problem of change points detection can be considered as a binary classification problem. Specifically, for each generated profile, the true change-points locations are known. There are several ways that can be used to measure the performance of a binary classification model, each providing different insights into the model's effectiveness. Here are some performance metrics for binary classification that is used in this chapter:

- True positive rate (TPR). The TPR is the proportion of true positive predictions out of all actual positive instances in the dataset. It focuses on the ability of the model to capture all positive instances and is valuable when the cost of false negatives is high.

- False positive rate (FPR). The FPR is the proportion of negative instances that are incorrectly classified as positive by the classifier. A lower FPR indicates that the classifier is better at distinguishing true negatives from false positives.

- Receiver operating characteristic (ROC) curve and its area under the curve (AUC). The ROC curve plots the TPR against the FPR at various discrimination thresholds. The AUC quantifies the overall performance of the classifier, providing a single scalar value that summarizes its ability to distinguish between the two classes. A perfect classifier has an AUC of 1, while a random or no-discrimination classifier has an AUC of 0.5. Generally, the higher the AUC, the better the classifier's ability to distinguish between the classes.

Besides the metrics mentioned above, there are other evaluation metrics such as F1 score which is effective in evaluating the trade-off between false positives and false negatives and the precision-recall curve which focuses specifically on the performance of the positive class (change-points). However, in the context of change-points problem, the ROC curve is used as the focus is to evaluate the performance of each method in estimating change-point with minimal false positives. By using the ROC curve, the trade-off between identifying actual change-points (true positive rate) and the number of false alarms (false positive rate) across various decision thresholds can be obviously illustrated.

In this simulation, For the computation of performance metrics, correctly identified change-points (true positive, TP) are defined as those whose locations are found within two windows and closest to the true change point. If there are two closest change-points detected, one is assigned as TP and the other one as false positive (FP). The remaining change points detected, FP = P − TP, where P denotes positives or the total number of estimated change points $N$, are considered spurious estimates (FP) (Pierre-Jean *et al.*, 2015). The illustration of these definitions is presented in Figure 3.2. Based on this definition, the average of true positive rate (aTPR) and the average false positive rate (aFPR) were computed over 1000 replicates. To assess the ability of the method in estimating short segments, the average true positive rate in estimating short segments (aTPRsh) was also calculated. Moreover, the similarity of the estimates and the true function was measured by calculating average mean squared error (aMSE) over 1000 replicates. The results of these performance metrics are presented in Section 4.4.2.

In Section 4.4.3, to further evaluate the operating characteristics of each method, the Receiver Operating Characteristic (ROC) curve was calculated for each method across different values of $\sigma^2$. The ROC curves are plotted based on the mean TPR and FPR across replicate data sets for each segmentation method. The classification threshold for the ROC curve varied from 4.5 to $-0.1$.

To investigate and compare the performance of each method in estimating each of the change-points of the test functions, in Section 4.4.4, the frequency of a change-point estimated at the correct locations was plotted.

### 4.4.1 Comparative Methods

For all the simulations above, the TGUHm segmentation was performed and compared to the original TGUH method (Fryzlewicz, 2018). To evaluate the practical impact of the constant $m^*$, two values of $m^*$ were considered, $m^* = 1$ and $m^* = 2$. For $m^* = 1$, both TGUHm and TGUH will produce exactly the same results, which is denoted by TGUH1.

Besides the original TGUH method, another TGUH-based segmentation method was also considered. The original TGUH combined with a localised pruning method using the R package breakfast ver 2.2 (Anastasiou *et al.*, 2021) which is denoted by TGUHb.

Several well-known published methods listed below were also considered as competitors.

1. Circular Binary Segmentation (Olshen *et al.*, 2004) using package DNAcopy (Seshan & Olshen, 2020). Circular Binary Segmentation (CBS) is a statistical method used for copy number segmentation in DNA sequences. It is designed to detect regions of the genome with distinct copy number changes. The key idea behind CBS is to iteratively divide the DNA sequence into segments of equal copy number, allowing for both gains and losses of genetic material to be accurately identified.

   The algorithm starts with an initial segment that covers the entire DNA sequence. The initial segment is then tested for any copy number change using a statistical test. CBS typically employs a statistical test, such as the t-test or Wilcoxon rank-sum test, to evaluate if the data points in a segment have a significantly different mean from the neighbouring segment. If the statistical test detects a significant difference, the segment is split into two smaller segments. This process continues iteratively. The algorithm iteratively applies the statistical test to each segment. If a segment is found to have a significant copy number change, it is split into two new segments. The process continues until no further significant changes are detected, or a predefined stopping criterion is reached. The circular aspect of CBS comes into play if the DNA sequence is circular (e.g., a circular chromosome). In such cases, the algorithm ensures that the segmentation process takes

into account the circularity, avoiding potential artefacts at the sequence's endpoints. Once the segmentation process converges, the algorithm outputs a set of segments, each representing a region with a constant copy number. These segments correspond to regions of copy number gain and loss in the DNA sequence.

2. HaarSeg (Ben-Yaacov & Eldar, 2008) using package HaarSeg (Ben-Yaacov & Eldar, 2009). HaarSeg method is a segmentation method based on wavelet denoising principles. HaarSeg identifies statistically significant breakpoints in the data, using the maxima of the Haar wavelet transform, and segments accordingly. The method starts with applying the non-decimated discrete wavelet transform (NDWT) on the input data using the Haar wavelet. A group of detail subbands are chosen from the transform, and then the local maxima in the chosen detail subbands can be identified. After finding the local maxima, an FDR (False Discovery Rate) thresholding procedure is applied to the maxima of each subband. By combining the selected maxima from all subbands a comprehensive list of significant breakpoints in the data can be created to reconstruct the final segmentation result.

3. CumSeg (Muggeo & Adelfio, 2010) using package cumSeg (Muggeo, 2020). The "CumSeg" method is a cumulative approach to time series segmentation, and it can be used to identify segments in time series data with distinct characteristics or trends. Instead of using a fixed threshold or a predefined number of segments, CumSeg determines the number of segments based on the data itself. It does this by cumulatively adding segments until a specified criterion, often related to goodness-of-fit or information criteria, is met.

4. FDRseg (Li *et al.*, 2016) using package FDRSeg (Li & Sieling, 2017). The FDRSeg method is a multiscale segmentation method that effectively controls the false discovery rate (FDR). This means that the number of false jumps is limited proportionally to the number of true jumps, enabling the method to adjust its sensitivity based on the actual number of true jumps. A

non-asymptotic upper bound is provided in the method for the FDR in a Gaussian scenario, allowing proper calibration of FDRSeg's single parameter.

5. CopyNumber (or PCF) method Nilsen *et al.* (2012) using package copynumber (Nilsen *et al.*, 2013). This method employs penalized least squares regression to estimate a piecewise constant fit to the data. In the CopyNumber method, a single penalty parameter $\gamma$ is introduced to control the balance between high sensitivity (minimizing missed true aberrations) and high specificity (reducing false aberrations) which is very critical for all segmentation procedures. Therefore, in the simulation, the CopyNumber method was applied twice, with its main parameter $\gamma$ set to be 12 and 40 as suggested by Nilsen *et al.* (2012) to give different balances between sensitivity and specificity. The results for these two separate analyses are denoted as Copy12 and Copy40, respectively.

### 4.4.2 Simulation Results

Figures 4.6, 4.7, and 4.8 show the result of the simulation study using the first, second, and third true functions. The corresponding quantitative results are presented in the Appendix A. Figure 4.6 indicates that for the basic Gaussian noise TGUHm, TGUH, and TGUH1 outperform the other competitors in terms of estimating both short and long segments by showing the highest aTPRsh and aTPR values for all noise levels but it comes with slightly larger aFPR and aMSE than most of the tested methods. In particular, different $m^*$ values ($m^* = 1$ or $m^* = 2$) do not affect the results much when the noise is standard Gaussian noise. The differences in performance due to these choices are more apparent in the case when the noise comes from the mixture of two normal distributions with different variance; see the right side plots of Figure 4.6. For aTPRsh, all of TGUHm, TGUH, and TGUH1 do not show a significant difference (still the best) but TGUH1 is marginally the worst in terms of aFPR and aMSE. On the other hand, TGUHm and TGUH are much better than TGUH1 for both aFPR and aMSE which indicate that setting $m^*$ equal to two successfully reduces spurious change-points (spikes) which are caused by the occasional extreme outliers. Moreover, compared to TGUH, TGUHm tends to return fewer false positives. This shows that adding

the unconnected thresholding is preferable as it allows us to use the $m^*$ to control the minimum segment width compared to using connected thresholding alone.

Figures 4.7 and 4.8 show the performance metric of the tested methods when the simulated data only contain short segments. TGUHm still performs very well by showing excellent results in terms of aTPRsh and aMSE without excess false positives. Besides TGUHm, the TGUH1, TGUH, Copy12, and FDRSeg methods do well in terms of estimating short segments. CBS also performs well in estimating short segments for the standard Gaussian noise but it is not as good as those methods when the noise is the Gaussian mixture noise. It also shows poor performance in terms of aTPRsh when the true function only contains an isolated short segment in the middle of a very long segment as shown in Figure 4.8. The FDRSeg method, while showing good performance for short test signals, it is sensitive to occasional extremely noisy observations. This reflects in larger aFPR and aMSE for Gaussian mixture noise.

Figure 4.9 show the performance metric of the tested methods based on the fourth type of test function. Unlike the three previous simulations which only consider one fixed test function, in the fourth type of simulation, there were 1000 different test functions. This condition enables us to see the performance of the tested method in more general. Even in this setting, TGUHm still performs very well in terms of aTPRsh and aMSE with relatively lower aFPR which is similar to the results in the previous simulations.

Compared to the other methods, based on all of the simulation models considered, CumSeg and Copy40 tend to miss some change-points and fail to estimate short segments even for low level of noise contamination. This also indicates that the performance of CopyNumber method is sensitive to the selection of $\gamma$. Therefore, in practice, it may be necessary to test a number of $\gamma$ values to find the optimal one.

### 4.4.3 Receiver Operating Characteristic of the Simulation

To show in a graphical way the connection/trade-off between sensitivity and specificity for every possible cut-off, ROC curves of each of the simulations across different noise levels were considered and the corresponding area under the curve

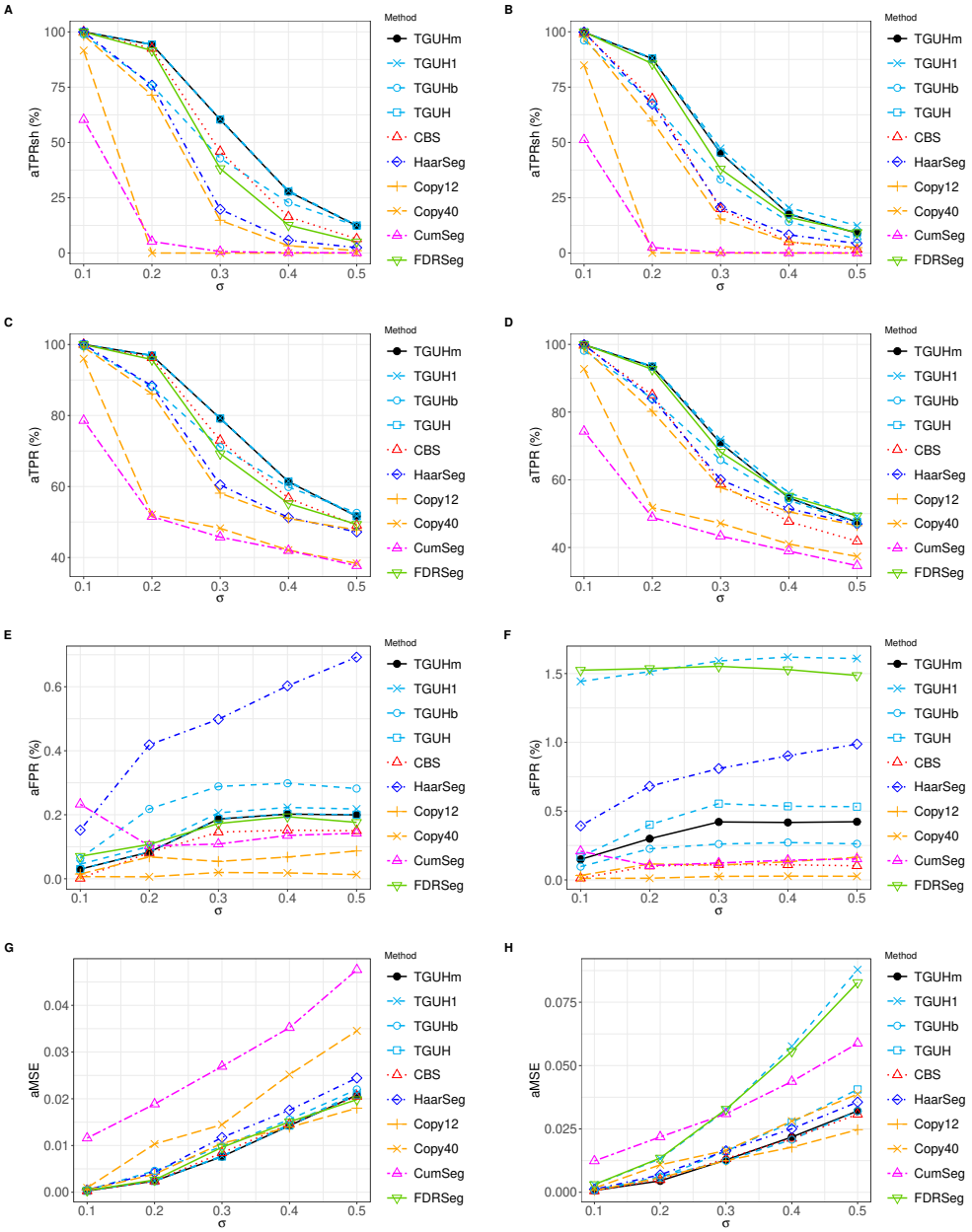


Figure 4.6: Performance metrics for 1000 replicates of the first test function (see panel **A** of Figure 4.5). **(A) (B)** Average of true positive rate in estimating change-points that corresponds to short segments (aTPRsh). **(C) (D)** Average true positive rate (aTPR). **(E) (F)** Average of false positive rate (aFPR). **(G) (H)** Average of mean-square error (aMSE) of the estimated piecewise constant signal to the true function. The left column (panels **A**,**C**, **E**, and **F**) show results for i.i.d. Gaussian noise $N(0, \sigma^2)$, while the right column (panels **B**,**D**, **F**, and **H**) show results for noise from a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$). For a quick reminder, TGUH1 denotes both TGUH and TGUHm method with $m^* = 1$ while TGUHm and TGUH denote TGUHm and TGUH method with $m^* = 2$, respectively. TGUHb denotes TGUH method with a localised pruning algorithm. Copy12 and Copy40 denote CopyNumber method with $\gamma$ parameter equal to 12 and 40, respectively.

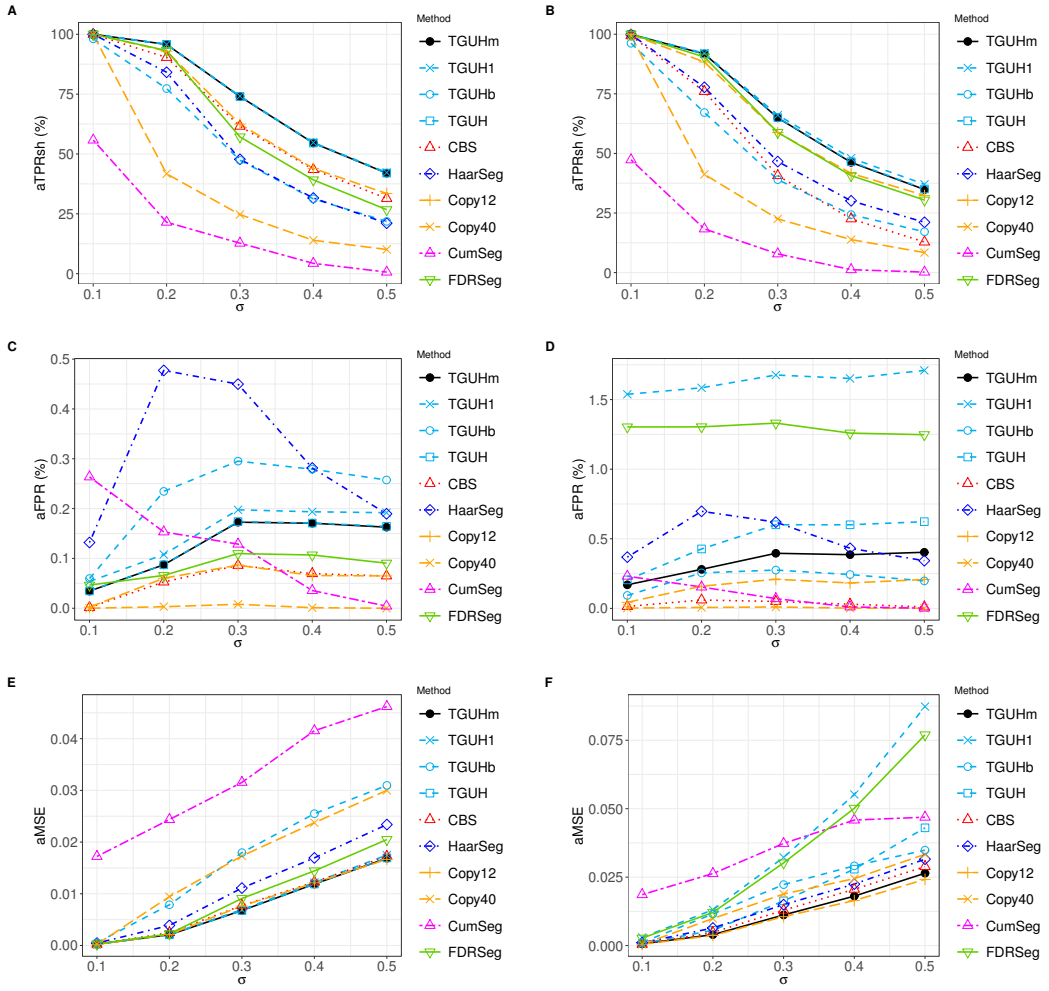


Figure 4.7: Performance metrics for 1000 replicates of the second test function (see panel **B** of Figure 4.5). **(A) (B)** Average of true positive rate in estimating change-points that corresponds to short segments (aTPRsh). **(C) (D)** Average of false positive rate (aFPR). **(E) (F)** Average of mean-square error (aMSE) of the estimated piecewise constant signal to the true function. The left column (panels **A**,**C**, and **E**) show results for i.i.d. Gaussian noise $N(0, \sigma^2)$, while the right column (panels **B**,**D**, and **F**) show results for noise from a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$). The aTPR results are omitted as the simulated data only contains short segments. For a quick reminder, TGUH1 denotes both TGUH and TGUHm method with $m^* = 1$ while TGUHm and TGUH denote TGUHm and TGUH method with $m^* = 2$, respectively. TGUHb denotes TGUH method with a localised pruning algorithm. Copy12 and Copy40 denote CopyNumber method with $\gamma$ parameter equal to 12 and 40, respectively.

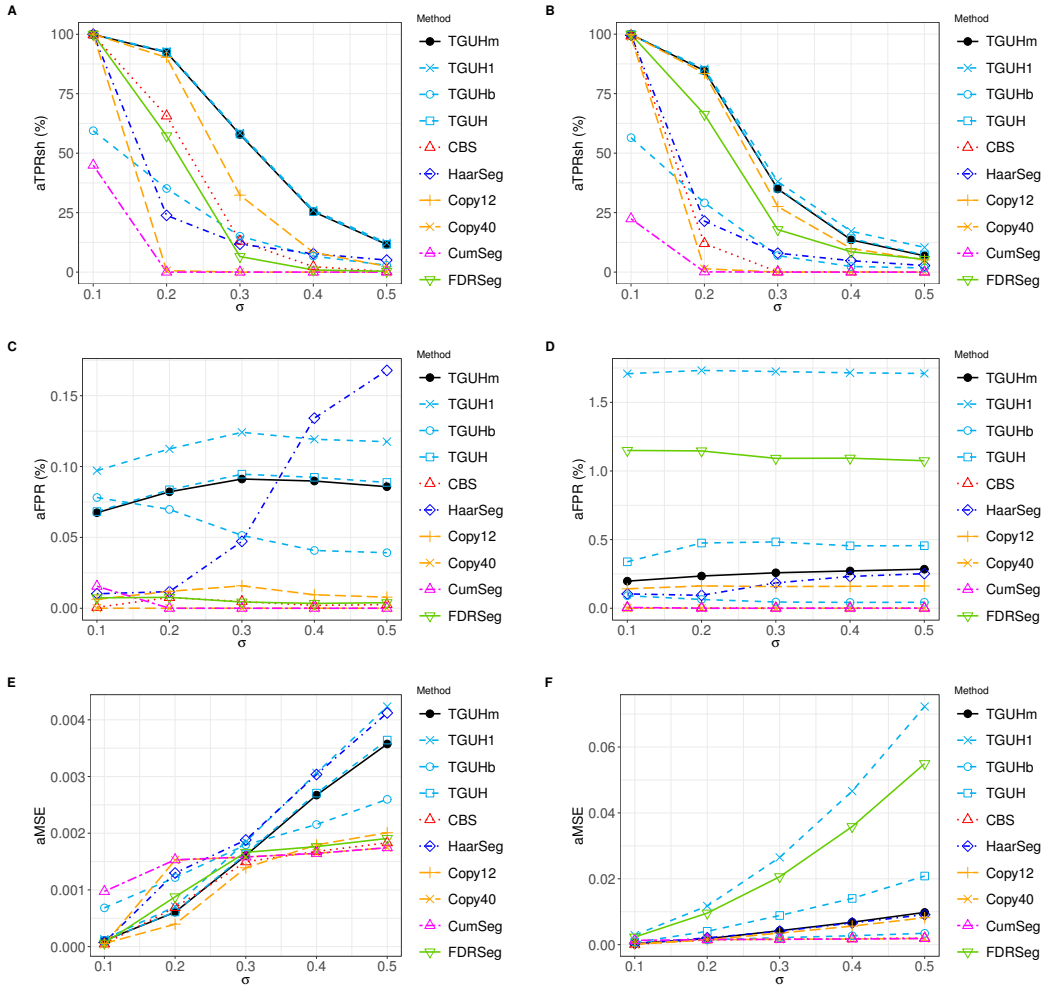


Figure 4.8: Performance metrics for 1000 replicates of the third test function (see panel **C** of Figure 4.5). **(A) (B)** Average of true positive rate in estimating change-points that correspond to short segments (aTPRsh). **(C) (D)** Average of false positive rate (aFPR). **(E) (F)** Average of mean-square error (aMSE) of the estimated piecewise constant signal to the true function. The left column (panels **A**,**C**, and **E**) show results for i.i.d. Gaussian noise $N(0, \sigma^2)$, while the right column (panels **B**,**D**, and **F**) show results for noise from a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$). The aTPR results are omitted as the simulated data only contains an isolated short segment. For a quick reminder, TGUH1 denotes both TGUH and TGUHm method with $m^* = 1$ while TGUHm and TGUH denote TGUHm and TGUH method with $m^* = 2$, respectively. TGUHb denotes TGUH method with a localised pruning algorithm. Copy12 and Copy40 denote CopyNumber method with $\gamma$ parameter equal to 12 and 40, respectively.

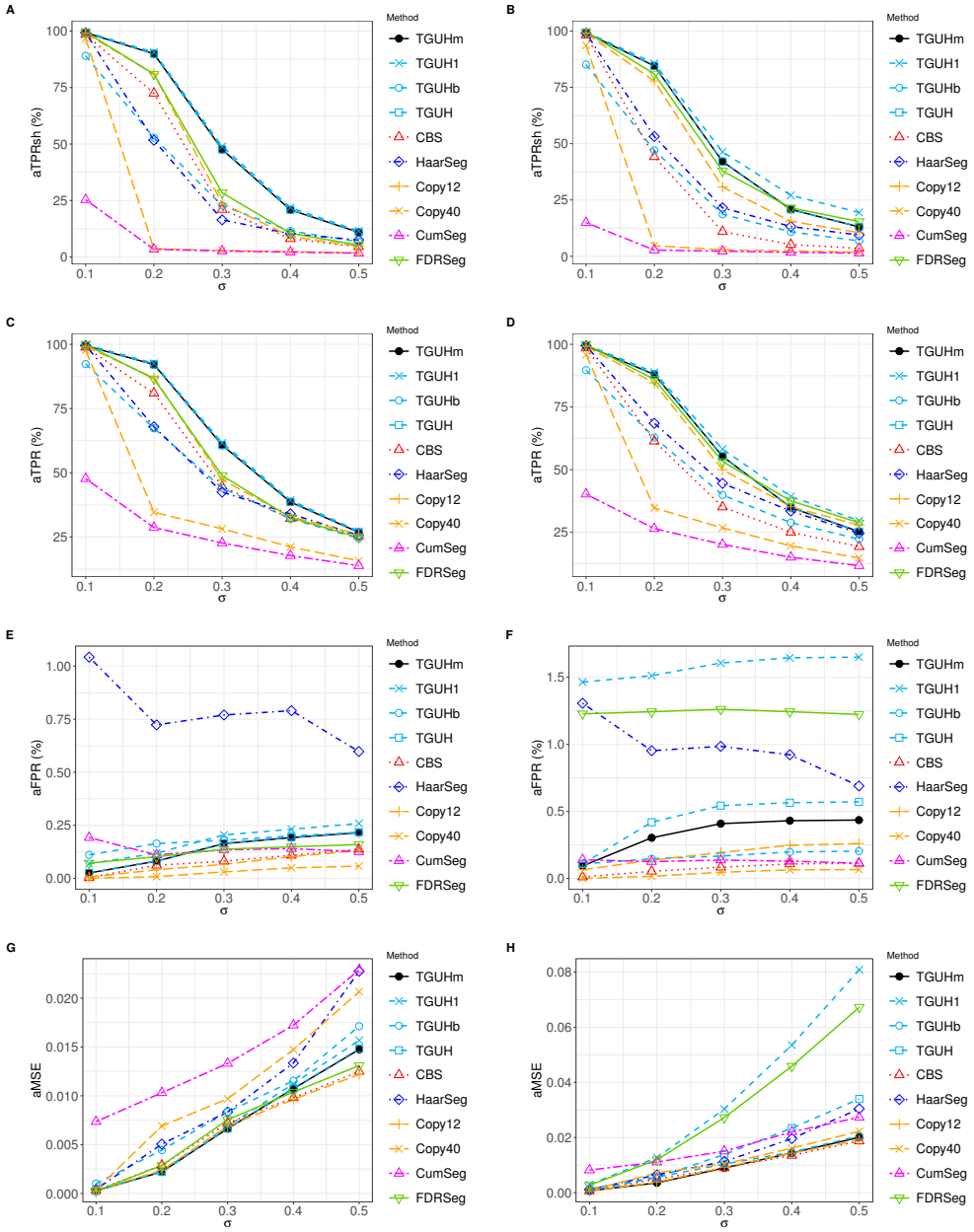


Figure 4.9: Performance metrics for 1000 replicates of the fourth test function as explained in Section 4.4. **(A) (B)** Average of true positive rate in estimating change-points that corresponds to short segments (aTPRsh). **(C) (D)** Average of false positive rate (aFPR). **(E) (F)** Average of mean-square error (aMSE) of the estimated piecewise constant signal to the true function. The left column (panels **A**,**C**, and **E**) show results for i.i.d. Gaussian noise $N(0,\sigma^2)$, while the right column (panels **B**,**D**, and **F**) show results for noise from a mixture of two Gaussian distributions $0.95\times N(0,\sigma^2)+0.05\times N(0,3\sigma^2)$). The aTPR results are omitted as the simulated data only contains an isolated short segment. For a quick reminder, TGUH1 denotes both TGUH and TGUHm method with $m^*=1$ while TGUHm and TGUH denote TGUHm and TGUH method with $m^*=2$, respectively. TGUHb denotes TGUH method with a localised pruning algorithm. Copy12 and Copy40 denote CopyNumber method with $\gamma$ parameter equal to 12 and 40, respectively.

(AUC) is reported in Figure 4.10. For both noise types used in the simulation, it is quite clear that performance in terms of AUC severely deteriorates when the noise level increases.

The results in panels E and F in Figure 4.10 show that, in a very extreme case where there is only a short altered segment in the underlying test function, the performance of the CBS, Copy40, and CumSeg methods are very poor. Those methods tend to produce a long flat segment and are unable to estimate the short segment even for low noise levels. This is reflected by their AUC scores that drastically drop to 0.5 for noise level with standard deviation $\sigma$ greater than 0.1.

Figure 4.10 also indicates that TGUHm, TGUH1, and TGUH has better AUC than the other methods in most of the noise levels considered. Their results almost overlap. This is due to the number of false positive that corresponds to spikes being very low compared to the number of negative cases. To avoid the issues caused by this condition (imbalanced dataset), Figure 4.11 shows partial AUC for early retrieval area ($FP < 20$). From the results shown in Figure 4.11, it is quite clear that for acceptable FP ($FP < 20$), TGUHm method still provides excellent results over all the noise levels for both noise types considered. For simulation using the standard Gaussian noise, the results of TGUHm, TGUH1, and TGUH are very close but for the heavier-tailed noise type that caused extreme outliers in the data, TGUHm shows a significant improvement over the original TGUH (TGUH1, TGUH) method (see right side column of Figure 4.11). This indicates that TGUHm method is competent to reduce spurious change-points commonly found in the original TGUH method caused by extreme outliers.

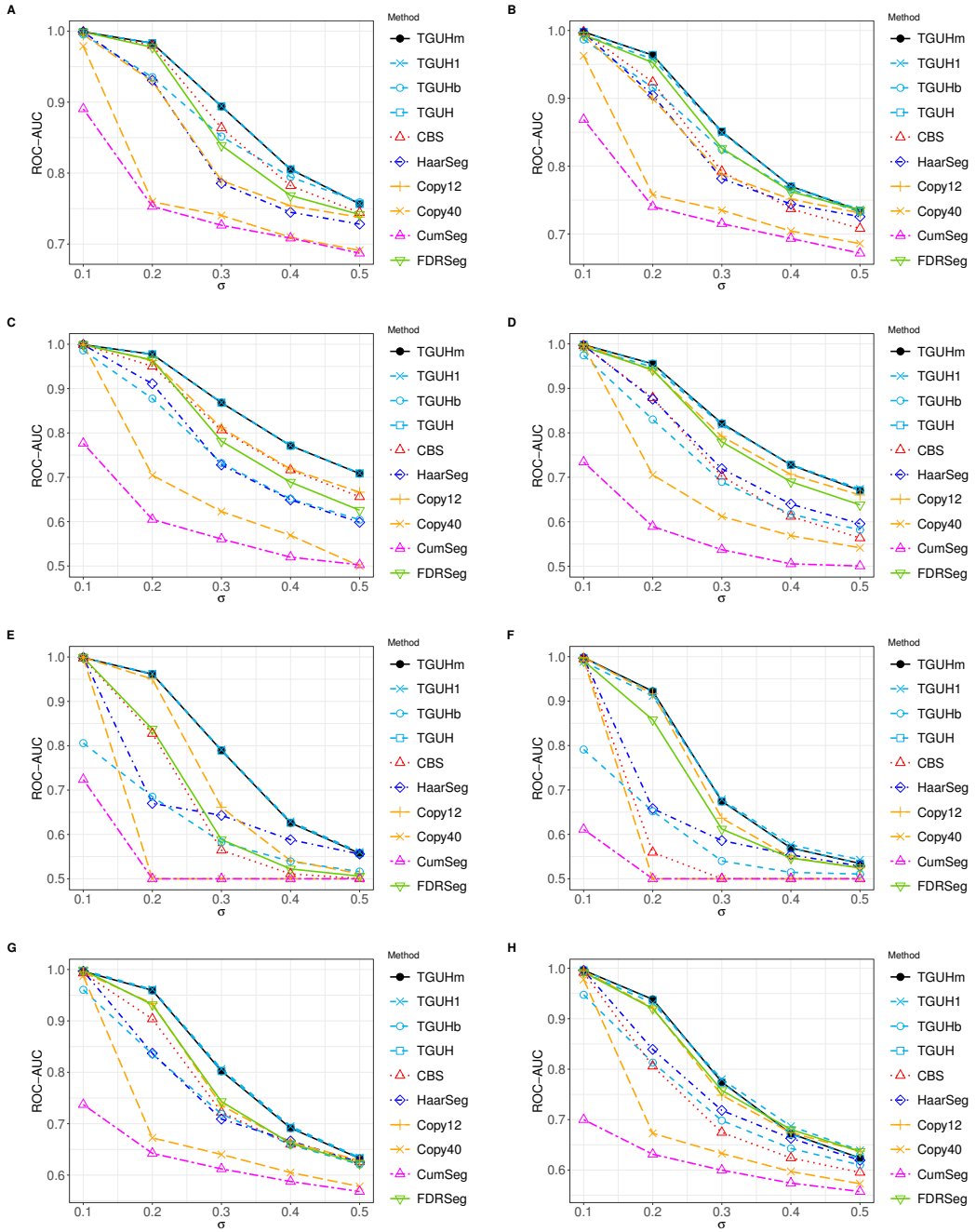


Figure 4.10: AUC of ROC curve of the methods applied to the first, second, and third test functions described in Section 4.4. The left column (panels **A**,**C**, and **E**) show results for i.i.d. Gaussian noise $N(0, \sigma^2)$, while the right column (panels **B**,**D**, and **F**) show results for noise from a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$). The first (panels **A** and **B**), second (panels **C** and **D**), third (panels **E** and **F**), and fourth (panels **G** and **H**) row correspond to the first, second, third, and fourth test function presented in Figure 4.5, respectively. For a quick reminder, TGUH1 denotes both TGUH and TGUHm method with $m^* = 1$ while TGUHm and TGUH denote TGUHm and TGUH method with $m^* = 2$, respectively. TGUHb denotes TGUH method with a localised pruning algorithm. Copy12 and Copy40 denote CopyNumber method with $\gamma$ parameter equal to 12 and 40, respectively.

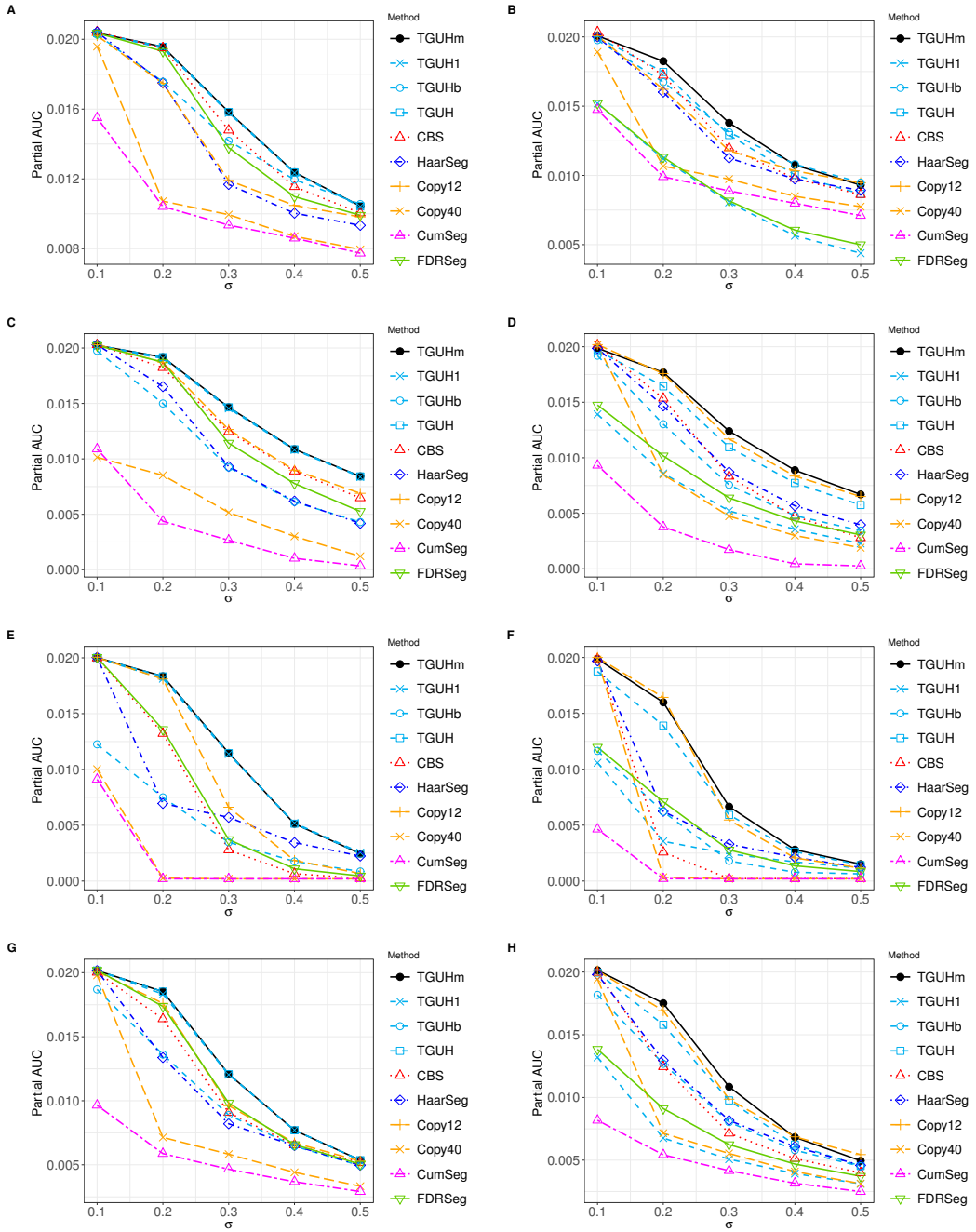


Figure 4.11: Partial AUC for $FP < 20$ of ROC curve of the methods applied to the test functions described in Section 4.4. The left column (panels **A**,**C**, **E**, and **G**) show results for i.i.d. Gaussian noise $N(0, \sigma^2)$, while the right column (panels **B**,**D**, **F**, and **H**) show results for noise from a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$). The first until fourth row correspond to the first, second, third, and fourth test function described in Section 4.4, respectively. For a quick reminder, TGUH1 denotes both TGUH and TGUHm method with $m^* = 1$ while TGUHm and TGUH denote TGUHm and TGUH method with $m^* = 2$, respectively. TGUHb denotes TGUH method with a localised pruning algorithm. Copy12 and Copy40 denote CopyNumber method with $\gamma$ parameter equal to 12 and 40, respectively.

### 4.4.4 Proportion of Times a Change-point is Estimated

To investigate the performance of each method in estimating the correct location of each change-point, Figures 4.12, 4.13, and 4.13 show the proportion of times (from 1000 simulated datasets) that each method detects a change-point at each location along the sequence base on test function as shown in Figure 4.5. Here, the results shown are only for simulated data contaminated with noise from a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ where $\sigma = 0.3$; results for the remaining results with basic Gaussian noise can be found in Appendix B. Since the results of TGUHm and TGUH almost overlap, the resultsof TGUHm and TGUH were plotted as one line.

Based on Figure 4.12, the proposed method, TGUHm method, has the highest sensitivity in terms of detecting short segments while still showing a relatively good performance in estimating long segments. This superiority of TGUHm method in estimating short segments is seen clearer in 4.13, and 4.13. Most of the methods have narrow 'peaks' in the location of the true changes, which indicate the ability of the methods to estimate change-point exactly at the true location over 1000 iteration. But careful inspection shows that the HaarSeg method has small peaks near the true change-points. This shows the tendency of HaarSeg to produce spurious change-points near the true change-point locations. This is commonly found in Haar wavelet-based methods and is its main weakness which has successfully been overcome by the proposed method.

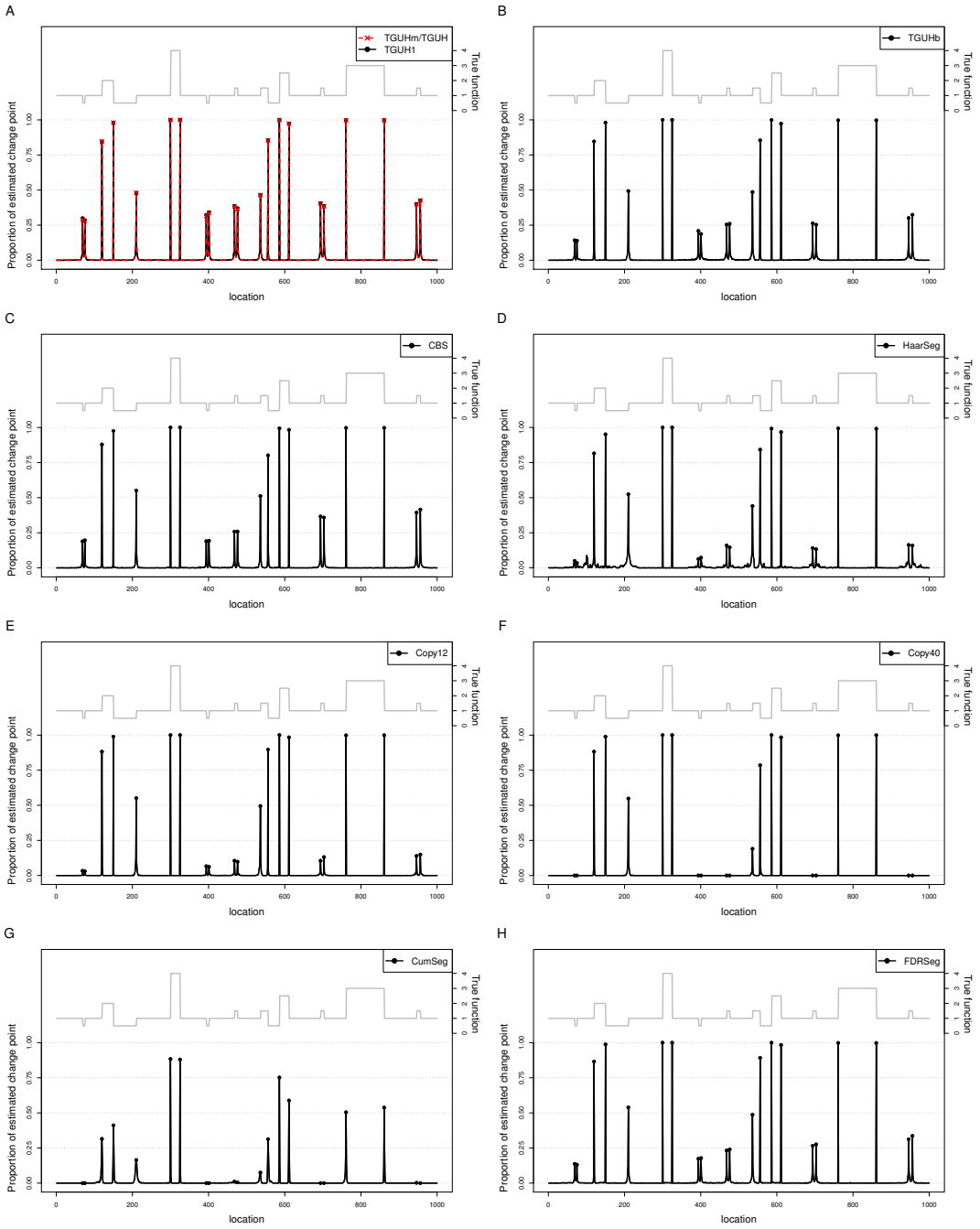


Figure 4.12: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.3^2$. The dots denote the proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **A** of Figure 4.5 for ease of reference. The left and right vertical axis show the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

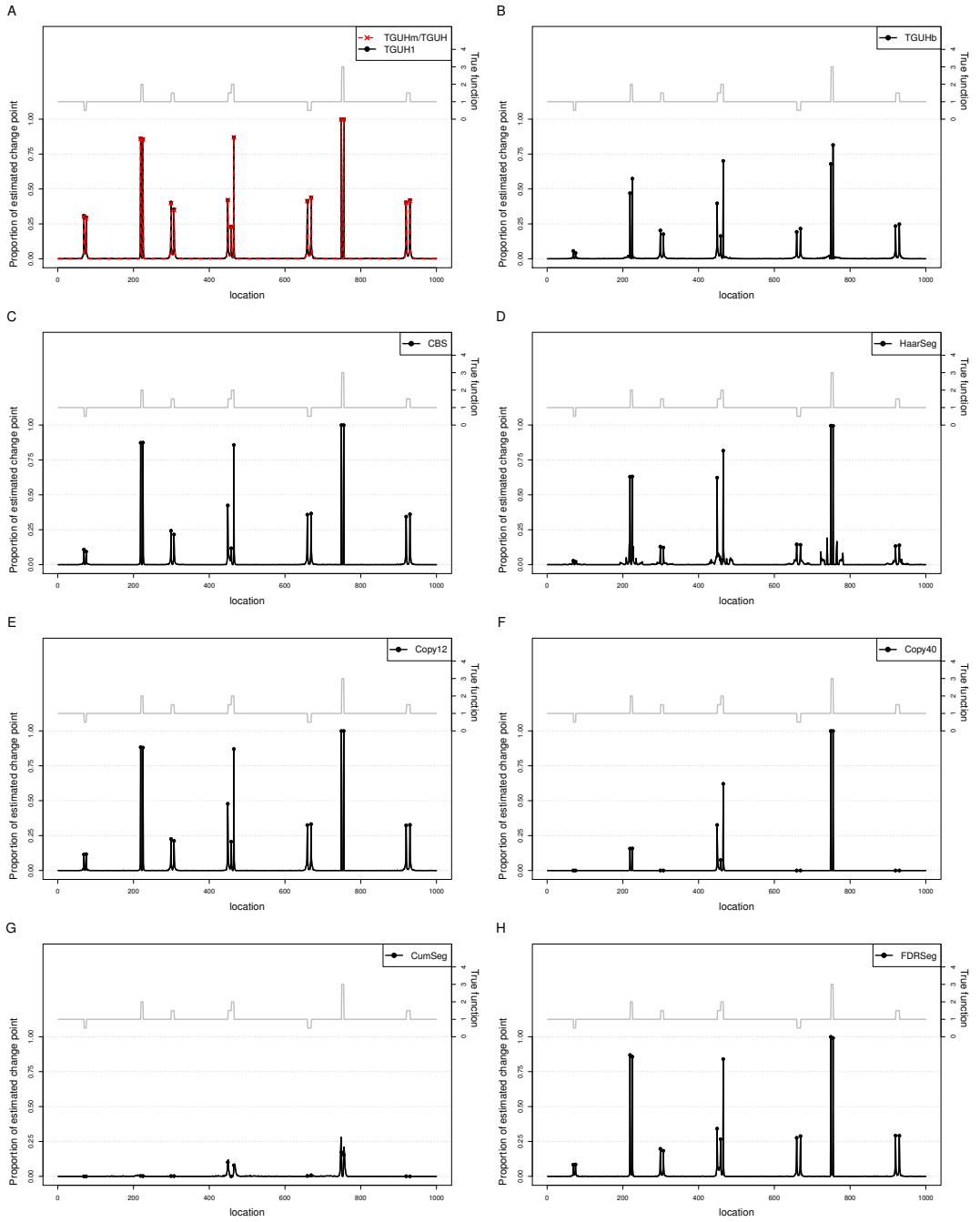


Figure 4.13: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.3^2$. The dots denote the proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **B** of Figure 4.5 for ease of reference. The left and right vertical axis show the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

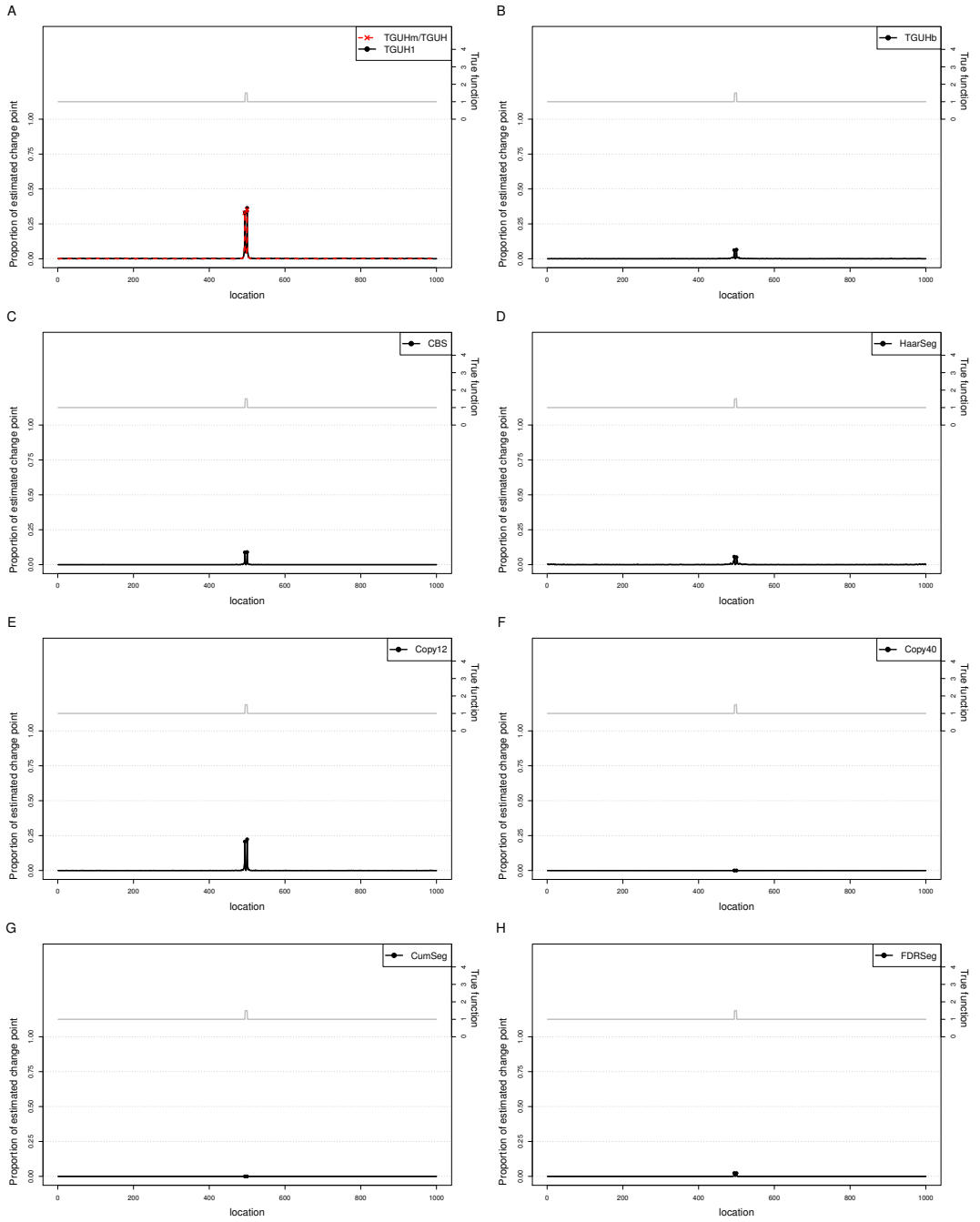


Figure 4.14: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.3^2$. The dots denote the proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **C** of Figure 4.5 for ease of reference. The left and right vertical axis show the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

### 4.4.5 Comparison of TGUHm Segmentation with Various $m^*$ Values

To see the effect of the choice of $m^*$ value in TGUHm segmentation, the TGUHm method was performed with several small $m^*$ values to the simulated data; $m^* = 1, 2, 3, 4, 5, 8, 10$. As a comparison, the results for the basic TGUH method with $m^*$ is equal to one and two were also shown. For simplicity, TGUH1 denotes both of the results of the TGUH and TGUHm methods with $m^* = 1$ since their results are almost the same. The TGUH method with $m^* = 2$ is denoted as TGUH2 while TGUHm2, TGUHm3, TGUHm4, and TGUHm5 denote TGUHm method with $m^*$ value equal to 2, 3, 4, and 5, respectively. TGUHm8 and TGUHm10 denote TGUHm method results where $m^* = 8$ and 10, respectively, these choices are used to see the performance of the proposed method in estimating short segment when $m^*$ is greater than the shortest altered segment length in the underlying data.

Figure 4.15– 4.17 show a performance metrics plot for 1000 replicates of the true functions. Those figures indicate that the differences in performance due to the choice of $m^*$ are more apparent in the case when the noise comes from the mixture of two normal distributions with different variances. The most significant improvement appeared on aFPR and aMSE between $m^* = 1$ and $m^* = 2$ while for larger $m^*$ values, the improvement is not significant compare to it. This motivates the setting of $m^* = 2$ as default and indicates that $m^* = 2$ significantly reduces the occurrences of single-point spikes. Especially, from Figure 4.17, if $m^*$ is larger than the shortest true segment length ($m^* > 6$, TGUH8 and TGUH10), the proposed method was almost unable to estimate the short segment which confirms the ability of $m^*$ in controlling the minimum estimated segment length.

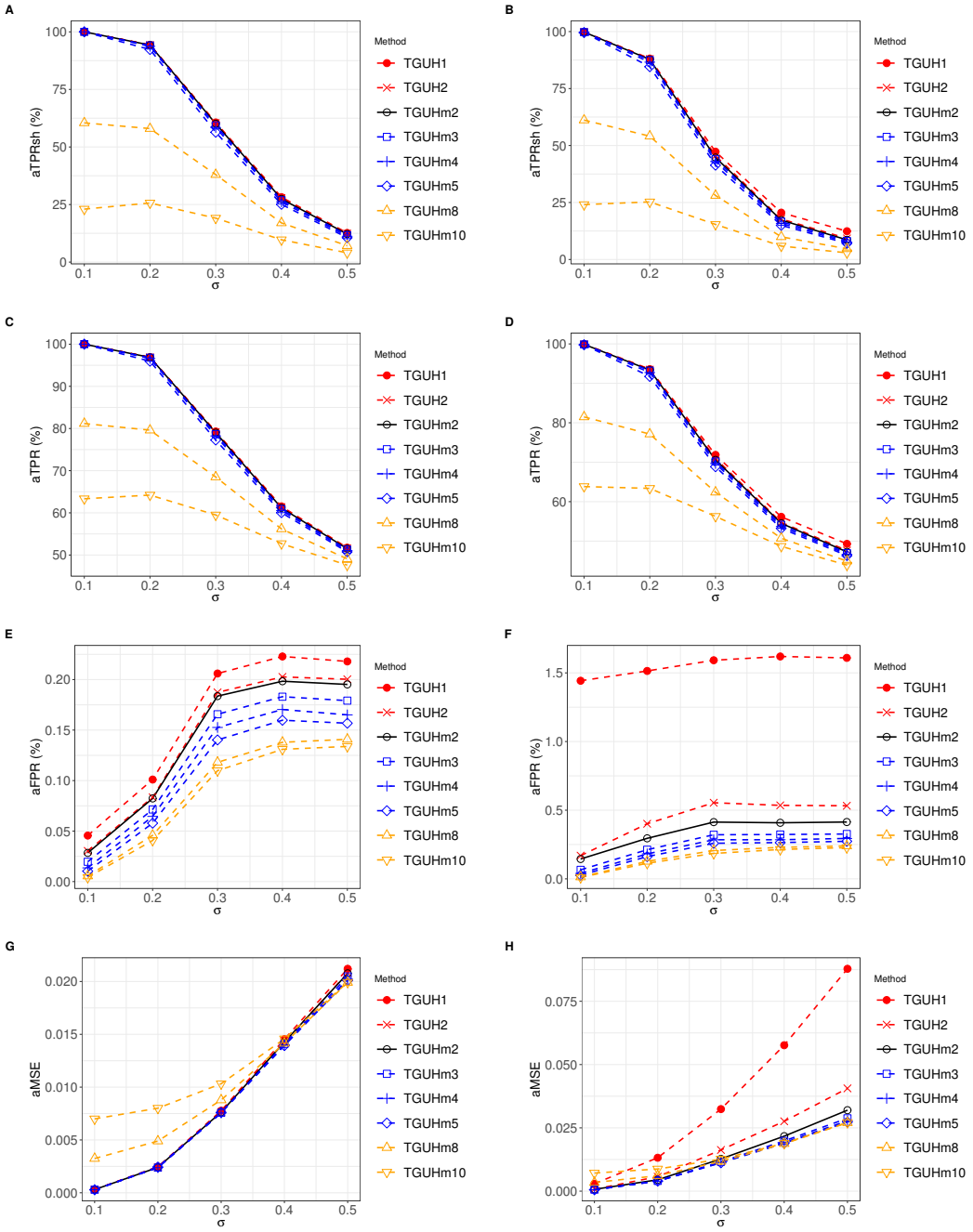


Figure 4.15: Performance metrics of TGUHm method with various $m^*$ values for 1000 replicates of the first test function (see panel **A** of Figure 4.5). **(A) (B)** Average of true positive rate in estimating change-points that corresponds to short segments (aTPRsh). **(C) (D)** Average of false positive rate (aFPR). **(E) (F)** Average of mean-square error (aMSE) of the estimated piecewise constant signal to the true function. The left column (panels **A**,**C**, and **E**) show results for i.i.d. Gaussian noise $N(0, \sigma^2)$, while the right column (panels **B**,**D**, and **F**) show results for noise from a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$). The aTPR results are omitted as the simulated data only contains an isolated short segment. For a quick reminder, TGUH1 denotes the results of both the basic TGUH method and TGUHm method with $m^* = 1$. TGUH2 denotes the basic TGUH method with $m^* = 2$. TGUHm2, TGUHm3, TGUHm4 and TGUHm5 denote TGUHm method with $m^*$ value equal to 2, 3, 4, and 5, respectively.

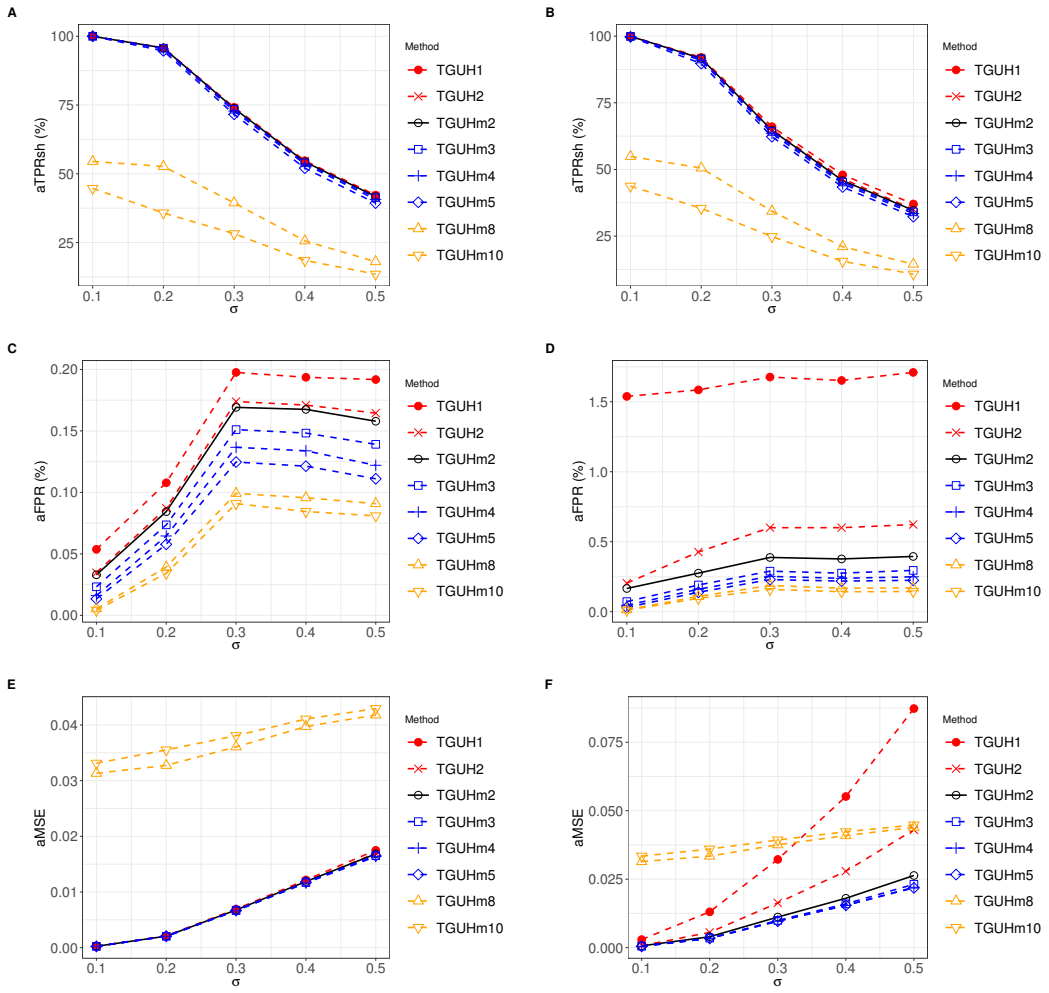


Figure 4.16: Performance metrics of TGUHm method with various $m^*$ values for 1000 replicates of the second test function (see panel **B** of Figure 4.5). **(A)** **(B)** Average of true positive rate in estimating change-points that corresponds to short segments (aTPRsh). **(C)** **(D)** Average of false positive rate (aFPR). **(E)** **(F)** Average of mean-square error (aMSE) of the estimated piecewise constant signal to the true function. The left column (panels **A**,**C**, and **E**) show results for i.i.d. Gaussian noise $N(0, \sigma^2)$, while the right column (panels **B**,**D**, and **F**) show results for noise from a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$). The aTPR results are omitted as the simulated data only contains short segments. For a quick reminder, TGUH1 denotes the results of both the basic TGUH method and TGUHm method with $m^* = 1$. TGUH2 denotes the basic TGUH method with $m^* = 2$. TGUHm2, TGUHm3, TGUHm4 and TGUHm5 denote TGUHm method with $m^*$ value equal to 2, 3, 4, and 5, respectively.

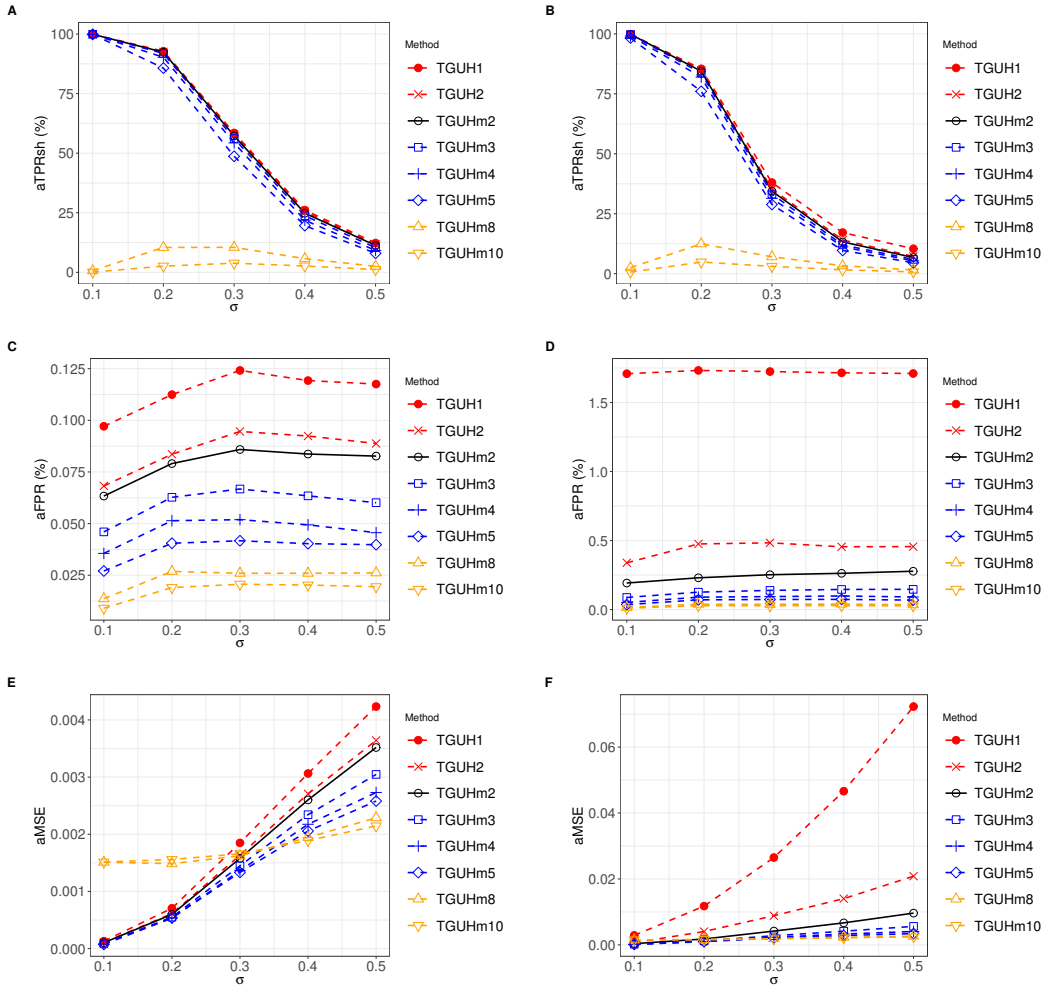


Figure 4.17: Performance metrics of TGUHm method with various $m^*$ values for 1000 replicates of the third test function (see panel **C** of Figure 4.5). **(A)** **(B)** Average of true positive rate in estimating change-points that corresponds to short segments (aTPRsh). **(C)** **(D)** Average of false positive rate (aFPR). **(E)** **(F)** Average of mean-square error (aMSE) of the estimated piecewise constant signal to the true function. The left column (panels **A**,**C**, and **E**) show results for i.i.d. Gaussian noise $N(0, \sigma^2)$, while the right column (panels **B**,**D**, and **F**) show results for noise from a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$). The aTPR results are omitted as the simulated data only contains short segments. For a quick reminder, TGUH1 denotes the results of both the basic TGUH method and TGUHm method with $m^* = 1$. TGUH2 denotes the basic TGUH method with $m^* = 2$. TGUHm2, TGUHm3, TGUHm4 and TGUHm5 denote TGUHm method with $m^*$ value equal to 2, 3, 4, and 5, respectively.

### 4.4.6 Comparison of TGUH-based Methods

In the TGUHm method, a two-stage thresholding procedure is applied to replace the basic connected thresholding used in the original TGUH method. But in practice, there are several options for thresholding techniques that can be applied in the denoising stage of the TGUH method. This section introduces three other possible TGUH-based segmentation methods and presents a simple simulation study to compare their performances.

The first method is the UTGUH method, a TGUH-based method that uses only the unconnected thresholding technique to set to zero all the coefficients below the threshold $\lambda$. The second method is the UTGUHmean method. This method is similar to the UTGUH method, but instead of using the inverse TGUH transform, it takes the mean of the data between two consecutive change-points to construct the segmentation result, as in the TGUHm method. The third method is the TGUHb method, a segmentation method that combines the basic TGUH method with the localised pruning algorithm (Cho & Kirch, 2021). In the simulation, these three TGUH-based methods were compared to the basic TGUH method and the TGUHm method.

Below is the list of five TGUH-based methods considered in the simulation.

1. TGUH (Basic TGUH method (TGUH transform - **connected** thresholding - inverse TGUH)

2. TGUHm (TGUH transform - two-stage thresholding (**connected** thresholding to delete all the detail coefficients lower than the threshold value $\lambda$ - **unconnected** thresholding to delete all detail coefficients which one of its wings is less than a $\beta$ parameter) - reconstruct the result using mean of the data between two consecutive estimated change-points)

3. UTGUH (TGUH transform - **unconnected** thresholding - inverse TGUH transform)

4. UTGUHmean (TGUH transform - **unconnected** thresholding - reconstruct the result using mean of the data between two consecutive estimated change-points)

5. TGUHb (Basic TGUH method combined with a localized pruning algorithm (Cho & Kirch, 2021))

The constant $m^*$ was set equal to 2 for all of the above methods to control the minimum segment length. Those five methods are also compared with the TGUH method using $m^* = 1$ which is denoted as TGUH1.

The simulation study was conducted using the first test function as explained in section 4.4 (please see the top panel of Figure 4.5 for the plot of the test function used). Same as in the previous simulations, 1000 replicates were generated for each of the true functions and two kinds of noise models were used to contaminate the test function. The first noise model is i.i.d. Gaussian noise $N(0, \sigma^2)$ and the second is a heavier-tailed noise model that reflects extreme observations that often occur in NGS copy number ratio data. Then, the simulations were repeated for $\sigma = 0.1, \ldots, 0.5$ for both noise.

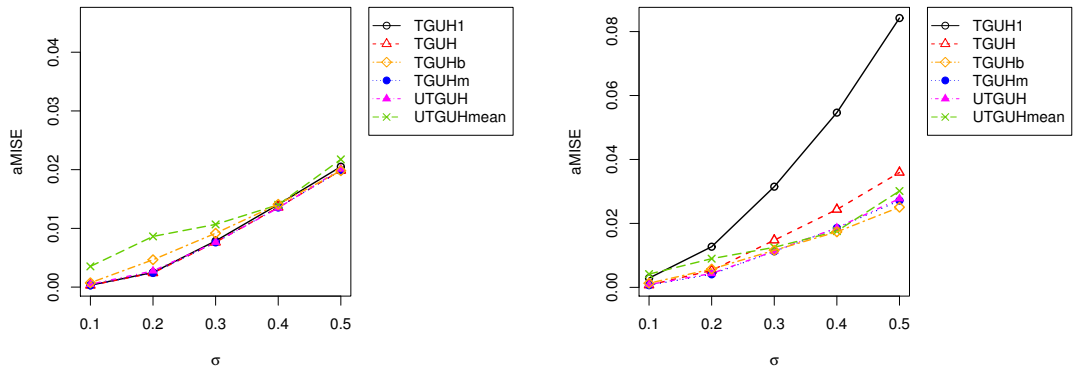


Figure 4.18: Plot of average mean-square error (aMSE) of the estimated piecewise constant signal to that of the signals estimated using the true change points.

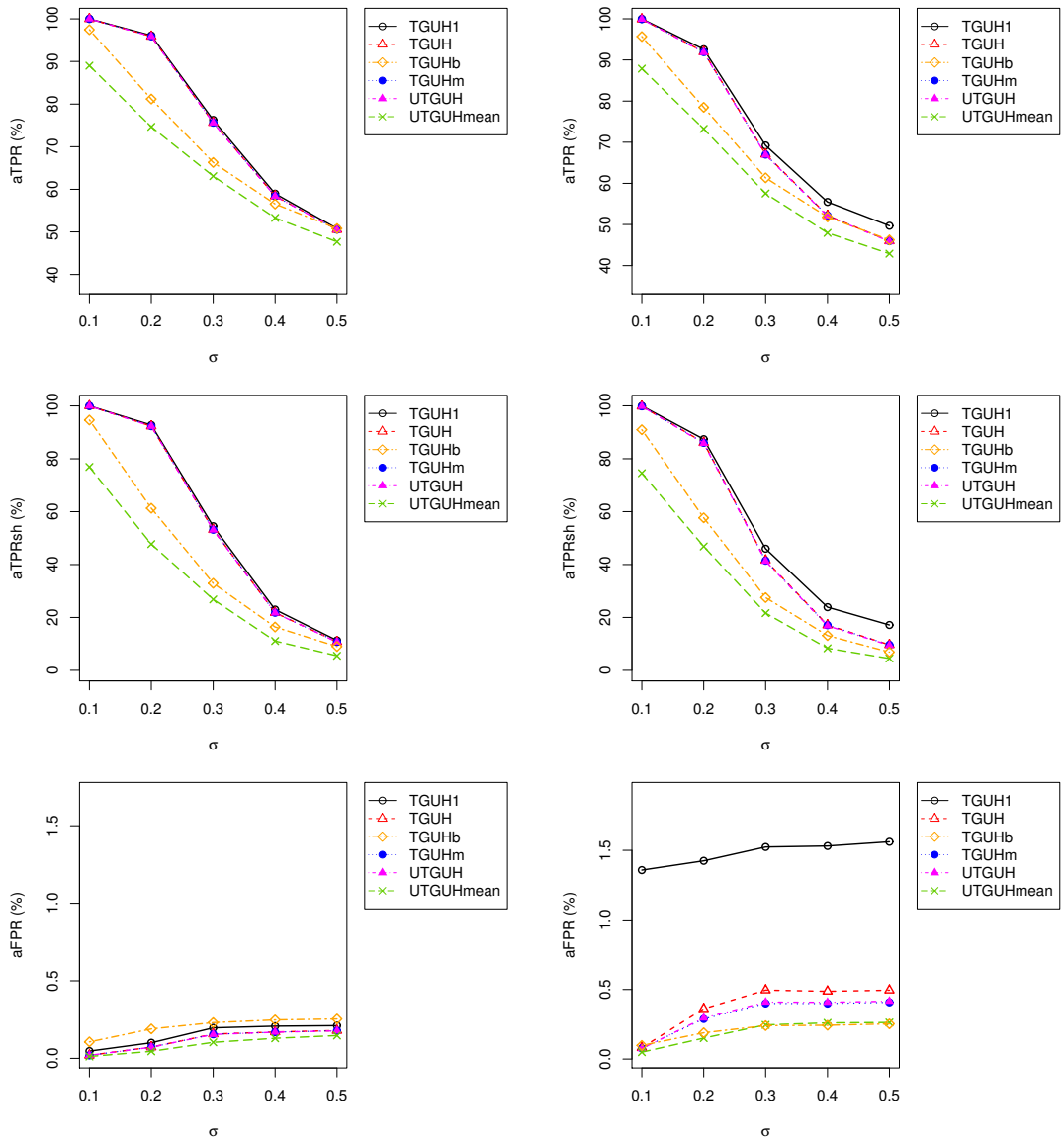


Figure 4.19: Average true positive rate (aTPR; first row). Average true positive rate in estimating change-points that correspond to short segments (aTPR.sh; second row), false positive rate (aFPR; third row) in estimating correct change-points over 1000 replicates of TGUH-based methods. The left and right side plot corresponds to the noise distribution used to contaminate the simulated data (left: i.i.d Gaussian noise $N(0, \sigma^2)$, right: a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$).

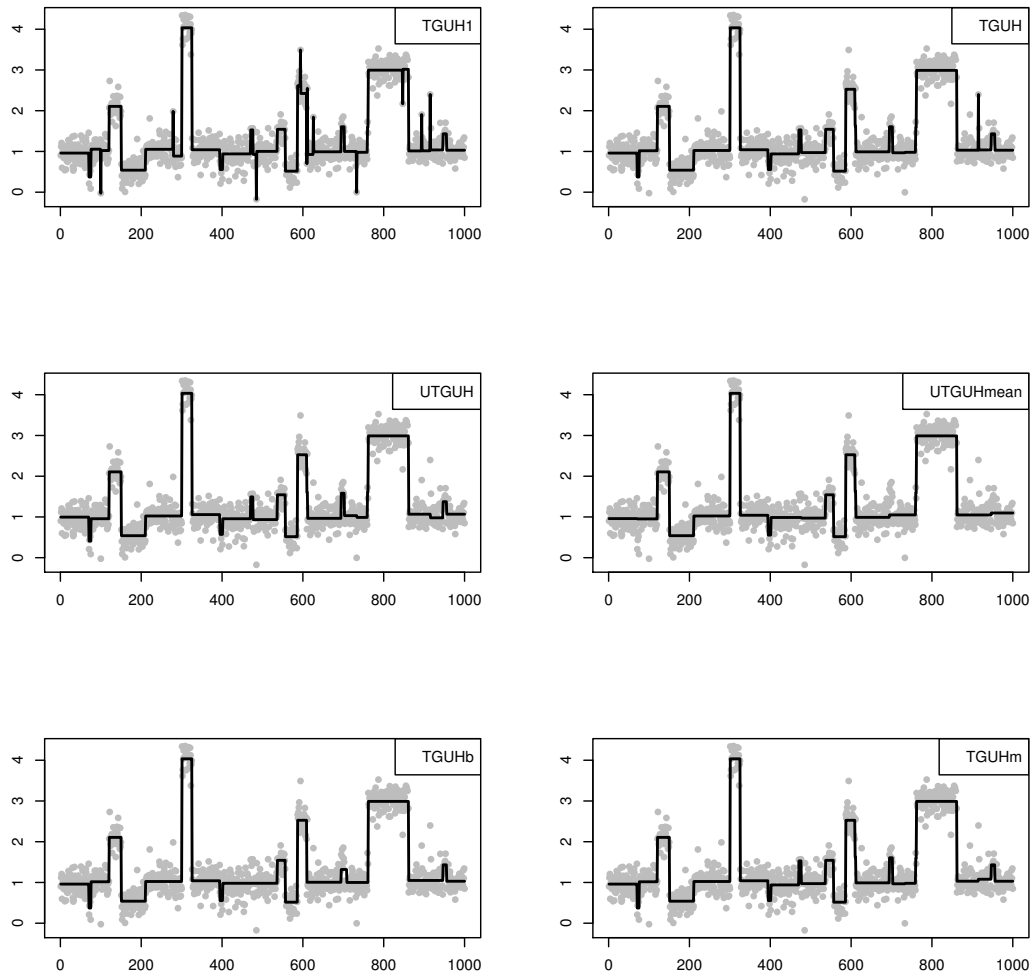


Figure 4.20: Example of TGUH1, TGUH, UTGUH, UTGUHmean, TGUHb, and TGUHm estimates corresponds to the first test function. Noise is a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2))$ with $\sigma = 0.2$.

Figure 4.19 shows the results of the simulation. Among the TGUH-based methods, TGUH1, TGUH, TGUHm, and UTGUH almost overlap and give the best results in terms of aMSE, aTPRsh, and aFPR for simulated data contaminated with i.i.d Gaussian noise, while for the mixture Gaussian noise, TGUHm outperforms the others.

Furthermore, UTGUHmean performance is very poor in terms of aTPRsh.

Its ability to estimate short segments (aTPRsh) is far below the other TGUH methods. Figure 4.20 shows that UTGUHmean is the only one that could not estimate short segment properly even the noise is relatively low ($\sigma = 0.2$). The aTPR and aTPR.sh of TGUHb are also low compared to the other methods, as it tends to over prune change-points. The UTGUH and TGUHm methods, on the other hand, perform well in terms of estimating change-points in long and short segments without significantly harming or increasing the AFPR.

Based on the result of the simulation above, below is the table of rank of each method evaluated.

Table 4.1: Table of the average ranking of performance measurement for simulation using for i.i.d. Gaussian noise $N(0, \sigma^2)$ over all of the noise level $\sigma$. Lower the rank denotes a better method.

| Method | Average Rank | | | | Rank |
|---|---|---|---|---|---|
| | aMSE | aTPR | aTPR.sh | aFPR | |
| TGUH1 | 4.4 | 1.5 | 1.5 | 6.0 | 3.35 |
| TGUH | 2.8 | 2.7 | 2.5 | 4.8 | 3.20 |
| TGUHb | 6.0 | 6.0 | 6.0 | 1.0 | 4.75 |
| TGUHm | 1.8 | 3.1 | 2.9 | 3.6 | 2.85 |
| UTGUH | 2.2 | 3.4 | 4.0 | 2.6 | 3.05 |
| UTGUHm | 3.8 | 4.3 | 4.1 | 3.0 | 3.80 |

Table 4.1 and 4.2 present the mean performance metric ranking across all noise levels, along with its overall ranking. These findings highlight the TGUHm and UTGUH methods as the leading pair compared to the rest. While they may not consistently excel in every individual metric, they consistently maintain a remarkable proximity to the optimal outcomes.

Table 4.2: Table of the average ranking of performance measurement for simulation using for noise from a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ over all of the noise level $\sigma$. Lower the rank denotes a better method.

| Method | Average Rank | | | | Rank |
|---|---|---|---|---|---|
| | aMSE | aTPR | aTPR.sh | aFPR | |
| TGUH1 | 6.0 | 1.6 | 1.2 | 6.0 | 3.70 |
| TGUH | 3.6 | 3.0 | 2.8 | 5.0 | 3.60 |
| TGUHb | 5.0 | 6.0 | 6.0 | 1.0 | 4.50 |
| TGUHm | 1.8 | 3.8 | 3.8 | 3.2 | 3.15 |
| UTGUH | 1.6 | 2.0 | 2.6 | 2.0 | 2.05 |
| UTGUHm | 3.0 | 4.6 | 4.6 | 3.8 | 4.00 |

Even though UTGUH show a good performance, the estimated signal for a segment, for example, $[a, b]$, is not obtained as a sample mean of the corresponding segment and it tends to form a small spike as seen at the position around 200 in Figure 4.20. This phenomenon occurs because UTGUH directly employs the inverse TGUH transform to reconstruct the segmentation. Hence, considering this result, it is preferable to use the utilization of TGUHm.

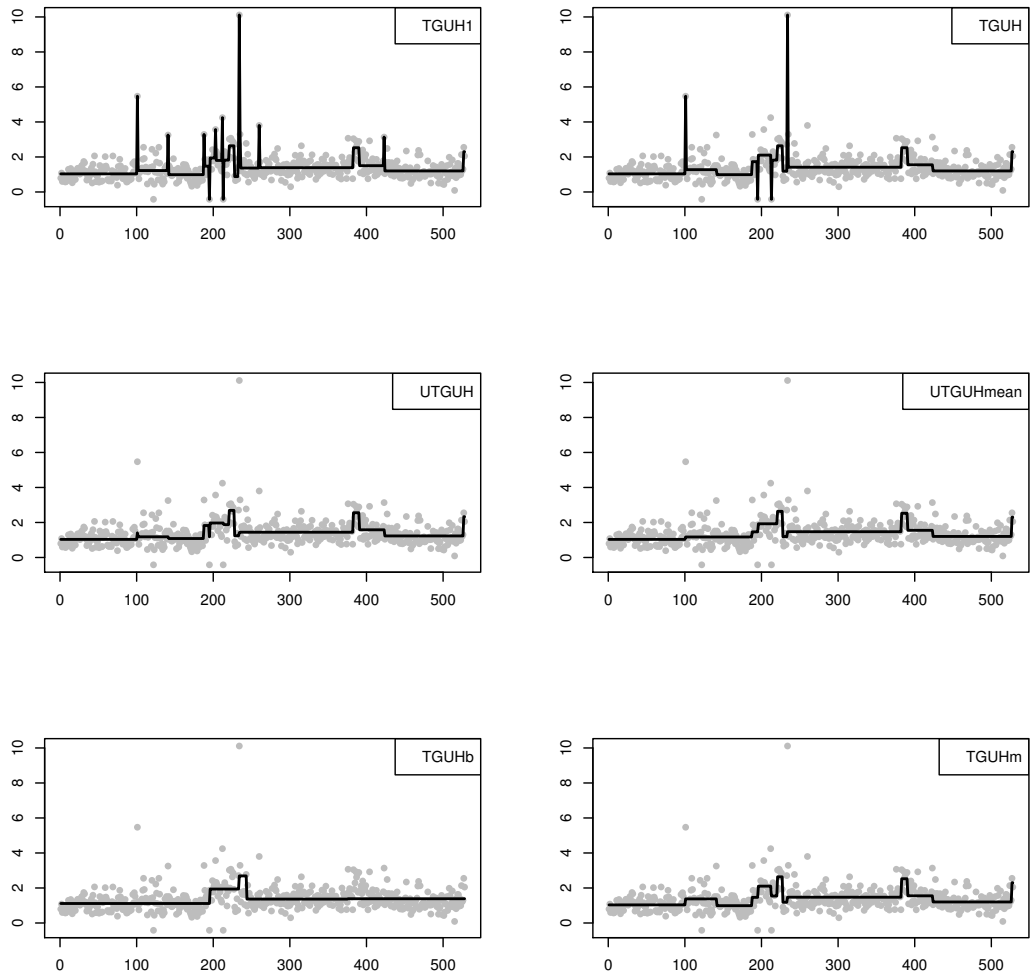


Figure 4.21: Comparison of segmentation result of chromosome 16 LA11 patient data using TGUH1, TGUH, UTGUH, UTGUHmean, TGUHb, and TGUHm methods.

## 4.5 Application to Real Data

To illustrate the types of segments produced in more detail, Figure 4.22 presents the results of segmentation based on TGUHm, TGUH, TGUH1, TGUHb, CBS, HaarSeg, CumSeg, and FDRSeg method in chromosome 8 of Patient TMA- 93

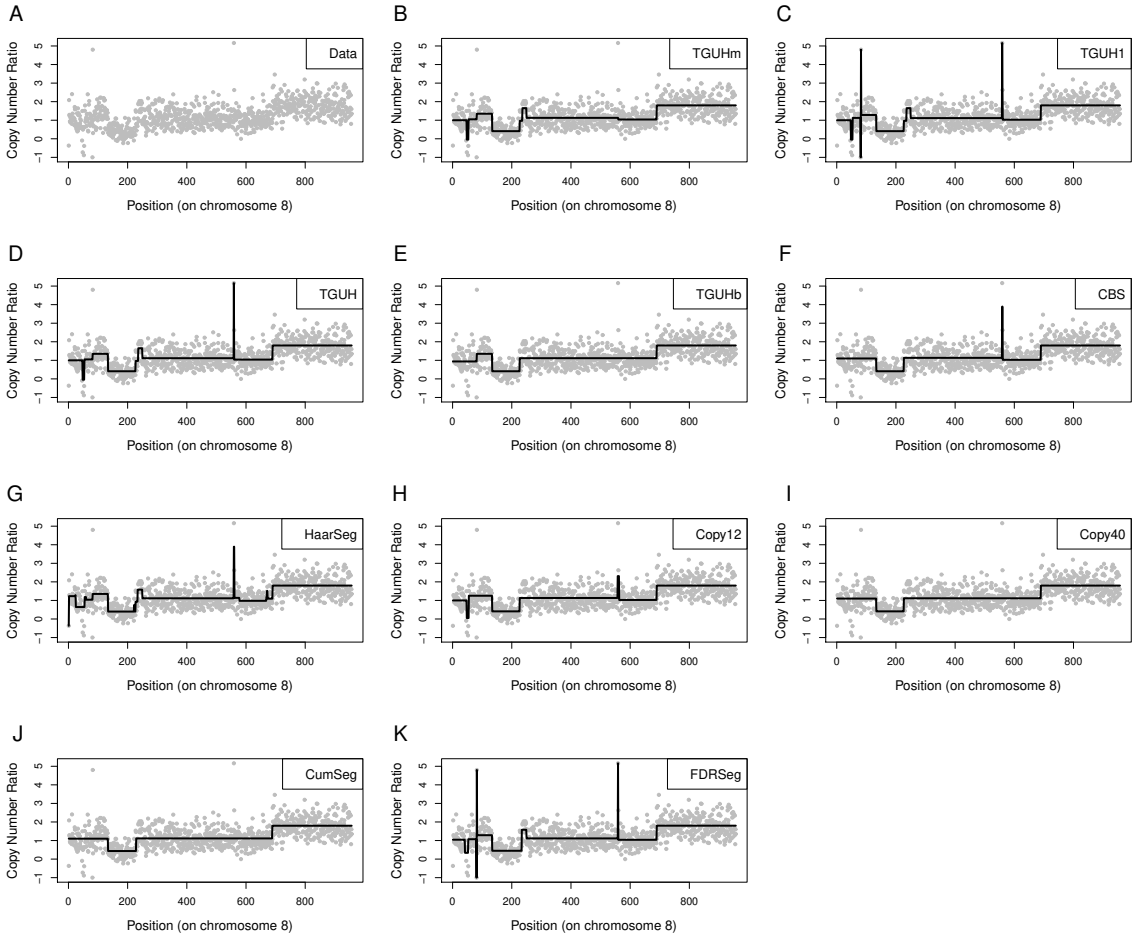


Figure 4.22: CNA estimate as a result of segmentation of chromosome 8 from patient TMA-93. **(A)** The copy number ratio data of chromosome 8 in Patient TMA-93. **(B)** TGUHm segmentation. **(C)** TGUH1 segmentation. **(D)** TGUH segmentation. **(E)** TGUHb segmentation. **(F)** CBS segmentation, **(G)** HaarSeg segmentation, **(H)** CopyNumber segmentation with $\gamma = 12$ and **(I)** $\gamma = 40$, **(J)** CumSeg segmentation, and **(K)** FDRSeg segmentation. For a quick reminder, TGUH1 denotes both TGUH and TGUHm method with $m^* = 1$ while TGUHm and TGUH denote TGUHm and TGUH method with $m^* = 2$, respectively. TGUHb denotes TGUH method with a localised pruning algorithm. Copy12 and Copy40 denote CopyNumber method with $\gamma$ parameter equal to 12 and 40, respectively.

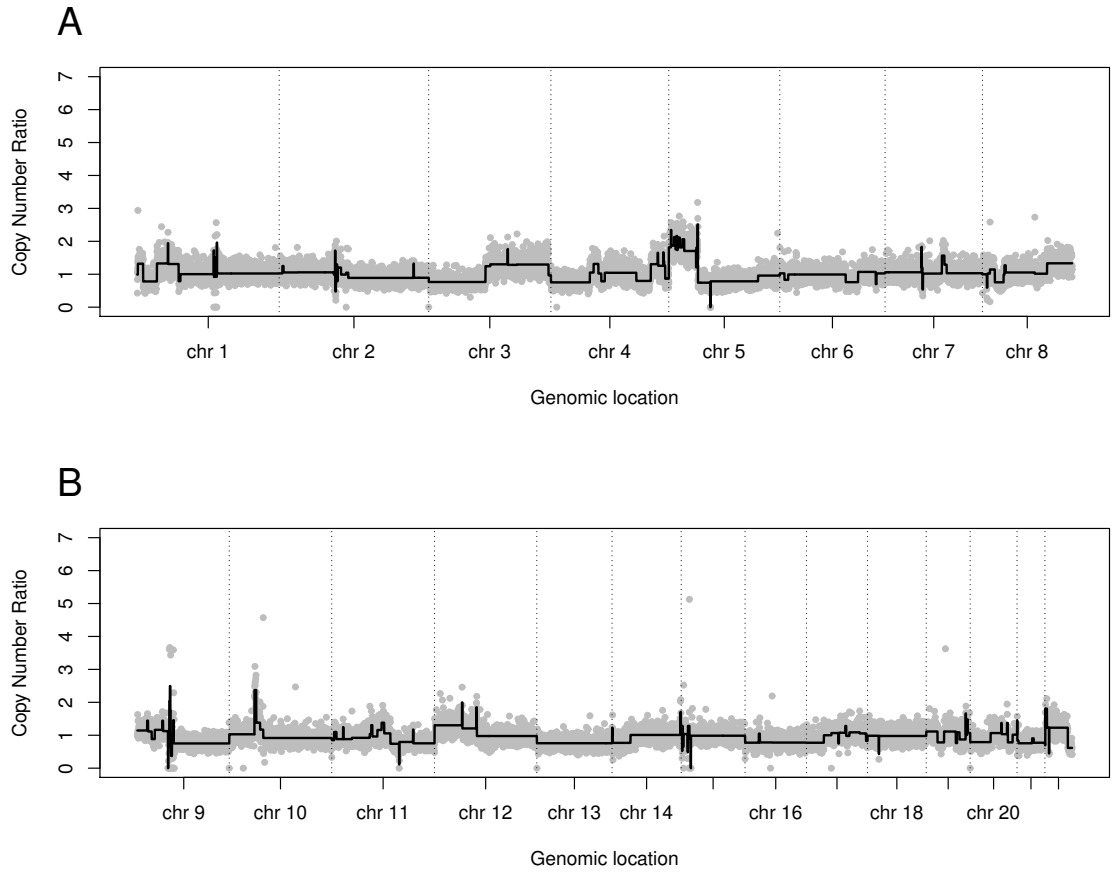


Figure 4.23: TGUHm CNA estimate as a result of segmentation of **(A)** chromosome 1–8 and **(B)** chromosome 9–22 in patient TMA-93.

Belvedere *et al.* (2012). The TGUH segmentation of the whole genome is shown in Figure 4.23. Figure 4.22 shows that most of the methods except CBS, Copy40, TGUHb and CumSeg estimate short segments at position around 50 and 250. From the simulation study, it has been shown that the CBS, CumSeg, TGUHb And Copy40 are not sensitive to short segments compared to the remaining methods. This indicates that there may be short altered segments at that region with high probability. TGUHm estimates more short segments than Copy12 but less than HaarSeg and FDRSeg which corresponds to results from the simulation, where Copy12 is less sensitive to short segments while both HaarSeg and FDRSeg tend to form more false positives than TGUHm.

Moreover, in this example, the differences between TGUH and TGUHm are clearly seen. The spikes (due to extreme single points) that are remained in TGUH1 ($m^* = 1$) are completely removed in TGUHm ($m^* = 2$). While the standard TGUH without the unconnected thresholding could not remove all those spikes, even when $m^* = 2$. Since the truth in real data is unknown, it is difficult to confirm whether the spikes are real changes or not although, given that they are single points, we expect a priori that they are not. However, these results indicate that one should consider TGUHm when it is appropriate to assume a value for the minimum segment length $m^*$.

### 4.5.1 Array Comparative Genomic Hybridization (aCGH) Data

Figure 4.24 and 4.25 show the TGUHm segmentation for the breast cancer dataset from Snijders *et al.* (2001). The CNA data are from array comparative genomic hybridization (aCGH) technology. From our observation, the aCGH data are not as noisy as NGS data but the results indicate the TGUHm method shows a similar segmentation pattern as seen in NGS data. It estimates more short aberrations compared to the CBS method but fewer than the FDRseg method. This is consistent with the simulation results, CBS is less sensitive to short segments while FDRseg tends to overestimate the change-points.

The true CNA pattern of this example is unknown but, for example from Figure 4.24, even with eyes, there can be seen that there might be a drop around

location probe number 360–430 and then a jump around 430–470 which are indicated by red dotted lines. The CBS method is unable to estimate these alterations, which is consistent with its strength in estimating long segments but often misses short segments. On the other hand, both the TGUHm and FDRseg method are able to estimate those alterations, but they are more sensitive to outliers in the data, leading to short segments corresponding to noise. One advantage of the TGUHm method is that by the application of two-stage thresholding, the sensitivity of the TGUHm method to outliers can be restrained so that its segmentation results are not as noisy as the FDRseg.

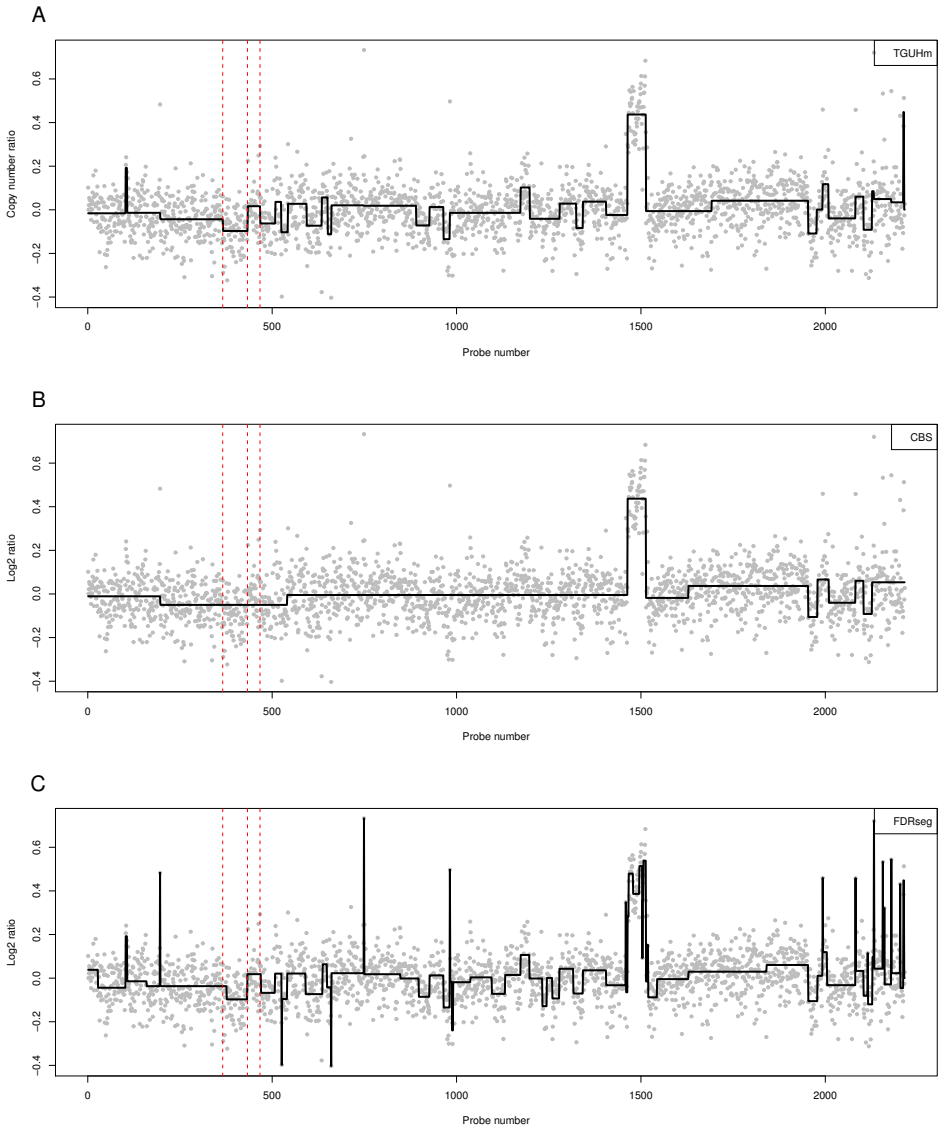


Figure 4.24: CNA estimate as a result of segmentation of array comparative genomic hybridization (aCGH) data GSM799. The points are normalized log ratios and graph is in genomic coordinates. **(A)** TGUHm segmentation. **(B)** CBS segmentation, **(C)** FDRSeg segmentation.

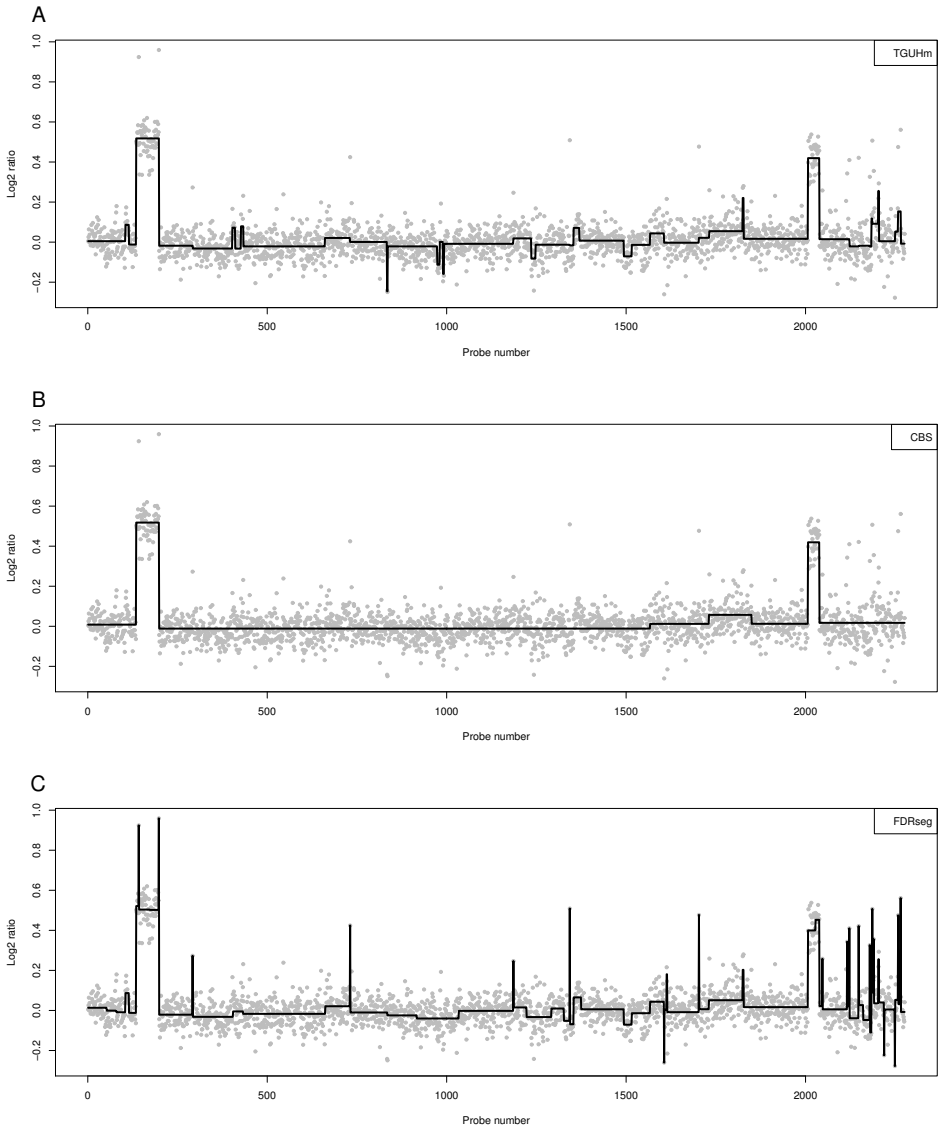


Figure 4.25: CNA estimate as a result of segmentation of array comparative genomic hybridization (aCGH) data GSM802. The points are normalized log ratios and graph is in genomic coordinates. **(A)** TGUHm segmentation. **(B)** CBS segmentation, **(C)** FDRSeg segmentation.

## 4.6 Conclusion

This chapter has described how 'spikes' occurred in the TGUH segmentation. The extremely short length of either of the unbalanced Haar wavelet wings becomes the main cause of the occurrence of spikes. Spikes in TGUH estimate are likely to occur when the detail coefficients that correspond to these extremely short wing unbalanced Haar wavelets survive the thresholding.

To address this tendency of the standard TGUH method to overestimate CNA spikes, the TGUH method was adapted for use with copy number data by modifying its thresholding technique so that it is no longer constrained to the 'unary-binary tree' structure; this adaptation is named TGUHm method. By modifying the thresholding procedure, the TGUHm method is shown to be successful in reducing those spikes.

The simulation study showed that setting $m^*$ to two gives the most benefit in terms of reducing the occurrences of single-point spikes in the segmented CNA. When $m^*$ is increased further to three, four, or even five, say, the results are very similar to $m^* = 2$. This indicates that, for reasonably low values of $m^* \geq 2$, the conclusion is not sensitive to the choice of $m^*$. From the simulation, for $m^*$ higher than the length of the shortest true segment, the TGUHm method was almost unable to detect the short segments. This confirms the ability of $m*$ to control the minimum length of the estimated segment. Based on this result, if users do not generally know the minimum length of expected altered segments, it is safe to set $m^* = 2$ as it provides a significant improvement to the performance of the method.

The simulation results also suggested that the proposed method has good operating characteristics to detect segments of different sizes. Some methods may have a tendency to identify more short segments or long segments. The proposed methods demonstrably work well for both short and long segments. This result becomes increasingly crucial in the case of low-coverage NGS data such as is the case in this study. This is because, for example, a 1 Mb segment is represented by only 5-7 windows or data points (Gusnanto *et al.*, 2014). In this case, segmentation methods are tested to the limit of detection, and the choice of the method becomes crucial. In the context of high-coverage NGS data, then

the same 1 Mb segment can be represented by hundreds of points. In such cases, most of the segmentation methods are expected to perform well with very little difference between their results.

Even though it has been shown that the TGUHm method perform well in identifying change-points where the data contain Gaussian noise with constant variance, analysing data with more complex noise structures is still challenging. This is a subject for the next chapter.

# Chapter 5

# Data-Driven TGUH-Fisz (DDTF) Method for Copy Number Segmentation with heteroscedastic noise

## 5.1 Introduction

This chapter explores a new wavelet approach named data-driven TGUH-Fisz (DDTF) that extends the data-driven wavelet-Fisz methodology (Fryzlewicz, 2008) to TGUHm wavelets denoising for handling non-negative data with heteroscedastic noise whose variance is non-decreasing function of the mean. The performance of the proposed method is assessed by simulation study and application to the real copy number ratio data. A paper based on the work in this chapter is currently in preparation.

Copy number alteration (CNA) data, especially those from NGS technology, have some characteristics that pose two inter-related challenges for change-point detection. The first challenge is the presence of non-constant random variation in the data, partly due to the normalisation pre-processing needed for this type of data (Gusnanto *et al.*, 2012). Specifically, the variance exhibits some association with the mean, which is exacerbated in the context of our study by low-coverage sequencing (<0.1X) (Wood *et al.*, 2010). When a segment has a high copy number

ratio, higher random variations are generally observed in the CNA data. Secondly, with such non-constant error variance in our copy number data, the detection of short segments is extremely challenging with some spurious changes often detected. In a low-coverage setting, a short segment, e.g. 1 Mbp, may be represented by just a few data points. In the analysis of CNA data, it is important to be able to detect accurately both short and long segments because they may give information on the location of oncogenes or tumour suppressor genes (Lengauer *et al.*, 1998). This chapter aims to address these challenges in a unified analytical framework.

For the first challenge, segmentation methods that rely on the homoscedastic error assumption are not ideal for CNA data since they can produce a large number of spurious change-points. Therefore, to handle this, a method that acknowledges the heteroscedasticity of error by using a method that applies a variance stabilisation process before proceeding to change-point estimation is generally needed.

Fryzlewicz *et al.* (2007) proposed a data-driven Haar-Fisz (DDHF) which utilises the balanced Haar wavelet transform to perform stabilisation in the wavelet domain prior to denoising the variance-stabilised data. This approach assumes that the noise variance is linked to the mean level of the data by an unknown function whose estimation is data-driven. Fryzlewicz *et al.* (2007) show that their DDHF transform is able to stabilise the variance better than some of time-domain variance stabilising transforms.

However, this approach in its standard form is not sufficient to address the above problem in CNA data. Like other methods based on balanced Haar wavelets, it creates spurious change-points at dyadic locations as an artefact of the balanced Haar wavelet transform. Therefore, in this chapter, the DDHF approach is extended by considering the tail-greedy unbalanced Haar (TGUH) wavelet transformation (Fryzlewicz, 2018) instead of the balanced Haar wavelet. As described in the previous chapter, unlike the balanced Haar transform, this application of the TGUH transform gives the advantage of adaptively adjusting the breakpoints in each wavelet basis function to sparsely describe the likely structure of the signal compared. This important feature enables us to reduce the number of spurious change-points commonly found in balanced Haar methods due to its

transformation structure. In section 5.4, a segmentation method named data-driven TGUH-Fisz (DDTF) method that utilises the TGUH wavelet transform for both variance stabilisation and denoising to replace any use of balanced Haar wavelet in DDHF method is introduced.

With regard to the challenge of estimating short segments, some spurious 'spikes' (segments with only one data point) can still be present, even with the adoption of the TGUH wavelet transformation in our proposed method. This makes it difficult to interpret the results because the short segments estimated are only represented by approximately 6-7 data points (Gusnanto *et al.*, 2014). Therefore, the estimation of short segments is at or close to the limit of detection and to address this challenge, as in the TGUHm method, an additional unconnected thresholding step is added as an improvement to the standard thresholding technique used in the TGUH method. This additional step enables control of the minimum segment length which can be estimated and, therefore, is able to control the occurrence of 'spikes' while at the same time still allowing the identification of these short segments.

This chapter proposes a unified analytical framework to address these challenges and presents good operating characteristics of the proposed method in a simulation study. Analysis of a real data example also shows that the proposed method is able to deal with the challenges and produce sensible results. The rest of the chapter is organised as follows. Section 5.3 reviews the key concepts of the DDHF method. The details of the proposed DDTF methodology are described in Sections 5.4. Section 5.5 illustrates in detail the differences between DDHF and DDTF methods. Sections 5.6 and 5.7 compare the performance of the proposed method with existing methods through simulated and real copy number DNA data, respectively.

## 5.2 Dataset

In this chapter, DNA sequence data from Belvedere *et al.* (2012) were considered. DNA extraction and libraries were prepared and sequenced using methods explained in detail in Section 2.4. The regions with missing values were removed,

such as the centromeres. An example of CNA data from patient TMA-93 is shown in Figure 5.1.

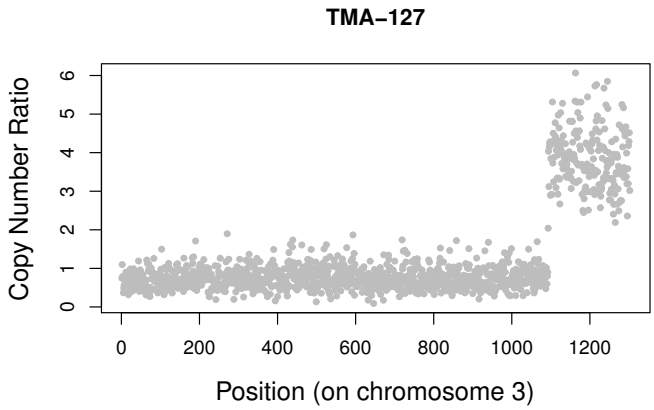


Figure 5.1: Example of chromosome 3 copy number ratio data from one patient, TMA-127. The data was normalised using CNAnorm (Gusnanto *et al.*, 2012) and regions with missing values, such as the centromeres, are removed. Each point in the figure denotes the copy number ratio of TMA-127 which corresponds to a specific genomic window (150 kb).

## 5.3 Data-Driven Haar-Fisz method

In general, the pattern of copy number alterations (CNA) can be considered as a piecewise constant function. CNAs can be identified by estimating the locations of regions with DNA copy number ratio deviating from one (Gusnanto *et al.*, 2012). This process is closely connected to the problem of detecting the locations of change-point.

To proceed with the statistical modelling, let $r_i$ be a sequence of the observed ratio between the tumour and normal genomes in the $i$-th window at genomic locations $x_i$ for $i = 1, 2, \ldots, n$. The genomic locations $x_i$ are known and satisfy $x_1 < x_2 < \cdots < x_n$. Generally, an additive measurement error model for relating the true copy number ratio signal $f_i$ and the observed copy number ratio $r_i$ can

be expressed as the following canonical change-point model

$$r_i = f_i + \epsilon_i, \quad i = 1, \ldots, n, \tag{5.1}$$

where $\epsilon_i$ is the error term and $f_i$ is unknown and believed to be a piecewise-constant signal with change-points at unknown location $\eta_1, \ldots, \eta_N$.

CNA data, as illustrated in Figure 5.1, often exhibit a feature where the noise variance may be linked to the mean level of the data. Because of this, the observed data $r_i$ is assumed to have distributional properties as specified below.

1. $(r_i)_{i=1}^n$ is a sequence of independent, nonnegative random variables with finite positive means $\mu_i > 0$ and positive variances $\sigma_i^2 > 0$.

2. The variance $\sigma^2$ is a non-decreasing function of the mean $\mu$:

$$\sigma_i^2 = h(\mu_i), \tag{5.2}$$

   where the function $h$ is independent of $i$.

Thus the task is to estimate change-points in $f_i$ from the noisy data $r_i$ under the above assumptions when $h$ is a non-decreasing function of the mean and unknown.

Fryzlewicz (2008) introduced a fully automatic multiscale technique named data-driven wavelet-Fisz (DDWF) method for approximately stabilising the variance of sequence of non-negative independent random variables whose variance is non-decreasing function of the mean. The term 'data-driven' here refers to an automatic way to estimate the function $h$ from the data as part of the variance stabilisation procedure. The DDWF methodology performs variance stabilisation in the wavelet domain, not in the time domain, like the standard square root transformation. One advantage of DDWF is it can make use of any wavelet denoising to remove noise from data whose variance is non-constant. In the context of change-point analysis, by employing Haar wavelets which have "square-shaped" characteristic, the piecewise constant estimation of noisy data can easily be obtained.

### 5.3.1 Literature Review of Data-Driven Haar-Fisz Method

The Data-Driven Haar-Fisz (DDHF) transform applies wavelet-fisz Methodology using Haar wavelets which allows us to obtain piecewise constant estimates (Fryzlewicz *et al.*, 2007). This approach requires the length of input data $r_i$ to be the power of two and the assumption that the noise variance is linked to the mean level of the data $\mu$ by an unknown function $h(\mu)$. Under the assumption that variance is a non-decreasing function of the mean, the DDHF method is able to transform the heteroscedastic noisy signal to one where the variance of the noise is constant. Fryzlewicz *et al.* (2007) show that DDHF transform is able to stabilise the variance better than some of 'time-domain' variance stabilising transform.

In the DDHF method, the appropriate variance-stabilising transformation is estimated from the data by first estimating the mean-variance relationship (the non-decreasing function $h(\mu)$) and then performing Haar-Fisz transform (Fryzlewicz & Nason, 2004) for stabilising the variance of the sequences. This variance stabilisation is called DDHF transformation.

Before describing further the DDHF transform, let us briefly recall the formula for the Haar transform. The Haar transform is a linear orthogonal transform $\mathbb{R}^n \to \mathbb{R}^n$ where $n = 2^J$. Given an input vector $\mathbf{X} = (X_i)_{i=1}^n$, the Haar transform can be performed as follows:

1. Let $c_i^J = X_i$.
2. For each $j = J-1, J-2, \ldots, 0$, recursively form vectors of smooth (or scaling) and detail coefficients, $\mathbf{c}^j$ and $\mathbf{d}^j$:

$$c_k^j = \frac{c_{2k-1}^{j+1} + c_{2k}^{j+1}}{2}; d_k^j = \frac{c_{2k-1}^{j+1} - c_{2k}^{j+1}}{2}, k = 1, \ldots, 2^j. \tag{5.3}$$

The inverse Haar transform simply reverses the formula 5.3 and is performed as follows:

1. For each $j = 0, 1, \ldots, J-1$, recursively forms $\mathbf{c}^{j+1}$:

$$c_{2k-1}^{j+1} = c_k^j + d_k^j; c_{2k}^{j+1} = c_k^j - d_k^j, k = 1, \ldots, 2^j. \tag{5.4}$$

2. Set $X_i = c_k^J$.

Note that, throughout this chapter, the requirement of balanced Haar wavelet transform input data to be the power of two is handled using the symmetric signal extension.

The main idea of the Haar-Fisz transform is to decompose $\mathbf{X}$ by Haar transform, stabilise the variance of detail coefficients $d_k^j$, and then apply the inverse Haar transform to obtain a vector with variance approximately stabilised. In set up (5.1), a simple Haar-Fisz transform would proceed as follows:

1. Take the Haar transform of $\mathbf{X}$ to obtain the detail coefficients $d_k^j$ and the smooth coefficients $c_k^j$.

2. Modify the smooth coefficients at each scales $j = 1, \ldots, J-1$ to transform them into local means of the data, $c_k^{*j} = 2^{(j-J)/2} c_k^j$.

3. Form the Haar-Fisz stabilised coefficients

$$d_k^{*j} = \frac{d_k^j}{\hat{h}^{1/2}(c_k^{*j})}, \tag{5.5}$$

   where $\hat{h}$ is the estimated $h$ function. This can be viewed as a kind of 'studentisation' in the wavelet domain. Note that the variance of $d_k^j$ is approximately equal to $h(\mu_k^j)$, where $\mu_k^j$ denotes the local mean of the sequence $X_i$ computed over the same support as the corresponding coefficients $d_k^j$ and $c_k^j$ so that $\mu_k^j$ can be pre-estimated by $c_k^{*j}$.

4. Take the inverse Haar transform of the transformed coefficients $d_k^{*j}$,

$$c_{2k-1}^{j+1} = c_k^j + d_k^{*j}; c_{2k}^{j+1} = c_k^j - d_k^{*j}, k = 1, \ldots, 2^j, \tag{5.6}$$

   for $k = 1, \ldots, 2^j - 1$ and $j$ going from 1 to $J$. Call the final $c_k^J$ vector, obtained from the Fisz–modified coefficients, $u_k$ for $k = 1, ..., 2^J = n$. The variance of the sequence $u_k$ is now stabilised.

Then now, after the Haar-Fisz transform, any wavelet denoising for homoscedastic noise can be applied. The most common yet effective wavelet denoising is Haar wavelet denoising with hard universal thresholding (Donoho & Johnstone, 1994).

For the remaining of this chapter, for simplicity, the term DDHF method refers to DDHF transform with hard universal thresholding.

Finally, the final piecewise constant estimator of $X_i$ can be obtained by taking the inverse Haar-Fisz transform. The inverse Haar-Fisz transform can be achieved by reversing the above steps: take the Haar wavelet transform of $u_k$, remultiply the $d_k^{*j}$ coefficients by $h(c_k^{*j})^{1/2}$, and then perform the inverse Haar wavelet transform.

### 5.3.2 Data-Driven Haar-Fisz with TGUHm Thresholding

As described in the previous subsection, it is important to note that any wavelet denoising technique that is appropriate for Gaussian noise (i.e., Discrete Haar Wavelet Transform (DHWT)—thresholding—inverse DHWT) can be used for the denoising stage. In this subsection, the DDHF method is extended using TGUHm denoising to utilise the superiority of TGUHm in estimating change-point location by perform it in the denoising stage of DDHF method. For the remaining text, this approach is called as DDHF+T method.

The detailed procedure of the DDHF+T method can be outlined as the following four stages.

1. Stage 1: Variance stabilisation stage. Apply a DDHF transform to the noisy data $r_i$ which will result in data that is contaminated with approximately Gaussian homoscedastic noise $r_i^s$.

2. Stage 2: Denoising stage. Denoise the stabilised data $r_i^s$ using TGUHm method as explained in Section 4.3 to obtain the piecewise constant estimate $\hat{r}_i^s$.

3. Stage 3: Reconstruction stage. Reconstruct the segmentation result by taking the inverse Haar-Fisz transform. This can simply be achieved by reversing the Haar-Fisz transform procedure: take the Haar wavelet transform of $\hat{r}_i^s$, remultiply it by $h^{1/2}(\mathbf{c})$, and then perform the inverse Haar wavelet transform.

## 5.4 Data-driven TGUH-Fisz Method

Despite the simplicity and flexibility of the DDHF method, the use of Haar wavelet transform often causes oversegmentation due to the dyadic structure of its transformation. A more detailed explanation of this oversegmentation tendency is discussed in Section 5.5. For that reason, in this section, a new wavelet-based approach is proposed to replace the balanced Haar wavelet transform with the TGUH wavelet transform (Fryzlewicz, 2018) in the variance stabilisation step. The aim is to take benefit from the unbalanced Haar wavelet used in the TGUH transform which makes the transformation not take the dyadic structure anymore. Rather, it follows the structure of the data by adjusting the location of breakpoints of the unbalanced Haar wavelet. This transformation, which called as TGUH-Fisz transform, allows us to translate the signal into a set of unbalanced Haar wavelet coefficients that are approximately Gaussian. Then the denoising/thresholding can be done to those coefficients via universal thresholding.

In the following subsections, the proposed method named data-driven TGUH-Fisz (DDTF) method which is a change-point detection method based on a wavelet-domain variance stabilising transform. The procedure of the DDTF method can be outlined into the following four main steps:

1. Estimation of the variance function $h$ as a function of the mean. In the real data problem, it is often to face a situation when the function $h$ is unknown so we need to estimate it from the data. Here, the function $h$ is estimated using isotone regression (Johnstone & Silverman, 2005a), as suggested by Fryzlewicz *et al.* (2007). This is described further in Section 5.4.1.

2. Variance stabilisation. In this stage, the heteroscedastic noise problem is addressed by performing variance stabilisation in the wavelet domain. the TGUH-Fisz transform is applied to transform data with non-constant variance noise to a set of unbalanced Haar wavelet coefficients named detail coefficients which are approximately Gaussian with mean zero and variance equal to one. The details are described in Section 5.4.2.

3. Denoising or thresholding. The main purpose of this stage is to determine which wavelet coefficients are likely to represent the true signals and should

be retained in the wavelet reconstruction phase by performing a two-stage threholding process which includes the connected thresholding introduced in Fryzlewicz (2018) and then followed by unconnected thresholding to prune the spurious change-points which are commonly estimated due to extreme observations found in NGS data. The details are described in Section 5.4.3.

4. Signal reconstruction. The final step, reconstruct the signal by taking the sample mean of the observed data within each segment between consecutive estimated change-points.

### 5.4.1 Estimation of Function $h$

As mentioned earlier, in practice, there may occur a condition when one believes there is a mean-variance relation in the data where $\sigma^2 = h(\mu)$ but does not know exactly what is the function $h$. The $h$ function is needed to be estimated from the data first. Since $\sigma^2 = h(\mu)$, the standard deviation can be written as a non-decreasing function of the mean, $\sigma = h(\mu)$. Due to the piecewise-constant pattern of the underlying signal, $\sigma_i^2$ can be estimated by $\hat{\sigma}_i^2 = (r_i - r_{i+1})^2/2$ and the empirical mean $\hat{\mu}_i$ can also be estimated by $\hat{\mu}_i = (r_i + r_{i+1})/2$. This discussion motivates the following regression setup:

$$\hat{\sigma}_i^2 = h(\hat{\mu}_i) + \epsilon_i. \tag{5.7}$$

As $h$ should be a non-decreasing function of $\mu_i$, $h$ can be estimated via a monotonic regression. Also, the standard deviation $\sigma$ can be estimated by $h^{1/2}$.

In this chapter, a "pool-adjacent-violators" algorithm for least-squares isotone regression described in Johnstone & Silverman (2005a) is used. Given a set of data points $(\hat{\mu}_i, \hat{\sigma}_i^2)$, the objective of isotone regression is to find a non-decreasing function $h(\hat{\mu}_i)$ that minimizes the sum of squared differences between the observed response $\hat{\sigma}_i^2$ and the fitted values $h_i$. Mathematically, this problem can be written by

$$f(\hat{\mu}) = \sum_i w(\hat{\sigma}_i^2 - h_i)^2 \to \text{min!} \tag{5.8}$$

which has to be minimized over $h_i$ under the inequality restrictions $h_1 \leq h_2 \leq \ldots \leq h_n$. The $w_i$ in the equation above are some optional observation weights. The weights $w_i \geq 0$ are added to the equation (5.8) for generality, although usually $w_i = 1$ (in this chapter $w_i$ is equal to 1 also).

Let us assume that the pairs $(\hat{\mu}_i, \hat{\sigma}_i^2)$ are ordered with respect to the predictors $\hat{\mu}_i$. Let $l$ be the index of iteration and $p$ be the index of the blocks where $p = 1, \ldots, B$. The pool-adjacent-violators algorithm can be performed as follows.

1. For $l = 0$, set the initial solution as $h_i^{(0)} := \hat{\sigma}_i^2$ and set $p = n$ which means that each observation $h_p^{(0)}$ is set to be a block.
2. Merge $h^{(l)}$-values into blocks if $h_{p+1}^{(l)} < h_p^{(l)}$.
3. Solve $f(\hat{\mu})$ in equation (5.8) for each block $p$.
4. If there is $h_{p+1}^{(l)} < h_p^{(l)}$ increase $l := l + 1$ and go beck to step 2.

The iteration stops when all the blocks are increasing, i.e., $h_{p+1}^{(l)} \geq h_p^{(l)}$. Finally, the block values are increased with respect to the observations $i = 1, \ldots, n$ such that the final result is the vector $\hat{\mathbf{h}}$ of length $n$ with elements in increasing order.

### 5.4.2 Variance Stabilisation: TGUH-Fisz Transformation

To bring the heteroscedastic noise problem into homoscedastic noise, the variance stabilisation procedure holds an important role. In the proposed method, variance stabilisation is performed in the unbalanced Haar wavelet domain. Due to the 'unbalanced' nature of the wavelet used, it allows us to remove the dyadic artefact of the original DDHF method. The variance stabilisation procedure is performed by first, applying TGUH wavelet transform (Fryzlewicz, 2018) to decompose the data into several scales and bring them into unbalanced wavelet domain and then performing Fisz transform to stabilised the decomposed data.

The TGUH wavelet transform is a bottom-up method that utilises unbalanced wavelets to translate the sequence $r_i$ into a set of different type coefficients that form an unary-binary tree structure (Fryzlewicz, 2018). The detailed procedure of the TGUH transform has been described in Section 3.3.1. But for a quick reminder, here a brief explanation is presented.

The TGUH transform is started by constructing adjacent pairs of local rescaled average coefficients named smooth coefficients, $c_{s,e}$. Given the sequence $r_i$ of copy number ratio observations, the smooth coefficients, $c_{s,e}$, in the region $i = s, \ldots, e$, is given by

$$c_{s,e} = \frac{1}{\sqrt{e-s+1}} \sum_{i=s}^{e} r_i. \tag{5.9}$$

At each iteration $j$, let $C^j = \{c_{s,e}\}$ to be the set of smooth coefficients of the data $r_i$, and $\alpha_j$ to be the length of the smooth coefficients, after the $j$-th iteration.

The main idea of the TGUH transform is to concentrate as little as possible power of the data at the 'finer' or lower levels scale. This is attained by merging $\lceil \rho\alpha_j \rceil$ pairs of regions which are thought to have the smallest variability. The merged regions are determined by computing the detail coefficients, $d_{s,b,e}$, which represent the 'difference' between two consecutive regions and select its $\lceil \rho\alpha_j \rceil$ smallest absolute value. The $\rho$ parameter describes the number of regions merged at each iteration and the parameter $\rho$ is set to 0.01 as suggested by (Fryzlewicz, 2018) in the remainder of the chapter.

For $j = 1$, initial smooth coefficients are assigned to be the data, $C^1 = \{c_{1,1}, c_{2,2}, \ldots, c_{i,i}, \ldots, c_{n,n}\} = \{r_1, r_2, \ldots, r_i, \ldots, r_n\}$. Then the detail coefficients, $d_{s,b,e}$, for each adjacent pair in $C^j$ can be computed as

$$d_{s,b,e} = l_{s,b}c_{s,b} - r_{b+1,e}c_{b+1,e}, \tag{5.10}$$

where $(l_{s,b}, -r_{b+1,e})$ is the 'detail' filter with restriction: $l_{s,b}^2 + r_{b+1,e}^2 = 1$ and $d_{s,b,e} = l_{s,b}c_{s,b} - r_{b+1,e}c_{b+1,e}$ should be zero if $(r_s, \ldots, r_e)$ is a constant vector. Roughly speaking, the indices $s, b$ and $e$ correspond to the approximate location of the start, breakpoint, and end of the unbalanced Haar wavelet used, respectively.

The detail coefficients of scale $j$ are chosen by sorting the sequence $|d_{s,b,e}|$ in ascending order then save the $\lceil \rho\alpha_j \rceil$ pairs of the smallest absolute value of the detail coefficient vector. Then, the new local smooth coefficients, $C^{j+1}$, are produced by merging the regions that correspond to the selected detail coefficients and computing the scaled average of the data in those merged regions. These procedures are repeated until only one detail coefficient is extracted. Therefore, at the end of the iteration, a set of detail coefficients $d_{s,b,e}^{j,k}$ is obtained where $j$

denotes the iteration or scale and $k$ indexes the detail coefficients according to increasing $s$ at each scale $j$; $k = 1, \ldots, K(j)$.

One advantage of the TGUH transform due to the 'greedy' and 'unbalanced' characteristic is its flexibility to have detail coefficients $d_{s,b,e}$ that correspond to wavelets whose either left/right $(b - s + 1/e - b)$ region length is very short at the coarser (larger) scale. This means that it is possible to store the significant 'difference' related to short segments at the coarser scale that tend to survive after thresholding. This gives the benefit of improving the sensitivity of the method in estimating short segments.

Note that now we have a set of detail coefficients, $\{d^{j,k}_{s,b,e}\}$, and the estimation of $h^{1/2}(\mu_i)$ which is approximately equal to the standard deviation of $\{d^{j,k}_{s,b,e}\}$. Thus, the stabilised detail coefficients can be obtained through a simple standardisation procedure as follows:

1. Modify the smooth coefficients $c^{j,k}_{s,e}$ to transform them into local means of the data
$$c^{*j,k}_{s,e} = (e - s + 1)^{-1/2} c^{j,k}_{s,e}. \tag{5.11}$$

2. If $\hat{h}(c^{*j,k}_{s,e}) \neq 0$, form the TGUH–Fisz stabilised coefficients $d^{*j,k}_{s,b,e}$ by dividing the detail coefficients $d^{j,k}_{s,b,e}$, by its local estimated standard deviation $\hat{h}^{1/2}(c^{*j,k}_{s,e})$
$$d^{*j,k}_{s,b,e} = \frac{d^{j,k}_{s,b,e}}{\hat{h}^{1/2}(c^{*j,k}_{s,e})}. \tag{5.12}$$
Otherwise, set $d^{*j,k}_{s,b,e} = d^{j,k}_{s,b,e}$. In a non-wavelet setting, the above ratio transformation is similar to that studied by Fisz (1955) which justifies the name of TGUH-Fisz transform.

At this stage, the stabilised detail coefficients, $d^{*j,k}_{s,b,e}$, are obtained which are approximately Gaussian with mean zero and variance one. This means that now the heteroscedastic noise has been transformed into a homoscedastic noise and, therefore, any suitable Gaussian wavelet thresholding can be used.

### 5.4.3 Thresholding

Unlike the standard TGUH denoising which only uses connected thresholding, a two-stage thresholding technique as explained in the Chapter 4.3.2 is used for the DDTF method. It applies connected thresholding to set those detail coefficients whose values are less than a specific threshold to zero and unconnected thresholding to control the minimum estimated segment length, $m^*$, say.

Specifically, let the children coefficients of $d^{*j,k}_{s,b,e}$ be the set of finer-scale coefficients whose support is entirely inside $[s, e]$:

$$\mathcal{C}^{j,k}_{s,b,e} = \{d^{*j',k'}_{s',b',e'} : [s', e'] \subseteq [s, e] \text{ for all } j' = 1, \ldots, j-1\}.$$

The connected thresholded stabilised coefficients $d^{*j,k}_{s,b,e}$ is given by

$$\hat{d}^{*j,k}_{s,b,e} = d^{*j,k}_{s,b,e}\mathbb{I}\{\exists d^{*j',k'}_{s',b',e'} \in \mathcal{C}^{j,k}_{s,b,e} > \lambda\}, \tag{5.13}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function.

Using only connected thresholding often leads to the problem of single-point spikes occurring in the estimation. 'Spikes' are likely to occur when the detail coefficients $d^{*j,k}_{s,b,e}$, with either $e - b$ or $b - s + 1$ equals to one, survive the connected thresholding. To control the occurrence of these 'spikes', the unconnected thresholding is applied to the $\tilde{d}^{*j,k}_{s,b,e}$ which is given by

$$\tilde{d}^{*j,k}_{s,b,e} = \hat{d}^{*j,k}_{s,b,e}\mathbb{I}\{(b - s + 1) > m^*\}\mathbb{I}\{(e - b) > m^*\}. \tag{5.14}$$

Based on the simulation in Section 4.4.5, setting $m^* = 2$ has enabled a significant reduction of spurious change-points that are generally caused by single-point outliers. For reasonably low values of $m^* \geq 2$ (i.e. three and four), the results are very similar to $m^* = 2$ which indicates that the conclusion is not sensitive to the choice of $m^*$, for $m^* \geq 2$. This is shown in the Appendix.

### 5.4.4 Signal Reconstruction

The last step is the reconstruction of the estimated segment. The final estimator of $f_i$ is obtained by setting the value of the signal between two consecutive breakpoints to be the average of all copy number ratio data in $r_i$ over that interval.

Since the estimated location of breakpoints are given by index $q$ in the survived stabilised detail coefficients $\tilde{d}^{*j,k}_{s,b,e}$, our final estimator $\hat{f}$ is given by

$$\hat{f}_t = \frac{1}{\eta_{l+1} - \eta_l} \sum_{s=\eta_l}^{\eta_{l+1}} r_s, \tag{5.15}$$

for $t \in [\eta_l, \eta_{l+1}]$ and $\eta_l = \{0, b_1, b_2, \ldots, b_N, n\}$. The $n$ is the length of the copy number ratio data $r_i$ and $\{b_l\}$ denote the collection of $b \in \tilde{d}^{*j,k}_{s,b,e}$ in ascending order where $l = 1, .., N$ and $N$ is the number of estimated change-points.

## 5.5 Comparison of DDHF, DDHF+T, and DDTF

So far, the standard DDHF method (Fryzlewicz & Delouille (2005)), the DDHF method with TGUHm thresholding (DDHF+T method), and the proposed DDTF method have been described. In this section, the focus is to discuss the differences of these methods in more detail from the detail coefficients' points of view. For this matter, it is important to be able to visualise the detail coefficient produced by both Haar and TGUH transformations. Here, two kinds of visualisation of the wavelet coefficients are used to illustrate each of those transformations.

For the Haar wavelet transform, the first visualisation is shown in the middle left panel of Figure 5.2. The figure is produced by `ywd` function from `wavethresh` R package Nason (2016). Each of Haar wavelet coefficients $d^j_k$ is plotted with the finest-scale coefficients at the bottom of the plot and the coarsest at the top. The level or parameter $j$ is indicated by the left-hand axis. The value of the coefficient is displayed by a vertical mark located along an imaginary horizontal line centred at each level. The magnitude of coefficients at each level is plotted according to a scale that varies according to level. The $k$, or location parameter, of each $d^j_k$ wavelet coefficient is labelled 'Translate', and the horizontal positions of the coefficients indicate the approximate position in the original sequence from which the coefficient is derived. The second visualisation of the Haar wavelet transform is shown in the bottom left panel of Figure 5.2. This attempts to visualise the magnitude of the wavelet coefficients more clearly by plotting the value/magnitude of coefficients with regards to their location hence each coefficients are comparable.

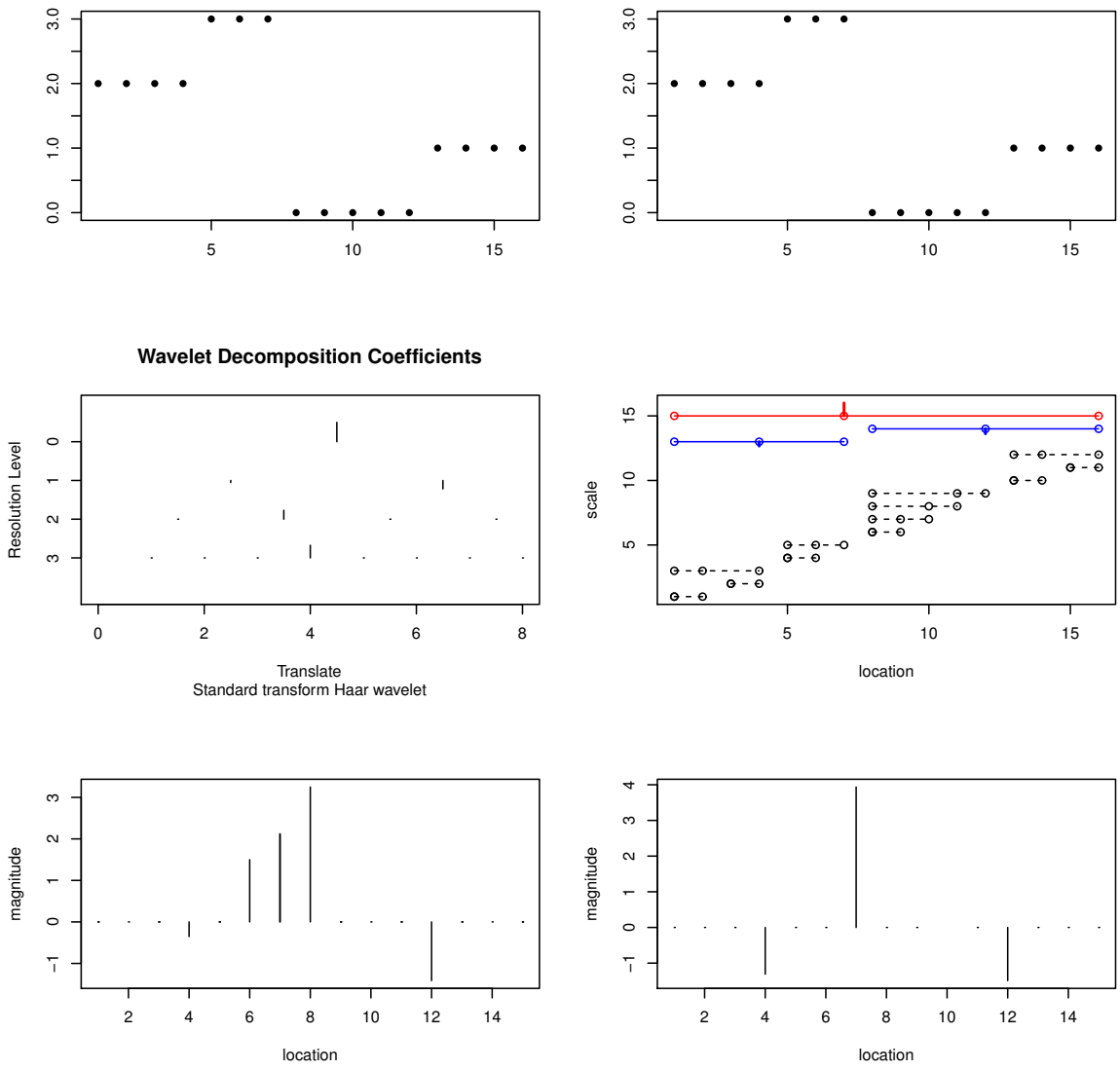


Figure 5.2: Top row: identical copies of the simulated data. Middle: the first visualisation of Haar wavelet coefficients (left) and TGUH coefficients (right) of the simulated data. The level or parameter $j$ is indicated by the left-hand axis. The magnitude of the coefficient is displayed by a vertical mark located along an imaginary horizontal and the horizontal positions of the coefficients indicate the approximate position in the original sequence from which the coefficient is derived. For the TGUH coefficients plot, the length of the basis used is denoted by the horizontal lines. Bottom: the second visualisation of Haar wavelet coefficients (left) and TGUH coefficients (right). The second visualisation of both the Haar and TGUH coefficient plot the magnitude of the coefficients with regards to their corresponding location of the original data.

For TGUH wavelet transform, two kinds of visualisations are also considered. The first visualisation (see middle right panel of Figure 5.2) intends to explain the region merges to produce a detail coefficient $d_{s,b,e}$ at each scale. The region merges are denoted by a horizontal line with 3 dots. The first, second, and third dot denote indices $s, b$, and $e$ of the detail coefficient $d^{j,k}_{s,b,e}$ respectively. The line that only has 2 dots indicates that $s = b$ which means the first dot denotes both $s$ and $b$. The region merges are plotted with the finest-scale coefficients at the bottom of the plot and the coarsest at the top. The level is indicated by the left-hand axis. The value of the detail coefficient is displayed by a vertical mark located along the region merges line that corresponds to the coefficient. The position of the coefficients on the region merges line indicates the index $b$ or in terms of its wavelet shape, it denotes the location of the breakpoint of the corresponding wavelet. The red and blue colours of the lines show the positive and negative signs of the coefficients, respectively. Whereas the black dashed lines indicate the detail coefficients which are equal to zero. For the second visualisation of the TGUH transform, the TGUH coefficients are plotted using the exact same way as the second visualisation of the Haar wavelet transform (see bottom right panel of Figure 5.2).

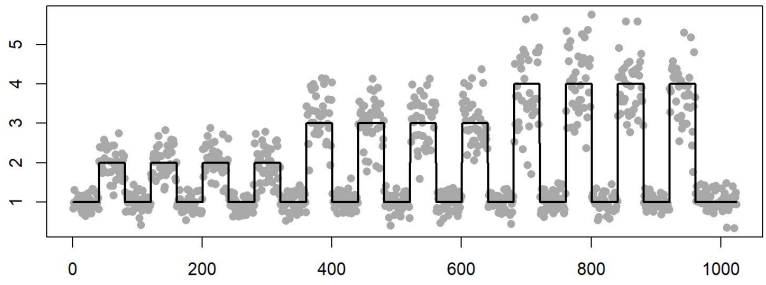


Figure 5.3: An example of simulated data contaminated by Gaussian noise with mean zero and variance $\sigma^2 = 0.2^2 f_i^2$. Grey dots denote the simulated data. Black solid line denotes the true pattern.

In the following subsections, the characteristic of each method is evaluated

using the wavelet coefficients visualisations defined above. For the purpose of illustration, a simple heteroscedastic change-points detection problem was considered. A simulated data contaminated by Gaussian noise with mean zero and variance $\sigma^2 = 0.2^2 f_i^2$ was used as shown in Figure 5.3.

### 5.5.1 Data-Driven Haar-Fisz (DDHF)

The variance stabilisation procedure of the DDHF method is performed by applying Fizs transform in Haar wavelet domain. In this section, for the simplicity of illustration, the DDHF method procedure is divided into three main stages: (i)variance stabilisation stage, (ii)denoising stage, and (iii)reconstruction stage. Figures 5.4, 5.5, and 5.6 show each those stages, respectively.

First, the Haar transform is applied to the data to obtain Haar wavelet coefficients. Then, Fisz transform is performed by dividing wavelet coefficients by their approximate standard deviation. The change of wavelet coefficient by this Fisz transform can be seen in Figure 5.4. The finest or scale 9 coefficients that relate to the higher variability in the original data tend to have higher values. But after the Fisz transform (right side panel of Figure 5.4), even only using eyes, we can see that the magnitude of coefficient become more uniform which indicates that the variance is stabilised. If one takes an inverse of these stabilised coefficients, as shown in the top right panel of Figure 5.4, one can see that the data is stabilised as well.

The next procedure is denoising. For this purpose, Haar wavelet shrinkage with hard thresholding is used. The threshold value, $\lambda$, is obtained by the universal threshold which is given by $\lambda = \sigma\sqrt{2\log n}$. In practice, $\sigma$ can be easily estimated by computing the Median Absolute Deviation (MAD) of $2^{1/2}|X_{i+1} - X_i|_{i=1}^{n-1}$, in which a MAD of the finest scale Haar wavelet coefficients. The plots of the wavelet coefficients before and after thresholding are shown in Figure 5.5. The left column of Figure 5.5 shows the simulated data and wavelet coefficients before thresholding and the right column shows them after thresholding. Most of the wavelet coefficients at the finer level are set to zero by thresholding and only leave the coefficient which likely to relate to the true function. In this case, the locations of significant changes in the simulated data are captured by coefficients at scale 6.

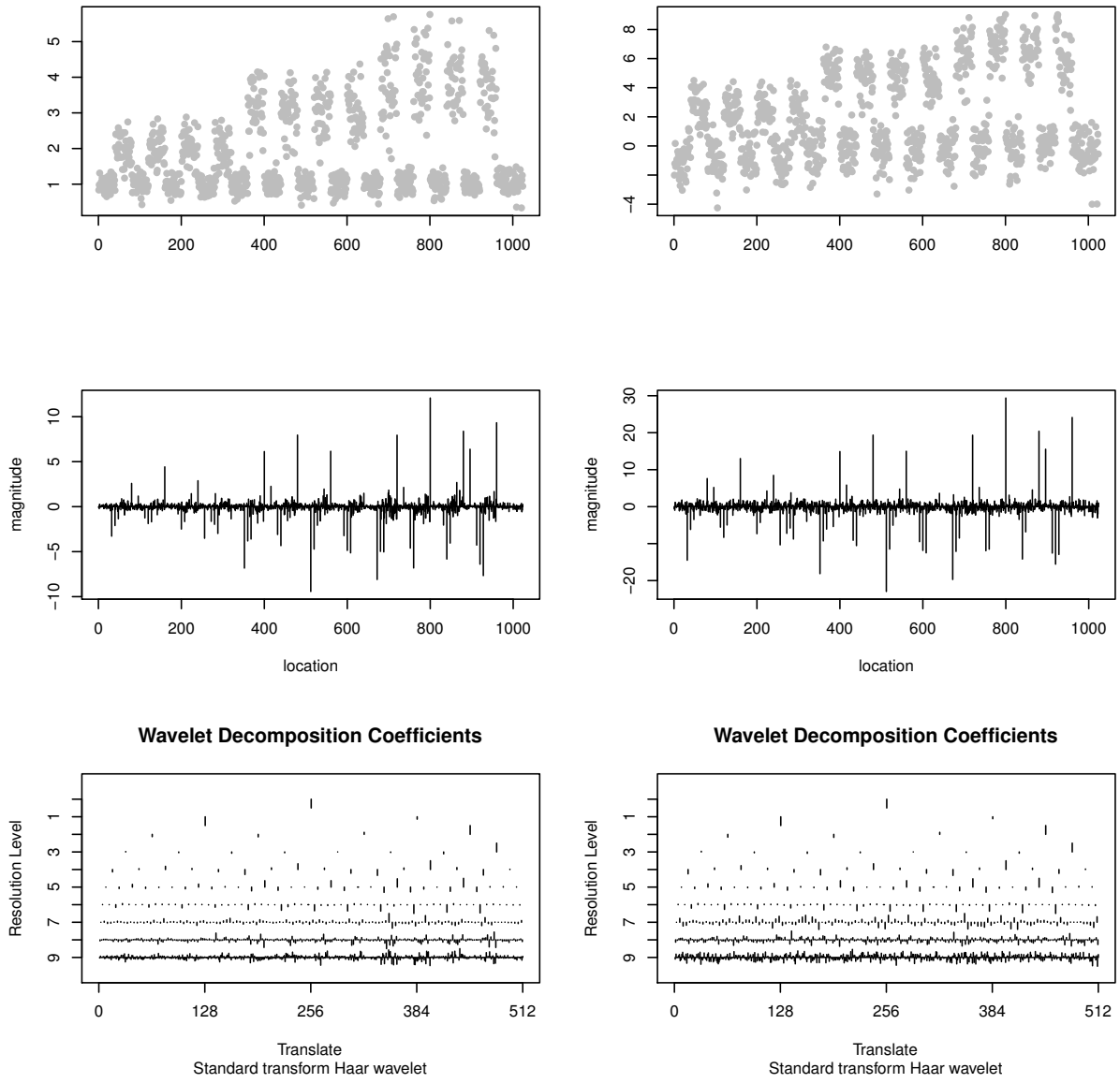


Figure 5.4: Illustration of the change of wavelet coefficients in variance stabilisation stage of DDHF method. Left hand column corresponds to the wavelet coefficients before stabilisation and right hand column to the coefficients after stabilisation. Top row: Simulated data before (left) and after (right) variance stabilisation by Haar-Fisz transform. Middle row: Plot of the magnitude of Haar wavelet coefficient before (left) and after (right) the Fisz transform. Bottom row: Plot of the magnitude of Haar wavelet coefficients with regard to their scale and location before (left) and after (right) the Fisz transform.

But due to the structure of Haar transformation, the coefficients at coarser scales (greater than 6) whose corresponding wavelet basis overlaps with the location of those significant changes location also survived the thresholding. Each of these survived coefficients will result in a single change-points. This cause the tendency of the DDHF method to overestimate change-points by forming many additional change-points at the dyadic location.

After the piecewise estimation for stabilised data is obtained, the last procedure is done by performing the inverse Haar-Fizs transform to bring back the segmentation to fit the original simulated data. This can be done by applying Haar transform to the stabilised data estimate, undoing the Fisz transform procedure by remultiplying the wavelet coefficients by the estimated local standard deviation relate to those coefficients, and then performing the inverse Haar wavelet transform. Figure 5.6 shows the illustration of the estimation before (left column) and after (right column) the inverse Haar-Fisz transform. The top left panel of Figure 5.6 shows the final result of the DDHF method. Each of change-points estimated from the denoising step is isolated which causes the final estimator to have the exact same number of change-points.

### 5.5.2 DDHF Method Using TGUHm Wavelet Shrinkage (DDHF+T)

As in the previous subsection, here, the DDTF+T method is also divided into three main stage: (i)variance stabilisation stage, (ii)denoising stage, and (iii)reconstruction stage. The illustration of the first stage of DDHF+T method is exactly same as DDHF method, and is shown in Figure 5.4. The difference over DDHF method is located in the second stage, here TGUHm denoising is performed and the illustration of the change of wavelet coefficients is presented in Figure 5.7. The second row of Figure 5.7 shows the TGUH coefficients before (left) and after (right) thresholding. Most of the small coefficients that correspond to the noise were thresholded and only leave those whose values are high and likely to relate to the true change. Compare to the Haar wavelet threshold in Figure 5.5, the TGUHm shrinkage gives a better estimation in terms of estimating the change-point number and location.

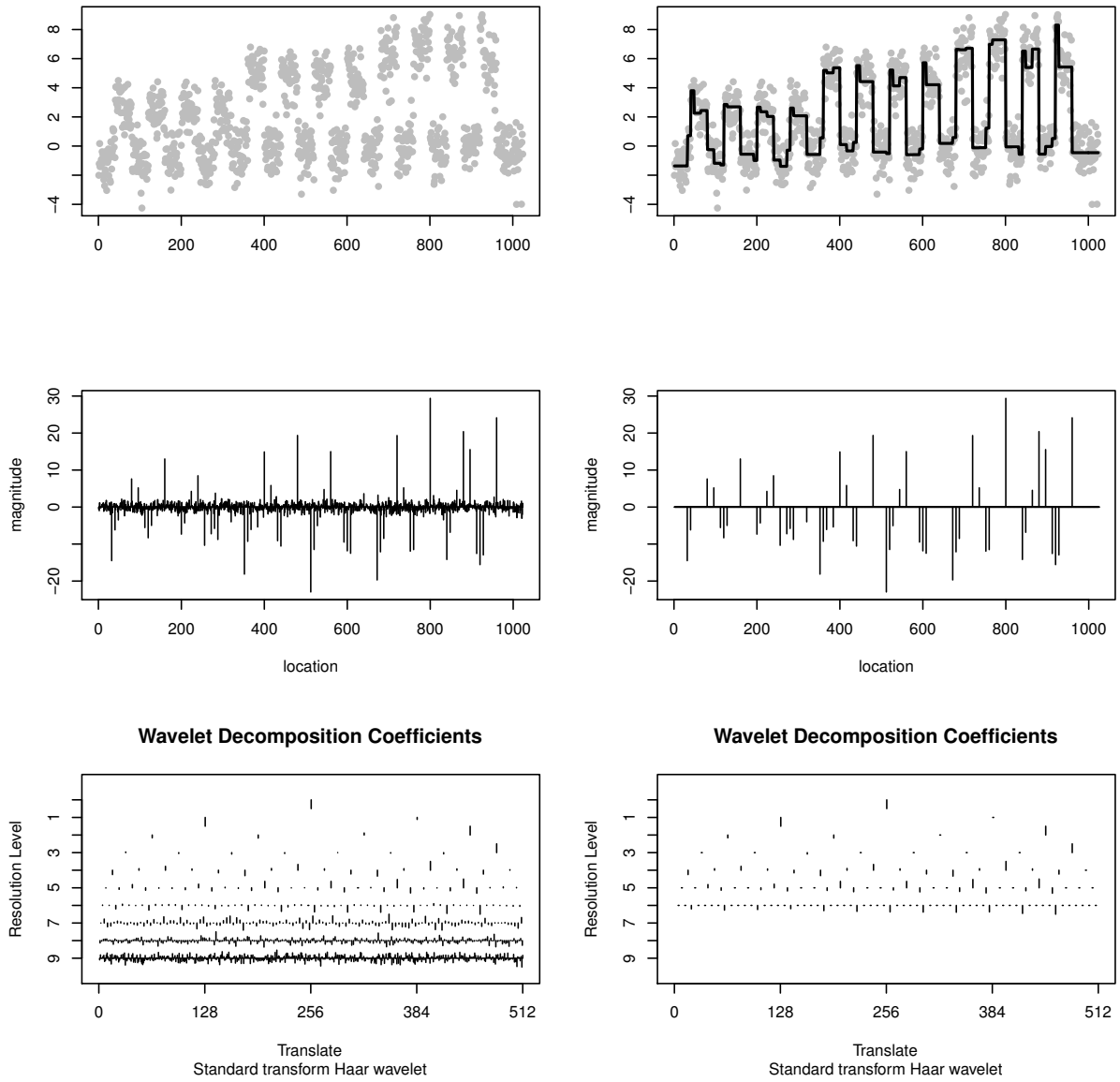


Figure 5.5: Illustration of the change of wavelet coefficients in denoising stage of DDHF method. Left-hand column corresponds to the wavelet coefficients before denoising and the right-hand column to the coefficients after denoising. Top left: Simulated data after variance stabilisation by Haar-Fisz transform before denoising. Top right: Black solid line denote the denoising result. Middle row: Plot of the magnitude of Haar wavelet coefficient before (left) and after (right) the denoising. Bottom row: Plot of the magnitude of Haar wavelet coefficients with regard to their scale and location before (left) and after (right) the denoising.

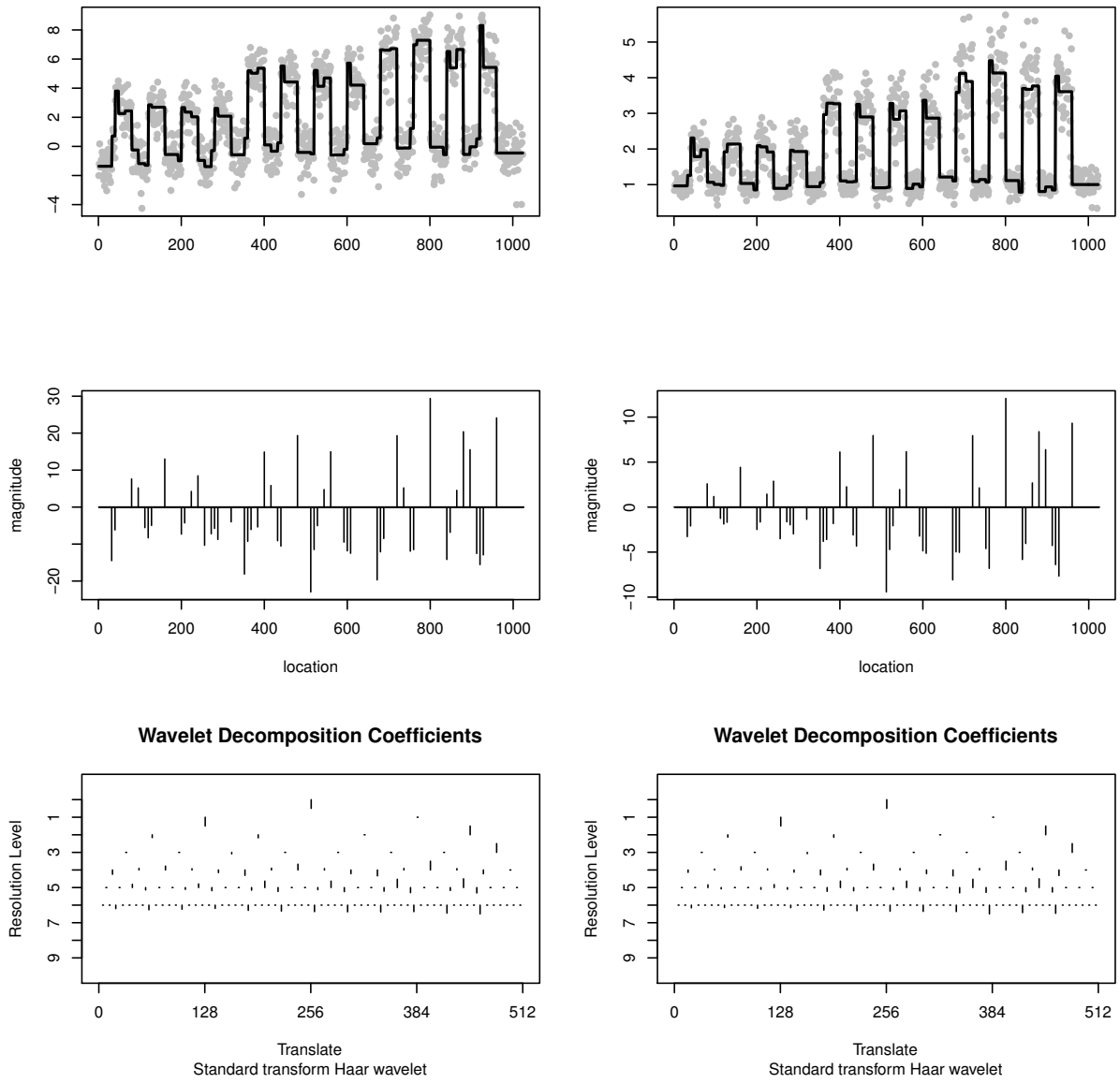


Figure 5.6: Illustration of the change of wavelet coefficients in reconstruction stage of DDHF method. The left-hand column corresponds to the wavelet coefficients before reconstruction and right-hand column to the coefficients after reconstruction (final estimator). Top left: Simulated data after variance stabilisation. Black solid line denote the denoising result. Top right: Black solid line DDHF estimate result after reconstruction. Middle row: Plot of the magnitude of Haar wavelet coefficient before (left) and after (right) the inverse Haar-Fisz transform. Bottom row: Plot of the magnitude of Haar wavelet coefficients with regard to their scale and location before (left) and after (right) the inverse Haar-Fisz transform.

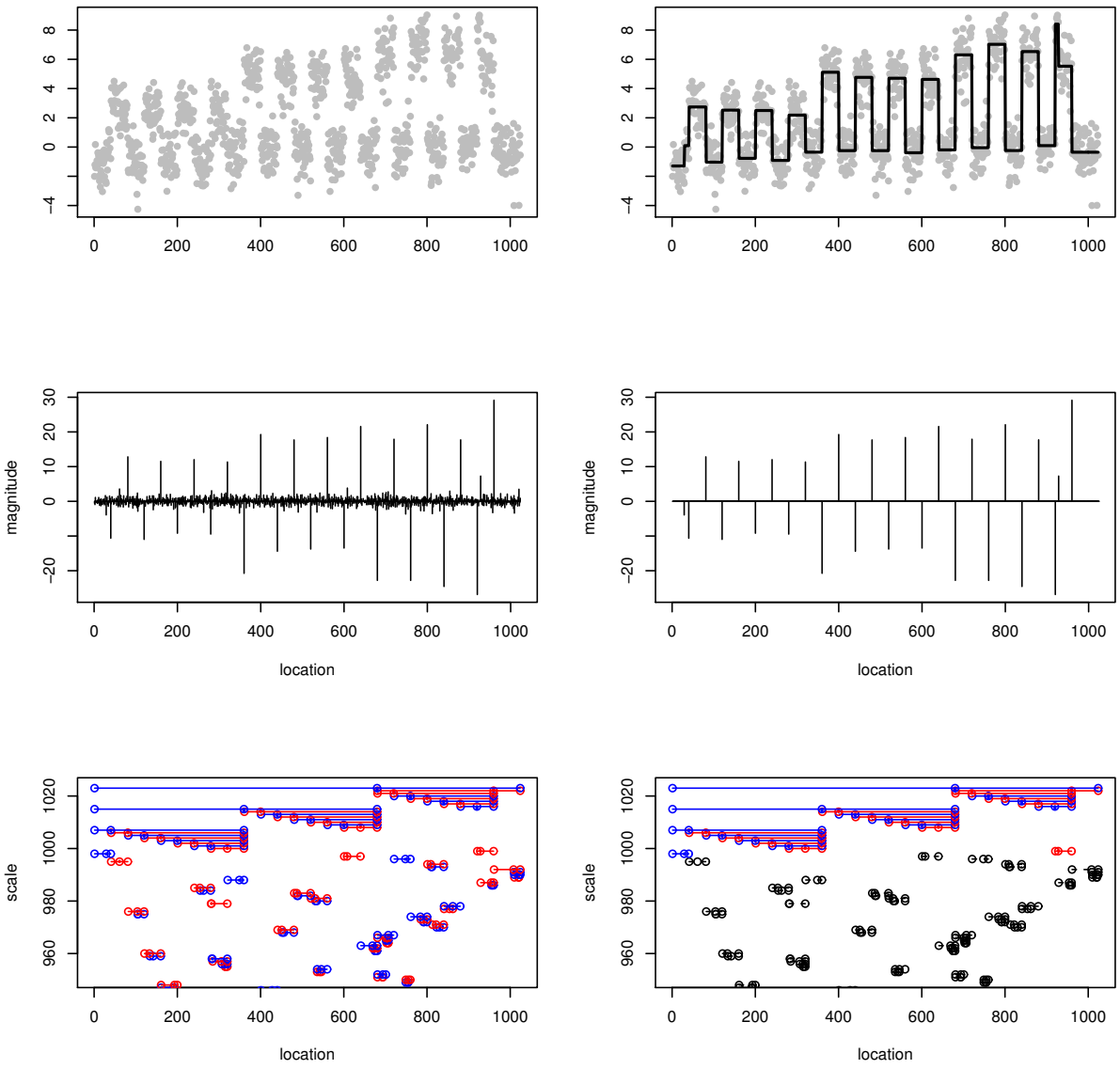


Figure 5.7: Illustration of the change of wavelet coefficients in denoising stage of DDHF+T method. Left hand column corresponds to the wavelet coefficients before denoising and right hand column to the coefficients after denoising. Top left: Simulated data after variance stabilisation by Haar-Fisz transform before the denoising. Top right: Black solid line denote the denoising result using TGUH shrinkage. Middle row: Plot of the magnitude of Haar wavelet coefficient before (left) and after (right) the denoising. Bottom row: Plot of the magnitude of Haar wavelet coefficients with regard to their scale and location before (left) and after (right) the denoising.

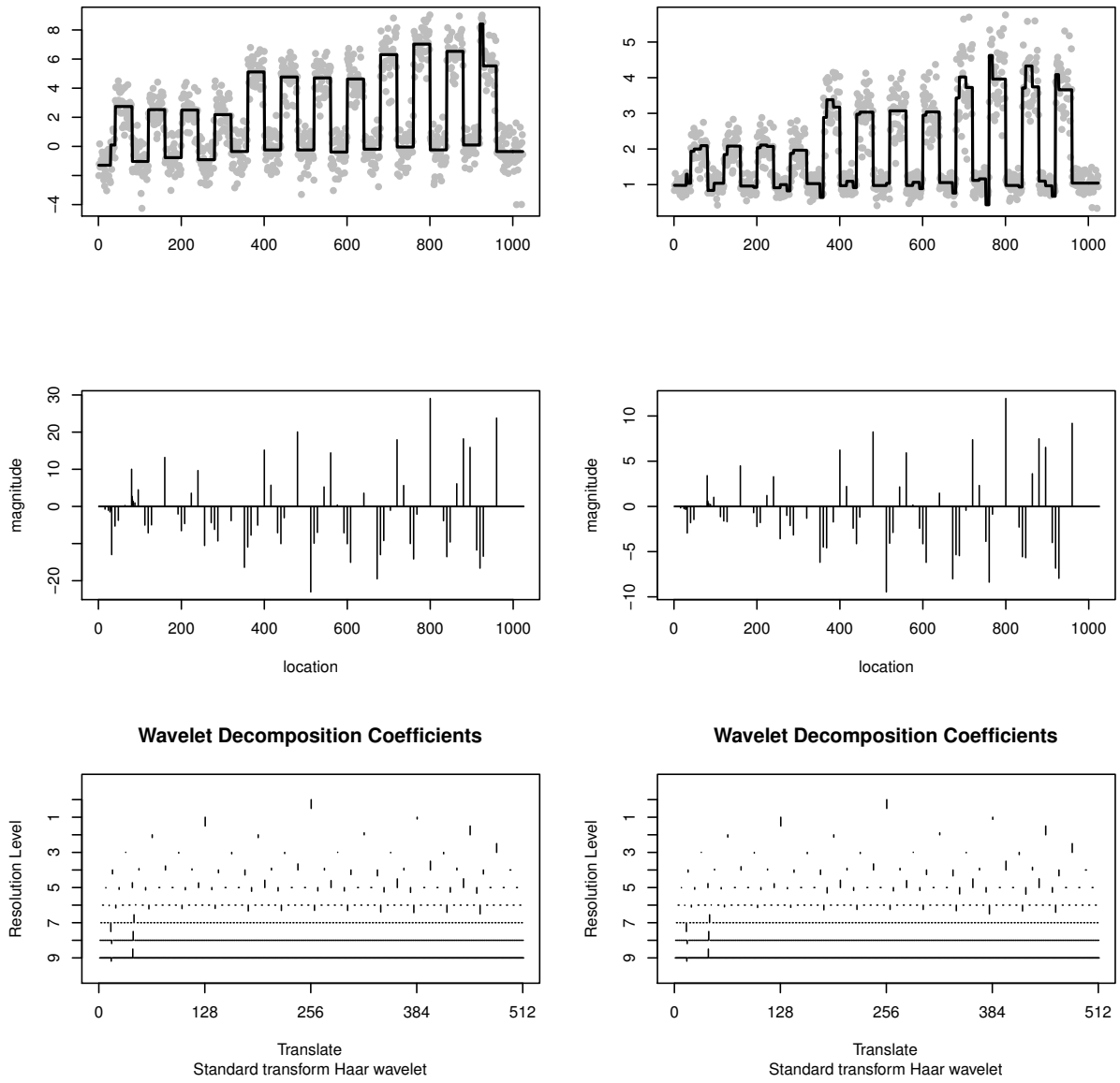


Figure 5.8: Illustration of the change of wavelet coefficients in reconstruction stage of DDHF+T method. Left hand column corresponds to the wavelet coefficients before reconstruction and right hand column to the coefficients after reconstruction (final estimator). Top left: Simulated data after variance stabilisation and the denoising stage. Black solid line denotes the denoising result. Top right: black solid line denotes DDHF+T estimate (final estimator). Middle row: Plot of the magnitude of Haar wavelet coefficient before (left) and after (right) the inverse Haar-Fisz transform. Bottom row: Plot of the magnitude of Haar wavelet coefficients with regard to their scale and location before (left) and after (right) the inverse Haar-Fisz transform.

After the TGUHm estimate of the stabilised data is obtained, the inverse Haar-Fisz transform is performed to reconstruct the result. This is done by first, taking the Haar wavelet transform to the TGUHm estimate, then remultiplying the wavelet coefficients by the estimated local standard deviation, and lastly performing the inverse Haar wavelet transform. Due to the dyadic structure of the Haar wavelet transform, each of the changes (jumps/drops) in the TGUHm estimate will be represented by one or more Haar wavelet coefficients. Only if the change is located exactly in the middle of the data, it will be represented by only a coefficient, otherwise, several coefficients will carry out this information. This will result in more change-points found in the final estimator as shown in the top right panel of Figure 5.8.

### 5.5.3 Data-Driven TGUH-Fisz (DDTF) Method

In order to address the problem that arise in DDHF and DDHF+T method, instead of only changing the shrinkage procedure with TGUHm denoising, the DDTF method directly applies the variance stabilisation via Fisz transform in TGUH wavelet domain.

DDTF method starts by performing TGUH transform into the simulated data. The resulting TGUH coefficients are shown in the middle and bottom left panel of Figure 5.9. One can see that the TGUH coefficients related to the high variability region have higher values. Then the Fisz transform is applied to these coefficients to stabilise the variance. The right column of Figure 5.9 shows the simulated data and TGUH coefficient after Fisz transform. Now, the coefficients that correspond to the noise are more uniform while all the large/significant coefficients are still prominent.

The next step is denoising. Similar to the DDHF+T method, two-stage thresholding with a universal threshold is used. By this thresholding procedure, most of the small coefficients corresponding to the noise are removed/set to zero leaving the large coefficients which are likely related to the true changes. Figure 5.11 illustrates the simulated data and TGUH coefficients before and after thresholding. The black solid line in the top right panel of Figure 5.10 illustrates the piecewise pattern that could be produced by applying inverse TGUH transform

to the survived coefficients after the thresholding stage. Note that this is just for illustration as in the DDTF method, it does not need to bring back the domain from wavelet to 'time' domain.

The final step is the reconstruction stage. The final estimator of $f_i$ is obtained by setting the value of the signal between two consecutive breakpoints to be the average of all copy number ratio data over that interval. The illustration of the change of wavelet coefficients in the reconstruction stage of the DDTF method is presented in Figure 5.11. Unlike the DDHF+T method, each of the survived coefficients from the thresholding stage only corresponds to a single change-point in the final estimator, hence it will result in cleaner estimation.

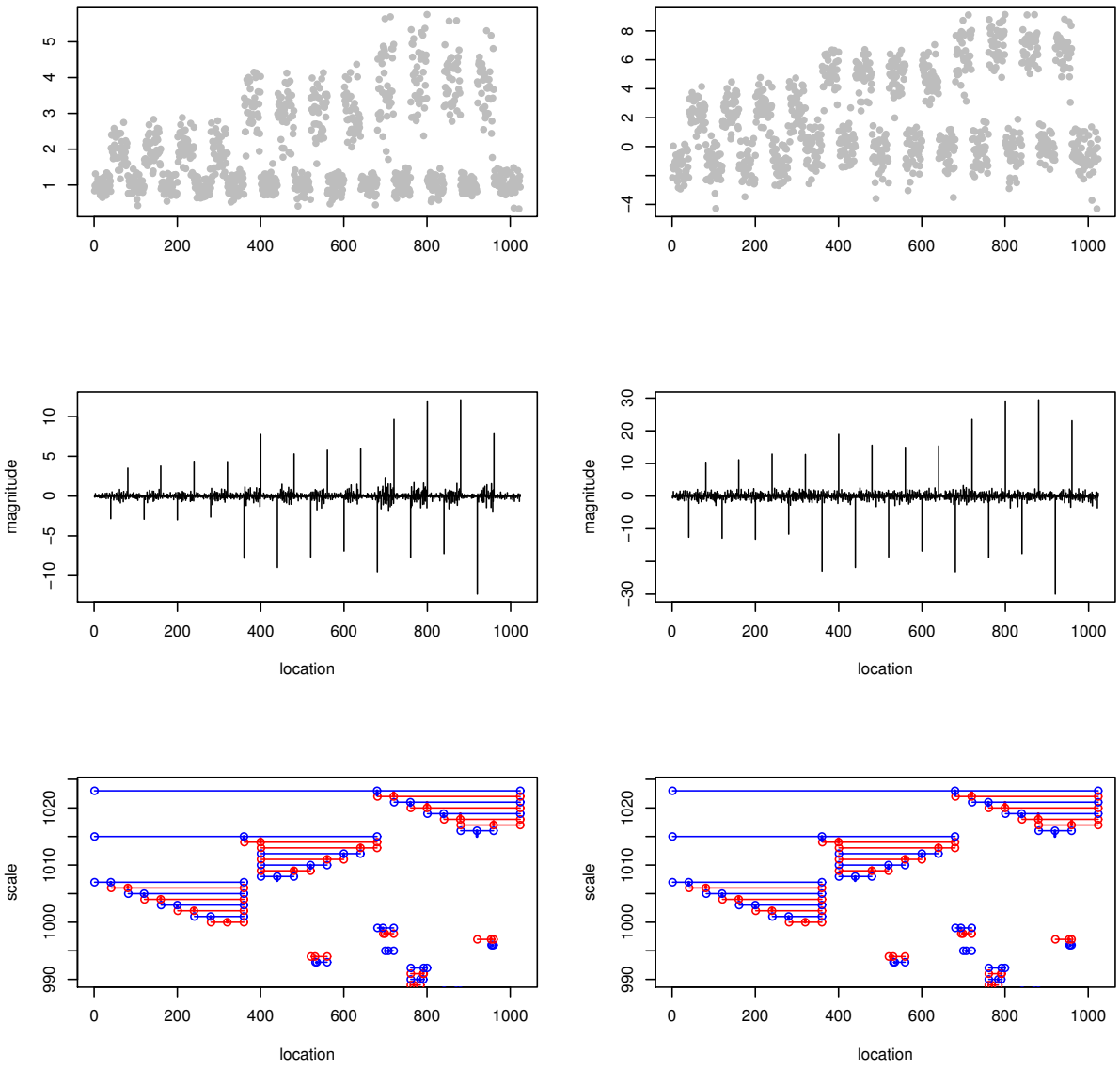


Figure 5.9: Illustration of the change of wavelet coefficients in variance stabilisation stage of DDTF method. Left hand column corresponds to the wavelet coefficients before variance stabilisation and right hand column to the coefficients after variance stabilisation. Top left: Simulated data. Top right: Simulated data after variance stabilisation by TGUH-Fisz transform. Middle row: Plot of the magnitude of TGUH wavelet coefficient before (left) and after (right) the Fisz transform. Bottom row: Plot of the magnitude of TGUH wavelet coefficients with regard to their scale and location before (left) and after (right) the Fisz transform.

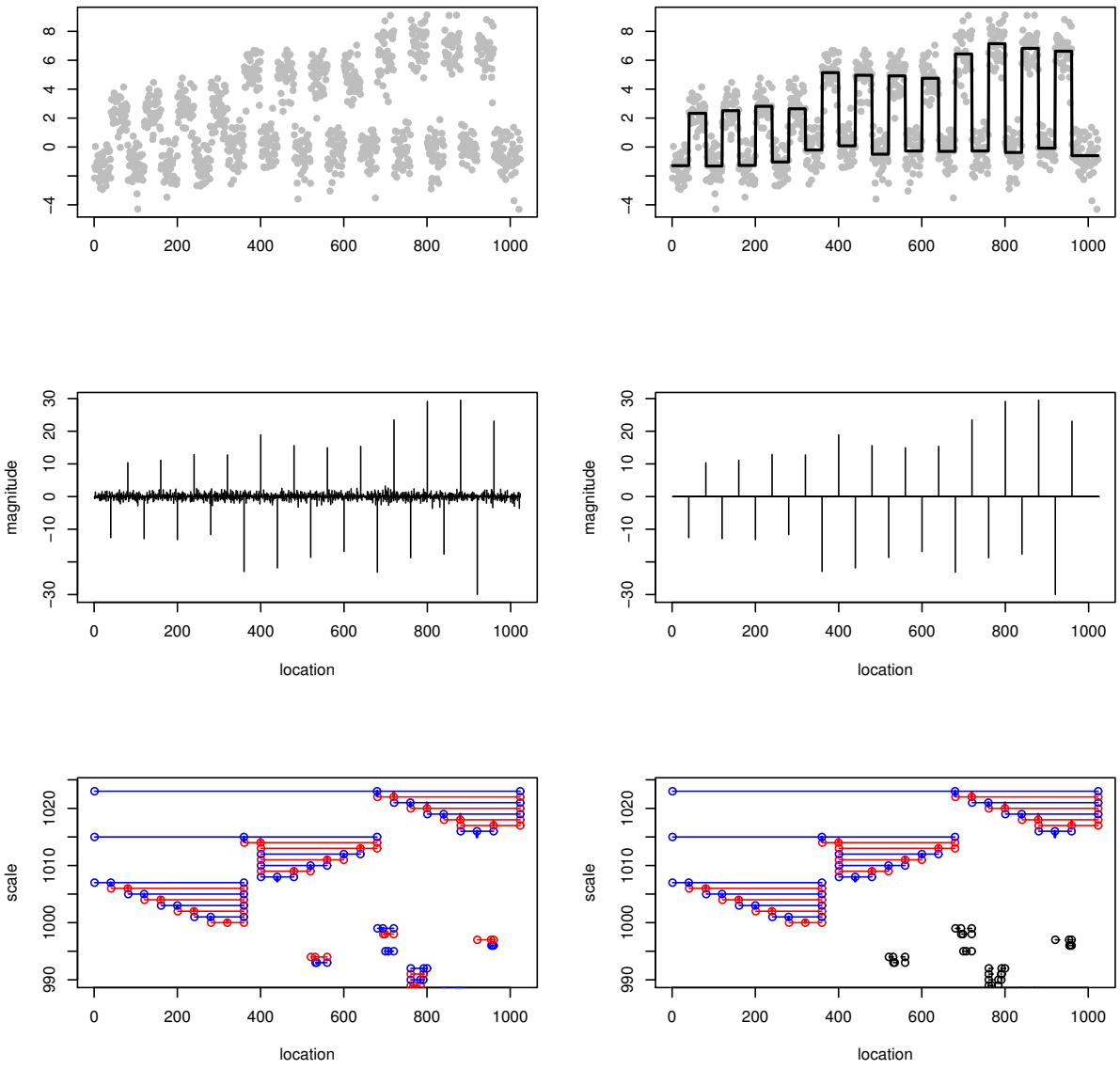


Figure 5.10: Illustration of the change of wavelet coefficients in denoising stage of DDTF method. Left hand column corresponds to the wavelet coefficients before denoising and right hand column to the coefficients after denoising. Top left: Simulated data after variance stabilisation by TGUH-Fisz transform before the denoising. Top right: Black solid line denote the denoising result using two-stage thresholding. Middle row: Plot of the magnitude of TGUH wavelet coefficient before (left) and after (right) the denoising. Bottom row: Plot of the magnitude of TGUH wavelet coefficients with regard to their scale and location before (left) and after (right) the denoising.

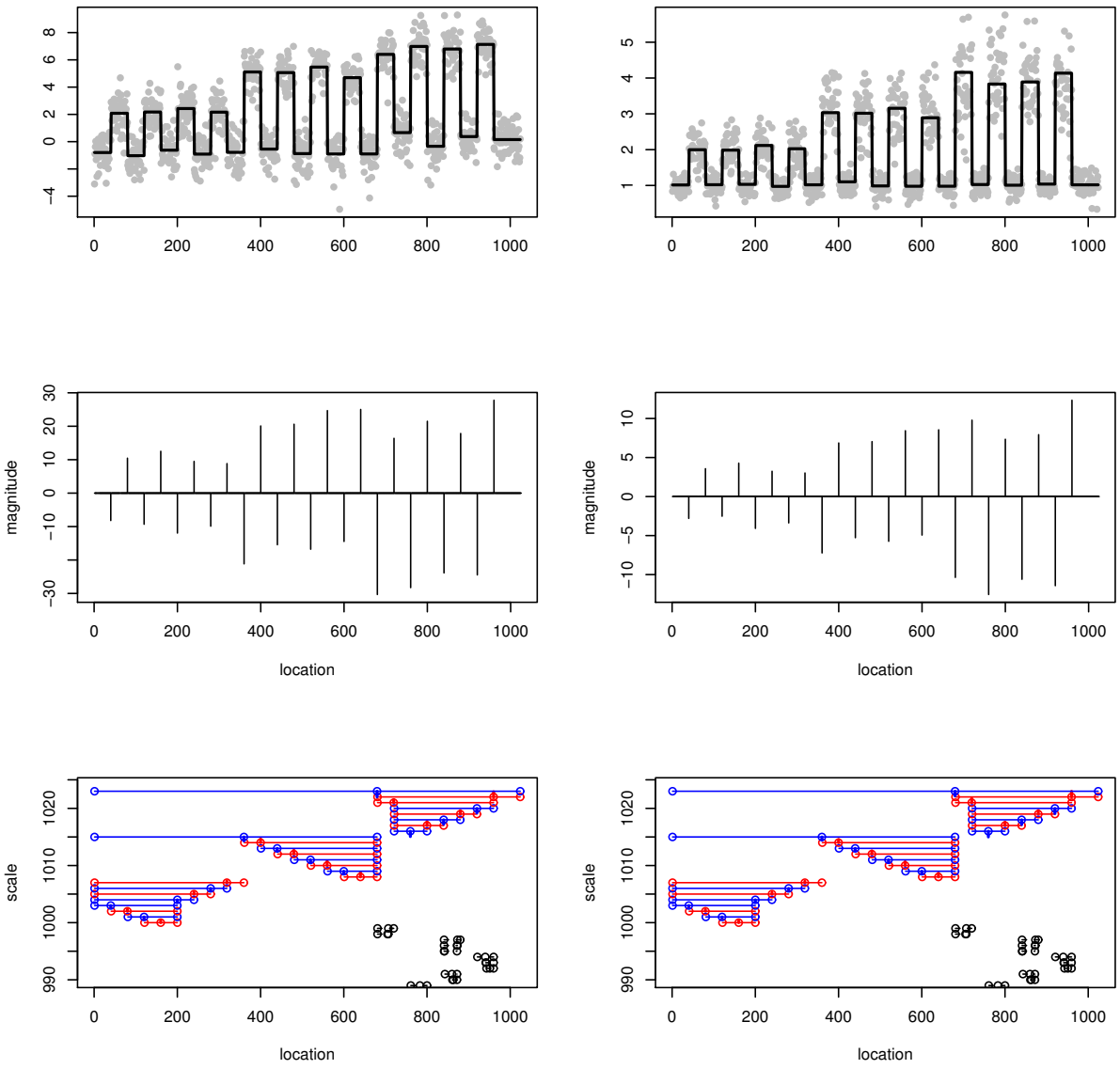


Figure 5.11: Illustration of the change of wavelet coefficients in reconstruction stage of DDTF method. Left hand column corresponds to the wavelet coefficients before reconstruction and right hand column to the coefficients after reconstruction (final estimator). Top left: Simulated data after variance stabilisation and the denoising. Black solid line denote the denoising result. Top right: DDTF estimate result. Middle row: Plot of the magnitude of TGUH wavelet coefficient before (left) and after (right) reconstruction procedure. Bottom row: Plot of the magnitude of Haar wavelet coefficients with regard to their scale and location before (left) and after (right) the reconstruction procedure

## 5.6 Simulation Study

To understand the operating characteristics of the proposed methodology, a simulation study using four kinds of test functions as shown in Figure 5.12 was conducted. The first test function includes both of long segments and short segments to evaluate the performance of the proposed method in estimating both of those altered segments. Short segments are defined as aberrations with length between 6–10 data points since, in our real data, a 1 Mb segment is represented by only 6-7 windows or data points. The second function only contains short altered segments with various aberrations height to assess the sensitivity of the methods toward short altered segments. The third one is a simple repetitive pattern with various segment height which aims to assess the ability of the method in estimating altered segment with different height at regular location. The last test function is similar to the third one but the change points are set to be located at dyadic locations.

The simulated datasets were generated from the model $r_i = f_i + \varepsilon_i$ where $\epsilon_i$ is a random error term that follows the assumption described earlier in Section 5.3. Two kinds of noise types were considered; additive i.i.d Gaussian noise $N(0, \sigma^2)$ and a mixture of two normal distributions. The mixture of normal distribution noise is considered to illustrate the extreme values that are commonly found on copy number data (Nilsen *et al.*, 2012); with probability $1 - \alpha$ the error is drawn from a distribution $N(0, \sigma^2)$, and with probability $\alpha$ from $N(0, d^2\sigma^2)$, typically with $d = 3$ and $\alpha = 0.05$.

For both noise types, the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ to represent the mean-variance relation in NGS copy number data; several noise level is considered $\sigma_0 = 0.1, 0.2, 0.3, 0.4,$ and $0.5$. Here as one of the interests is to evaluate the performance of the method deal with a very noisy signal, where the human eye is not of much help in estimating the true signal and a reliable automatic statistical technique becomes important. As shown in the second and third rows of Figure 5.13, it is difficult to identify short altered segments underlying the noisy signal only by eyes even for the lowest noise level that we consider ($\sigma_0 = 0.1$, especially those with the smallest change (0.5)). This setting

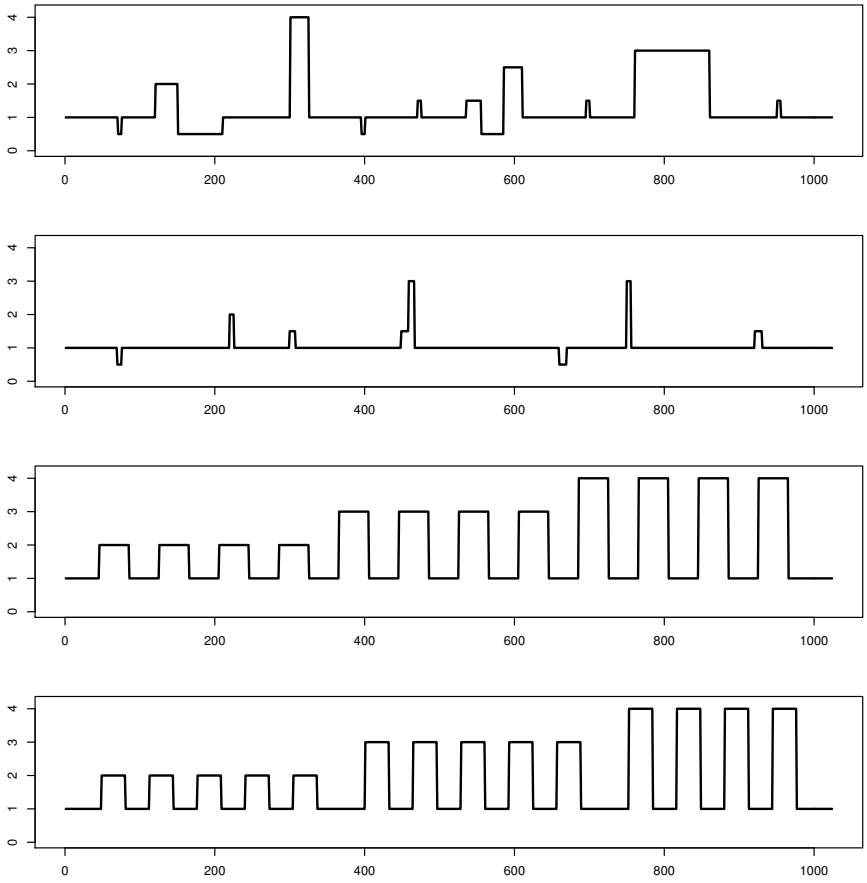


Figure 5.12: The true patterns of copy number alterations, denoted $f$, in simulated examples. First row: First true function. The irregular pattern of segment length is based on common patterns observed in real data. Second row: Second true function, which aims to characterise the proposed method's performance in a case where the underlying true pattern only contains short altered segments. Third row: Third true function. A test function to assess the ability of the method in estimating altered segment with different height. Fourth row: Fourth true function, where contains repetitive pattern with change-points located at dyadic locations.

is considered to represent a difficult real data situation where the height difference between segments is almost at its limit.

To assess the performance of the proposed method, the sensitivity or true positive rate (TPR) and specificity or false positive rate (FPR) were computed. The TPR is the proportion of the correctly identified change point out of the $k$ true change points. While the FPR is the proportion of the spurious estimators or false positives out of the total length of the dataset minus $k$. The true positives (TP) or correctly identified change-points is defined as the number of estimated change-points that are located closest to the true change-point location and inside a given distance tolerance, $\delta = 2$. While, the number of false positives (FP) is defined as the remaining change points estimated, FP=P-TP, where P denotes positives or the total number of change points estimated. The illustration of these definitions is presented in Figure 3.2.

For illustration, an estimated change-point is classified as TP if it is located closest and at least two points to the right or left to the true change-point location. Based on this definition the average of true positive rate (aTPR) and the average false positive rate (aFPR) were computed over 1000 replicates. To evaluate the ability of the method in estimating short segments, the average true positive rate in estimating short segments (aTPRsh) was also calculated. The average mean squared error (aMSE) to the estimated piecewise constant signal and the true function also were reported to measure the similarity between the estimated and true segmentation.

To understand the operating characteristics of the proposed method, it is compared with some segmentation methods that have been used widely in the analysis of copy number such as Circular Binary Segmentation (CBS) (Olshen *et al.*, 2004), HaarSeg (Ben-Yaacov & Eldar, 2008), CopyNumber (Nilsen *et al.*, 2012) and FDRseg (Li *et al.*, 2016). For the CopyNumber method, same as in Chapter 4, the CopyNumber method is applied twice, with its main parameter $\gamma$ set to be 40 and 12 to give different balances between sensitivity and specificity as suggested in Nilsen *et al.* (2012). The results for these two separate analyses are denoted as Copy12 and Copy40, respectively. The TGUHm method introduced in Chapter 4 which has been proven as a powerful segmentation method that employs unbalanced Haar wavelets is also evaluated. Moreover, to emphasise

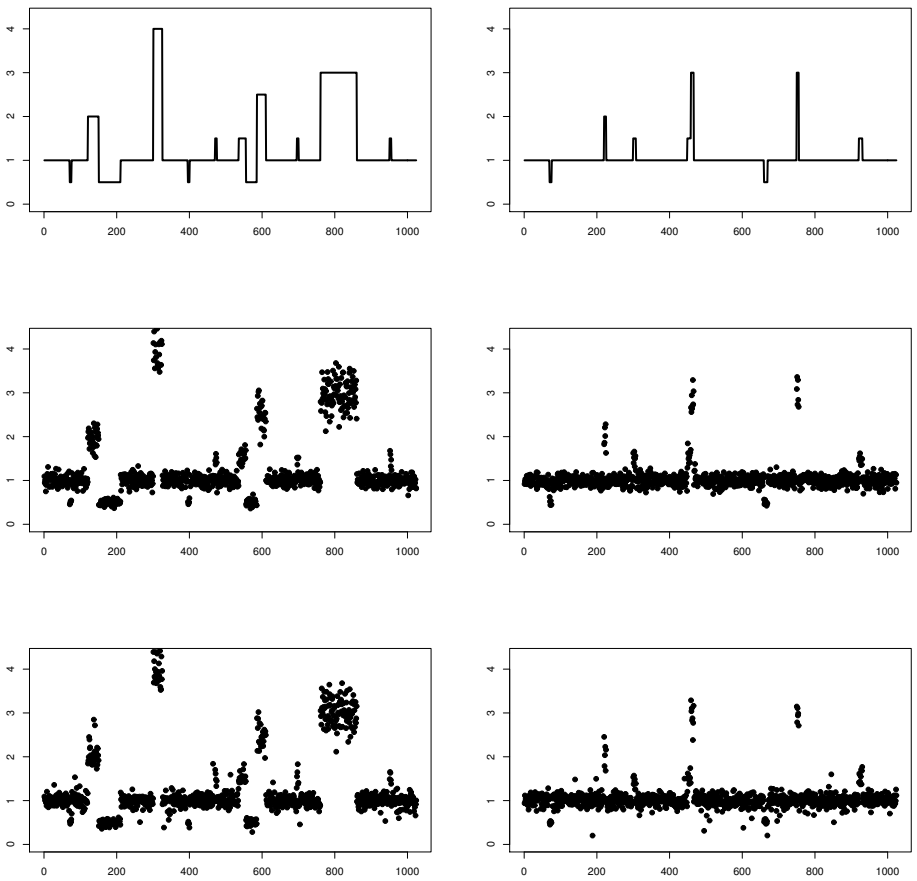


Figure 5.13: First row: The first (left panel) and second (right panel) type of test function denoted $f$. Second row: The simulated data contaminated with i.i.d Gaussian noise $N(0, \sigma_0^2)$ that corresponds to the first row test functions. Third row: The simulated data contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ that corresponds to the first row test function, where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$. This is an example for $\sigma_0 = 0.1$

the advantage of the use of TGUH wavelets compared to the 'balanced' Haar wavelets in variance stabilisation, the data-driven Haar-Fisz (DDHF) method (Fryzlewicz & Delouille, 2005) which performs variance stabilisation in 'balanced' Haar wavelet domain is also considered as comparison method.

The following list labels and describes three variation wavelet denoising of the DDHF methods considered in the simulation.

- **DDHF**: The DDHF method with the universal thresholding from Donoho & Johnstone (1994) which uses median absolute deviation (MAD) variance estimation on all coefficients. This is the default setting of DDHF method suggested in Fryzlewicz & Delouille (2005)
- **DDHF+T**: The DDHF method with the TGUHm denoising as explained in Section 4.3.
- **DDHF+B**: The DDHF method with the eBayes thresholding as described in Johnstone & Silverman (2005a).

### 5.6.1 Results

Figures 5.14, 5.15, 5.16, and 5.17 show the results of the average true positive rate in estimating correct change-points (aTPR) and in estimating those which are related to short segments (aTPRsh), the average false positive rate (aFPR) and mean square error (aMSE) over 1000 simulated data for different noise levels and types using first, second, third, and fourth type of true function, respectively. Based on Figure 5.14, for the first type of true function, DDTF shows excellent results in terms of aTPR and aTPRsh. At some noise levels, it is below DDHF+B and DDHF+T, but both DDHF+B and DDHF+T come with high aFPR and aMSE, which means that both of those methods have a tendency to produce many spurious change-points, especially for DDHF+B. Otherwise, DDTF is able to present relatively low aFPR and aMSE compared to most of the evaluated methods for both noise types used in the simulation.

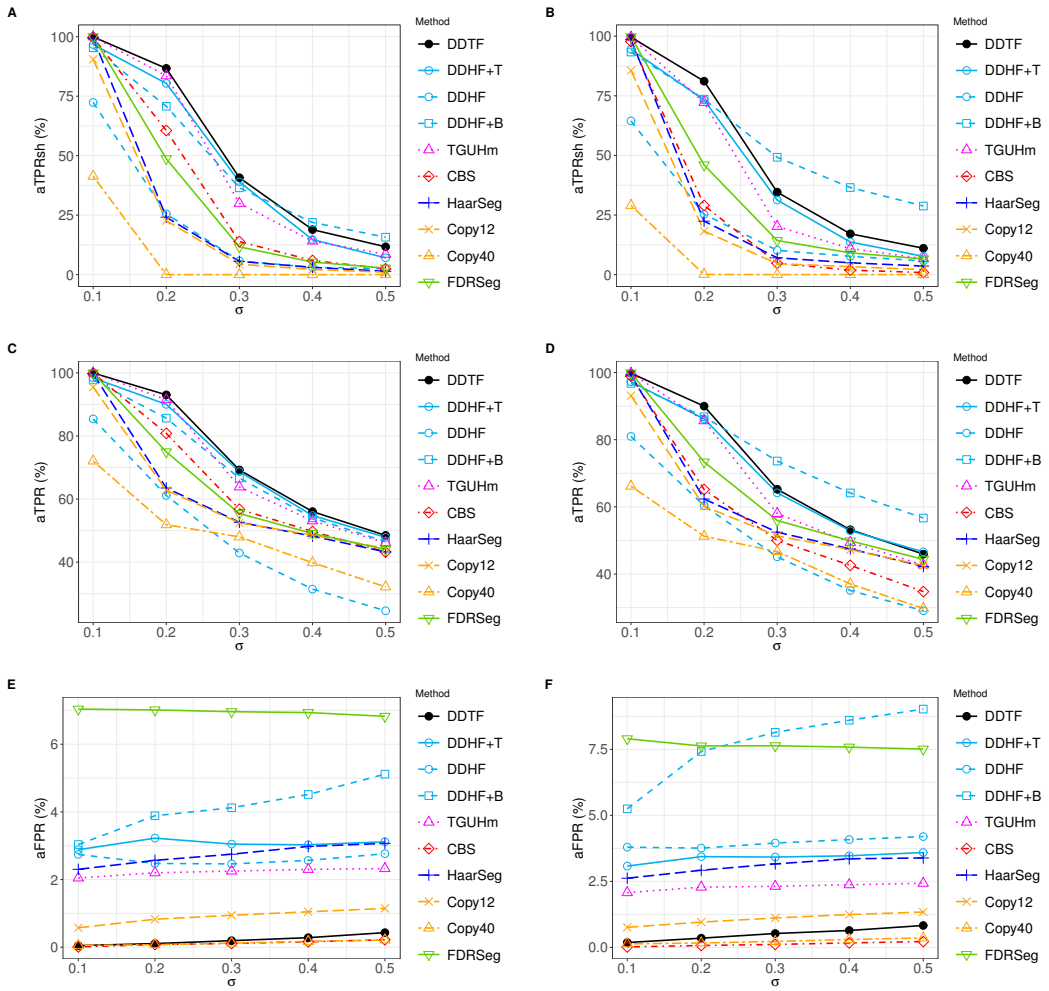


Figure 5.14: Performance metrics of the simulation based on first type of true function (see top panel of Figure 5.12). The left (**A**,**C**,**E**, and **G**) and right (**B**,**D**,**F**, and **H**) side corresponds to noise distribution used to contaminate the simulated data (left: i.i.d Gaussian noise $N(0, \sigma^2)$, right: a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$), where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$. **(A)** **(B)** Average of true positive rate in estimating change-points that corresponds to short segments (aTPRsh). **(C)** **(D)** Average true positive rate (aTPR). **(E)** **(F)** Average of false positive rate (aFPR). **(G)** **(H)** Average of mean-square error (aMSE) of the estimated piecewise constant signal to the true function. The average is taken over 1000 replicates.

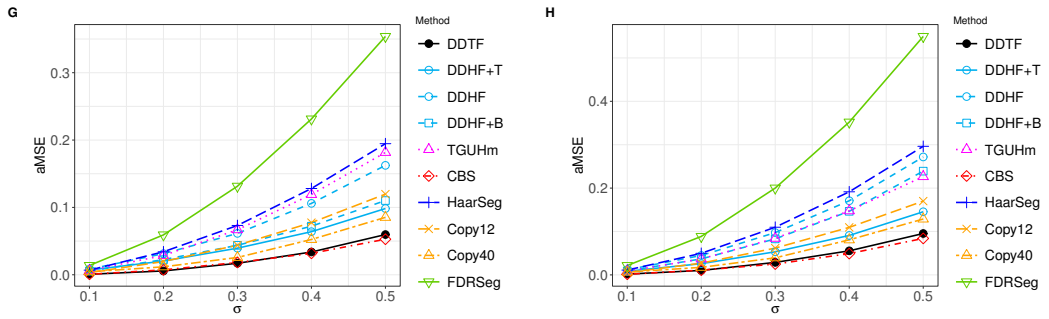


Figure 5.14: Continued.

For the simulation study using the second type of true function, DDHF+T is the best in terms of aTPR and aTPRsh but, same as the first simulation, it comes with high aFPR as shown in Figure 5.15. Even though DDTF slightly below DDHF+T in terms of aTPR and aTPRsh, its aFPR and aMSR are relatively low. Furthermore, an interesting point to see here is the results of DDTF and TGUHm are almost overlap for all of the performance metrics. This is happen due to the pattern of the test function which only contains short altered segments. In this particular example, DDTF and other data-driven methods only have very small information to estimate the function $h$, which gives a disadvantage to our proposed method. But even in this difficult situation, the performance of the DDTF method never gets worse than the TGUHm method.

For the results of the simulation study using the third type of true function (see Figure 5.16, DDTF, DDHF+T, TGUHm, HaarSeg, FDRSeg methods show an excellent result in terms of aTPR for both types of noise considered. But most of them are unable to present low aFPR and aMSE, only DDTF has good results in both metrics. An interesting point here is when the simulation is done using the true function whose shape is very similar to the third true function but all change points located at dyadic location, the aTPR or the sensitivity of the DDHF-based method (DDHF, DDHF+B, and DDHF+T) is much higher, as shown in Figure 5.17. This superiority comes from one of the unique characteristics of the 'balanced' Haar wavelet transform used where the discontinuity or breakpoints of the basis are always aligned at the dyadic location of the input data. But it comes with high aFPR since due to this characteristic, it also tends to estimate spurious change points at other dyadic locations.

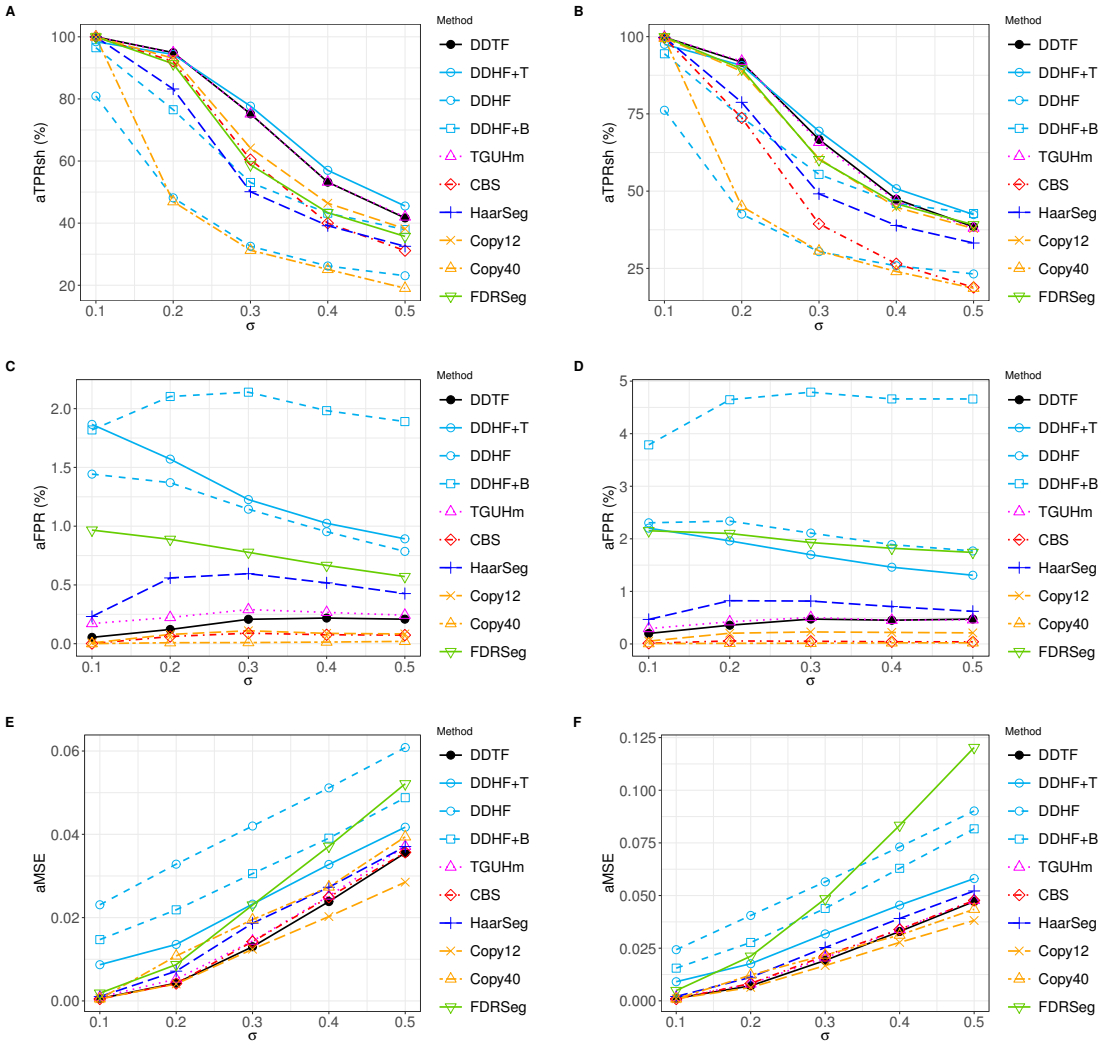


Figure 5.15: Performance metrics of the simulation based on second type of true function (see second row of Figure 5.12). The left (**A**,**C**,**E**, and **G**) and right (**B**,**D**,**F**, and **H**) side corresponds to noise distribution used to contaminate the simulated data (left: i.i.d Gaussian noise $N(0, \sigma^2)$, right: a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$), where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$. **(A)** **(B)** Average of true positive rate in estimating change-points that corresponds to short segments (aTPRsh). **(C)** **(D)** Average of false positive rate (aFPR). **(E)** **(F)** Average of mean-square error (aMSE) of the estimated piecewise constant signal to the true function. The average is taken over 1000 replicates. The aTPR results are omitted as the simulated data only contains an isolated short segment.

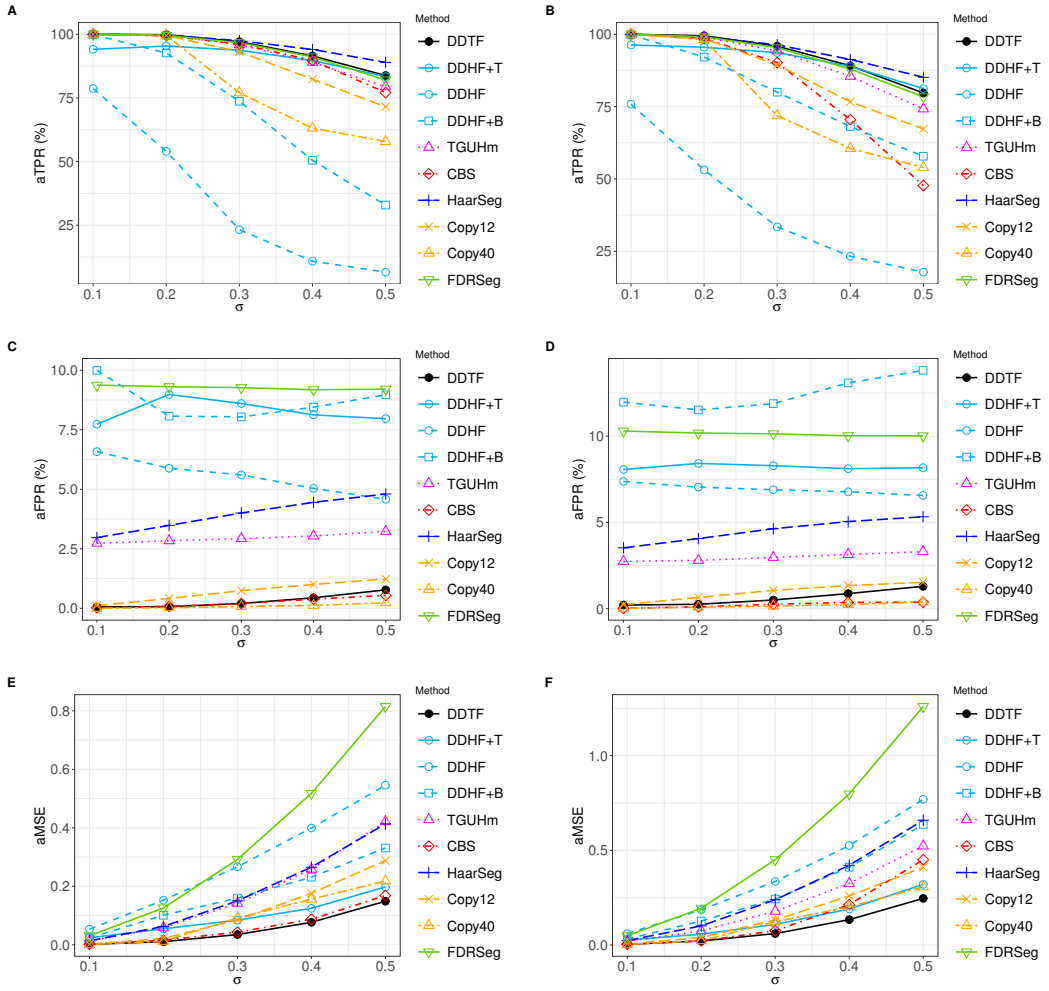


Figure 5.16: Performance metrics of the simulation based on third type of true function (see third row of Figure 5.12). The left (**A**,**C**,**E**, and **G**) and right (**B**,**D**,**F**, and **H**) side corresponds to noise distribution used to contaminate the simulated data (left: i.i.d Gaussian noise $N(0, \sigma^2)$, right: a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$), where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$. **(A)** **(B)** Average of true positive rate in estimating change-points that corresponds to short segments (aTPRsh). **(C)** **(D)** Average of false positive rate (aFPR). **(E)** **(F)** Average of mean-square error (aMSE) of the estimated piecewise constant signal to the true function. The average is taken over 1000 replicates. The aTPRsh results are omitted as the simulated data only contains long segments.

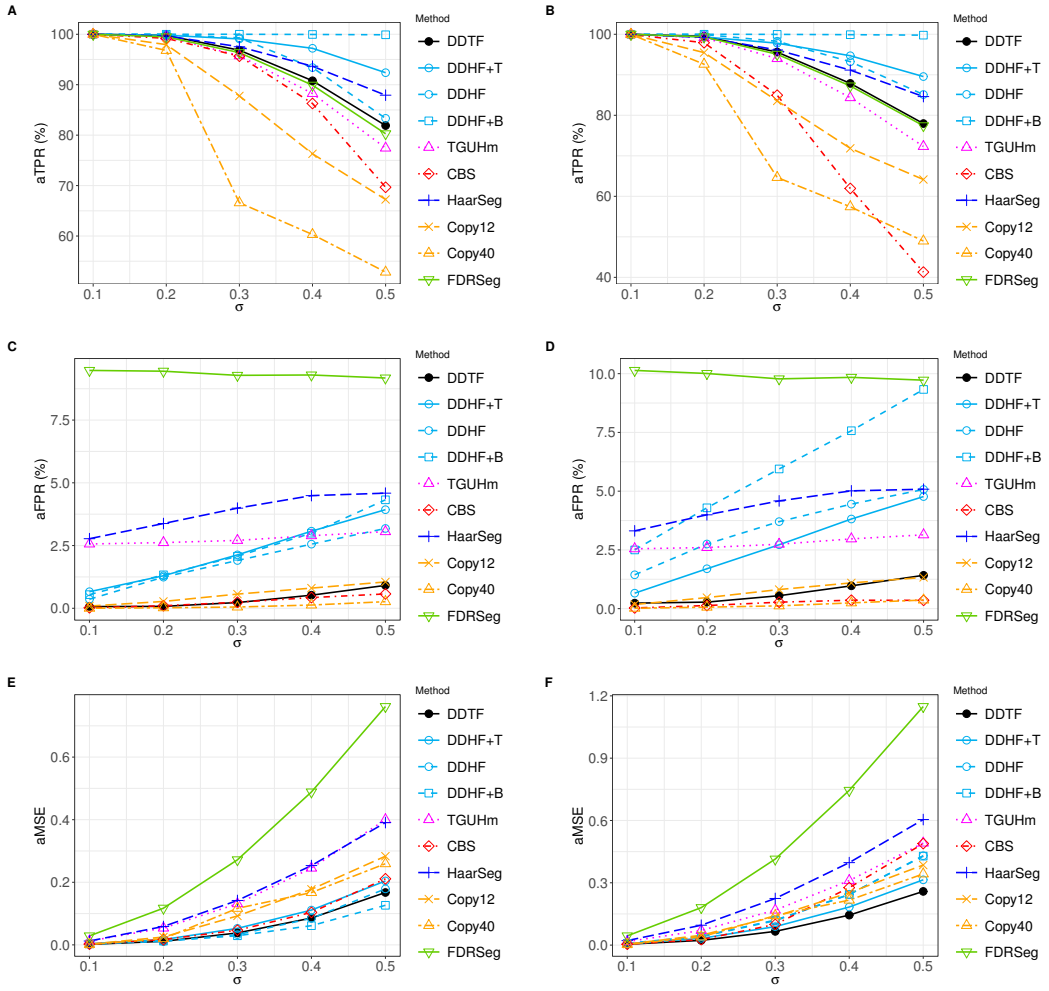


Figure 5.17: Performance metrics of the simulation based on fourth type of true function (see fourth row of Figure 5.12). The left (**A**,**C**,**E**, and **G**) and right (**B**,**D**,**F**, and **H**) side corresponds to noise distribution used to contaminate the simulated data (left: i.i.d Gaussian noise $N(0, \sigma^2)$, right: a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$), where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$. **(A)** **(B)** Average of true positive rate in estimating change-points that corresponds to short segments (aTPRsh). **(C)** **(D)** Average of false positive rate (aFPR). **(E)** **(F)** Average of mean-square error (aMSE) of the estimated piecewise constant signal to the true function. The average is taken over 1000 replicates. The aTPRsh results are omitted as the simulated data only contains long segments.

To further evaluate the operating characteristics of each method, the area under the curve (AUC) of the receiver operating characteristic (ROC) curve is also calculated for each segmentation method across different values of $\sigma^2$. Figure 5.18 reports the area under the ROC curve at each noise level. It shows that for both true functions, our proposed method is relatively better than most of the other methods for both of the noise types and all of the $\sigma_0$ levels.

A more careful inspection was done by plotting the proportion of estimated change-point against location as shown in Figure 5.19, 5.20, 5.21, and 5.22. The reason for the poor aFPR results of DDHF-based methods (DDHF, DDHF+B, and DDHF+T) is obviously seen here. All of Figure 5.19, 5.20, and 5.21 show that only change-points estimated by DDTF, CBS and Copy40 methods concentrated in the true locations while DDHF-based methods have a propensity to estimate change-points at some particular locations (dyadic location) and the remaining methods (TGUHm, HaarSeg, Copy12, and FDRSeg) tend to estimate false change-points at high segments (with higher variance). This tendency is clearly shown in Figure 5.21 and Figure 5.22, higher the mean level, most of the methods fail to present a clean segmentation. Here, only the plot for datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ is presented, where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.2$. But this behaviour also can be seen through all the noise levels evaluated. The results for other noise levels are shown in Appendix C.

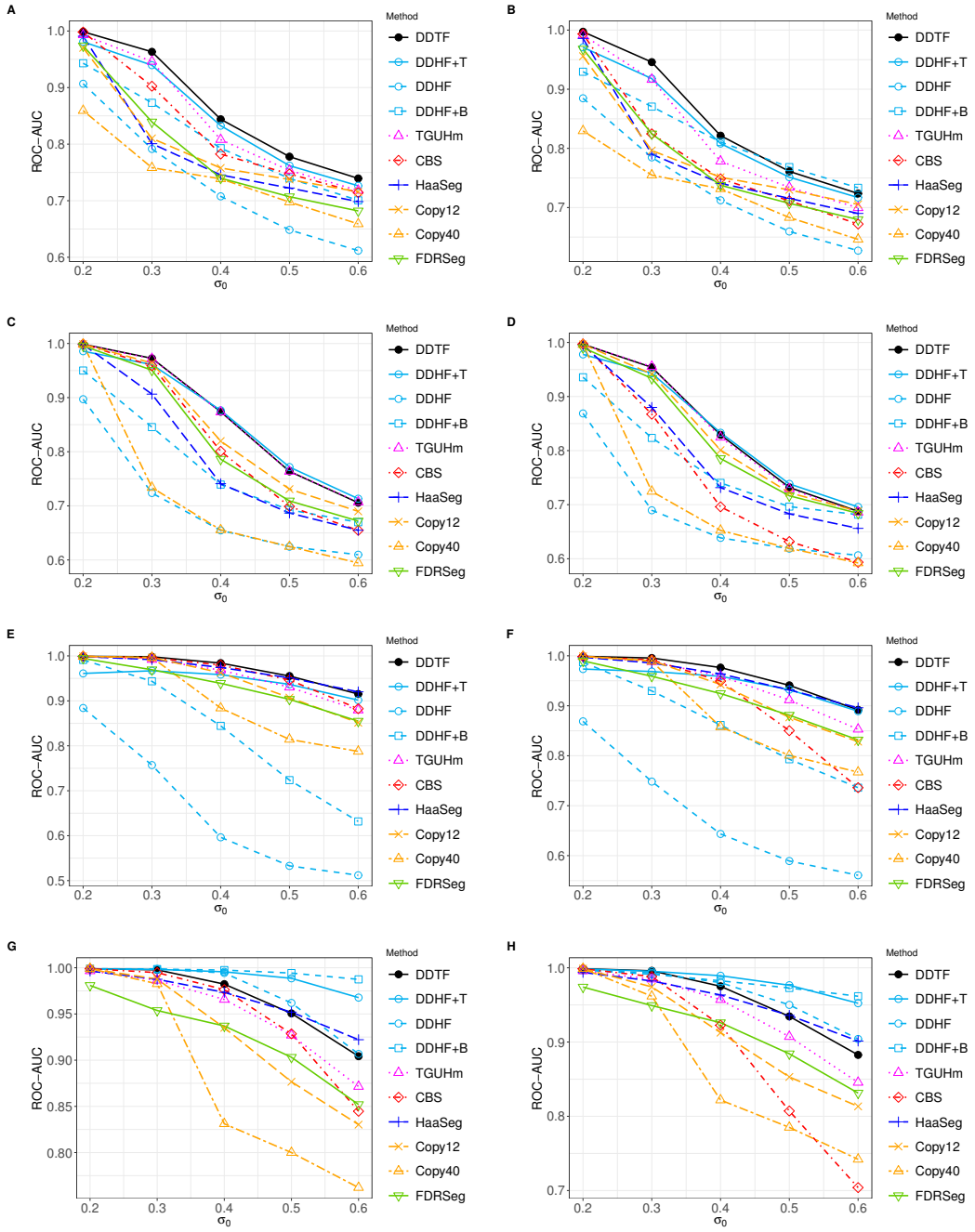


Figure 5.18: AUC of ROC of the methods correspond to the first type (first row), second type (second row), third type (third row), and fourth type (fourth row) simulated data. The left and right side corresponds to noise distribution used to contaminate the simulated data (left: i.i.d Gaussian noise $N(0, \sigma^2)$, right: a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$, where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$).

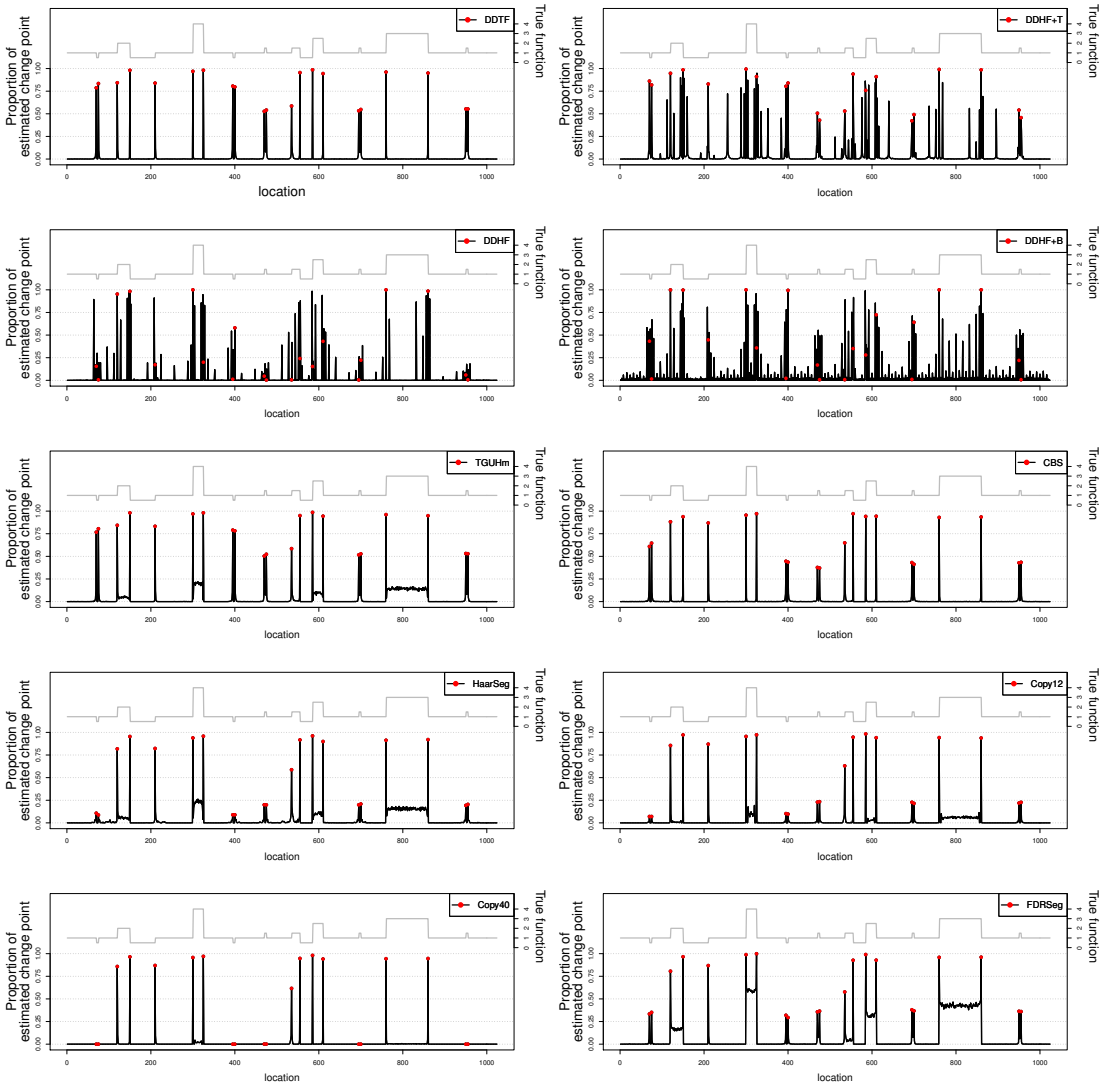


Figure 5.19: Proportion of times a change-point is estimated against location corresponds to the first test function (first row of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$, where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.2$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

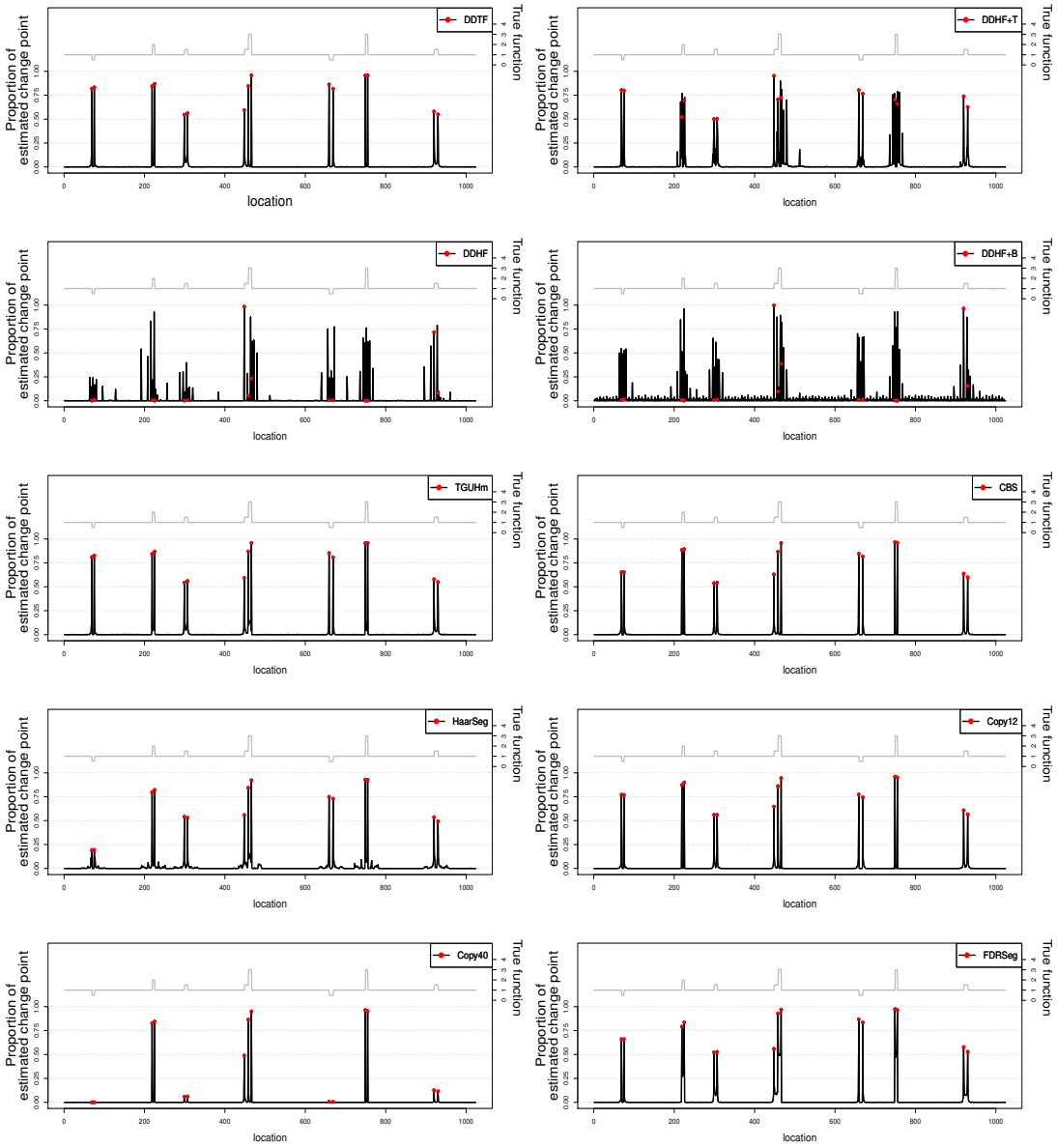


Figure 5.20: Proportion of times a change-point is estimated against location corresponds to the second test function (second row of Figure 5.12). Each value denotes the proportion of a change-point found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$, where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.2$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of the estimated change point and the corresponding test function's height, respectively.

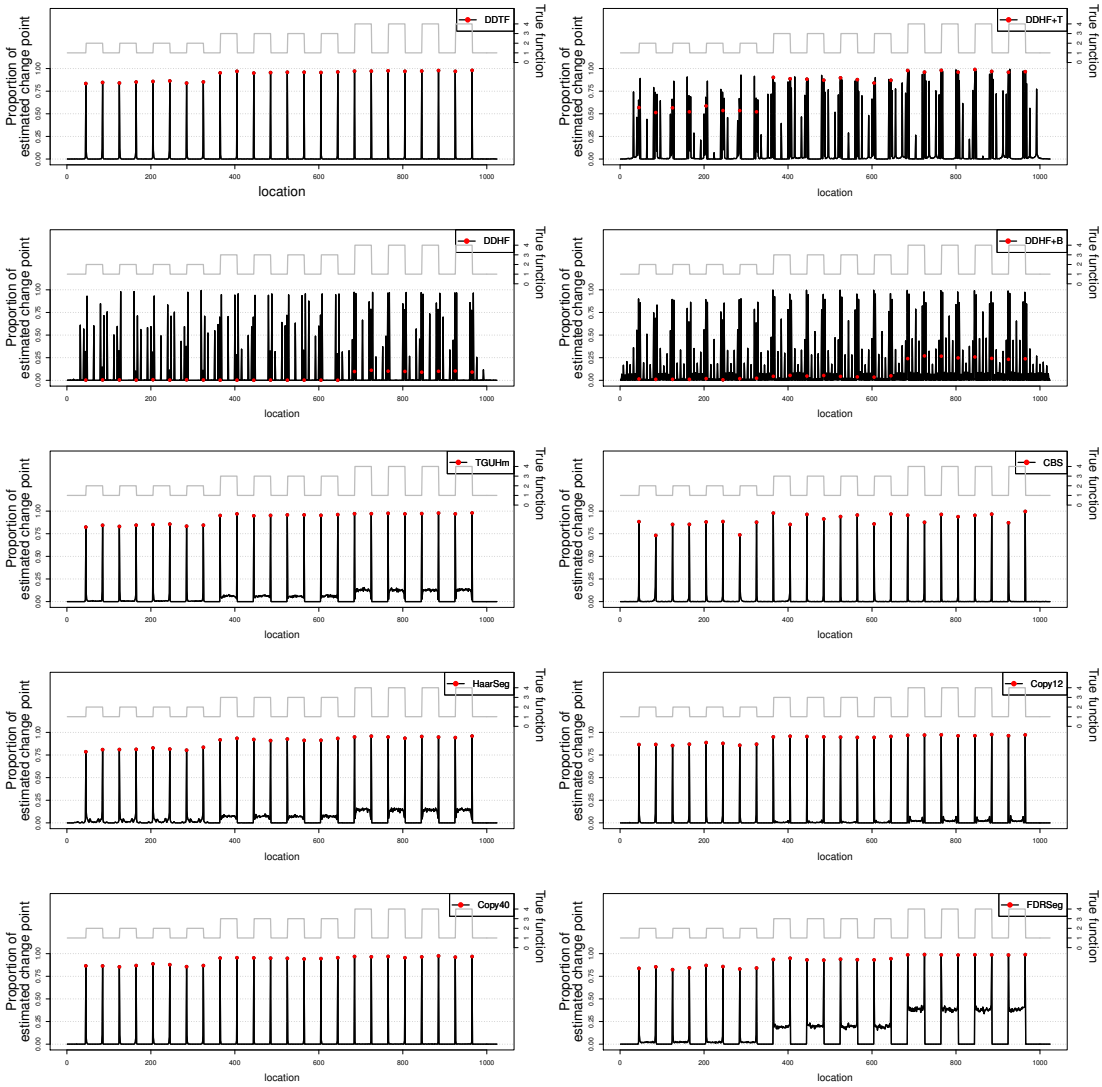


Figure 5.21: Proportion of times a change-point is estimated against location corresponds to the third test function (third row of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$, where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.2$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

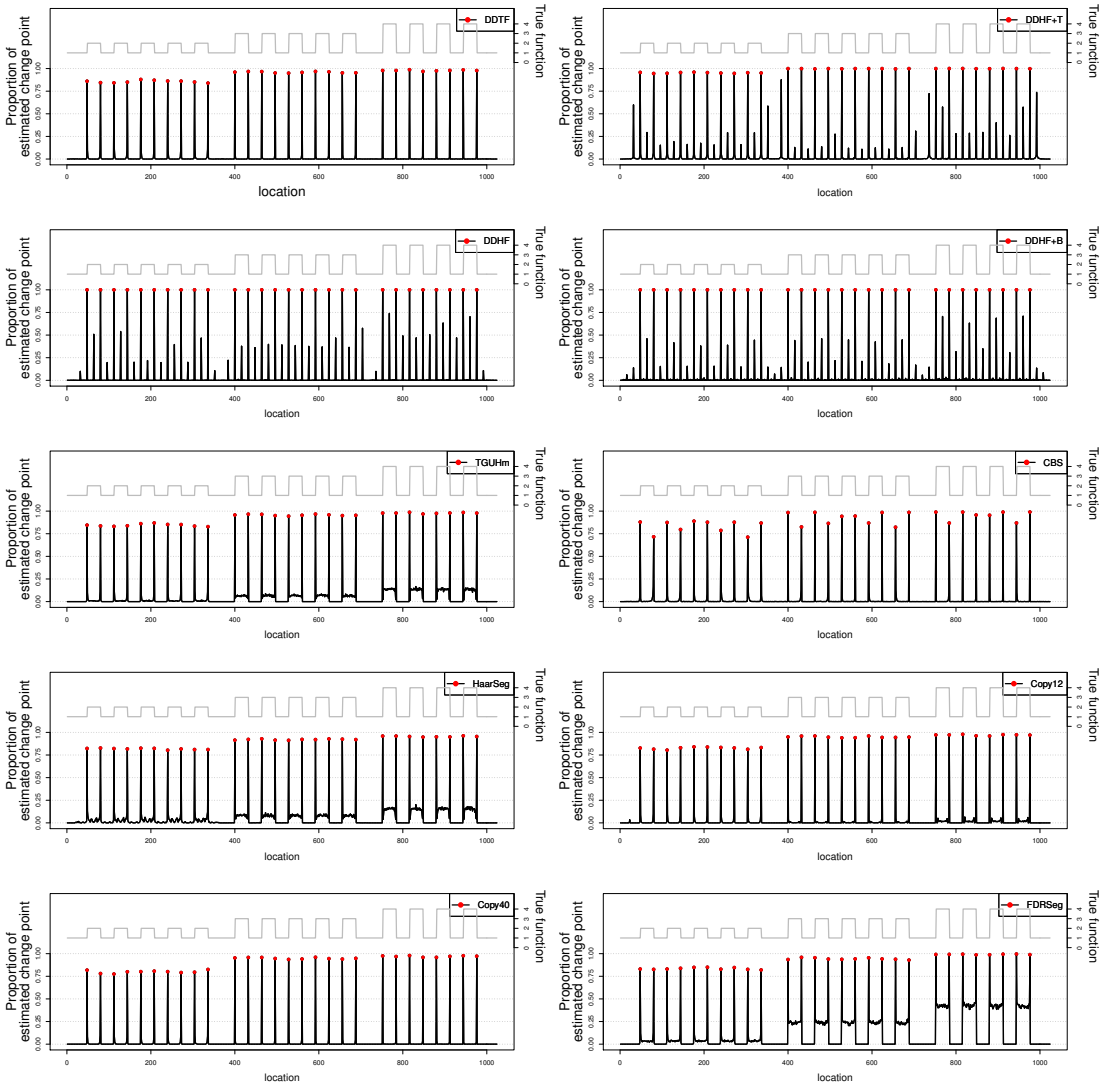


Figure 5.22: Proportion of times a change-point is estimated against location corresponds to the fourth test function (fourth row of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$, where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.2$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

## 5.7 Application to Copy Number DNA Data

In order to see the types of segmentation from each method, Figure 5.23 shows the results of segmentation in chromosome 3 of TMA-127 patient data (Belvedere *et al.*, 2012). The whole genome segmentation using the DDTF method is presented in the Supplementary material. Figure 5.23 indicates that only the DDTF and CBS methods are able to present clear segmentation for the higher mean level segment with high variance which is located around position 1200. In more detail, the DDTF method also estimates short altered segments at low mean level segments. Since the truth in real data is unknown, it is difficult to confirm whether these short altered segments are real changes or not, but based on the simulation results, we speculate that they are true changes.

## 5.8 Conclusion

In this chapter, a segmentation method, DDTF method, was proposed for detecting change-points with applications in copy number segmentation where the data variance depends on the mean. The method was developed based on the DDHF methodology which is known to be effective in variance stabilisation. A novel application was presented which includes the TGUH denoising method to improve its performance in estimating change-point location in the DDHF method.

The simulation study suggested that the proposed method yields excellent results in terms of estimating change-point locations, especially in estimating short segments. This advantage also found in some of the DDHF-based methods but it is followed by a high false positive rate due to the Haar wavelet transformation used in variance stabilisation and reconstruction stages. Unlike those methods, the DDTF method replaces the use of the balance Haar wavelet transform with unbalanced Haar wavelet transform. This enables us to match the likely structure of the data by adjusting the breakpoint of the unbalanced Haar wavelets which results in more accurate estimates of change-points. The spurious change-points at dyadic locations that often occurs in the DDHF-based methods are well addressed by DDTF method. This is important for the identification of copy number alterations as the alterations may occur in any location in the genome. Therefore

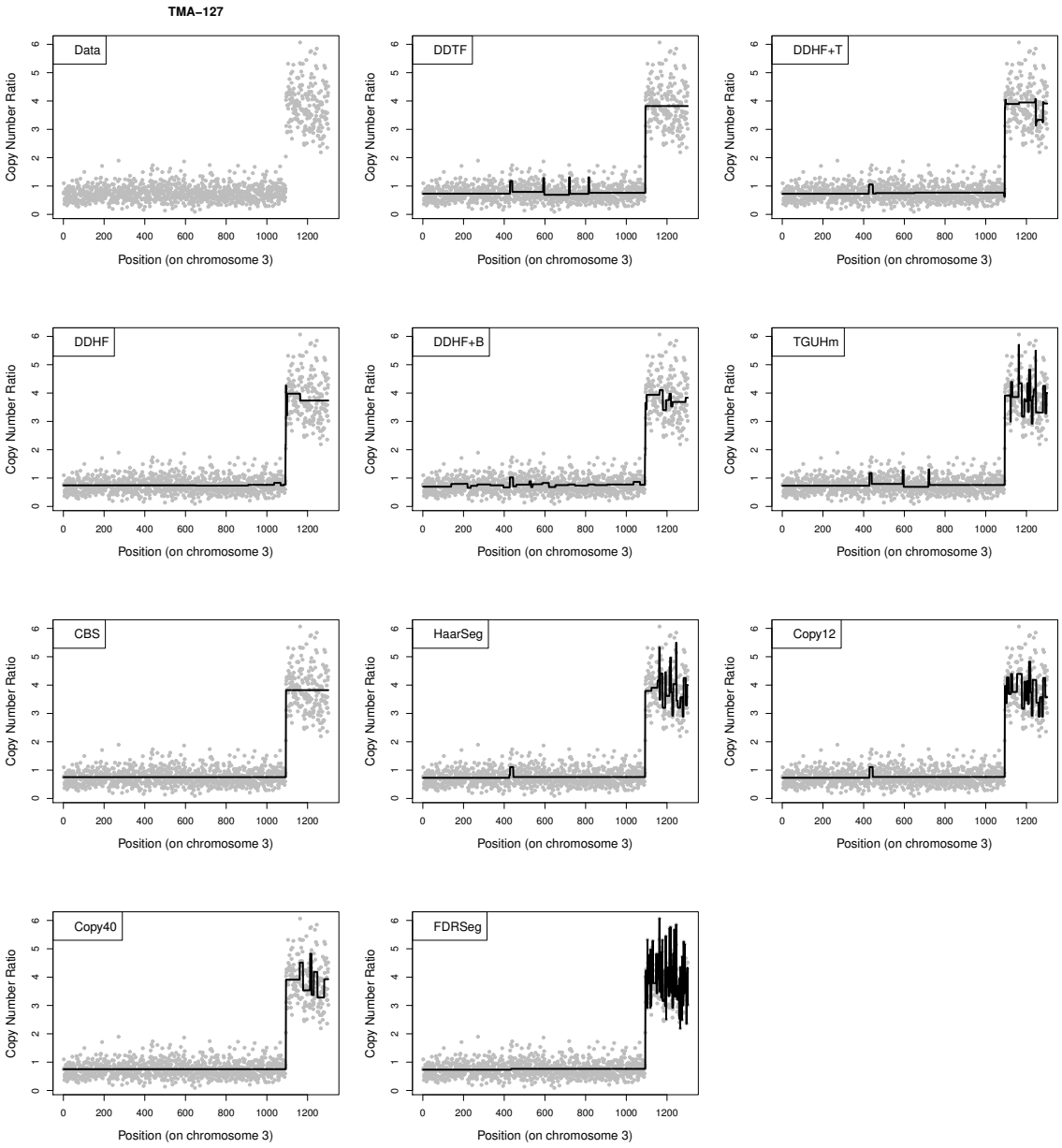


Figure 5.23: CNA estimate as a result of segmentation of chromosome 3 in patient TMA-127 using ten different segmentation methods.

based on the simulation done, in the real data application, when the user thinks that the data might contains many short altered segments, the DDTF method would be the good choice to be used.

It is also interesting to note that DDTF still performs well even when the data do not give enough information to estimate mean-variance relationship, i.e. in the case when the underlying pattern only contains short altered segments. It can preserve its performance to at least not worse than the TGUHm method but still better than the basic DDHF method. This advantage has made the proposed method a flexible alternative for change-points estimation even though there is insufficient mean-variance information in the data.

# Chapter 6

# Wavelet-based Cancer Subtypes Classification

## 6.1 Introduction

In previous chapters, two unbalanced wavelet-based segmentation methods have been introduced. Those methods can be used to separate noise from the CNA data. The resulting CNA estimates can then be processed into a classification procedure. In this chapter, the aim is to explore the advantage of wavelet analysis in classifying cancer subtypes, particularly lung cancer.

Lung cancer is one of the major causes of cancer mortality in the world (Siegel *et al.*, 2012). The most common lung cancer that contributes to this is non–small cell lung cancer (NSCLC) which can be further divided into lung adenocarcinoma (LA) and lung squamous cell carcinoma (LS). These two subtypes are often classified together as NSCLC even though they have different biological signatures (Herbst *et al.*, 2008). Hence, it is essential to investigate statistical models to distinguish these two subgroups clinically.

Changes in DNA copy number or copy number alteration (CNA) is a hallmark of cancer cells (Hanahan & Weinberg, 2011). Each LS and LA tumour subtype has unique patterns of copy number alteration (CNA) due to the differences in their development process (Jamal-Hanjani *et al.*, 2017). Previous studies have shown that CNAs bring important key information for the prediction of the NSCLC subtypes (Gusnanto *et al.*, 2015; Li *et al.*, 2014).

The use of the CNA segmented line itself is already known to be useful for investigating the important gains and losses along the genome that contribute to cancer subtype classification. But in many cases of data classification, representing data as wavelet-transformed variables may improve classification performance. Functions of wavelet coefficients, which enable to emphasise the localised information of the original data, can be used as explanatory variables in a predictive regression method. The wavelet transform allows rapid identification of sudden segment changes in the original data and represents them as a set of wavelet coefficients. It also allows decomposing data into different scales that bring the opportunity for improved interpretation by identifying which resolution scales are the most informative.

This chapter presents a general framework for the application of wavelet transformation to the classification of lung cancer CNA data. This framework is started by first performing segmentation to separate noise from the CNA data and splitting it into regions of equal copy number. For this step, the segmentation methods that have been developed in Chapter 5 can be utilised. Then wavelets are used to transform the segmented CNA data into a set of wavelet coefficients that bring its localised information. The key information extracted by the wavelet transform here, is the localised mean and difference. Finally, a classification method uses the resulting wavelet coefficients to classify the CNA data into one of the non–small cell lung cancer subtypes; lung adenocarcinoma (LA) or lung squamous cell carcinoma (LS). For the classification method, a logistic regression model with a sparse solution is condidered. The term 'sparse' refers to the case where the coefficients of some variables are forced to be exactly zero, while the others are estimated to be away from zero. Due to this sparseness, only coefficients related to the most significant variables are kept, which enables us to identify the key variables that are informative to distinguish lung cancer subtypes.

The details of the proposed framework are described in Section 6.2. Then it is evaluated using simulated data as shown in Section 6.3 and applied to the real CNA data as shown in Section 6.4.

## 6.2 Methodology

In this section, the detailed classification procedure for CNA data using wavelet transform is described. The procedure can be outlined into four main stages as follows.

1. **Stage 0. Data preparation.** For easier comparison between CNA profiles, several data preparation steps need to be performed such as determining the optimal window size of the raw CNA data and data normalisation. This step is particularly suggested for CNA data obtained from NGS technology and can be ignored for other copy number data.

2. **Stage 1. Segmentation.** The normalised CNA profiles are segmented to translate noisy data into regions of equal copy number.

3. **Stage 2. Non-decimated Haar wavelet transform (NDWT).** Take the NDWT of segmented CNA profiles which results in a set of NDWT detail and scaling coefficients. The detailed explanation about NDWT transform is explained in Section 3.2.

4. **Stage 3. Classification using Logistic regression.** Perform prediction algorithm (logistic regression) using the NDWT coefficients.

In this chapter, logistic regression is used as it has been proven to be a valid choice for classification using copy number alteration (CNA) data (Ghosh & Chinnaiyan, 2005; Kaymaz *et al.*, 2021). Furthermore, because the wavelet transform translates signals into sparse representation, and this chapter aims to pinpoint key genomic markers for lung cancer subtype classification (squamous carcinoma and adenocarcinoma), using logistic regression with Lasso is advantageous. This method provides a sparse solution and automatically selects relevant features while shrinking the coefficients of less important features to zero.

For future study, some classification methods such as Naive Bayes, Decision Trees, and Random Forests may offer distinct advantages. However, those methods were not chosen to be used in this chapter because based on practical knowledge, some of those methods have certain disadvantages that we aim to avoid for CNA analysis: the Naive Bayes assumes feature independence, which may not

align well with correlated genomic markers in CNAs, and the decision Trees and Random Forests capture nonlinearities and interactions but might struggle with high-dimensional data.

### 6.2.1 Data Preparation

In our case, copy number alterations data obtained from 76 lung cancer patients ($n = 76$) from the Leeds Teaching Hospitals NHS Trust (UK) as explained in Section 2.4 are used. For this dataset, with the 150 kb window size, the reads are binned approximately into 20,000 genomic windows. Since missing values can be problematic for the analysis, the sex chromosomes and the centromere regions are removed. At the end of this removing procedure, the number of genomic windows $q$ becomes 17,931 ($q = 17,931$).

### 6.2.2 Segmentation

To separate the informative copy number pattern from noise, the data-driven Haar-Fisz (DDTF) segmentation, which has been explained in detail in Chapter 5, is performed on the normalised CNA samples. The minimum wing length ($m^*$) is set to two to minimise the effect of a single extreme data point. The CNA estimates from the patients are then summarised in a matrix of size $n = 76$ by $q = 17,931$. The column is ordered based on genomic locations.

### 6.2.3 Non-decimated Haar Wavelet Transform of CNA Profiles

For data such as CNA, where the true pattern underlying the 'noisy' data is a piecewise constant function, the standard transformation such as the Fourier transform is unsuitable since it captures global frequency information, meaning frequencies that persist over an entire signal. In contrast, wavelets are able to decompose a signal into a set of wavelets (or scales). In the context of CNA analysis, if we use the Fourier transform, localised information in the signal like a discontinuity will affect the entire coefficients of the series. But if we use the wavelet transform, it only affects the coefficients produced by the wavelets that

overlap with this discontinuity, which will enable us to identify the 'informative' coefficients.

In this study, the non-decimated discrete wavelet transform (NDWT) is considered to represent CNA estimates in the location and frequency domains. Unlike the 'basic' discrete Haar wavelet transform (DWT) that uses dyadic wavelets, the NDWT has complete location localisation at each scale, resulting in an over-complete basis. Therefore, it suits the characteristics of copy number alteration data as the change points in CNA data do not always occur at dyadic locations (can be found anywhere along the sequence).

The NDWT will result in two kinds of coefficients: (i) detail coefficient and (ii) scaling coefficients. The detail coefficients carry the 'different' information of two consecutive pairs while the scaling coefficients carry the 'average' information. For the detail coefficients, at a fine scale (high frequency) resolution level, wavelets are highly localised which means that the coefficients representing the information of the number and location of change-points in the corresponding signal more precisely, while those at a coarser scale measure lower frequency activity meaning that the coefficient carries the 'different' information of two consecutive pairs for a larger region. Meanwhile, the scaling coefficients contain coarsening of that in the CNA estimates. As explained in Section 3.2, the procedure of obtaining scaling coefficients is similar to a moving average smoothing operation, in which, coarser the scale smoother the signal. However, the NDWT only represents data of length $q$ at $\lfloor \log_2(q) \rfloor$ resolution levels or scales. For both detail and scaling coefficients, each scale (resolution level) represents activity at approximately twice the frequency of the previous scale.

For a quick reminder, a brief explanation of the NDWT transform is presented here. A more details explanation of the NDWT transform is explained in Section3.2. In the NDWT, the wavelet function $\psi$ (mother wavelet) and Haar scaling function $\phi$ (father wavelet) are used, where

$$\psi(\tau) = \begin{cases} 1 & \text{for } \tau \in [0, 1/2), \\ -1 & \text{for } \tau \in [1/2, 1), \\ 0 & \text{otherwise.} \end{cases} \quad \text{and} \quad \phi(\tau) = \begin{cases} 1 & \text{for } \tau \in [0, 1), \\ 0 & \text{otherwise.} \end{cases} \tag{6.1}$$

By using dilation and translation, the wavelet detail and scaling functions at location $k$ and scale $j$ can be obtained by:

$$\psi_{j,k}(\tau) = 2^{j/2}\psi(2^j(\tau - k)), \quad \text{and} \quad \phi_{j,k}(\tau) = 2^{j/2}\phi(2^j(\tau - k)), \tag{6.2}$$

where $j = 1, 2, \ldots, \lfloor \log_2(q) \rfloor$ and $k = 0, \ldots, q - 1$.

The NDWT maps a discrete CNA estimate vector $r$ to a collection of NDWT detail $(d_{j,k})$ and scaling $(c_{j,k})$ coefficients at levels $j = 1, 2, \ldots, \lfloor \log_2(q) \rfloor$ and locations $k = 0, \ldots, q - 1$ defined by

$$d_{j,k} = \langle r, \psi_{j,k} \rangle \quad \text{and} \quad c_{j,k} = \langle r, \phi_{j,k} \rangle. \tag{6.3}$$

Figure 6.1 shows the wavelet transform of a simple piecewise constant signal and Figure 6.2 is the wavelet transform of a LS patient. The top row shows the piecewise constant signal and the middle and bottom rows show the plot of its NDWT detail and scaling coefficients, respectively. The coefficients $d_{j,k}$ and $c_{j,k}$ are plotted with the finest-scale coefficients at the bottom of the plot, and the coarsest at the top. The left-hand axis indicates the scale. The magnitude of the coefficient is denoted by a vertical mark located along an imaginary horizontal line centred at each level. The horizontal positions of the coefficients indicate the approximate position in the original data from which the coefficient is derived.

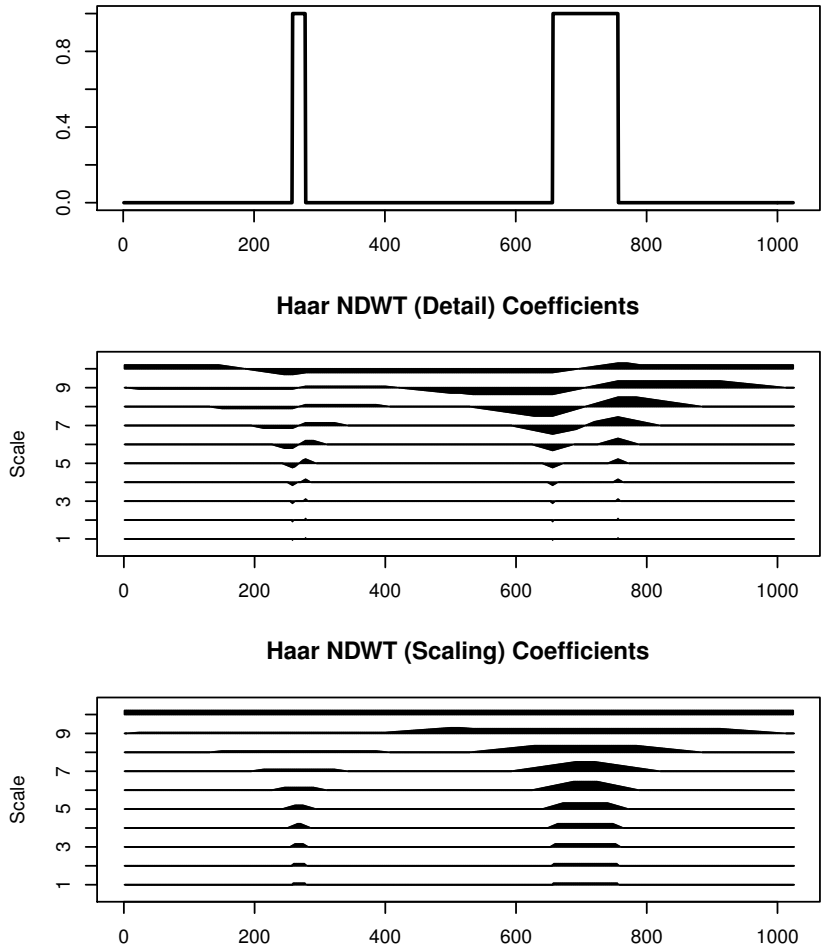


Figure 6.1: Discrete wavelet detail (middle row) and scaling (bottom row) coefficients of a piecewise constant signal. The left-hand axis indicates the scale. The magnitude of the coefficient is denoted by a vertical mark located along an imaginary horizontal line centred at each level. The horizontal positions of the coefficients indicate the approximate position in the original data from which the coefficient is derived.

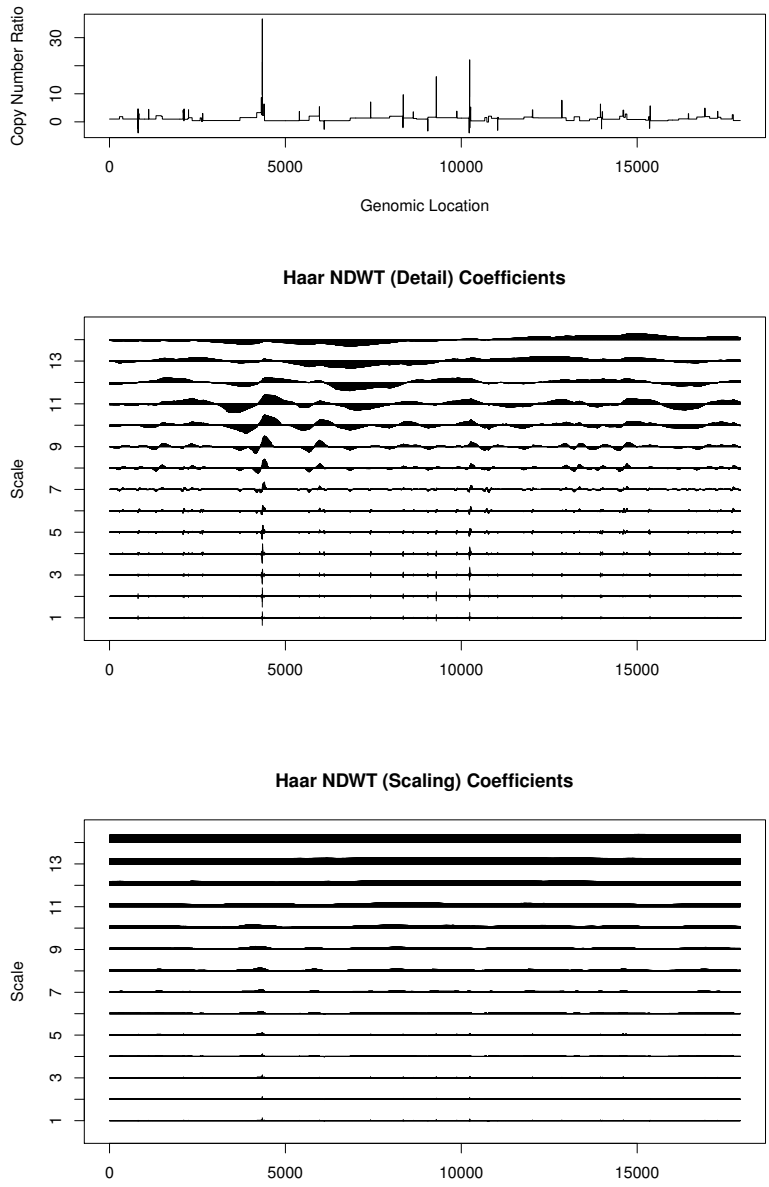


Figure 6.2: First row: copy number ratio of the LS80 cancer patient. Second row: Haar NDWT detail coefficient. Third row: Haar NDWT detail coefficient. For the second and third rows, the left-hand axis indicates the scale. The magnitude of the coefficient is denoted by a vertical mark located along an imaginary horizontal line centred at each level. The horizontal positions of the coefficients indicate the approximate position in the original data from which the coefficient is derived.

### 6.2.4 Classification using Logistic Regression

Let $q$ be the number of samples and $y$ as a vector of binary cancer subtype with $y_i = 1$ if the cancer is LS and $y_i = 0$ if the cancer is LA for $i = 1, 2, \ldots, n$. Let $x_i \equiv (x_{i1}, ..., x_{im})$ be a fixed covariate vector, including the intercept, and $z_i \equiv (z^j_{i1}, ..., z^j_{in})$, where $z^j_{ik}$ the NDWT detail/scaling coefficients of the $k$-th location at a scale $j$ and $q$ is the length of CNA data (since the length of NDWT detail/scaling coefficients of the CNA data $\mathbf{r}$ at each scale is equal to the length of CNA data itself).

The $p_i$, can be modelled in vector notation as

$$\log \frac{p_i}{1 - p_i} = x_i' b + z_i' \beta \tag{6.4}$$

where $b$ is fixed regression parameter and $\beta$ is parameter vectors for the NDWT coefficients. In matrix notation 6.4 can be written as

$$\log \frac{p}{1 - p} = \mathbf{X}b + \mathbf{Z}\beta, \tag{6.5}$$

where $\mathbf{X}$ and $\mathbf{Z}$ are the $n \times m$ and $n \times q$ matrices by storing row vectors $x$'s and $z$'s and the function on the left-hand side is understood to apply element-wise. In general, $\mathbf{X}$ can contain any clinical predictors but in the specific application of this chapter, $\mathbf{X}$ is set as a vector of ones or a fixed intercept. The term $\mathbf{X}b$ in the equation (6.5) then can be replaced by $b_0$.

Here the case is the number of subjects $n$ is much less than the number of explanatory variables $q$, $n << q$. One problem that commonly occurs due to this condition is multicollinearity or several (groups) of $z^j_{ik}$'s show identical patterns. In the context of CNA data, the multicollinearity is very understandable: there will be genes that have a nearly identical pattern. This characteristic is clearly shown by the presence of correlation blocks in the correlation between genomic regions in the CNA dataset as shown in Kaymaz *et al.* (2021). This characteristic also isolates the wavelet-transformed variables.

Many solutions have been proposed to solve the multicollinearity problem such as variable selection, principal component analysis, partial least squares and penalised estimation. This chapter only considers the latter. The most commonly used penalised regression include (i) ridge regression, (ii) Lasso regression, and

(iii) elastic net regression. In this chapter, the Lasso regression is used to make the coefficients of some less contributed variables are forced to be exactly zero and only the most significant variables are kept in the final model. The logistic Lasso estimator is defined as

$$\hat{\beta}_\lambda = \arg\min_\beta(||Y - Z\beta||_2^2 + \lambda \sum_{l=1}^{n} |\beta_l|). \tag{6.6}$$

Here, $\lambda > 0$ is a tuning parameter that controls the sparsity of the estimator (i.e., the number of coefficients with a value of zero). In practice, it is selected by cross-validation. For obtaining the logistic Lasso estimator, the glmnet package in R (Simon *et al.*, 2011) was used.

In practice, two kinds of explanatory variables are considered in this study as follows:

1. **NDWT detail coefficients of each scale**. At each of the models, **Z** is defined as the NDWT detail coefficients of one specific scale, $j$. The extension part of the signal is truncated so that the size of the matrix **Z** is $n = 76 \times q = 17931$. As an illustration, a logistic regression model 1 uses the scale 1 NDWT detail coefficients only as the matrix **Z**, model 2 uses the scale 2 NDWT detail coefficients, and so on.

2. **NDWT scaling coefficients of each scale**. Similar to the previous model, here the scaling coefficients of one specific scale $j$ are considered as the explanatory variable. The coefficients of each scale are considered and compared to obtain the best result where each of the models uses the NDWT scaling coefficients of a scale on its own as the matrix **Z**.

Then by comparing the misclassification rate results, the most informative scale can be identified.

### S-fold Cross-validation

Cross-validation is considered for estimating $\lambda$ and as a method to assess the method performance in terms of classification error. The cross-validation can be

performed by, first, splitting $q$ observations in the data into a training set of size $n_t$ and a validation set of size $n_v$ where $n_t + n_v = n$ such that

$$y := \begin{bmatrix} y_t \\ \dots \\ y_v \end{bmatrix}, X := \begin{bmatrix} X_t \\ \dots \\ X_v \end{bmatrix}, Z := \begin{bmatrix} Z_t \\ \dots \\ Z_v \end{bmatrix}. \tag{6.7}$$

In cross-validation, "S-fold" refers to the number of subsets or "folds" into which the original dataset is divided during the evaluation process. The term "S" represents the number of folds, and it determines how many iterations of training and testing will be performed during the cross-validation procedure. In n-fold cross-validation, the dataset is partitioned into S subsets of approximately equal size.

In practice, four-fold cross-validation is performed and the classification error is calculated using validation sets across the four folds, which means that if there are 100 elements in a dataset, it is partitioned into four subsets and each subset has 25 elements. One set is taken among those four sets for validation and the remaining three are for training. Then the process is repeated for all four sets.

The training set is used for estimating the model parameters $\hat{b}_t$ and $\hat{\beta}_t$. The resulting estimates are then used in the validation set to obtain model prediction

$$\hat{y}_v = \mathbb{I}\left(\frac{1}{1 + e^{-(X_v\hat{b}_v + Z_v\hat{\beta}_v)}} \geq 0.5\right) \tag{6.8}$$

where $\mathbb{I}$ is an indicator function such that $\hat{y}_v$ equals one if the expression inside the brackets is true, and zero otherwise.

From this prediction, the classification error in the validation set is calculated by comparing the prediction $\hat{y}_v$ with the observed group labels $y_v$. The classification error is defined as

$$CE = \frac{1}{n_v}\sum_{k=1}^{n_v} \mathbb{I}(y_{vk} \neq \hat{y}_{vk}), \tag{6.9}$$

where $y_v = (y_{v1}, y_{v2}, \dots, y_{vn_v})^T$ and $\hat{y}_v = (\hat{y}_{v1}, \hat{y}_{v2}, \dots, \hat{y}_{vn_v})^T$.

At the end of the proposed approach, a set of CE for each scale $j$ will be obtained, and then, let $CE^j$ be the median of CE at each scale. By finding

the smallest $CE^j$ across each scale, the scale that contributes the most to the classification can be identified. It can be written mathematically as

$$j_b = \arg\min_j CE^j, \tag{6.10}$$

where $j_b$ is the best scale.

## 6.3 Simulation Study

To understand the characteristics of the proposed methodology, a simple simulation study with a smaller signal length was considered under the following setting. To obtain realistic CNA estimates, the simulated data is produced from the 'known truth' of the CNA pattern of LA and LS tumours then contaminated with standard Gaussian noise and lastly, segmented by the DDTF method. In this simulation study, one hundred simulated datasets, $D^v$ $(v = 1, \ldots, 100)$, are generated and each of these datasets consists of 50 samples of group one (LA), $a^{v,i}$, and 50 samples of group two (LS), $s^{v,i}$, where $i = 1, \ldots, 50$.

Let the 'true' functions of the CNA pattern of LA and LS be the same within the group and denoted as $f^a$ for the LA group $f^s$ for the LS group. Several types of the 'true' functions or test functions are considered and explained further in the next subsections. Both $f^s$ and $f^a$ are piecewise constant functions with length $q = 1024$. The noisy cancer CNA data of each patient are generated independently. Mathematically, each samples of group one (LA) are generated from model $a_l^{v,i} = f_l^a + \epsilon_l$ and group two (LS) from $s_l^{v,i} = f_l^s + \epsilon_l$, where $l = 1, \ldots, q$ and $\epsilon_l$ is additive i.i.d Gaussian noise $N(0, \sigma^2)$.

After 100 noisy CNA datasets are obtained, the DDTF method was used to segment all of these simulated samples to produce piecewise constant estimates $\hat{D}^v$. Any segmentation method can be used here, but here the DDTF method was chosen based on the results in Section 5.6.1 which indicates its superiority compared to some other well-known segmentation methods. Then, the NDWT was applied to each of $\hat{a}_l^{v,i}$ and $\hat{s}_l^{v,i}$ in $\hat{D}^v$. The NDWT maps each of discrete vector $\hat{a}^{v,i}$ and $\hat{s}^{v,i}$ to a collection of NDWT detail $(d_{j,k})$ and scaling $(c_{j,k})$ coefficients at levels $j = 1, 2, \ldots, 10$ and locations $k = 0, \ldots, 1023$. As described in

Section 6.2.4, for each test function considered, the performance of: (i) NDWT detail coefficients, and (ii) NDWT scaling coefficients of each scale, with those that only use CNA estimates/segmented dataset $\hat{D}$ (untransformed dataset) as predictors are compared to the model that uses both detail and scaling coefficients from all scales and the CNA segmented dataset itself. This simulation procedure is summarised in a flowchart shown in Figure 6.3

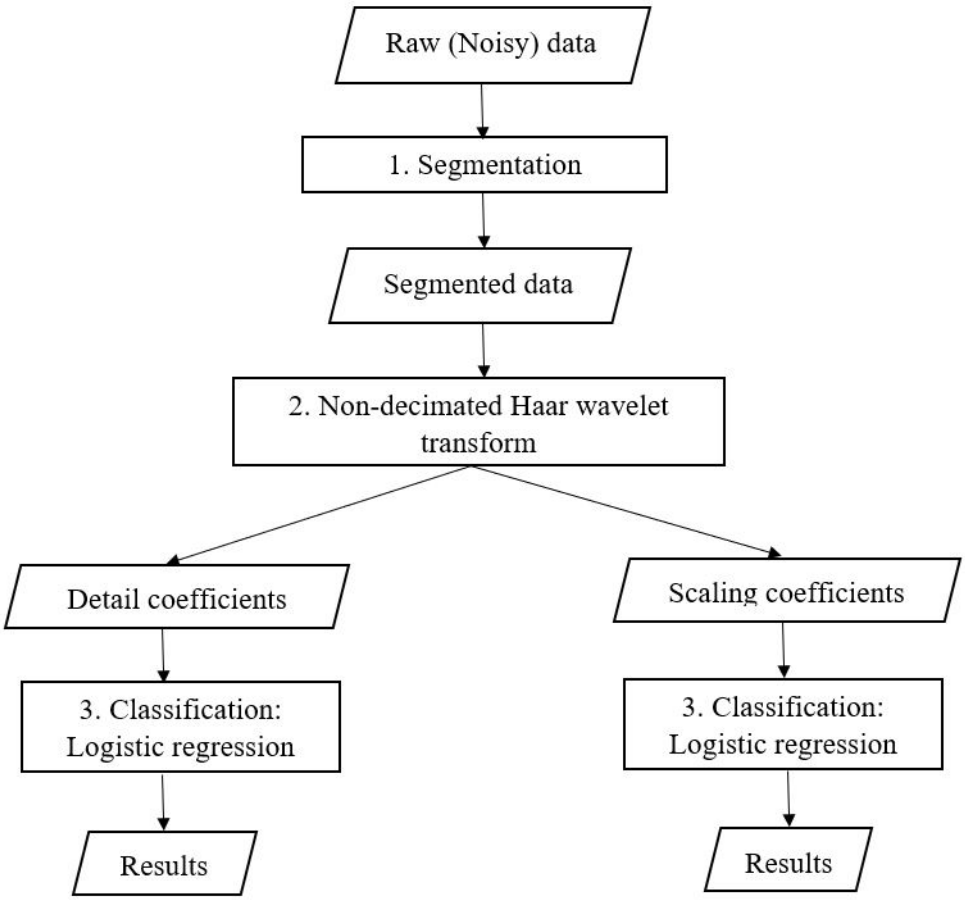


Figure 6.3: Flowchart for simulation of wavelet-based copy number data classification

Furthermore, to estimate the optimal $\lambda$ in equation (6.6) of each model, additional 10 datasets are generated in the same way as explained above and by

performing 4-fold Cross-validation, the optimal $\lambda$ is estimated as the one that minimises the (4-folds) misclassification rate.

### Test Function with One Altered Region

In this section, two kinds of simulated LA and LS CNA segmented patterns (a piecewise constant function that is used to generate noisy data) are considered as shown in the top panel of Figures 6.4 and 6.5. For both of the simulated patterns, the difference between LA and LS groups only locate at a single altered segment. The test function of the LS group has an altered segment while the LA group does not have any altered region. The length of the altered segment of the first simulated LS data is 50 (top panel of Figure 6.5) and the second one is 20 (top panel of Figure 6.5).

The bottom panel of Figures 6.4 and 6.5 shows the misclassification rate from 4-fold cross-validation of 100 datasets across different models using the Lasso penalty. The first simulated data, scale-5 NDWT scaling coefficients as predictors (Lasso penalty) gives the lowest misclassification rate. Besides scale-5, the misclassification rate of scale-6 is also quite close to scale-5 and significantly better than the untransformed data (seg) results. For the second simulated data (see Figure 6.5), the lowest misclassification rate was produced by scale-4 NDWT scaling coefficients (Lasso penalty). An interesting point here is that for both cases, the lowest misclassification rate is obtained by the model which uses coefficients that are produced by wavelets with length close to the length of the altered segment in the LS group. The altered segment's length in the first simulation is 50 and the lowest misclassification rate was presented by scale-5 and scale-6 which is associated with wavelets with length $2^5 = 32$ and $2^6 = 64$, respectively.

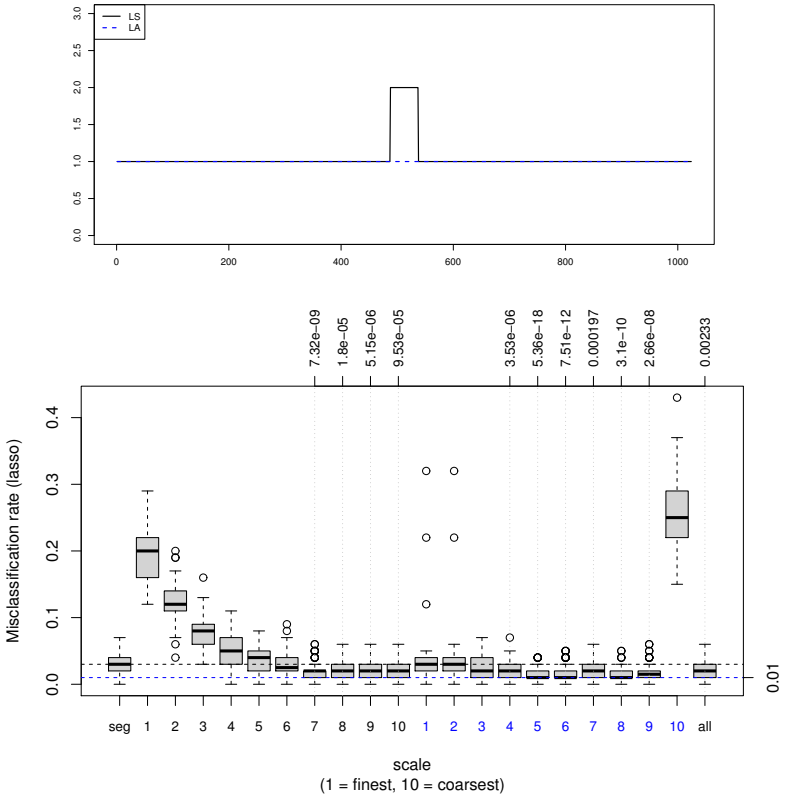


Figure 6.4: Misclassification rate of the first simulated dataset from 4-fold cross-validations of 100 datasets (for each dataset 75 samples in the training set and 25 samples in the validation set) where the predictors are NDWT detail (denoted by black x-axis label) and scaling (denoted by blue x-axis label) coefficients across different scales. The 'seg' and 'all' labels indicate the result for untransformed segmented CNA data and NDWT coefficients from all the scales as predictors, respectively. The upper x-axis label shows the p-values of the models that are significantly lower than the 'seg' model. Top panel: Plot of test functions, the black solid line denotes the test function for LS group and the blue dashed line denotes LA group. The length of the altered segment is 50. Bottom panel: Misclassification rate using Lasso regularisation.

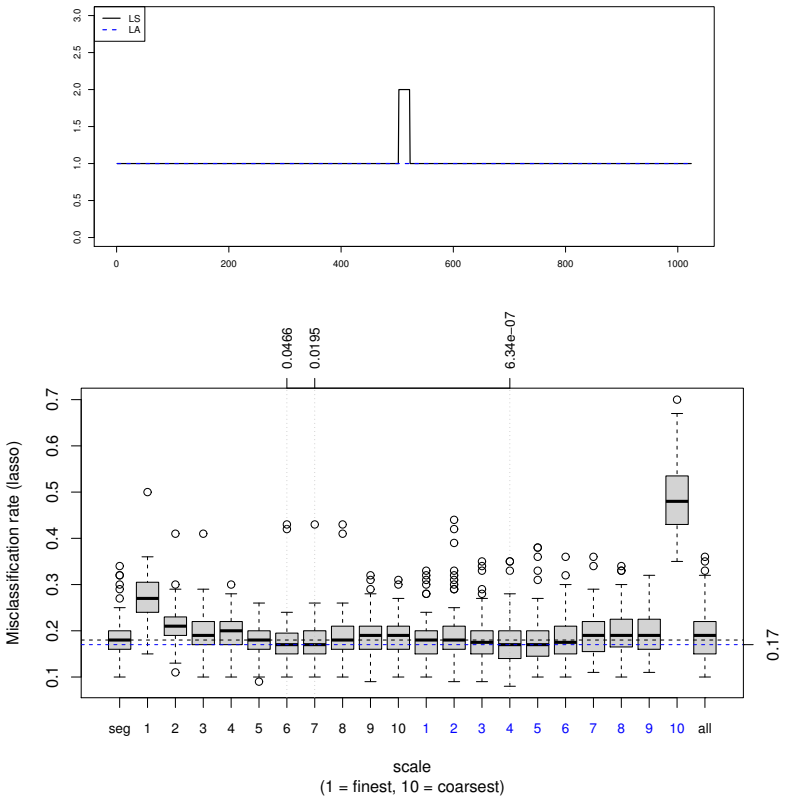


Figure 6.5: Misclassification rate of the second simulated dataset from 4-fold cross-validations of 100 datasets (for each dataset 75 samples in the training set and 25 samples in the validation set) where the predictors are NDWT detail (denoted by black x-axis label) and scaling (denoted by blue x-axis label) coefficients across different scales. The 'seg' and 'all' label indicate the result for untransformed segmented CNA data and NDWT coefficients from all the scales as predictors, respectively. The upper x-axis label shows the p-values of the models that are significantly lower than the 'seg' model. Top panel: Plot of test functions, the black solid line denotes the test function for LS group and the blue dashed line denotes LA group. The length of the altered segment is 20. Bottom panel: Misclassification rate using Lasso regularisation.

## Test Function with Two Altered Regions

Two kinds of simulated LA and LS CNA segmented patterns are considered as shown in the top panel of Figures 6.6 and 6.9. For both of the simulated patterns, the difference between LA and LS is located at two regions of the altered segment. The test function of the LS group has two altered segments while the LA group does not have any altered region. The length of the altered segments is 20 and 50.

In the first simulated data (Figure 6.6), the height of both altered segments is the same and set to one. Meanwhile, for the second simulated data (Figure 6.9), the height of the narrower altered segment is two and the wider one is one.

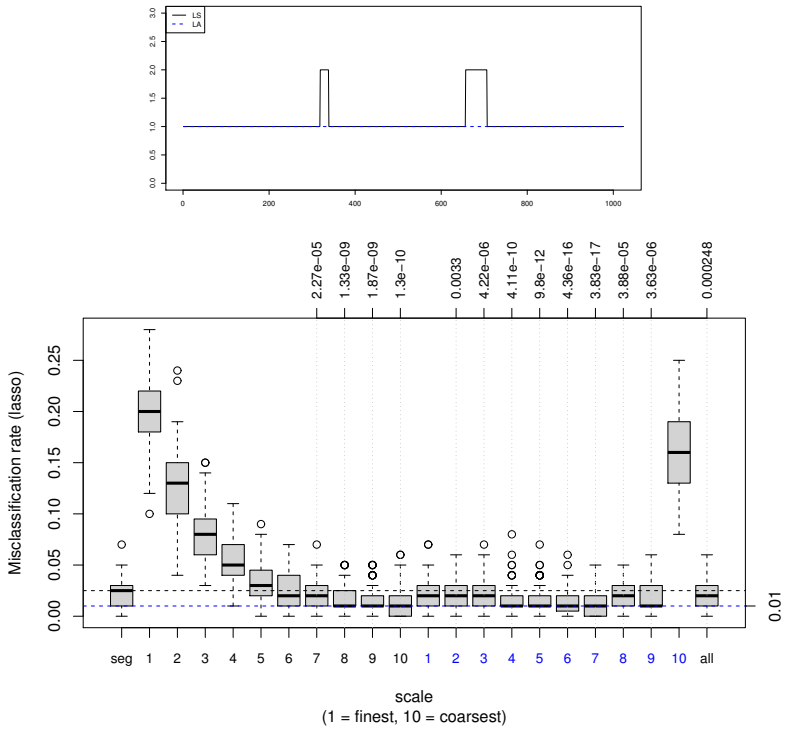


Figure 6.6: Misclassification rate of the third simulated dataset from 4-fold cross-validations of 100 datasets (for each dataset 75 samples in the training set and 25 samples in the validation set) where the predictors are NDWT detail (denoted by black x-axis label) and scaling (denoted by blue x-axis label) coefficients across different scales. The 'seg' and 'all' labels indicate the result for untransformed segmented CNA data and NDWT coefficients from all the scales as predictors, respectively. The upper x-axis label shows the p-values of the models that are significantly lower than the 'seg' model. Top panel: Plot of test functions, the black solid line denotes the test function for the LS group and the blue dashed line denotes the LA group. The length of the altered segments is 20 and 50. Bottom panel: Misclassification rate using Lasso regularisation.

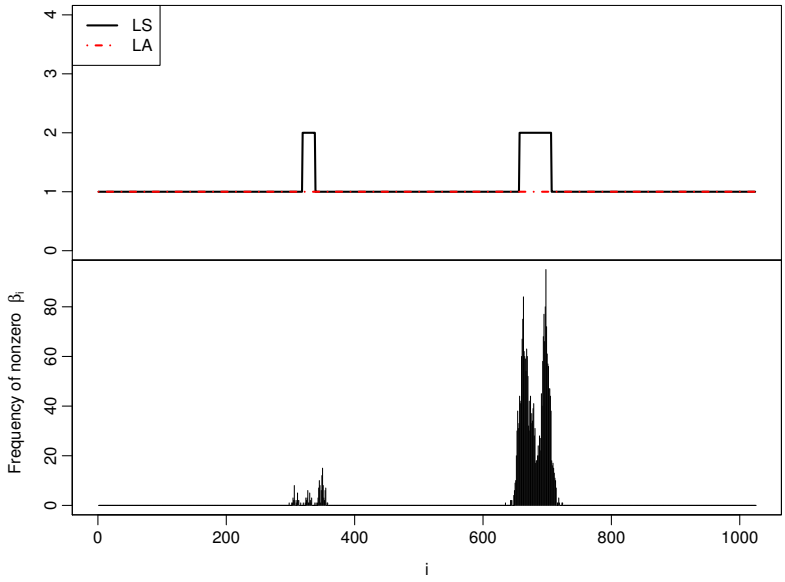


Figure 6.7: Frequency of times nonzero $\beta$ are estimated for model with scale-6 of scaling coefficients over 4-folds cross-validation of 100 dataset of the third simulated dataset.

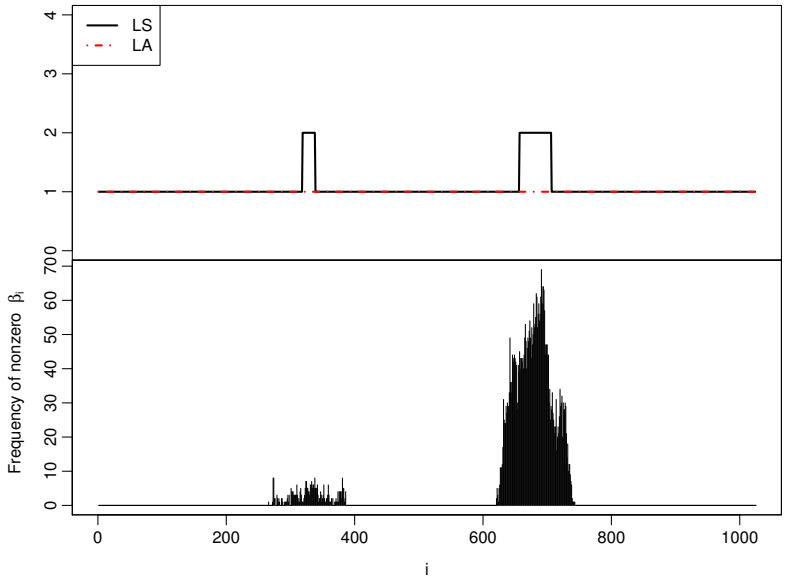


Figure 6.8: Frequency of times nonzero $\beta$ are estimated for model with scale-7 of scaling coefficients over 4-folds cross-validation of 100 dataset of the third simulated dataset.

For the first simulation, Figure 6.6 indicates that the misclassification rate of the model with scaling coefficients of scale-7 followed by scale-6 is the lowest. The frequency of times non-zero $\beta_i$ estimated over 4-fold cross-validation of 100 datasets of these models is shown in Figures 6.7 and 6.8. The results show that variable that corresponds to a region with a wider altered segment are chosen more often than the narrower one. This indicates that in this case, the wider altered segment region is more dominant or contributes more to distinguishing the LS and LA groups.

In the second simulation, the results in Figure 6.9 show that the lowest misclassification rate is achieved by scaling coefficients of scale-2 and then followed by scale-6. For this case, the $\beta_i$ that corresponds to the narrower altered segment is chosen more by Lasso penalty as shown in Figures 6.10 and 6.11.

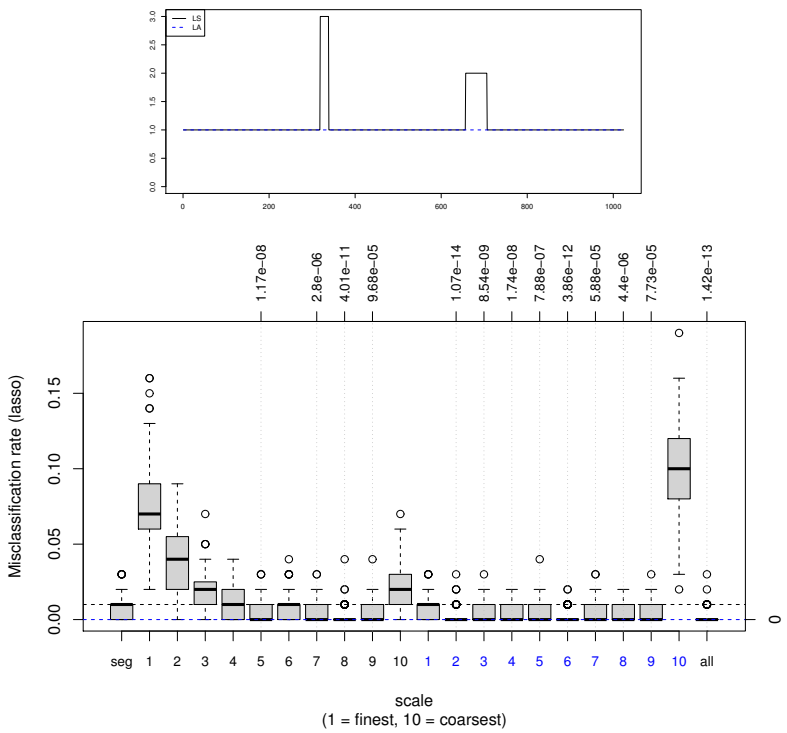


Figure 6.9: Misclassification rate of the fourth simulated dataset from 4-folds cross-validations of 100 datasets (for each dataset 75 samples in the training set and 25 samples in the validation set) where the predictors are NDWT detail (denoted by black x-axis label) and scaling (denoted by blue x-axis label) coefficients across different scales. The 'seg' and 'all' labels indicate the result for untransformed segmented CNA data and NDWT coefficients from all the scales as predictors, respectively. The upper x-axis label shows the p-values of the models that are significantly lower than the 'seg' model. Top panel: Plot of test functions, the black solid line denotes the test function for the LS group and the blue dashed line denotes the LA group. The length of the altered segments is 20 and 50. Bottom panel: Misclassification rate using Lasso regularisation.

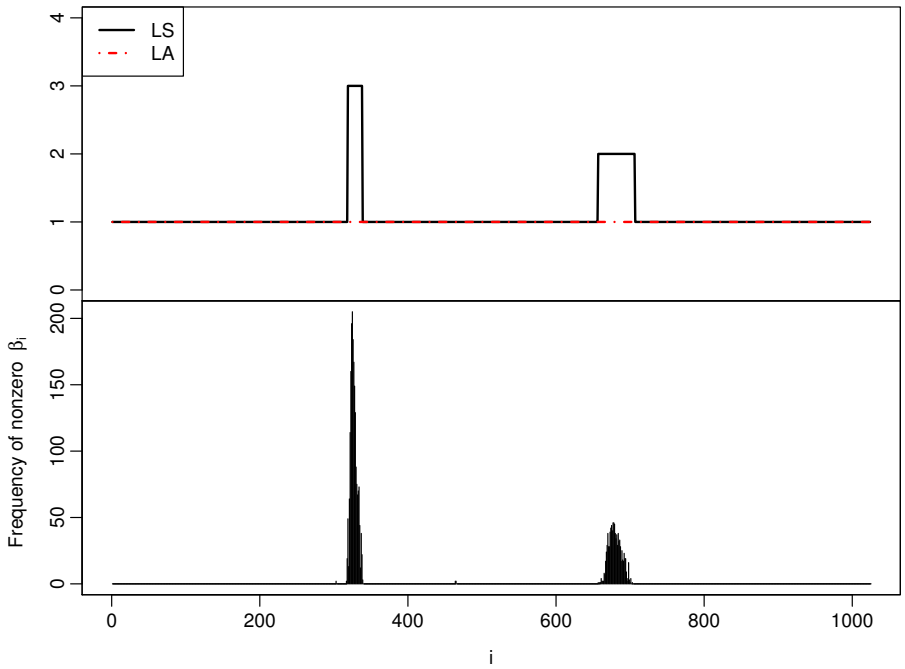


Figure 6.10: Frequency of times nonzero $\beta$ are estimated for model with scale-2 of scaling coefficients over 4-folds cross-validation of 100 dataset of the fourth simulated dataset.

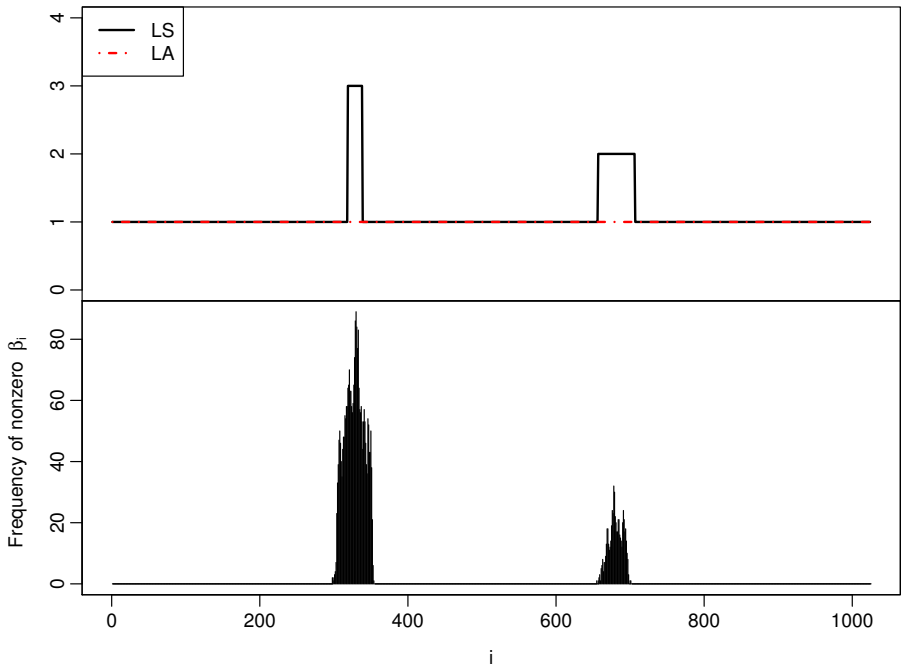


Figure 6.11: Frequency of times nonzero $\beta$ are estimated for model with scale-6 of scaling coefficients over 4-folds cross-validation of 100 dataset of the fourth simulated dataset.

These two cases indicate that the proposed methodology using scaling coefficients allows us to identify whichever feature that more dominant to distinguish the two groups. The dominant feature does not always correspond to the wider region.

## Test Function with Multiple Altered Regions

In this section, a test function that has a more complicated pattern than the previous ones is considered. The test functions used are shown in the top panel of Figure 6.12. The different features between the two groups are located at several segments with various heights.

Based on Figure 6.12, the best model that minimises the misclassification rate is scale-4 scaling coefficients. From the plot of the frequency of times nonzero $\beta$ are estimated for scale-4 as shown in Figure 6.13, the most important feature that is informative to distinguish the two groups are located at position 760–780. This

region has the largest difference between LA and LS groups in terms of alteration height. This result is consistent with what was shown in the previous simulation that the important feature does not always correspond to the widest region but also can be a narrower region but with larger differences in height.

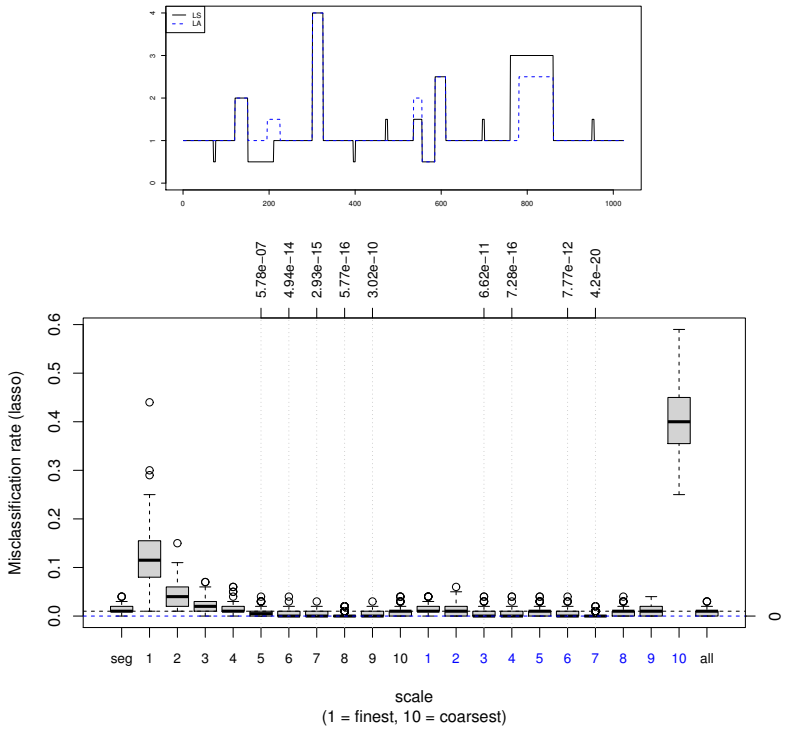


Figure 6.12: Misclassification rate of the fifth simulated dataset from 4-fold cross-validations of 100 datasets (for each dataset 75 samples in the training set and 25 samples in the validation set) where the predictors are NDWT detail (denoted by black x-axis label) and scaling (denoted by blue x-axis label) coefficients across different scales. The 'seg' and 'all' labels indicate the result for untransformed segmented CNA data and NDWT coefficients from all the scales as predictors, respectively. The upper x-axis label shows the p-values of the models that are significantly lower than the 'seg' model. Top panel: Plot of test functions, the black solid line denotes the test function for LS group and the blue dashed line denotes LA group. Bottom panel: Misclassification rate using Lasso regularisation.

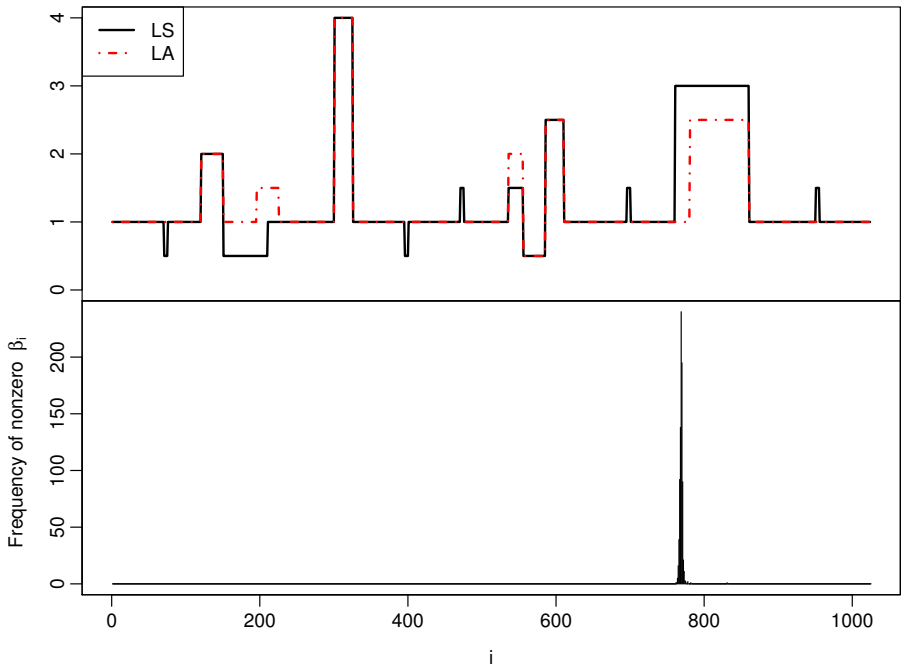


Figure 6.13: Frequency of times nonzero $\beta$ are estimated for model with scale-4 of scaling coefficients over 4-folds cross-validation of 100 dataset of the fifth simulated dataset.

## Simulation with Unique Test Function

In the real application, the underlying true pattern behind each sample of the LS or LA group differs due to biological variation. To get a realistic simulation, in this section, a unique test function is generated for each of the samples. First, for each of the groups, some fixed altered regions to distinguish LS and LA groups are assigned. Then, for the remaining regions, the altered segment is generated from the empirical profile constructed from the CBS segmentation of 76 lung cancer patients. Then, randomly sampled copy number levels from the empirical distribution of segment mean values, where mean values were binned into the intervals less than 0.25 (0 copies), between 0.25, and 0.75 (one copy), between 0.75 and 1.25 (2 copies), between 1.25 and 1.75 (three copies), between 1.75 and 2.25 (four copies), between 2.25 and 2.75 (five copies), between 2.75 and 3.25 (six copies). The length of normal segments (copy number 2) is assigned by randomly

sampling the segment length from the empirical length distribution of copy number levels belonging to the $[0.75, 1.25]$ bin. Similarly, the lengths are assigned to the altered segments by sampling from the length distribution for segments with levels outside that bin, without distinguishing among length distributions with different copy numbers. The illustration of this simulation setting is illustrated in Figure 6.14.

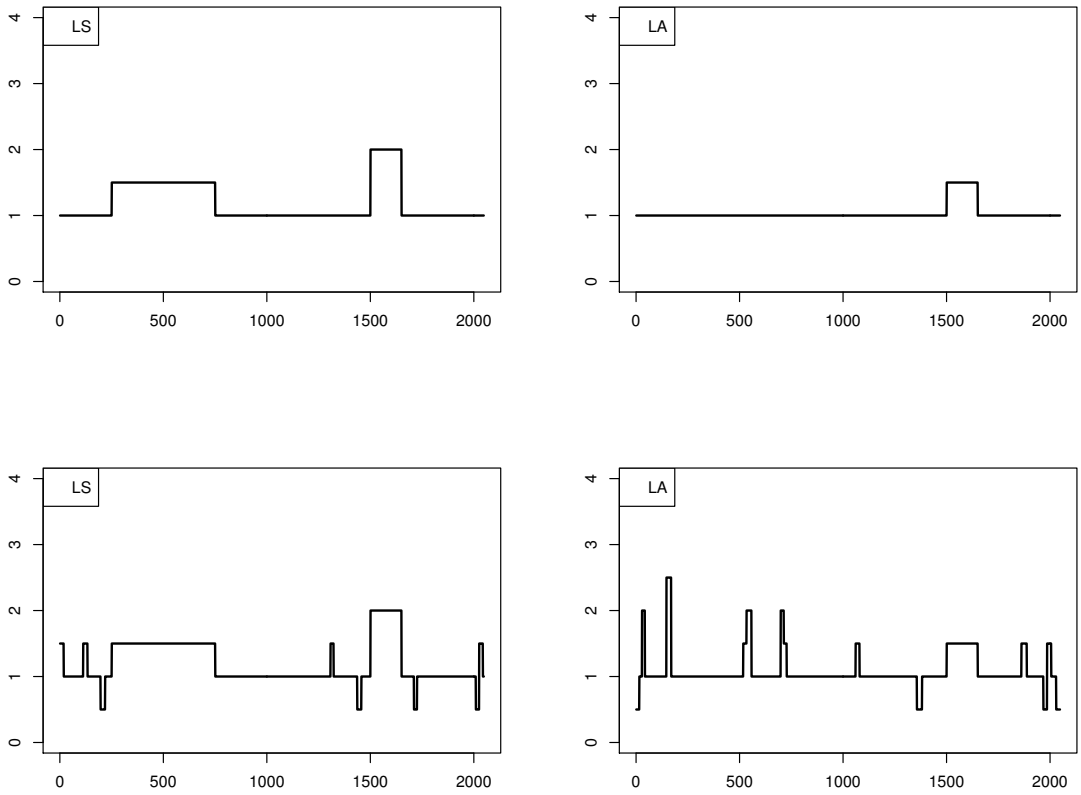


Figure 6.14: Top row: The fixed altered regions of each of the LS (left) and LA (right) groups. Bottom row: An example of the simulated test function of each of the groups before noise contamination.

Figure 6.15 shows the misclassification error of each of the logistic models tested. The results show that only models with scaling coefficients from each of the scales 4, 5, 6, 7, 8, 9, and 10 as predictors are significantly better than the

model with untransformed segmented CNA data and scale 9 is the best among those. The length of wavelets used to extract scale 9 NDWT scaling coefficients is $2^9 = 512$. This indicates that the length of the feature that contributes the most to the classification is close to 512 and this is consistent with the length of the larger altered segment is 500. It can be confirmed by looking at the plot of the frequency of times nonzero $\beta$ are estimated for scale-9 as shown in Figure 6.16, the variables that more informative to distinguish the two groups correspond to the larger altered region.

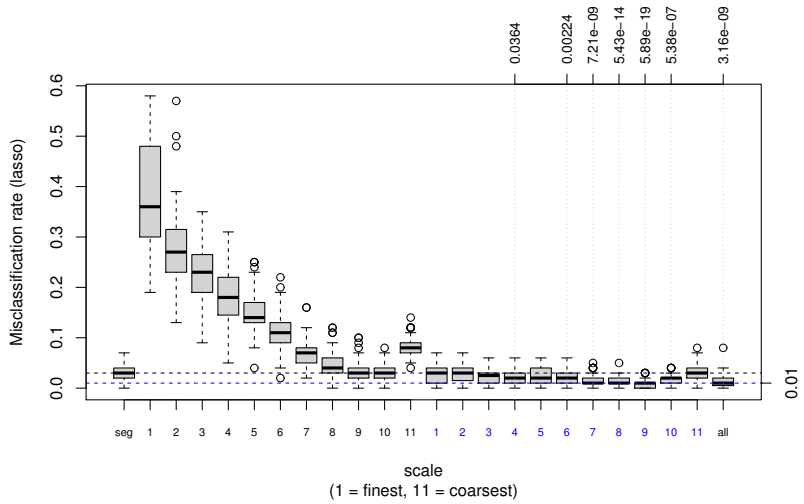


Figure 6.15: Misclassification rate of the seventh simulated dataset from 4-fold cross-validations of 100 datasets (for each dataset 75 samples in the training set and 25 samples in the validation set) where the predictors are NDWT detail (denoted by black x-axis label) and scaling (denoted by blue x-axis label) coefficients across different scales. The 'seg' and 'all' labels indicate the result for untransformed segmented CNA data and NDWT coefficients from all the scales as predictors, respectively. The upper x-axis label shows the p-values of the models that are significantly lower than the 'seg' model.

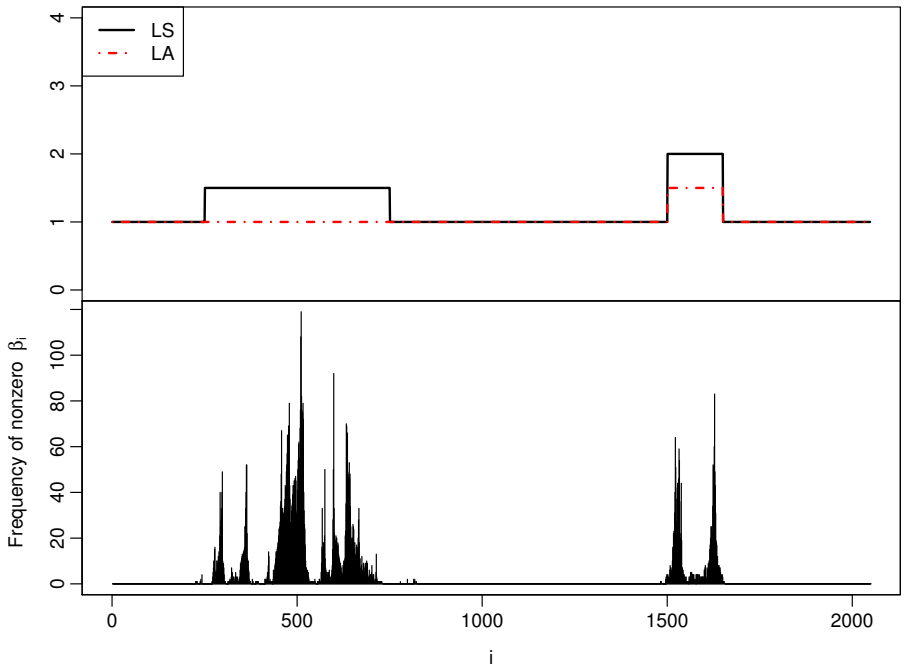


Figure 6.16: Frequency of times nonzero $\beta$ are estimated for model with scale-9 of scaling coefficients over 4-folds cross-validation of 100 dataset.

## 6.4 Application to Real Data

In this section, the proposed classification procedure was applied to the real copy number dataset as described in Section 6.2.1. The four-fold cross-validation was performed 100 times, where, out of 76 observations, 57 (75%) observations are randomly selected to be in the training set and the remaining 19 (25%) observations are in the validation set. For each of the 100 iterations of cross-validation, the same dataset was used but with different arrangements of training and validation set.

Based on the result in Figure 6.17, the lowest misclassification error rate is given by scale-4 scaling coefficients followed by scale-4 detail coefficients. Further investigation in the non-zero $\beta$ estimates as shown in Figures 6.18 and 6.19 indicate that for classification using the scaling coefficients, the most frequently chosen variables are from chromosomes 3, 10 and 17 while for detail coefficients are from chromosomes 3, 10, and 14. Compared the plot of the frequency of non-zero $\beta$ in

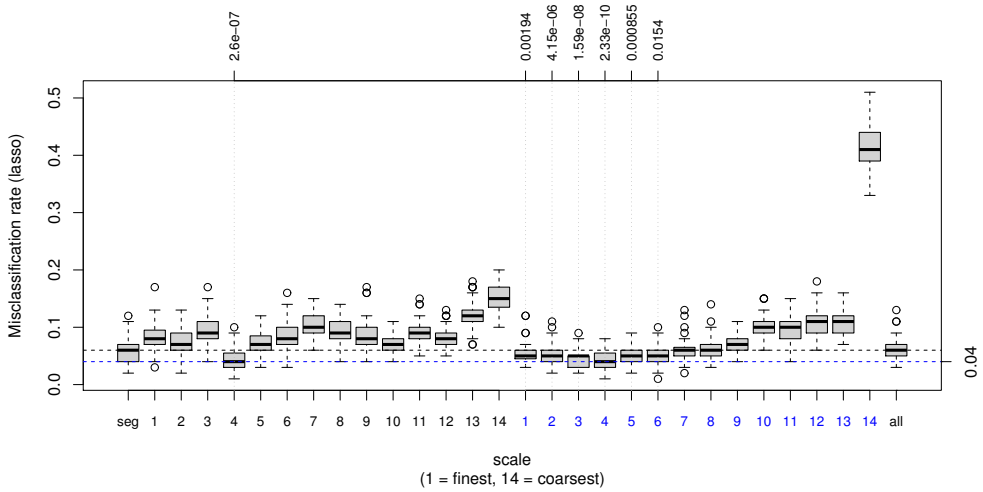


Figure 6.17: Misclassification rate from 4-fold cross-validations of 100 datasets (for each dataset 75 samples in the training set and 25 samples in the validation set) where the predictors are NDWT detail (denoted by black x-axis label) and scaling (denoted by blue x-axis label) coefficients across different scales. The 'seg' and 'all' labels indicate the result for the untransformed segmented CNA data and NDWT coefficients from all the scales as predictors, respectively. The upper x-axis label shows the p-values of the results that are significantly lower than 'seg' model.

Figures 6.18 and 6.19, the frequency of non-zero $\beta$ of scaling coefficients is more dispersed than the detail coefficients. Each scaling coefficient brings information about the scaled average of its surroundings while the detail coefficients bring information about sudden jumps/drops in the original data. This causes the scaling coefficients to have more variables to be chosen so that in most cases, scaling coefficients offer better results.

One advantage of the use of the detail coefficients in the classification is to identify the location of the sudden jumps/drops that contribute to the classification of the LA and LS groups. For example, in Figure 6.19, there is a variable that is highly chosen by Lasso regularisation which is located close to the transition between chromosome 3 and 4 and its value is positive. This indicates that there is a sudden drop in the CNA of the LS group that is significant for LA and LS group classification.

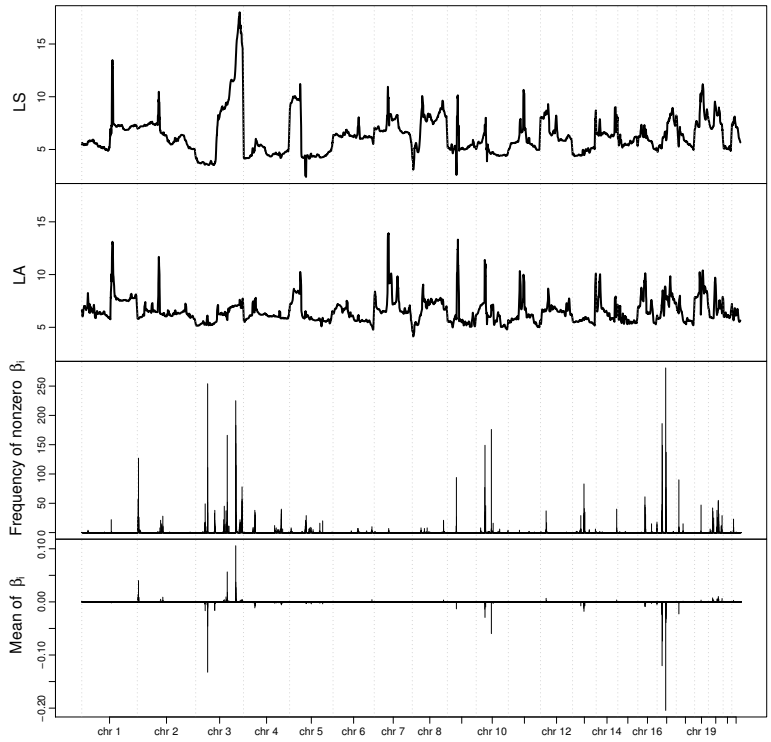


Figure 6.18: Plot of the results for a logistic regression model that only allows scaling coefficients from scale-4 to be chosen by Lasso regularisation. First row: plot of scale-4 NDWT scaling coefficients of segmented LS data averaged over 38 patients. Second row: plot of scale-4 NDWT scaling coefficients of segmented LA data averaged over 38 patients. Third row: Frequency of nonzero $\beta$ coefficients over 100 times cross-validation of logistic regression. Fourth row: mean of the magnitude of $\beta$ over 100 times cross-validation.

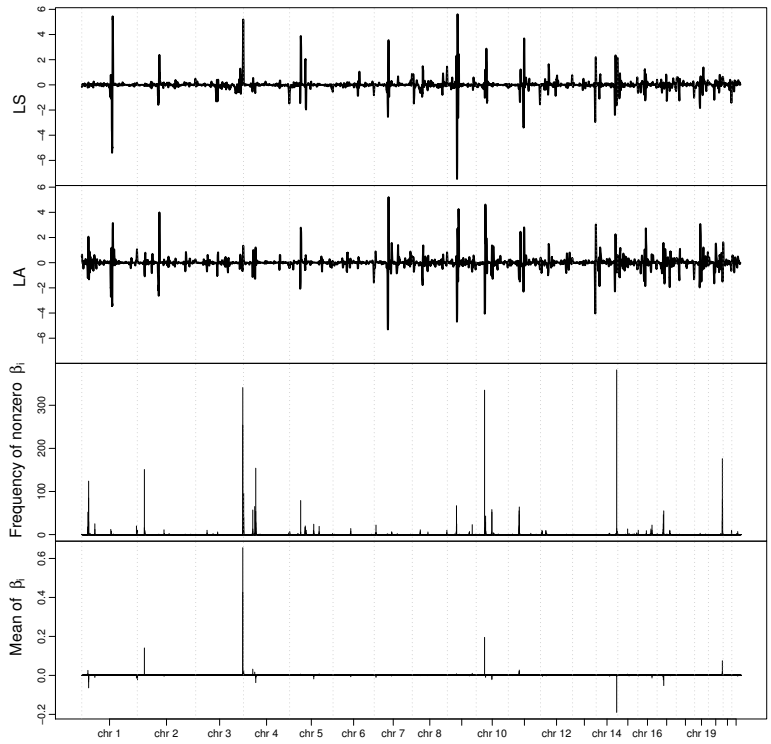


Figure 6.19: Plot of the results for a logistic regression model that only allows scaling coefficients from scale-4 to be chosen by Lasso regularisation. First row: plot of scale-4 NDWT detail coefficients of segmented LS data averaged over 38 patients. Second row: plot of scale-4 NDWT detail coefficients of segmented LA data averaged over 38 patients. Third row: Frequency of nonzero $\beta$ coefficients over 100 times cross-validation of logistic regression. Fourth row: mean of the magnitude of $\beta$ over 100 times cross-validation.

Table 6.1: Confusion Matrix

| | Predicted LS | Predicted LA |
|---|---|---|
| Actual LS | TP = 36 | FN = 2 |
| Actual LA | FP = 1 | TN = 37 |

For a better illustration, Table 6.1 shows the confusion matrix from an iteration of cross-validation of logistic regression with scale-4 detail coefficients as the predictors which give a misclassification rate equal to 0.04. The misclassification for this case is quite low but in the medical context, misclassification errors in distinguishing between lung squamous carcinoma and lung adenocarcinoma are crucial as they can lead to improper treatment choices, delayed interventions, and inaccurate prognostic information. Therefore, a careful diagnosis from a biological point of view is still needed. However, this classification approach is effective in identifying, for example, potential biomarkers that might not be obvious to human experts.

## 6.5 Conclusion

In the context of discovering an appropriate medicine for lung cancer, a main objective for statistical modelling is the accurate prediction of tumours' subtypes. In this chapter, a wavelet-based classification framework is presented. Wavelets provide a tool to extract features of the data in several scales, allowing the classification technique (logistic regression with Lasso regularisation in our case) to select from localised means and differences over a range of scales. The wavelet transformation decomposes the original data into detail (localised difference) and scaling (localised means) coefficients into different resolution levels. This would bring an advantage to discovering hidden features or information that are difficult to find from original data only. Each resolution level corresponds to different lengths of wavelet basis and by considering which levels are most useful in a model, the length of the region that may be responsible for the prediction can be identified.

Analysis of the simulated data indicates that scaling coefficients consistently present the best misclassification rate, which means that the important information of CNA data lies in the means of explanatory variables. The NDWT scaling coefficients have characteristics similar to the moving average and higher (or larger) the scale, more variables will carry out the altered segment information. It will give more variable selection to be chosen compared to the detail coefficients which only carry out the discontinuity of the data in the logistic regression with the Lasso penalty.

But it is need to note here that this works where the simulated data of each patient are generated independently. In a case where there is when the simulated data are highly correlated in blocks, where the location of change-points (or discontinuity) is easier to identify, the detail coefficient might be more informative.

It has been discussed that the lowest misclassification error rate tends to be produced by models with NDWT scaling coefficients corresponding to wavelets that have lengths close to the key region. Therefore, by comparing the misclassification error rate of models across different scales, it is possible to identify the approximate length of the region that contributes the most to distinguish the two groups. Furthermore, by observing the sparse solution given by Lasso regularisation, its variable selection effect makes it easier to identify important variables responsible for prediction.

Regarding the wavelet-based cancer subtype prediction framework, currlently, only logistic regression is used as the primary classification tool. However, it is acknowledged that other classification methods, such as random forest, support vector machines, neural networks, and K-Nearest Neighbors, could be considered as alternatives. To better understand which method is best suited for the proposed wavelet-based lung cancer subtype prediction framework, a simulation study is planned to be conducted in future work.

From the analysis of the real data, the lowest misclassification error rate is given by scale-4, which indicates that the length important feature that contributes the most to the subtypes prediction is approximately $2^4 = 16$ point width or in our case around 2.4 Mb. Further investigation of the sparse solution results showed that the most frequently chosen variables are from chromosomes 3, 10, and 17.

# Chapter 7

# Conclusion

## 7.1 Summary

The thesis focused on the multiscale analysis of DNA copy number alteration using wavelets, specifically in the development of copy number segmentation methods and the prediction of cancer subtypes using wavelets.

In Chapter 3, a comparison study of three kinds of Haar wavelet-based segmentation methods is conducted; (i)the basic Haar wavelet denoising method using universal thresholding (Donoho & Johnstone, 1994), (ii)the HaarSeg method (Ben-Yaacov & Eldar, 2008), and (iii) the tail-greedy unbalanced Haar (TGUH) method (Fryzlewicz, 2018). Analysis of the simulated and real data suggests that the tail-greedy unbalanced Haar (TGUH) method has good operating characteristics to detect segments of different sizes and provide a clear segmentation result compare to the 'balanced' Haar wavelet-based methods. The original Haar wavelet and HaarSeg methods which utilise the 'balanced' Haar wavelets have a tendency to identify more spurious breakpoints due to the dyadic structure of the balanced Haar wavelet transformation. Only the TGUH method offers clean segmentation results with high sensitivity but a low false positive rate.

But further analysis of the TGUH segmentation results has shown that the occurrence of extreme observation (outliers) in NGS data causes the TGUH method to estimate spurious change points as spikes (very short altered segments of only one or two data points). Chapter 4 particularly focused on the investigation of these spurious change points and also proposed a modification to the TGUH

method to reduce them. The extremely short length of either of the unbalanced Haar wavelet wings used in TGUH transformation becomes the main cause of the occurrence of spikes. Spikes are likely to occur when the detail coefficients that correspond to these extremely short-wing unbalanced Haar wavelets survive the thresholding.

To address this problem, the TGUH method was adapted for use with copy number data by modifying its thresholding technique so that it is no longer constrained to the 'unary-binary tree' structure. This modified TGUH method is named the TGUHm method. In the TGUHm method, an additional procedure named unconnected thresholding was added to the connected thresholding used in the original TGUH method. The simulation study has shown that this additional thresholding procedure is effective in reducing the spikes.

In Chapter 5, based on a good performance of the TGUHm method shown in Chapter 4, the data-driven wavelet-Fisz methodology (Fryzlewicz, 2008) further was combined with the TGUHm method for handling non-negative data with heteroscedastic noise whose variance is a non-decreasing function of the mean. Actually, CNA data, as illustrated in Figure 5.1, often exhibit a feature where the noise variance may be linked to the mean level of the data where the variance increases as the mean level increases. This method was named as data-driven TGUH-Fisz (DDTF) method.

The proposed DDTF method was developed to address two key challenges for change-point detection in CNA data. The first challenge is the presence of non-constant random variation in the data where the variance exhibits some association with the mean. The second one is with such non-constant error variance in the copy number data, the detection of short segments is extremely challenging with some spurious changes often detected. In the DDTF method, the first challenge was handled by a variance stabilisation method that combine Fisz transform and the TGUH transform. This transformation has shown effective to bring the TGUH detail coefficients approximately Gaussian with mean zero and variance one. Then the second problem was addressed by applying the TGUHm thresholding procedure in the thresholding stage of the DDTF method.

The simulation study in Chapter 5 suggested that the proposed DDTF method offers excellent results in terms of estimating change-point locations, especially

in estimating short segments. This advantage is also found in some of the data-driven Haar-Fisz (DDHF)-based methods but it is followed by a high false positive rate due to the Haar wavelet transformation used in variance stabilisation and reconstruction stages. Unlike those methods, the DDTF method replaces the use of the balance Haar wavelet transform with the unbalanced Haar wavelet transform. This enables us to match the likely structure of the data by adjusting the breakpoint of the unbalanced Haar wavelets which results in more accurate estimates of change points. The spurious change points at dyadic locations that often occur in the DDHF-based methods are well addressed by the DDTF method. This is important for the identification of copy number alterations as the alterations may occur in any location in the genome.

From Chapters 4 and 5, two unbalanced wavelet-based segmentation methods have been introduced. Those methods can be used to separate noise from the CNA data resulting chromosomes to split into regions of equal copy number. The resulting CNA estimates can then be processed into a classification procedure. Chapter 6 aimed to explore the use of wavelets in this procedure. Particularly, to analyse the circumstances under which wavelet-transformed variables have a better classification performance.

A wavelet-based classification framework was proposed in Chapter 6 which employs the non-decimated Haar wavelet transform to extract localised differences and means of the original data into several scales. Analysis of the simulated data indicates that when the noise of the simulated data is generated independently for each patient, scaling coefficients are consistent to present the best misclassification rate which means that the important information of CNA data lies in the means of explanatory variables. It has been discussed that the lowest misclassification error rate tends to be produced by the model with NDWT scaling coefficients corresponding to wavelets that have lengths close to the key region. Therefore, by comparing the misclassification rate over different scales, one can identify the approximate length of the region that contributes the most to distinguish the two groups. Furthermore, by observing the sparse solution given by Lasso regularisation, its variable selection effect make it easier to identify important variables responsible for prediction.

## 7.2 Future Work

It has been seen that compared to the Haar wavelet-based segmentation method and some other well-known segmentation methods, the TGUHm and DDTF segmentation methods perform very well. This thesis only explored the use of those methods in CNA segmentation, but in practice, they can be used for a wider range of data structures. For example for the prediction of transmembrane helix locations Lio & Vannucci (2000), estimation of phase transitions in pain symptoms Desmond *et al.* (2002), speech segmentation Shriberg *et al.* (2000), and adaptive trend estimation in markets Schroeder & Fryzlewicz (2013).

But currently, those methods are particularly designed for a change-point model with homoscedastic noise or heteroscedastic data where there is a non-decreasing relationship between mean and variance. When there is a different level of noise variance at the same mean, the proposed method still needs further extension. An example of this condition is when there is a burst in the centromere region (see Figure). This is a subject for future research.

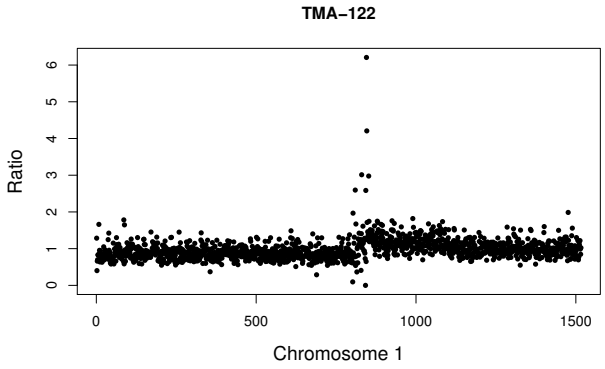


Figure 7.1: The observed copy number ratio of chromosome 1 from TMA-122 patient data.

One possibility to deal with this problem is by considering the spike and slab model. The main idea of this model is to set the prior of the parameter to have mass at zero which is suitable to model the sparsity of the detail coefficient $d^{j,k}$ of the true function $f$. This technique is similar to the model in the Ebayesh threshold

(Johnstone & Silverman, 2005a) which sets each of the $d^{j,k}$ as zero with probability $(1-w)$, while, with probability $w$, $d^{j,k}$ is drawn from a symmetric heavy-tailed density. Then $w$ is chosen automatically from the data, using a marginal maximum likelihood approach, and then substituted back into the Bayesian model. But in our case, the aim is to use Bayesian to estimate the location of the variance instability in the real data. Therefore, to better illustrate the variance instability in the real data, the following model may be considered

$$\varepsilon_i | p_i \sim N(0, \sigma_i^2(h(\mu_i) + p_i k)) \tag{7.1}$$

$$p_i \sim Bernoulli(\omega) \tag{7.2}$$

and to adjust the burst location, set

$$\omega = \omega_0 I(i^s \leq i \leq i^e) \tag{7.3}$$

where $I(\cdot)$ is an indicator function, $i^s$ and $i^e$ are the start and end of the location where the burst is located, respectively. Figure 7.2 shows data generated from this model with $h(\mu_i) = 0.2\mu_i$, $\omega_0 = 0.35$, $k = 4$, $i^s = 780$ and $i^e = 820$. This model seems to fit the copy number data better. But of course, further work is needed to evaluate whether this approach is effective in reducing the influence of the burst in the final estimator.

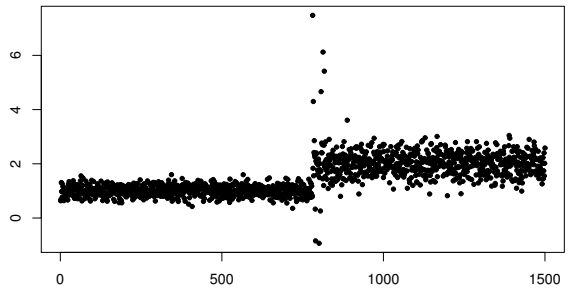


Figure 7.2: the spike and slab model with $\omega_0 = 0.35$ and $k = 4$.

Besides the above spikes and slab model, another simple way that might work to deal with the sudden burst near the centromere region is by applying

a pre-processing procedure, for example, the wisorization used in Nilsen *et al.* (2012) prior to the proposed segmentation methods. Winsorization is a simple transformation to reduce the outliers by replacing a specified number of extreme values outside a certain fractile with a smaller/higher data value which belongs to that range. For identically distributed observations $y_1, \ldots, y_p$, the corresponding Winsorized observations are defined as $y_j^w = \psi(y_j)$ where

$$\psi(y_j) = \psi(y_j|c) = \begin{cases} -c, & y_j < -c \\ y_j, & |y_j| < c \\ c, & y_j > c. \end{cases}$$

Here, $c > 0$ determines how extreme an observation must be to be relocated, as well as the replacement value. A common choice is $c = \tau s$, where typically $\tau \in [1.5, 3]$ and $s$ is a robust estimate of the standard deviation (SD). A robust scale estimator is the Median Absolute Deviation (MAD), defined as the median of the values $|y_j - \hat{m}|$, where $\hat{m}$ is the median of $y_1, \ldots, y_p$.

Winsorization of copy number data can be achieved by first estimating the trend in the data and then Winsorizing the residuals. Let the observations represent a copy numbers ratio in $p$ location be $y = (y_1, \ldots, y_p)$, ordered according to the genomic position. A simple estimator of the trend is the median filter. The trend estimate $\hat{m}_j$ in the $j$th position is then given by the median of $y_{j-k}, \ldots, y_{j+k}$ for some $k > 0$. The SD of the residuals $y_j - \hat{m}_j$ may then be estimated with the MAD estimator $s_M$, and Winsorized observations $y_1^w, \ldots, y_p^w$ obtained by $y_j^w = \hat{m}_j + \psi(y_j - \hat{m}_j|\tau s_M)$.

In terms of the wavelet-based cancer subtype prediction framework, currently, only logistic regression is considered as the tool to perform classification. However, other classification methods such as random forest, support vector machines, neural networks, and K-Nearest Neighbors can be the alternative to this. A simulation study that compares those methods would be needed in our future work to determine the most suitable method for the proposed wavelet-based lung cancer subtypes prediction framework.

Also, it is important to notice that there can be "leakage" between different levels of the NDWT. When the length of the key segment is somewhere between two levels, there may be difficulty in choosing a single "best" level. For future

work, continuous wavelet transformation can be considered to fill this 'gap' in wavelet length between the scales. This may provide a better misclassification rate and more accurate information on the length of the important region.

# Appendix A

# Additional Tables of Section 4.4

## A.1 Tables related to Figure 4.7

Table of the first, second, and third simulation study in chapter 4.4.2 (which corresponds to Figure 4.6, 4.7, and 4.6): The average of mean-square error (aMSE), average of true positive rate (aTPR), average of false positive rate (aFPR), and average of true positive rate in estimating change-points that corresponds to short segments (aTPRsh) over 1000 replicates of the simulation that were contaminated by Gaussian noise $N(0, \sigma^2)$ and Gaussian mixture noise $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$.

Table A.1: Table of the first simulation study in the main article (which corresponds to Figure 4.6 of the main article)

| | | Gaussian noise | | | | Gaussian mixture noise | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\sigma$ | Method | aMSE | aTPR | aFPR | aTPRsh | aMSE | aTPR | aFPR | aTPRsh |
| 0.1 | TGUHm | 0.00030 | 100.00 | 0.030 | 100.00 | 0.00060 | 99.87 | 0.068 | 99.76 |
| | TGUH1 | 0.00032 | 100.00 | 0.046 | 100.00 | 0.00294 | 99.93 | 1.443 | 99.88 |
| | TGUH | 0.00030 | 100.00 | 0.030 | 100.00 | 0.00063 | 99.87 | 0.078 | 99.76 |
| | TGUHb | 0.00044 | 99.47 | 0.064 | 98.88 | 0.00094 | 97.73 | 0.063 | 95.28 |
| | CBS | 0.00028 | 99.99 | 0.001 | 99.98 | 0.00052 | 99.79 | 0.012 | 99.57 |
| | HaarSeg | 0.00051 | 100.00 | 0.152 | 99.99 | 0.00114 | 99.92 | 0.392 | 99.85 |
| | Copy12 | 0.00046 | 99.26 | 0.015 | 98.44 | 0.00073 | 98.85 | 0.032 | 97.64 |
| | Copy40 | 0.00095 | 95.99 | 0.007 | 91.57 | 0.00164 | 92.75 | 0.011 | 84.84 |
| | CumSeg | 0.01158 | 78.58 | 0.233 | 60.31 | 0.01236 | 74.31 | 0.209 | 51.17 |
| | FDRSeg | 0.00036 | 100.00 | 0.070 | 100.00 | 0.00304 | 99.98 | 1.498 | 99.95 |
| 0.2 | TGUHm | 0.00241 | 96.88 | 0.083 | 94.27 | 0.00404 | 93.27 | 0.214 | 87.59 |
| | TGUH1 | 0.00247 | 96.93 | 0.101 | 94.37 | 0.01320 | 93.60 | 1.514 | 88.16 |
| | TGUH | 0.00241 | 96.88 | 0.083 | 94.27 | 0.00516 | 93.30 | 0.307 | 87.64 |
| | TGUHb | 0.00459 | 88.14 | 0.218 | 75.86 | 0.00568 | 82.15 | 0.177 | 64.03 |
| | CBS | 0.00239 | 96.26 | 0.081 | 92.49 | 0.00493 | 85.03 | 0.099 | 69.61 |
| | HaarSeg | 0.00419 | 88.37 | 0.418 | 75.92 | 0.00690 | 84.01 | 0.680 | 67.35 |
| | Copy12 | 0.00387 | 86.07 | 0.068 | 71.36 | 0.00566 | 80.22 | 0.116 | 59.65 |
| | Copy40 | 0.01032 | 52.08 | 0.006 | 0.03 | 0.01083 | 51.76 | 0.012 | 0.10 |
| | CumSeg | 0.01887 | 51.54 | 0.103 | 5.25 | 0.02183 | 48.88 | 0.104 | 2.52 |
| | FDRSeg | 0.00264 | 95.76 | 0.107 | 91.57 | 0.01343 | 92.82 | 1.515 | 85.97 |
| 0.3 | TGUHm | 0.00763 | 79.15 | 0.187 | 60.36 | 0.01160 | 70.40 | 0.327 | 44.42 |
| | TGUH1 | 0.00777 | 79.29 | 0.206 | 60.65 | 0.03241 | 71.86 | 1.592 | 47.30 |
| | TGUH | 0.00764 | 79.16 | 0.187 | 60.37 | 0.01496 | 70.54 | 0.455 | 44.70 |
| | TGUHb | 0.00979 | 71.10 | 0.288 | 42.86 | 0.01182 | 64.98 | 0.213 | 31.83 |
| | CBS | 0.00843 | 73.00 | 0.146 | 45.93 | 0.01329 | 58.74 | 0.110 | 19.99 |
| | HaarSeg | 0.01172 | 60.45 | 0.498 | 19.78 | 0.01637 | 60.05 | 0.809 | 20.72 |
| | Copy12 | 0.01048 | 58.12 | 0.054 | 14.83 | 0.01251 | 57.74 | 0.107 | 15.45 |
| | Copy40 | 0.01446 | 48.23 | 0.020 | 0.00 | 0.01651 | 47.15 | 0.026 | 0.02 |
| | CumSeg | 0.02692 | 45.74 | 0.108 | 0.69 | 0.03096 | 43.37 | 0.123 | 0.24 |
| | FDRSeg | 0.00963 | 69.25 | 0.173 | 38.22 | 0.03274 | 68.30 | 1.539 | 38.24 |
| 0.4 | TGUHm | 0.01425 | 61.34 | 0.202 | 27.81 | 0.01989 | 54.28 | 0.329 | 16.90 |
| | TGUH1 | 0.01454 | 61.53 | 0.223 | 28.21 | 0.05764 | 56.15 | 1.620 | 20.55 |
| | TGUH | 0.01427 | 61.35 | 0.203 | 27.83 | 0.02527 | 54.40 | 0.438 | 17.12 |
| | TGUHb | 0.01556 | 59.86 | 0.298 | 22.84 | 0.01867 | 53.64 | 0.239 | 13.06 |
| | CBS | 0.01458 | 56.71 | 0.152 | 16.39 | 0.02128 | 47.65 | 0.110 | 5.06 |
| | HaarSeg | 0.01757 | 51.26 | 0.603 | 5.80 | 0.02506 | 51.47 | 0.902 | 8.25 |
| | Copy12 | 0.01391 | 51.00 | 0.068 | 3.31 | 0.01776 | 50.54 | 0.132 | 5.02 |
| | Copy40 | 0.02518 | 42.08 | 0.018 | 0.00 | 0.02811 | 41.03 | 0.027 | 0.01 |
| | CumSeg | 0.03522 | 41.95 | 0.135 | 0.22 | 0.04369 | 38.94 | 0.143 | 0.11 |
| | FDRSeg | 0.01506 | 55.21 | 0.194 | 12.61 | 0.05539 | 55.30 | 1.512 | 16.43 |
| 0.5 | TGUHm | 0.02079 | 51.61 | 0.200 | 12.38 | 0.02885 | 47.18 | 0.333 | 8.55 |
| | TGUH1 | 0.02120 | 51.74 | 0.218 | 12.66 | 0.08783 | 49.30 | 1.609 | 12.40 |
| | TGUH | 0.02081 | 51.61 | 0.200 | 12.38 | 0.03678 | 47.28 | 0.435 | 8.73 |
| | TGUHb | 0.02201 | 52.58 | 0.282 | 12.30 | 0.02700 | 47.44 | 0.229 | 6.36 |
| | CBS | 0.02052 | 49.08 | 0.150 | 6.42 | 0.03076 | 41.86 | 0.102 | 1.71 |
| | HaarSeg | 0.02445 | 47.20 | 0.693 | 2.44 | 0.03560 | 46.92 | 0.987 | 4.38 |
| | Copy12 | 0.01800 | 47.77 | 0.087 | 1.08 | 0.02470 | 46.45 | 0.164 | 2.46 |
| | Copy40 | 0.03455 | 38.35 | 0.013 | 0.00 | 0.03872 | 37.35 | 0.026 | 0.00 |
| | CumSeg | 0.04762 | 37.70 | 0.142 | 0.10 | 0.05880 | 34.62 | 0.152 | 0.08 |
| | FDRSeg | 0.01983 | 49.34 | 0.176 | 5.08 | 0.08259 | 49.10 | 1.472 | 9.31 |

Table A.2: Table of the first simulation study in the main article (which corresponds to Figure 4.7 of the main article).

| | | Gaussian noise | | | | Gaussian mixture noise | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\sigma$ | Method | aMSE | aTPR | aFPR | aTPRsh | aMSE | aTPR | aFPR | aTPRsh |
| 0.1 | TGUHm | 0.00025 | 100.00 | 0.035 | 100.00 | 0.00065 | 99.94 | 0.169 | 99.94 |
| | TGUH1 | 0.00026 | 100.00 | 0.054 | 100.00 | 0.00295 | 99.96 | 1.538 | 99.96 |
| | TGUH | 0.00025 | 100.00 | 0.035 | 100.00 | 0.00072 | 99.94 | 0.204 | 99.94 |
| | TGUHb | 0.00053 | 98.11 | 0.060 | 98.11 | 0.00116 | 96.21 | 0.093 | 96.21 |
| | CBS | 0.00023 | 99.97 | 0.002 | 99.97 | 0.00046 | 99.38 | 0.014 | 99.38 |
| | HaarSeg | 0.00045 | 99.99 | 0.132 | 99.99 | 0.00104 | 99.89 | 0.368 | 99.89 |
| | Copy12 | 0.00021 | 100.00 | 0.002 | 100.00 | 0.00046 | 99.87 | 0.044 | 99.87 |
| | Copy40 | 0.00021 | 99.91 | 0.000 | 99.91 | 0.00044 | 99.31 | 0.002 | 99.31 |
| | CumSeg | 0.01722 | 55.80 | 0.264 | 55.80 | 0.01862 | 47.31 | 0.231 | 47.31 |
| | FDRSeg | 0.00027 | 100.00 | 0.046 | 100.00 | 0.00273 | 99.98 | 1.304 | 99.98 |
| 0.2 | TGUHm | 0.00207 | 95.75 | 0.087 | 95.75 | 0.00399 | 91.74 | 0.280 | 91.74 |
| | TGUH1 | 0.00215 | 95.79 | 0.108 | 95.79 | 0.01308 | 92.05 | 1.585 | 92.05 |
| | TGUH | 0.00207 | 95.75 | 0.087 | 95.75 | 0.00558 | 91.79 | 0.427 | 91.79 |
| | TGUHb | 0.00787 | 77.33 | 0.235 | 77.33 | 0.01154 | 67.15 | 0.255 | 67.15 |
| | CBS | 0.00248 | 90.32 | 0.053 | 90.32 | 0.00482 | 75.99 | 0.060 | 75.99 |
| | HaarSeg | 0.00388 | 84.13 | 0.477 | 84.13 | 0.00643 | 77.73 | 0.697 | 77.73 |
| | Copy12 | 0.00214 | 92.91 | 0.060 | 92.91 | 0.00368 | 88.34 | 0.159 | 88.34 |
| | Copy40 | 0.00944 | 41.70 | 0.003 | 41.70 | 0.00987 | 41.21 | 0.006 | 41.21 |
| | CumSeg | 0.02436 | 21.50 | 0.153 | 21.50 | 0.02630 | 18.33 | 0.153 | 18.33 |
| | FDRSeg | 0.00225 | 93.20 | 0.066 | 93.20 | 0.01211 | 90.45 | 1.304 | 90.45 |
| 0.3 | TGUHm | 0.00676 | 73.99 | 0.173 | 73.99 | 0.01119 | 64.91 | 0.396 | 64.91 |
| | TGUH1 | 0.00693 | 74.15 | 0.198 | 74.15 | 0.03222 | 66.06 | 1.676 | 66.06 |
| | TGUH | 0.00677 | 73.99 | 0.174 | 73.99 | 0.01633 | 65.03 | 0.601 | 65.03 |
| | TGUHb | 0.01796 | 47.37 | 0.295 | 47.37 | 0.02227 | 38.99 | 0.275 | 38.99 |
| | CBS | 0.00773 | 61.57 | 0.086 | 61.57 | 0.01290 | 40.68 | 0.049 | 40.68 |
| | HaarSeg | 0.01111 | 47.78 | 0.450 | 47.78 | 0.01495 | 46.58 | 0.620 | 46.58 |
| | Copy12 | 0.00764 | 62.48 | 0.087 | 62.48 | 0.01057 | 58.75 | 0.209 | 58.75 |
| | Copy40 | 0.01732 | 24.69 | 0.008 | 24.69 | 0.01882 | 22.53 | 0.010 | 22.53 |
| | CumSeg | 0.03155 | 12.73 | 0.129 | 12.73 | 0.03731 | 7.86 | 0.070 | 7.86 |
| | FDRSeg | 0.00910 | 57.13 | 0.110 | 57.13 | 0.03021 | 58.81 | 1.331 | 58.81 |
| 0.4 | TGUHm | 0.01188 | 54.58 | 0.171 | 54.58 | 0.01806 | 46.23 | 0.386 | 46.23 |
| | TGUH1 | 0.01217 | 54.76 | 0.194 | 54.76 | 0.05519 | 48.01 | 1.652 | 48.01 |
| | TGUH | 0.01188 | 54.58 | 0.171 | 54.58 | 0.02791 | 46.53 | 0.601 | 46.53 |
| | TGUHb | 0.02549 | 31.24 | 0.279 | 31.24 | 0.02912 | 24.32 | 0.243 | 24.32 |
| | CBS | 0.01228 | 43.50 | 0.069 | 43.50 | 0.02057 | 22.55 | 0.030 | 22.55 |
| | HaarSeg | 0.01694 | 31.60 | 0.282 | 31.60 | 0.02247 | 30.11 | 0.432 | 30.11 |
| | Copy12 | 0.01212 | 43.96 | 0.066 | 43.96 | 0.01645 | 41.62 | 0.182 | 41.62 |
| | Copy40 | 0.02375 | 13.94 | 0.001 | 13.94 | 0.02447 | 13.86 | 0.002 | 13.86 |
| | CumSeg | 0.04154 | 4.31 | 0.036 | 4.31 | 0.04585 | 1.23 | 0.010 | 1.23 |
| | FDRSeg | 0.01443 | 39.17 | 0.107 | 39.17 | 0.05014 | 40.69 | 1.259 | 40.69 |
| 0.5 | TGUHm | 0.01688 | 41.99 | 0.163 | 41.99 | 0.02644 | 34.85 | 0.403 | 34.85 |
| | TGUH1 | 0.01749 | 42.25 | 0.192 | 42.25 | 0.08731 | 37.06 | 1.710 | 37.06 |
| | TGUH | 0.01693 | 42.01 | 0.164 | 42.01 | 0.04302 | 35.09 | 0.623 | 35.09 |
| | TGUHb | 0.03096 | 21.83 | 0.257 | 21.83 | 0.03486 | 17.13 | 0.197 | 17.13 |
| | CBS | 0.01723 | 31.42 | 0.065 | 31.42 | 0.02885 | 12.83 | 0.011 | 12.83 |
| | HaarSeg | 0.02339 | 21.07 | 0.190 | 21.07 | 0.03156 | 21.11 | 0.342 | 21.11 |
| | Copy12 | 0.01666 | 33.48 | 0.065 | 33.48 | 0.02418 | 32.37 | 0.207 | 32.37 |
| | Copy40 | 0.03000 | 10.09 | 0.000 | 10.09 | 0.03333 | 8.40 | 0.001 | 8.40 |
| | CumSeg | 0.04622 | 0.65 | 0.004 | 0.65 | 0.04690 | 0.21 | 0.002 | 0.21 |
| | FDRSeg | 0.02049 | 26.72 | 0.090 | 26.72 | 0.07698 | 30.43 | 1.248 | 30.43 |

Table A.3: Table of the first simulation study in the main article (which corresponds to Figure 4.8 of the main article).

| | | Gaussian noise | | | | Gaussian mixture noise | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\sigma$ | Method | aMSE | aTPR | aFPR | aTPRsh | aMSE | aTPR | aFPR | aTPRsh |
| 0.1 | TGUHm | 0.00011 | 99.90 | 0.067 | 99.90 | 0.00021 | 99.70 | 0.091 | 99.70 |
| | TGUH1 | 0.00013 | 99.90 | 0.097 | 99.90 | 0.00288 | 99.85 | 1.709 | 99.85 |
| | TGUH | 0.00011 | 99.90 | 0.068 | 99.90 | 0.00038 | 99.75 | 0.175 | 99.75 |
| | TGUHb | 0.00068 | 59.45 | 0.078 | 59.45 | 0.00088 | 49.10 | 0.064 | 49.10 |
| | CBS | 0.00005 | 99.70 | 0.001 | 99.70 | 0.00009 | 99.10 | 0.003 | 99.10 |
| | HaarSeg | 0.00007 | 100.00 | 0.010 | 100.00 | 0.00034 | 99.60 | 0.104 | 99.60 |
| | Copy12 | 0.00006 | 100.00 | 0.006 | 100.00 | 0.00031 | 99.85 | 0.142 | 99.85 |
| | Copy40 | 0.00004 | 100.00 | 0.000 | 100.00 | 0.00008 | 98.90 | 0.000 | 98.90 |
| | CumSeg | 0.00097 | 44.95 | 0.016 | 44.95 | 0.00128 | 22.40 | 0.006 | 22.40 |
| | FDRSeg | 0.00006 | 100.00 | 0.007 | 100.00 | 0.00225 | 99.95 | 1.099 | 99.95 |
| 0.2 | TGUHm | 0.00061 | 92.40 | 0.082 | 92.40 | 0.00120 | 83.85 | 0.129 | 83.85 |
| | TGUH1 | 0.00071 | 92.80 | 0.112 | 92.80 | 0.01175 | 85.45 | 1.733 | 85.45 |
| | TGUH | 0.00061 | 92.50 | 0.084 | 92.50 | 0.00306 | 83.90 | 0.324 | 83.90 |
| | TGUHb | 0.00122 | 35.15 | 0.070 | 35.15 | 0.00147 | 22.90 | 0.046 | 22.90 |
| | CBS | 0.00069 | 65.65 | 0.008 | 65.65 | 0.00146 | 11.95 | 0.004 | 11.95 |
| | HaarSeg | 0.00130 | 23.85 | 0.012 | 23.85 | 0.00207 | 21.45 | 0.094 | 21.45 |
| | Copy12 | 0.00040 | 90.25 | 0.012 | 90.25 | 0.00147 | 83.50 | 0.163 | 83.50 |
| | Copy40 | 0.00153 | 0.60 | 0.000 | 0.60 | 0.00154 | 1.40 | 0.000 | 1.40 |
| | CumSeg | 0.00153 | 0.00 | 0.000 | 0.00 | 0.00155 | 0.10 | 0.000 | 0.10 |
| | FDRSeg | 0.00088 | 57.30 | 0.008 | 57.30 | 0.00934 | 63.95 | 1.091 | 63.95 |
| 0.3 | TGUHm | 0.00161 | 57.85 | 0.091 | 57.85 | 0.00281 | 33.70 | 0.143 | 33.70 |
| | TGUH1 | 0.00185 | 58.50 | 0.124 | 58.50 | 0.02649 | 38.00 | 1.725 | 38.00 |
| | TGUH | 0.00163 | 58.05 | 0.095 | 58.05 | 0.00629 | 33.90 | 0.313 | 33.90 |
| | TGUHb | 0.00179 | 15.10 | 0.051 | 15.10 | 0.00193 | 4.05 | 0.027 | 4.05 |
| | CBS | 0.00150 | 13.00 | 0.005 | 13.00 | 0.00167 | 0.15 | 0.002 | 0.15 |
| | HaarSeg | 0.00188 | 11.75 | 0.047 | 11.75 | 0.00408 | 7.95 | 0.185 | 7.95 |
| | Copy12 | 0.00139 | 32.35 | 0.016 | 32.35 | 0.00352 | 27.55 | 0.158 | 27.55 |
| | Copy40 | 0.00158 | 0.00 | 0.000 | 0.00 | 0.00161 | 0.00 | 0.000 | 0.00 |
| | CumSeg | 0.00158 | 0.00 | 0.000 | 0.00 | 0.00161 | 0.00 | 0.000 | 0.00 |
| | FDRSeg | 0.00166 | 6.55 | 0.004 | 6.55 | 0.02015 | 15.85 | 1.049 | 15.85 |
| 0.4 | TGUHm | 0.00267 | 25.40 | 0.090 | 25.40 | 0.00423 | 12.30 | 0.151 | 12.30 |
| | TGUH1 | 0.00306 | 26.10 | 0.119 | 26.10 | 0.04661 | 17.15 | 1.715 | 17.15 |
| | TGUH | 0.00271 | 25.45 | 0.092 | 25.45 | 0.00903 | 12.70 | 0.275 | 12.70 |
| | TGUHb | 0.00215 | 6.75 | 0.041 | 6.75 | 0.00229 | 1.20 | 0.024 | 1.20 |
| | CBS | 0.00168 | 2.30 | 0.002 | 2.30 | 0.00186 | 0.10 | 0.002 | 0.10 |
| | HaarSeg | 0.00304 | 7.75 | 0.134 | 7.75 | 0.00658 | 4.80 | 0.232 | 4.80 |
| | Copy12 | 0.00179 | 8.30 | 0.010 | 8.30 | 0.00562 | 9.90 | 0.160 | 9.90 |
| | Copy40 | 0.00164 | 0.00 | 0.000 | 0.00 | 0.00172 | 0.05 | 0.000 | 0.05 |
| | CumSeg | 0.00164 | 0.00 | 0.000 | 0.00 | 0.00171 | 0.00 | 0.000 | 0.00 |
| | FDRSeg | 0.00176 | 0.90 | 0.003 | 0.90 | 0.03485 | 7.80 | 1.042 | 7.80 |
| 0.5 | TGUHm | 0.00357 | 11.65 | 0.086 | 11.65 | 0.00573 | 6.05 | 0.152 | 6.05 |
| | TGUH1 | 0.00423 | 12.30 | 0.118 | 12.30 | 0.07229 | 10.40 | 1.711 | 10.40 |
| | TGUH | 0.00364 | 11.70 | 0.089 | 11.70 | 0.01224 | 6.25 | 0.257 | 6.25 |
| | TGUHb | 0.00260 | 2.85 | 0.039 | 2.85 | 0.00281 | 1.00 | 0.025 | 1.00 |
| | CBS | 0.00183 | 0.25 | 0.003 | 0.25 | 0.00202 | 0.05 | 0.002 | 0.05 |
| | HaarSeg | 0.00412 | 5.05 | 0.168 | 5.05 | 0.00911 | 2.75 | 0.252 | 2.75 |
| | Copy12 | 0.00201 | 2.40 | 0.008 | 2.40 | 0.00818 | 5.15 | 0.162 | 5.15 |
| | Copy40 | 0.00174 | 0.00 | 0.000 | 0.00 | 0.00189 | 0.00 | 0.000 | 0.00 |
| | CumSeg | 0.00174 | 0.00 | 0.000 | 0.00 | 0.00186 | 0.00 | 0.000 | 0.00 |
| | FDRSeg | 0.00191 | 0.35 | 0.004 | 0.35 | 0.05328 | 4.70 | 1.024 | 4.70 |

# Appendix B

# Additional Figures of Section 4.4

## B.1 Additional figures of the proportion of times change-points estimated at each location: noise model 1

Figures B.1–B.15 show the proportion of times (from 1000 simulated datasets) that each method detects a change-point at each location along the sequence based on 1000 simulated datasets contaminated with Gaussian noise with mean zero and variance $\sigma^2$.

## B.2 Additional figures of the proportion of times change-points estimated at each location: noise model 2

Figures B.16–B.27 show the proportion of times (from 1000 simulated datasets) that each method detects a change-point at each location along the sequence based on 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$.

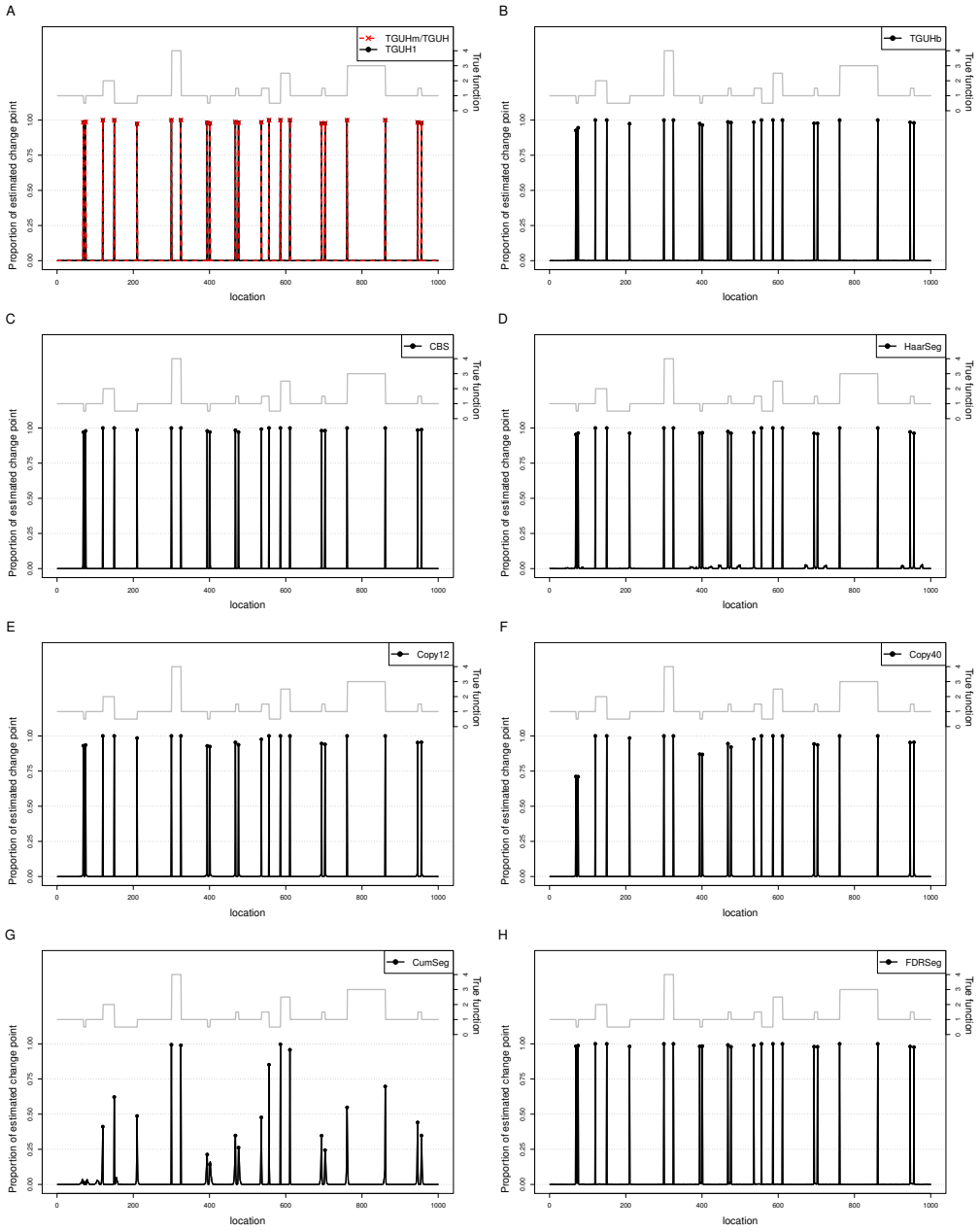


Figure B.1: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.1^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **A** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

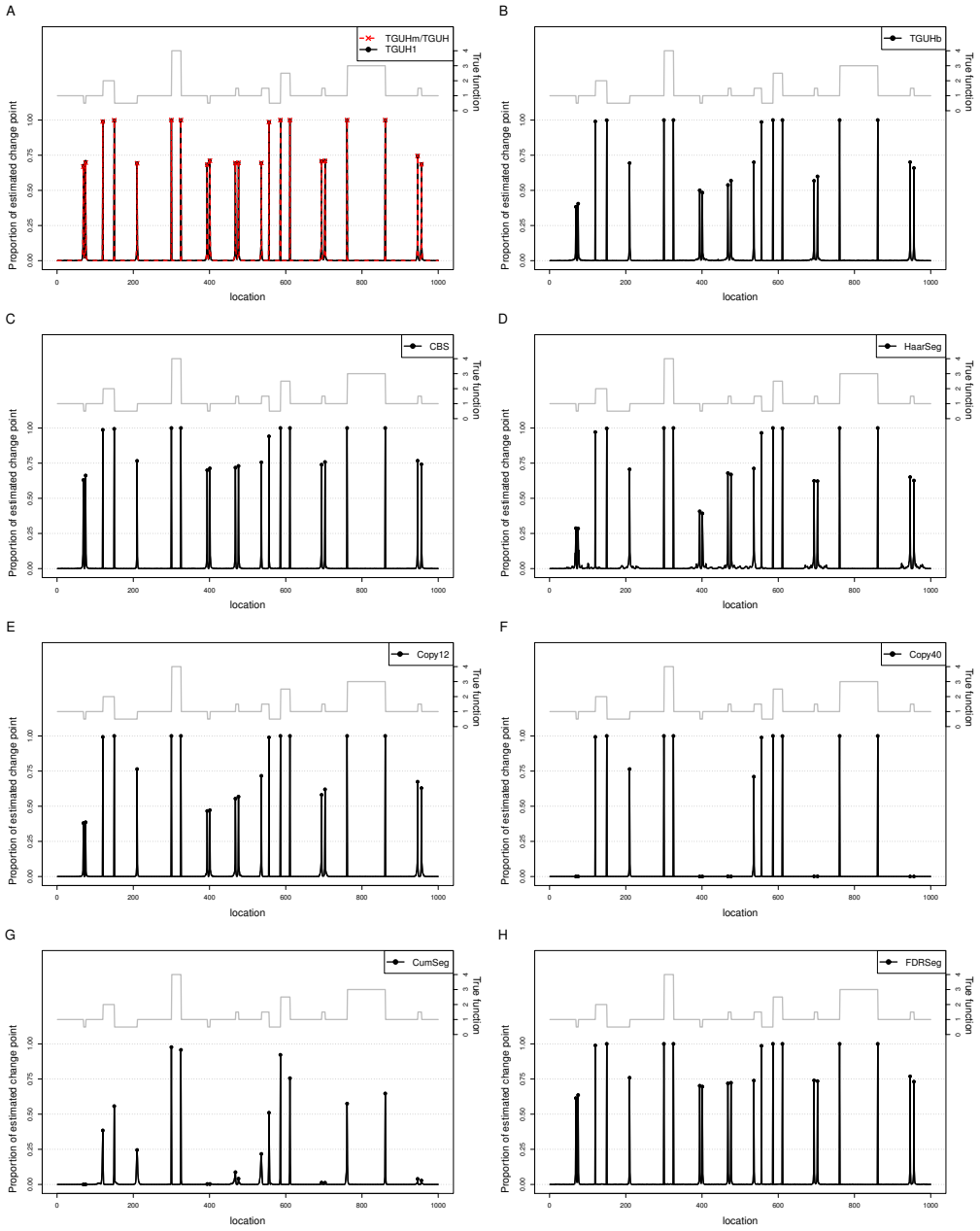


Figure B.2: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.2^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **A** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

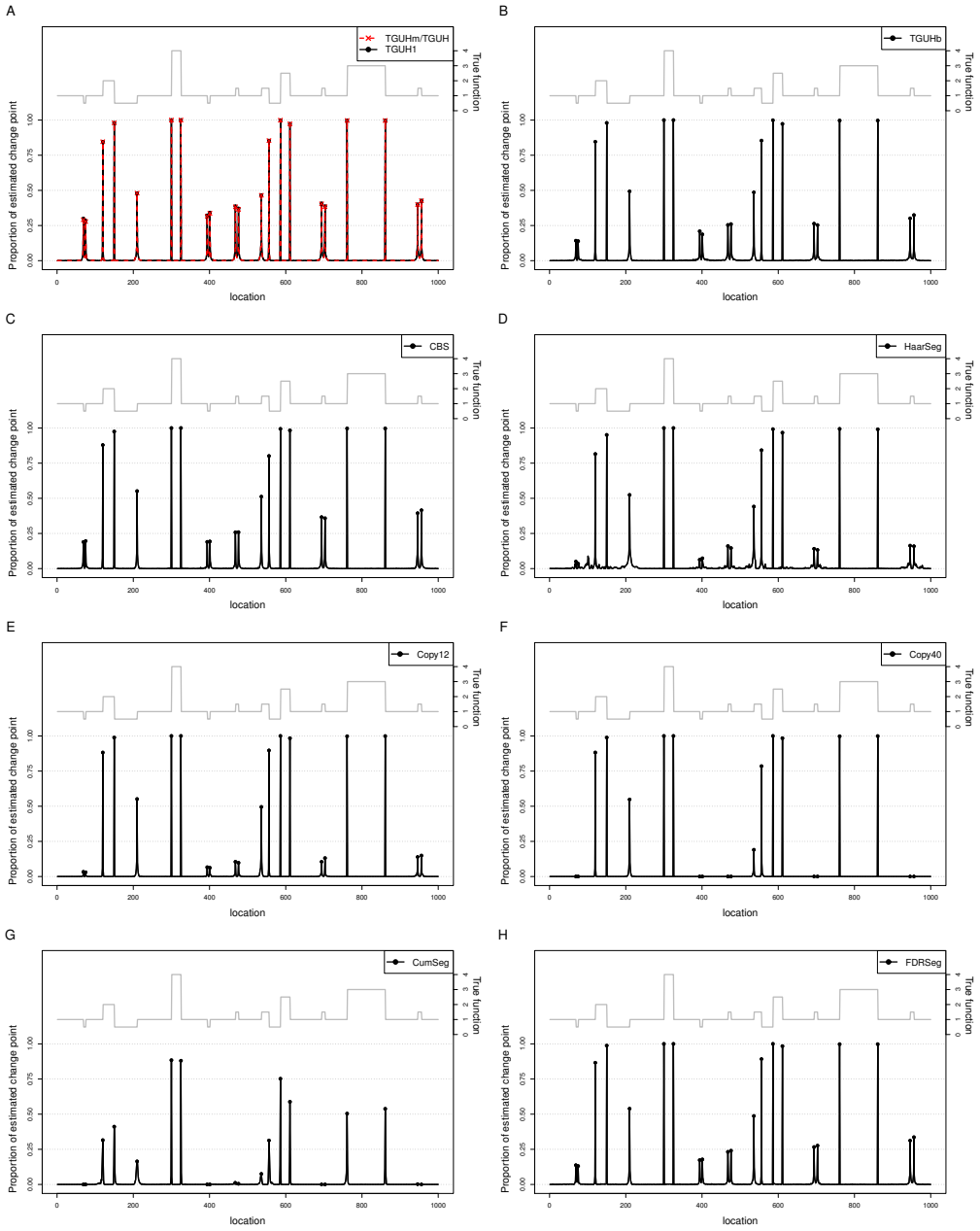


Figure B.3: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.3^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **A** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

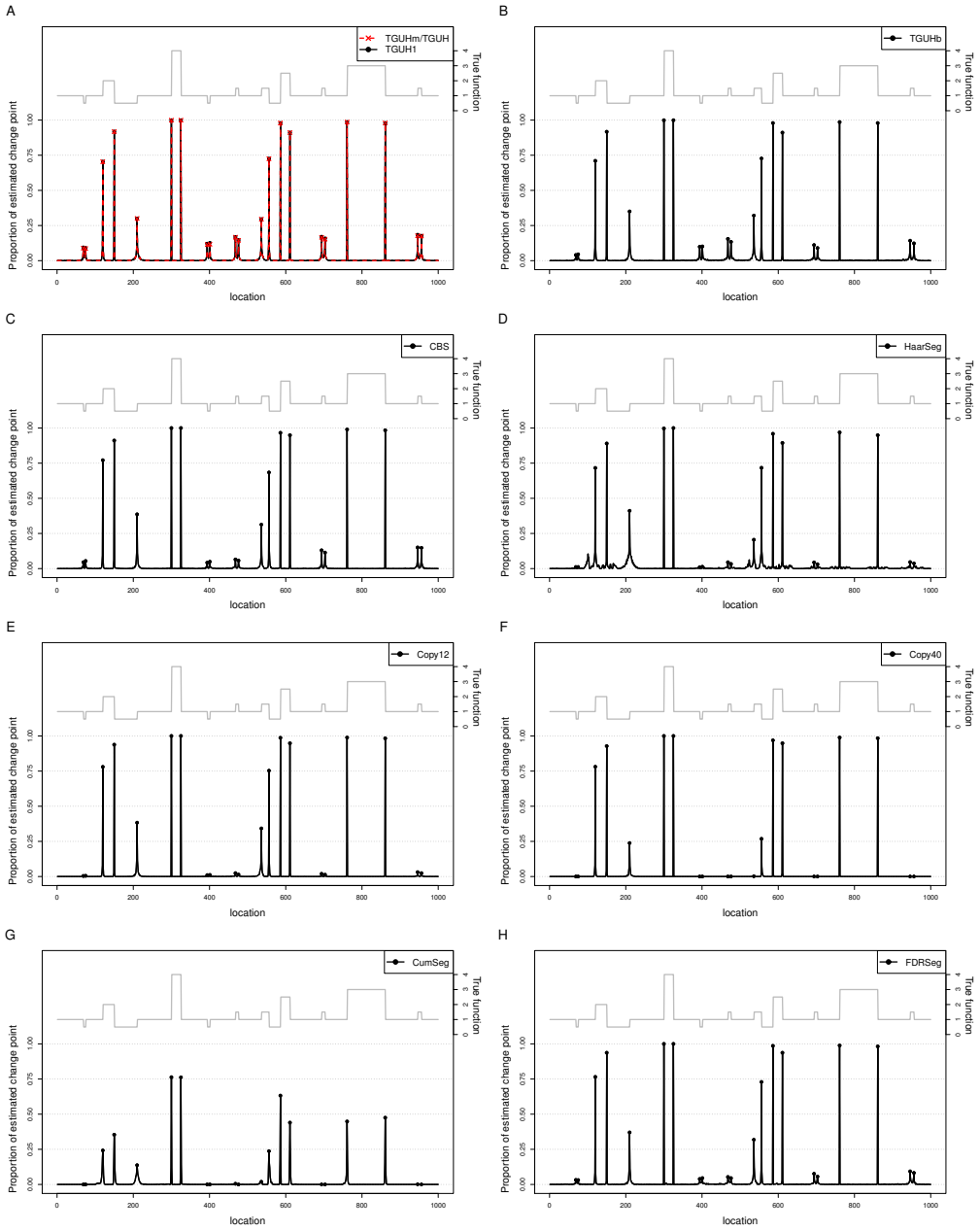


Figure B.4: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.4^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **A** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

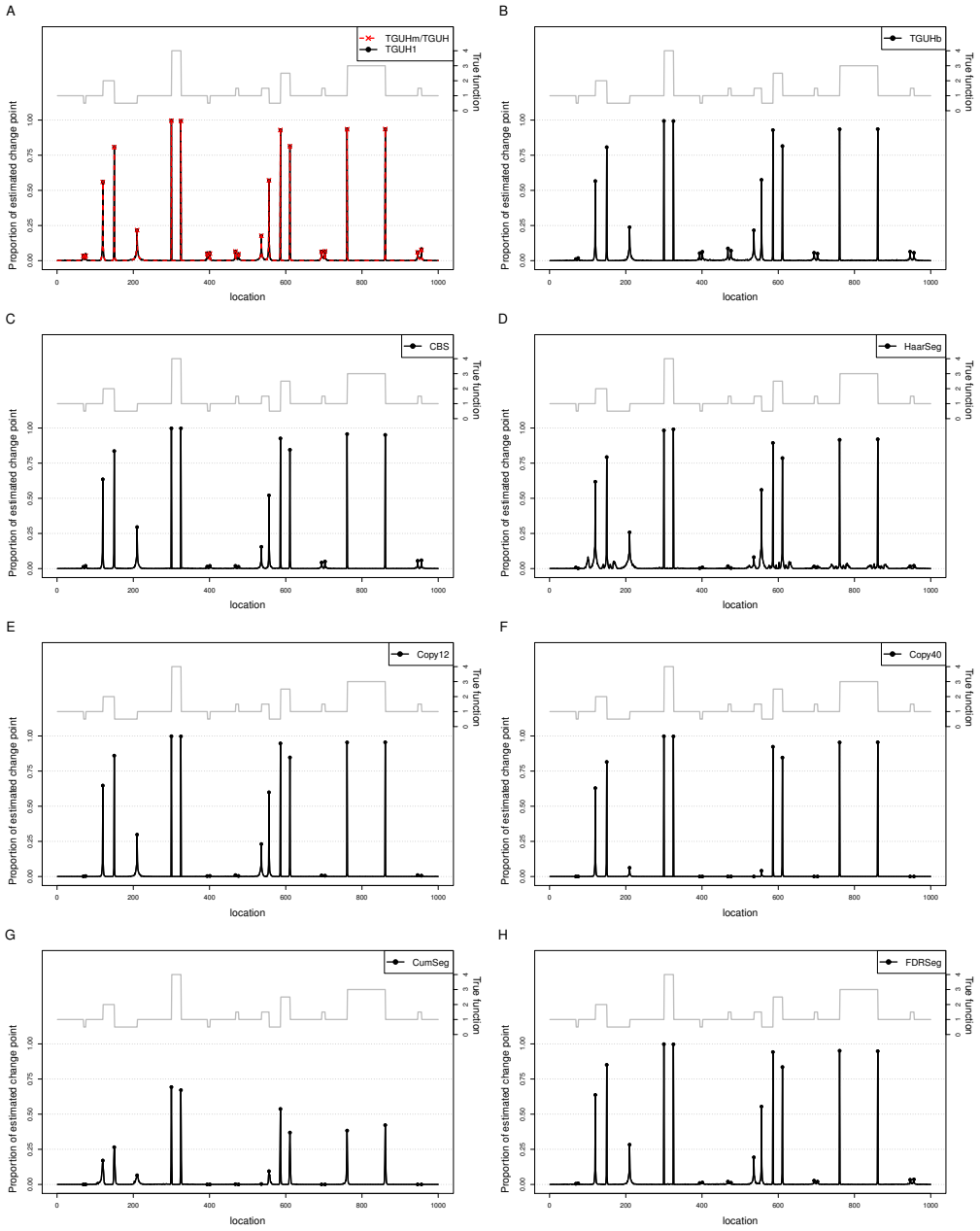


Figure B.5: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.5^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **A** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

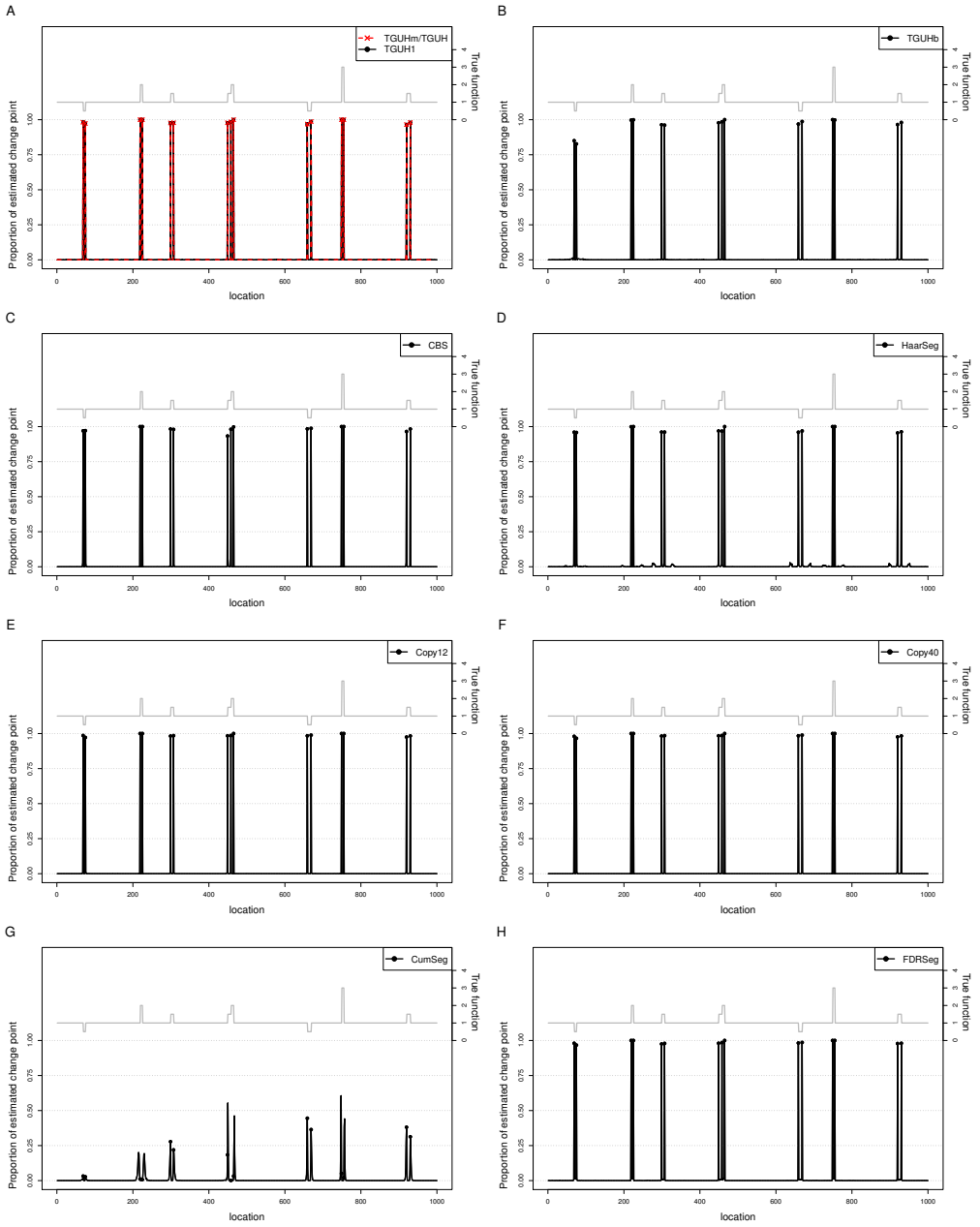


Figure B.6: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.1^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **B** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

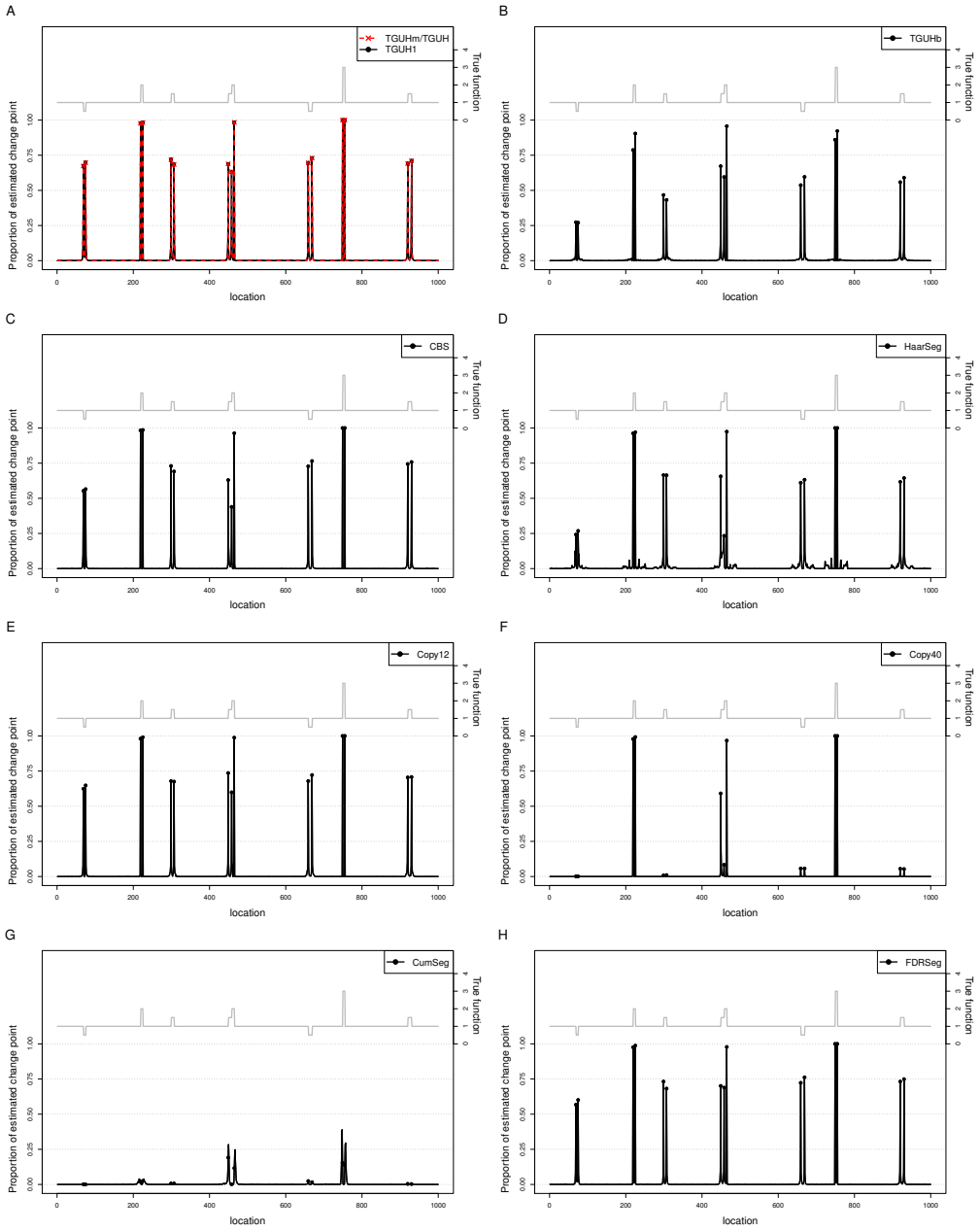


Figure B.7: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.2^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **B** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

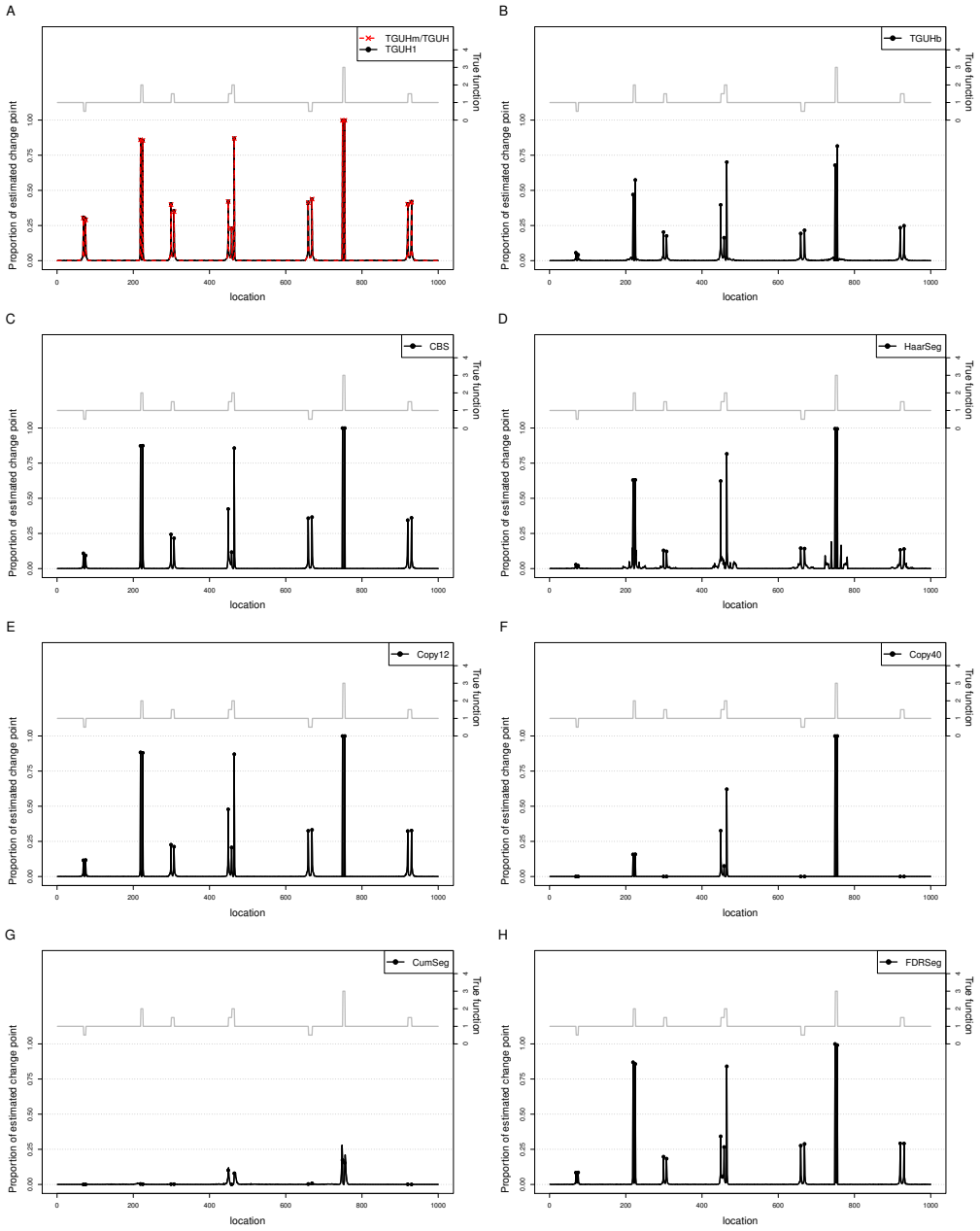


Figure B.8: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.3^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **B** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

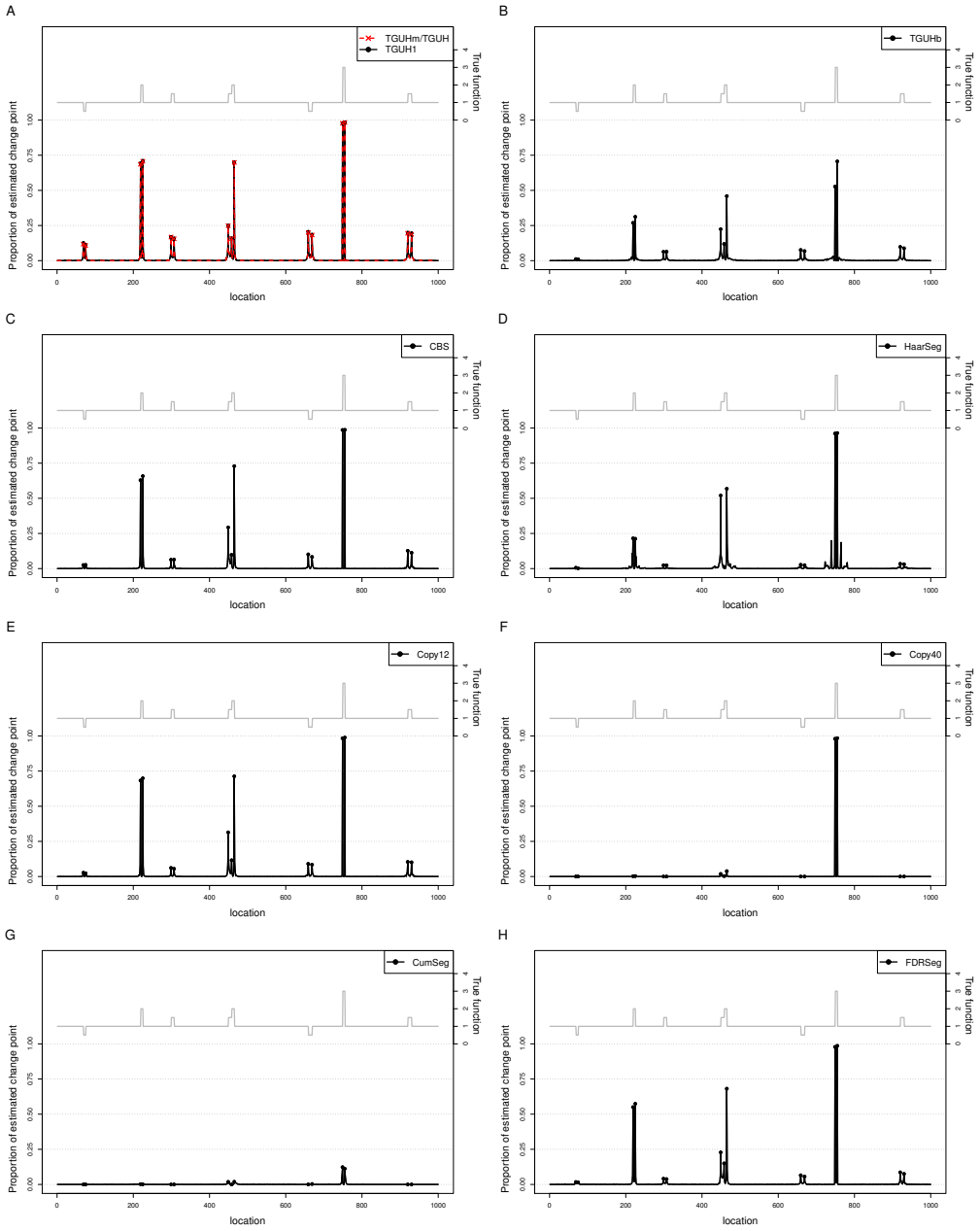


Figure B.9: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.4^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **B** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

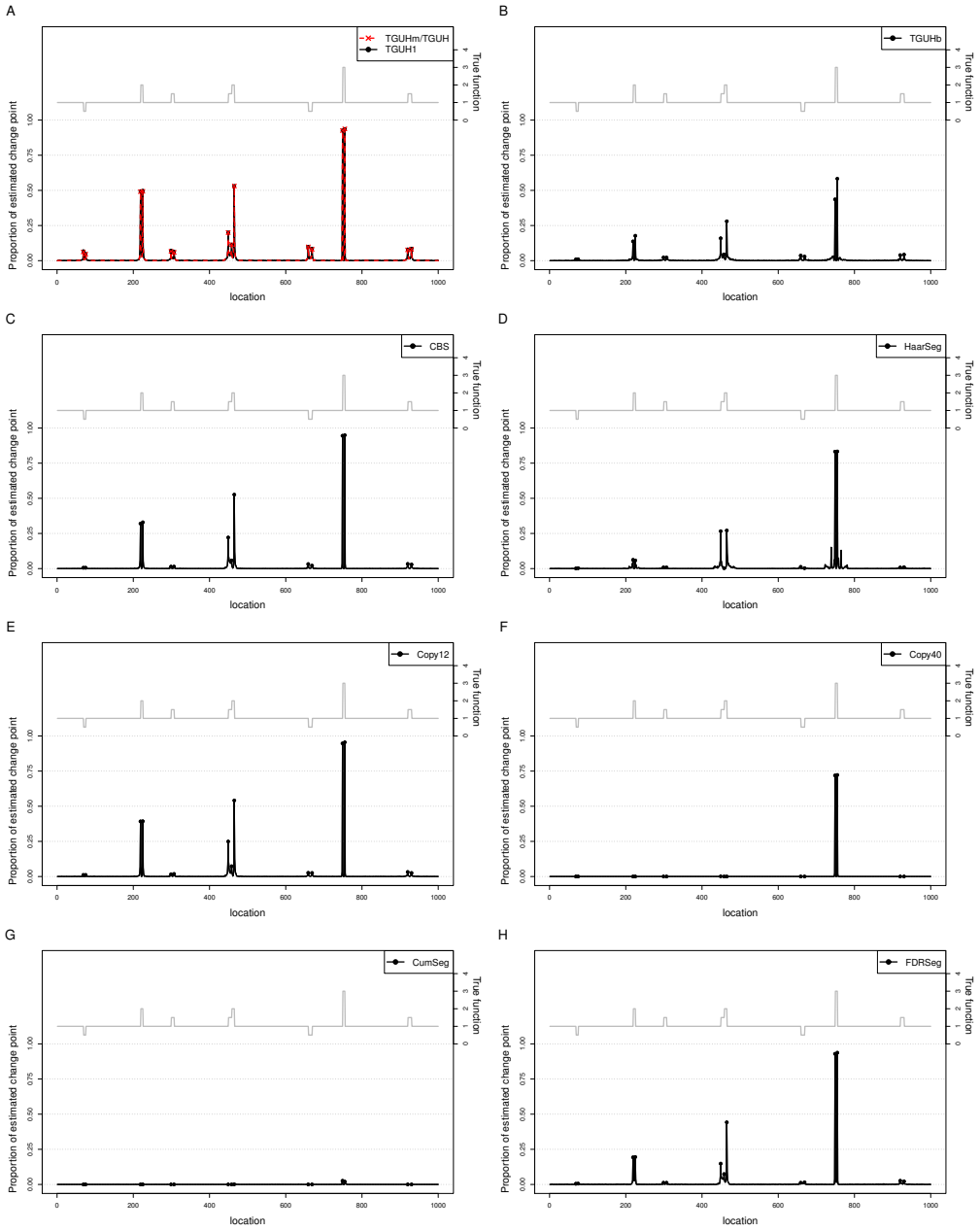


Figure B.10: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.5^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **B** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

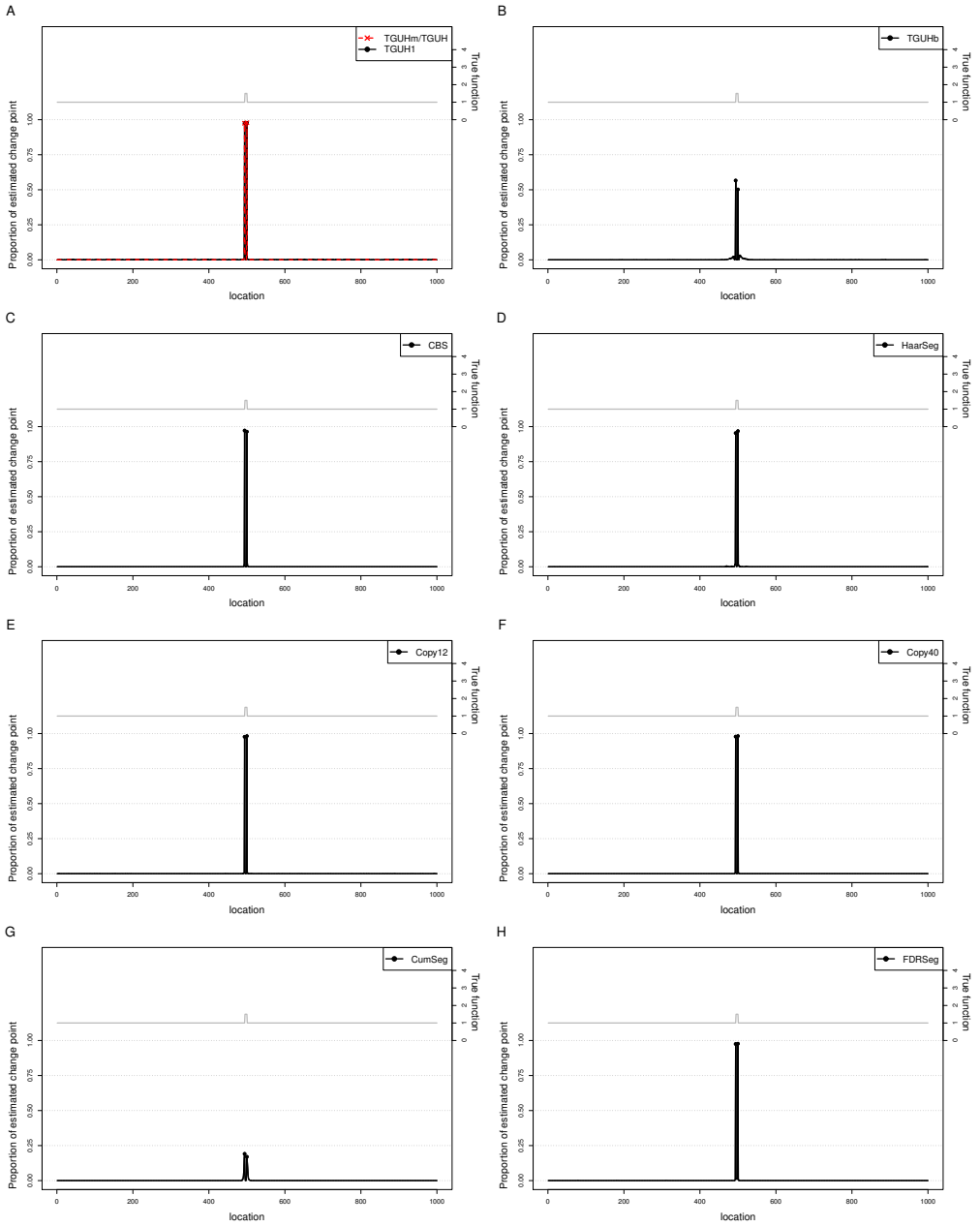


Figure B.11: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.1^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **C** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

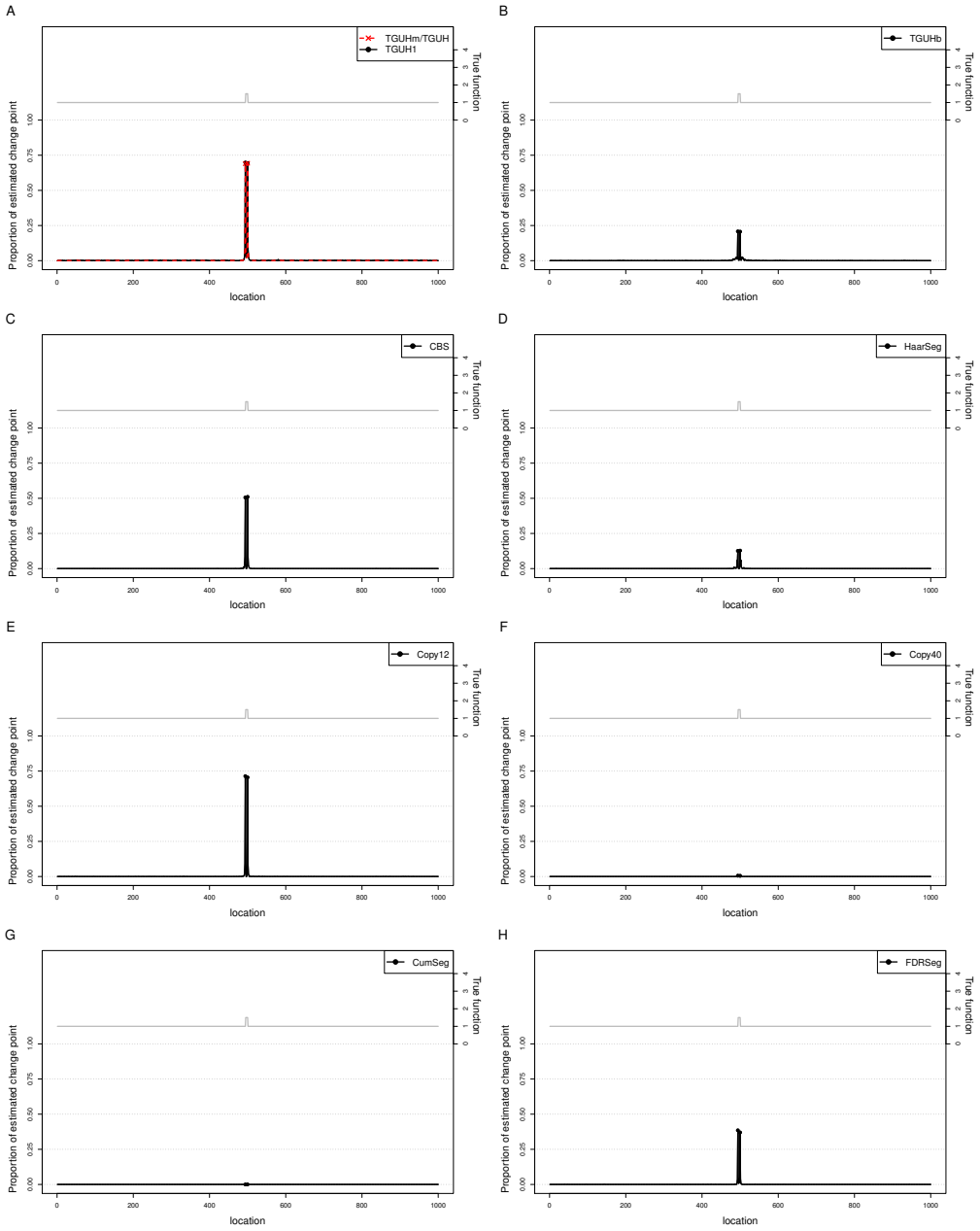


Figure B.12: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.2^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **C** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

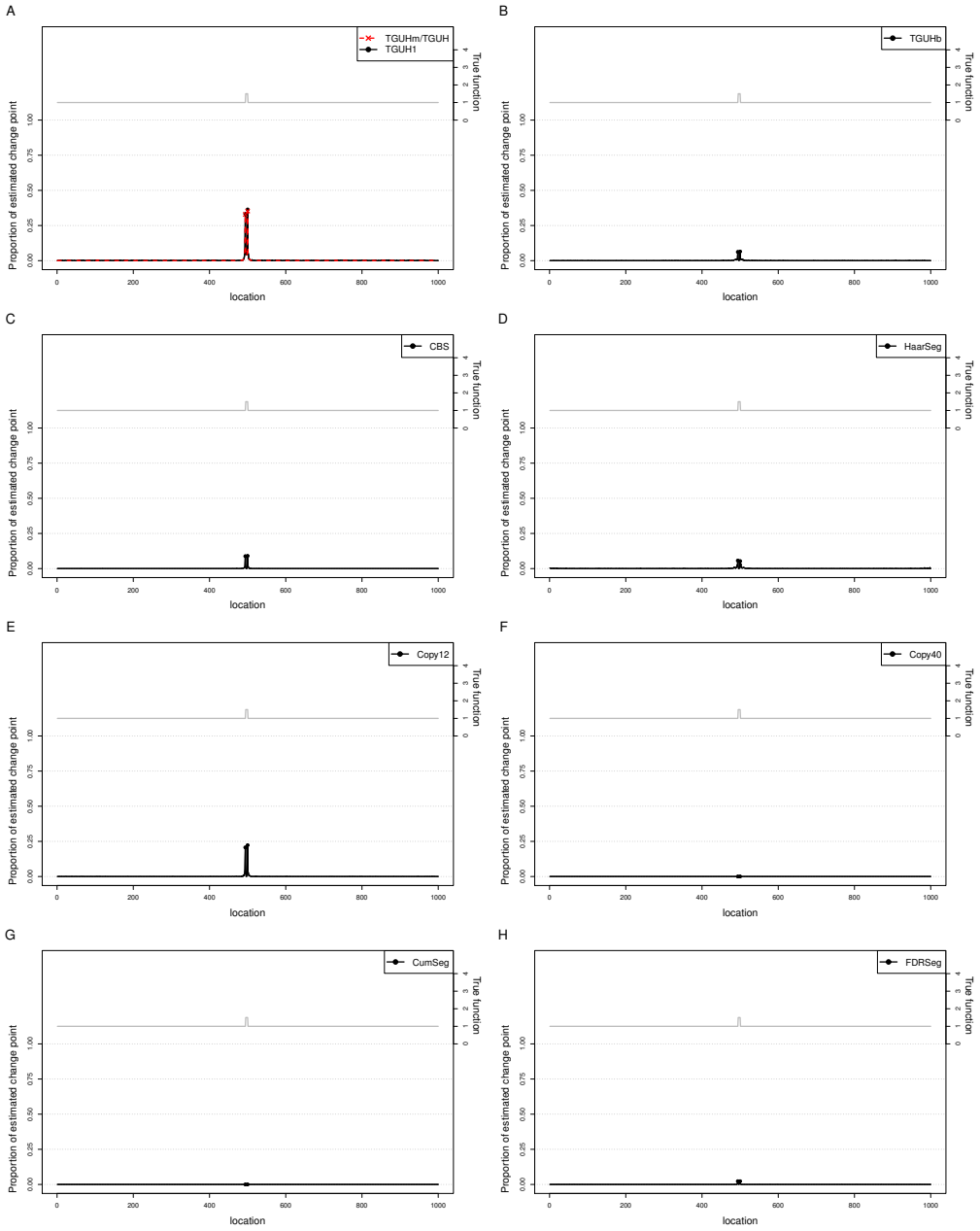


Figure B.13: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.3^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **C** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

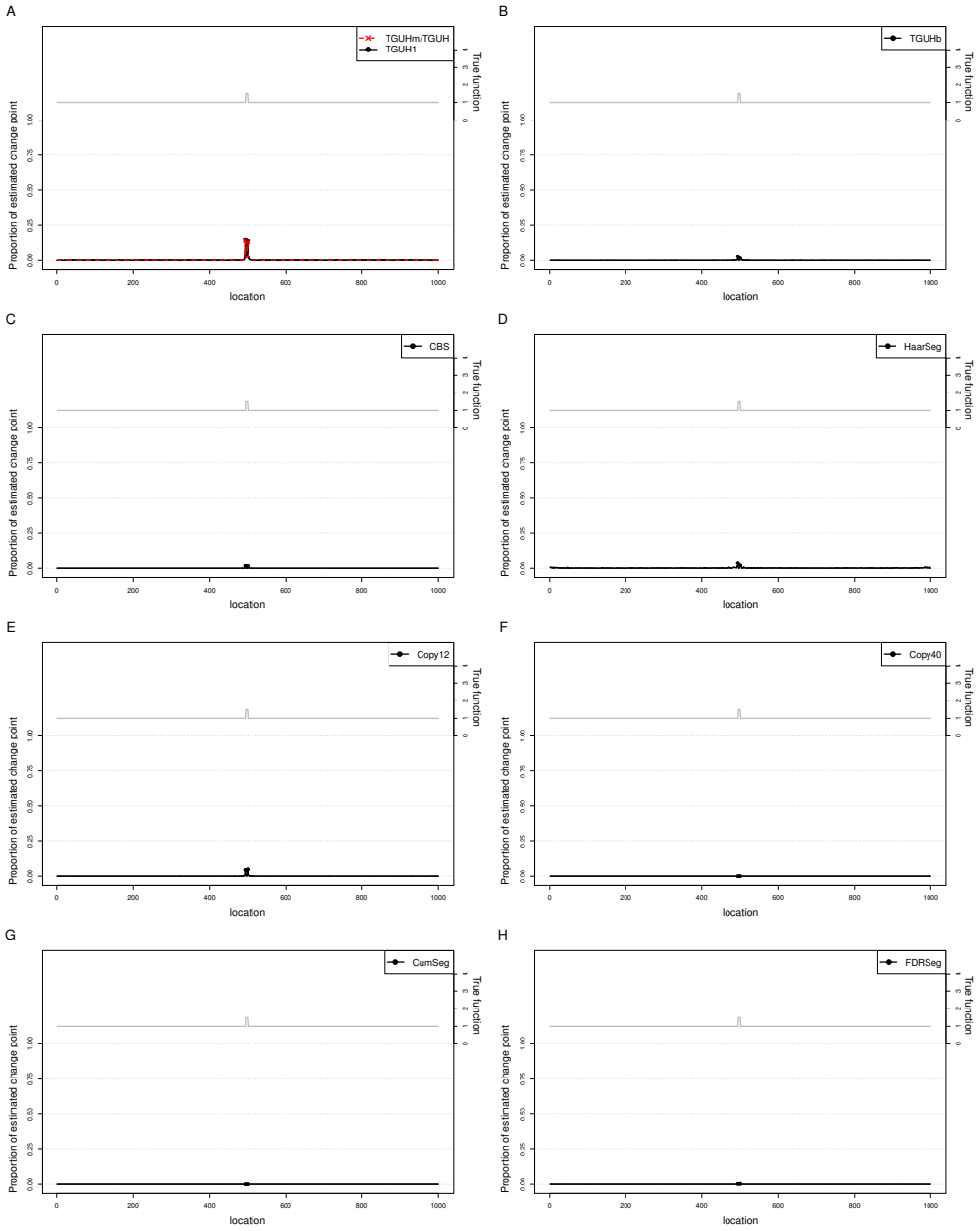


Figure B.14: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.4^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **C** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

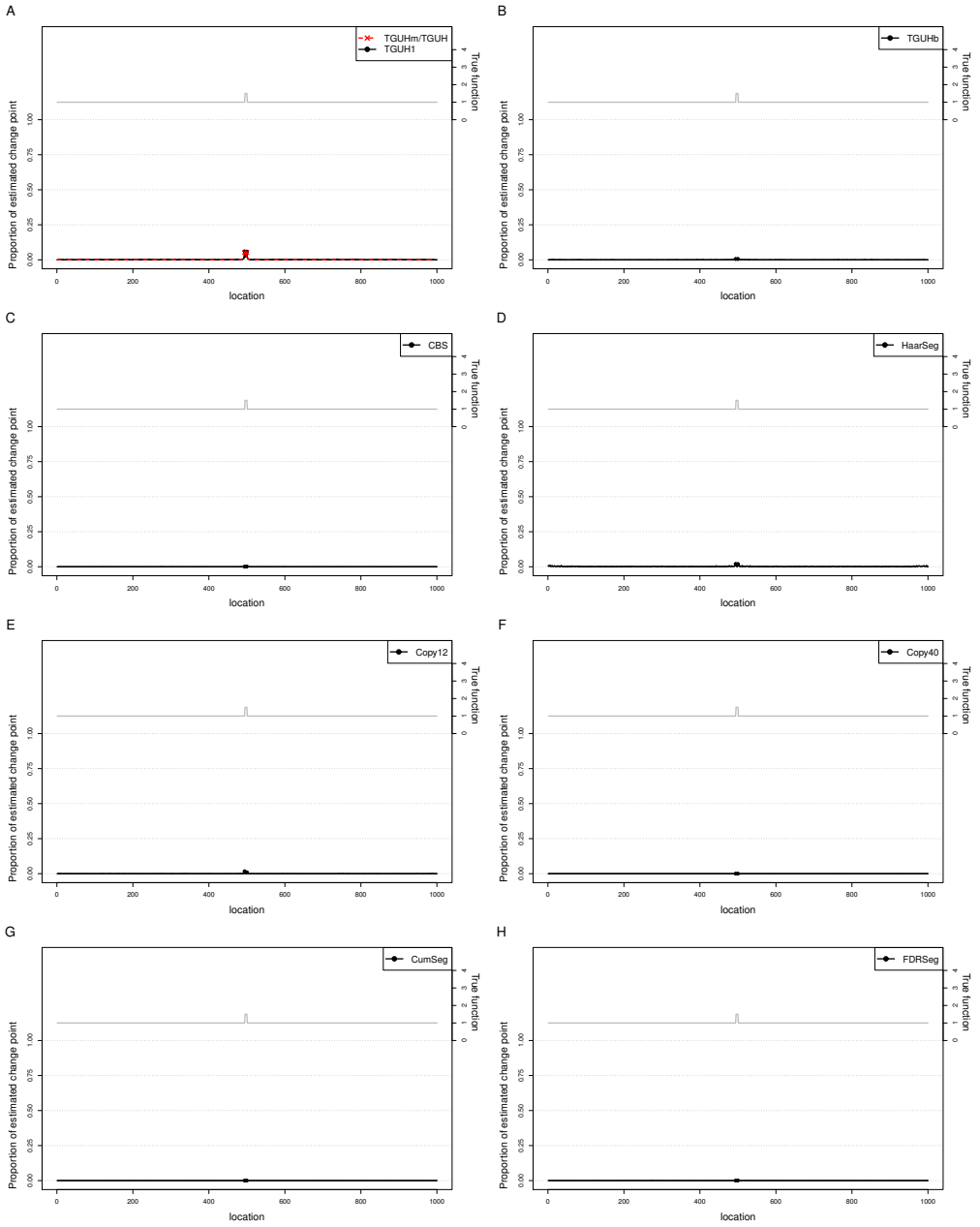


Figure B.15: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.5^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **C** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

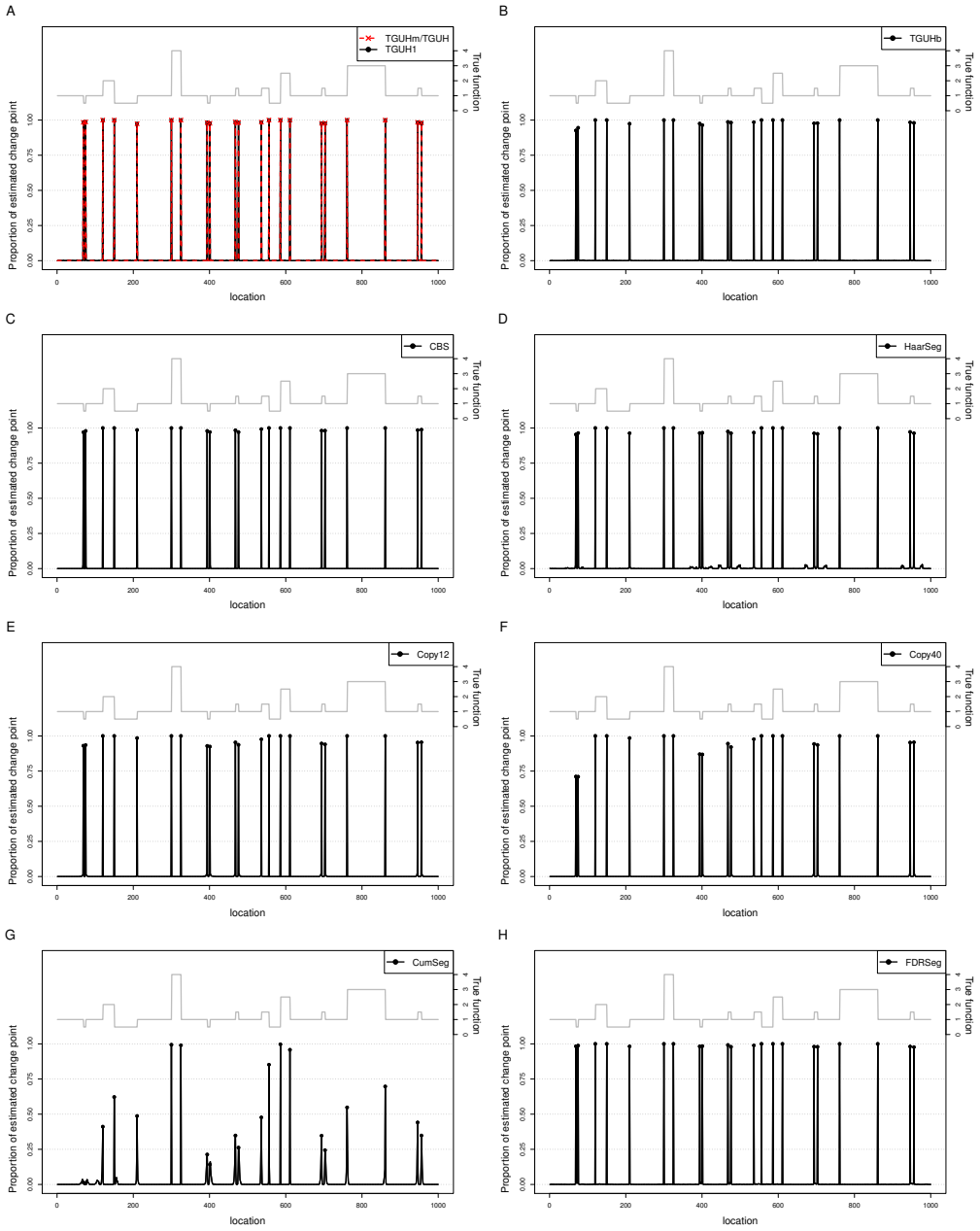


Figure B.16: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.1^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **A** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

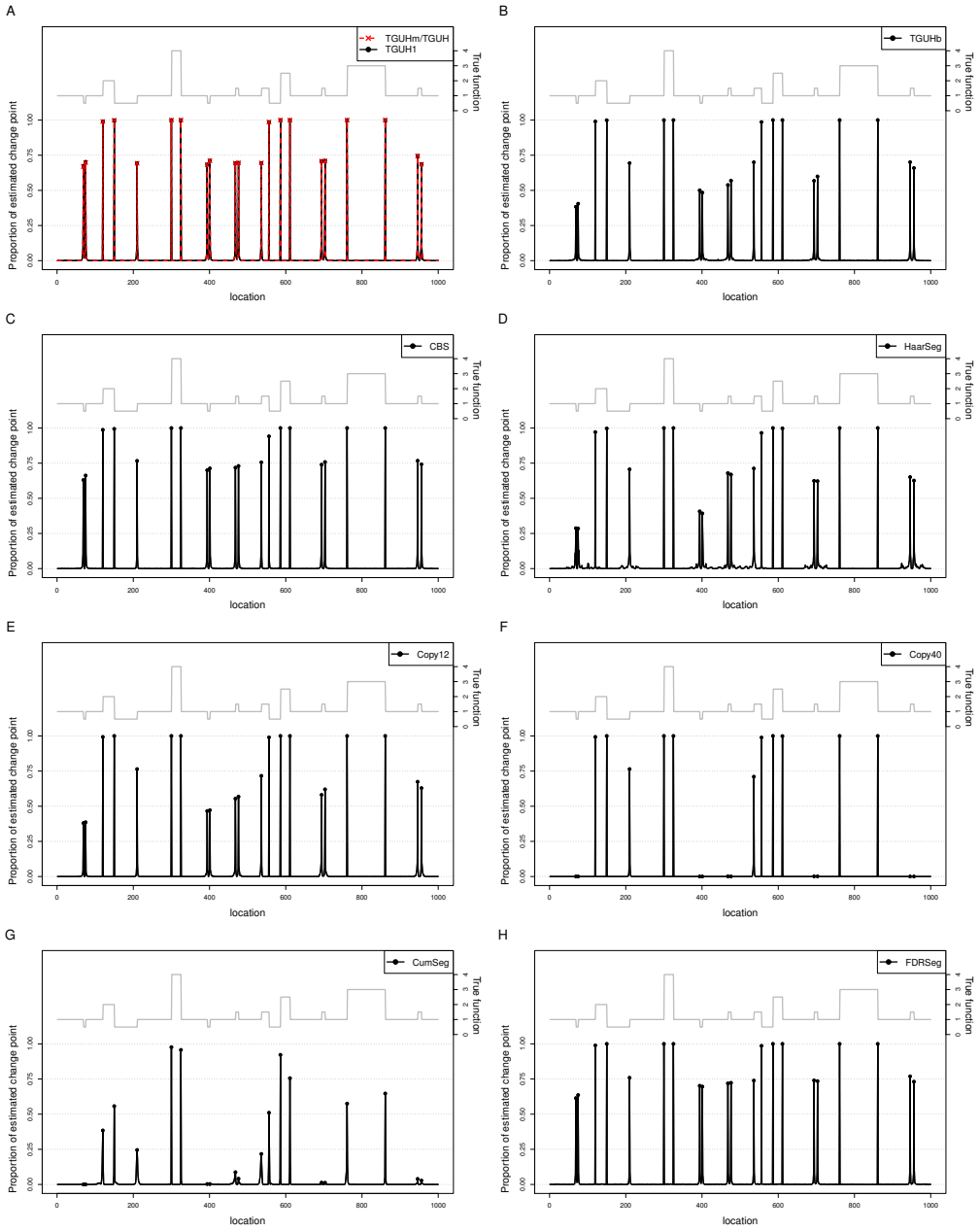


Figure B.17: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.2^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **A** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

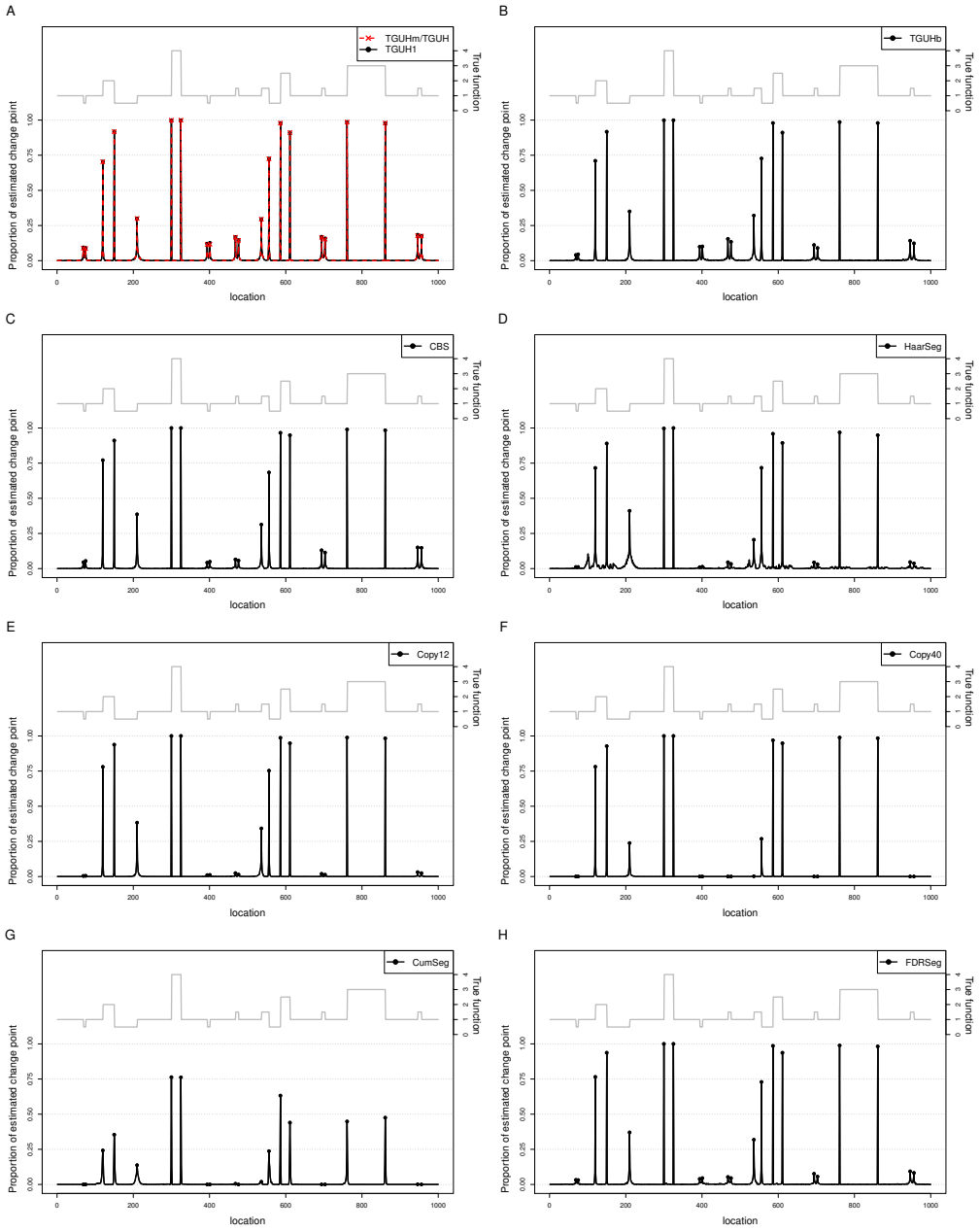


Figure B.18: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.4^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **A** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

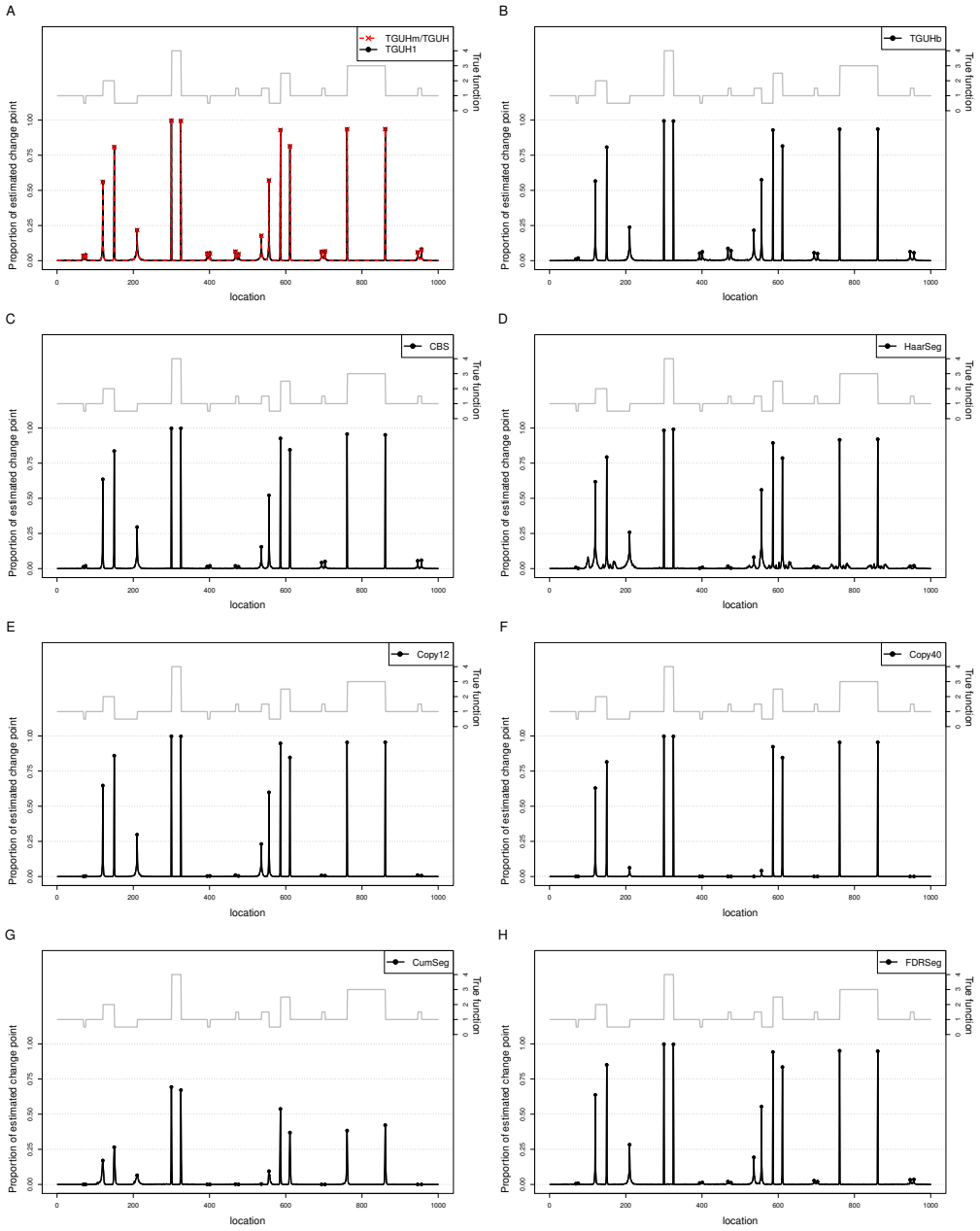


Figure B.19: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.5^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **A** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

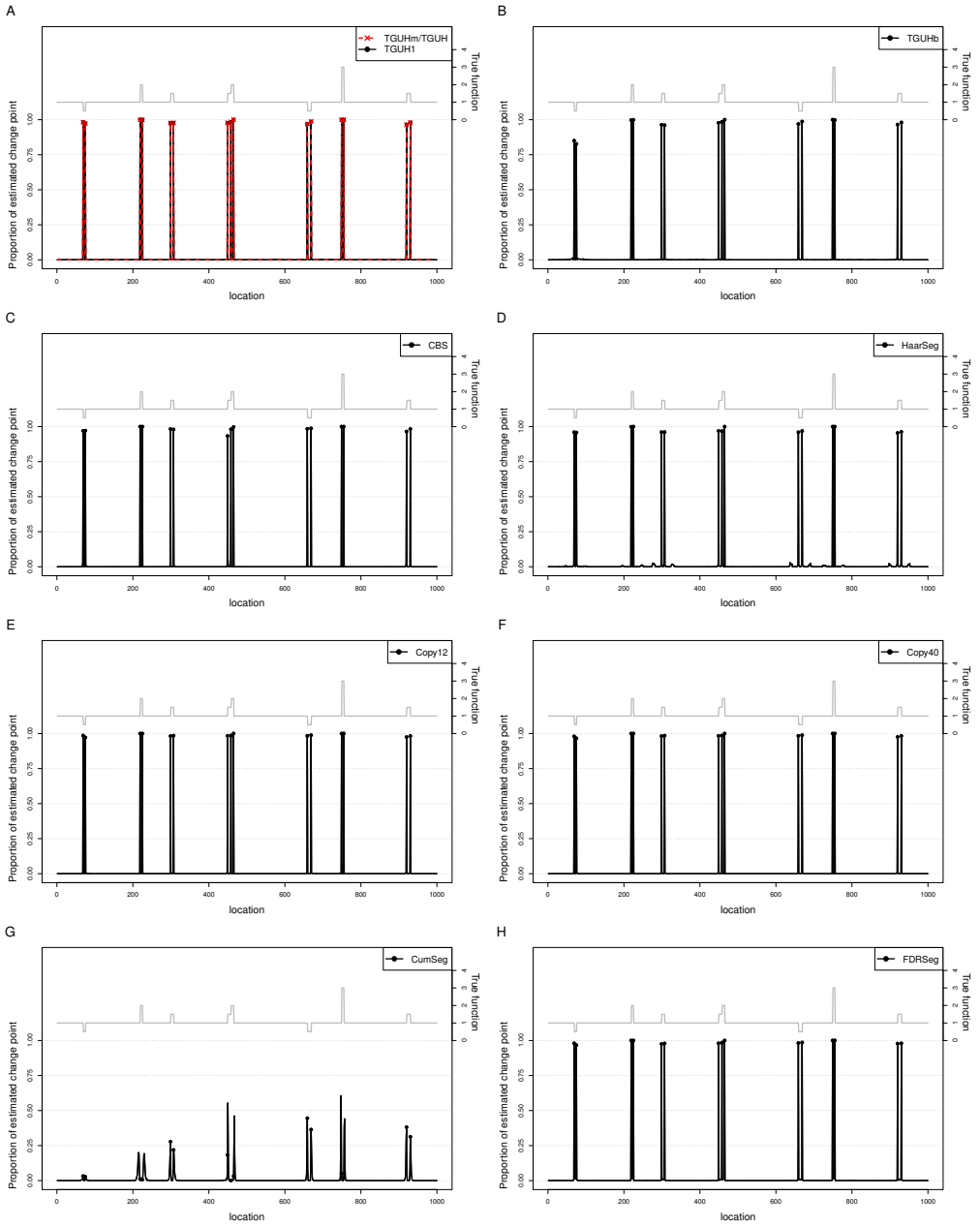


Figure B.20: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.1^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **B** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

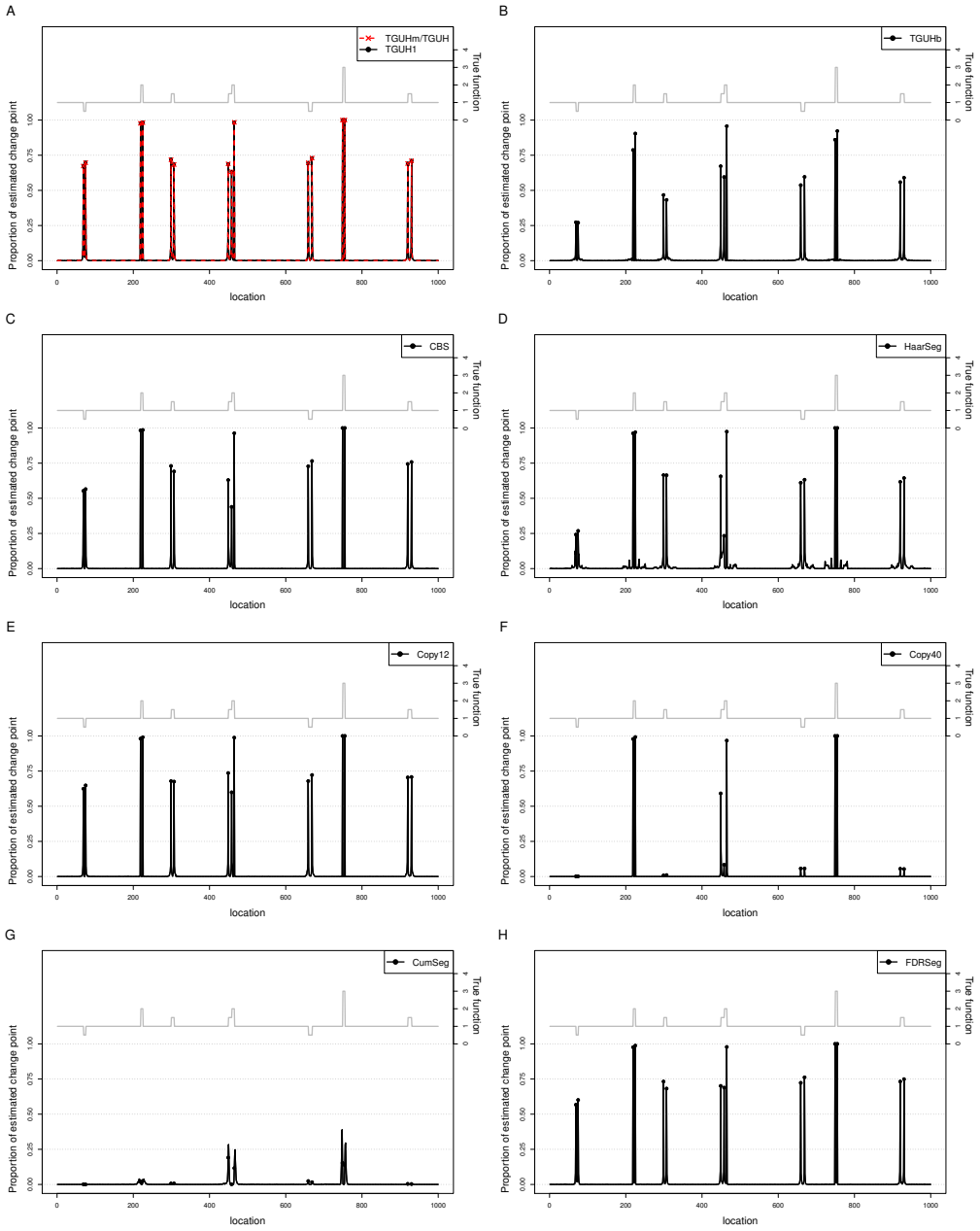


Figure B.21: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.2^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **B** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

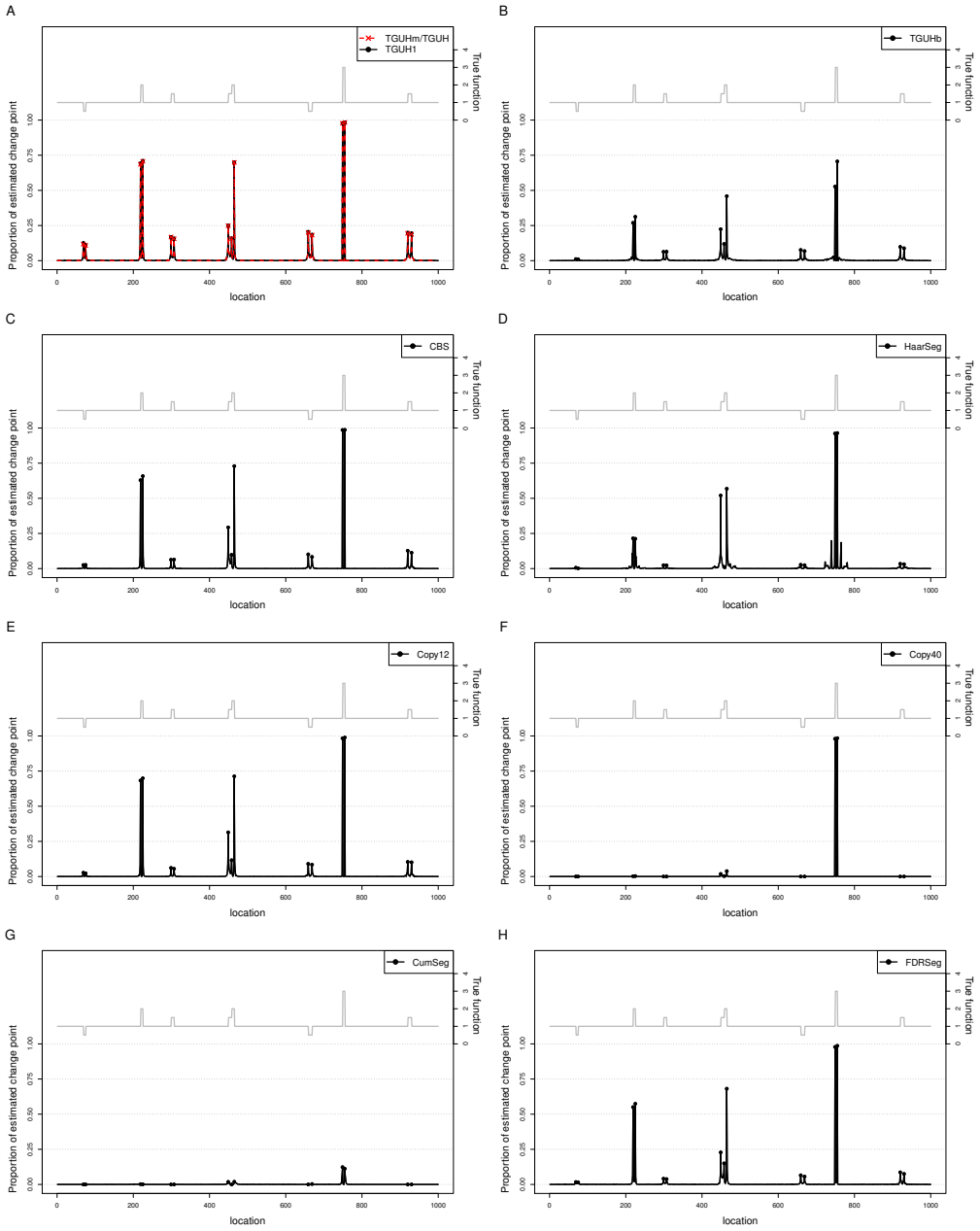


Figure B.22: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.4^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **B** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

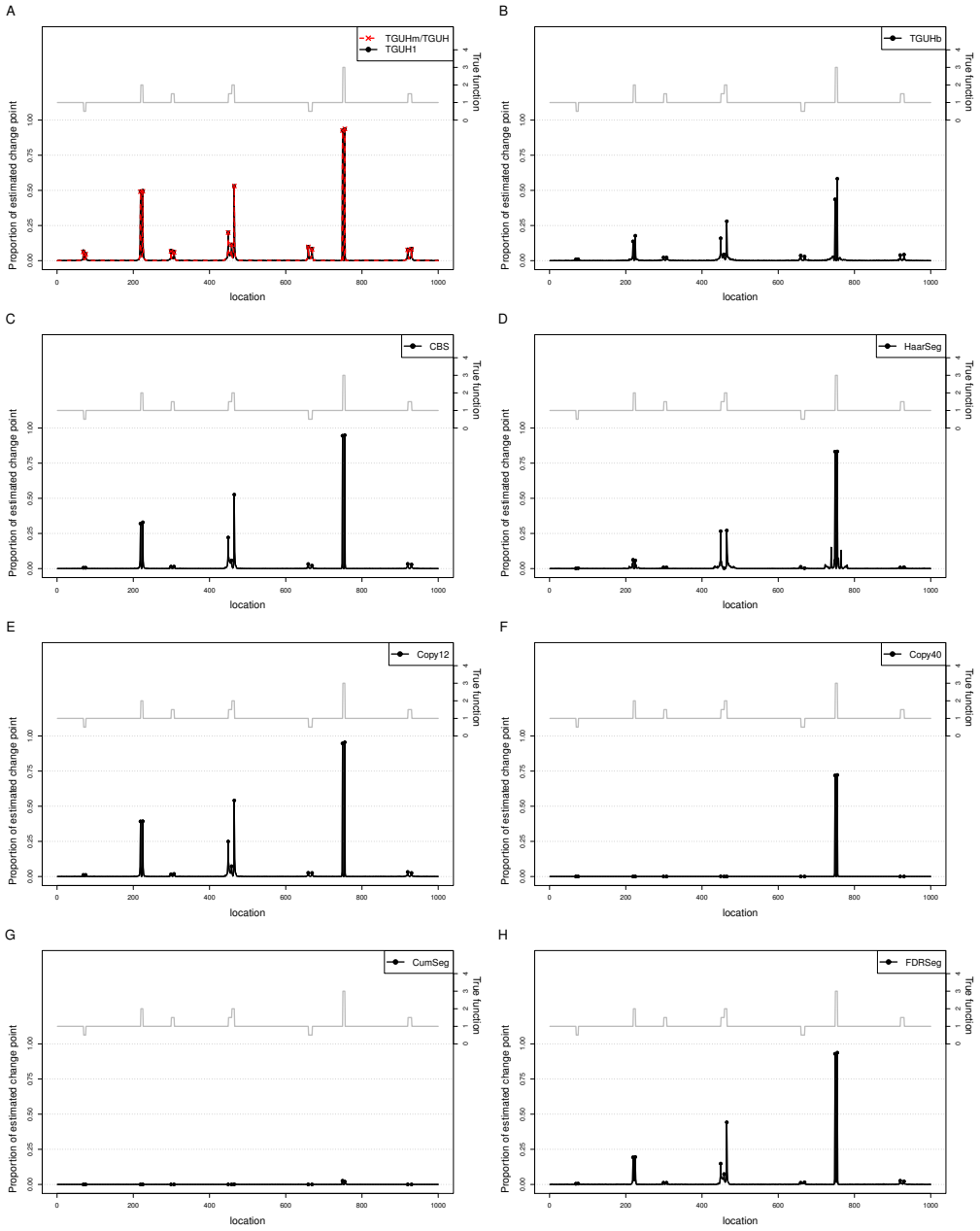


Figure B.23: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.5^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **B** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

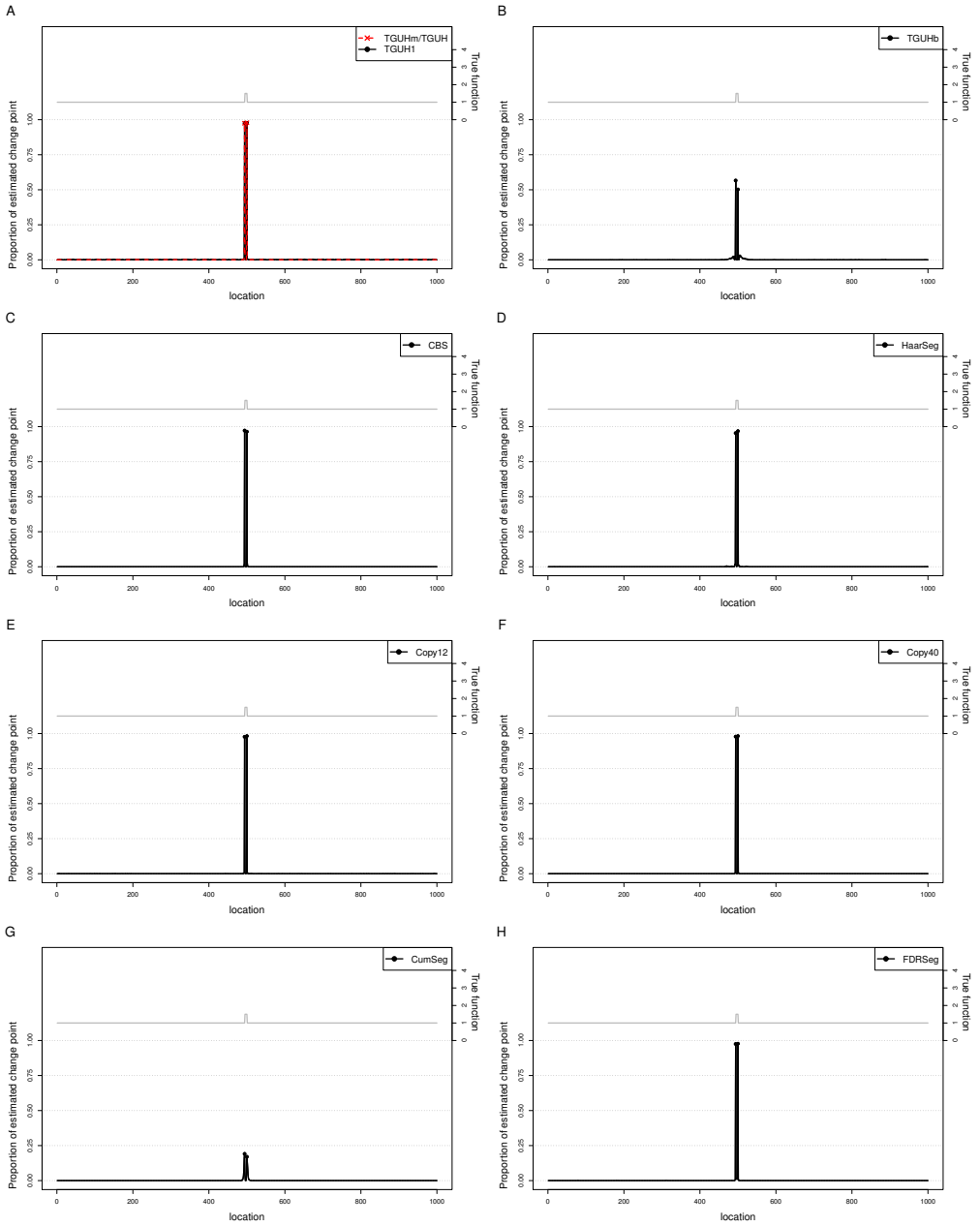


Figure B.24: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.1^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **C** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

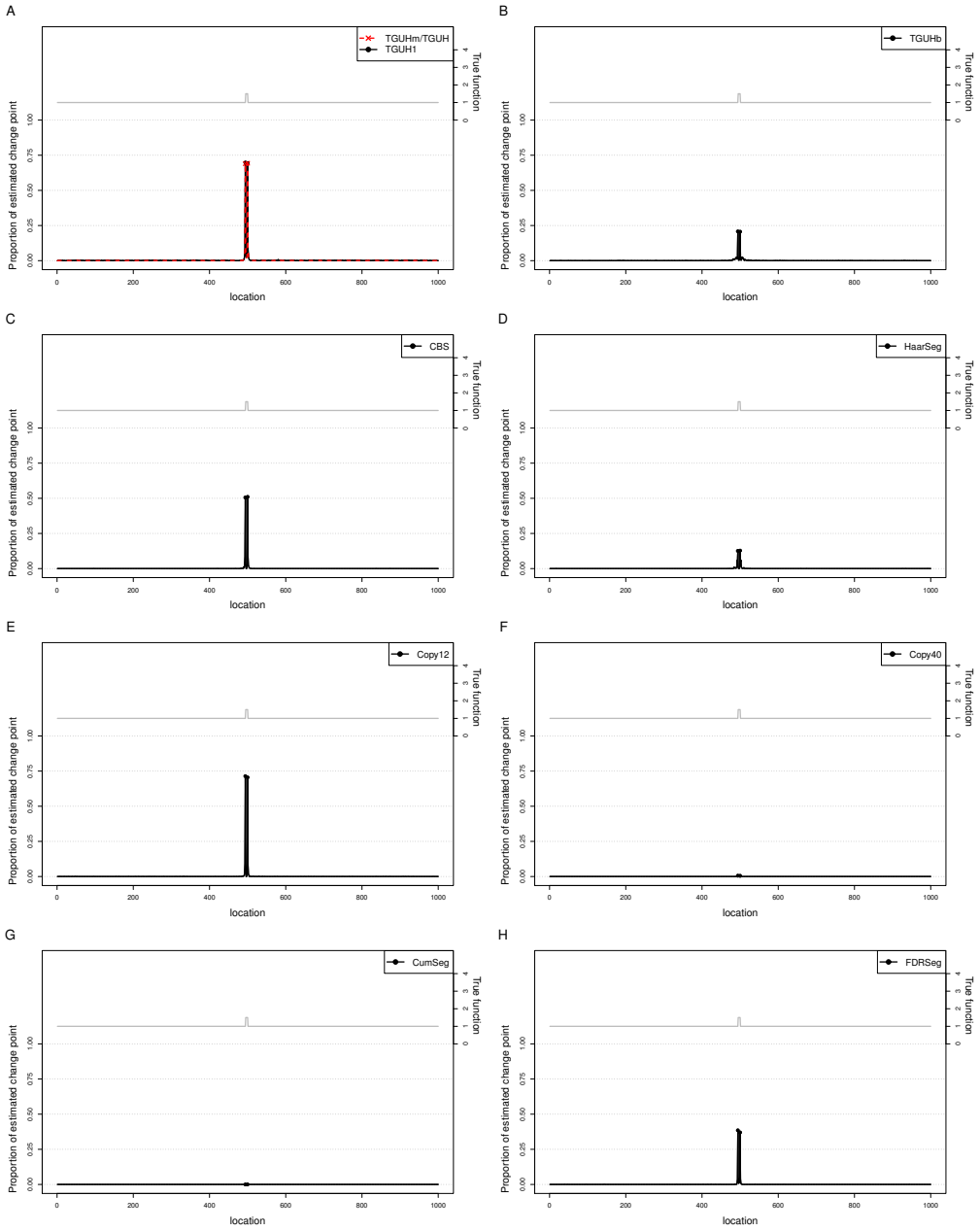


Figure B.25: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.2^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **C** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

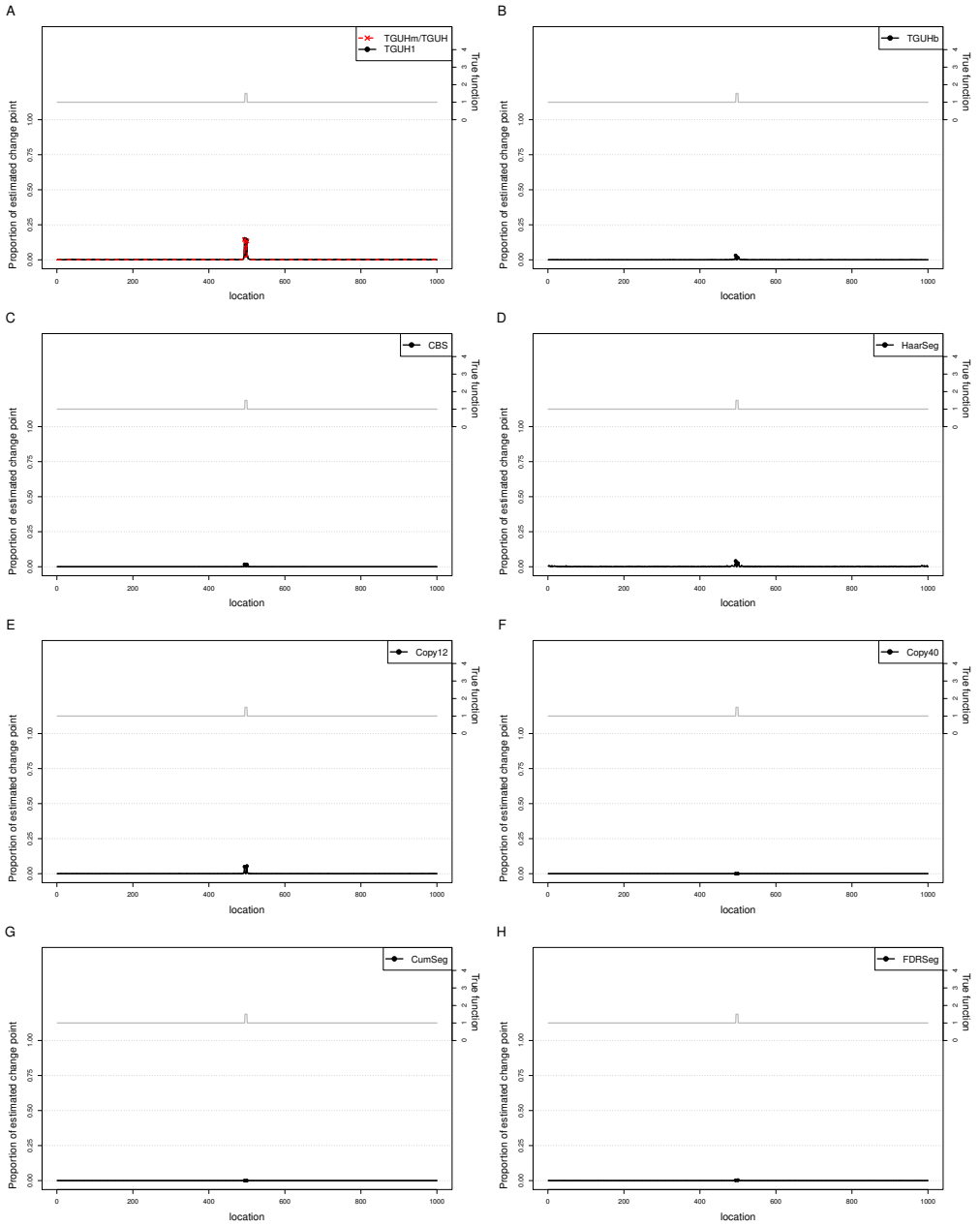


Figure B.26: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.4^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **C** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

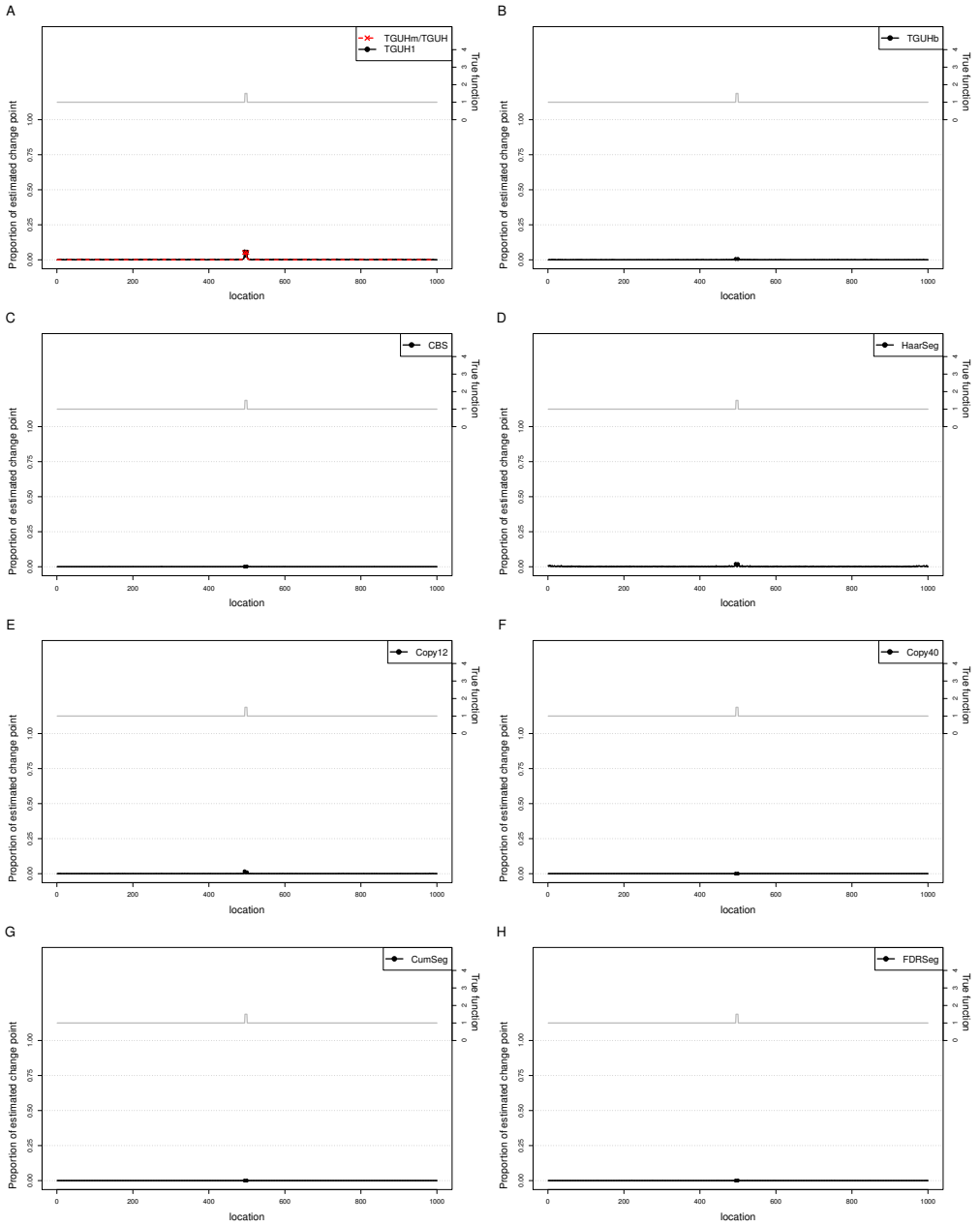


Figure B.27: Proportion of times a change-point is estimated against location out of 1000 simulated datasets contaminated with a mixture of two Gaussian distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$ for $\sigma^2 = 0.5^2$. The dots denote proportion of detection at locations where there are actual change-points. The grey solid line is the corresponding test function, repeated here from panel **C** of Figure 5.12 for ease of reference. The left and right vertical axis shows the proportion of replicates where a change-point is estimated and the corresponding test function's height, respectively.

# Appendix C

# Additional Figures of Section 5.6

## C.1 Additional figures of the proportion of times change-points estimated at each location: noise model 1

Figures C.1–C.12 show the proportion of times (from 1000 simulated datasets) that each method detects a change-point at each location along the sequence based on 1000 simulated datasets contaminated with Gaussian noise with mean zero and variance $\sigma^2$ where $\sigma_i^2 = \sigma_0^2 f_i^2$.

## C.2 Additional figures of the proportion of times change-points estimated at each location: noise model 2

Figures C.13–C.20 show the proportion of times (from 1000 simulated datasets) that each method detects a change-point at each location along the sequence based on 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2))$, where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.1, 0.3, 0.4$, and 0.5.

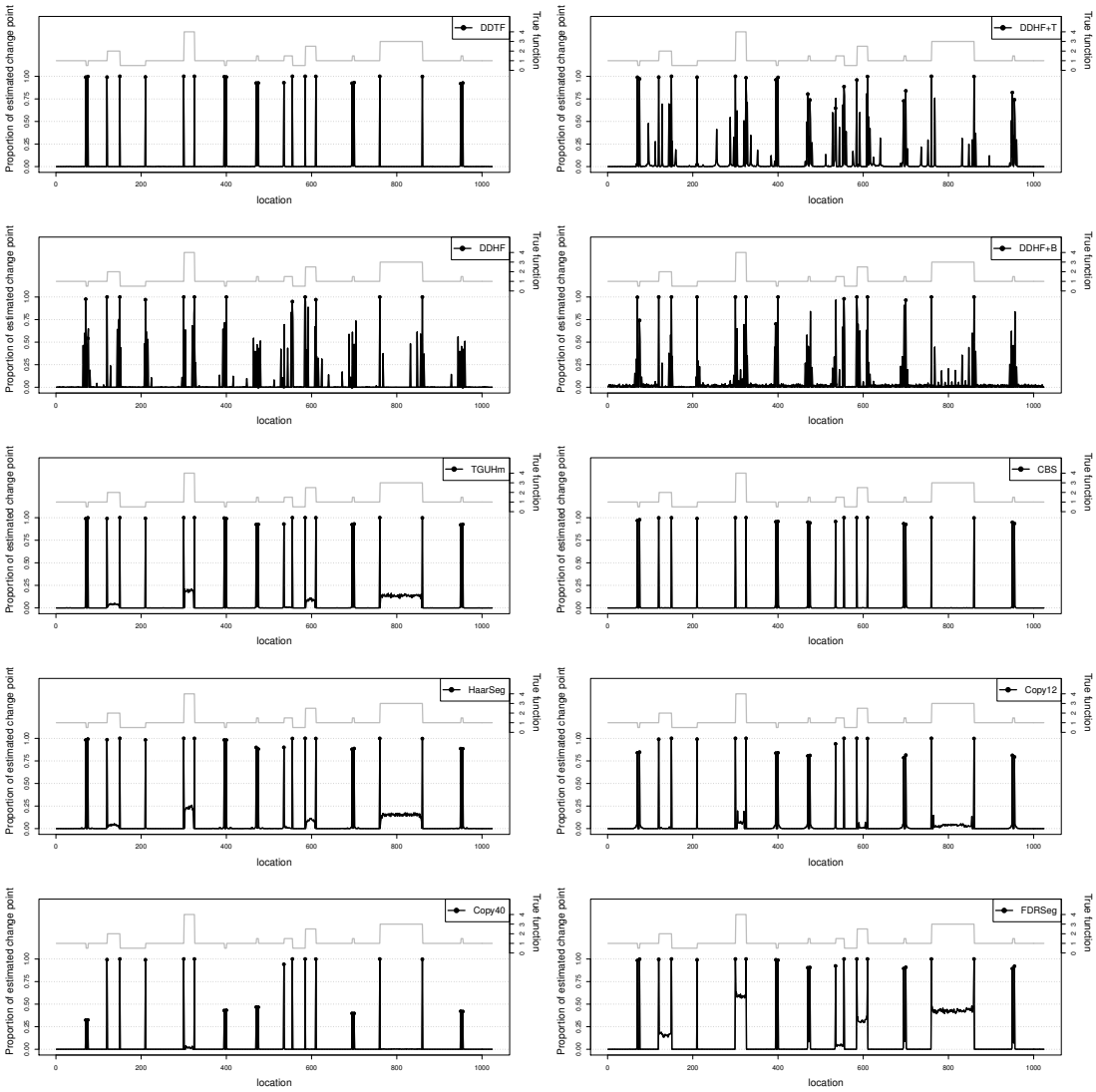


Figure C.1: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.1$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

## C.2 Additional figures of the proportion of times change-points estimated at each location: noise model 2

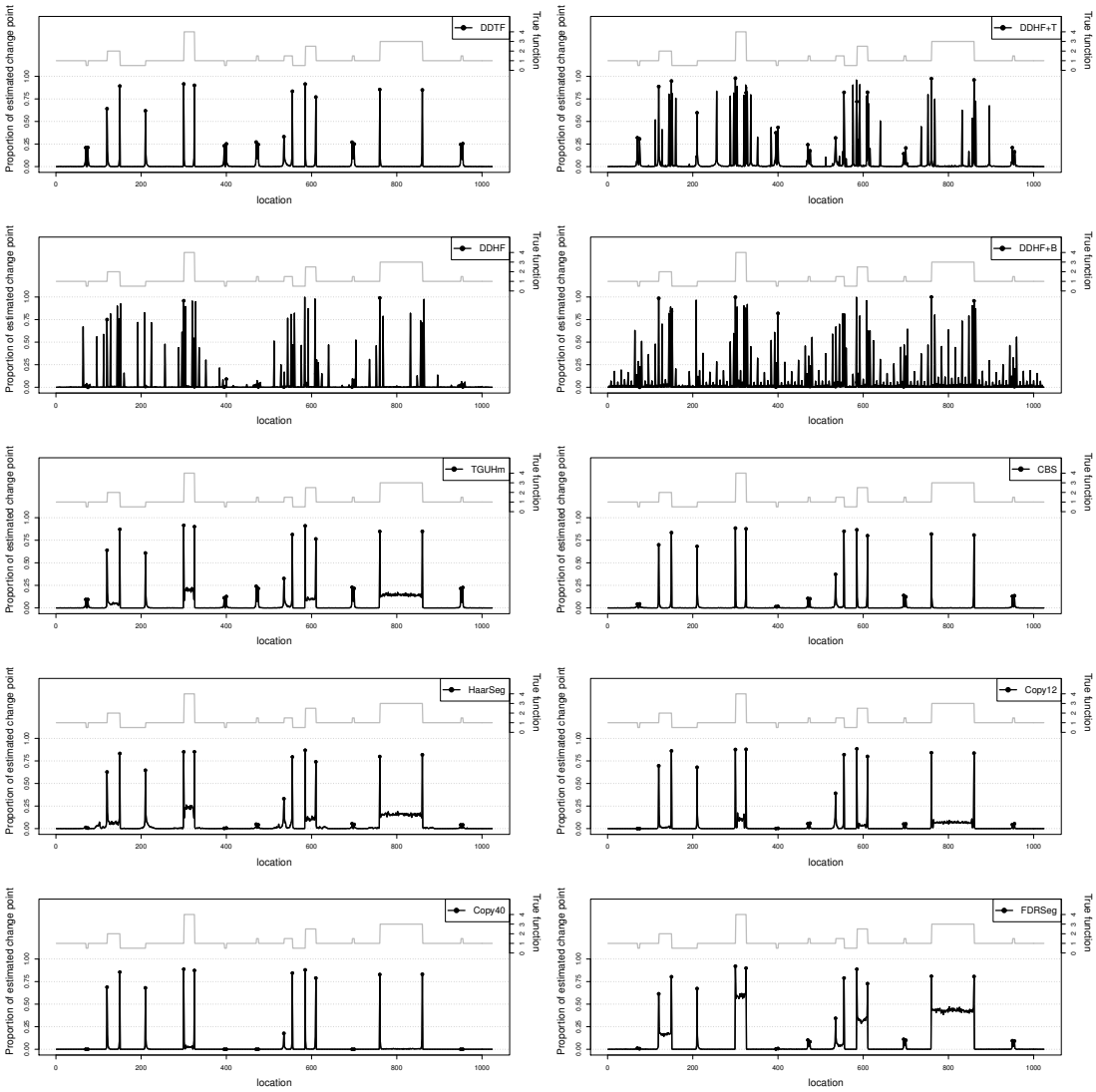


Figure C.2: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.3$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

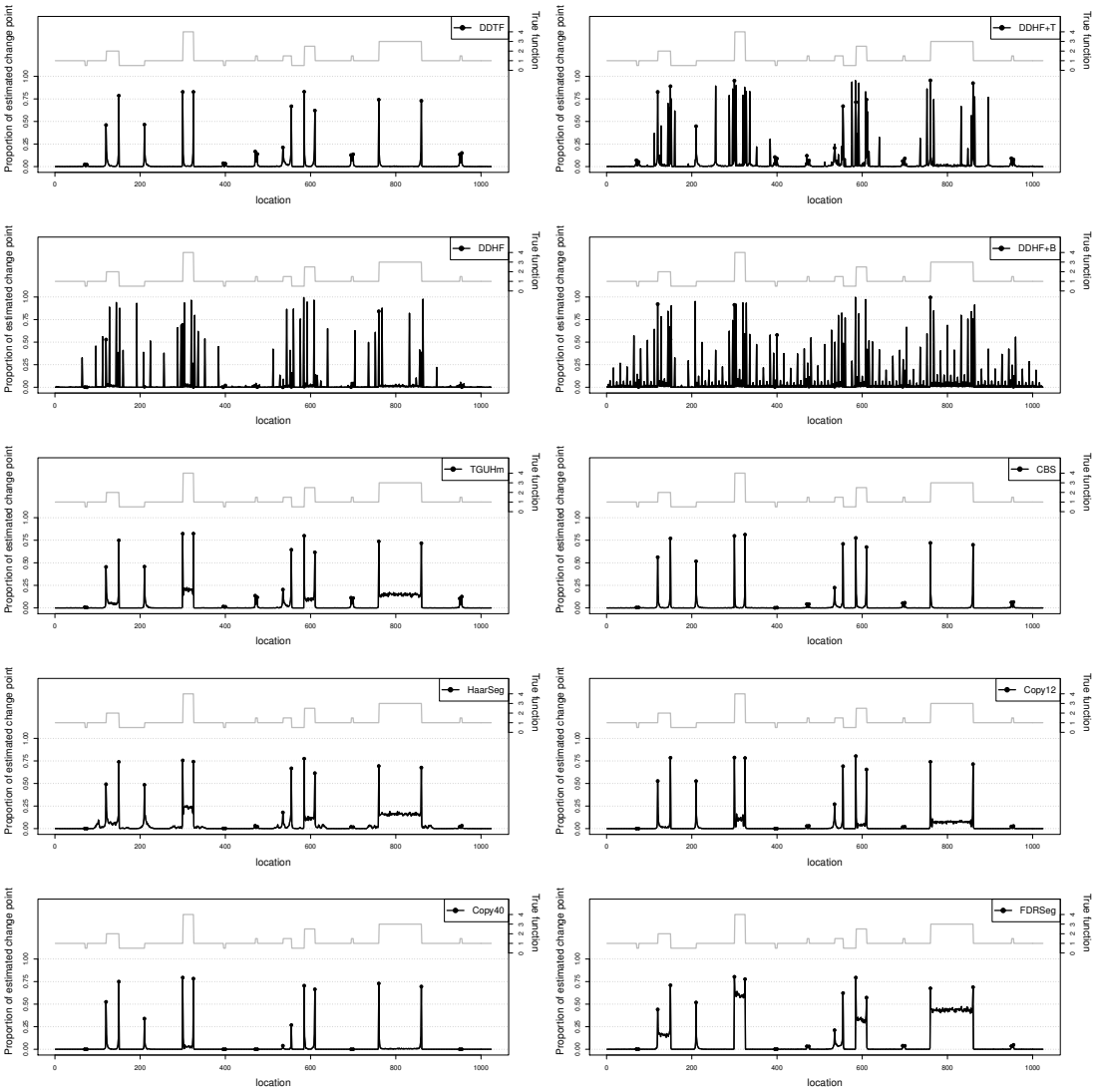


Figure C.3: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.4$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

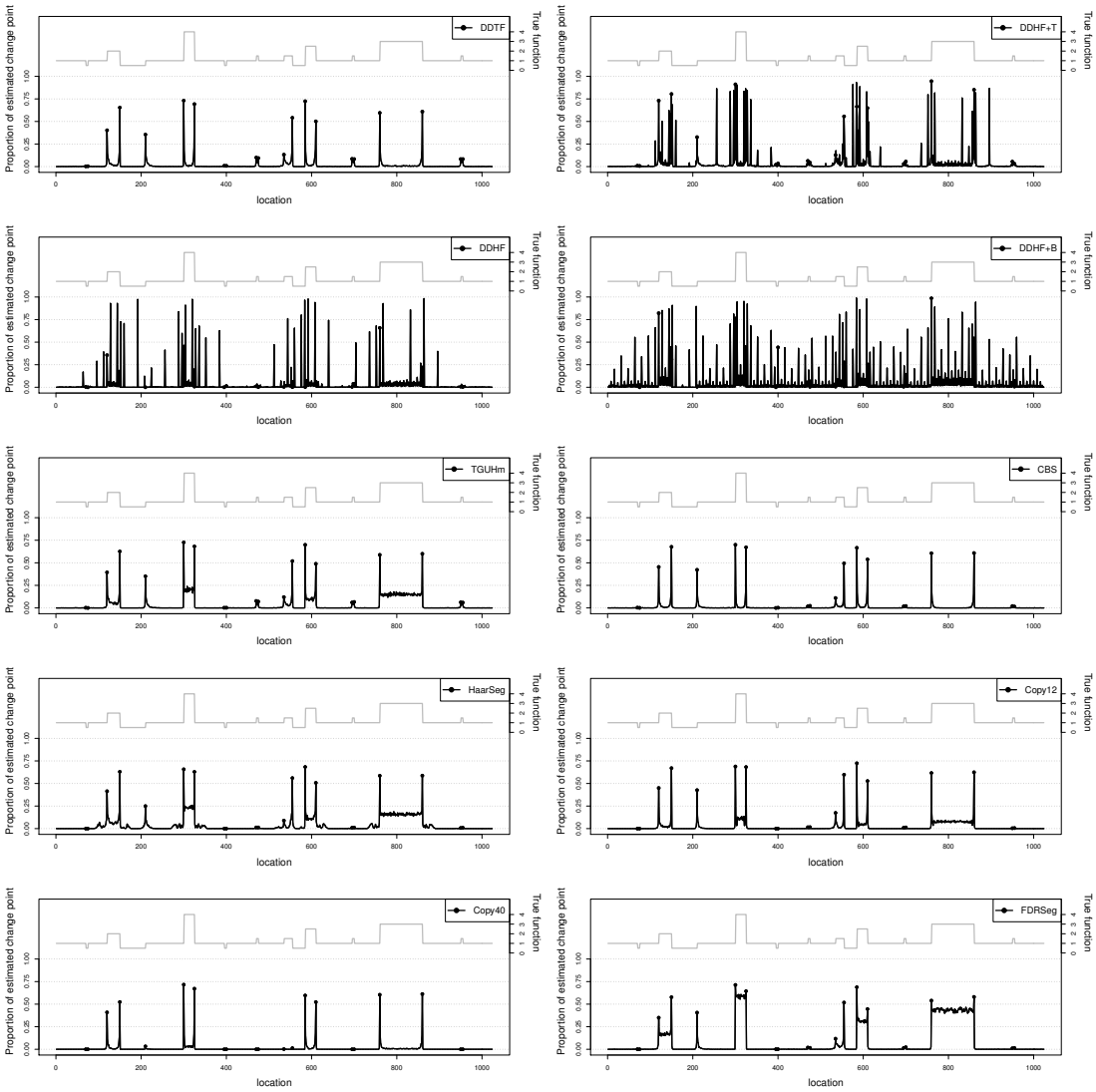


Figure C.4: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.5$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

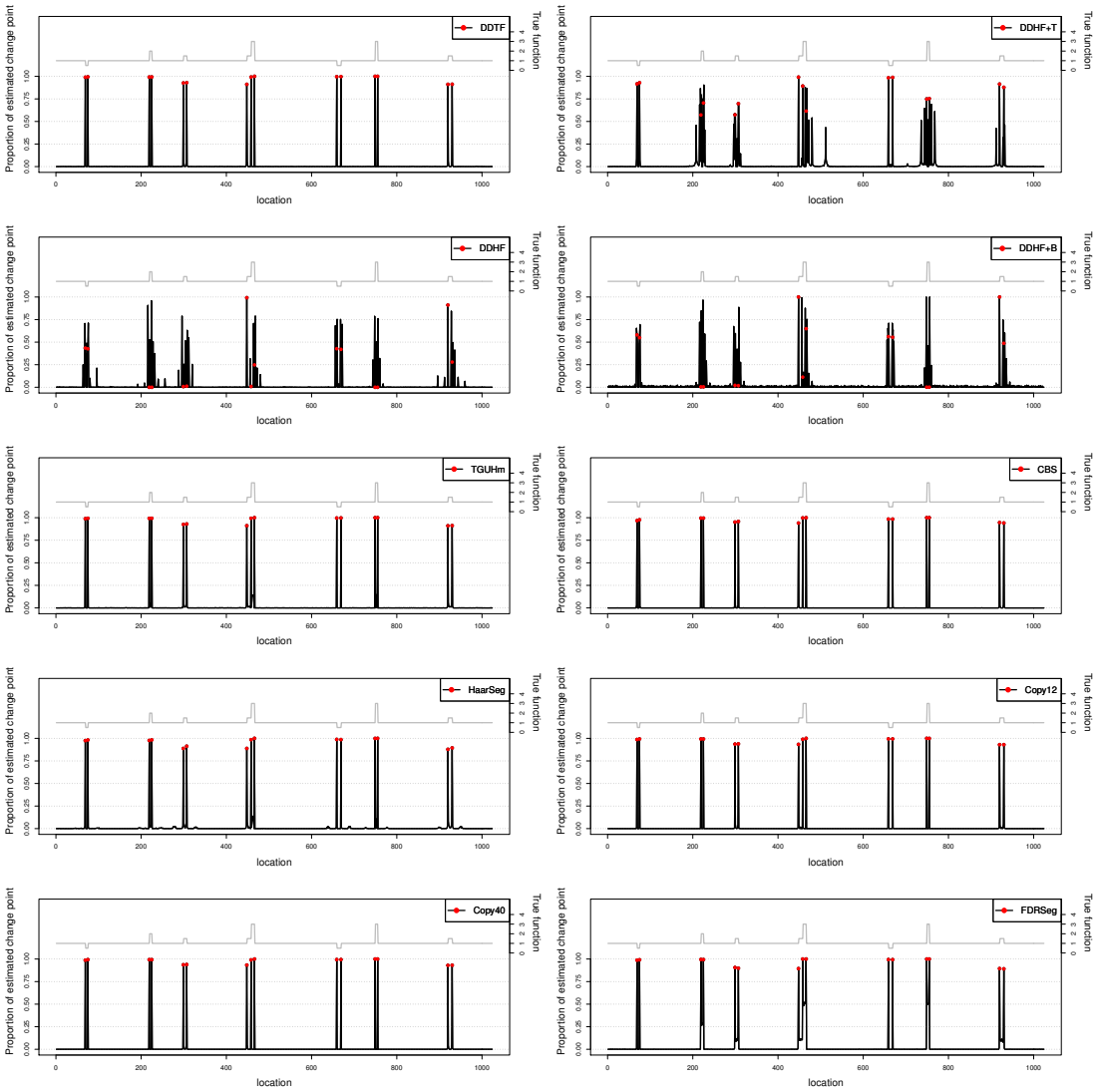


Figure C.5: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.1$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

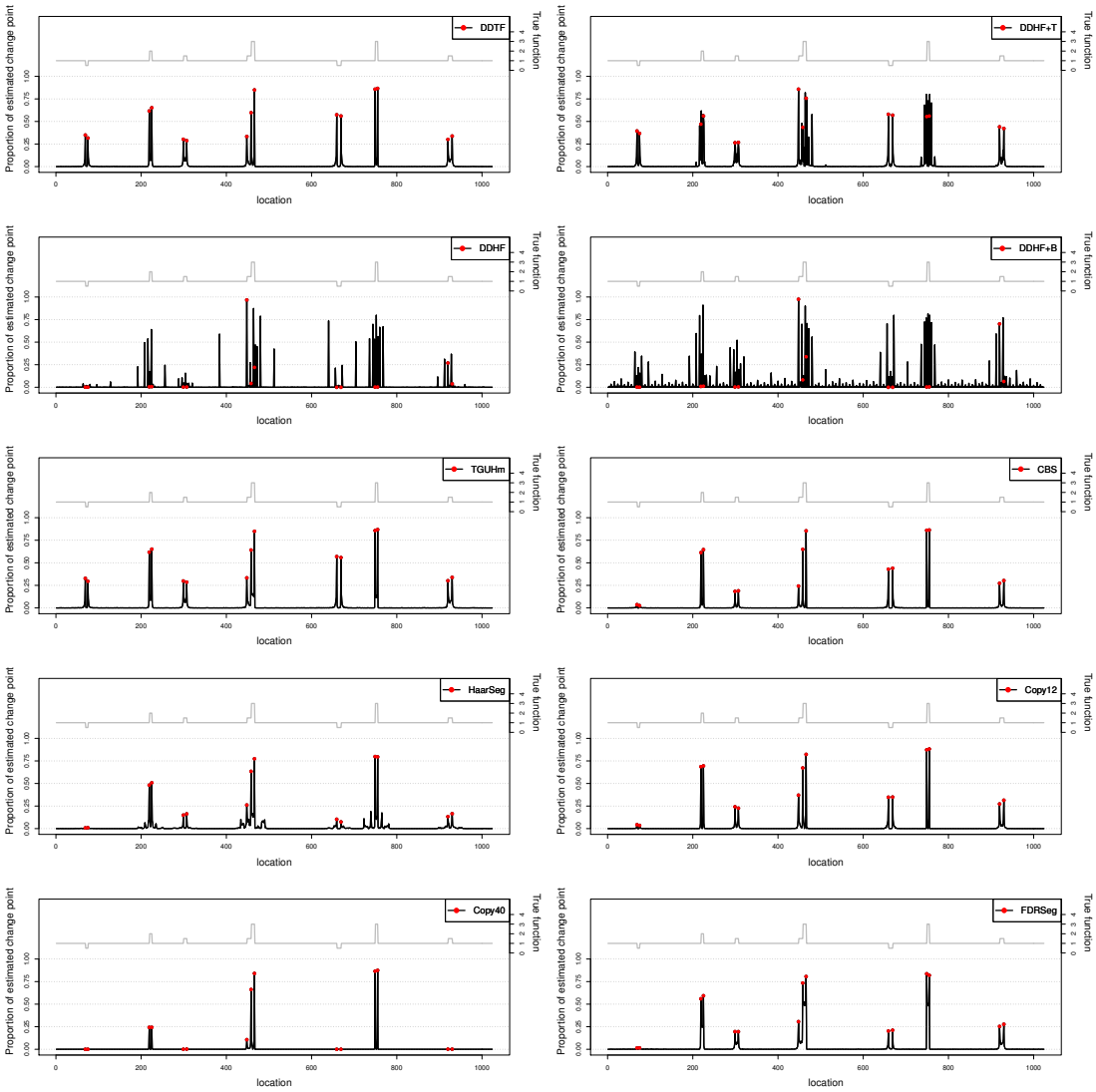


Figure C.6: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.3$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

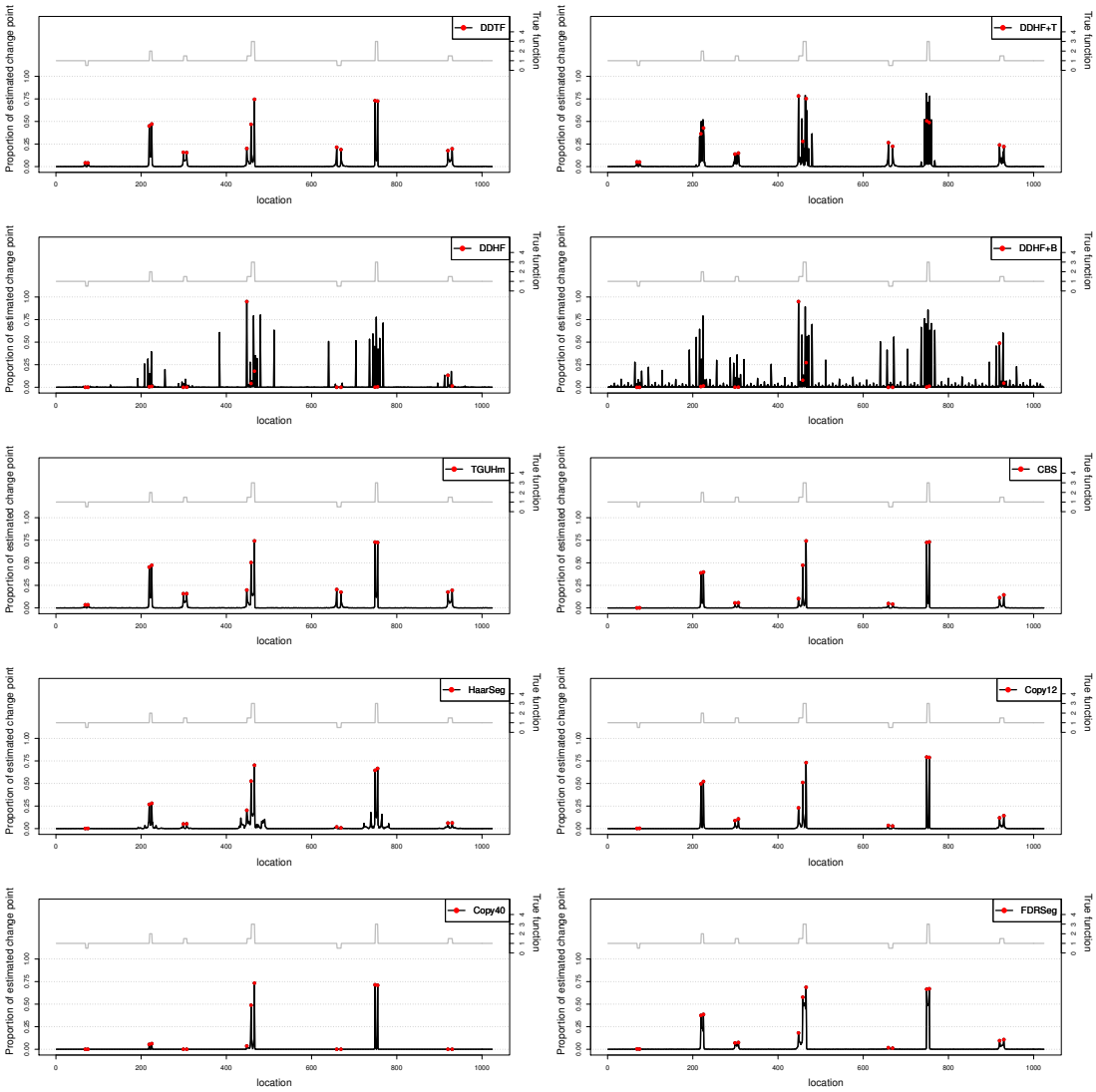


Figure C.7: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.4$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

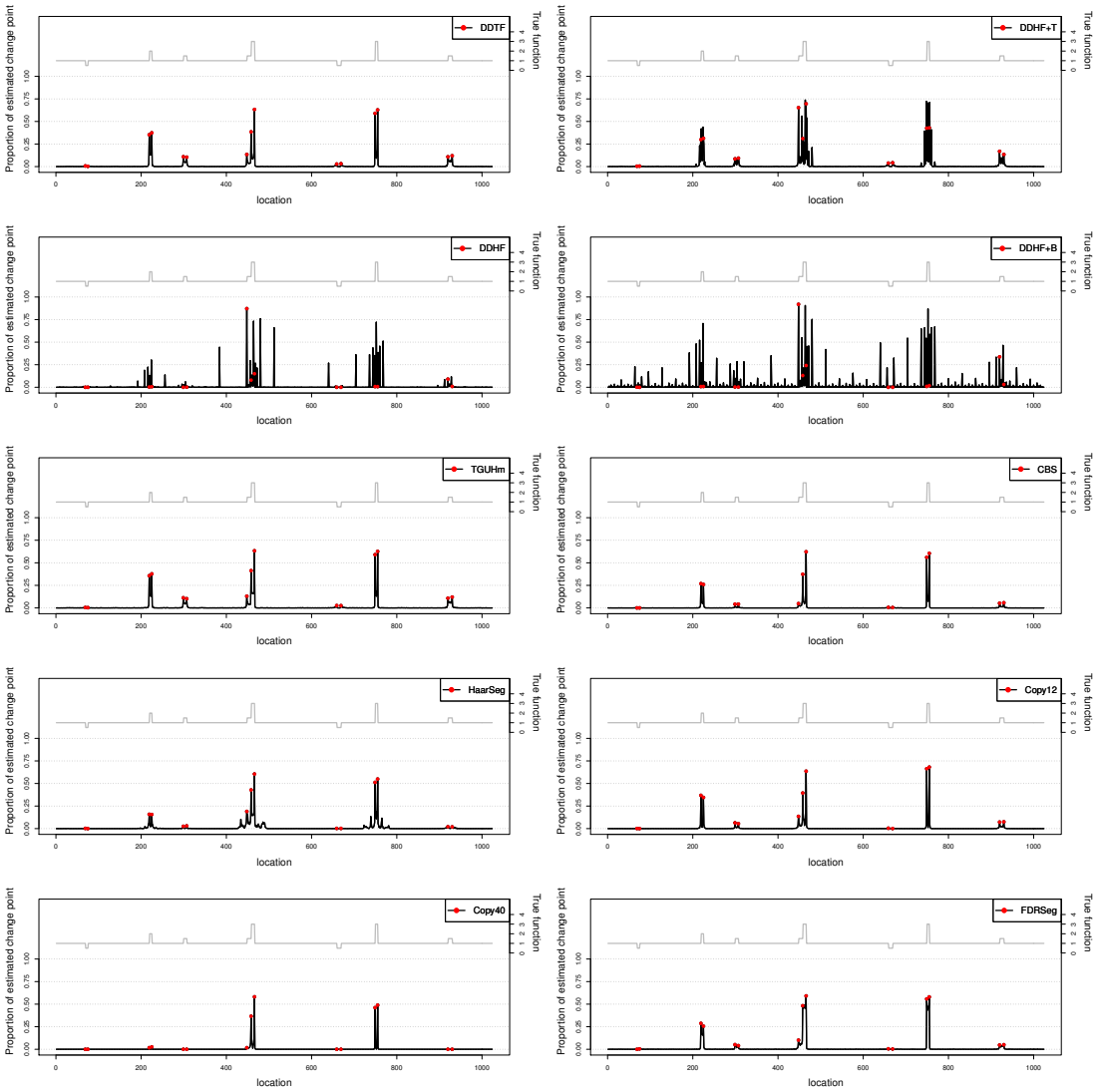


Figure C.8: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.5$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

## C.2 Additional figures of the proportion of times change-points estimated at each location: noise model 2

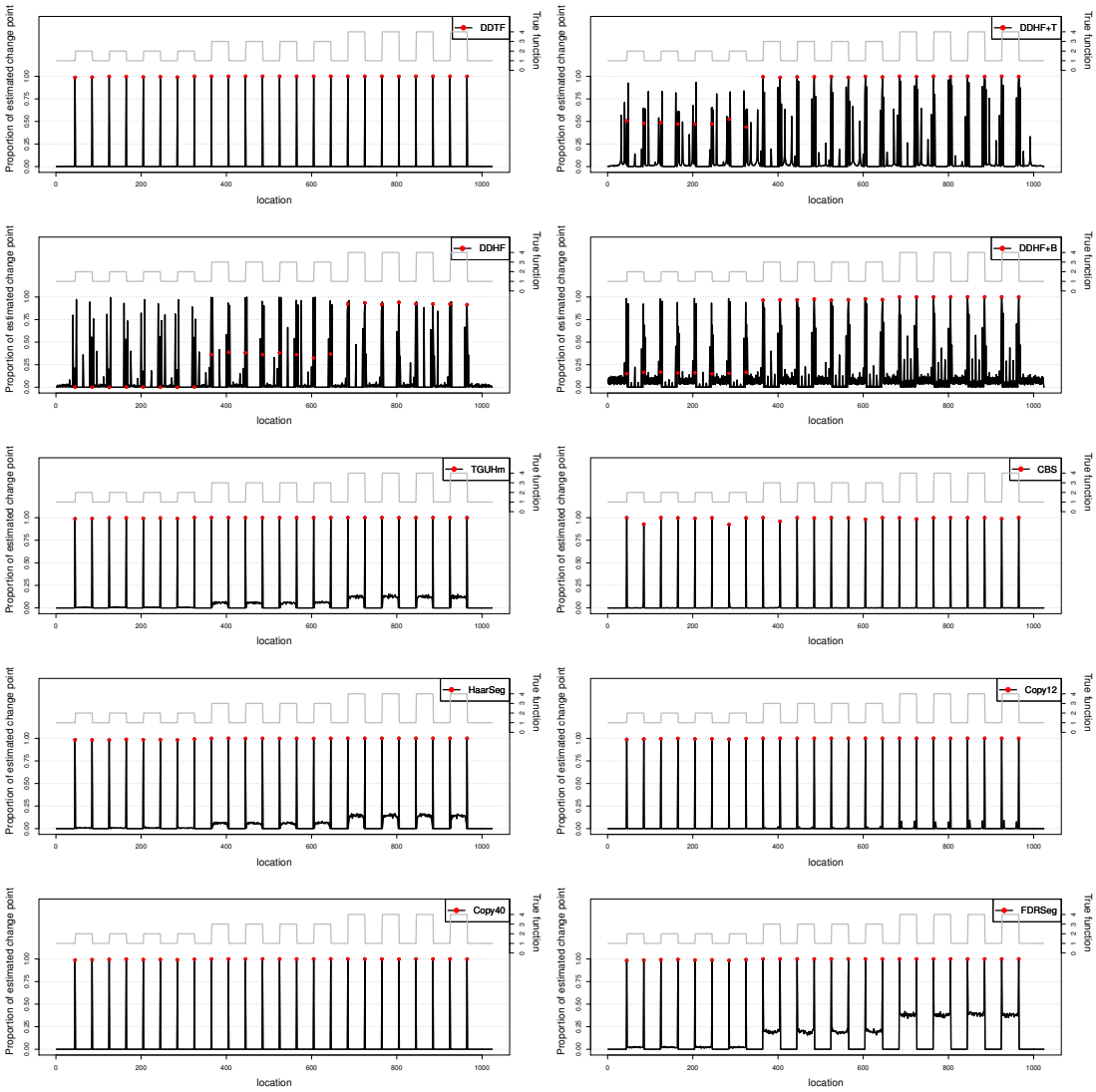


Figure C.9: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.1$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

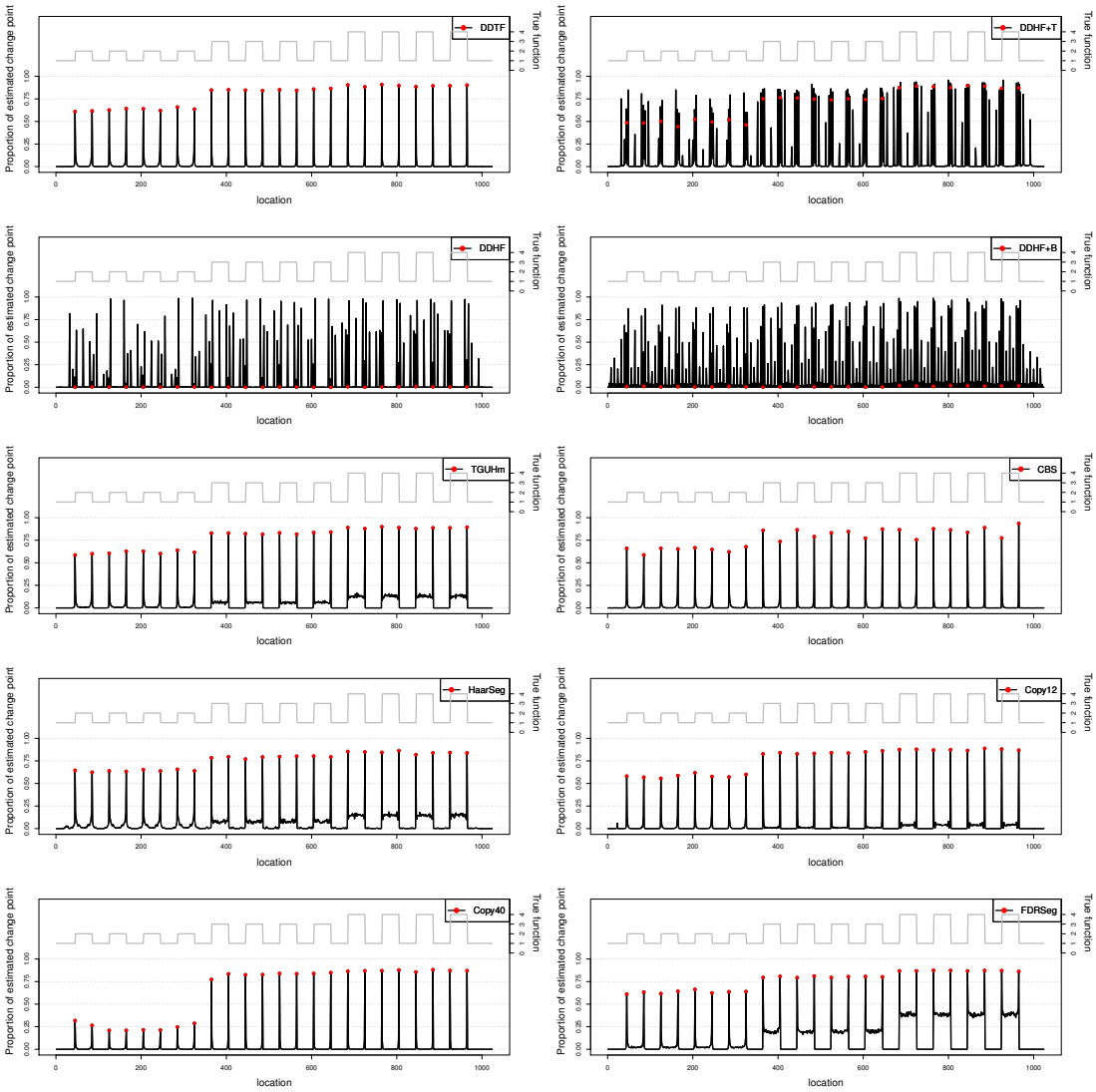


Figure C.10: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.3$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

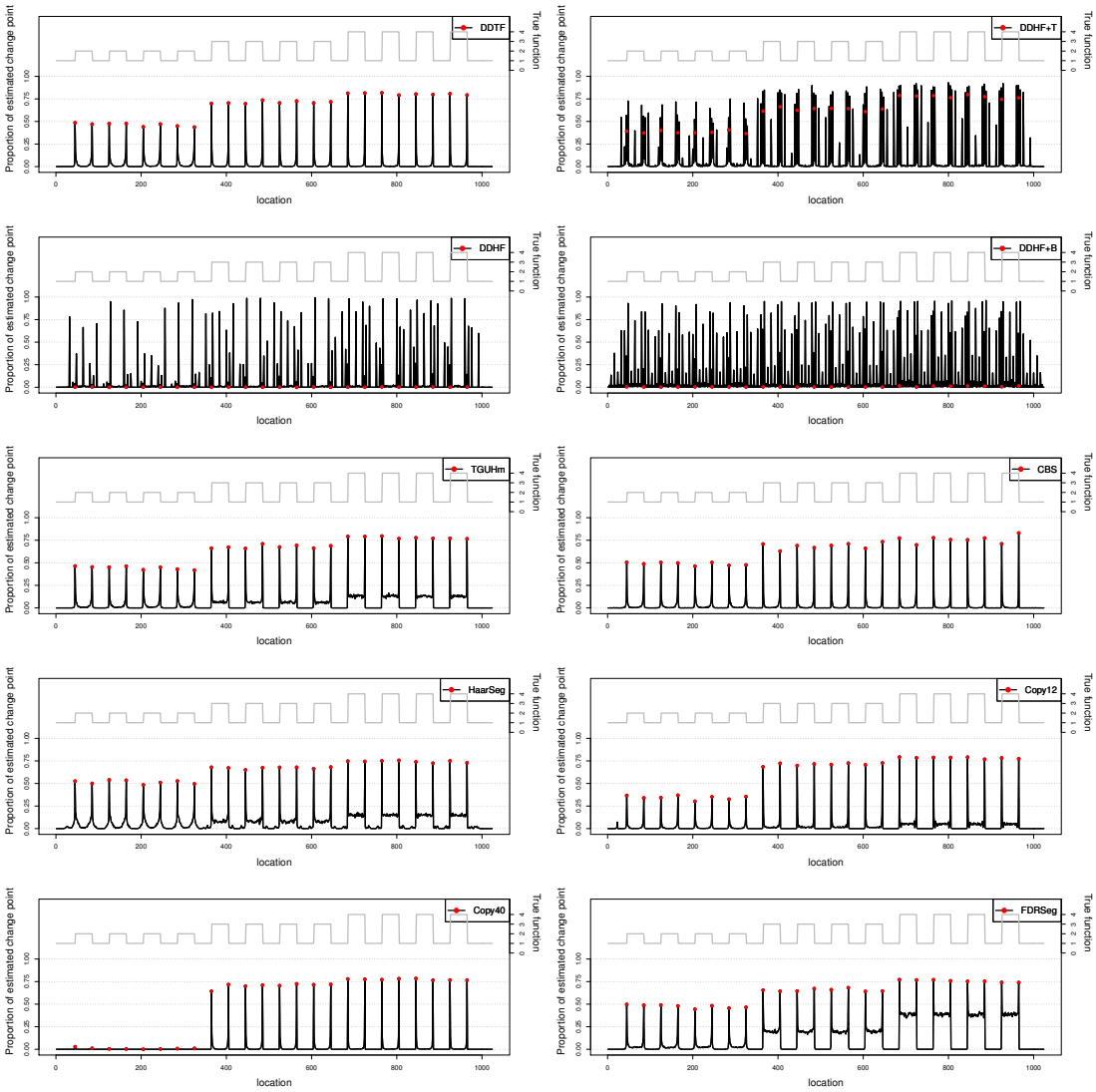


Figure C.11: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.4$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

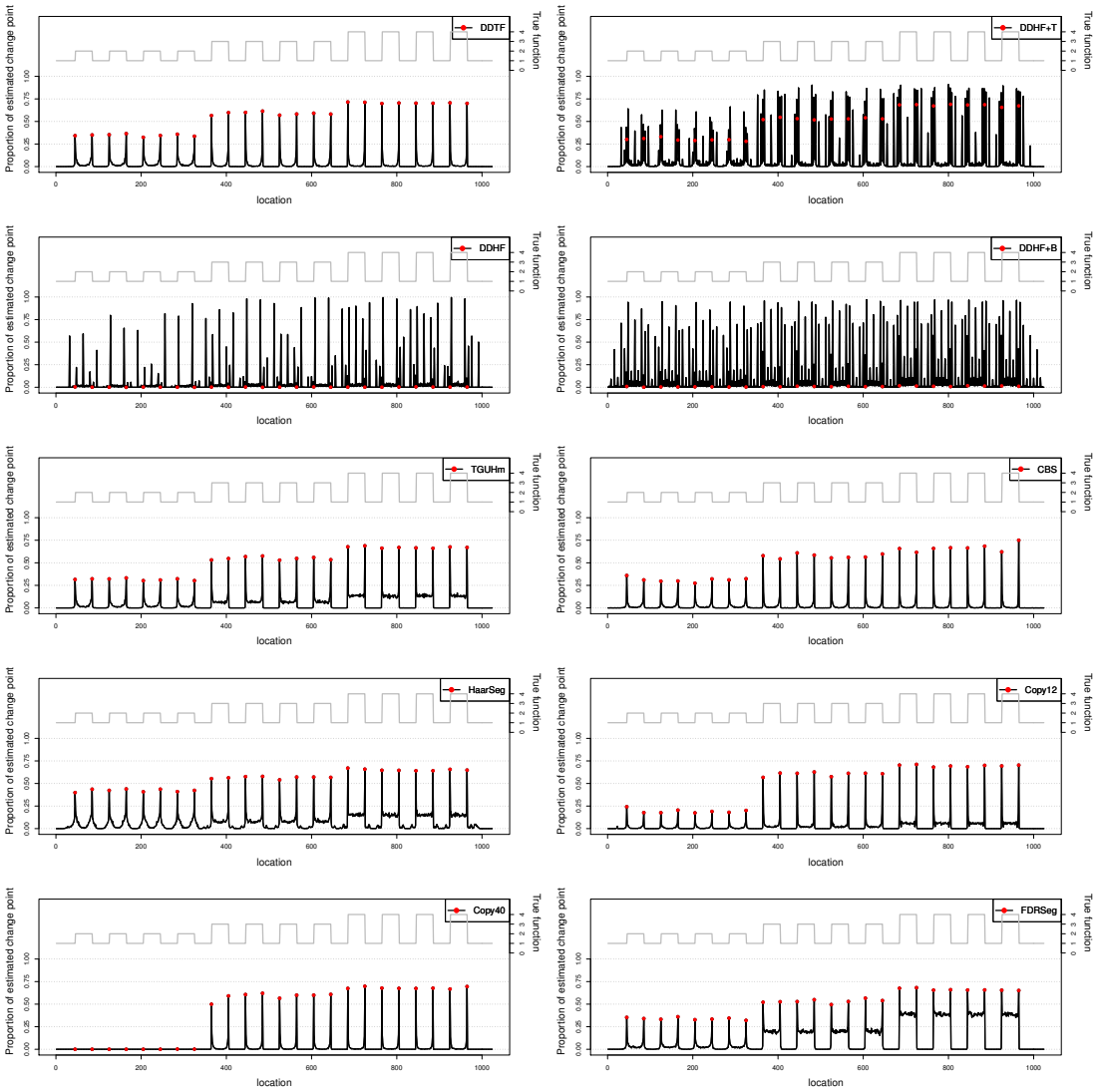


Figure C.12: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with an additive i.i.d Gaussian noise $N(0, \sigma_0^2)$ where the variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.5$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

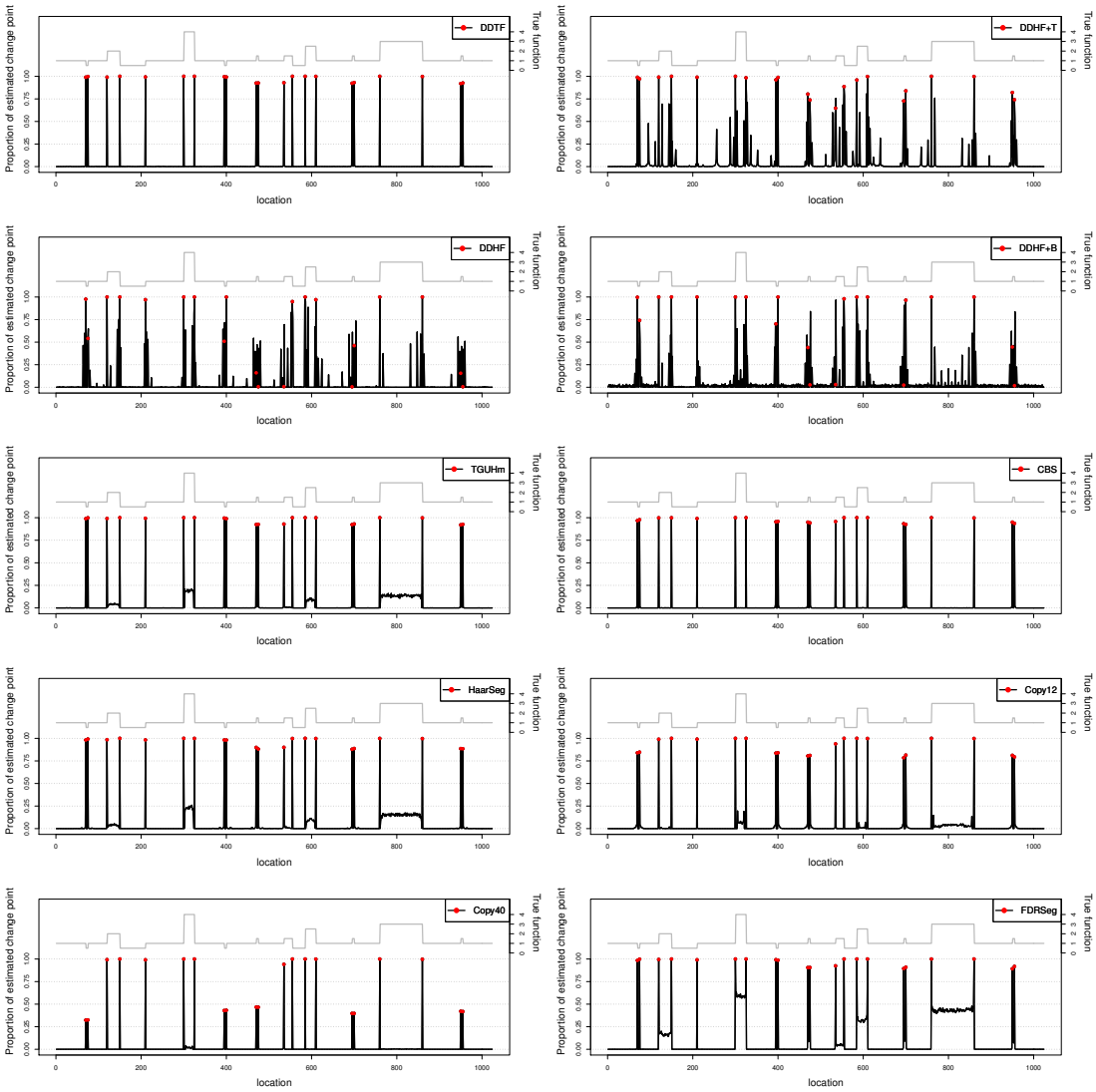


Figure C.13: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2))$, where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.1$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

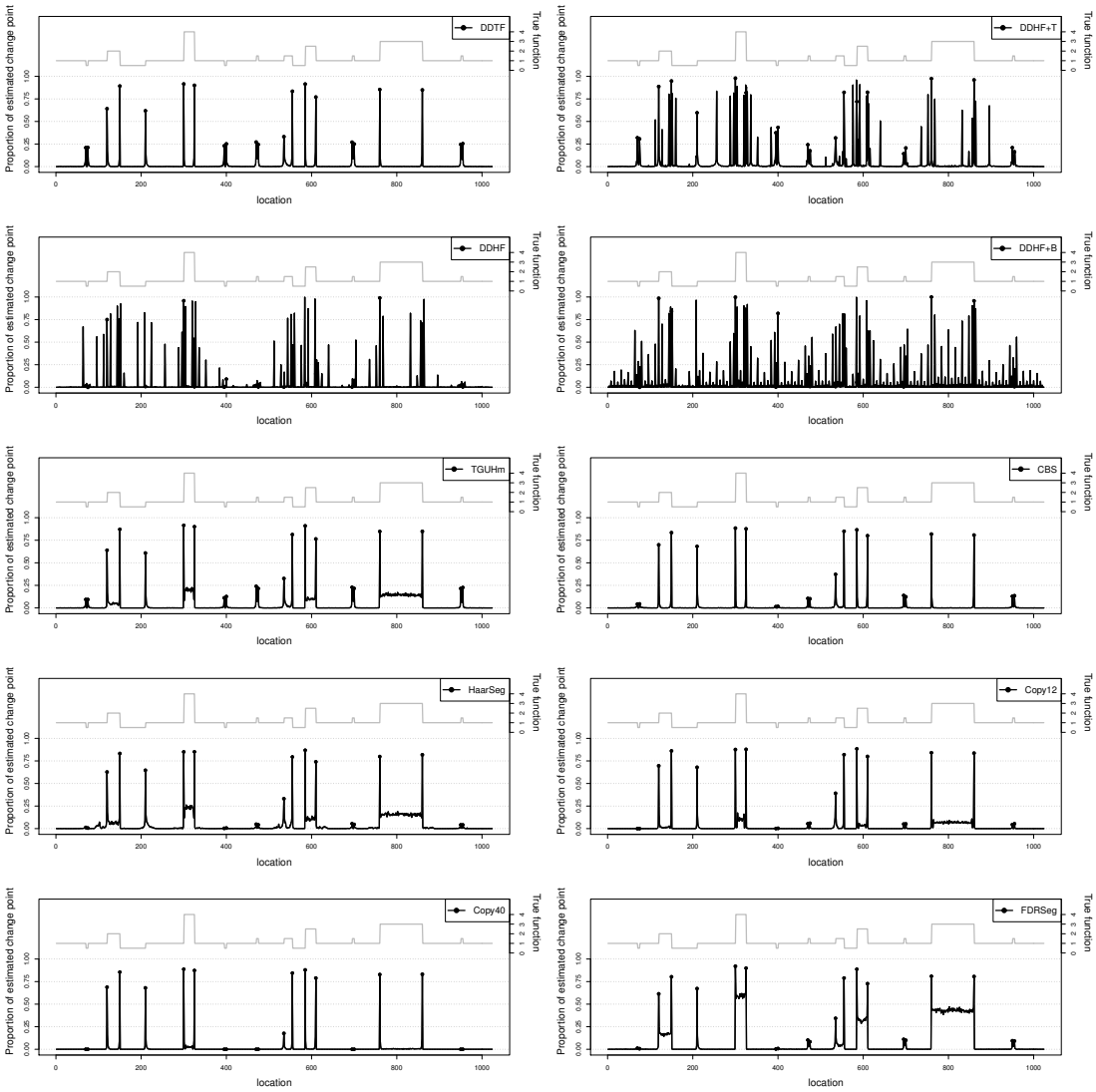


Figure C.14: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$), where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.3$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

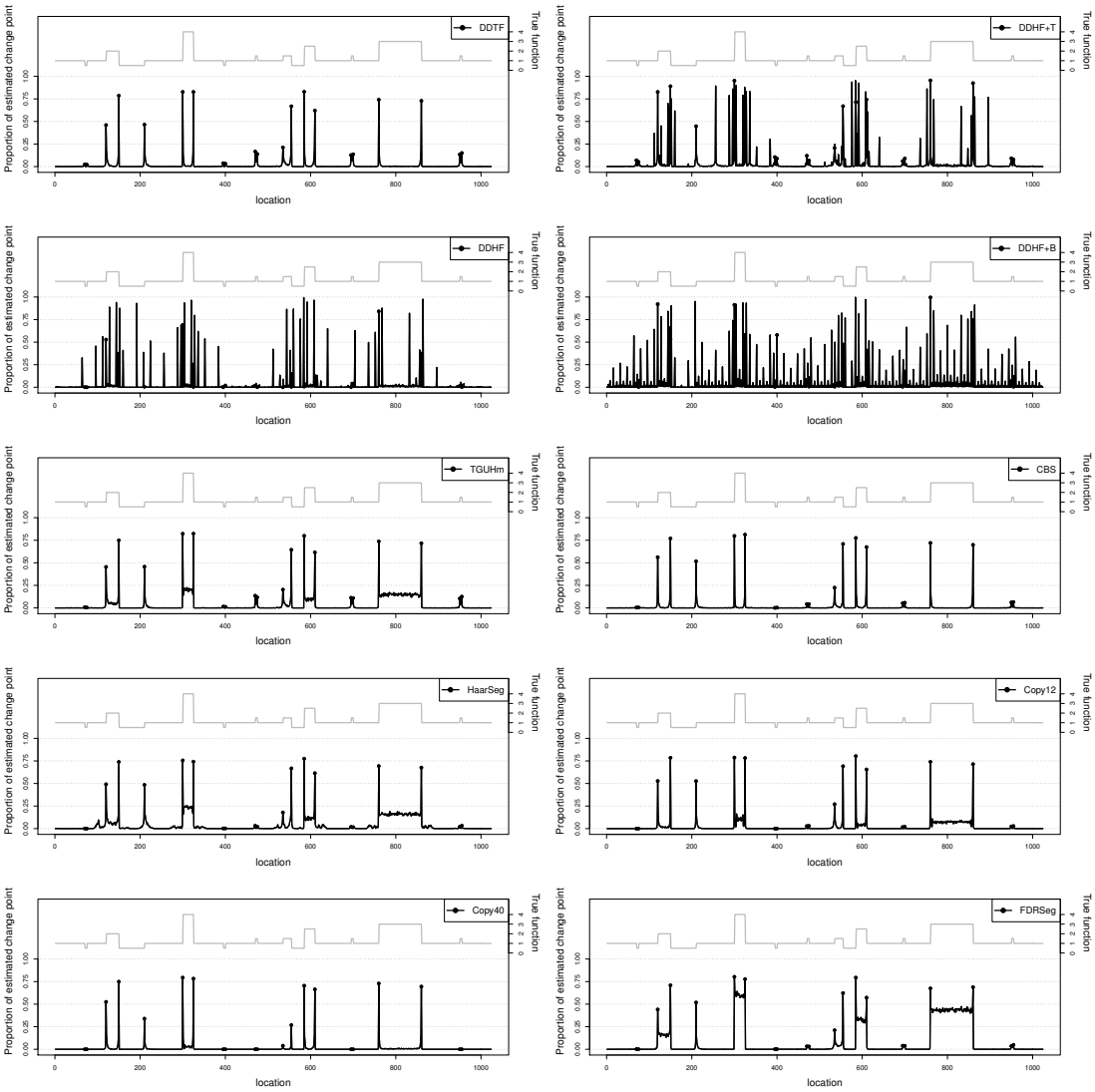


Figure C.15: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$), where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.4$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

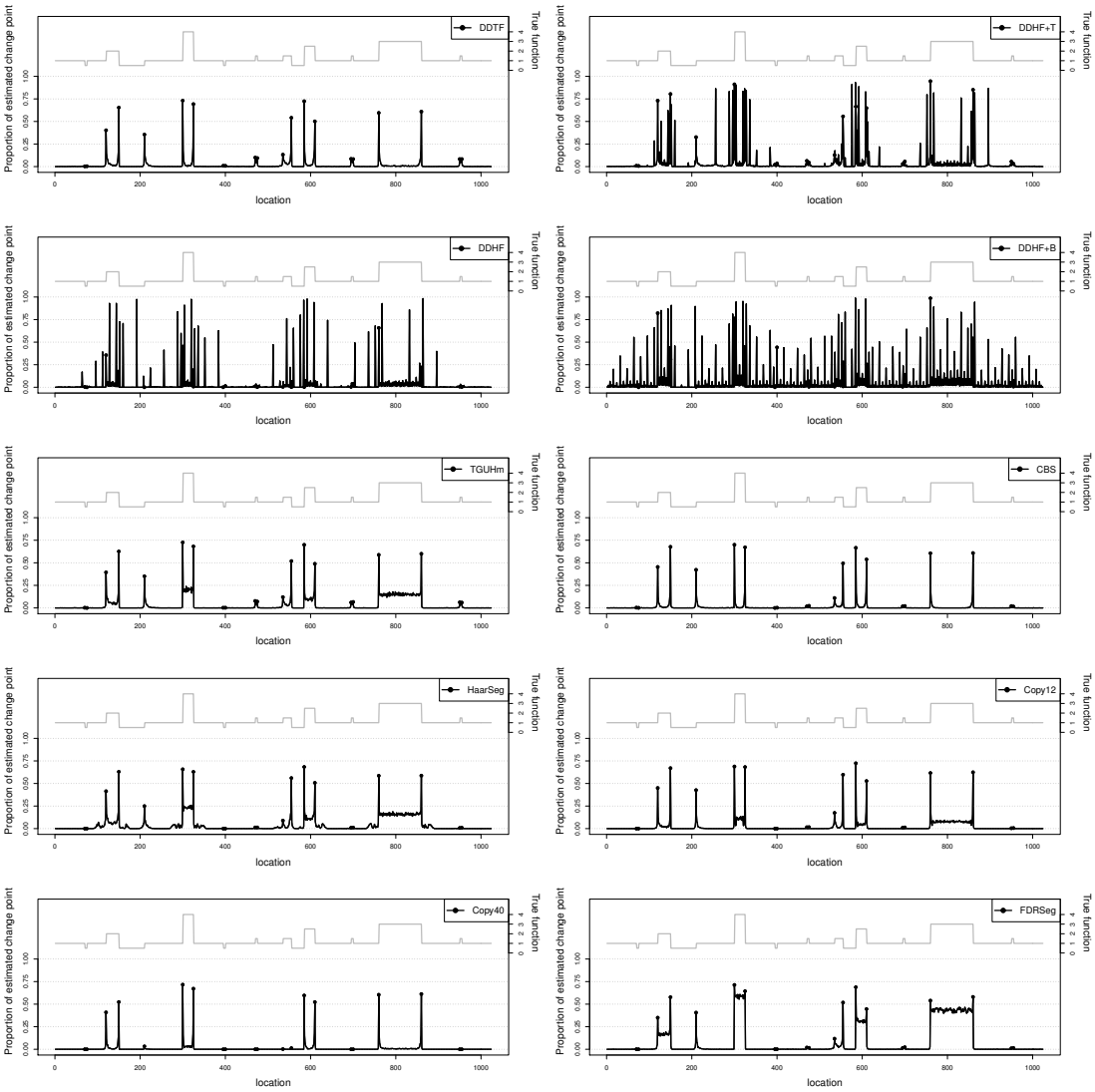


Figure C.16: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$), where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.5$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

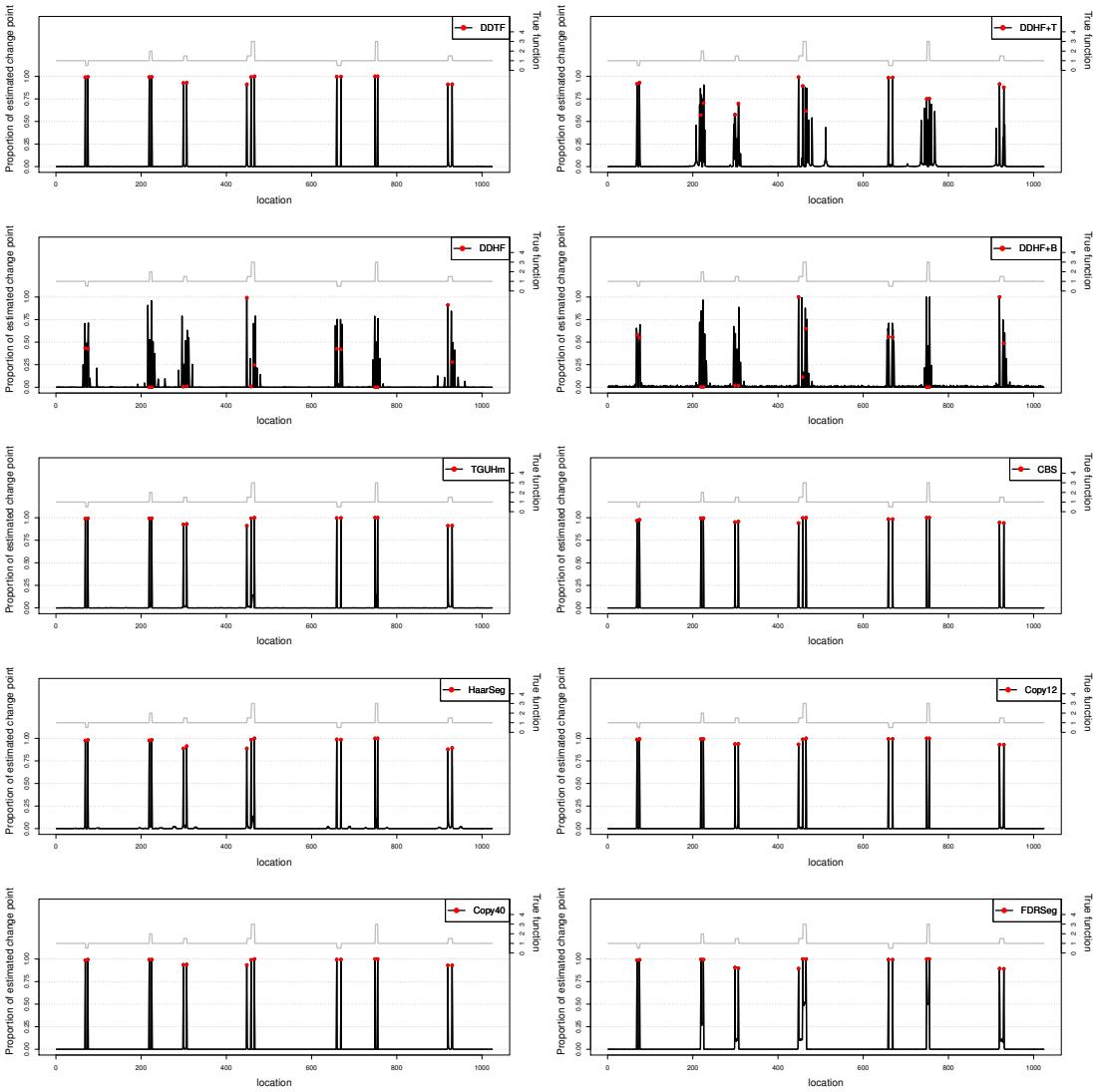


Figure C.17: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$), where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.1$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

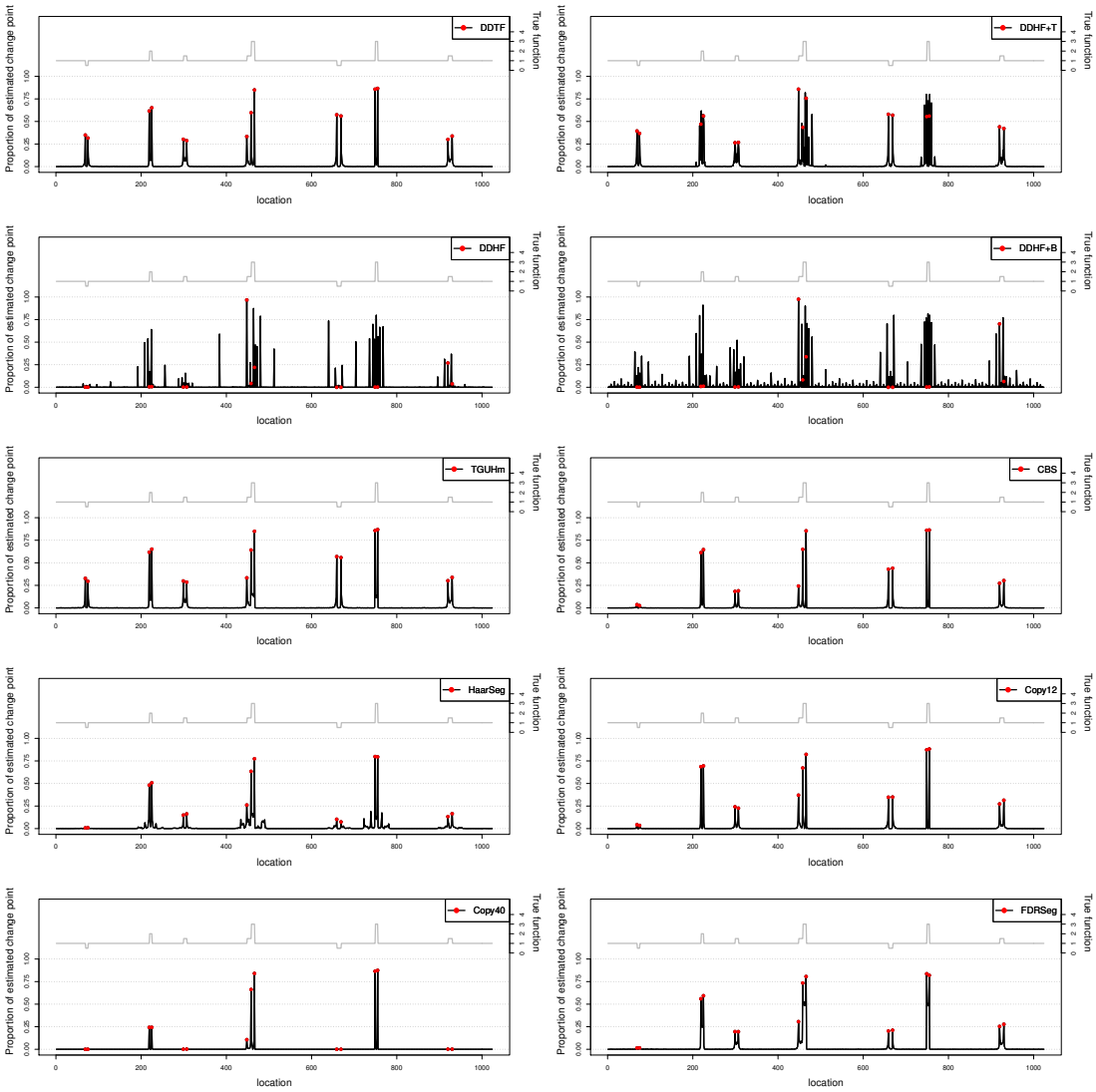


Figure C.18: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$), where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.3$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

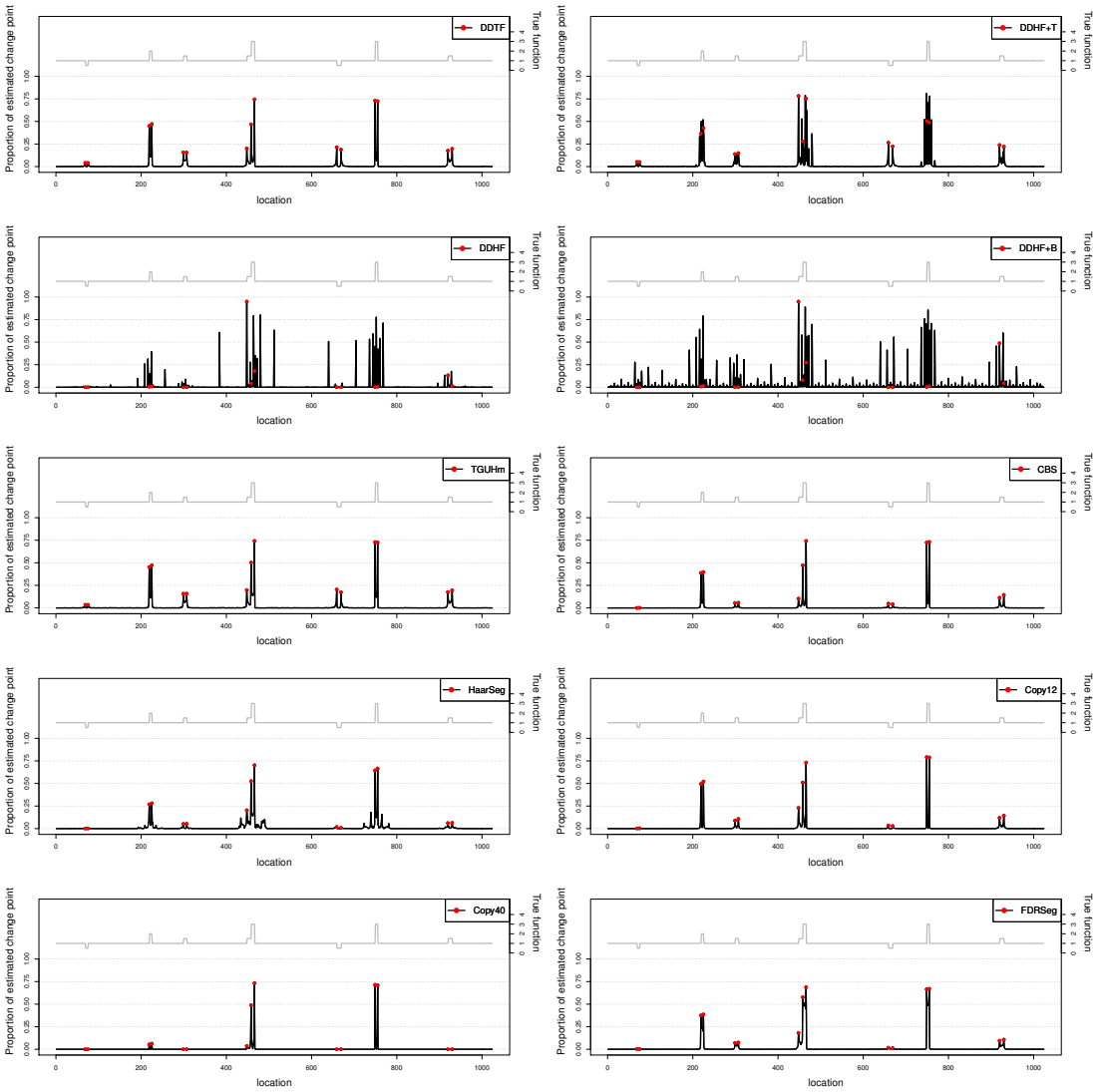


Figure C.19: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$), where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.4$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

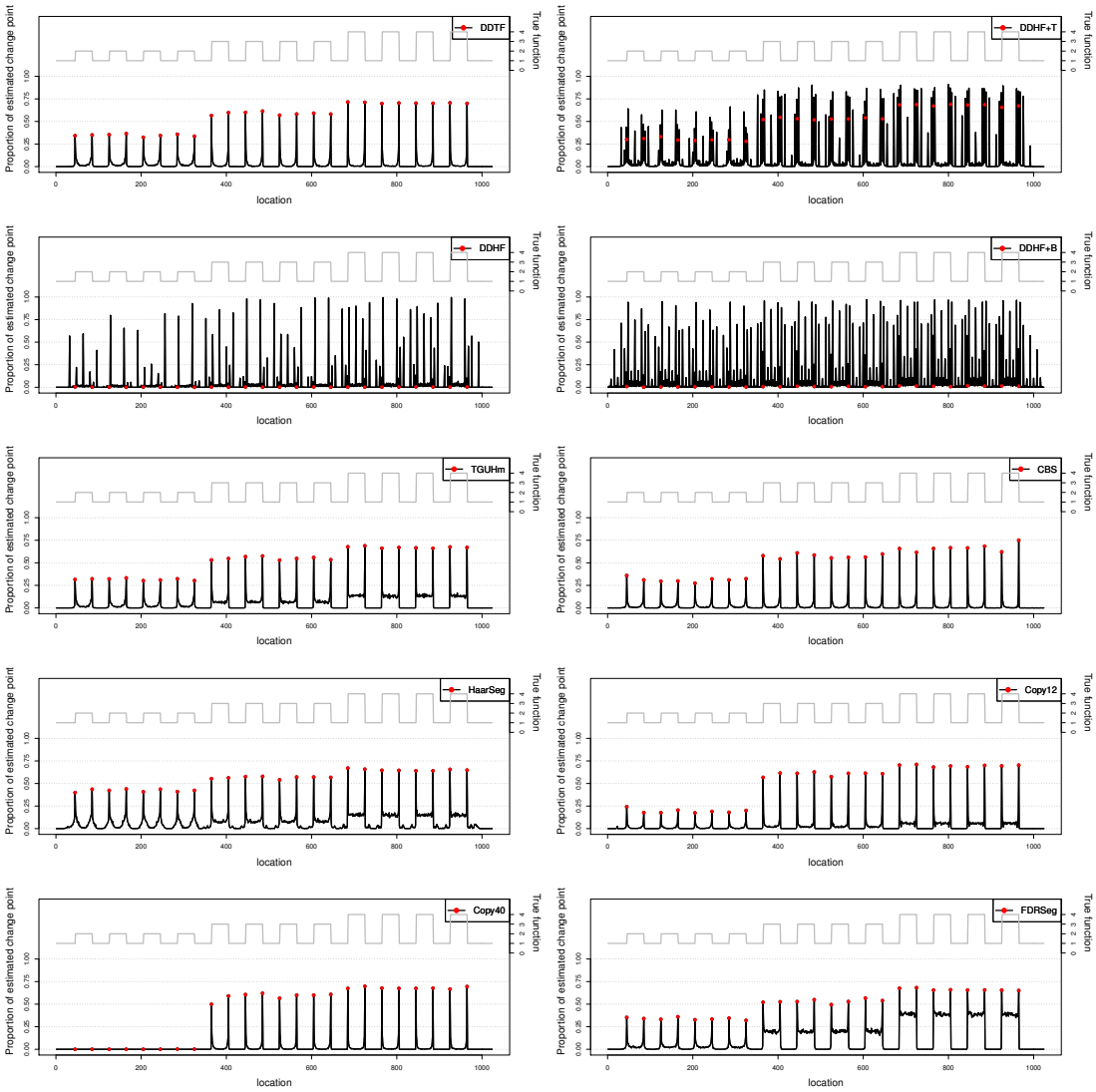


Figure C.20: Proportion of times a change-point is estimated against location corresponds to the first test function (top panel of Figure 5.12). Each value denotes the proportion of a change-point is found at the corresponding location out of 1000 simulated datasets contaminated with a mixture of two normal distributions $0.95 \times N(0, \sigma^2) + 0.05 \times N(0, 3\sigma^2)$), where variance $\sigma^2$ is defined as $\sigma_i^2 = \sigma_0^2 f_i^2$ and $\sigma_0 = 0.5$. The red dots denote proportion of each of the methods produce change-points at the correct location. The grey solid line is the corresponding test function. The left and right vertical axis shows the proportion of estimated change point and the corresponding test function's height, respectively.

# References


Abramovich, F. & Benjamini, Y. (1995). *Thresholding of Wavelet Coefficients as Multiple Hypotheses Testing Procedure*, 5–14. Springer New York, New York, NY. 28

Anastasiou, A., Chen, Y., Cho, H. & Fryzlewicz, P. (2021). *breakfast: Methods for Fast Multiple Change-Point Detection and Estimation*. R package version 2.2. 62

Belvedere, O., Berri, S., Chalkley, R., Conway, C., Barbone, F., Pisa, F., MacLennan, K., Daly, C., Alsop, M., Morgan, J., Menis, J., Tcherveniakov, P., Papagiannopoulos, K., Rabbitts, P. & Wood, H.M. (2012). A computational index derived from whole-genome copy number analysis is a novel tool for prognosis in early stage lung squamous cell carcinoma. *Genomics*, **99**, 18–24. 10, 90, 98, 141

Ben-Yaacov, E. & Eldar, Y.C. (2008). A fast and flexible method for the segmentation of aCGH data. *Bioinformatics*, **24**, i139–i145. 2, 7, 26, 27, 28, 63, 127, 176

Ben-Yaacov, E. & Eldar, Y.C. (2009). *HaarSeg: HaarSeg*. R package version 0.0.3/r4. 63

Bergamaschi, A., Kim, Y.H., Wang, P., Sørlie, T., Hernandez-Boussard, T., Lonning, P.E., Tibshirani, R., Børresen-Dale, A.L. & Pollack, J.R. (2006). Distinct patterns of dna copy number alteration are associated with different clinicopathological features and gene-expression subtypes of breast cancer. *Genes, Chromosomes and Cancer*, **45**, 1033–1040. 7

Cho, H. & Kirch, C. (2021). Two-stage data segmentation permitting multiscale change points, heavy tails and dependence. *Annals of the Institute of Statistical Mathematics*, 1–32. 81, 82

Dancey, J.E., Bedard, P.L., Onetto, N. & Hudson, T.J. (2012). The genetic basis for cancer treatment decisions. *Cell*, **148**, 409–420. 1, 8

Daubechies, I. (1992). *Ten Lectures on Wavelets*. SIAM. 13, 17

Desmond, R., Weiss, H., Arani, R., Soong, S.J., Wood, M., Fiddian, P., Gnann, J. & Whitley, R. (2002). Clinical applications for change-point analysis of herpes zoster pain. *Journal of Pain and Symptom Management*, **23**, 510–6. 179

Donoho, D.L. & Johnstone, I.M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, **81**, 425–455. 21, 22, 24, 26, 31, 102, 129, 176

Fisz, M. (1955). The limitingdistribution of a function of two independent random variables and its statistical application. *Colloquium Mathematicum*, **3**, 138–146. 108

Fridlyand, J. (2004). Hidden Markov models approach to the analysis of array CGH data. *Journal of Multivariate Analysis*, **90**, 132–153. 58

Fryzlewicz, P. (2007). Unbalanced Haar technique for nonparametric function estimation. *Journal of the American Statistical Association*, **102**, 1318–1327. 28

Fryzlewicz, P. (2008). Data-driven wavelet-fisz methodology for nonparametric function estimation. *Electronic Journal of Statistics*, **2**, 863–896. 5, 96, 100, 177

Fryzlewicz, P. (2018). Tail-greedy bottom-up data decompositions and fast mulitple change-point detection. *Annals of Statistics*, **46**. 4, 26, 28, 29, 30, 31, 32, 46, 54, 55, 56, 62, 97, 104, 105, 106, 107, 176

Fryzlewicz, P. & Delouille, V. (2005). A data-driven Haar-Fisz transform for multiscale variance stabilization. In *IEEE/SP 13th Workshop on Statistical Signal Processing, 2005*, 539–544. 110, 129

Fryzlewicz, P. & Nason, G.P. (2004). A Haar-fisz algorithm for poisson intensity estimation. *Journal of computational and graphical statistics*, **13**, 621–638. 101

Fryzlewicz, P., Delouille, V. & Nason, G.P. (2007). Goes-8 x-ray sensor variance stabilization using the multiscale data-driven Haar–Fisz transform. *Journal of the Royal Statistical Society Series C: Applied Statistics*, **56**, 99–116. 97, 101, 104

Ghosh, D. & Chinnaiyan, A.M. (2005). Classification and selection of biomarkers in genomic data using lasso. *Journal of Biomedicine and Biotechnology*, **2005**, 147. 146

Girardi, M. & Sweldens, W. (1997). A new class of unbalanced Haar wavelets that form an unconditional basis for Lp on general measure spaces. *Journal of Fourier Analysis and Applications*, **3**, 457–474. 28

Gusnanto, A., Wood, H.M., Pawitan, Y., Rabbitts, P. & Berri, S. (2012). Correcting for cancer genome size and tumour cell content enables better estimation of copy number alterations from next-generation sequence data. *Bioinformatics*, **28**, 40–47. 13, 8, 11, 41, 53, 60, 96, 99

Gusnanto, A., Taylor, C.C., Nafisah, I., Wood, H.M., Rabbitts, P. & Berri, S. (2014). Sequence analysis Estimating optimal window size for analysis of low-coverage next-generation sequence data. *Bioinformatics*, **30**, 1823–1829. 8, 10, 94, 98

Gusnanto, A., Tcherveniakov, P., Shuweihdi, F., Samman, M., Rabbitts, P. & Wood, H.M. (2015). Stratifying tumour subtypes based on copy number alteration profiles using next-generation sequence data. *Bioinformatics*, **31**, 2713–2720. 144

Hanahan, D. & Weinberg, R. (2011). Hallmarks of cancer: The next generation. *Cell*, **144**, 646–74. 1, 7, 144

Härdle, W., Kerkyacharian, G., Picard, D. & Tsybakov, A. (2012). *Wavelets, Approximation, and Statistical Applications*, vol. 129. Springer Science & Business Media. 13

Herbst, R.S., Heymach, J.V. & Lippman, S.M. (2008). Molecular origins of cancer: lung cancer. *New England Journal of Medicine*, **359**, 1367–1380. 3, 144

Hsu, L., Self, S.G., Grove, D., Randolph, T., Wang, K., Delrow, J.J., Loo, L. & Porter, P. (2005). Denoising array-based comparative genomic hybridization data using wavelets. *Biostatistics*, **6**, 211–226. 7

Jamal-Hanjani, M., Wilson, G.A., McGranahan, N., Birkbak, N.J., Watkins, T.B., Veeriah, S., Shafi, S., Johnson, D.H., Mitter, R., Rosenthal, R., Salm, M., Horswell, S., Escudero, M., Matthews, N., Rowan, A., Chambers, T., Moore, D.A., Turajlic, S., Xu, H., Lee, S.M., Forster, M.D., Ahmad, T., Hiley, C.T., Abbosh, C., Falzon, M., Borg, E., Marafioti, T., Lawrence, D., Hayward, M., Kolvekar, S., Panagiotopoulos, N., Janes, S.M., Thakrar, R., Ahmed, A., Blackhall, F., Summers, Y., Shah, R., Joseph, L., Quinn, A.M., Crosbie, P.A., Naidu, B., Middleton, G., Langman, G., Trotter, S., Nicolson, M., Remmen, H., Kerr, K., Chetty, M., Gomersall, L., Fennell, D.A., Nakas, A., Rathinam, S., Anand, G., Khan, S., Russell, P., Ezhil, V., Ismail, B., Irvin-Sellers, M., Prakash, V., Lester, J.F., Kornaszewska, M., Attanoos, R., Adams, H., Davies, H., Dentro, S., Taniere, P., O'Sullivan, B., Lowe, H.L., Hartley, J.A., Iles, N., Bell, H., Ngai, Y., Shaw, J.A., Herrero, J., Szallasi, Z., Schwarz, R.F., Stewart, A., Quezada, S.A., Le Quesne, J., Van Loo, P., Dive, C., Hackshaw, A. & Swanton, C. (2017). Tracking the evolution of non–small-cell lung cancer. *New England Journal of Medicine*, **376**, 2109–2121. 144

Johnstone, I. & Silverman, B. (2005a). Ebayesthresh: R programs for empirical bayes thresholding. *Journal of Statistical Software, Articles*, **12**, 1–38. 104, 105, 129, 180

Johnstone, I. & Silverman, B.W. (2005b). Ebayesthresh: R programs for empirical bayes thresholding. *Journal of Statistical Software*, **12**, 1–38. 23

Johnstone, I.M. & Silverman, B.W. (2004). Needles and straw in haystacks: Empirical bayes estimates of possibly sparse sequences. 23

Johnstone, I.M. & Silverman, B.W. (2005c). Empirical bayes selection of wavelet thresholds. 23

Kaymaz, Ö., Alqahtani, K., Wood, H.M. & Gusnanto, A. (2021). Prediction of tumour pathological subtype from genomic profile using sparse logistic regression with random effects. *Journal of Applied Statistics*, **48**, 605–622. 146, 152

Lengauer, C., Kinzler, K.W. & Vogelstein, B. (1998). Genetic instabilities in human cancers. *Nature*, **396**, 643–649. 97

Li, B.Q., You, J., Huang, T. & Cai, Y.D. (2014). Classification of non-small cell lung cancer based on copy number alterations. *PLoS One*, **9**, e88300. 144

Li, H. & Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, **25**, 1754–1760. 10

Li, H. & Sieling, H. (2017). *FDRSeg: FDR-Control in Multiscale Change-Point Segmentation*. R package version 1.0-3. 63

Li, H., Munk, A. & Sieling, H. (2016). FDR-control in multiscale change-point segmentation. *Electron. J. Stat.*, **10**, 918–959. 2, 63, 127

Lio, P. & Vannucci, M. (2000). Wavelet change-point prediction of transmembrane proteins. *Bioinformatics (Oxford, England)*, **16**, 376–82. 179

Magi, A., Tattini, L., Pippucci, T., Torricelli, F. & Benelli, M. (2011). Read count approach for DNA copy number variants detection. *Bioinformatics*, **28**, 470–478. 8

Mallat, S.G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, **11**, 674–693. 18

Mermel, C.H., Schumacher, S.E., Hill, B., Meyerson, M.L., Beroukhim, R. & Getz, G. (2011). Gistic2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. *BioMed Central Ltd*, **12**, R41–R41. 32, 53

Morlet, J., Arens, G., Fourgeau, E. & Giard, D. (1982). Wave propagation and sampling theory. *Geophysics*, **47**, 222–236. 13

Muggeo, V.M. (2020). *cumSeg: Change Point Detection in Genomic Sequences*. R package version 1.3. 63

Muggeo, V.M.R. & Adelfio, G. (2010). Efficient change point detection for genomic sequences of continuous measurements. *Bioinformatics*, **27**, 161–166. 63

Nason, G. (2016). *wavethresh: Wavelets Statistics and Transforms*. R package version 4.6.8. 110

Nilsen, G., Liestøl, K., Loo, P., Vollan, H., Brodtkorb, M., Rueda, O., Chin, S.F., Russell, R., Baumbusch, L., Caldas, C., Børresen-Dale, A.L. & Lingjærde, O. (2012). Copynumber: Efficient algorithms for single- and multi-track copy number segmentation. *BMC Genomics*, **13**, 591. 2, 60, 64, 125, 127, 181

Nilsen, G., Liestol, K. & Lingjaerde, O.C. (2013). *copynumber: Segmentation of single- and multi-track copy number data by penalized least squares regression*. R package version 1.28.0. 64

Olshen, A., Venkatraman, E., Lucito, R. & Wigler, M. (2004). Circular binary segmentation for the analysis of array-based dna copy number data. *Biostatistics*, **5**, 557–572. 2, 62, 127

Pei, J., Balsara, B.R., Li, W., Litwin, S., Gabrielson, E., Feder, M., Jen, J. & Testa, J.R. (2001). Genomic imbalances in human lung adenocarcinomas and squamous cell carcinomas. *Genes, Chromosomes and Cancer*, **31**, 282–287. 7

Pierre-Jean, M., Rigaill, G. & Neuvial, P. (2015). Performance evaluation of dna copy number segmentation methods. *Briefings in Bioinformatics*, **16**, 600–615. 32, 61

Schroeder, A.L. & Fryzlewicz, P. (2013). Adaptive trend estimation in financial time series via multiscale change-point-induced basis recovery. *Statistics and Its Interface*, **6**, 449–461. 179

Seshan, V.E. & Olshen, A. (2020). *DNAcopy: DNA copy number data analysis*. R package version 1.62.0. 62

Shriberg, E., Stolcke, A., Hakkani-Tur, D. & Tur, G. (2000). Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, **32**, 127–154. 179

Siegel, R., Naishadham, D. & Jemal, A. (2012). Cancer statistics. *CA Cancer J Clin*, **62**, 10–29. 3, 144

Simon, N., Friedman, J., Hastie, T. & Tibshirani, R. (2011). Regularization paths for Cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, **39**, 1–13. 153

Snijders, A.M., Nowak, N., Segraves, R., Blackwood, S., Brown, N., Conroy, J., Hamilton, G., Hindle, A.K., Huey, B., Kimura, K., Law, S., Myambo, K., Palmer, J., Ylstra, B., Yue, J.P., Gray, J.W., Jain, A.N., Pinkel, D. & Albertson, D.G. (2001). Assembly of microarrays for genome-wide measurement of DNA copy number. *Nature Genetics*, **29**, 263–264. 90

Starck, J.L., Elad, M. & Donoho, D.L. (2004). Redundant multiscale transforms and their application for morphological component separation. *Advances in Imaging and Electron Physics*, **132**, 287–348. 27

Sy, S.H., Wong, N., Lee, T.W., Tse, G., Mok, T.K., Fan, B., Pang, E., Johnson, P. & Yim, A. (2004). Distinct patterns of genetic alterations in adenocarcinoma and squamous cell carcinoma of the lung. *European journal of cancer*, **40**, 1082–1094. 7

Tukey, J. (1960). A survey of sampling from contaminated distributions. *In Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, 448–485. 60

Ummi, M.A., Barber, S. & Gusnanto, A. (2023 planning to be published). Comparison of wavelet methods for segmentation of high- throughput data. *Proceedings of Sriwijaya International Conference on Basic and Applied Sciences 2021*. 26

Ummi, M.A., Barber, S., Wood, H.M. & Gusnanto, A. (2023 submitted). Tail-greedy unbalanced Haar wavelet segmentation for copy number alteration data. *Journal of Applied Statistics*. 45

Vidakovic, B. (2009). *Statistical modeling by wavelets*. John Wiley & Sons. 13

Wagle, N., Berger, M.F., Davis, M.J., Blumenstiel, B., DeFelice, M., Pochanard, P., Ducar, M., Van Hummelen, P., MacConaill, L.E., Hahn, W.C. *et al.* (2012). High-throughput detection of actionable genomic alterations in clinical tumor samples by targeted, massively parallel sequencing. *Cancer discovery*, **2**, 82–93. 1

Wood, H.M., Belvedere, O., Conway, C., Daly, C., Chalkley, R., Bickerdike, M., McKinley, C., Egan, P., Ross, L., Hayward, B., Morgan, J., Davidson, L., MacLennan, K., Ong, T.K., Papagiannopoulos, K., Cook, I., Adams, D.J., Taylor, G.R. & Rabbitts, P. (2010). Using next-generation sequencing for high resolution multiplex analysis of copy number variation from nanogram quantities of DNA from formalin-fixed paraffin-embedded specimens. *Nucleic Acids Research*, **38**, e151–e151. 10, 96

Wu, H.T., Hajirasouliha, I. & Raphael, B. (2014). Detecting independent and recurrent copy number aberrations using interval graphs. *Bioinformatics (Oxford, England)*, **30**, i195–i203. 7

Yakut, T., Schulten, H.J., Demir, A., Frank, D., Danner, B., Egeli, Ü., Gebitekin, C., Kahler, E., Gunawan, B., Ürer, N. *et al.* (2006). Assessment of molecular events in squamous and non-squamous cell lung carcinoma. *Lung Cancer*, **54**, 293–301. 7