Stockholms universitet

Statistical process control for next-generation sequencing quality control data

Erik Thorsén

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2016:7
Matematisk statistik
December 2016

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Matematiska institutionen

Stockholm University

Mathematical Statistics
Stockholm University
Master Thesis **2016:7**
http://www.math.su.se

Statistical process control for next-generation sequencing quality control data

Erik Thorsén*

December 2016

Abstract

Assuring quality output is a vital part of the management of production processes. This is no different to the area of sequencing where the quality of sequenced data needs to be assured. In this thesis we have provided a solution to the issue of detecting and finding changes in transformed next-generation sequencing (NGS) quality control data using control charts from statistical process control (SPC) together with a change-point estimation procedure. The transformed data was assumed to follow a multivariate normal distribution. We monitored the mean vector using Hotelling's T^2 statistic and Croiser's MCUSUM control chart (cf. Hotelling, (1947), Croiser, (1988)). To monitor the covariance matrix we made use of properties of the singular Wishart distribution, introduced by Bodnar et al., (2009). Change-points were estimated using a generalized likelihood ratio when Croiser's MCUSUM chart gave a indication of a change. Our model was applied to data from an individual machine with a certain setting. A simulation study was performed to test how the control charts performed under tran- sient and persistent changes. The results of this simulation showed that Hotelling's T^2 control chart was efficient for detecting transient and large changes but poorly for persistent and small. The MCUSUM control charts detected small and persistent changes in mean and covariance matrix while worse than Hotelling's T^2 for transient and large. In the simulation study, the change point estimation procedure was shown to be accurate for large persistent changes. The constructed control charts were also applied to transformed quality control data from other machines of the same sort. In this application, Hotelling's T^2 control chart showed no great difference for transformed quality control data between the machines but a difference between the settings on these machines. However, the MCUSUM charts detected large structural differences between the machines on the same setting. These differences were discovered in the mean and covariance matrix. All control charts and simulations where migrated to C++ using the Rcpp package together with OpenMP, a parallel programming model, to increase R's computational power. The improvement was shown to be large compared to base R performance in a benchmark.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: ethorsn@gmail.com. Supervisor: Taras Bodnar.

Acknowledgements

I want to thank my two supervisors Ass. Prof. Taras Bodnar and Phd. stud. Johan Dahlberg who supported me in the writing of this thesis. Taras, who kindly answered all my questions on statistical process control and multivariate statistical analysis and Johan, who helpfully described the procedures of sequencing with detailed instructions on the problem. Without their help, this thesis would not have been possible. I would also like to thank the SNP&SEQ platform in Uppsala for the opportunity of doing my thesis in their facility. Last, but not least, I would like to thank the people at the platform who answered all my questions regarding the machines and programming.

Contents

Introduction

Quality assurance is vital in the management of all production processes. This statement is just as valid for the SNP&SEQ technology platform (SNP&SEQ) at the Science for Life Laboratory in Uppsala. They provide state of the art sequencing and genotyping techniques to Swedish researchers. Ensuring that sequenced data is of high quality is therefore no exception to any other production process. The platform currently use fixed quality limits, which are decided by the machine manufacturer, to decide if a sample passes quality control or not. It is possible that using these quality limits, we may fail to take the inherent variance of the process into account. We may also fail to detect persistent changes in the process which manifests inside the quality limits. A process may produce products of lower quality on average, while still being inside the limits. It may also be so that the process produce products whose quality vary more inside these limits, which may not be desirable. The aim of this thesis is to use statistical methods to detect changes in next generation sequencing quality control data. Both transient- as well as persistent changes will be considered. Transient changes will be considered in the mean and persistent changes will be considered in the mean and variance/covariance structure. If we discover a persistent change, we would like to estimate when it occurred. This will only be considered for changes in the mean. In this thesis we will use the framework of statistical process control to detect changes and a change point estimation procedure to say where in time these changes occured.

Statistical process control (SPC) was initially developed by Shewhart, (1931) to provide a framework for monitoring a process in a sequential setting. In his book, Shewhart introduced several concepts but we will mainly focus on the control chart, which is to be considered one of his major contributions. The control chart is an effective and intuitive statistical tool, originally constructed for monitoring and analysing industrial production processes. It is used to deduce if the monitored process is performing as expected or desired, referred by Shewhart as the process is in control (IC). Times where the process is not performing as desired was referred to as out of control (OC).

SPC has been applied to many different areas. Lim et al., (2014) discussed the use and impact of SPC in the food industry and Thor et al., (2007) provided a systematic review of SPC's application in health care improvement. In Golosnoy et al., (2010) the authors discussed the use of SPC to monitor portfolio weights. SPC has been used in the area of genomics, as shown in Model et al., (2002). Here, the authors considered transient changes in data from large scale microarray experiments. To our knowledge, no previous research has been performed on how statistical process control can be applied on NGS quality control data for detecting persistent and transient changes in the mean and variance/covariance structure.

As described in chapter 1.3 Qiu, (2013), statistical process control is usually divided into 2 phases. Phase 1 includes deduction of the process' ordinary behaviour and the characteristics of it. This could include estimating in-control parameters or simply specifying desired in-control paramters (cf. Chakraborti et al., (2008)). It can also include validating necessary assumptions for the control charts. In the next phase, which is called Phase 2, we observe the process in a sequential manner. The process is monitored with the help of our control chart together with

its charting statistic. If the charting statistic exceeds a pre-specified control limit we deem the process as being OC. This thesis will mainly focus on Phase 2 monitoring. This will include a simulation study of how the control charts react to different OC behaviour and a application on quality control data from next generation sequencing machines.

Since the introduction of statistical process control, new refinments have been introduced. The cumulative sum (CUSUM) chart and exponential moving average (EWMA) (cf. Page, (1954), Roberts, (1959)) are some of these refinements. The CUSUM chart was introduced to answer the issue of detecting a small and persistent change in the mean of a process, which had not been covered in Shewhart, (1931). In this thesis, we will use two multivariate control charts to monitor changes in the mean and covariance matrix of a multivariate normal process. These are Hotelling's T^2 statistic and Croisers MCUSUM chart presented in Hotelling, (1947) and Croiser, (1988), respectively. Hotelling's T^2 control chart have been documented to detect large and transient changes well, while being very poor at detecting small and persistent changes (cf. Croiser, (1988)). Croisers MCUSUM chart will be used as a complement to detect small and persistent changes of the process. Both charts are used to monitor the mean of a multivariate normal distribution. To monitor the covariance matrix we will make use of properties of the singular Wishart distribution which was introduced in Bodnar et al., (2009). To estimate potential change points of the observed process we will use an estimate based on a generalized likelihood ratio, described in Gombay and Horvath, (1994). This method will only be used for the mean of the process.

The data to be used in this thesis consists of multivariate irregular time series. Each observation represent the machine being run at a specific time. We will sometimes use "run" as a reference to an observation which provide quality control data. For each machine there is a number of settings which provide different quality characteristic but also changes the inherent dimensionality of the problem. The machine may be used on different settings from run to run and it may stand idle from time to time. Therefore, the following assumptions will be made in this thesis; the irregularity will be disregarded and a observation is independent from previous observations. The data to be used in this thesis will be transformed and the transformed data will be assumed to follow a multivariate normal distribution. Before the transformation is performed, those runs which are poor according to today's quality criteria are removed. A small section, introducing the transformation methods and evaluating the validity of these assumptions are presented in the Appendix section 7.1.

The in-control parameters where estimated from transformed quality control data from a specific machine. Using these in-control parameters, we constructed Hotelling's T^2 and Croisers MCUSUM control chart for the mean and the covariance matrix. In a simulation study it was shown that the MCUSUM chart was able to detect persistent and small changes quickly on average. Hotelling's T^2 statistic showed very poor performance in the same simulation study while being proficient in discovering transient changes in the mean. The change-point detection procedure was seen to estimate the change-point well in the case of large changes in the mean and when the in control period was larger or equal to the out-of-control period. The simulations and code were implemented in C++ using the Rcpp extension (c.f. Eddelbuettel, (2013)) together with OpenMP, a parallel programming model (c.f. Chandra et al., (2001)), to increase computational efficiency. In a benchmark, Rcpp computational power was compared to base R's and base R running in parallel using the foreach package. In this benchmark, Rcpp showed to be very proficient in large simulation studies.

When applying the control charts to transformed quality control data from other machines of the same sort, the chart showed the following results. Hotelling's T^2 statistic confirmed that the different settings of the machines give different quality characteristics but did not show any clear difference between the machines themselves. However, Croisers MCUSUM control chart showed large evidence that the estimated in-control parameters do not fit the other machines' transformed quality control data.

All code used in this thesis can be obtained at the Github repository `https://github.com/Ethorsn/SPC-NGS-Rcpp`.

1.1 Outline

This thesis will be outlined as follows. First, a brief introduction on Next Generation Sequencing (NGS) and the operational routines at the SNP&SEQ platform is presented. In this section we introduce next generation sequencing, the variables which are collected when the sequencing is performed. Thereafter a chapter presenting the methods to be used in this thesis is presented. Next, an exploratory data analysis is conducted of the two datasets at our disposal and continuing on with results in form of simulations and an application of our methods on quality control data from NGS machines. We end this thesis with a discussion and conclusions.

Introduction to NGS and operational routines at the SNP&SEQ platform

This section aims to provide some understanding to Next Generation Sequencing (NGS), the structure of data, the quality measurements which are collected when sequencing a sample and how the machines work. All information presented here have been attained from Illuminas website (June 6, 2016) (machine manufacturer), through discussions with technicians at the SNP&SEQ platform or otherwise as cited.

Next Generation Sequencing (NGS) was introduced in 2005, Illumina, (2016). As described in Metzker, (2010), NGS is a collection of approaches or work flows containing library preparation and sequencing. Here, we will only consider one sequencing procedure, sequencing by synthesis (SBS), which Illumina's machines use. The general work flow with SBS follows the steps seen in Table 2.1 with accompanying Figure 7.1 which is found in the Appendix, section 7.2. These two provide a simplification of what is presented in Illumina, (2016).

Table 2.1: The general workflow using sequencing by synthesis.

1. Library preparation	The DNA sample is randomly fragmented and then prepared for placement on a plate (flowcell).
2. Cluster generation or amplification	The prepared library is loaded onto a plate (flowcell) which has some special features. These features together with what is called solid-phase amplification make it possible to clone each fragment placed on the flowcell such that each fragment results in a cluster.
3. Sequencing	The flowcell is placed inside the machine where the sequencing can begin using the SBS technology.
4. Preliminary & data analysis	The sequenced data and quality variables are extracted.

The two first steps, library preparation and cluster generation, is performed manually or by a machine. The third step where the sequencing is performed, constitutes of a process where of one of four fluorescent-labelled nucleotides binds to their complementary base pair, contained in the DNA fragment on the flowcell. This process results in a florescent light which can be captured by a camera in the machine. There is a trade-off between cluster generation and sequencing performance. Too much cluster generation results in over-clustering and the machine can not distinguish the florescent lights that appear between adjacent clusters. On the other hand, too few clusters results in under-clustering and the machine can not detect the florescent light properly, resulting in low daya yeild.

The flowcell, which was previously mentioned in table 2.1, has 8 lanes and each of these lanes can provide two reads. Figure 2.1 illustrates this hierarchical structure of a flowcell. The measurements made on a lane will in general be worse for Read 2. The camera in the machine emits a light in each step of the sequencing procedure. This light is said to damage the sample

placed on the flowcell. The second reason is that the sequencing procedure takes time. The second read can be performed up to two or three days after the run was started. The quality of the sample can deteriorate under the sequencing procedure because of the amount of time it is inside the machine.

Each read represents a measurement of the same cluster but from opposite directions. Between each read, the fragments which are to be sequenced, turns 180 degrees. We can therefore interpret it as read 1 sequence the DNA fragment from the top to the bottom and read 2 sequence the DNA fragment from the bottom to the top. This procedure is performed by the machine with the help of a set of chemicals. The procedure is not deterministic and errors often occur.

When a client provides their sample, they will also specify a number of settings which is to be taken into account in a run. A client will specify which tags corresponds to a specific sample. A tag corresponds to a unique sequence of base pairs. A flowcell can provide several observations from each read in a given lane, corresponding to the search for different tags. This is seen in Figure 2.1. The number of observations may differ between flowcells. This is the structure of the lowest level in a flowcell, which we will refer to as a flowcells "tag level". A higher level exists and at this level only one observation for each read is provided. We will call this level the "read level". The client also specifies the number of cycles, lanes and reads to be used in a run. The cycles represents how many times we are repeating the SBS procedure. A higher cycle number implies a longer run time. For some machines, there is only one cycle setting and for others several.

In Table 2.2 we can see the quality variables provided by the machines and the data pipeline at SNP&SEQ. These measurements are derived from the output of the machines. We list what level they are measured on and a very brief description of them. Tag level measurements can be aggregated, by the mean or sum, to receive their respective variable measurements found on read level. This is true to the first decimal. The completed run cycles are not equal to the actual setting which was used. We will therefore make the following assumption. A run which has a specific number of completed run cycles had a cycle setting equal to the largest closest possible run setting. Under this assumption we can deduce what type of cycle setting the run was performed with.

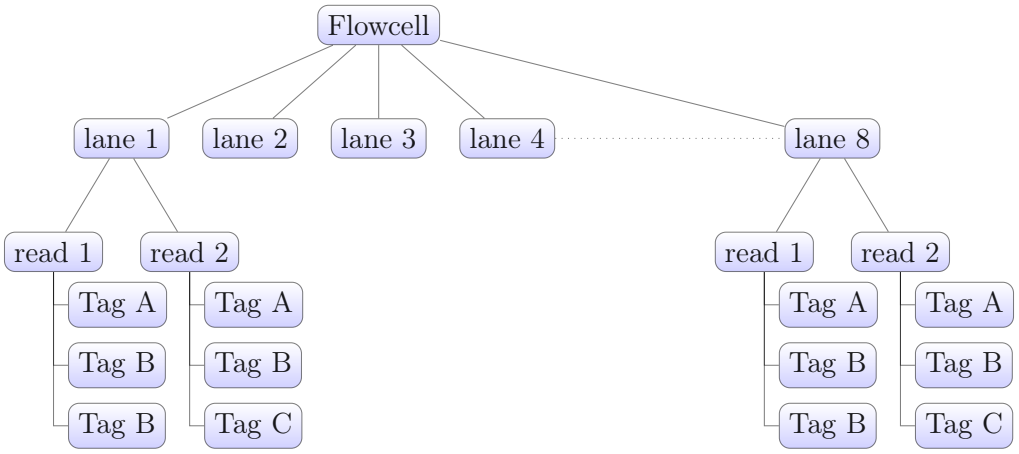

Figure 2.1: The hierachical structure of data at the lowest level, tag level, of quality measurement.

There are a total of 10 Next Generation Sequencing (NGS) machines of three types (MiSeq, HiSeq2500 and HiSeqX) at the SNP&SEQ platform. The HiSeqX machines are all of the same model whereas the HiSeq2500 are mixed, upgraded from previous models amongst others. These upgrades do not imply that the machines are equal in terms of their specifications. The three different types of machines are generally used for different things. The single MiSeq machine is mainly used for experimental samples. This machine provides less data and the cost of a run is less compared to the others. The MiSeq machine is expected to perform worse since a large portion of variation will come from the sample itself. The HiSeq machines are used for common

samples and are expected to perform better than MiSeq. They cost more to run but has the ability to produce larger output. Lastly, the HiSeqX machine is the least flexibile machine which also is the most expensive to run. Although being the least flexible and the most expensive one, a HiSeqX mechine is able to provide the largest amount of data and is deemed to be the most accurate.

Before the NGS machines are used they are cleaned. If a machine is idle for too many days the machine is put through the same maintenance to ensure that it does not deteriorate. It should be noted that the maintenance performed only covers a certain set of parts in the machine, such as drain pipes and pumps amongst others.

Table 2.2: Table containing quality variables, what level they are measured on and a short description of them.

Variable	Level	Description
Mean Q	tag/read	The mean quality score of a read
Completed cycles	tag/read	Number of completed cycles
Percent Q30	tag/read	Percentage of base calls which had a Q-value which where over 30
Error rate	read	Error rate of read sequence compared to a reference genome
Percent tag error	tag	The percent of error of the alignment of this tag
Raw cluster	read	The number of cluster's detected in a read
Post filter Cluster	tag/read	The number of cluster's detected, post filter
Raw Density	read	The cluster density
Post filter Density	read	The cluster density, post filter

In the next section we will introduce the methods to be used in this thesis.

Methods

In this section we will introduce the model, control charts and the change-point estimation procedure to be used in this thesis. We begin with introducing our problem, and how we will approach it from a mathematical point of view. Thereafter, we introduce the control charts and last the change-point estimation procedure.

3.1 Problem description

In Phase 1 we assumed that we have observed the target process, which we denote $\{\mathbf{Y}_t\}$. The target process represents what was called the IC state or IC behaviour of the process. It is assumed that realisations from this process are independent and identically distributed according to a p-variate normal distribution with mean vector $\boldsymbol{\mu}_0$ and non-singular covariance matrix $\boldsymbol{\Sigma}_0$, defined in section 2.3 Hair et al., (2006). In this thesis, the parameters are assumed to be unknown and will therefore be estimated using a random sample. Let $\hat{\boldsymbol{\mu}}_0$ and $\widehat{\boldsymbol{\Sigma}}_0$ denote the maximum likelihood estimate based on a random sample $\mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_M\}'$ from the target process. Their closed forms can be found in section 3.2 Hair et al., (2006).

In Phase 2 we consider a sequential p-dimensional process $\{\mathbf{X}_t\}$, which we will refer to as the observed process. If the observed process were to coincide with the target process, then we will say that the process is IC. On the contrary, if they do not share the same distribution we refer to the process as being OC.

We will consider two types of changes, transient and persistent changes. Transient changes are those where the observed process shows out of control behaviour but only for a specific observation. The observed process then goes back to the target process. Persistent changes are defined as whenever the observed process departs from the target process and do not return to it. The problem of discovering persistent changes can be placed in a change-point framework (cf. Chen and Gupta, (2011)), i.e.

$$X_t \sim \begin{cases} \mathcal{N}_p(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0), & t < \tau \\ \mathcal{N}_p(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), & t \geq \tau \end{cases} \tag{3.1}$$

where $\boldsymbol{\mu}_0$, $\boldsymbol{\mu}_1$ are two p-dimensional mean vectors and $\boldsymbol{\Sigma}_0$, $\boldsymbol{\Sigma}_1$ are two non-singular covariance matrices. If $\tau < \infty$ then a change occurred at time τ. Thus, up until time τ the observed process coincides with the target process. In the change-point framework, we aim to test *if* as well as *when* a change has occurred.

Transient changes will only be considered in the mean. For persistent changes, we will consider shifts in the location, i.e. $\boldsymbol{\mu}_0 \neq \boldsymbol{\mu}_1$, and also changes in the covariance matrix of the distribution i.e. $\boldsymbol{\Sigma}_0 \neq \boldsymbol{\Sigma}_1$.

3.2 Statistical process control - SPC

In this section we will introduce the control charts to be used in this thesis. These are Hotelling's T^2 chart (cf. Hotelling, (1947)) and Croiser's multivariate cumulative sum (MCUSUM) chart (cf.

Croiser, (1988)). The charts provide different characteristics and possibilities to detect different behaviour. We will use Hotelling's T^2 chart to monitor transient and large changes and Croiser's MCUSUM to monitor persistent and small changes.

We will start with introducing Hotelling's T^2 statistic for Phase 2 monitoring. Initially, Hotelling's T^2 statistic was derived as the generalisation of the univariate one-sample t-test to the multivariate setting where he was also first to use the T^2 statistic in multivariate statistical process control (cf. Hotelling, (1947)).

3.2.1 Hotelling's T^2 control chart

In Phase 2 monitoring we observe $\mathbf{X}_t$ one at a time, in a sequential manner. As presented by Qiu, (2013) in section 7.2.2, Hotelling's T^2 statistic for Phase 2 monitoring uses the following charting statistic

$$T_t^2 = (\mathbf{X}_t - \hat{\boldsymbol{\mu}}_0)' \widehat{\boldsymbol{\Sigma}}_0^{-1} (\mathbf{X}_t - \hat{\boldsymbol{\mu}}_0).$$

where $\hat{\boldsymbol{\mu}}_0$ and $\widehat{\boldsymbol{\Sigma}}_0$ are the maximum likelihood estimates based on the IC sample $\mathbf{Y}$. The charting statistic T_t^2 can be interpreted as the squared Mahalanobis distance (cf. Mitchell and Krzanowski, (1985)) of $\mathbf{X}_t$ to the estimated IC mean $\hat{\boldsymbol{\mu}}_0$ with respect to the estimated IC covariance matrix $\widehat{\boldsymbol{\Sigma}}_0$.

Under the assumption that the observed process is IC, $\mathbf{X}_t$ follows the target process. From Hair et al., (2006), Collary 5.2.1, we have that

$$\frac{(M-p)M}{p(M-1)(M+1)} T_t^2 \sim \mathcal{F}_{p,M-p}$$

where $\mathcal{F}_{a,b}$ is the F-distribution with parameters a and b, M is the sample size of the in-control sample and p is the number of elements in the target process. Under the assumption that the target distribution is multivariate normal the distribution of the T_t^2 statistic is exact. If the assumption is violated the F-distribution is still the asymptotic distribution, which makes Hotelling's T^2 statistic robust against distributional assumptions (cf. Kariya, (1981)).

The control limit h is calculated according to

$$h = \frac{p(M-1)(M+1)}{(M-p)M} \mathcal{F}_{1-\alpha,p,(M-p)} \tag{3.2}$$

where $\mathcal{F}_{1-\alpha,p,(M-p)}$ is the $(1-\alpha)\%$ percentile of the $\mathcal{F}_{p,M-p}$ distribution. If $T_t^2 > h$ we signal a alarm and state that the process has gone OC.

In the following section we introduce the CUSUM chart by Page, (1954) in the univariate setting and continue on to the multivariate CUSUM chart by Croiser, (1988).

3.2.2 The cumulative sum (CUSUM) chart

The univariate CUSUM chart was originally presented by Page, (1954). Page constructed the CUSUM chart based on what is called the sequential probability ratio test (SPRT). We start with presenting some theory for the ordinary hypothesis testing and then extend it to the SPRT framework. We then introduce the CUSUM chart and show how it is connected to SPRT.

Let Z denote a continuous random variable distributed according to some distribution $\mathcal{F}$. Let $\mathcal{F}$ have probability density function $f_{\mathcal{F}}$ and z be a realisation from this distribution. As presented by Siegmund, (1985) in chapter 1, the regular framework of hypothesis testing considers the null hypothesis together with the alternative in the following fashion

$H_0 : Z \sim \mathcal{F}$
$H_1 : Z \sim \mathcal{G}$

where $\mathcal{G}$ is some distribution with density $f_{\mathcal{G}}(z)$. The two distributions $\mathcal{F}$ and $\mathcal{G}$ are known and therefore the hypothesis can be tested using a likelihood ratio test. Let

$$\lambda(z) = f_{\mathcal{G}}(z)/f_{\mathcal{F}}(z)$$

be the likelihood ratio. Using the single observation z the likelihood ratio can be computed. The value of the likelihood ratio is then compared to a constant r_1. If the likelihood ratio, $\lambda(z)$, is larger then the constant r_1 we reject the null hypothesis. If it is less than r_1 we fail to reject the null. A sequential probability ratio test introduces a third possibility for intermediate values of $\lambda(z)$. That is, for $r_0 < \lambda(z) < r_1$ where $r_0 < r_1$, we can neither reject nor fail to reject the null. Intermediate values indicate that we need more information and should therefore continue to observe, or gather, more observations from the process.

Now, let z_t be realisations of the random variable Z at time point t. These realisations are obtained in a sequential order at regular intervals. Page, (1954) constructed the following charting statistic

$$C_t = \max(C_{t-1} + z_t, 0) \quad C_0 = 0, \tag{3.3}$$

in order to detect an increase in the mean of the distribution of Z. From equation (3.3), $C_t = 0$ when $C_t < \min_{0 \leq i < t} C_i$. Whenever the sequence C_t receives a new minimum, the process resets and starts again from zero. The charting statistic gives a signal of an increase in the mean if $C_t > h$ where h is a pre-specified control limit.

To see the connection to the SPRT framework we follow Qiu, (2013), section 4.2.4. Assume that Z follows a normal distribution with mean μ and variance σ^2. We are interested in testing the hypothesis

$$H_0 : Z \sim \mathcal{N}(\mu_0, \sigma^2)$$
$$H_1 : Z \sim \mathcal{N}(\mu_1, \sigma^2)$$

or in more compact form $H_0 : \mu = \mu_0$ against $H_1 : \mu = \mu_1$ where $\mu_0 < \mu_1$. Let $\mathbf{z}_t = \{z_1, z_2, ..., z_t\}$ represent a random sample of size t. The log likelihood ratio can then be written as

$$\begin{aligned}
\log(\lambda(\mathbf{z}_t)) &= \log(f_1(\mathbf{z}_t)/f_0(\mathbf{z}_t)) \\
&= \log\left(\prod_{i=1}^{t} f_1(z_i)/f_0(z_i)\right) \\
&= \sum_{i=1}^{t} \log(f_1(z_i)) - \log(f_0(z_i))
\end{aligned} \tag{3.4}$$

where the sub-index of the densities f_i, $i = 0, 1$, refers to the distribution under the null and alternative hypothesis, respectively. Using the density of the normal distribution in equation (3.4) we have that

$$\begin{aligned}
\log(\lambda(\mathbf{z}_t)) &= \sum_{i=1}^{t} \left(-\frac{(z_i - \mu_1)^2}{2\sigma^2} + \frac{(z_i - \mu_0)^2}{2\sigma^2}\right) \\
&= \frac{1}{\sigma^2} \sum_{i=1}^{t} \left(z_i\mu_1 - z_i\mu_0 - \frac{(\mu_1^2 - \mu_0^2)}{2}\right) \\
&= \frac{1}{\sigma^2} \sum_{i=1}^{t} \left(z_i(\mu_1 - \mu_0) - \frac{(\mu_1 + \mu_0)(\mu_1 - \mu_0)}{2}\right) \\
&= \frac{\mu_1 - \mu_0}{\sigma^2} \sum_{i=1}^{t} (z_i - \mu_0 - k)
\end{aligned} \tag{3.5}$$

where $k = (\mu_1 - \mu_0)/2$. Let

$$\begin{aligned}\tilde{C}_t &= \frac{\sigma^2}{\mu_1 - \mu_0} \log(\lambda(\mathbf{z}_t)) \\ &= \sum_{i=1}^{t} (z_i - \mu_0 - k) \qquad (3.6) \\ &= \tilde{C}_{t-1} + (z_t - \mu_0) - k \qquad (3.7)\end{aligned}$$

where $\tilde{C}_0 = 0$. If we compare the statistic C_t, equation (3.3), to $\tilde{C}_t$, seen in (3.7), the main difference can be seen in the maximum and the addition of k. The maximum can be interpreted as Page's CUSUM chart will never fail to reject the null hypothesis. The CUSUM chart only considers the two options of rejecting the null or what was described as the third option, we need more information.

A natural extension of the chart suggested in equation (3.3), under the assumption of normally distributed data, would be to include the constant k, which is referred to as the allowance constant (cf. Qiu, (2013) section 4.2.2). This implies the following charting statistic

$$C_t = \max(C_{t-1} + (z_t - \mu_0) - k, 0). \qquad (3.8)$$

Moustakides, (1986) showed that Page's CUSUM chart is quickest out of all SPRT tests to detect persistent shifts of size $\mu_1 - \mu_0 = 2k$, given an average run length (ARL) and a allowance constant k. The average run length will be further explained in section 3.2.3. The size of the shifts are seldom known beforehand which implies that k should be chosen such that we detect a *desirable* shift as soon as possible.

The CUSUM chart was first extended by Croiser, (1988) to the multivariate setting from a two-sided univariate chart he introduced in Croiser, (1986). We will now introduce Croiser's multivariate CUSUM chart presented in 1988.

3.2.2.1 Croiser's multivariate CUSUM chart

The natural and somewhat blunt extension of the univariate CUSUM chart, equation (3.8), to the multivariate setting would be to include vector variables in the scheme, i.e.

$$\mathbf{S}_t = \max(\mathbf{S}_{t-1} + (\mathbf{X}_t - \boldsymbol{\mu}_0) - \mathbf{k}, 0). \qquad (3.9)$$

where $\mathbf{X}_t \sim \mathcal{N}_p(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$. However, deducing which is the largest, a vector or $\mathbf{0}$, is not trivial nor is it clear how to choose the column vector of allowance constants $\mathbf{k}$. Therefore, Croiser, (1988) suggested the following. Consider the vector $\mathbf{k}$, it must have the same direction as $\mathbf{S}_{t-1} + (\mathbf{X}_t - \boldsymbol{\mu}_0)$, or otherwise increasing elements of $\mathbf{k}$ would not shrink $\mathbf{S}_{t-1} + (\mathbf{X}_t - \boldsymbol{\mu}_0) - \mathbf{k}$ towards the zero vector. Also, Croiser suggested that $\mathbf{k}$ should shrink $\mathbf{S}_{t-1} + (\mathbf{X}_t - \boldsymbol{\mu}_0)$ w.r.t the variance, thus he suggested, $(\mathbf{k}'\boldsymbol{\Sigma}_0\mathbf{k})^{1/2} = k$, it should have length k. Therefore Croiser set

$$\mathbf{k} = (k/C_t)(\mathbf{S}_{t-1} + (\mathbf{X}_t - \boldsymbol{\mu}_0))$$

given that $k < C_t$, where C_t is the length of $\mathbf{S}_{t-1} + (\mathbf{X}_t - \boldsymbol{\mu}_0)$ w.r.t. $\boldsymbol{\Sigma}_0$, i.e.

$$C_t = \sqrt{(\mathbf{S}_{t-1} + \mathbf{X}_t - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}_0^{-1}(\mathbf{S}_{t-1} + \mathbf{X}_t - \boldsymbol{\mu}_0)}$$

Now that we have constructed $\mathbf{k}$ in such a way that it will shrink the vector $\mathbf{S}_{t-1} + (\mathbf{X}_t - \boldsymbol{\mu}_0)$ towards the zero vector, the maximum taken in equation 3.9 can be seen as setting $\mathbf{S}_t = \mathbf{0}$ whenever $C_t \leq k$. Rather than considering the multivariate CUSUM in equation (3.9) we can

consider the following, let

$$C_t = \sqrt{(\mathbf{S}_{t-1} + \mathbf{X}_t - \boldsymbol{\mu}_0)' \boldsymbol{\Sigma}_0^{-1} (\mathbf{S}_{t-1} + \mathbf{X}_t - \boldsymbol{\mu}_0)} \tag{3.10}$$

$$\mathbf{S}_t = \begin{cases} \mathbf{0} & \text{if } C_t \leq k \\ (\mathbf{S}_{t-1} + \mathbf{X}_t - \boldsymbol{\mu}_0)(1 - k/C_t) & \text{otherwise} \end{cases} \tag{3.11}$$

where k is the allowance constant and $\mathbf{S}_0 = \mathbf{0}$. Let

$$H_t = \sqrt{\mathbf{S}_t' \boldsymbol{\Sigma}_0^{-1} \mathbf{S}_t}, \tag{3.12}$$

be the charting statistic. The chart gives a signal if $H_t > h$ where h is a pre-specified control limit.

Croiser, (1988) proved that the chart is directional invariant. It only depends on the non-centrality parameter

$$\lambda = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)' \boldsymbol{\Sigma}_0 (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

where $\boldsymbol{\mu}_1$ is the OC process mean vector and $\boldsymbol{\mu}_0$ the IC mean vector. The non-centrality can be interpreted as the statistical distance of the new mean $\boldsymbol{\mu}_1$ to the in control mean. Also, the result implies that we only need to use one chart to monitor all possible changes in the mean vector $\boldsymbol{\mu}_0$. The allowance constant k can be chosen in the same way as described in previous section. The choice of h is not trivial and will be extended upon more thoroughly in the section 3.2.3.

In the next section we will introduce one method for monitoring the covariance matrix, introduced in Bodnar et al., (2009). They constructed numerous charts for monitoring the covariance matrix based on properties of the singular Wishart distribution.

3.2.2.2 Monitoring the covariance matrix using properties of the singular Wishart distribution

Let $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n)'$ be a random sample of size n from $\mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma}_0)$. The dimensions of $\mathbf{X}$ is equal to $n \times p$, $n > p$. Let $\mathbf{V} = \mathbf{X}\mathbf{X}'$, then $\mathbf{V}$ follows a p-dimensional Wishart distribution with n degrees of freedom. The p-dimensional Wishart distribution, which we will denote $W_p(n, \boldsymbol{\Sigma}_0)$ has the following probability density function (cf. Forbes et al., (2011), chapter 47)

$$f(\mathbf{V}; \boldsymbol{\Sigma}_0) = \frac{\exp\left(-\frac{1}{2}\mathbf{tr}(\boldsymbol{\Sigma}_0^{-1}\mathbf{V})\right) |\mathbf{V}|^{(n-p-1)/2}}{\Gamma_p(n/2)|2\boldsymbol{\Sigma}_0|^{n/2}} \tag{3.13}$$

where $|\mathbf{A}|$ is the determinant of the matrix $\mathbf{A}$, $\mathbf{tr}(\mathbf{A})$ is the trace of the matrix $\mathbf{A}$ and $\Gamma_p(g)$ is the p-dimensional gamma function. Consider the case where $n < p$, then $\mathbf{V}$ is a rank deficient matrix since

$$\begin{aligned} &\text{rank}(\mathbf{X}) = \min(n, p) = n \\ &\text{rank}(\mathbf{V}) = \text{rank}(\mathbf{X}\mathbf{X}') \leq \min(\text{rank}(\mathbf{X}), \text{rank}(\mathbf{X}')) = n < p, \end{aligned}$$

by 3.12, Seber, (2008). By the definition of the rank (cf. definition 4.2 Seber, (2008)), the matrix $\mathbf{V}$ does not have an inverse and the determinant of the matrix is equal to zero. Under these circumstances, the density in equation (3.13) is zero for all outcomes $\mathbf{V}$. The distribution on the other hand, still exists and does so under the name of the singular Wishart distribution. The properties of the Wishart and singular Wishart distribution was thoroughly investigated in Bodnar and Okhrin, (2008). These properties were then placed in the multivariate SPC framework by Bodnar et al., 2009.

To monitor the covariance matrix, Bodnar et al., (2009) proposed the following. Let $n = 1$, then $\mathbf{X} = \mathbf{X}_t$, which is a single observation from the p-dimensional observed process at time

point t. Let $\mathbf{V}_t = \mathbf{X}_t\mathbf{X}_t'$ be the maximum likelihood estimate for the covariance matrix at time point t and partition the matrices $\mathbf{V}_t$ and $\mathbf{\Sigma}_0$ in the following way

$$\mathbf{V}_t = \begin{pmatrix} \mathbf{V}_{t;11} & \mathbf{V}_{t;12} \\ \mathbf{V}_{t;21} & \mathbf{V}_{t;22} \end{pmatrix} \qquad \mathbf{\Sigma}_0 = \begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{pmatrix}. \tag{3.14}$$

Note that we removed the subscript 0 when partitioning the covariance matrix. This was done in order to keep the notation readable. Consider the case where $\mathbf{V}_{t;12}$ and $\mathbf{\Sigma}_{12}$ are row vectors. For the i-th row and column we may reorder $\mathbf{\Sigma}_0$ and $\mathbf{V}_t$ such that the element $\mathbf{\Sigma}_{11}$ is the i-th diagonal element and $\mathbf{\Sigma}_{12}, \mathbf{\Sigma}_{21}$ the i-th row and column. Let σ_{ii}^2 and $\nu_{t,ii}^2$ be the i-th diagonal elements of the covariance matrix $\mathbf{\Sigma}_0$ and the matrix $\mathbf{V}_t$, respectively. Let $\mathbf{\Sigma}_{21,i}$ and $\mathbf{V}_{t;21,i}$ denote the i-th column of $\mathbf{\Sigma}_0$ and $\mathbf{V}_t$ but without their respective i-th element diagonal element. Let $\mathbf{\Sigma}_{22,-i}$ denote the $(p-1)\times(p-1)$ matrix without the i-th column and row of $\mathbf{\Sigma}_0$. The Schur complement (cf. Seber, (2008) definition 14.1) for the i-th row is defined as

$$\mathbf{\Sigma}_{22,-i}^* = \mathbf{\Sigma}_{22,-i} - \mathbf{\Sigma}_{21,i}\mathbf{\Sigma}_{21,i}'/\sigma_{ii}^2.$$

Partition the out-of-control covariance matrix $\mathbf{\Sigma}_1$ in equation (3.1) in the same manner as above. Let $\mathbf{\Sigma}_{1;22,-i}^*$ be the Schur complement for the i-th row of the out-of-control covariance matrix $\mathbf{\Sigma}_1$. The following theorem was displayed in Bodnar et al., (2009)

Theorem 1. *Let $\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_n$ be an i.i.d. p dimensional Gaussian process with $\mathbf{Y}_i \sim \mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma}_0)$. Let observed process $\{X_t\}$ be defined as*

$$X_t \sim \begin{cases} \mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma}_0), & t < \tau \\ \mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma}_1), & t \geq \tau. \end{cases} \tag{3.15}$$

Then

(a) in the IC state

$$\boldsymbol{\eta}_{i,t} = {\mathbf{\Sigma}_{22,-i}^*}^{-1/2} \left(\mathbf{V}_{t;21,i}/\nu_{t,ii}^2 - \mathbf{\Sigma}_{21,i}/\sigma_{ii}^2 \right) \nu_{t,ii} \sim \mathcal{N}_{p-1}(\mathbf{0}_{p-1}, \mathbf{I}_{p-1}). \tag{3.16}$$

where $\mathbf{0}_{p-1}$ is the zero vector of length $p-1$ and $\mathbf{I}_{p-1}$ is the $p-1$ dimensional identity matrix.

(b) in the OC state

$$E[\boldsymbol{\eta}_{i,t}] = (\mathbf{\Sigma}_{22,-i}^*)^{-1/2} \Omega_i \sigma_{ii} \frac{\sqrt{2}}{\sqrt{\pi}} \tag{3.17}$$

$$Var(\boldsymbol{\eta}_{i,t}) = (\mathbf{\Sigma}_{22,-i}^*)^{-1/2} \left(\mathbf{\Sigma}_{1;22,i}^* + \Omega_i \sigma_{ii}^2 (1 - 2\pi^{-1}) \Omega_i' \right) (\mathbf{\Sigma}_{22,-i}^*)^{-1/2} \tag{3.18}$$

where $\Omega_i = \mathbf{\Sigma}_{1;21,i}/\sigma_{1;ii}^2 - \mathbf{\Sigma}_{21,i}/\sigma_{ii}^2$.

Moreover, $\{\boldsymbol{\eta}_{i,t}\}$ are independent in the IC and OC state.

Proof of (a) was shown by Bodnar and Okhrin, (2008) and whereas part (b) was shown by Bodnar et al., (2009). The process $\{\eta_{i,t}\}$ is independent in time if the observed process $\{\mathbf{X}_t\}$ is.

If $\Omega_i = \mathbf{0}_{p-1}$ then no shift has occurred in the covariance matrix of the original observed process $\{\mathbf{X}_t\}$. A shift in the covariance matrix would imply a shift in the mean of the transformed quantity in equation (3.16). The result provides us with a way to monitor the covariance matrix with methods to monitor changes in the mean of a multivariate normal distribution. Also, if a shift in the mean vector would occur in the observed process in equation (3.15) the distribution of $\eta_{i,t}$, $i = 1, ..., p$ is no longer a $(p-1)$-variate normal distribution, shown by Bodnar et al., (2009). Any control chart which is constructed based on the process $\{\eta_{i,t},\ i = 1, ..., p\}$ will thus be sensitive to shifts in the covariance matrix *and* the mean vector of the initial observed process.

In order to monitor the whole covariance matrix we need to use p different charts. As suggested by Bodnar et al., (2009) we define the joint control chart, using Croiser's MCUSUM control chart as the foundation, as

$$C_{i,t} = \sqrt{(\mathbf{S}_{i,t-1} + \boldsymbol{\eta}_{i,t})'(\mathbf{S}_{i,t-1} + \boldsymbol{\eta}_{i,t})} \tag{3.19}$$

$$\mathbf{S}_{i,t} = \begin{cases} \mathbf{0} & \text{if } C_{i,t} \leq k \\ (\mathbf{S}_{i,t-1} + \boldsymbol{\eta}_{i,t})(1 - k/C_{i,t}) & \text{otherwise} \end{cases} \tag{3.20}$$

where k is the allowance constant and $\mathbf{S}_{i,0} = \mathbf{0}$. Let

$$H_{i,t} = \sqrt{\mathbf{S}'_{i,t}\mathbf{S}_{i,t}}, \tag{3.21}$$

for $i = 1, 2, 3, ..., p$. We define the charting statistic as

$$H_t = \max(H_{1,t}, H_{2,t}, ..., H_{p,t}).$$

We will now continue with specifying how to determine the control limit h and properly define the average run length.

3.2.3 Control limits and average run length

The IC average run length was first introduced by Page, (1954) together with the CUSUM chart. It is defined as the average number of observations we can observe in the sequential setting before the chart gives an alarm. The literature differentiate between the IC ARL (ARL_0) and the OC ARL (ARL_1) (cf. Qiu, (2013) or Mezzenga and Benassi, (2016)). The ARL_0 is defined as the average number of observations until the chart gives a alarm when the process is in control. This is closely related to the type 1 error in the regular hypothesis testing framework. The ARL_1 represents the average number of observations for the chart to discover a change which is actually present. It is closely related to the power of a test in the regular hypothesis testing framework.

Consider a random variable Z_t with continuous support which appears in a sequential order. In the case when Z_t are i.i.d for all t, we can see the events $Z_t > h$ as independent Bernoulli trials with probability of success α. One can define N_h as the number of independent Bernoulli trials it takes until a successful event, $Z_t > h$. For a finite value of h, the stopping time

$$N_h = \inf\{t \in \mathbb{Z}_+ : Z_t > h\}$$

is also a random variable, which follows a geometric distribution with probability of success α, described by Grimmett and Stirzaker, (2001) page 487. The expectation of N_h is by definition equal to the ARL_0. In the case of Hotellings T_t^2 charting statistic all are independent and therefore the disitribution holds of its ARL_1. However, in the CUSUM setting the charting statistics H_t are not independent which implies that the distribution of the stopping time is not geometric. The distribution of the ARL_0 was first investigated by Page, (1954) in the univariate setting through renewal equations. Generalizing the renewal equations into the multivariate setting becomes increasingly difficult as Croiser's MCUSUM control chart as it is not of them same form. While the calculations of the ARL_0 can not be done by closed forms it can be done through simulations.

From section 3.9 where we defined the MCUSUM control chart, we can see that the charting statstic H_t depends on the previous value of itself. This is famously known as the *markov property*, described by Resnick, (2002) page 63. Using a partition of the charting statistics support, one can describe this stochastic process as a markov chain, where the last state is an absorbing one. The ARL can be approximated using a markov chain approach, as described by Hawkins and Olwell, (1998) page 152. However, by Hawkins and Olwell, (1998) page 156, one can simply use

Monte Carlo simulation to construct trajectories of the the charting statistics, record the run lengths and take the average of these. The average of the individual run lengths serve as an estimate for the ARL_0. We will use this approach to calculate the ARL_0, i.e.

$$\text{ARL}_0 \approx \frac{1}{n}\sum_{i=1}^{n} N_{h,i} \tag{3.22}$$

where $N_{h,i}$ is the ith simulated run length under a fixed control limit h and allowance constant.

The Monte Carlo approximation of $\text{E}[N_h]$ using Croiser's MCUSUM chart is outlined in Algorithm 1. The monte carlo approximation is implemented in the functions `SimulateARL0` and `SimulateARL0Sigma` which are written in C++ together with OpenMP using the Rcpp extension. The Rcpp extension is presented by Eddelbuettel, (2013). It provides syntatic sugar for C++, extending some of `R`'s syntax into C++. OpenMP is a parallell programming model which is written for Fortran, C, C++ and is thoroughly described by Chandra et al., (2001) which can be used together with the Rcpp package by changing to a compiler which supports OpenMP. In the Appendix, section 7.3, we present a comparison of `SimulateARL0` performance inmplemented in base R, base R running in parallel using the **foreach** package and Rcpp using the OpenMP extension.

input : An allowance constant k, a control limit h, the in control parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$
output : A vector with Run lengths

Initialize: $t = 0$, $H = 0$, $\mathbf{S}_0 = \mathbf{0}$
while $H_t \leq h$ *and* t *is less than some large number* **do**
  Simulate $\mathbf{X}$ from $\mathcal{N}_p(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$.
  Calculate C_t with the help of $\mathbf{S}_{t-1}$ and $\mathbf{X}$, according to equation (3.10).
  Calculate $\mathbf{S}_t$ according to equation (3.11) and then calculate H_t.
  Update $t = t + 1$
end
Repeat n times.

Algorithm 1: Simulation of the IC average run length, given a control limit h and allowance constant k.

In equation (3.22) and Algorithm 1 we are assuming a fixed control limit and allowance constant. In order to find the optimal control limit given a target IC average run length, reffered to as ARL_0^*, and an allowance constant k, we consider the following function

$$f(h) = \text{E}[N_h] - ARL_0^* \approx \left(\frac{1}{n}\sum_{i=1}^{n} N_{i,h}\right) - ARL_0^*. \tag{3.23}$$

We aim to find the h^* which fulfils $f(h^*) = 0$. In this thesis, we will use the bisection algorithm described by Conte and Boor, (1980) page 75, to find h^* on a interval (a_0, b_0). In order for the bisection algorithm to converge we need to choose a pair a_0, b_0 such that $f(a_0)f(b_0) < 0$ which can be done by setting a_0 relatively small and b_0 sufficiently large. The choice of a_0 and b_0 will have a large impact on the time until convergence. The bisection algorithm is implemented in the R function `CalculateControlLimit` which makes use of the C++ functions previously described. An outline of the algorithm is presented in Algorithm 2.

In the next section we introduce the change-point estimation procedure. It will be used as a retrospective tool for diagnostics when the MCUSUM chart gives a indication of a change. The method to be used in this thesis is a change-point estimate based on the generalized likelihood ratio.

input : An allowance constant k, two endpoints in an interval $[a_0, b_0]$, a target ARL_0^*, a maximum number **Nmax** of iterations, a small number ϵ and IC parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$

output : Simulated ARL_0 for each iteration together with the interval used in that iteration step.

for $i \leftarrow 0$ **to** ***Nmax*** **do**
  Set $h_i = (a_i + b_i)/2$
  Use function `SimulateARL0` or `SimulateARL0Sigma` to simulate $\text{E}[N_h]$ given h_i, k, $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$. Let $ARL_0 = \left(\frac{1}{n}\sum_{i=1}^{n} N_{i,h}\right)$
  if $|ARL_0 - ARL_0^*| < \epsilon$ **then**
    break, convergence achieved.
  else if $ARL_0 < ARL_0^*$ **then**
    $a_{i+1} = h_i$
    $b_{i+1} = b_i$
  else
    $a_{i+1} = a_i$
    $b_{i+1} = h_i$
  end
end

Algorithm 2: Bisection algorithm used to find the control limit h

3.3 Change-point estimation using a generalized likelihood ratio

Consider the model defined in equation (3.1). Let $f(\cdot;\boldsymbol{\mu})$ denote the density of a multivariate normal distribution with mean vector $\boldsymbol{\mu}$. Consider the case where Croiser's MCUSUM control chart for the mean has given an alarm at the nth observation in Phase 2 monitoring. Let $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n)'$ denote the sample obtained in Phase 2 monitoring. Under the assumption of i.i.d observations, the joint density for the sample is the product of each individual density. However, since the Croiser's MCUSUM chart has given an indication of a change, we assume that the process, and therefore the distribution, has changed somewhere throughout the sample. To estimate the change-point τ we will consider a generalized likelihood ratio estimation procedure defined by Gombay and Horvath, (1994).

In the context of hypothesis testing, we would consider the following hypothesis

$$H_0 : \tau > n \;\text{ against }\; H_1 : \tau \leq n.$$

Let $l_i(\cdot;...)$, $i = 1, 2$, denote the likelihood of the density under the different hypothesis. The authors define the generalized likelihood ratio as

$$\begin{aligned}
\boldsymbol{\lambda}(\tau, \boldsymbol{\mu}_1; \mathbf{X}) &= \frac{l_1(\tau, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1; \mathbf{X})}{l_0(\boldsymbol{\mu}_0; \mathbf{X})} \\
&= \frac{f_1(\mathbf{X}; \tau, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1)}{f_0(\mathbf{X}; \boldsymbol{\mu}_0)} \\
&= \frac{\prod_{i=1}^{\tau-1} f_0(\mathbf{X}_i; \boldsymbol{\mu}_0) \prod_{i=\tau}^{n} f_1(\mathbf{X}_i; \boldsymbol{\mu}_1)}{\prod_{i=1}^{n} f_0(\mathbf{X}_i; \boldsymbol{\mu}_0)} \\
&= \prod_{i=\tau}^{n} \frac{f_1(\mathbf{X}_i; \boldsymbol{\mu}_1)}{f_0(\mathbf{X}_i; \boldsymbol{\mu}_0)} \\
&= \exp\left(-\frac{1}{2}\left(\sum_{i=\tau}^{n} (\mathbf{X}_i - \boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_0^{-1}(\mathbf{X}_i - \boldsymbol{\mu}_1) - (\mathbf{X}_i - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}_0^{-1}(\mathbf{X}_i - \boldsymbol{\mu}_0)\right)\right).
\end{aligned}$$

The generalized likelihood ratio depends on two parameters through the likelihood under the alternative hypothesis. In this context the OC mean $\boldsymbol{\mu}_1$ may be considered a nuisance parameter. The profile likelihood (cf. Sundberg, ("Statistical modelling by exponential families") page 234) of the likelihood under the alternative hypothesis can be attained by can by ordinary means since the score function exists, holding τ fixed. Using the profile likelihood instead of the likelihood we have that the generalized likelihood ratio is equal to

$$\begin{aligned}\boldsymbol{\lambda}_p(\tau;\mathbf{X}) &= \boldsymbol{\lambda}(\tau,\hat{\boldsymbol{\mu}}_1(\tau);\mathbf{X}) \\ &= \exp\left(-\frac{1}{2}\left(\sum_{i=\tau}^{n}(\mathbf{X}_i-\hat{\boldsymbol{\mu}}_1(\tau))'\boldsymbol{\Sigma}_0^{-1}(\mathbf{X}_i-\hat{\boldsymbol{\mu}}_1(\tau)) - (\mathbf{X}_i-\boldsymbol{\mu}_0)'\boldsymbol{\Sigma}_0^{-1}(\mathbf{X}_i-\boldsymbol{\mu}_0)\right)\right)\end{aligned}$$

where

$$\hat{\boldsymbol{\mu}}_1(\tau) = \frac{1}{n-(\tau+1)}\sum_{i=\tau}^{n}\mathbf{X}_i.$$

This profile likelihood ratio can be used to estimate the change-point τ. The estimator for τ based on the generalized likelihood ratio is defined as

$$\hat{\tau} = \arg\max_{1<j\leq n}\{\boldsymbol{\lambda}_p(j;\mathbf{X})\}.$$

The estimate is obtained by calculating all values of the generalized profile likelihood ratio and then picking the index j which results in the largest value of the likelihood ratio. This index j which supplied the largest value corresponds to the estimated change-point.

Exploratory data analysis

In this section we will conduct an exploratory data analysis of the two datasets at our desposal.

The data to be used consists of two sets. The first set contains observations on the lowest level, what we called the tag level. There is no fixed number of tags for each run and therefore each flowcell can contain a different number of measurements. This dataset contains a total of 786 runs (unique flowcells) from 2012 up until the end of 2015.

The second dataset contains observations from what we called the read level. This dataset contains a total of 801 runs. This implies that there is a difference between the datasets. A total of 15 runs are missing from the tag level. The missing runs are from the MiSeq 1, HiSeq 3 and 6 machines. These missing runs will be excluded from the data. Also, it was advised that data from 2012 was not to be used since runs performed in 2012 was done so under different circumstances. The quality control data from 2012 will be removed from both datasets.

In Table 4.1 we can see the completed run cycles for the HiSeq (Hi) and HiSeqX (HiX) machines. The different machines are labeled with different index. We can start by noticing that HiSeq 1 and 2 are not present in the table. These two have been taken out of production. The HiSeq machines show a large variance in the completed cycles. This is a consequence of the wide range of settings that have been used. We can see that HiSeq 6 (Hi6) have most runs in the vicinity of 124-125 completed cycles. A cycle setting of 126 is one of the most common cycle settings for HiSeq machines at the SNP&SEQ platform. We will use the HiSeq 6 machine to represent the HiSeq machines. The HiSeqX machines have all been run on the same cycle setting, with every completed cycle equal to 150. This is the only setting used at the SNP&SEQ platform for HiSeqX machines. As the HiSeqX machines do not differ in the cycle setting we will use the HiSeqX 1 to represent the HiSeqX machines. The MiSeq 1 machine has a wide range of 0-500 completed cycles with a lot of different run settings. It is not included since the table would be a page long but will be included, to some extent, in the exploratory analysis. All runs which have 0 completed cycles have been documented to be malfunctions.

The last row in Table 4.1 shows the total number of runs performed on each machine. The HiSeq 4 machine has most runs of all but also a large diversity in the run settings.

We will now investigate the Mean Q values of each successive run at a tag level. In Figure 4.1 we can see the mean of Mean Q tag level measurements together with the range (min to max) for three different machines of different types for lane 1 stratified on read. The observations are presented in their order of appearance.

For lane 1 measurements, the mean for HiSeq 6 of Mean Q tag level measurements correlates well with its range. If the range is large then the mean is usually lower. The variability of mean tag level measurements in read 2 is larger compared to read 1. HiSeqX is seen to have a small range in each run, for read 1 and 2 measurements. Read 2 measurements are lower on average but do not show any substaintial increase in variance. MiSeq 1 is seen to be the worst of all in terms of its Mean Q tag level measurements. It is clearly seen in the large variance of the means and the larger range throughout the runs. This is to be expected since the MiSeq machine was used for experimental samples on several different settings.

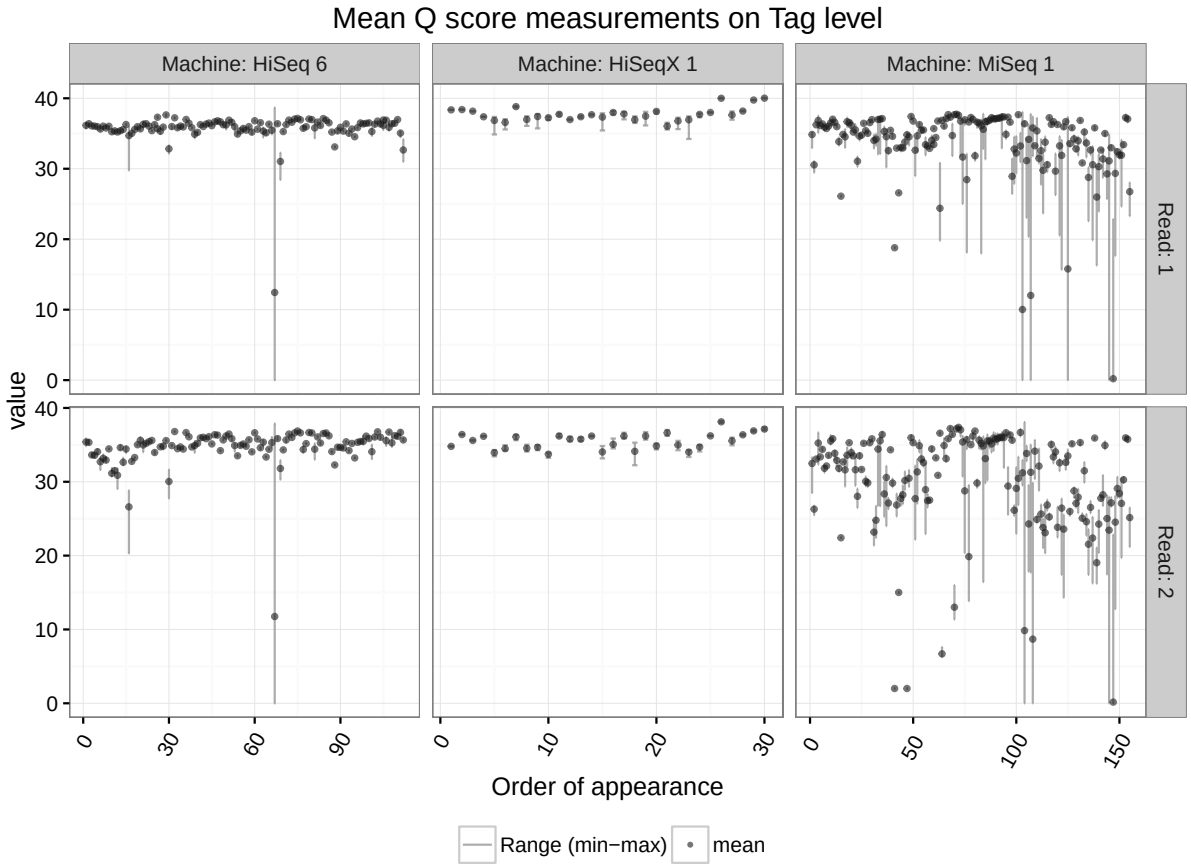

Figure 4.1: Figure containing the range (min to max) and mean of each successive run (flowcell). Here, we are showing read 1 and 2 in lane 1, disregarding what type of setting the run is performed on.

Table 4.1: Table showing the number of flowcells with a specific number of completed cycles for HiSeq (Hi) and HiSeqX (HiX) machine.

	Machine								
Cycles	Hi3	Hi4	Hi5	Hi6	HiX1	HiX2	HiX3	HiX4	HiX5
0	1	2		3		3		1	1
49	7								
50	16	10	11	7					
60				3					
99	2	4	2	1					
100	73	49	15	6					
124		22	22	30					
125		53	46	50					
150	11	2	14	16	30	27	22	32	16
200				1					
250	6	1							
Σ	115	143	110	117	30	30	22	33	17

We will now focus on the HiSeq machines with 102 to 126 completed cycles, in order to make this EDA sufficiently short.

In Figure 4.2 the range together with the mean of lane 1 and read 1 for HiSeq 6 is shown in their order of appearance. The variables shown here are Percent Q30 and the percent tag error. These measurements are from tag level for the HiSeq 6 machine with a cycle setting of 102 to 126. The last figure contains the number of observations in each lane 1 and read 1. The first figure with values of the Percent Q30 variable, shows an overall small amount of variation between runs. The range is very small, except for a single observation where the mean is lower compared to previous observations. The percent tag error can be seen to be very close to zero and at some times equal to zero. This is surprising since the construction of the variable is connected to the Error rate. If one is zero, the other should be as well. However, for *some* runs with zero percent tag error, the Error rate is well above zero. We will refrain from using the percent of tag error since the quality data can not be assured. The last figure in Figure 4.2 illustrates the number of observations contained in lane 1, read 1, in each successive run. This illustrates that the number of observations between runs does not need to be equal and that they vary a lot.

We will now continue with the read level measurements. At this level one observation per read and lane is supplied. We have 7 different variables, with 16 measurements in each. Since the HiSeq 6 machine has been our main interest so far, we will continue in this fashion and compare it to the other HiSeq machines. All runs will henceforth be using all 8 lanes and a cycle setting on 126. The HiSeq 3 machine does not have any runs on this specific cycle setting and will therefore be omitted.

Figure 4.3 shows the mean together with the range of the Error rate variable. We can see that no measurements are zero in this case. The HiSeq 5 machine seems to have a lower Error rate on average compared to the other machines. The HiSeq 6 machine has shown one, or possibly several, runs with large Error rates in different lanes. To further investigate the distribution of the Error rate together with those variables which have not been looked upon, we will look at them in a histogram.

In Figure 4.4 we have histograms of the Error rate, raw cluster density and the number of raw clusters for lane 1 and 2, both reads. Note that the scale of the variables differ. In the Error rate (row one), we can see that read 2 contains more variability compared to read 1. The distribution of read 1 is more peaked while the distribution for read 2 is quite flat. The density

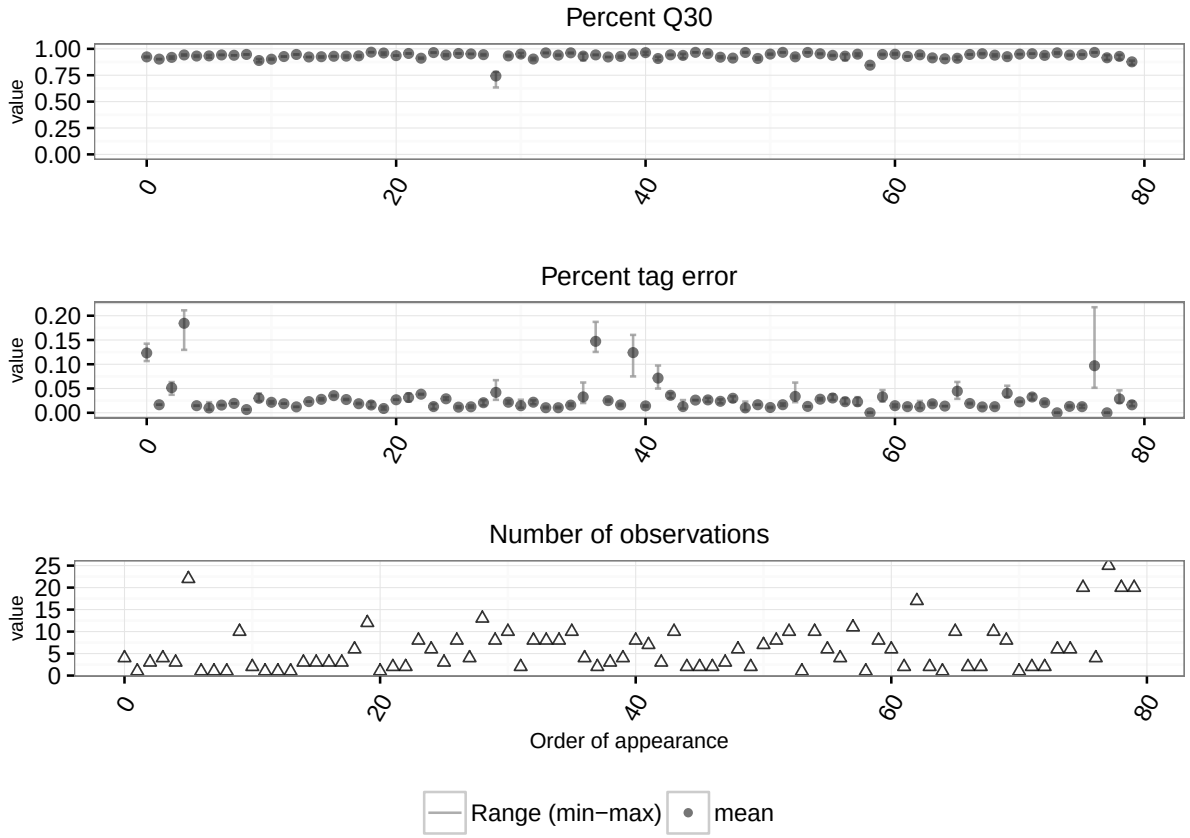

Figure 4.2: Figure showing the range (min to max) and mean of each succsessive run (flowcell) in lane 1, read 1, of the two variables Percent Q30 and Percentage tag error. All runs shown where performed on 126 cycles.

and cluster variable can be seen to be close to symmetric. The distribution of these two variables look very much alike.

The Spearman correlation matrix is visualized in Figure 4.5. The number of variables in the Figure is equal to 112. In this figure the axis labels where omitted but a header for each group of variables is placed next to them. As an example, the top 16 variables in Figure 4.5 corresponds to the Error rate for each read and lane which is denoted by the label. We will refer to this as a section of variables.

We can see that the density and cluster sections of variables correlate almost perfectly. This is especially true for measurements on the same read in a lane. The Mean Q and Percent Q30 sections seem to be correlated to each other while not having much correlation to the cluster and density variables. The Error rate is negatively correlated with Percent Q30 and Mean Q measurements from the same read and lane, while not showing much correlation to other reads and lanes. The correlation matrix can almost be placed on a block diagonal form where three first sections of variables create one block and the last four create another.

For further analysis we will consider the quality control data for HiSeq 6 with the variables; Mean Q, Error rate and Percent Q30. We will continue to use a cycle setting of 126. The Mean Q, Error rate can be assumed to have support on the positive real line. These variables can not be assumed to follow a normal distribution and will therefore be transformed. We will use a Box-Cox transformation (c.f. Box and Cox, (1964)) on these variables and estimate the transformation parameter λ using the Guerro method (cf. Guerrero, (1993)). Also, if necessary, we will divide the transformed variables by a constant to change the scale. The Percent Q30 variable has limited support on $(0, 1)$ and will be transformed using the quantile normal function.

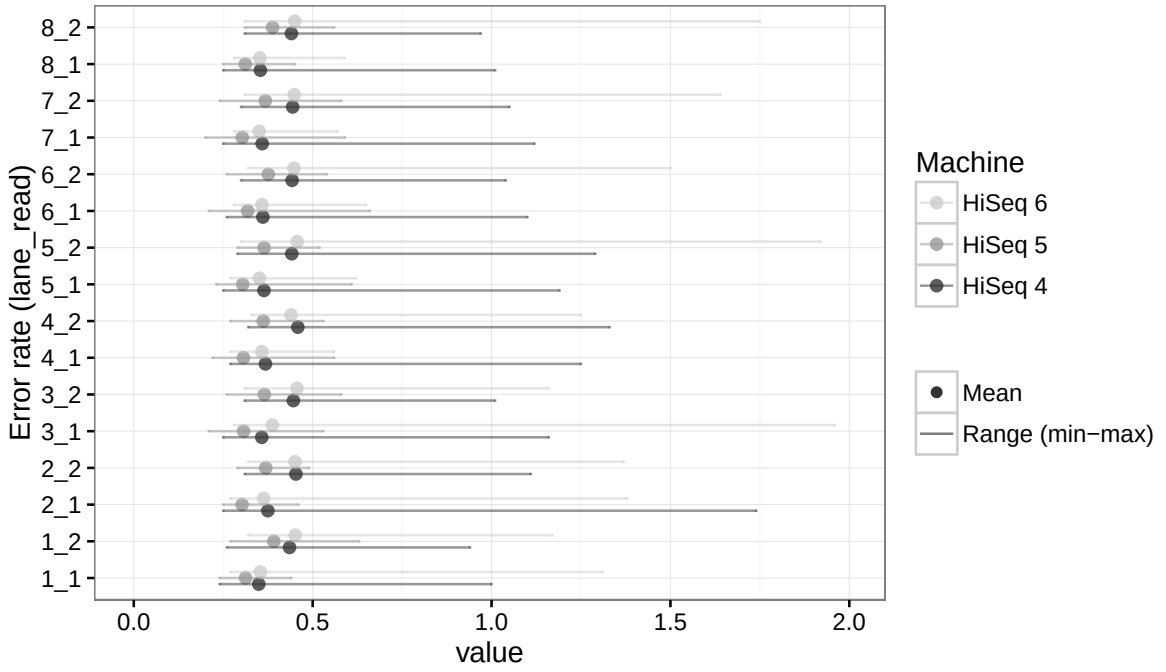

Figure 4.3: Mean together with the range of the Error rate of each lane and read (lane_read). Notice that the HiSeq 5 has the lowest mean Error rate of all HiSeq machines.

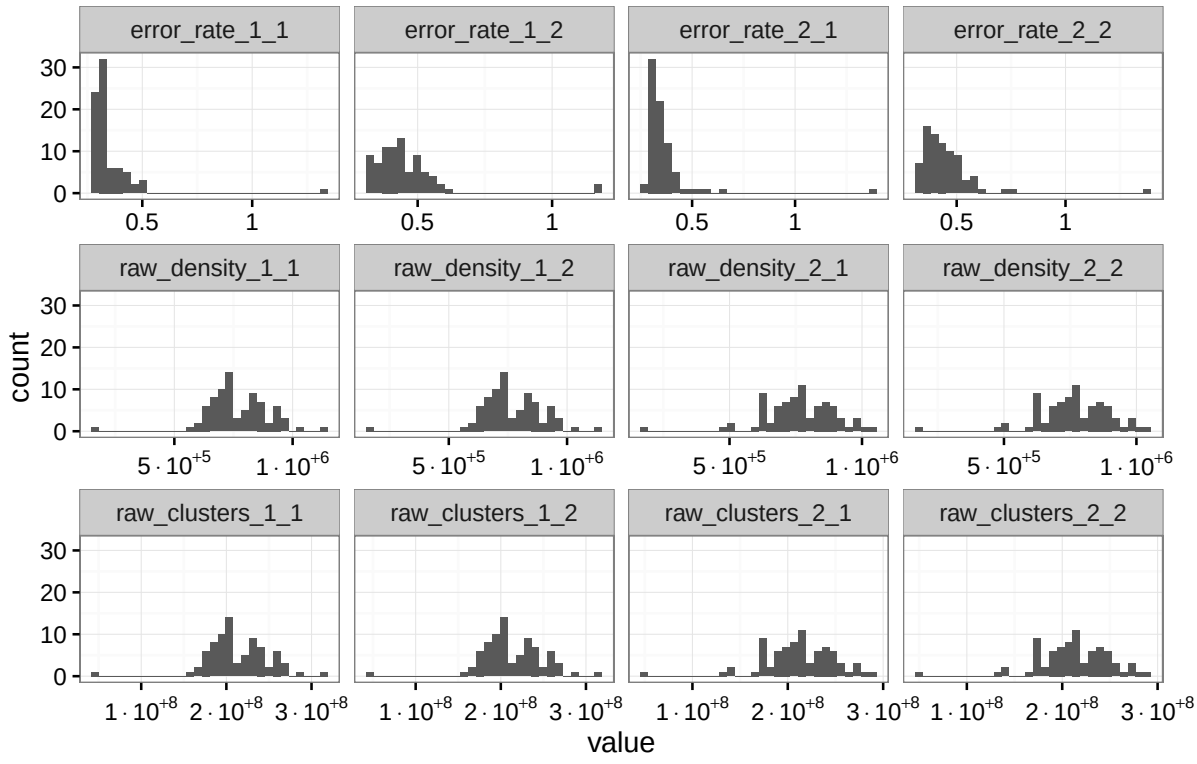

Figure 4.4: Error rates, the raw density and the number of raw clusters for each read in lanes 1 and 2. The variable name are listed in the following manner: Variable_lane_read.

Before the transformation and estimation of transformation parameters are performed we will remove those runs which are poor. A run will be classified as poor if it does not fulfill todays quality control criterias.

The transformation methods are more thoroughly presented in the Appendix, section 7.1. In this section, we also assess the assumption of normality for the transformed HiSeq 6 quality control data, for the variables previously mentioned, together with the a short investigation of autocorrelation. For further analysis, we assume that the transformed data of the Mean Q, Error rate and Percent Q30 variables are generated by a multivariate normal distribution.

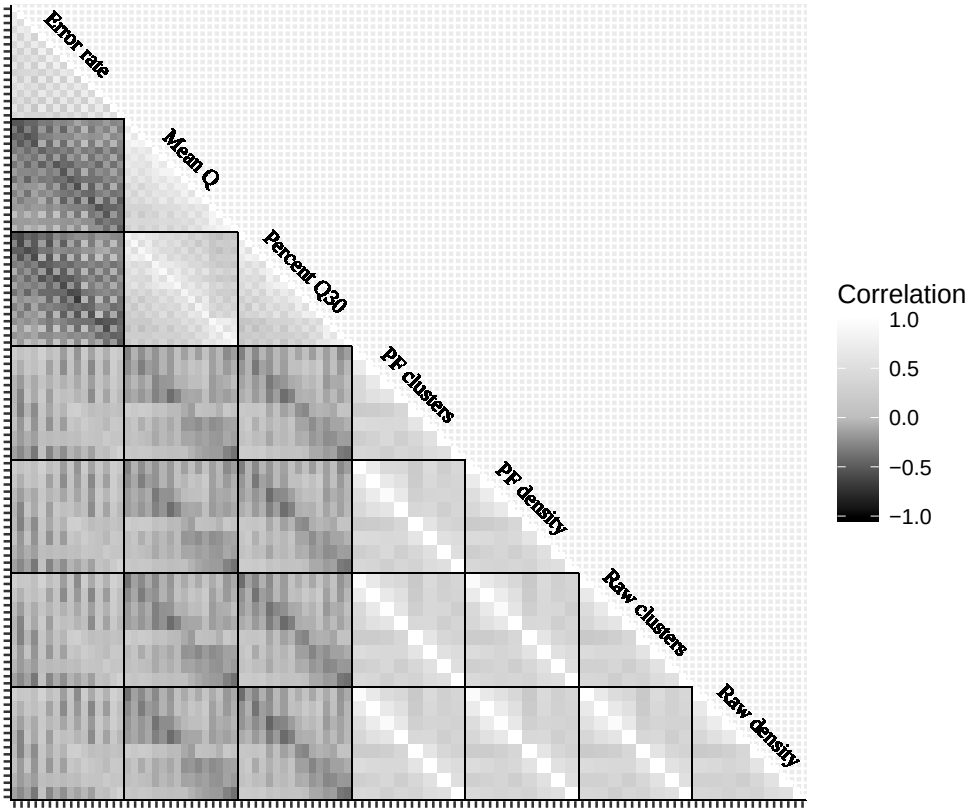

Figure 4.5: Spearman correlation matrix of HiSeq 6 Read level measurements.

Results

This section aims to show how the control charts perform in fictive scenarios and in practice. The first section will present the calculated control limits and what parameters were used to calculate these. We will introduce the performance measures to be used in three simulated scenarios of different characteristics. The last section includes an application on HiSeq quality control data. The control charts are constructed from transformed HiSeq 6 quality control data using the Mean Q, Error rate and Percent Q30 variables from each respective lane and read. The transformation methods are described in the Appendix, section 7.1.

5.1 Calculation of control limits

Using the Mean Q, Error rate and Percent Q30 variables from lane 1 to 8 with their respective measurement from read 1 and 2, the total number of variables is equal to $p = 48$. Using a cycle setting of 126 and excluding those observation that do not live up to todays quality control limits the number of observation of the IC sample is equal to $M = 73$. The IC parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ were estimated from the transformed IC sample.

The control limit for Hotelling's T^2 control chart was calculated using $\alpha = 0.01$ and is equal to 337.57. To calculate the control limits for the MCUSUM control charts we use Algorithm 2, described in section 3.2.3, which is implemented in the function `CalculateControlLimit`. In this thesis we will use a set of allowance constants k to see how the different control charts act with the use of different allowance constants. The following inputs was used in the calculations of the control limits.

- Allowance constants $k = \{0.30, 0.40, 0.50\}$.
- Target IC average run length $\text{ARL}_0^* = 100$.
- Maximum number of iterations Nmax= 40.
- The convergence error ϵ was set to 0.05.
- The number of simulations was set to 10^5

The constant a_0 was set sufficiently small in each simulation to ensure convergence, often close or equal to zero. The upper limit b_0 was tailored for each allowance constant, k. For $k = 0.3$, b_0 was chosen large, equal to 1000 or 10000 since no prior information on the control limit is available. In the next calculation with a larger allowance constant k, b_0 was chosen close to the calculated control limit in previous step with a smaller allowance constant. Table 5.1 displays the MCUSUM control limits for the mean and covairance control chart with the use of different allowance constants.

Table 5.1: The control limits calculated using the function `CalculateControlLimit` for a set of allowance constants k.

	k		
Type	0.3	0.4	0.5
Mean	2580.242	2100.029	1713.218
Covariance	382.812	226.562	116.547

5.2 Performance measures

In this section we present the performance measures to be used when evaluating the control charts and the change-point detection procedure in the simulation study.

5.2.1 Control charts

To evaluate the performance of the control charts, three measures will be used. The first measure will be used for transient changes. We will use proportion of discovered changes when the process has gone OC for one simulated observation. Since our MCUSUM control chart depends on a sequence of observations and not only the present we will supply a warm-up period before the transient change occurs. Thus, we will perform the following simulation. We will simulate a Phase 2 sequence of length k where the last observation is of OC nature. We will report the proportion of detected changes on the kth position, i.e.

$$\hat{P} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{H_k > h}$$

where $\mathbb{1}_{\cdot}$ is the indicator function, n is the number of simulations performed and H_k is the value of the charting statistic at observation k. The second and third performance measures will be used for persistent changes. It is the ARL_1, which is described as the time it takes until we discover a change which is present. It is defined in the same manner as the ARL_0, i.e.

$$\begin{aligned} N^* &= \inf\{t \in \mathbb{Z}_+ : H_t > h\} \\ \text{ARL}_1 &= \text{E}[N^*] = \text{E}[\inf\{t \in \mathbb{Z}_+ : H_t > h\}]. \end{aligned} \tag{5.1}$$

We have removed the subscript h and added a star to distinguish between the ARL_0 and ARL_1. In the case of persistent changes, we will set a max ARL_1 to 500. If any control chart does not indicate a change after our max ARL_1 in a OC scenario we will say that it failed to detect the change.

The ARL_1 assumes that the process goes OC as soon as we start to monitor it. To assume that a machine breaks as soon as Phase 2 monitoring begins may not be a very realistic case. As a third measure we will use the conditional expected delay (cf. Lai, 1995). It can be used to emulate scenarios where changes occur after some time. The conditional expected delay (ED) is defined as

$$\text{ED}_\tau(N^*) = \text{E}_\tau[N^* - \tau | N^* \geq \tau]$$

which allows for shifts at arbitrary times τ. In our simulations we will set $\tau = 20$.

All expectations will be approximated using monte carlo approximation with 10^5 repetitions.

5.2.2 Change-point estimation

The performance of the change-point estimation model will be evaluated using following performance measure

$$\bar{D} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{D}_i = \frac{1}{n}\sum_{i=1}^{n} (\hat{\tau}_i - \tau)$$

which we will call the average offset. It will give a indication on how our estimated change-point compares to the true value, on average. Since the change-point estimation assumed a fixed sample size we will perform the simulations in the following way:

- Simulate $\tau = 20$ observations from the target process.
- Simulate $\lfloor \mathrm{ED}_\tau[N^*] \rfloor$ number of observations for the given scenario, size of change and allowance constant.

Here $\lfloor \cdot \rfloor$ is the floor, i.e. the closest lowest integer of the conditional expected delay as the number of simulations for a given scenario and size of change. In the simulation of $\bar{D}$ we will use the result of simulated expected delay with an allowance constant equal to 0.3. Also, the suggested change-point procedure assumed that the change manifested in the mean and not in the covariance matrix. We will investigate how the change-point estimation procedure would detect the change-point if a change in the covariance matrix occured at the same time as the mean. Let $\bar{D}_\Delta$ be the simulated average offset under a change in the covariance matrix by $\boldsymbol{\Delta}$.

We will continue with describing the simulation study and defining the three different scenarios we will consider.

5.3 Simulation study

We will consider three different OC scenarios. The first scenario will consider transient changes whereas the later two will consider persistent changes. The first scenario will constitute of the following. A total of 20 observations will be simulated from the target process. The 21st observation will be of OC nature. This could be described as the process going OC because of a flowcell being processed poorly or perhaps a poor sample on the flowcell. We aim to investigate if the charts discover the 21st observation. In this scenario, we will not test the performance of the change point estimation procedure, as we assume that the process would go back to the target process afterwards.

The second scenario will emulate a broken lane, all measurements on this lane will persistently show worse behaviour. The third scenario considers persistently worse behaviour in the Error rate of lane 1, while every other variable performs as expected. We assume that these scenarios can manifest itself in the mean and the covariance matrix.

Let the observed process $\mathbf{X}_t$ be ordered in the following manner. The first three variables are the Mean Q, percent q30 and Error rate variables from the first lane and read. The second triplet of variables are from the first lane but the second read and so forth. Define the OC mean vector $\boldsymbol{\mu}_1$ and OC covariance matrix $\boldsymbol{\Sigma}_1$ in the following way

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 + \begin{pmatrix} -\delta_1 \\ -\delta_2 \\ \delta_3 \\ -\delta_1 \\ -\delta_2 \\ \delta_3 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_0 + \boldsymbol{\Delta} \tag{5.2}$$

where

$$\boldsymbol{\Delta} = \begin{pmatrix} \boldsymbol{\Delta}_1 & \mathbf{0}_{(p-6)\times(p-6)} \\ \mathbf{0}_{(p-6)\times(p-6)} & \mathbf{0}_{(p-6)\times(p-6)} \end{pmatrix}$$

where $\mathbf{0}_{k\times k}$ is a $k \times k$ matrix with all entries equal to zero. The submatrix $\boldsymbol{\Delta}_1$ have dimension 6×6. We will now continue simulations for scenario one.

5.3.1 Scenario 1 - Transient changes from poor samples on flowcells

In this scenario we consider transient changes. We will simulate Phase 2 sequence of length 21, i.e. 21 observations where the last is of out-of-control nature. We will assume that no change occur in the covariance matrix. We also assume that the poor sample will only manifest itself in the Mean Q measurements of the first lane. Let $\delta_1 > 0$, $\delta_2 = \delta_3 = 0$ and $\Delta_1 = \mathbf{0}_{6\times 6}$. The proportion of detected changes at the 21st observation over different values of δ is seen in Figure 5.1.

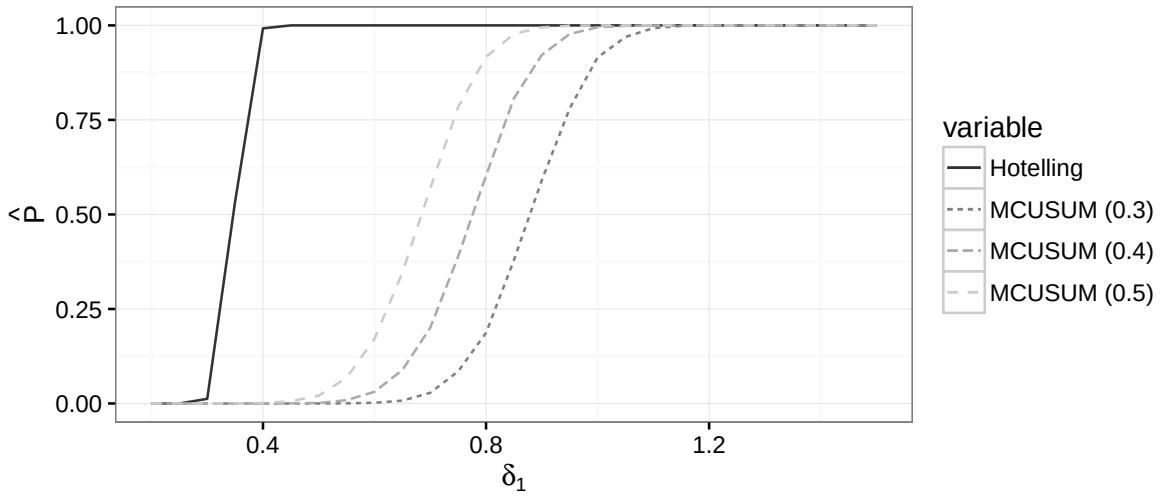

Figure 5.1: The proportion of detected changes at the 21st observation in scenario 1. Note that the legend for Croisers MCUSUM is on the following form MCUSUM (k).

In Figure 5.1 we can see that Hotelling's T^2 statistic is more proficient in detecting the transient change at the 21st observation for small values of δ_1 compared to Croiser's MCUSUM. Croiser's MCUSUM chart with a allowance constant of 0.5 detects the transient changes best among the MCUSUM charts.

We will continue with simulating scenario 2 where we assume that quality control data indicates persistently bad performance in lane 1.

5.3.2 Scenario 2 - All quality control variables in lane 1 show persistently poor behaviour

In this scenario the quality control data for lane 1 shows persistently worse behaviour which we assume emulates a broken lane. Let $\delta_i = \delta > 0$ for $i = 1, 2, 3$. We also assume that this change can manifest itself in the variance. Therefore, all off-diagonal elemets in $\boldsymbol{\Delta}_1$ are zero and the diagonal elements are equal to a constant $\Delta > 0$.

5.3.2.1 Simulation results of control charts

Figure 5.2 shows the ARL_1 and ED of the MCUSUM for different values of δ. The values of δ are small which is a result of the scale of the transformed data and also the number of components that are changing in the mean vector. The ARL_1 goes down quickly for increasing values of δ.

The ED is seen to decrease quicker than the ARL_1 for small values of δ. When δ becomes large, the two measures are almost equal in this simulation study. The optimal value of the allowance constant seem to be equal to 0.5 in both cases. The worst allowance constant switch between Hotelling's T^2 statistic showed no indication of a change, the smallest OC ARL_1 for a subset (evenly distributed) of the values of δ was equal to 500 for this simulated scenario.

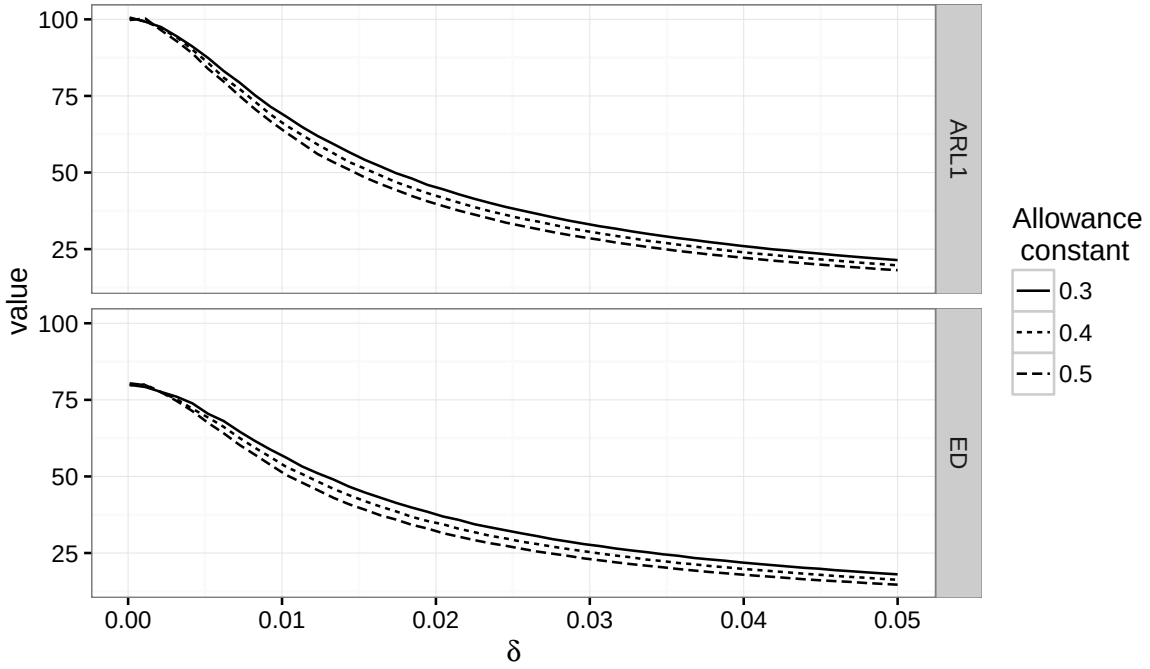

Figure 5.2: OC ARL and ED for the MCUSUM chart of Scenario 2 - changes in the mean of all variables of lane one.

In Table 5.2 we have the simulated ARL_1 and ED for persistent changes in the covariance matrix. For small changes in the variance of the covariance matrix it takes close to the 100 observations to discover a change. As Δ grows the MCUSUM chart detects changes faster when monitoring the covariance matrix. The ARL_1 and ED is in general smaller for an allowance constant equal to 0.5 compared to the results of 0.3 or 0.4. For this specific scenario, the allowance constant k seem to be optimal at 0.5 in the sense that it detects these changes the fastest.

Table 5.2: Scenario 2. MCUSUM simulated OC ARL and ED for changes in covariance matrix. Each cell is on the following form: ARL_1 (ED). The ARL_0 is equal to 100. 10^5 replications was used in this simulation study.

	Δ				
k	0.01	0.1325	0.255	0.3775	0.5
0.3	92.84 (72.00)	59.59 (43.69)	30.55 (23.19)	18.84 (11.49)	13.72 (6.84)
0.4	94.18 (72.23)	57.54 (45.90)	25.71 (21.32)	14.49 (8.97)	10.87 (4.44)
0.5	87.16 (76.87)	48.35 (39.75)	18.11 (12.80)	10.44 (3.47)	7.85 (1.10)

5.3.2.2 Simulation results of change-point estimation procedure

In Table 5.3 we can see the results from the simulation study of the change-point estimation procedure. In this table we have listed the floor of the simulated ED, that is how many observations was used after 20 in control observations in our simulation study. As an example, for a δ equal to 0.0001 we simulated 20 in control observations and 79 OC observations with the use of the OC mean vector. The second row shows the average offset, $\bar{D}$, with no change in the covariance matrix. We can see that for a small values of δ we heavily overestimate the change point. As δ grows we start to underestimate the change-point. Note that while it might take us one observation to discover a change, the estimated change-points differ. The third row shows the result of $\bar{D}_\Delta$, where $\Delta = 0.1325$. This type of change in the covariance matrix skewes the estimated change-point heavily!

Table 5.3: Table containing change-point estimation simulations for Scenario 2. 20 in control observations was used. The simulated ED originates from the use of a allowance constant of 0.3. Δ was set equal to 0.1325.

	δ									
	1e-04	0.0052	0.0103	0.0164	0.0215	0.0276	0.0327	0.0388	0.0439	0.05
$\lfloor ED \rfloor$	79.00	70.00	56.00	43.00	35.00	29.00	25.00	22.00	20.00	18.00
$\bar{D}$	48.74	36.83	18.97	6.26	1.90	0.20	-0.21	-0.30	-0.17	-0.11
$\bar{D}_\Delta$	-10.19	-10.87	-11.04	-11.37	-11.34	-11.36	-11.68	-11.27	-11.54	-11.24

5.3.3 Scenario 3 - The Error rate of lane 1 shows persistently poor behaviour

In this scenario we will investigate how the charts behave in a situation where only two variables in a lane is effected by some unknown change. In this case we assume that $\delta_1 = \delta_2 = 0$ and $\delta_3 > 0$. This implies that the Error rate increases on average, for lane 1 quality measurements. In the case of changes in the covariance matrix, we assume that the variance will increase and that covariance is held constant. All elements in the matrix Δ_1 are zero except for the third and sixth diagonal element.

5.3.3.1 Simulation results of control charts

In Figure 5.3 we can see the results for ARL_1 and ED. Note that the values of δ are larger than the values used in scenario 2. In this scenario the ARL_1 and ED decreases quickly for increasing values of δ. The optimal allowance constant is equal to 0.5 in this scenario. Hotelling's T^2 statistic did not show any indication of a change, the smallest out-of-control ARL for all δ was equal to 500.

In Table 5.4 the results for the ARL_1 and ED for simulated changes in the covariance matrix. Here, the value of the allowance constant is seen to have an impact on detecting changes in variance structure of the covariance matrix. The optimal choice of k is equal to 0.5 for this specific scenario.

5.3.3.2 Simulation results of change point estimation procedure

In Table 5.5 we can see the average offset for different shifts in scenario 3. For increasing values of δ the average offset $\bar{D}$ decreases, our change-point estimation procedure becomes more accurate on average. As previously seen in scenario 2, for the smallest value of δ the average offset is large, i.e. we overestimate the change-point. For all other values, the estimated change-point is quite accurate on average.

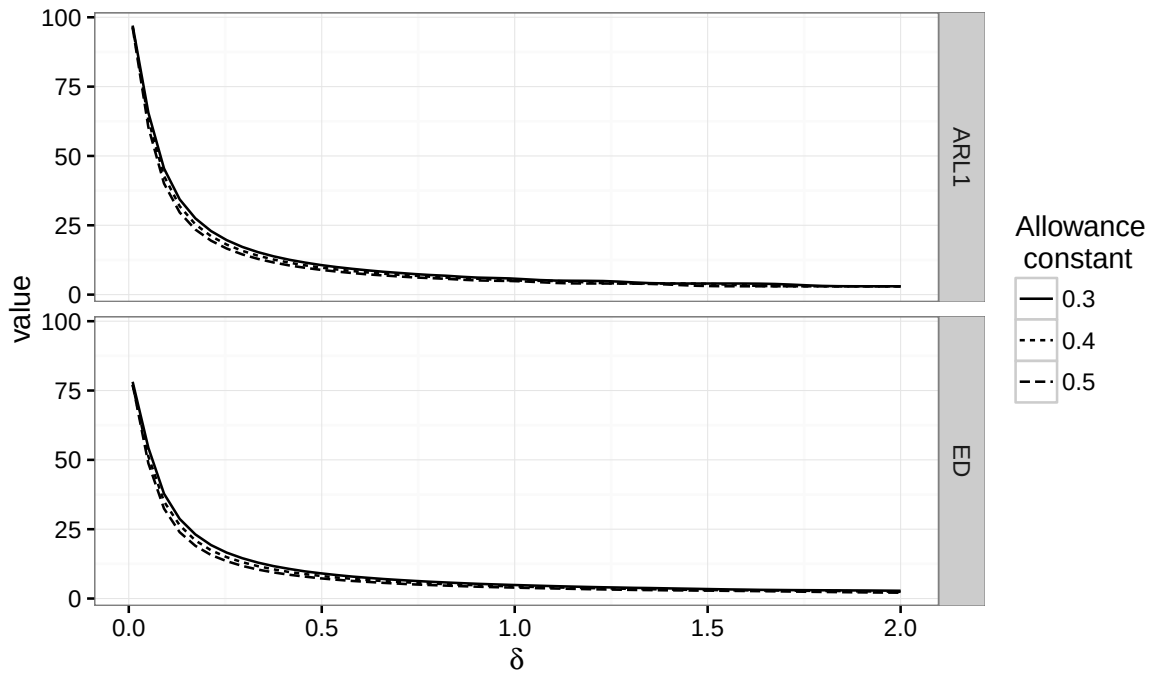

Figure 5.3: OC ARL simulations of scenario 3 - changes in the mean of the Error rate of lane one.

Table 5.4: Scenario 3. MCUSUM simulated OC ARL and ED for changes in covariance matrix. Each cell is on the following form: ARL_1 (ED). The ARL_0 is equal to 100. 10^5 replications was used in this simulation study.

	Δ				
k	0.05	0.6625	1.275	1.8875	2.5
0.3	94.45 (74.05)	70.15 (51.24)	49.53 (34.93)	35.46 (23.63)	26.87 (16.88)
0.4	95.46 (73.17)	68.24 (49.39)	46.28 (31.53)	30.98 (19.90)	22.21 (12.89)
0.5	88.29 (66.32)	58.54 (42.01)	37.56 (23.46)	22.31 (12.33)	15.44 (6.60)

In the third row we can see how the change in the covariance matrix impacts the average offset. In general, the increase in the variance of the Error rate by $\Delta = 0.6625$ causes the average offset to increase. In this simulated scenario the average offset is seen to increase when the covariance matrix change according to $\Delta = 0.6625$. This change is however quite small.

Table 5.5: Table containing change-point estimation simulations for scenario 3. 20 IC observations were used. The simulated ED originates from the use of an allowance constant of 0.3. In these simulations Δ was set to 0.6625.

	δ									
	0.01	0.2131	0.4161	0.6598	0.8629	1.1065	1.3096	1.5533	1.7563	2
$\lfloor ED \rfloor$	78.00	19.00	10.00	7.00	5.00	4.00	3.00	3.00	3.00	2.00
$\bar{D}$	47.16	-0.19	-0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$\bar{D}_\Delta$	52.56	8.65	3.18	1.40	0.73	0.39	0.23	0.14	0.07	0.04

5.4 An application on HiSeq quality control data.

In this section we will test the control charts, constructed from transformed quality control HiSeq 6. These will be tested on three other HiSeq machines, namely HiSeq 3, 4 and 5. First, we transform the quality control data from HiSeq 3, 4 and 5 using the Box-Cox transformation with the estimated parameters from the HiSeq 6 data. For variables with limited support, we use the quantile function of the normal distribution to transform the data. After the transformation we test the constructed control charts on the new data. Any alarm that is given will only be an indication that the IC parameters for HiSeq 6 does not fit the other machines. The HiSeq 3 machine has no runs on the same type of setting which we used for estimating our HiSeq 6 IC parameters. The data from the HiSeq 3 machine is performed on a mixture of settings. The HiSeq 4 and 5 machines have runs performed on the same setting as those on HiSeq 6.

We will use a allowance constant $k = 0.3$ for both MCUSUM charts. The control limits can be seen in Table 5.1 for the mean and covariance chart, respectively. With a $\alpha = 0.01$ Hotelling's T^2 control limit, defined in (3.2), was calculated to 337.57.

In Figure 5.4 we can see Hotelling's T^2 statistic of the transformed HiSeq 3, 4 and 5 quality control data. First, notice the difference in scale of the y-axis between the figures. The HiSeq 3 machine was used with different settings which can be a cause to what is seen in the left figure of Figure 5.4. Almost all observations are well above the control limit. The first run of HiSeq 3, where Hotellings T^2 shows a value of 2.071688×10^4, stands out in terms of great quality measurements in lane 7 while having very poor quality measurements in read 2, lane 5 and lane 8. The majority of Hotelling's T^2 statistics based on transformed HiSeq 4 and 5 quality control data are below the control limit.

The MCUSUM charts for the mean vector and covariance matrix are shown in Figure 5.5 for the HiSeq 4 and 5 machines. These were calculated according to the order of appearance. The first observation in the sequence is the oldest one and the last the most recent one. HiSeq 3 was removed since it did not have any runs on the same setting. Here we can see that all charts show evidence of a strong OC situation. They give a strong indication that the estimated IC mean vector and covariance matrix do not fit the transformed quality data of HiSeq 4 and 5 machines.

Assuming that the transformed quality control data for HiSeq 4 and 5 represents an IC sample from their respective processes, we can calculate the non-centrality parameter, described in section 3.2.2.1. Note that we are not removing any observations from the HiSeq 4 and 5 quality control data and that we are using the transformation parameter estimated from HiSeq 6 quality control data. Let $\hat{\boldsymbol{\mu}}_i$ be the maximum likelihood estimator of the mean vector based on the i-th HiSeq (transformed) quality control data. Under the assumption that they share the same covariance, the non-centrality parameter comparing IC mean between HiSeq 6 and HiSeq 4 can be calculated to

$$(\hat{\boldsymbol{\mu}}_0 - \hat{\boldsymbol{\mu}}_4)\widehat{\Sigma}_0(\hat{\boldsymbol{\mu}}_0 - \hat{\boldsymbol{\mu}}_4)' = 0.205.$$

The non-centrality parameter comparing HiSeq 6 and HiSeq 5 is calculated to 4.926. Under the

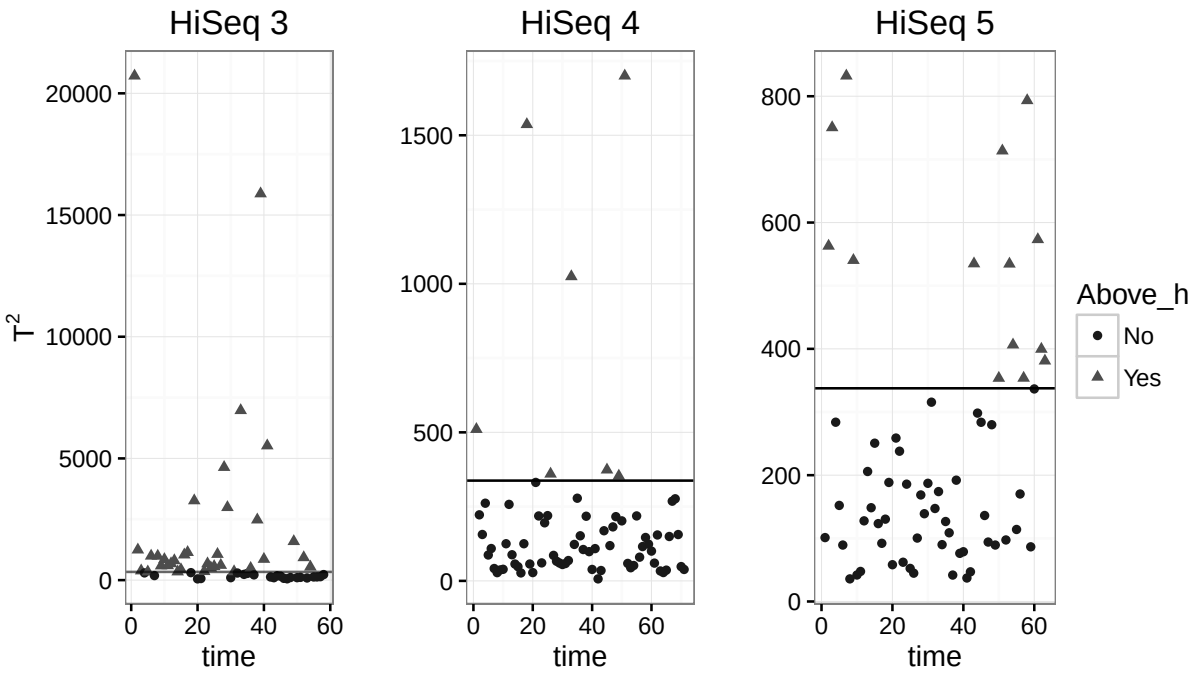

Figure 5.4: Hotelling's control chart for the HiSeq 3 (left), HiSeq 4 (middle) and HiSeq 5 machines with IC parameters based on HiSeq 6 data. The horizontal line represents the control limit.

assumption that the processes share the same covariance matrix, the means are distant from each other in the transformed space.

The control charts for the covariance showed a even stronger OC scenario compared to the mean. The covariance matrices can be compared using the determinant of the covariance matrices. It represents the squared volume of the parallelotope in $\mathcal{R}^p$ where the eigenvectors are the principal edges (cf. Hair et al., (2006, page 385)). The ratio of the determinants can serve as a measure of how the squared volume of the parallelotope relates to eachother. Let $\widehat{\Sigma}_i$ be the maximum likelihood estimator based on transformed quality control data from the i-th HiSeq machine. Let

$$R_i = \frac{|\widehat{\Sigma}_0|}{|\widehat{\Sigma}_i|}$$

be the ratio between the IC covariance matrix and the estimated covariance matrix based on the i-th HiSeq transformed quality control data. The ratio R_i for HiSeq 4 transformed quality control data is equal to $R_4 = 3.0064497 \times 10^{12}$ and for HiSeq 5 we have $R_5 = 5.5434293 \times 10^8$. In the transformed space, these covariance matrices are not equal in terms of their parallelotope volume. This will only give a illustration of whether or not their volume is equal. It does not take the inherent structure of the covariance matrix into account. Since the charts gave a indication of such a heavy OC scenario the change-point detection estimation procedures will not be used in this application.

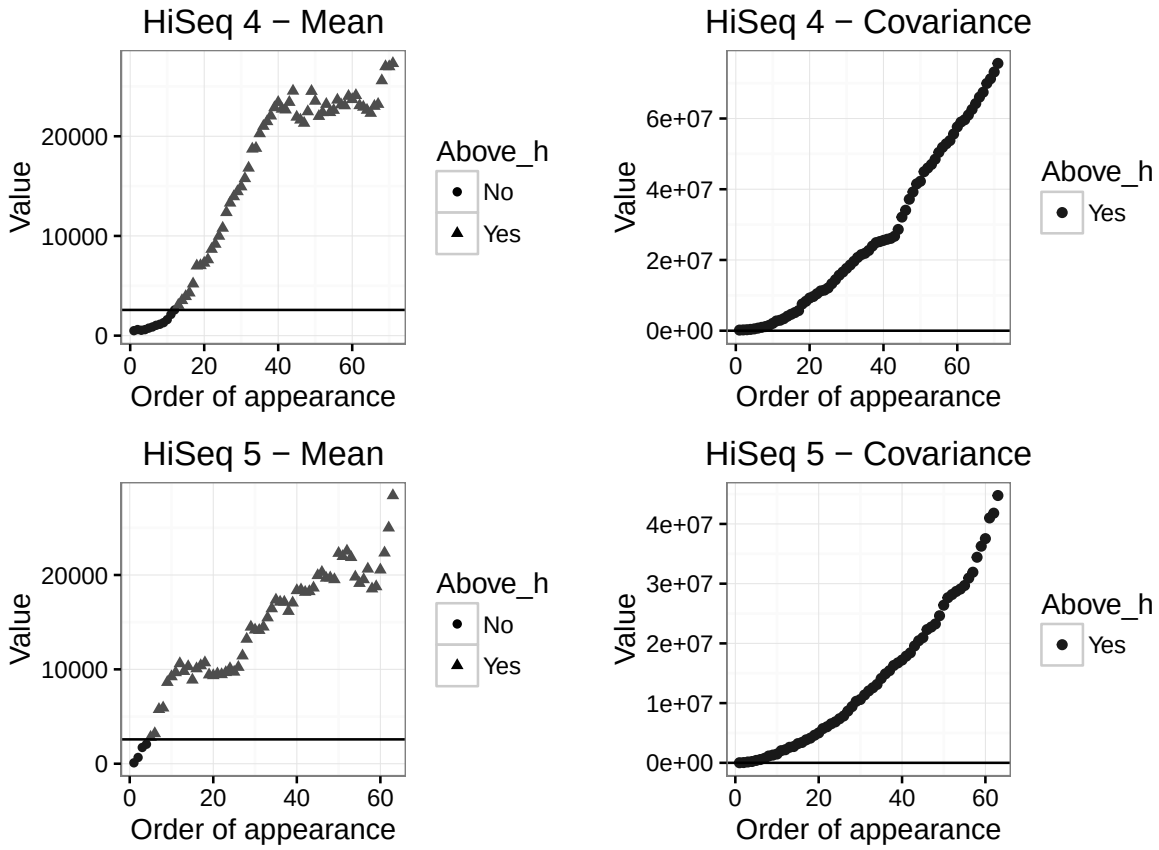

Figure 5.5: MCUSUM control charts monitoring the mean vector and covariance matrix. The IC parameters is estimated from transformed data from the HiSeq 6 machine. The horizontal line is the control limit calculated with k=0.3.

Discussion and conclusions

In this thesis we have investigated how statistical process control and a change-point estimation procedure can be used to detect and estimate changes in transformed next generation sequencing quality control data. The control charts presented in this thesis were applied on transformed HiSeq 6 quality control data. The performance of these control charts were tested in a simulation study. Three different scenarios were considered. The first scenario considered transient changes and compared Hotelling's T^2 control chart and Croiser's MCUSUM control chart ability to detect these changes. In the second scenario we investigated how quickly the control charts would detect a persistent small change in all variables in a lane. In the third scenario we tested how quick the control chart would detect a persistent change in a variable, read one and two, in a lane.

The results from the simulation study showed that Hotelling's control chart was proficient in detecting large and transient changes while poor in detecting small and persistent changes. The two MCUSUM charts were shown to detect small and persistent changes well. These were not as quick as Hotelling's to detect transient and large changes. Hotelling's control chart showed poor performance in detecting small and persistent changes. For simulated persistent changes, the change-point detection model was shown to be efficient in estimating the time of change if the change was large and the number of runs in OC was short relative to the IC period. The number of OC observations used in the estimation was determined by the conditional expected delay. It was also shown that if a change had occurred in the covariance matrix, the change-point estimation procedure was less accurate.

We applied the constructed control charts on similar machines' transformed quality control data. In this application, Hotelling's T^2 control chart showed large differences between runs performed on different cycle settings. It did not detect any large differences between machines with runs on the same setting. However, the MCUSUM control charts showed indications of large structural differences in the mean and the covariance matrix between HiSeq 6 and HiSeq 4 and 5. These differences were also shown with the use of the non-centrality parameter and a ratio between the determinants of the covariance matrices.

The strong OC scenarios shown by the control charts in section 5.4 can partially be explained by the nature of the machines. Some of the HiSeq machines have been upgraded from older versions and are not the same in terms of their specifications. The MCUSUM control chart showed a very strong OC situation for the covariance matrix. As described in section 3.2.2.2, any control chart constructed from the transformed quantities, defined in equation (3.16), would be sensitive to shifts in the mean. These results could partially be explained by the differences in the non-centrality parameter.

The framework of SPC enables us to specify a *desired* mean and variance structure for the target process. This could be applicable to our application as well. However, in the multivariate setting a new issue arises. Specifying a desired mean for a multivariate distribution does not pose a issue but specifying $p(p+1)/2$ elements in a covariance matrix may be very hard, especially if p is large. Also, the transformation used in this thesis provides several complications. Consider the case when the mean vector is known and is specified in the initial parameter space. We assumed that the transformed data is normally distributed and therefore, we would need to transform

the known mean to the new parameter space. The transformation was done by using a Box-Cox transformation, where the parameter λ was estimated from data. Any estimate contains uncertainty and therefore the transformation is not deterministic. The problem becomes even more complicated if the parameter λ would be random.

In the Appendix, section 7.1 the assumption of normality and temporal independence was investigated. The transformed quality control data showed little evidence of being normally distributed. The evidence shown by these statistical tests could perhaps be increased by further investigating transformation methods. Since there was little evidence for the assumption of normally distributed data any conclusions on temporal independence should be done with great care. However, temporal independence is not only a desirable property for the quality control data but should also be considered a necessary assumption. Not only did the data consist of irregular time series, it was made even more irregular from the different run settings a machine could be run on. Each run setting provided different quality characteristics and number of measurements. For some types of settings only certain parts of the flowcell is used. The autocorrelation assumes that data consists of regularly spaced time series, which is not the case. Therefore, using the autocorrelation as a measure of temporal dependency is not only misleading but also violates the assumptions it is built upon. A solution to this problem is to monitor results quality variables at a flowcell level, aggregating each quality measurements for each read and lane to receive one observation per variable. In this setting one could possibly consider the run settings as fixed, regress upon these fixed settings and then monitor the residuals.

The use of Rcpp (c.f. Eddelbuettel, (2013)) significantly reduced the time to perform the simulations in this thesis. The benchmarks, presented in the Appendix section 7.3, provide a great indication of how fast Rcpp together with OpenMP can be. It also provides a good indication of what they can do for computer intensive methods in R. R provides a trade-off between performance and readability. The programming language C++ does not follow this paradigm. Rcpp tries to combine these two by using the performance of C++ and the syntax of R. This makes the transition from R to C++ substantially easier.

The framework of SPC together with the change-point estimation procedure provides a solution to the problem of monitoring changes in NGS quality control data. However, the presented methods all share the underlying assumption of normally distributed data. This advocates the use of non-parametric SPC methods which could be a great subject for future research.

Bibliography

Bodnar, O. et al. (2009). "Surveillance of the covariance matrix based on the properties of the singular Wishart distribution". In: *Computational Statistics and Data Analysis*.

Bodnar, T. and Y Okhrin (2008). "Properties of the singular, inverse and generalized inverse partitioned Wishart distributions". In: *Journal of Multivariate Analysis*.

Box, G.E. and D.R. Cox (1964). "An analysis of transformations". In: *Journal of the Royal Statistical Society Series B (Methodological)*.

Chakraborti, S. et al. (2008). "Phase 1 Statistical Process Control Charts: An Overview and Some Results". In: *Quality Engineering*.

Chandra, R. et al. (2001). *Parallel programming in OpenMP*. Morgan Kaufmann Publishers.

Chen, J. and A.K. Gupta (2011). *Parametric statistical change point analysis: with applications to genetics, medicine, and finance*. Springer Science Business Media.

Conte, S.D. and C.W.D. Boor (1980). *Elementary numerical analysis: an algorithmic approach*. McGraw-Hill Higher Education.

Croiser R., B. (1988). "Multivariate Generalizations of Cumulative Sum Quality-Control Schemes". In:

Croiser, R.B. (1986). "A new two-sided cumulative sum quality control scheme". In: *Technometrics*.

Eddelbuettel, D. (2013). *Seamless R and C++ Integration with Rcpp*. Springer.

Forbes, C. et al. (2011). *Statistical Distributions*. John Wiley and Sons.

Golosnoy, Vasyl et al. (2010). "On the Application of SPC in Finance". In: *Frontiers in Statistical Quality Control 9*. Ed. by Hans-Joachim Lenz et al. Heidelberg: Physica-Verlag HD, pp. 119–130. ISBN: 978-3-7908-2380-6. DOI: 10.1007/978-3-7908-2380-6_8. URL: http://dx.doi.org/10.1007/978-3-7908-2380-6_8.

Gombay, E. and L. Horvath (1994). "An application of the maximum likelihood test to the change-point problem". In: *Stochastic processes and their applications*.

Grimmett, G. R. and D. R. Stirzaker (2001). *Probability and random processes*. Oxford University Press.

Guerrero, V.M. (1993). "Time series analysis supported by power transformations". In: *Journal of forecasting*.

Hair, J.F. et al. (2006). *An introduction to multivariate statistical analysis*. Upper Saddle River, NJ: Pearson Prentice Hall.

Hawkins, D.M. and D.H. Olwell (1998). *Cumulative Sum Charts and Charting for Quality Improvement*. Springer Science Business Media.

Henze, N. and B. Zirkler (1990). "A class of invariant consistent tests for multivariate normality". In: *Communications in Statistics-Theory and Methods*.

Hotelling, H. (1947). "Multivariate quality control illustrated by the air testing of sample bombsights". In: *Techniques of statistical analysis*.

Illumina (2016). *An Introduction to Next-Generation Sequencing Technology*. Tech. rep. Illumina.

Kariya, T. (1981). "A Robustness Property of Hotelling's T2 -Test". In: 9.1, pp. 211–214.

Lai, T. L. (1995). "Sequential Changepoint Detection in Quality Control and Dynamical Systems". In: *Journal of the Royal Statistical Society*.

Lim, S. et al. (2014). "Review: Statistical Process Control (SPC) in the food industry – A systematic review and future research agenda". In: *Trends In Food Science and Technology*.

Metzker, M.L. (2010). "Sequencing technologies — the next generation". In: *Nature genetics*.

Mezzenga E., D'Errico V. Sarnelli A. Strigari L. Menghi E. Marcocci F. Bianchini D. and M. Benassi (2016). "Preliminary Retrospective Analysis of Daily Tomotherapy Output Constancy Checks Using Statistical Process Control." In: *PloS one*.

Mitchell, Ann F. S. and Wojtek J. Krzanowski (1985). "The Mahalanobis Distance and Elliptic Distributions". In: *Biometrika*.

Model, Fabian et al. (2002). *Statistical process control for large scale microarray experiments*. Vol. 18. Bioinformatics.

Moustakides, G.V. (1986). "Optimal stopping times for detecting changes in distributions". In: *The Annals of Statistics*.

Page, E.S. (1954). "Continuous Inspection Schemes". In: *Biometrika* 41.1/2, pp. 100–115. ISSN: 00063444. URL: `http://www.jstor.org/stable/2333009`.

Qiu, P. (2013). *Introduction to statistical process control*. CRC Press, Taylor and Francis group.

Resnick, I. S. (2002). *Adventures in stochastic processes*. Springer Science Business Media.

Roberts, S.W. (1959). "Control chart tests based on geometric moving averages". In: *Technometrics*.

Seber, G. A. F. (2008). *A matrix handbook for statisticians*. John Wiley and Sons.

Shewhart, W.A. (1931). *Economic control of quality of manufactured products*. Quality Press.

Siegmund, D. (1985). *Sequantial analysis*. Springer.

Sundberg, R. "Statistical modelling by exponential families". Published by the mathematics department at Stockholm university.

Thor, J. et al. (2007). *Application of statistical process control in healthcare improvement: systematic review*. Vol. 16. 5. Quality Safety in Health Care.

Villasenor Alva, J.A. and E.G. Estrada (2009). "A generalization of Shapiro–Wilk's test for multivariate normality". In: *Communications in Statistics—Theory and Methods*.

Appendix

7.1 Transformation, normal assumption and autocorrelation.

In this section we will present the transformation methods and evaluate the assumption that the process follows a multivariate normal distribution. We will also look upon the autocorrelation and to what extent it is present in the data. However, the autocorrelation should be interpreted with caution. Not only does the assumption of temporal independence depend upon the normality assumption but it also assumes that the timeseries is regular.

For the variables which have support on the positive real line we will use the Box-Cox transformation, presented in Box and Cox, (1964), i.e.

$$Z = \begin{cases} \frac{X^\lambda - 1}{\lambda} & \text{for } \lambda \neq 0 \\ \log(X) & \text{else.} \end{cases}$$

for transformation. The variables are transformed independently. The parameter λ is estimated using the method suggested in Guerrero, (1993). The Box Cox transformation and the guerro estimation method is implemented in the `forecast` package. For variables which have limited support on $(0, 1)$ we will use the standard normal quantile function as a transformation method. Consider X having support on (0,1), then we have that

$$Z = \Phi^{-1}(X),$$

where Z will follow a normal distribution.

Two statistical tests of the normal assumption are presented in Table 7.1, performed on the transformed data. Henze-Zirkler's multivariate test of normality, presented in Henze and Zirkler, (1990), shows some evidence that the data is normally distributed. The genaralized Shapiro-Wilk test of normality, presented in Villasenor Alva and Estrada, (2009), shows no evidence at all.

	Test	P.value
1	Henze-Zirkler's	0.29
2	Generalized Shapiro-Wilk	0.00

Table 7.1: Two statistical tests of normality, Henze-Zirkler's and a generalized Shapiro-Wilk's test. One out of two tests approves of the normality assumption.

Under the assumption that the transformed data *is* normally distributed, the autocorrelation may be investigated. If the absolute value of the autocorrelation at a given lag is below the standard normal distributions 97.5% percentile we can assume that the data is independent in time. There are a total of 1176 correlation coefficients at each lag to investigate. In Table 7.2 we can see the proportion of absolute autocorrelationcoefficients which are greater than the standard normal distributions 97.5% percentile.

Lag	Proportion
1	0.21
2	0.11
3	0.05
4	0.04
5	0.03

Table 7.2: Proportion of autocorrelation greater than the normal 95 percentile, at lags 1 through 5.

7.2 Figure from Illumina

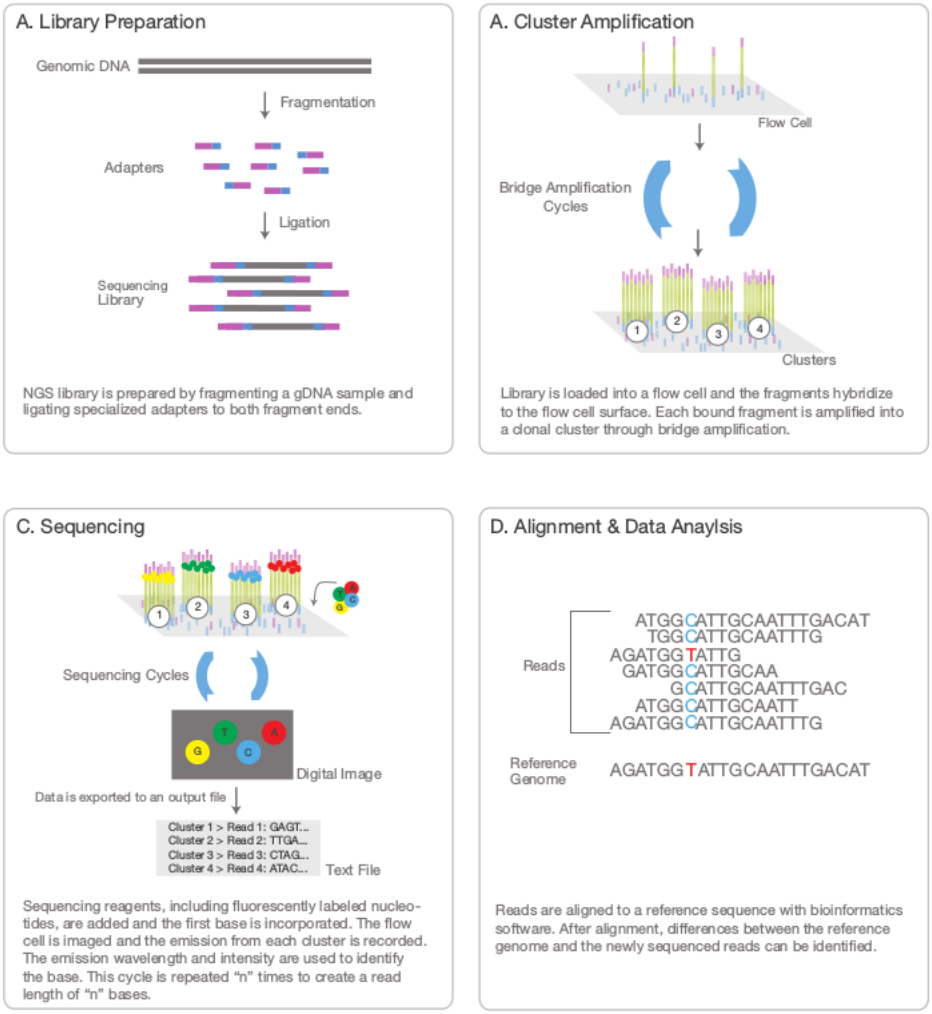

Figure 7.1: Figure illustrating the general workflow in sequencing by synthesis. The figure is taken from Illumina, (2016) with approval.

7.3 Benchmarks

In this section we will shortly present some benchmarks of the SimulateARL0 function implemented in Rcpp compared to the same function implemented in base R and R run in parallel, using the **foreach** package. To create the benchmarks we used the following inputs

- The in control mean vector and covariance matrix from HiSeq 6 transformed quality control data
- A allowance constant equal to 0.3
- The control limit 2580.24
- The number of threads was held constant, equal to 7.

The number of simulations N=$\{10^2, 10^3, 10^4, 10^5\}$. Each simulation is performed 10 times and the average time is taken as the benchmark for this specific setting. The test system was a Intel®Core i7-4770S@3.1Ghz with 16Gb system RAM running Ubuntu 14.04.4. In Figure 7.2 we can see the average time it takes to perform N simulations. Rcpp using OpenMP is around 10 times faster compared to using base R. Using Rcpp together with OpenMP, compared to base R in parallel using **foreach** package, only results in twice the performance. To perform 10^4 simulations with Rcpp takes 32.53 minutes on average.

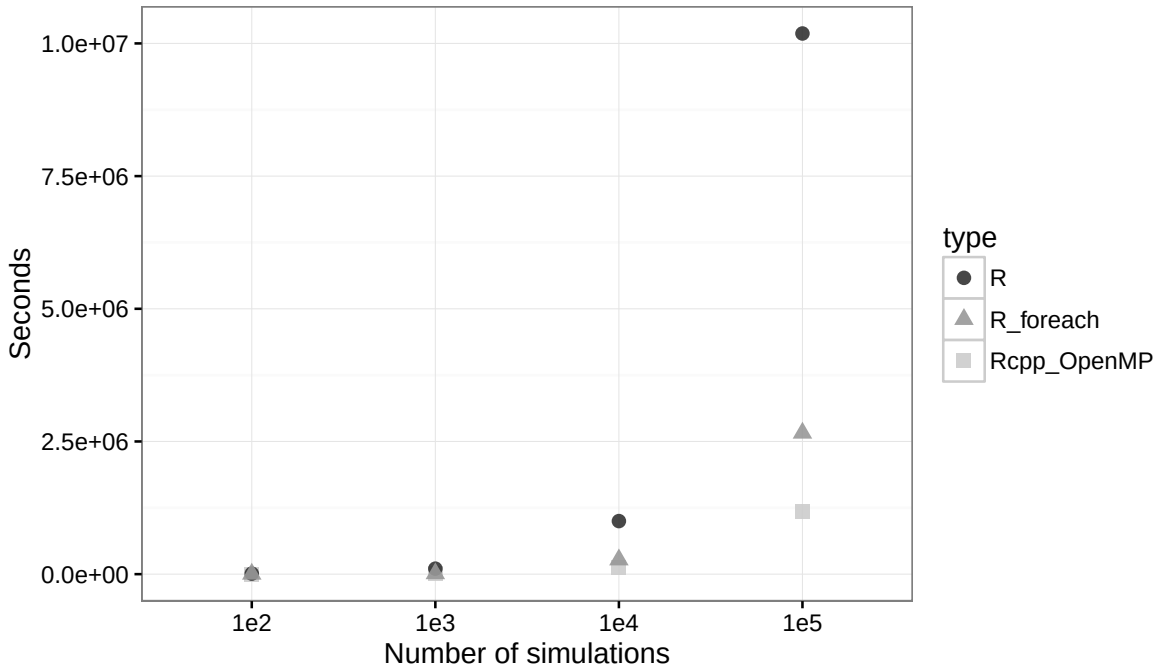

Figure 7.2: Benchmark of the SimulateARL0 function, implemented in base R, base R run in parallel using the foreach package and Rcpp together with OpenMP.