DATA NORMALIZATION AND CLUSTERING FOR BIG AND SMALL DATA AND AN APPLICATION TO CLINICAL TRIALS

BY YAYAN ZHANG

A dissertation submitted to the
Graduate School—New Brunswick
Rutgers, The State University of New Jersey
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
Graduate Program in Statistics and Biostatistics
Written under the direction of
Javier Cabrera
and approved by

New Brunswick, New Jersey
May, 2015

ABSTRACT OF THE DISSERTATION

Data Normalization and Clustering for Big and Small Data and an Application to Clinical Trials

by YAYAN ZHANG

Dissertation Director: Javier Cabrera

The purpose of this thesis is to propose new methodology for data normalization and cluster prediction in order to help us unravel the structure of a data set. Such data may come from many different areas, for example clinical responses, genomic multivariate data such as microarray, educational test scores, and so on. In addition and more specifically for clinical trials this thesis proposes a new cohort size adaptive design method that will adapt cohort size eventually and finally will save time and cost while still keep the accuracy to find the target maximum tolerate dose.

The new normalization method is called Fishe-Yates normalization and it has the advantage of being computationally superior than the standard quantile normalization and it improved the power of the following statistical analysis. Once the data has been normalized the observations are clustered by their pattern of response and cluster prediction is used to validate the findings. We propose a new method for cluster prediction which is a natural way to predict for hierarchical clustering. Our prediction method using nonlinear boundaries between clusters.

Normalization method and clustering prediction method can help to identify subgroups of patients which has positive treatment effect. For clinical trial study, this thesis also proposes a new adaptive design which will adapt cohort size thus save time and cost to locate the target maximum tolerated dose.

Acknowledgements

I am really grateful to went through my PhD in Statistics department of Rutgers. I am using this opportunity to express my gratitude to everyone who supported me throughout these years.

First of all, I would like to express my deepest gratitude to my thesis advisor Professor Javier Cabrera. He always had lot of brilliant ideas about the topics in this thesis. Without his invaluable guidance and constant support, this dissertation would be impossible. To me, he is far beyond my thesis advisor. He does not only guide me on how to do research, but also share his social connection and life experience with me. He brought me to attend the Cardiovascular Institute weekly meeting, to meet with people in outside companies. All of these experiences improved my communication skills and trained me a lot for my future career. I am really grateful to have such a great advisor like him.

To Professor John E. Kolassa, the Director of graduate program in our department. He is very nice and always devote his effort to help students. I thank for his guidance and help with graduate program requirements understanding. He encouraged me to take the written qualifying exam before I came to Rutgers four years ago thus I were able to save one year for my PhD study. With his help, I bypassed lot of obstacles both in my study and in my life.

Special thanks go to my committee members Professor Lee Dicker and Dr.Birol Emir for their precious time and efforts. I would thank Dr. Birol Emir who has kindly provided me access to the neuropathic pain data set used in the research projects.

My gratitude also goes to my summer intern supervisor from Pfizer: Dr.David Li. He spent lot of time to work with me even after my internship ended. Without his help, I could never extend the intern project into the third part of my dissertation.

A special warm thanks go to my beloved husband Xin. I cannot image how I get through all these thistles and thorns in my life without his encouragement and support. Thank you for always stayed strong and raised me up. Thank you for always being there no matter good or bad.

Finally, I would like to thank the Department of Statistics and Biostatistics for offering me this opportunity to receive my degree here and providing me support. I thank Dr. Kezhen Liu, Dr. Lan Yi, Dr. Jie Liu and Dr. Ning Tang, for their suggestions and encouragements on my thesis work and my job hunting. I thank my friends Chuan Liu and Die Sun for various course topics discussion through all these years. I also thank Long Feng, Ting Yang, Xinyan Chen, Xialu Liu and other friends, I deeply appreciate their friendship that made this experience so memorable.

Dedication

To my father Muliang Zhang and mother Qiaoyun Wu.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

One of the motivating examples for the statistical methodology presented in this thesis is the analysis of clinical trial data for the treatment of neuropathic pain. Neuropathic pain (NeP) is a complex, central nervous system related pain state. It is part of the disorder or dysfunction of peripheral nerves. Usually neuropathic pain is caused by diseases affecting the somatosensory system, and presents heterogeneous sensory deficits. The nerve fibers maybe damaged and thus will send incorrect information to other pain centers.

Though there is no obvious cause for neuropathic pain, there are several common causes: alcoholism, diabetes, HIV infection or AIDS, Chemotherapy etc. Generally, neuropathic pain symptoms may include shooting and burning pain, tingling and numbness. It actually is not a single disease, but is a complication that is based on various underlying medical conditions. Thus the research targeted to find effective treatment for neuropathic pain is always a challenge. The Institute for Advanced Reconstruction at the plastic surgery center indicated that "more than 100 types of peripheral neuropathy have been identified, each with its own particular set of symptoms, patterns of development and prognosis".

To diagnosis neuropathic pain, usually the doctor will conduct a few questionnaires which evaluate different pain symptom descriptors. The patient maybe asked to record a daily pain diary on a numeric rating scale, and also record when and how this pain arises. Blood tests and nerve tests are also necessary sometimes. Due to the limitations of current treatment, a number of neuropathic pain studies failed to meet the primary efficacy target and many patients felt no relief or even occasionally got worse instead of better over time.

A research study was conducted by Freeman et al. (2014) aimed to depict any phenotype of neuropathic pain and uncover any pattern of sensory symptoms if exist. Freeman et al. (2014) mentioned that "the management of neuropathic pain is challenging and many patients obtain only partial or no pain relief taking currently available pain medications".

Four clinical studies were conducted by Freeman et al. (2014), for testing the treatment effect of an undisclosed drug. The results of the four studies turned out to be negative, because they found no difference between treatment and placebo. We find out in this example that the reason why the analysis didn't work is that pain acts differently in different people. It is unlikely to find a treatment that reduces pain on everybody. However, the study also provides pain scale measures based on sensory scales and pain scales that can be used to find clusters or patients with similar pain patterns some of which will respond to treatment. The analysis that was initially performed using z-score normalization was not able to detect clusters that were predictive of good response to treatment. This is because the skewness of the pain measurements is obscuring the good clustering and z-score transformations are linear transformations and do not alter skewness and the underlying structure maybe missed. This is just an example but this problem is very generalized among scale data, microarray and genomic data, education testing data, imaging data and others.

The purpose of this thesis is to propose some new methodology for normalizing the data and for cluster prediction that help us unravel the structure that is presented in the data. In addition and more specifically for clinical trails this thesis propose a new adaptive design method that can save time to perform the study.

Normalization methods are very important in data sets in which the measurement scales depend on the observation. For example two patients fill up questionnaires one gives high values while the other gives low values but their pain maybe the same. Microarrays and image data are based in pictures or scans which are sensitive to the luminosity at the instance of the picture. The idea of normalization is to make the scales as similar as possible.

In our example of pain clinical trials we improve the data set by standardizing or normalizing the answers using the Fisher Yates normalization and the hierarchical clustering methods, which hopefully will be able to detect subgroups with positive treatment response. For the adaptive design methodology we promoted a new cohort adaptive method which could find the maximum tolerate dose more quickly while still keep the accuracy.

1.2 Thesis Goals and Overview

The goal of this thesis is *to develop new techniques to pre-process the big and small data as well as demonstrate new prediction algorithm and adaptive method that could better detect subgroups with positive treatment response, and could save cohort numbers and time when locating the target maximum tolerated dose in clinical trial studies by applying to the existing challenges and problems.*

The remainder of the thesis document is structured as follows:

In **Chapter2** we present a new data normalization method with Fisher-Yates transformation that could remove as much as possible, any systematic sources effects of variation. We also build extensive simulation analyses and conduct real data application on neuropathic pain questionnaire data and Sialin data.

In **Chapter3** we study the incorporating of new data to an existing hierarchical clustering partition that was obtained earlier using older training data.

We derive a novel method called hierarchical clustering tree prediction method, which will use the existing hierarchical tree to insert the new observation into the appropriate cluster by inter-point distance and inter-cluster distance.

Finally, in **Chapter4** we demonstrate in clinical trial study, our new approach called Dose and Size adaptive design can save study time and thus save cost while keep the accuracy to find the true maximum tolerate dose, which is very exciting result in clinical trial study.

1.3 Data Normalization and Fisher-Yates Transformation

Once the experiment has been done and data has been collected, it is necessary to preprocess the data before formally analyzing it. The purpose of normalization is to remove as much as possible, any systematic sources effects of variation. Data normalization is a very popular technique to remove technological noise in genomic, in test scoring, in questionnaire scoring of medicine, and social sciences etc.

Early researchers noticed that there are substantial differences in intensity measurements among questionnaire data and microarrays which were treated exactly alike. In this thesis, we will show there are statistical analysis challenges in using this questionnaire data and microarray data while conducting standard analyses such as linear and non-linear modeling, factor analysis, cluster analysis etc. It has been emphasized in many literature about the importance of preprocessing the predictors and responses before modeling and analysis.

We prove that the traditional transformation methods (centering, scaling, z-scores) are inadequate to fulfill this task and we propose a new data normalization method with Fisher-Yates transformation. We build extensive simulation analyses and conduct real data application on neuropathic pain questionnaire data and Sialin data to show Fisher-Yates transformation is more successful at removing noise and reducing skewness.

The idea of quantile normalization is motivated by making the common scales of the normalized data as close as possible to the true scale of the data. Conceptually it seems like the proper thing to do, but the problem is then the data analyzed using T-tests or F-tests will rely much on the assumption of normality. Suppose we have a data set $X = (X_1, ..., X_P) = \{x_{ij}\}$ $(i = 1, ..., I;\ j = 1, ...P)$, where X_j stands for the column array, and $X_j = x_{1j}, ..., x_{Ij}$. The column of X represent the observations and the row of X are the variables or questions for questionnaire data or gene.

The main idea of quantile normalization is to first sort each subject vector and calculate the coordinate-wise median of the vectors, say M_I. Then replace each x_{ij} in M_I with the corresponding rank, thus $Q_n(X) = \{M[r_{ij}]\}$, where $i = 1, ..., I$ and $j = 1, ..., P$. One concern for quantile normalization is that the median array may not look similar to any of the arrays and may has shorter tailer than other arrays in the data set. Another concern is, when the number of variables or genes or questions in our data set is not large, the median array is very variable and may not be adequate for normalization.

Quantile normalization tries to put all data into the same scale and this is scale as close as possible to the true scale. However, it does not remove the skewness of the original if it is very skewed. We proposed that Fisher-Yates normalization can bring data into same scale and reduce skewness of data at the same time. We also conclude that Fisher-Yates normalization handles skewness and outliers better than quantile normalization and as a result it increases the power to detect genes that are differentially expressed between arrays and also it gets better classification results by simulation study and application on real data example.

1.4 Hierarchical Clustering Tree Prediction

Cluster analysis, also known as data segmentation, is an *unsupervised* classification method to split the data into clusters or segments, so that within cluster objects are as homogenous as possible and between clusters are as heterogeneous as possible. Clustering methods can be classified as hierarchical (nested) or partitional (un-nested). However, they all depend on a dissimilarity or a similarity measure, depict the “distance” of two observations: how far or how close, the observations are from each other.

Here we want to study the incorporating of new data to an existing hierarchical clustering partition that was obtained earlier using older training data. The issue is on how do we use the existing hierarchical clustering to predict the clusters for the new data. The standard prediction method is to assign the new observation to the closest cluster using inter-cluster distance between that observations and the existing clusters. We derive a novel method called hierarchical clustering tree prediction method (HCTP), which will use the existing hierarchical tree to insert the new observation into the appropriate cluster.

We analyzed a data set about the treatment effect of Lyrica on neuropathic pain patients data from four randomized clinical trials (Freeman et al. (2013)). After the data on baseline symptoms of Neuropathic pain was normalized by Fisher-Yates we applied hierarchical clustering we identified three clusters. Once the clusters were established, a new clinical trial data became available for us. We wanted to assign new patients from the recent trial into the established clusters from the previous four trials.

The basic idea for this method is to include the new observed data with the original data and to perform hierarchical tree clustering until the new observation join a cluster A. Then we assign the new observation to cluster A' which is the cluster in the original configuration where points in cluster A falls into. But for different inter-cluster distance we may find a different way to do this. This new method depends on the inter-point distance and inter-cluster distance which was used to generate the hierarchical tree.

We will study the most commonly used distance measures like “Single linkage”, “Complete linkage”, “Average linkage” and “Ward’s method” for hierarchical clustering to compare our HCTP to the standard prediction method. The classification boundaries are different from HCTP and traditional method in our simulation study and also misclassification rate is reduced.

1.5 Dose and Cohort Size Adaptive Design

Early-phase clinical trials are first-in-human studies for new treatment. The primary objective of phase I oncology trial is to define the recommended phase II dose of a new drug, aiming at locating the MTD. The main outcome for most existing dose-finding designs is toxicity, and escalation action is guided by ethical considerations. It is very important to estimate the MTD as accurate as possible, since it will be further investigated for efficacy in Phase II study. The study will begins at low doses and escalate to higher doses eventually due to the severity of most DLTs. However, we also want the escalation of doses to be as quick as possible since the lower doses are expected to be ineffective in most cases.

A rich literature has been published for dose-finding designs of Phase I trials. The conventional 3+3 design, first introduced in the 1940s, is still the most widely utilized dose-escalation and de-escalation scheme. However, there are some limitations when applying 3+3. Statistical simulations demonstrated that 3+3 design is used to identify the MTD in as few as 30% of trials. Another very popular model-based method is Continual Reassessment Method (CRM) which estimate the MTD based on one-parameter model and eventually updated the estimator every time one cohort completes either by Bayesian methods given by

O'Quigley et al. (1990), or maximum likelihood methods given by O'Quigley and Shen (1996).

Traditional adaptive methods will adapt dose up and down eventually depend on the toxicity from the observed data. We promoted a novel dose assignment method called dose and cohort size (*D*&*S*) adaptive design, which is based on conjugate Beta prior, and will adapt dose and cohort size at the same time, thus able to detect the true MTD with less cohorts while still keep the accuracy.

For dose escalation rules, *D*&*S* follows the same principles as 3+3, TPI and CRM etc., that will "Escalate" if current dose has AE rate too high, "Stay" if around the target rate, and "De-escalate" if is too low. Also, we change the cohort size depending on whether the next dose is likely or unlikely to be the MTD. We will not change cohort size if we are uncertain it is MTD, add more subjects if the dose is likely to be the dose with targeted AE rate, and add much more subjects if the dose is highly likely to be the dose with targeted AE rate. Simulation results indicate that with appropriate parameters, *D*&*S* design performs better at estimating the target dose and at subject assignment of the target dose.

This new method may also appeal to physicians while its implementation and computation are very simple. To implement this new method, we will need to specify the target toxicity probability p_T, the number of doses D and true toxicity probabilities for each dose to start simulation study.

The main distinction of this new proposed method is: it requires all information from current dose, lower dose and higher dose to decide dose assignment action. And we will adapt cohort size at the same time when some specified criteria are satisfied.

Chapter 2

Data Normalization and Fisher- Yates Transformation

Data normalization is a very popular technique to remove technological noise in genomic, in test scoring, in questionnaire scoring of medicine, and social sciences etc. Early researchers noticed that there are substantial differences in intensity measurements among questionnaire data and microarrays which were treated exactly alike. There are statistical challenges in using these data when conducting standard analyses such as modeling or clustering and the main issue is to preprocess the data to construct a response score. We show that the traditional transformation methods (centering, scaling, z-scores) are inadequate to fulfill this task and we propose a new data normalization method with Fisher-Yates transformation. We build extensive simulation analyses and conduct real data application on neuropathic pain questionnaire data and Sialin data to show Fisher-Yates transformation is more successful at removing noise and reducing skewness.

2.1 Introduction and Motivation

2.1.1 Background and Introduction

Once the experiment has been done and data has been collected, it is necessary to preprocess the data before formally analyzing it. The purpose of normalization is to remove as much as possible, any systematic sources effects of variation. Normalization methods can greatly enhance the quality of any downstream analyses.

I present here two basic examples of data that are commonly needed preprocess of normalization:

(I) Questionaire data example.

Questionnaire data is a research tool utilized in many research areas. A questionnaire means eliciting the feelings, beliefs, experiences or attitudes from sample of individuals. Though there are economy and uniformity of questions advantages when using questionnaire data, the respondent's motivation is difficult to assess and sample bias does exist at the beginning of the study.

Many questionnaires are based on scales such like Likert scales that take values in a fixed range (1-7 or 0-10) and it is common to have many such questions on the same questionnaire. The response score is very personal dependent, with same stabbing, some subjects will return generally high scores while others will return relatively low scores. This is a very common behavior among population and has been studied in many different research areas.

Our concerns about this questionnaire data are:

1. The distribution of the response scores may differ substantially from subject to subject, spread and shape;
2. The boundary threshold effects at the low and high values make the distribution of the scores either left or right skewed.
3. Especially the introduction of online questionnaires or medical outcomes questionnaires that are recorded by health providers, the number of cases can grow very large and the data becomes big data.

(II) Microarray Data Example.

Early microarray researchers indicated there are substantial difference in intensity measurements among microarrays which were treated exactly alike. Microarray technology though popular, is well known of various technical noises due to the limitation of technology. Though the magnitude is reduced now due to improvements of technology, difference still persist.

The systematic effects introduced into intensity measurements due to the complexities of microarray experimental process can be substantial enough to dilute the effects that the experimenter is trying to detect. Sources of variability caused Systematic effects were summarized by Amaratunga and Cabrera (2004):

- "the concentration and amount of DNA placed on the microarrays, arraying equipment such as spotting pins that wear out over time, mRNA preparation, reverse transcription bias, labeling efficiency, hybridization efficiency, lack of spatial homogeneity of the hybridization on the slide, scanner setting, saturation effects, background fluorescence, linearity of detection response, and ambient conditions."

In order to make valid comparisons across microarrays, we need to remove the effects of such systematic variations and bring the data onto a common scale.

2.1.2 Case Study

There are many data sets used by Amaratunga and Cabrera (2004) in book "Exploration and Analysis of DNA microarray and Other High-Dimensional Data". The Golub data, the Mouse5 data, the Khan data, the Sialin data, the Behavioral study data, the Spiked-In data, the APOAI study data, the Breast Cancer data, the Platinum Spike data set, and Human Epidermal Squamous Carcinoma Cell Line A431 Experiment data. Amaratunga and Cabrera (2004) applied generally quantile normalization on these data set. In particular we are looking into the Sialin Data as microarray data example to demonstrate our Fisher-Yates normalization method.

The Sialin data was collected basically from two different types of mice: Slc17A5 gene knocked out mice, and the wild-type mice ("normal" mice), where

Slc17A5 is the gene expression for the production of Sialin. The RNA samples then collected from newborn and 18-day-old mice from these two types mice. The final profile has 496,111 probes corresponding to 45,101 genes collected from RNA samples using Affymetrix Mouse430-2 Gene-chips. Other biological data can achieve to much higher dimension like 30 million variables.

Questionnaire data used by Freeman et al. (2014) were from patients who were males or non-pregnant, non-lactating females aged ≥ 18 days with a diagnosis of NeP syndromes: CPSP, PTNP, painful HIV neuropathy and painful DPN. The NPSI questionnaire evaluated 10 different pain symptom descriptors: superficial and deep spontaneous ongoing pain; brief pain attacks or paroxysmal pain; evoked pain (pain provoked or increased by brushing, pressure, contact with cold on the painful area); and abnormal sensations in the painful area.

2.1.3 Why Fisher-Yates Normalization

There are statistical analysis challenges in using this questionnaire data and microarray data while conducting standard analyses such as linear and non-linear modeling, factor analysis, cluster analysis etc. It has been emphasized in many literature about the importance of preprocessing the predictors and responses before modeling and analysis.

The objective of questionnaire score normalization is to reduce as much as possible the difference in shape between set of scores belonging to the same subject. By doing so we improve the compatibility of the individual subject scales so that the variables that come out of the questionnaire are more homogeneous and could be better used in further analysis.

Traditionally the questionnaire scores are replaced by z-scores obtained from each individual subject data. This calibrates each subject to have zero mean and one standard deviation but it does not affect the skewness and other shape measures of the data. However, we find an application of Fisher-Yates scoring which we call FY normalization can remedy the shortages. Quantile normalization is widely used to analyze microarray data, but we will also demonstrate in the following sections why we prefer Fisher-Yates normalization.

2.2 Exist Normalization and Standardization Methodology

Once we have collected the data, it is necessary to pre-process it before formally analyzing it. There are several issues we could solve to enhance the quality of any downstream analyses:

1. to transform the data into a scale suitable for analysis;
2. to remove the effects of systematic sources of variation;
3. to identify discrepant observations and arrays.

The purpose of normalization is to remove the effects of any systematic sources of variation as much as possible by data processing. Systematic effects can dilute the effects that the experimenter is wanting to detect. Some source variability can be controlled by the experimenter, however, we can not eliminate them completely. Early researchers noticed this problem and did lots of work to remove the effects of such systematic variations.

2.2.1 Global or Linear Normalization

Early methods used normalization by the sum, by mean, by the median and by Q3 (third quantile). For example, for normalization by the sum, the sums for each individual of questionnaire data are forced to be equal to one another. Suppose the k original sums were $X_{1+}, ..., X_{k+}$ and we divide i^{th} sum by X_{i+}, then we force the sum to be 1. Similarly, normalization by the mean will force the arithmetic means of each individual to be equal; and normalization by the median equated the row medians.

We call these examples global or linear normalization. For linear normalization, we assume the spot intensities for every pair of individual scores are linearly related without intercept. Then we can apply normalization scheme to adjust intensity for every single score by the same amount to reduce the systematic variation and make the data more comparable.

2.2.2 Intensity-Dependent Normalization

We could use global normalization if the pair of our data is linearly related without intercept. But in most cases, the spot intensities are nonlinear. Different factors needed to adjust low-intensity and high-intensity measurements. We call this normalization scheme intensity-dependent normalization while the normalizing factor is a function of the intensity level. We denote the nonlinear normalization function as: $X \to f(X)$.

A lot of pre-work had been done for this intensity-dependent normalization including Amaratunga and Cabrera (2001), Li and Wong (2001), Schadt et al. (2001). Baseline array needs to be specified for intensity-dependent normalization. For example, the median mock array. If X_{ij} represents the transformed spot intensity measurement for the i^{th} individual (i=1,...,I) for the jth question (j=1,...,P), the median mock array for kth observation is:

$$M_k = median\{X_{k1}, ..., X_{kP}\}. \tag{2.1}$$

There are several ways to perform intensity-dependent normalization.

Smooth Function Normalization

For smooth function normalization, there is an inverse function $g_i = f_i^{-1}$ is estimated by fitting the model

$$X_{ij} = g_i(M_k) + \varepsilon_{ij} \tag{2.2}$$

where ε_{ij} is a random error term and the normalized values for the ith individual are obtained from

$$X_{ij}' = f_i(X_{ij}) \tag{2.3}$$

Stagewise Normalization

Stagewise normalization is used when data is combined with technical and biological replicates. Usually smooth function normalization is applied to technical replicates, and quantile normalization is applied to biological replicates.

Other Intensity-Dependent Normalization Methods

Quantile Normalization and Fisher-Yates Normalization are two other very popular intensity-dependent normalization methods, we will discuss quantile normalization and propose Fisher-Yates method in details in the following sections.

2.3 Quantile Normalization and Fisher-Yates Transformation

The idea of quantile normalization is motivated by making the common scales of the normalized data as close as possible to the true scale of the data. Conceptually it seems like the proper thing to do, but the problem is then the data analyzed using T-tests or F-tests will rely much on the assumption of normality.

Quantile normalization and Fishr-Yates normalization are performed basically when the data is non-normal. We expect to see some power loss on the t-test with respect to the non-normal method. Also the more skewed data is, the less reliable is the tail of the observed data.

We have a data set $X = (X_1, ..., X_P) = \{x_{ij}\}$ $(i = 1, ..., I;\ j = 1, ...P)$, where X_j stands for the column array, and $X_j = x_{1j}, ..., x_{Ij}$. The column of X represent the observations and the row of X are the variables or questions for questionnaire data or gene.

2.3.1 Quantile Normalzation

Quantile normalization introduced by Amaratunga and Cabrera (2001) is to make the distributions of the transformed spot intensity as similar as possible, or at least to the distribution of the median mock array. For quantile normalization, the shape of the normalized data is the median shape of the original data. But the data maybe skewed and median shape is deformed on the tails.

Amaratunga and Cabrera (2001) proposed the idea of standardization or normalization by quantiles for micro-array data under the name of quantile standardization and was changed to quantile normalization later by Irizarry et al. (2003). The differences between micro-array data and questionnaire data are: (a) the measurements are continuous, (b) the shapes of subject observations are more similar, and (c) the number of subjects I is usually much smaller than the number of predictors P, $I < P$.

The algorithm for constructing the quantile normalization of the rows of a data matrix X with I observations (rows) and P genes (columns) is as follows:

1. Construct the median subject. First we need to sort each of the subject vectors and calculate the coordinate-wise median of the vectors and lets call this vector M of length P. Let X^* represents the sorted data, we say

$$X^* = \{X_1^*, ..., X_P^*\}; \tag{2.4}$$

and

$$X_j^* = \{x_{(1)j}, ..., x_{(I)j}\}. \tag{2.5}$$

Where X_j^* is the ordered vector, and $x_{(1)j} \leq x_{(2)j} \leq ... \leq x_{(I)j}$. Then the median vector, M_I can be derived as

$$M[i] = median\{x_{(i)1}, ..., x_{(i)P}\}; \tag{2.6}$$

2. We know x_{ij} is the i^{th} score (question) of j^{th} column (subject). Let r_{ij} be the rank of i^{th} score among j^{th} column, $1 \leq i \leq I$ and $1 \leq j \leq P$. Then we will replace x_{ij} with $M(r_{ij})$ by column and the resulting array Y_{ij} is the normalized scores.

$$Q_n(X) = \{M[r_{ij}]\} \tag{2.7}$$

$$i = 1, ..., I; j = 1, ..., P.$$

2.3.2 Fisher-Yates Normalization

One concern for quantile normalization is that the median array may not look similar to any of the arrays and may has shorter tailer than other arrays in the data set. Another concern is, when the number of variables or genes or questions in our data set is not large, the median array is very variable and may not be adequate for normalization.

Also we know quantile normalization tries to put all data into the same scale and this is scale as close as possible to the true scale. However, quantile normalization does not remove the skewness of the original if it is very skewed. In order to fix skewness, we need to do other. Here we propose Fisher-Yates rank transformation to normalize the data, we call Fisher-Yates Normalization. Fisher-Yates normalization can bring data into same scale and reduce skewness of data at the same time, thus we can see in our simulation study power is improved due to skewness reduction. Also good properties of Fisher-Yates are listed and proved in the following section.

Generally, the algorithm for Fisher-Yates normalization is: suppose x_{ij} is the i^{th} score (question) of j^{th} column (subject). Let r_{ij} be the rank of i^{th} score among j^{th} column, $1 \leq i \leq I$ and $1 \leq j \leq P$. Then x_{ij} will be replaced by $\Phi^{-1}(r_{ij}/(I+1))$ and the resulting array Y_{ij} is the Fisher Yates normalization scores.

Fisher-Yates (1928) proposed Fisher-Yates transformation by replacing z-scores by scores based on ranks and assign the scores the corresponding quantile of a standard normal distribution.

$$FY(x_{ij}) = \Phi^{-1}(r_{ij}/(I+1)) \tag{2.8}$$

for Fisher-Yates Normalization $FY(X)$ can be proposed as

$$FY(X) = \{FY(X_1), ..., FY(X_P)\} \tag{2.9}$$

Theorem 1. *If X^M is a random variable from a distribution F and suppose we draw observations $\{x_1^M, ..., x_I^M\} \sim \mathcal{F}$. In reality we observe $x_{ij} = \psi_j(x_i^M + \epsilon_{ij})$ where ψ_j is also unobserved and strictly monotonic. Then $\{\psi_j, x_i^M\}$ are not identifiable.*

Proof. Suppose h is also strictly monotonic, let $\hat{\psi}_j$ and $\hat{x_i^M}$ be estimators of ψ_j and x_i^M, such that $x_{ij} = \hat{\psi}_j(\hat{x_i^M})$. Then if $\tilde{\psi}_j = \hat{\psi}_j(h^{-1})$ and $\tilde{X_i^M} = h(\hat{X}_i^M)$, we have $x_{ij} = \tilde{\psi}_j(\tilde{x_i^M}) = \hat{\psi}_j(\hat{x_i^M})$. Thus means that $\{\psi_j, x_i^M\}$ are not identifiable. $\square$

Quantile normalization solve this pattern of non-identifiability by setting

$$\hat{X}_{(i)}^M = \underset{j}{Median}\{X_{(i)j}\}.$$

And Fisher-Yates normalization solves by setting

$$\hat{X}_{(i)}^M = \Phi^{-1}(r_{ij}/(I+1)).$$

2.3.3 Properties for Fisher-Yates Transformation

Property 1. *The Fisher-Yates transformation FY can be obtained from the Quantile normalization algorithm Qn by replacing $M[r_{ij}]$ with $\Phi^{-1}(r_{ij}/(I+1))$.*

Proof. For Quantile Normalization

$$Q_n(X) = \{M[r_{ij}]\}$$

we replace with Fisher-Yates transformation $\Phi^{-1}(r_{ij}/(I+1))$, then we have

$$\{\Phi^{-1}(r_{ij}/(I+1))\} = \{FY(X_1), ..., FY(X_P)\} = FY(X)$$

□

Property 2. *If $Skew(X) > 0$:*

1. *If X has no ties, then $Skew(FY(X)) = 0$;*
2. *If X has tie but proportion of tie goes to zero when $n \to \infty$, then*

$$\lim_{n\to\infty} Skew(FY(X)) = 0$$

Proof. If the random variable X has n observations, then the estimator of population skewness can be written as:

$$skew(X) = \frac{E(X - E(X))}{s^3} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X})^3}{\sqrt{[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X})^2]}^3}$$

where $\bar{X}$ is the sample mean and s is the sample standard deviation.

$$\begin{aligned} skew(FY(X)) &= \frac{E(FY(X) - E(FY(X)))}{s(FY(X))^3} \\ &= \frac{\frac{1}{n}\sum_{i=1}^{n}[\Phi^{-1}(r_i/(n+1)) - mean(FY(X))]^3}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}[\Phi^{-1}(r_i/(n+1)) - mean(FY(X))]^2}^3} \end{aligned}$$

1. If X has no ties:

$$
\begin{aligned}
mean(FY(X)) &= \frac{1}{n}\sum_{i=1}^{n}[\Phi^{-1}(r_i/(n+1))] \\
&= \frac{1}{n}\{[\Phi^{-1}(\frac{1}{n+1}) + \Phi^{-1}(\frac{n}{n+1})] + [\Phi^{-1}(\frac{2}{n+1}) + \Phi^{-1}(\frac{n-1}{n+1})] + ...\} \\
&= \frac{1}{n}\{0+0+...\} = 0
\end{aligned}
$$

Then

$$
\begin{aligned}
skew(FY(X)) &= \frac{\frac{1}{n}\sum_{i=1}^{n}[\Phi^{-1}(r_i/(n+1))]^3}{sd(FY(X))^3} \\
&= \frac{\{[(\Phi^{-1}(\frac{1}{n+1}))^3 + (\Phi^{-1}(\frac{n}{n+1}))^3] + [(\Phi^{-1}(\frac{2}{n+1}))^3 + (\Phi^{-1}(\frac{n-1}{n+1})^3)] + ...\}}{n.sd(FY(X))^3} \\
&= \frac{\{0+0+...\}}{n \cdot sd(FY(X))^3} = 0
\end{aligned}
$$

2. If X has tie but proportion of tie goes to zero when $n \to \infty$, that is

$$\lim_{n\to\infty} \frac{O(\mathcal{N}_{\mathcal{T}})}{n} \to 0$$

Then

$$
\begin{aligned}
mean(FY(X)) &= \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n}[\Phi^{-1}(r_i/(n+1))] \\
&\leq \lim_{n\to\infty} \frac{2\mathcal{N}_{\mathcal{T}}}{n} = \lim_{n\to\infty} \frac{O(\mathcal{N}_{\mathcal{T}})}{n} = 0
\end{aligned}
$$

$$
\begin{aligned}
skew(FY(X)) &= \lim_{n\to\infty} \frac{\frac{1}{n}\sum_{i=1}^{n}[\Phi^{-1}(r_i/(n+1))]^3}{sd(FY(X))^3} \\
&\leq \lim_{n\to\infty} \frac{2 \cdot \mathcal{N}_{\mathcal{T}} \cdot 1^3}{n \cdot sd(FY(X))^3} \\
&= \lim_{n\to\infty} \frac{O(\mathcal{N}_{\mathcal{T}})}{n} = 0
\end{aligned}
$$

here $\mathcal{N}_{\mathcal{T}}$ is the number of ties.

□

Property 3. *If X is continuous:*

1. *$Skew(FY(X)) = 0$*
2. *Suppose that the median vector M_P of quantile normalization has $|Skew(M_P)| = k$ where $k \geq 0$, then $|Skew(FY(M_P))| \leq k$.*

Proof. 1. If X is continuous, X has no ties, then $Skew(FY(X)) = 0$.

2. X is continuous, then the median array M_P of quantile normalization is also continuous. Then

$$Skew(FY(M_P)) = 0$$

thus

$$|Skew(FY(M_P))| \leq k.$$

□

Property 4. *For discrete case, if vector $\{x_i\}$ (i=1,...,P) has only two values a_1 and a_2, then the skewness of $\{h(x_i)\}$ is the same as the skewness of $\{x_i\}$, where h is any monotonic transformation.*

Proof. For any monotonic transformation $h : \{a_1, a_2\} \to R$, can be represented by the linear transformation

$$h(x) = h(a_1) + \frac{(x - a_1)}{(a_2 - a_1)}(h(a_2) - h(a_1))$$

where we can show

$$h(a_1) = h(a_1) + 0 = h(a_1)$$

$$h(a_2) = h(a_1) + \frac{a_2 - a_1}{a_2 - a_1}(h(a_2) - h(a_1)) = h(a_2)$$

Thus h is a linear transformation when $\{x_i\}$ has only two values and the skewness is invariant under linear transformation. □

Property 5. *Fisher-Yates normalization is better for big data I:*
Fisher-Yates normalization is computationally faster than quantile normalization because it dose not require the computation of the median sample, or array. Therefor is more useful for big data.

Property 6. *Fisher-Yates normalization is better for big data II:*
Suppose that we obtain new observation from the same experiment or from new data that is added. Quantile normalization requires the re-computation of the median array which has changed with the new data. This may introduce changes in the analysis which are unsettling. However, Fisher-Yates normalization that's not change anything so the analysis from the prior data remains untouch. It also saves additional computing time thus good for big data.

2.4 Simulation Study for Two-Group Gene Comparative Experiments

The objective of many microarray experiments is to detect different gene expression levels across two or more conditions. For example for people who have lung cancer, we would like to differentiate the gene expression levels of lung tissue with cancer cells with lung tissue with normal cells. Due to the complexity of microarray experiments, we consider a simple and most common case: a comparative experiment to compare two groups.

2.4.1 Notations

Assuming we have two phenotypic groups of gene, say two groups of microarrays (Group1 and Group2); there are n_1 microarrays in Group1 and n_2 microarrays in Group2. In our simulation study, Group1 represents the normal microarrays, which will be simulated from a fixed distribution, while Group2 is the group of interest, for example, the disease group.

Let x_{ij}^k represent the intensity measure of the ith gene in the jth microarray from the kth group, where k=1,2, i=1,...,I, and j=1,...,n_k. The normalized counterparts are written as x_{ij}^{*k}. In addition, let μ_i^k and σ_i^k be the mean and standard deviation of the ith gene in the kth group. Also the normalized intensity observations are denoted as μ_i^{*k} and σ_i^{*k}.

2.4.2 Hypothesis Testing

For microarray data, to test if there is any difference between two groups, we need to construct I null hypotheses (The same I genes in each group). In practice, gene differential expression level and variance are unlikely to be constant. Due to the complexity of the microarray experiment, the variation depends very much on the measurement accuracy. However, there is a trade off between variance and mean difference when our interest is the statistical power. In our simulation study, we thus fix the gene expression variance, and test the equality of gene expression means between two groups:

$$H_0\text{: } \mu_i^1 = \mu_i^2$$

$$\text{vs } H_1\text{: } \mu_i^1 \neq \mu_i^2$$

where i=1,...,I.

Genes in the normal group (Group1) are simulated from a fixed distribution where $\mu_i^1 = \mu_0$, i=1,...,I. Genes in the group of interest (Group2) are generated from three hypotheses:

- G_0^2: takes up 70% of gene with $\mu_i^2 = \mu_i^1 = \mu_0$, $i \in [1, ...,\mathrm{I}]$.
- G_1^{2+}: takes up 15% of gene with $\mu_i^2 > \mu_0$, $i \in [1, ...,\mathrm{I}]$.
- G_1^{2-}: takes up 15% of gene with $\mu_i^2 < \mu_0$, $i \in [1, ...,\mathrm{I}]$.

2.4.3 Simulation Study for Normal Distribution Data

In our simulation study, each microarray has $N = 10,000$ genes or probes. This number N is not as big as the number of features as most genomic data sets that can goto $50,000$ and millions, but for simulation purpose, we think it is a reasonable number. Without loss of generalization, we simulate $10,000$ observations from $Normal(\mu_i^1 = 0, 1)$ for genes in Group1. For three parts of Group2, we simulate:

- G_0^2: $ng_0 = 7,000$, $x_{ij}^2 \sim Normal(0,1)$.
- G_1^{2+}: $ng_1^+ = 1,500$, $x_{ij}^2 \sim Normal(\mu_i^{2+}, 1)$.
- G_1^{2-}: $ng_1^- = 1,500$, $x_{ij}^2 \sim Normal(\mu_i^{2-}, 1)$.

For both groups, the variance is fixed at $\sigma^2 = 1$ and the α level is controlled at 5%. To simplify but without loss of generality, we set $n_1 = n_2 = n$. Table 2.1 lists the power and Type I error for different mean pairs (μ_i^{2+}, μ_i^{2-}) and also for each pair lists the results with different number of observations N.

From Table 2.1, we compare the power of the two-sample t-test across data sets that have been normalized using Fisher-Yates or quantile normalization with the old standard which in this case is the identity transformation. Notice, the identity transformation is not a choice for real data, because the data always need to be normalized. In Table 2.1, we see that all the normalization methods produce reasonable type I error, though have a small loss of type I error for both Quantile and Fisher-Yates normalization, and the power of Fisher-Yates and quantile are almost the same to true normalization.

To make Table 2.1 more intuitive, we use Figure 2.1 to capture the relationship between power and group size for each method. The red line chart is the power line for method 1. When sample size is small, power for method1 is always a little bit higher than other methods, while the powers for all three methods will go to

Table 2.1: $H_0 : Normal(\mu_i^1 = 0, 1)$ vs $H_1 : Normal(\mu_i^2, 1)$

$\mu_i^{2-} = -0.5$, $\mu_i^{2+} = 0.5$						
	True Normalization		Fisher-Yates		Quantile Normalization	
Group Size	Power	TypeI Error	Power	TypeI Error	Power	TypeI Error
n=6	0.1143	0.0506	0.1107	0.0501	0.1103	0.0497
n=10	0.2017	0.0463	0.2000	0.0454	0.2007	0.0453
n=20	0.3317	0.0499	0.3233	0.0503	0.3233	0.0504
n=50	0.6900	0.0470	0.6733	0.0464	0.6737	0.0463
n=100	0.9423	0.0524	0.9370	0.0526	0.9370	0.0527
$\mu_i^{2-} = -1$, $\mu_i^{2+} = 1$						
	True Normalization		Fisher-Yates		Quantile Normalization	
Group Size	Power	TypeI Error	Power	TypeI Error	Power	TypeI Error
n=6	0.3387	0.0516	0.3027	0.0526	0.3030	0.0521
n=10	0.5597	0.0516	0.5080	0.0513	0.5070	0.0511
n=20	0.8650	0.0467	0.8123	0.0457	0.8123	0.0460
n=50	0.9993	0.0503	0.9970	0.0509	0.9970	0.0506
n=100	1.0000	0.0527	1.0000	0.0527	1.0000	0.0527
$\mu_i^{2-} = -1.5$, $\mu_i^{2+} = 1.5$						
	True Normalization		Fisher-Yates		Quantile Normalization	
Group Size	Power	TypeI Error	Power	TypeI Error	Power	TypeI Error
n=6	0.6517	0.0469	0.5200	0.0484	0.5193	0.0486
n=10	0.8860	0.0533	0.7820	0.0549	0.7817	0.0547
n=20	0.9943	0.0493	0.9817	0.0487	0.9813	0.0493
n=50	1.0000	0.0520	1.0000	0.0534	1.0000	0.0526
n=100	1.0000	0.0493	1.0000	0.0491	1.0000	0.0490
$\mu_i^{2-} = -2$, $\mu_i^{2+} = 2$						
	True Normalization		Fisher-Yates		Quantile Normalization	
Group Size	Power	TypeI Error	Power	TypeI Error	Power	TypeI Error
n=6	0.8700	0.0499	0.6893	0.0534	0.6827	0.0530
n=10	0.9877	0.0510	0.9143	0.0524	0.9133	0.0524
n=20	1.0000	0.0493	0.9983	0.0504	0.9987	0.0503
n=50	1.0000	0.0523	1.0000	0.0516	1.0000	0.0514
n=100	1.0000	0.0519	1.0000	0.0529	1.0000	0.0524

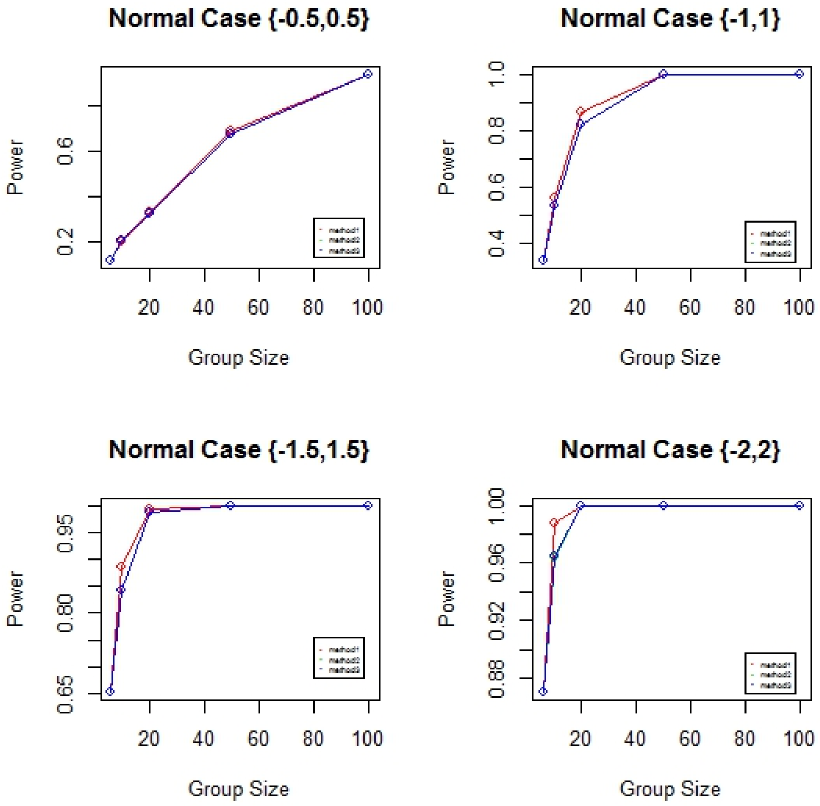


Figure 2.1: $H_0 : Normal(\mu_i^1 = 0, 1)$ vs $H_1 : Normal(\mu_i^2, 1)$

1 eventually when we increase our sample size.

2.4.4 Simulation Study for Gamma Distribution Data

To summarize, in the normal case, all the methods are similar and work reasonably well. But what if our data is generated from a Gamma distribution? Still we simulate $10,000$ observations from $Gamma(\mu_i^1 = 3, 1)$ for genes in Group1. For Group2, similarly:

- G_0^2: $ng_0 = 7,000$, $x_{ij}^2 \sim Gamma(\mu_i^1 = 3, 1)$.
- G_1^{2+}: $ng_1^+ = 1,500$, $x_{ij}^2 \sim Gamma(\mu_i^{2+}, 1)$.
- G_1^{2-}: $ng_1^- = 1,500$, $x_{ij}^2 \sim Gamma(\mu_i^{2-}, 1)$.

Table 2.2: $H_0 : Gamma(\mu_i^1 = 3, 1)$ vs $H_1 : Gamma(\mu_i^2, 1)$

$\mu_i^{2-} = 2$, $\mu_i^{2+} = 4$						
	True Normalization		Fisher-Yates		Quantile Normalization	
Group Size	Power	TypeI Error	Power	TypeI Error	Power	TypeI Error
n=6	0.1480	0.0460	0.1543	0.0473	0.1403	0.0459
n=10	0.2500	0.0460	0.2633	0.0480	0.2410	0.0470
n=20	0.4467	0.0497	0.4677	0.0509	0.4283	0.0499
n=50	0.8233	0.0527	0.8567	0.0550	0.8083	0.0526
n=100	0.9770	0.0541	0.9883	0.0547	0.9733	0.0539
$\mu_i^{2-} = 2$, $\mu_i^{2+} = 5$						
	True Normalization		Fisher-Yates		Quantile Normalization	
Group Size	Power	TypeI Error	Power	TypeI Error	Power	TypeI Error
n=6	0.2607	0.0459	0.2597	0.0491	0.2387	0.0449
n=10	0.4377	0.0509	0.4287	0.0543	0.3963	0.0533
n=20	0.6933	0.0449	0.7110	0.0491	0.6710	0.0490
n=50	0.9487	0.0514	0.9707	0.0536	0.9713	0.0631
n=100	0.9953	0.0493	0.9987	0.0600	0.9977	0.0893
$\mu_i^{2-} = 2$, $\mu_i^{2+} = 6$						
	True Normalization		Fisher-Yates		Quantile Normalization	
Group Size	Power	TypeI Error	Power	TypeI Error	Power	TypeI Error
n=6	0.3943	0.0446	0.3543	0.0516	0.3273	0.0466
n=10	0.5710	0.0439	0.5520	0.0497	0.5227	0.0510
n=20	0.7490	0.0471	0.7923	0.0591	0.7750	0.0679
n=50	0.9487	0.0544	0.9810	0.0711	0.9800	0.1063
n=100	0.9953	0.0501	0.9990	0.1001	0.9990	0.1816

Table 2.2 lists the output for power and Type I error with different mean pairs (μ_i^{2+}, μ_i^{2-}) (α level is controlled at 5%) and for each pair displays the results with different number of gene observations n.

It appears from table 2.2 that when the data is not normally distributed, Fisher-Yates method is better than Quantile normalization. In this gamma case, the power for Fisher-Yates is also higher than non-normalization in most cases.

Also from figure 2.2 we find the red curve (curve for method 1) is generally below the other two curves. So we could say identity transformation is optimal when our data for simulation is i.i.d normally distributed. But if it appears when

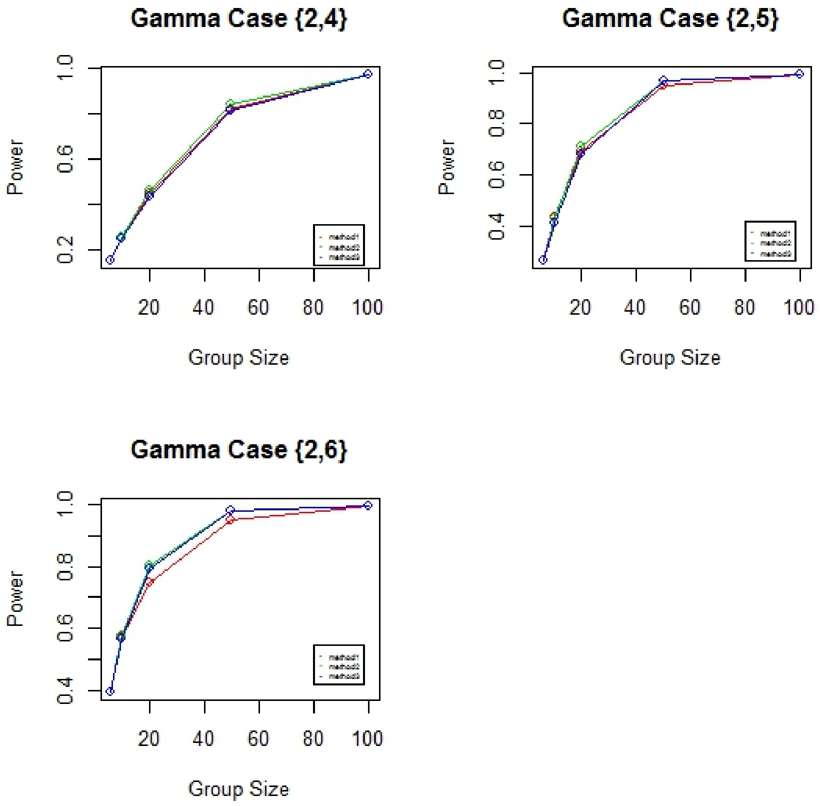


Figure 2.2: $H_0 : Gamma(\mu_i^1 = 3, 1)$ vs $H_1 : Gamma(\mu_i^2, 1)$

the data is not symmetric, Fisher-Yates normalization works better. Also there is a loss of type I error in quantile and Fisher-Yates normalization due to over-fitting.

2.4.5 Discussion

In this section, we compare Fisher-Yates transformation to quantile normalization and the true transformation, which in this case is identity. From our simulation example, we could see all methods are similar and work reasonably well when data is generated from normal distribution. Notice that in practice, the true transformation is always unknown and unlikely to be identity transformation. That's why we use Fisher-Yates and quantile normalization to preprocess the data (the true transformation here represents the optimal method you can do). And we can find when data is non-normal, Fisher-Yates is more successful at reducing skewness and thus improve the power.

2.5 Simulation Study for Scoring Scale Data

Fisher-Yates normalization handles skewness and outliers better than quantile normalization and as a result it increases the power to detect genes that are differentially expressed between arrays and also it gets better classification results.

We generate here three different cluster patterns similar to the questionnaire scoring scale data which we use as the real data example: generally horizontal, oblique with positive slope, oblique with negative slope. We first define six probability vectors which will be used to generate data: high-top and low-top put more weight on top(large) numbers; high-middle and low-middle put more weight on middle numbers; high-bottom and low-bottom put more weight on bottom(small) numbers. The specific probability vectors we used are listed in Table 2.3 (p_i is the probability used to generate scoring scale i), also in Figure 2.3 gives the probability trend for each pattern.

Table 2.3: Probability Vectors for Data Generation

Name	p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
High-top	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.1	0.63
Low-top	0.02	0.02	0.02	0.02	0.02	0.02	0.07	0.17	0.35	0.24	0.05
High-middle	0.00	0.01	0.01	0.04	0.05	0.20	0.22	0.17	0.10	0.10	0.10
Low-middle	0.10	0.10	0.10	0.17	0.22	0.20	0.05	0.04	0.01	0.01	0.00
High-bottom	0.10	0.24	0.35	0.12	0.07	0.02	0.02	0.02	0.02	0.02	0.02
Low-bottom	0.63	0.10	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03

1200 observations with 10 columns are generated from the above six probability vectors, where each column corresponds to one out of 10 questions. Observations are generated similarly to the patterns of the three clusters that we obtained from our pain scale data set. After we generate this data we need to

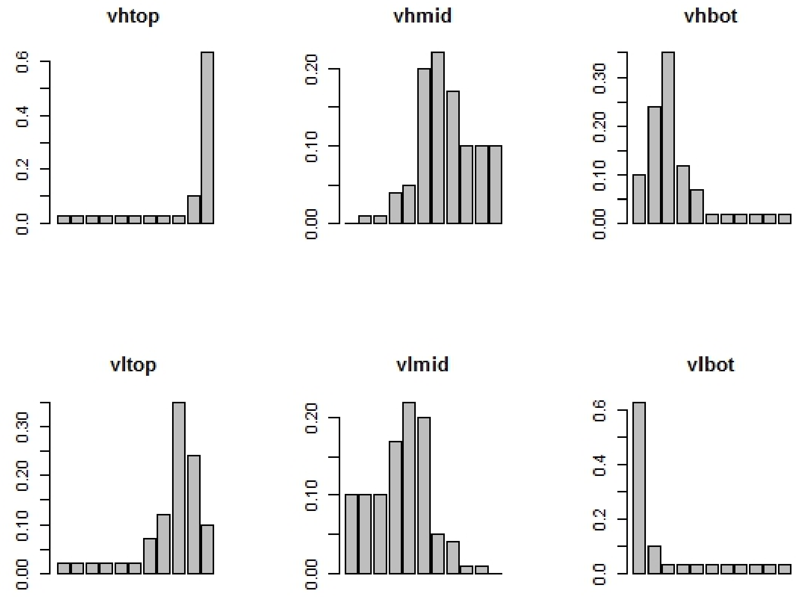


Figure 2.3: Probability Trend Barchart for Different Patterns.

perform the following three steps:

1. First normalize the data separately with the Z-score method, quantile normalization method and Fisher-Yates normalization method;
2. Perform a hierarchical clustering with “Ward” distance, and set the number of clusters to be three;
3. We compare the cluster predictions that were obtained from the hierarchical clustering to the real clusters, and then calculate the number of true classification.

The numbers of true classification for each method are listed in Table 2.4. Table 2.4 indicates quantile normalization and Fisher-Yates normalization are better classifiers than Z-score and Identity transformation for our simulated data.

Table 2.4: True Classification Numbers

Normalization Method	**True Classification Number**
Identity Transformation	491
Z-score Transformation	610
Quantile Normalization	706
Fisher-Yates Normalization	708

2.6 Normalization Application on Neuropathic Pain Data

2.6.1 Data Description

Neuropathic Pain Symptoms Inventory (NPSI) questionnaire consisted of 10 descriptors on a 0-10 pain scale, 0 means no pain and 10 means worse imaginable pain. The descriptors measured a range of symptoms: Burning, Pressure, Squeezing, Electric Shocks, Stabbing, Evoked by Brushing, Evoked by Pressure, Evoked by Cold Stimuli, Pins and Needles, Tingling.

Four clinical studies were conducted for testing the treatment effect of an undisclosed drug. The results of the three studies turned out to be negative, because they found no difference between treatment and placebo, and one was positive. This was followed by an attempt to find subgroups of the data here the pain scale questionnaire data by standardizing or normalizing the answers using z-sores but that was also unsuccessful. Our objective here is to find out if we improve the data by applying our F-Y normalization and hopefully we will be able to find subgroups with positive treatment response. Clinicians and physicians believe that the treatment effect should work for large subgroups of population that follow the specific pain pattern.

Our hope is that by using FY normalization of this questionnaire data we will be able to find subpopulation that are very responsive to the treatment.

2.6.2 Normalization Outcomes

Figure 2.4 displays boxplots of the raw symptom descriptors grouped by patient and sorted by the mean NPSI. To make the figure workable we skip 19 out of 20 patients. We observe differences in shape and strong skewness on the boundaries. When we applied the z-scores normalization to this data in Figure 2.5 we could see the skewness is not removed.

Figure 2.6 shows the boxplots of the quantile normalized symptom descriptors grouped by patient and sorted by the mean NPSI. It seems that quantile normalization could improve the skewness but it has a problem with the median subject because of the small number of predictors the “median subject” has a small range(score from 0 to 8). This effect happens also on microarray data but is less pronounced as the number of predictors is much larger.

Figure 2.7 shows the boxplots of the Fisher-Yates normalization grouped by patient and sorted by the patient median. Fisher-Yates can improve the skewness while doesn’t suffer from the median subject problem as quantile normalization does.

2.6.3 Skewness Reduction

As show in **Property 3**, the skewness of data after Fisher-Yate’s transformation is always 0 if the random variable is continuous. But in reality we may have data with many ties like this example, and therefore FY normalization not always produce data with 0 skewness. But we expect that overall the skewness of the data should be reduced more than other methods by FY. In our example data, you can see this is the case

The skewness improvement (or normalization result) between Quantile and Fisher-Yates is not obvious from boxplots listed(Figure 2.4 to Figure 2.7). Thus we calculate the skewness for each method and summarized in Table 2.5.

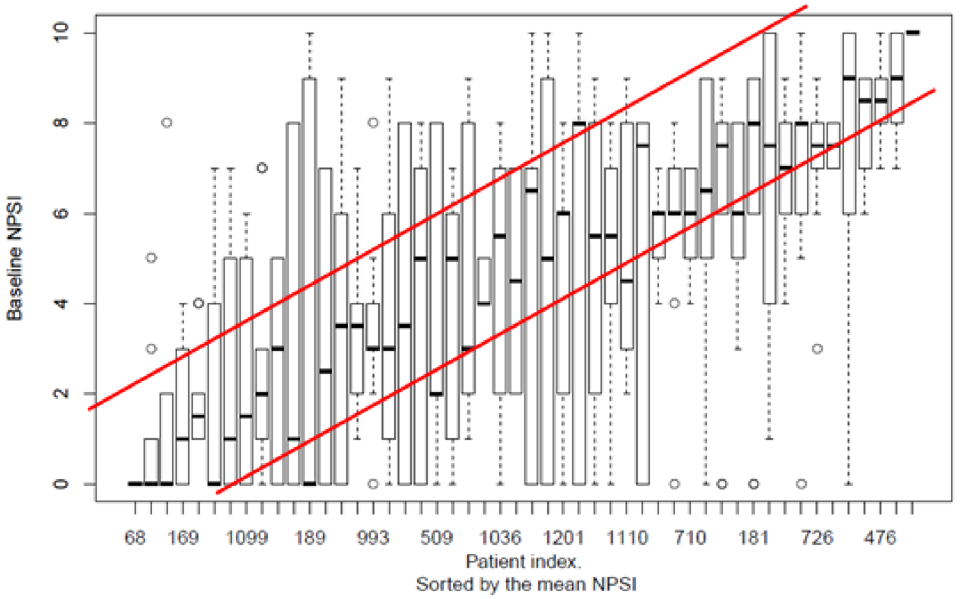


Figure 2.4: Boxplots of the raw symptoms grouped by the patient and sorted by the patient median. 1 out of every 20 patients are plotted.

Table 2.5: Skewness Reduction Comparison

	$\lvert Sk(Z)\rvert > \lvert Sk(FY)\rvert$	$\lvert Sk(Qn)\rvert > \lvert Sk(FY)\rvert$	$\lvert Sk(Z)\rvert > \lvert Sk(Qn)\rvert$
Ture	864	765	766
False	287	386	385

where $|Sk(Z)|$ is the absolute skewness value of z-score transformation, $|Sk(Qn)|$ is the absolute skewness of Quantile normalization and $|Sk(FY)|$ is the absolute skewness of Fisher-Yates.

We can see the Skewness Comparison Ratio between these three normalization methods:

- $R(|Skew(Z)| > |Skew(FY)|) \approx 75\%$
- $R(|Skew(Qn)| > |Skew(FY)|) \approx 66\%$
- $R(|Skew(Z)| > |Skew(Qn)|) \approx 67\%$

So Fisher-Yates transformation is most successful at reducing skewness effect: 75% times the skewness of Fisher-Yates is smaller than Z-score, and 66% times

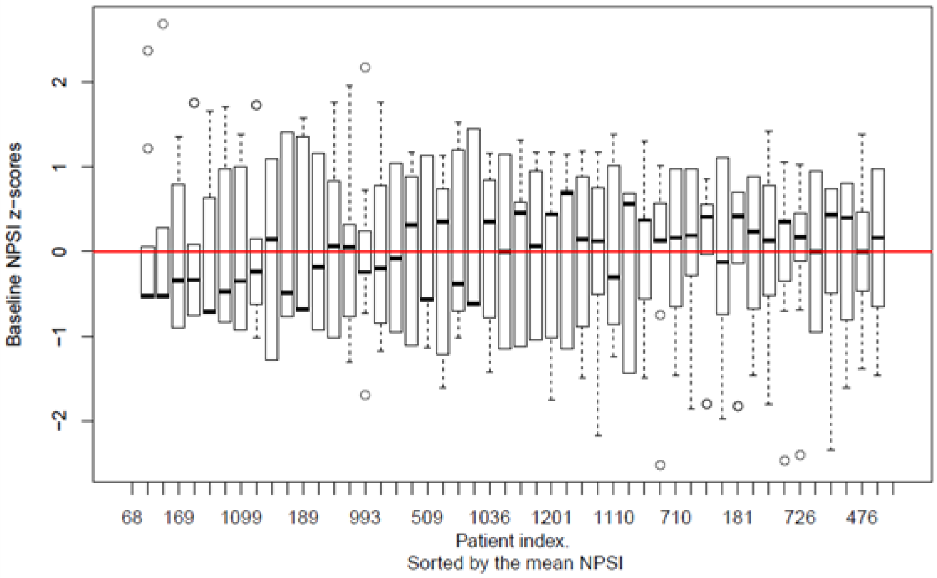


Figure 2.5: Boxplots of the z-scores on symptoms grouped by the patient and sorted by the patient median.

the skewness of Fisher-Yates is smaller than Quantile normalization.

2.7 Normalization Application on Sialin Data

2.7.1 Data Description

As described by Amaratunga and Cabrera (2004), Sialin data are gene expressions collected from a group of mice whose Slc17A5 gene was knocked out compared to gene expression of a group of "normal" mice. Slc17A5 is the gene responsible for Sialin production which is involved in the development of the mice. In the experiment, RNA samples were derived for each group from newborn and 18-day-old mice. There are total 24 observations which corresponds to 2 groups by 2 time points by 6 biological observations. The gene expressions were generated by hybridization of the observations using 24 Affymetrix Mouse430-2 gene chips. Each chip generated the gene expression profile of the sample, which contains 45,101 gene expressions.

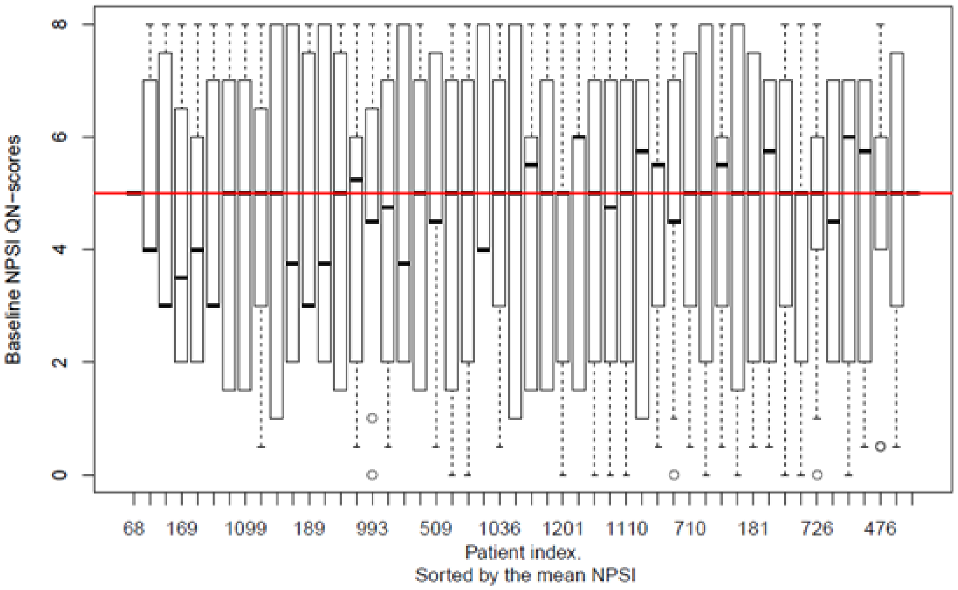


Figure 2.6: Boxplots of the Quantile Normalization of the symptoms grouped by the patient and sorted by the patient median.

2.7.2 Normalization Results

We conducted quantile normalization and Fisher-Yates transformation separately on the Sialin data, want to test the significance of the gene expression based on each method. There are two groups of 6 observations each for the 18 day data. In order to perform t-tests to compare the two groups after the normalization, we need to assume the observations in each group are normally distributed. This is actually not likely to be always true, because quantile normalization approximately preserves the shape of the original distribution which is unlikely to be normal. Thus our hypotheses will be:

$$H_0\colon\ \mu_i^1 = \mu_i^2$$

$$\text{vs } H_1\colon\ \mu_i^1 \neq \mu_i^2$$

where i=1,...,I, and μ^1 is the mean for group 1, and μ^2 is the mean for group 2.

We calculate the p-value at significance level α for testing for differential expression of each gene between group 1 and group 2. The most basic statistical

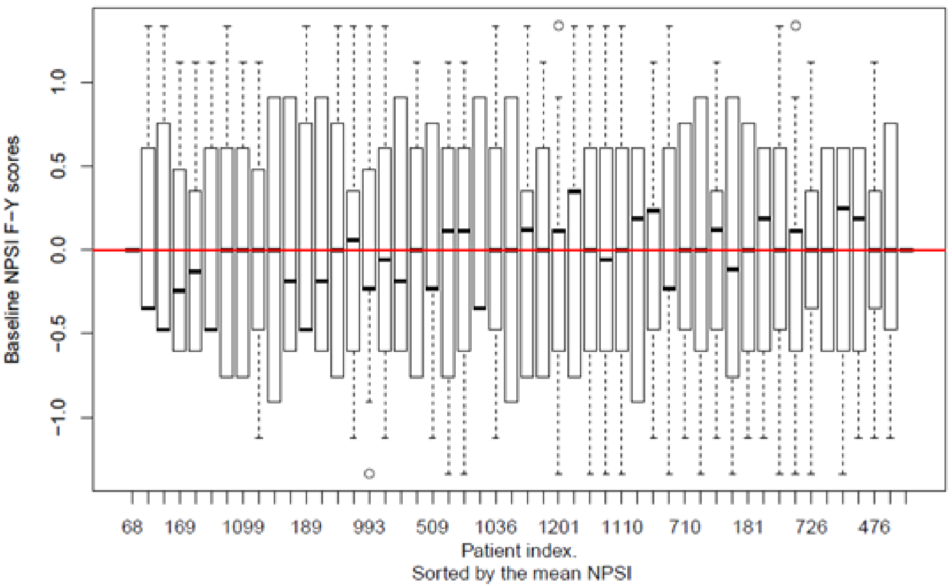


Figure 2.7: Boxplots of the Fisher-Yates Algorithm on symptoms grouped by the patient and sorted by the patient median. This improve the skewness and gives a more satisfactory shape result.

test for comparing two groups is the two-sample t-test:

$$T_e = \frac{|\bar{x_1}| - |\bar{x_2}|}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{2.10}$$

where

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2} \tag{2.11}$$

is the pooled estimate of variance.

After normalization, under the assumptions that the populations are Gaussian and the variances are homoscedastic, the null distribution of T_e is then a t-distribution with degrees of freedom $v = n_1 + n_2 - 2$. The p-value for each gene is calculated as $p_e = Prob(|T_e| > T_{eobs})$, where T_{eobs} is the observed value of T_e. A gene is declared as significant at level α if $p_e < \alpha$.

The significance table at significance level $\alpha = 0.01$ is given in Table 2.6 and significance level $\alpha = 0.05$ is given in Table 2.7. Tables listed below indicate Fisher-Yates transformation could detect more significantly differentially expressed gene than quantile normalization.

Table 2.6: Significance Level $\alpha = 0.01$

Fisher-Yate's Transformation	Quantile Normalization	
	Not Significant	Significant
Not Significant	24322	268
Significant	329	20182

Table 2.7: Significance Level $\alpha = 0.05$

Fisher-Yates Transformation	Quantile Normalization	
	Not Significant	Significant
Not Significant	23701	261
Significant	338	20801

We also present the boxplots of gene expression level in log-scale for probes of RMA18 under raw data and normalized data. From figure 2.8 we could see most of the log-scaled gene levels are in range 4 to 8 with median array around 6. Z-score normalization makes the data more normally distributed but very skewed. Fisher-Yate's transformation handles the skewness and the normalization result is quite good. In figure 2.8 here quantile normalization result is the same as raw data, because our data set has already been pre-processed with quantile normalization.

2.8 Discussion and Conclusion

The main reason to use quantile normalization is that the median chip is similar in shape to the individual chips, and it seems a reasonable idea to normalize to a function of shape that is close to the shape of the real data.

But in the case of small number of genes or small number of questions per subject the median chip is not informative because it does not necessarily have similar shape than the individual observations.

Fisher- Yates normalization is a very simple algorithm which normalizes each subject with multiple questions. It is not uncommon that data comes measured

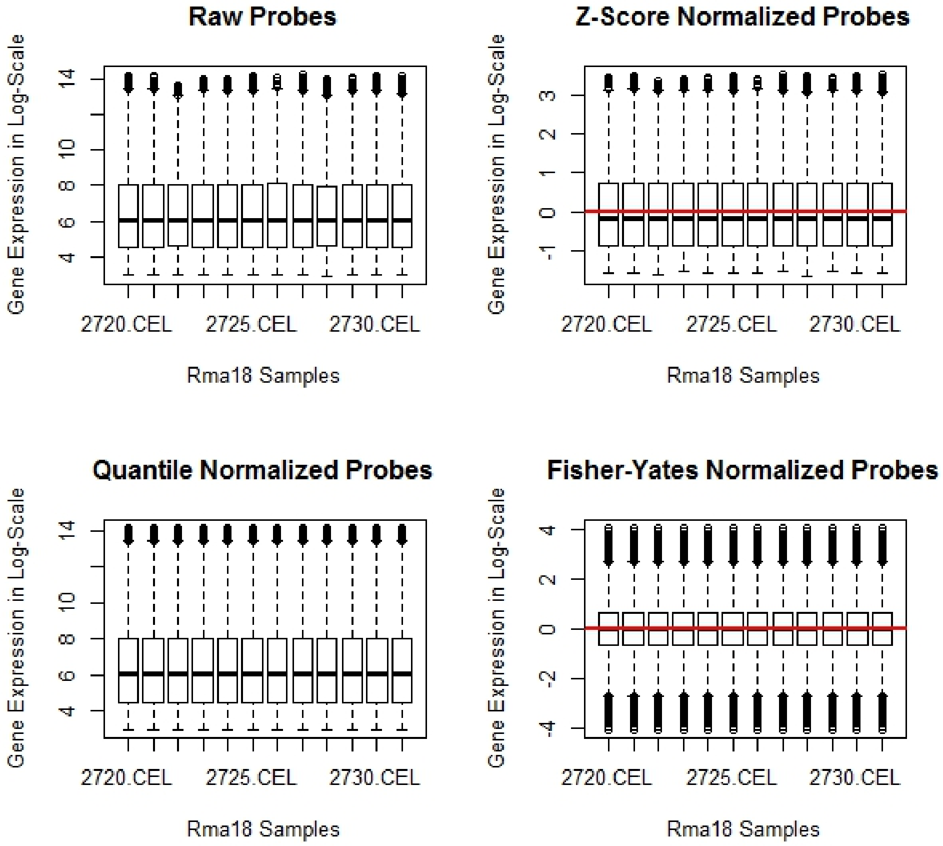


Figure 2.8: Gene Expression Level in Log-scale for probes of RMA18

on different scales. Here we consider the case when multivariate observations of similar quantities are measured in scales that are dependent on the observation. For example questionnaire data where the scale of the answers depends on the individuals perception, or microarray data where each microarray has its own scale or images where the luminosity of the image depends on the light level of the picture. We compare traditional z-scores normalization with quantile normalization that is the standard in genomic and with Fisher-Yates normalization which is our new proposal. We show that Fisher-Yates is more efficient when testing hypothesis following the normalization procedure when compared to the other two.

For micro-array normalization, quantile normalization is standard but there maybe situations where Fisher-Yates is a better alternative. In our questionnaire data example it appears that the Fisher-Yates algorithm does a better job at normalizing data.

For future work, we could concentrate on applying Fisher-Yates normalization to imaging data and educational testing data, or we can also explore the application on combined date sets from different sources.

Chapter 3

Hierarchical Clustering Tree Prediction; An Unsupervised Algorithm for Predicting New Data based on the Established Clusters

In the previous chapter we applied Fisher-Yates normalization to the clinical outcomes data which ended up generating a set of clusters, some of which had a positive response to treatment. In this Chapter we validate our method by adding data from a new study that was obtained after performing the previous analysis. For this we develop a new method for incorporating new data to the existing hierarchical clustering partition that was obtained earlier using older training data. The issue is on how to use the existing hierarchical clustering to predict the clusters for the new data. The standard prediction method is to assign the new observation to the closest cluster using inter-cluster distance between that observations and the existing clusters. Here we derive a novel method called hierarchical clustering tree prediction method (HCTP), which will use the existing hierarchical tree to insert the new observation into the appropriate cluster. This new method depends on the distance and inter-cluster distance which was used to generate the hierarchical tree. We will study the most commonly used distance measures used for hierarchical clustering to compare our HCTP to the standard prediction method.

3.1 Introduction

Cluster analysis, also known as data segmentation, is an *unsupervised* classification method to split the data into clusters or segments, so that within cluster objects are as homogenous as possible and between clusters are as heterogeneous as possible.

Clustering methods can be classified as hierarchical (nested) or partition (unnested). However, they all depend on a dissimilarity or a similarity measure, depict the "distance" of two observations: how far or how close, the observations are from each other. Given two observations, x_1 and x_2, there are many choices of the dissimilarity measure $D(x_1, x_2)$, generally D obeys the following rules (Amaratunga and Cabrera (2004)):

1. $D \geq 0$;
2. $D = 0$ if and only if $x_1 = x_2$;
3. D increase if x_1 and x_2 are further apart;
4. $D(x_1, x_2) = D(x_2, x_1)$.

Some choices for D also satisfy either

1. the triangle inequality, $D(x_1, x_2) \leq D(x_1, x_3) + D(x_3, x_2)$; OR
2. the ultra-metric inequality, $D(x_1, x_2) \leq max(D(x_1, x_3), D(x_2, x_3))$.

The most widely used dissimilarity measure is the Euclidean distance, D_E:

$$D_E(x_1, x_2) = \sqrt{\sum_{j=1}^{p} (x_{1j} - x_{2j})^2} \tag{3.1}$$

In this documnet we apply hierarchical clustering i based on Euclidean distance D_E.

3.1.1 Motivation

We recently analyzed a data set about the treatment effect of Lyrical on neuropathic pain patients data from four randomized clinical trials (Freeman et al. (2014)). After the data on baseline symptoms of Neuropathic pain was normalized by Fisher-Yates we applied hierarchical clustering, then identified three clusters. Once the clusters were established, a new clinical trial data became available for us. We wanted to assign new patients from the recent trial into the established clusters from the previous four trials.

Here we present a novel prediction method based on the usual hierarchical clustering methodology which will return a dendrogram with the original hierarchical cluster plus the insertion points of the new data, allowing us to directly observe and interpret our prediction results. The standard clustering prediction method is to use inter-cluster distance between new observations and the final cluster configuration for the training data, and assign to the new observation the clusters with the smallest distance. Also in some cases we use supervised classification methods where the response is the cluster number and the predictors are the cluster variables.

Our method consists of inserting the new observation into the hierarchical tree without changing the tree structure. The basic idea is to include the new observation data with the original data and to perform hierarchical tree clustering until the new observation joins a cluster A. Then we assign the new observation to cluster A', which is the cluster in the original configuration where cluster A falls into. As with the clustering itself, different inter-cluster distances may result in different predictions.

3.1.2 Two Simple Approaches to Prediction

Before we goto HCTP, let's first look at two simple approaches to prediction. Suppose we got 200 random points in $\mathbb{R}^2$ from an unknown distribution, and suppose they are clustered into 2 groups, $\mathfrak{G} = \{red, blue\}$. Then how to predict the color of future points? There are two simple approaches to prediction [Hastie et al. (2009)]: Least squares and Nearest neighbors.

Least Squares

For linear models, we know given a vector of inputs $X^T = (X_1, ..., X_p)$, we predict a response variable Y via

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j = X^T \hat{\beta} \tag{3.2}$$

The prediction method for least square is: we code $Y = 1$ if G is red and $Y = 0$ if G is blue. Then the classification will be

$$\hat{G} = \begin{cases} Red, & \text{if } \hat{Y} > 0.5; \\ Blue, & \text{if } \hat{Y} < 0.5. \end{cases}$$

In this case, the decision boundary is linear given by function (3.3)

$$\{x | x^T \hat{\beta} = 0.5\} \tag{3.3}$$

Nearest-Neighbor Methods

For k Nearest-Neighbor (kNN) prediction, we predict response variable Y via

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \tag{3.4}$$

The classification function is the same:

$$\hat{G} = \begin{cases} Red, & \text{if } \hat{Y} > 0.5; \\ Blue, & \text{if } \hat{Y} < 0.5. \end{cases}$$

The results of these two methods show kNN has far fewer misclassified training observations than linear model Hastie et al. (2009), and the error of misclassification is an increasing function of k. Means if $k = 1$, the misclassification error is minimized.

3.2 Hierarchical Cluster Tree Prediction Analysis on A Set of Dissimilarities and Methods

Hierarchical clustering is one of the most widely used data analysis tools. The idea is to build a tree configuration of data that successively merges similar groups of points. Usually hierarchical clustering will fall into two types: bottom-up or top-down.

Bottom-up clustering (also known as *agglomerative hierarchical clustering*) algorithms are started with each unit situated in its own cluster, and the algorithm grows the hierarchical tree that puts together the observations into different clusters. At each step, the closest pair of clusters is combined and finally the data will falls into one super cluster. At each stage, when agglomerating two clusters, distance between the new cluster and all the others rest will be recalculated according to the particular clustering method being used (for “ward” method, we use Lance-Williams dissimilarity).

Top-down clustering (also known as *divisive hierarchical clustering*) algorithms are initiated all units put together as one cluster. The cluster then is split into two clusters at the next step. Process can be continued until each unit is alone in its own cluster. A serious disadvantage of top-down clustering method is, at the early stage, there are a huge number of ways of splitting the initial cluster (e.g., $2^{G-1} - 1$ in the first stage).

Agglomerative (bottom-up) is more popular and simpler than divisive (top-down), but less accurate. There are some advantages of hierarchical clustering:

I) Partitions can be visualized using a tree structure (a dendrogram), which will provide a useful summary of the data.

II) Does not need the number of clusters as input, it can be decided by visually analyzing the hierarchical tree.

III) Can give different partitions depending on the level-of-resolution we are looking at. However, hierarchical clustering can be very slow, need to make lot of merge/split decisions. In following sections, we will look into different dissimilarity measures of hierarchical clustering.

3.2.1 Model Construction and Algorithm

We here note our training data as $X_{tr} = \{x_{ij}\}, i = 1, 2, ..., N; j = 1, 2, ..., P$. Where n is the number of independent variables and p is the number of features. Likewise, let X_{te} be our testing data. The objective of supervised classification is to use the training data for "training" purposes, that is, to develop a classification rule. The idea is given a new sample x, the classification rule is used to predict, as accurate as possible, the true class of the new sample.

The main idea of our HCTP is:

1. Perform hierarchical clustering on our training data;
2. Add a new observation into the training data and re-perform hierarchical clustering method, up to the point when the new observation joins a cluster A. Follow cluster A in the original hierarchical just adding the new observation until we stop a cluster $A^{'}$;
3. The new observation will be included in cluster $A^{'}$;
4. Repeat the previous steps for all new observations. The resulting hierarchical tree will include the old tree with the insertions of the new observations.

The idea is illustrated roughly in Figure 3.1 and Figure 3.2. We perform hierarchical clustering on the training data and cut into four clusters as show in Figure 3.1. Now if we have a new point, which cluster should we assign it to? For the traditional method, we calculate the distance between the new point to the four clusters, assigning it to the closest cluster. For our HCTP, we perform hierarchical clustering again on the combined data set (training data and the new point). For example x_1 and x_2 join together as one cluster first, then x_3 joins with them. And then our new point joins with x_8, we stop and assign this new point to cluster 4 (the same cluster as x_8) with respect to the tree configuration of the training data.

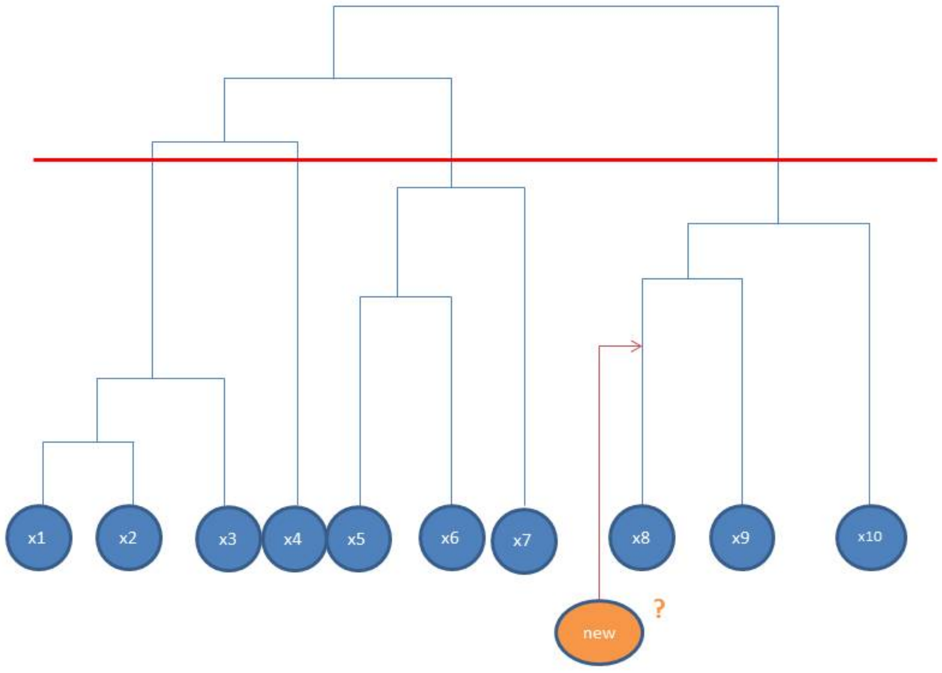


Figure 3.1: Algorithm Illustration Graph 1.

If a new point joins a cluster below the red line stage (before growing to 4 clusters), we take action as illustrated in Figure 3.1. But what if this new point doesn't join any existing clusters while we already established four clusters, like the new point 3 in Figure 3.2? This may happen in real occasion, but we still need a strategy. In this situation, we assign this new point to the cluster with minimum distance. That is we use the same algorithm as the traditional method in this special case.

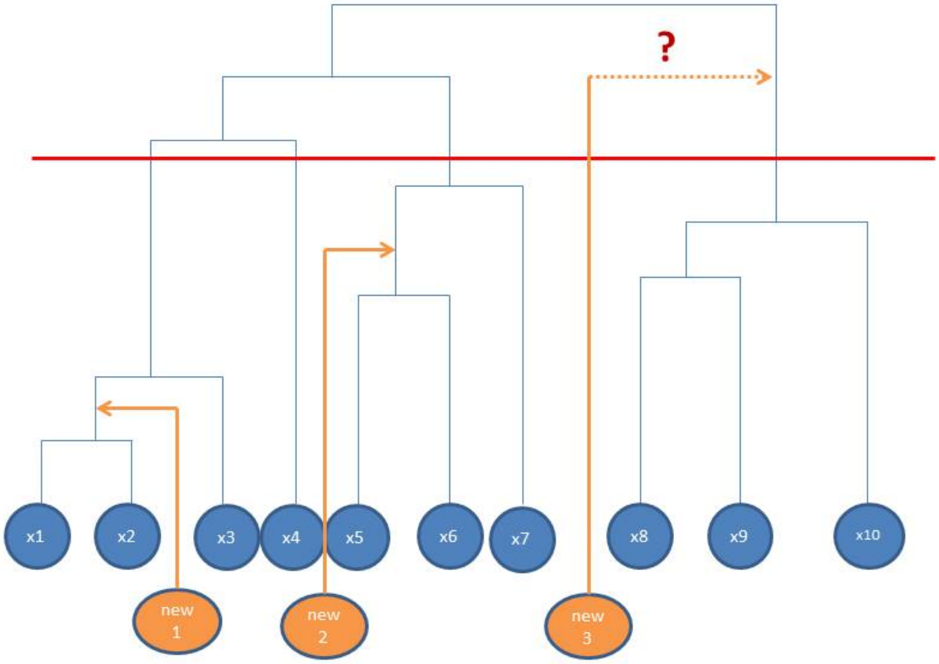
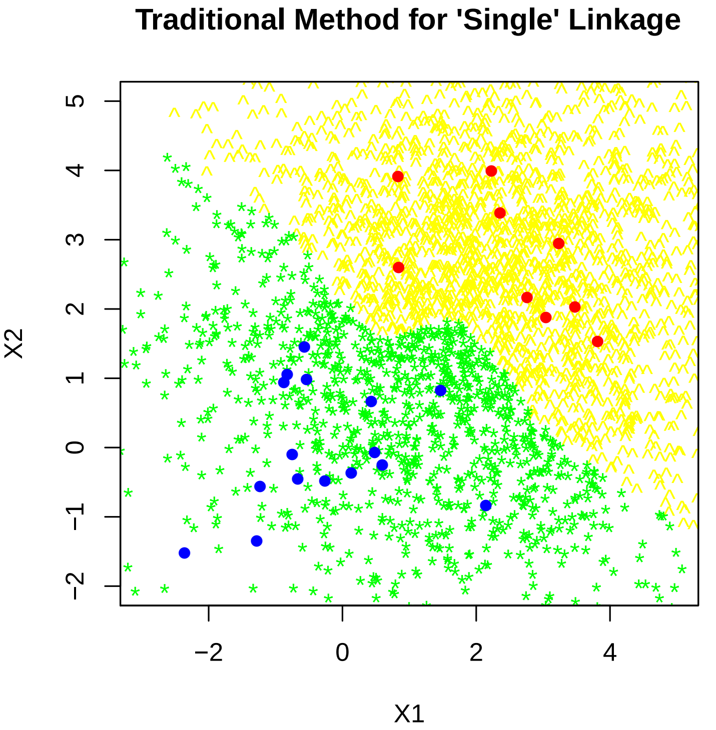


Figure 3.2: Algorithm Illustration Graph 2.

3.2.2 Single Linkage Prediction

The *single linkage hierarchical clustering* (also known as *nearest neighbor clustering*), is one of the oldest methods among cluster analysis and was suggested by researchers in Poland in 1951. The definition is: the distance between two clusters is the smallest dissimilarity measure between any two objects in different



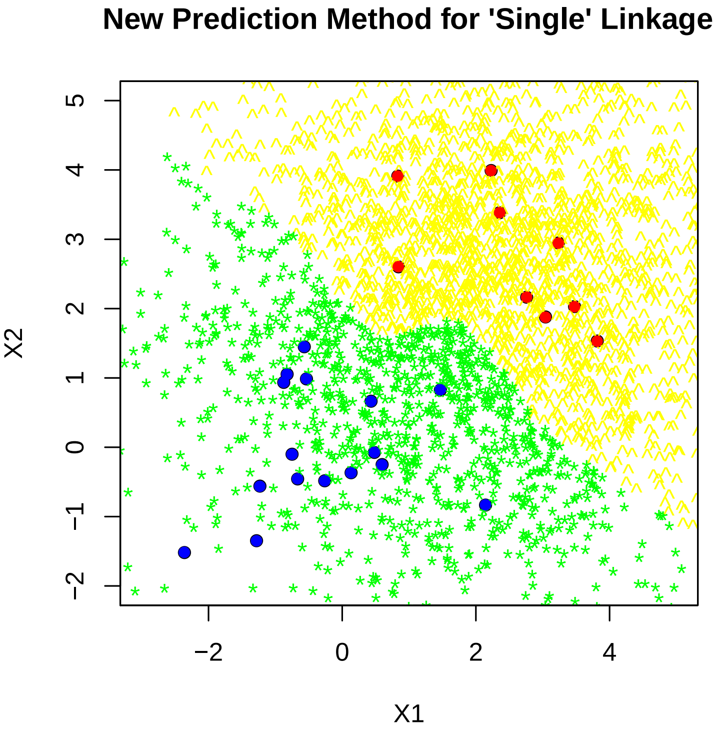


Figure 3.3: Traditional and New Prediction Methods with “Single” Linkage Distance.

clusters. Mathematically, the linkage function $D(X, Y)$ can be expressed as:

$$D(X, Y) = \underset{x \in X, y \in Y}{min} d(x, y) \tag{3.5}$$

where X and Y represent two clusters, while $d(x, y)$ represents distance between two elements $x \in X$ and $y \in Y$. The merge criterion is local, that means we only need to focus on the area where two clusters are closest to each other at each stage, and ignore the overall clusters or other more further parts.

There is a well known drawback called *chaining phenomenon*: two clusters may be forced together due to some single elements being close to each other, though actually some elements in each may be very far away to each other. This disadvantage promoted a bunch of other hierarchic and non-hierarchic methods (Lance and Williams 1957; Sneath 1957).

So in this case, our new observation will always cluster with the closest observation. We don't need to run the tree again, but just look for the cluster of the nearest neighbor to the new data. That means there is no difference between our HCTP and traditional method while the distance is "Single" linkage. Look at Figure 3.3, there are two groups $\mathfrak{G} = \{red, blue\}$, also these two groups correspond to two well separated clusters under "single" linkage, say red=cluster1 and blue=cluster2. The red and blue points are our training data, and yellow and green dots are the prediction results. Both methods predict yellow dots to cluster1 and blue dots to cluster2. From Figure 3.3, we could see the boundary is the same for two different prediction methods.

3.2.3 Complete Linkage Prediction

In *complete linkage hierarchical clustering* (also known as *farthest neighbor clustering*), the distance between two clusters is the largest dissimilarity measure

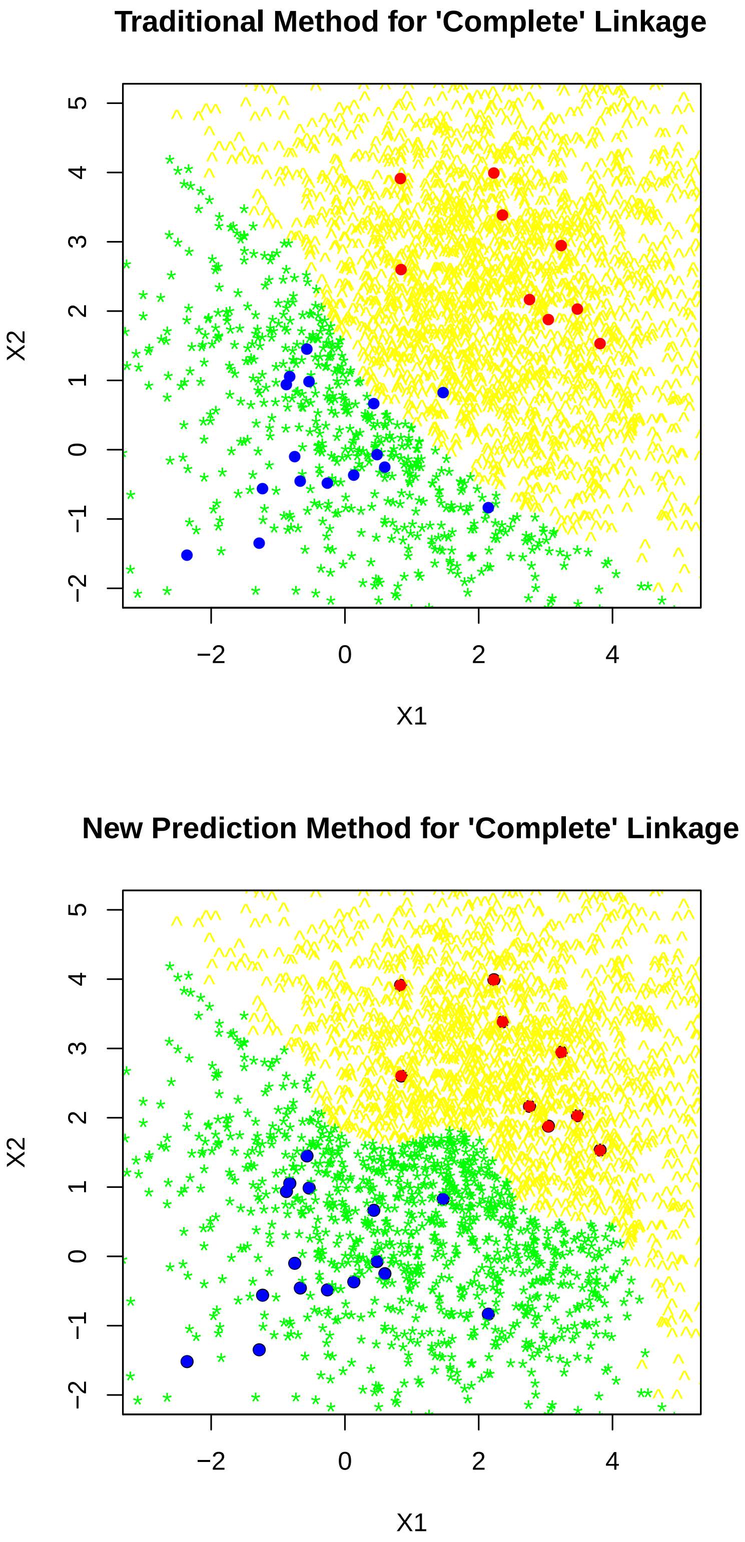


Figure 3.4: Traditional and New Prediction Method with “Complete” Linkage Distance.

between any two objects in different clusters. Mathematically, we define the complete linkage function $D(X, Y)$ as:

$$D(X, Y) = \max_{x \in X, y \in Y} d(x, y) \tag{3.6}$$

similar to the single linkage function, X and Y are two clusters, and $d(x, y)$ is the distance between two elements $x \in X$ and $y \in Y$. This complete linkage merge is non-local. The whole structure of the clustering has influence on our final decisions. Also complete linkage clustering avoids *chaining phenomenon* caused by single linkage.

From Figure 3.4 we could see for "Complete" linkage, our HCTP is different from traditional method, and seems better at classification. We will use the same training set as "Single" linkage, $\mathfrak{G} = \{red, blue\}$, and also red=cluster1, blue=cluster2. From the first graph in Figure 3.4, we could see there is one blue point falls into yellow dots region, which is the prediction region of cluster1. While in the second graph, the boundary classifies two clusters more reasonable.

3.2.4 Average Linkage Prediction

In *average linkage hierarchical clustering*, the distance between two clusters $\mathcal{A}$ and $\mathcal{B}$ is the arithmetic mean of the dissimilarity measures between all pairs of members in different clusters.

$$\frac{1}{|n_1|.|n_2|} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x, y) \tag{3.7}$$

here n_1 is the number of members in $\mathcal{A}$ and n_2 is number of members in $\mathcal{B}$.

From Figure 3.5 and Figure 3.5 we could see for "Average" linkage, our HCTP is also different from traditional method. The training set is unchanged, $\mathfrak{G} = \{red, blue\}$, and red=cluster1, blue=cluster2. The boundary for traditional method is linear, while nonlinear for our method.

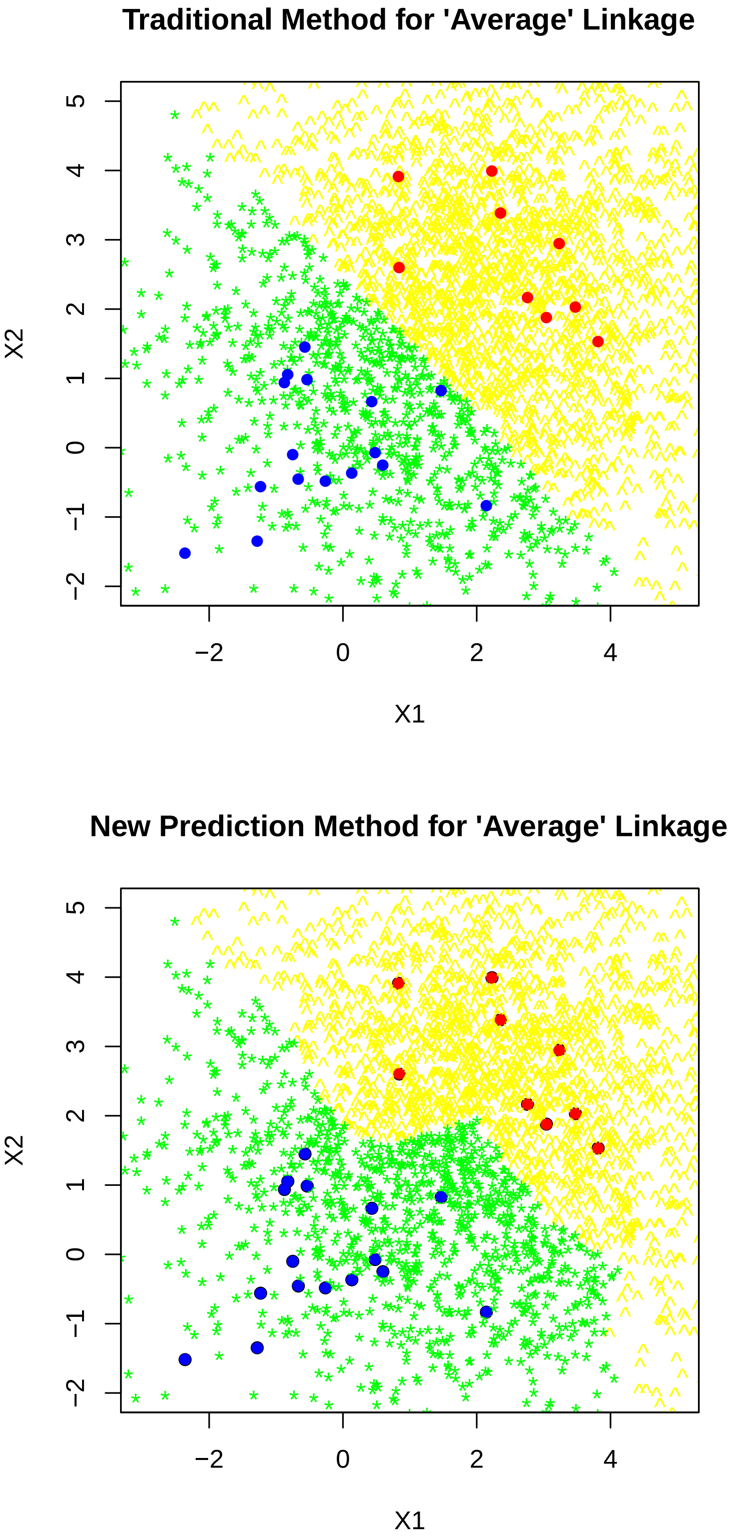


Figure 3.5: Traditional and New Prediction Method with "Average" Linkage Distance.

3.2.5 Ward's Clustering

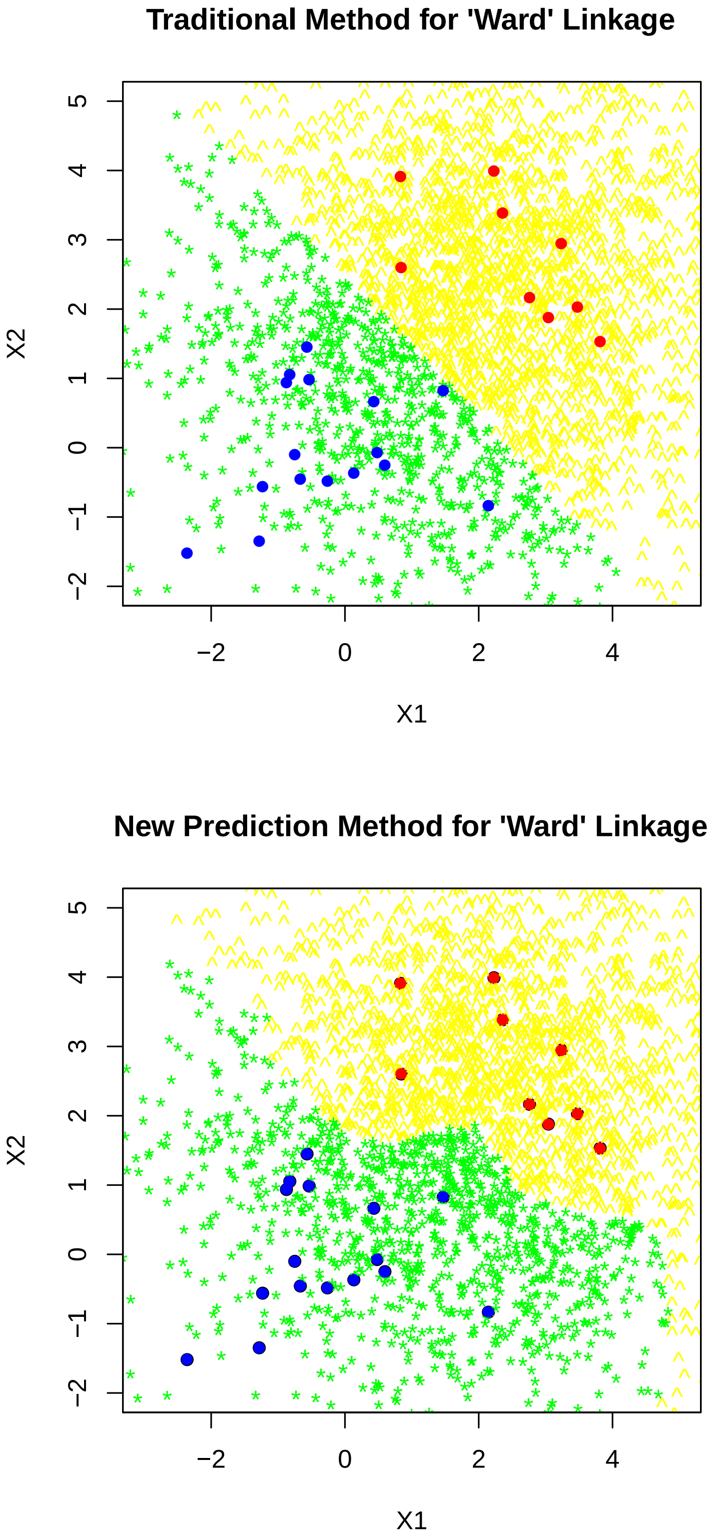
In *ward's clustering*, the distance between two clusters is calculated as the sum of squares between clusters divided by the total sum of squares, or equivalently, the change in R^2 when a cluster is split into two clusters. Here R^2 is the *coefficient of determination* , which is the percent of the variation that can be explained by the clustering.

Our prediction method for ward clustering, may not correspond to the same cluster as the prediction with the traditional method. The traditional prediction method calculate the distance between the new point and all the cluster centers and assign this point to the cluster with the closest distance. While our method will calculate the inter-cluster distance using *bottom-up clustering* algorithms until this new point joints to a cluster, say A. Finally We will assign the new point to the original cluster of the first point in A, namely A'. The difference is showed in Figure 3.6. We could see the boundary is linear for traditional clustering while nonlinear for our HCTP. In this case, the results for "Ward" are the same as "Average" linkage.

3.2.6 Comments on different Hierarchical Clustering Methods

All the hierarchical clustering methods we mentioned above, though have lot of similarities, they do persist different properties and will generally cluster the data in quite different ways and may even impose a structure of their own.

The single linkage is setting up to maximize the connectedness of a cluster and highly prefer to find chain-like clusters. A sequence of close observations in different groups may cause early merge by single linkage. The complete linkage has the opposite problem: it sets up to minimize the maximum within-cluster



Traditional Method for 'Ward' Linkage

New Prediction Method for 'Ward' Linkage

Figure 3.6: Traditional and New Prediction Method with “Ward” Linkage Distance.

distance and tends to find compact clusters but may overemphasize small differences between clusters. Complete linkage might not merge close groups if outlier members are far apart.

3.3 Simulation Set Up and Results

We simulated data from two normal distributions but constrained to some certain regions of the space. The distributions we used for simulation study are: $\mathfrak{G} = \{G_1, G_2\}$, where $G_1 = \{(x_{i1}, y_{i1})\}$, $i = 1, ..., n_1$, and $G_2 = \{(x_{i'2}, y_{i'2})\}$, $i' = 1, ..., n_2$. Here we use (x_{i1}, y_{i1}) to stand for the coordinate for i^{th} observation in group 1, and (x_{i2}, y_{i2}) to be the coordinate for i^{th} observation in group 2. Observations in G_1 satisfy conditions:

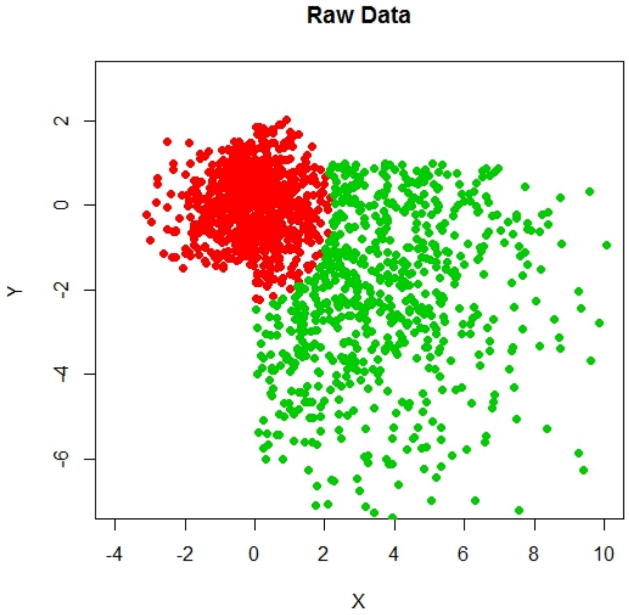


Figure 3.7: Simulation Data. Red and Green samples are corresponding to two groups.

$$\begin{cases} x_{i1} \geq 0 \\ \sqrt{x_{i1}^2 + y_{i1}^2} < 2.25 \end{cases} \tag{3.8}$$

Or

$$\begin{cases} x_{i1} < 0 \\ |y_{i1}| < 1.5 \end{cases} \tag{3.9}$$

for each $i \in \{1, ..., n_1\}$, here we set $n_1 = 1000$. In contrast, conditions for G_2 are:

$$\begin{cases} x_{i2} \geq 0 \\ y_{i2} \leq 1 \\ \sqrt{x_{i2}^2 + y_{i2}^2} > 2.25 \end{cases} \tag{3.10}$$

for each $i \in \{1, ..., n_2\}$, we set $n_2 = 2000$.

Data we generated above is plotted in Figure 3.7. We could see the raw data is clearly divided into two groups: colored with **Red** and **Green**.

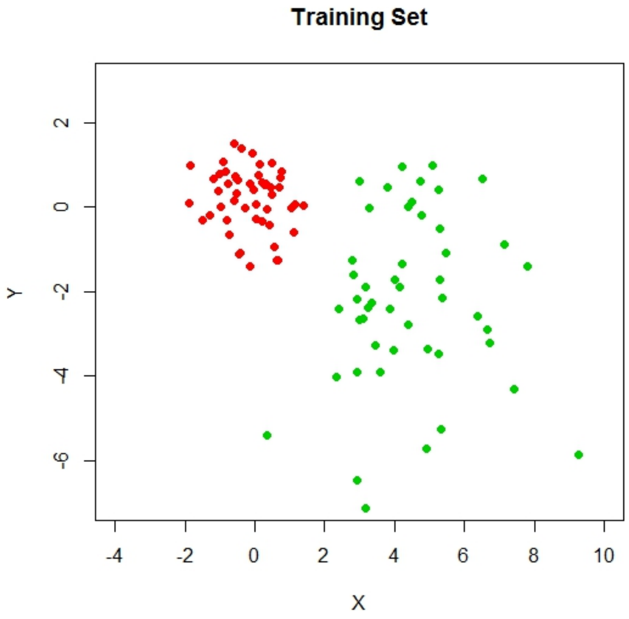


Figure 3.8: Training Data.Red and Green samples are corresponding to two groups.

There are 1000 observations for group 1 (Red) and 2000 observations for the group 2 (Green) in Figure 3.7. We took the first 50 observations in each group to be our training data as in Figure 3.8, and use the rest to be the validation set.

Hierarchical clustering with "Ward" method clustered our training observations into two clusters (colored as yellow and blue in Figure 3.9). We could see from Figure 3.9, yellow observations (cluster1) and red observations (group1) are perfectly matched, also blue observations (cluster2) and green observations (group2) are perfectly matched. So our training set is well clustered into two clusters under "Ward" method.

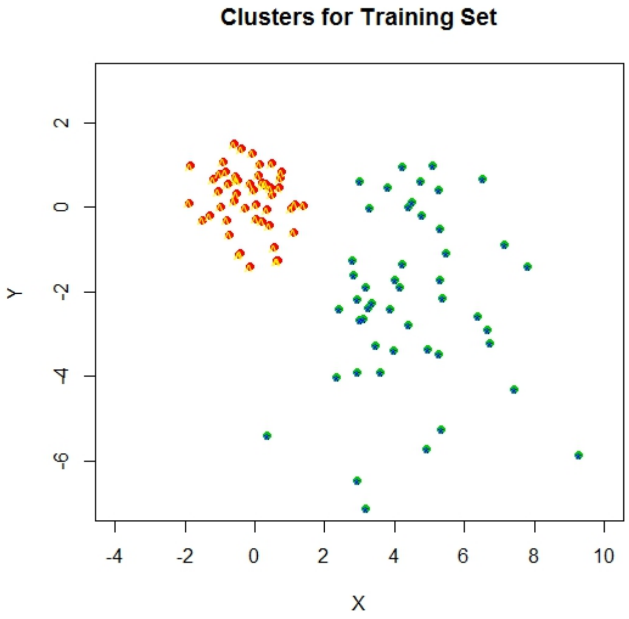


Figure 3.9: Clusters of Training Data.Yellow and Blue samples are corresponding to two clusters.

Both traditional method and our HCTP method were applied to the simulated data to assess their performance. We run 1000 reiteration and pick one to be an illustration example as show in Figure 3.10. We could see the classification boundaries are different it is linear for the traditional method while nonlinear for our HCTP. And it follows the data density. Then which one is better? We will compute the misclassification ratio to measure the performance for both methods.

The prediction scheme for each method assigned the simulated test observations into two clusters: $\mathcal{C} = \{C_1, C_2\}$. Let $G_1^* = \{g_{i1}^* : (x_{i1}^*, y_{i1}^*)\}$, and $G_2^* = \{g_{i2}^* : (x_{i'2}^*, y_{i'2}^*)\}$ be testing observations in group 1 and group 2. We use r_1 to

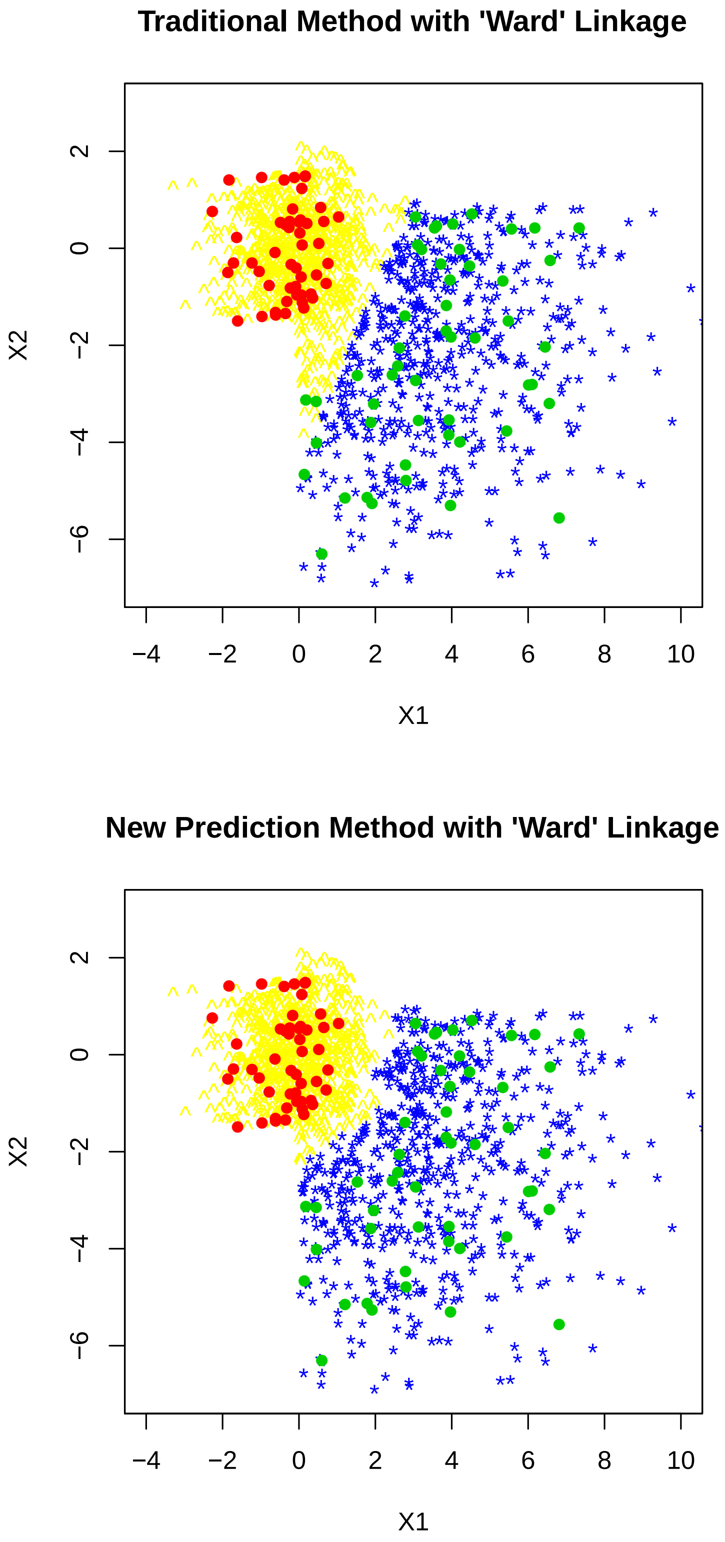


Figure 3.10: Prediction Result for both Methods.

present our HCTP misclassification ratio, and r_2 be the traditional prediction misclassification ratio. The formula for misclassification ratio r_k (k=1,2) is:

$$r_k = \frac{\#P_k(G_1) \in C_2 + \#P_k(G_2) \in C_1}{n'_1 + n'_2} \tag{3.11}$$

where P_1 is our HCTP result, P_2 is the traditional prediction result, n'_1 is the number of test observations in group 1 and n'_2 is the number of testing observations in group 2. With 1000 reiteration, the average values for r_1 and r_2 are: $r_1 = 0.0283$, $r_2 = 0.1067$. So the misclassification ratio $r_1 < r_2$ indicating misclassification error of our HCTP is smaller than traditional method. We will go further in next section by application of each method on real data example.

3.4 Real Data Example

3.4.1 Data Description

The NPSI questionnaire consists of 10 different pain symptom descriptors. The data came from 4 randomized, double-blind, placebo-controlled clinical studies of pregabalin (150-600 mg/day) in patients with neuropathic pain (NeP) syndromes: central post-stoke pain, post-traumatic peripheral pain, painful HIV Neuropathy, and painful diabetic peripheral neuropathy. Patients enrolled were males or non-pregnant, non-lactating females $aged \geq 18$ with a diagnosis of NeP syndromes: CPSP (219), PTNP (254), painful HIV neuropathy (302), and painful DPN (450).

Patients with specific NeP syndrome were enrolled in each study, and were asked to record their daily pain score on Numeric Rating Scale (NRS) with 11-points, where 0= no pain and 10= worst possible pain. The average of the NRS scores over the 7 days prior to randomization was used as mean pain score at baseline.

We will use NPSI as our training data set, which has 1161 observations 11 variables. The columns of NPSI are 10 different pain symptom descriptors and

1 mean vector: superficial and deep spontaneous ongoing pain (Questions 1 and 2); brief pain attacks or paroxysmal pain (Questions 5 and 6); evoked pain (pain provoked or increased by bushing, pressure, contact with cold on the painful area; Questions 8 through 10); abnormal sensations in the painful area (dysesthesia/paresthesia; Questions 11 and 12); and duration of spontaneous ongoing pain assessment (Question 3). Our testing data is a new clinical trial NPSI data set, with 210 observations and same 11 variables.

3.4.2 Clustering Prediction Result

The objective of this exercise is to try our new clustering prediction method on the fifth study which was finalized after we had performed the analysis of the four previous studies.

Once the new data has been normalized by Fisher Yate's, we predict the cluster number of each for the new observation. Table 3.1 and Table 3.2 summarize the new prediction results compared to the traditional method.

Figure 3.11 indicates the patients' overall NPSI scores' means are: cluster1=5.54, cluster2=2.41, and cluster3=4.29. Perform our HCTP and traditional prediction method on NPSI data, we have Table 3.1 to capture the clustering prediction results. We could see our HCTP predict more observations (102 observations) to cluster1, while traditional method predict more to cluster 2 and 3. Also the sum of off diagonal numbers "63" means our HCTP and traditional method assign 63 observations in testing data to different clusters.

Figure 3.12 and Figure 3.13 list some testing observations which will get different prediction result under different clustering prediction method. For example, our HCTP predicts sample 1 to cluster 1, while the traditional clustering method predicts it to cluster 2 as shown in Figure 3.12. We summarize the clustering ratio results for these two methods in Table 3.2, indicating that the predicted cluster ratio under our HCTP is more closer to the ratio of training data.

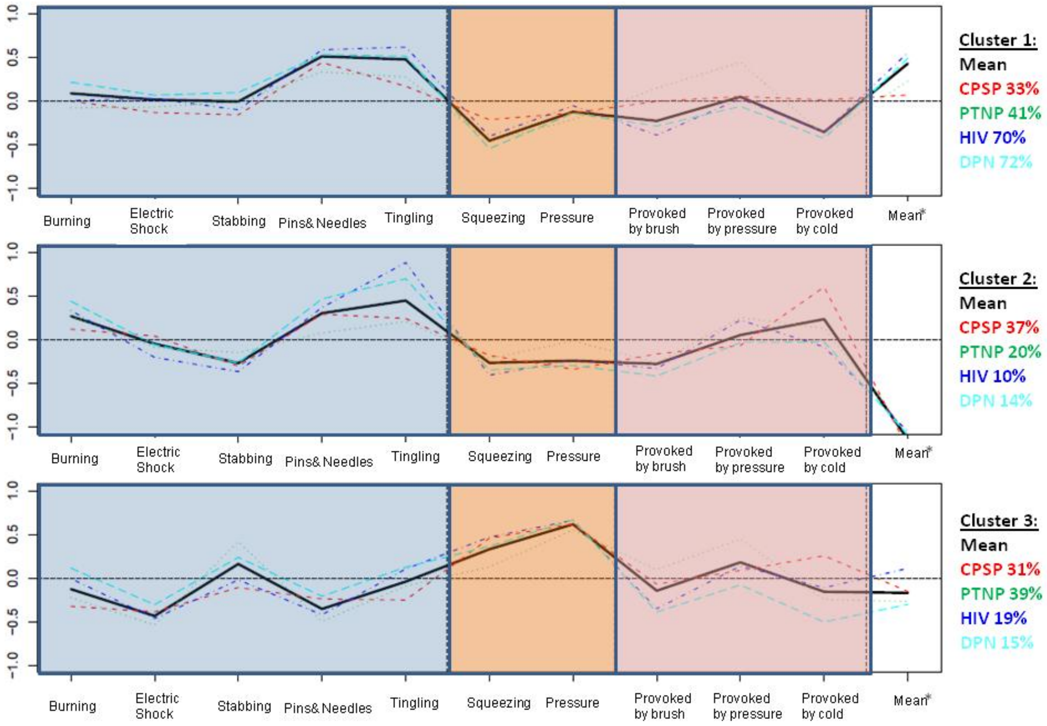


Figure 3.11: NPSI Cluster means by disease and individual pain dimension. CPSP= central post-stroke pain; DPN= painful diabetic peripheral neuropathy; HIV= painful HIV neuropathy; NPSI= Neuropathic Pain Symptom Inventory; PTNP=post-traumatic peripheral neuropathic pain.

3.5 Conclusions

In this chapter we propose a cluster prediction method for hierarchical clustering. The idea is to

I) Make predictions that follow the structure of the tree;

II) Produce a tree which is the original tree but where we have incorporated all the new observations, and therefor this new tree is more complete than just the predictions because it also includes the level at which the prediction happens.

We also show the results of using the new prediction method with a new clinical trial that was completed after the initial 4 trials were analyzed. We found that the cluster that had significant treatment in the initial 4 studies also showed

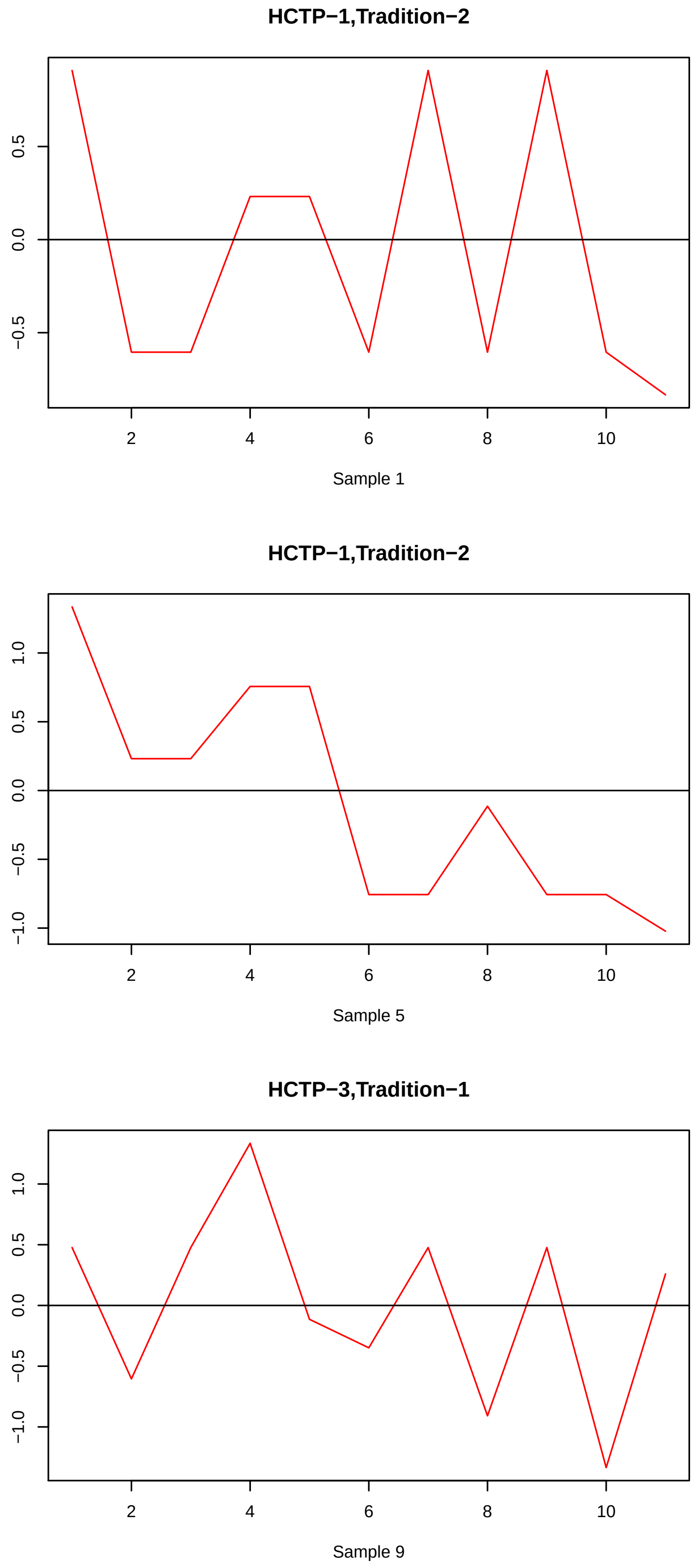


Figure 3.12: Testing observations 1. Show some testing observations which get different prediction result under HCTP and Traditional method. Main title for each sample indicates the prediction cluster result for each method.

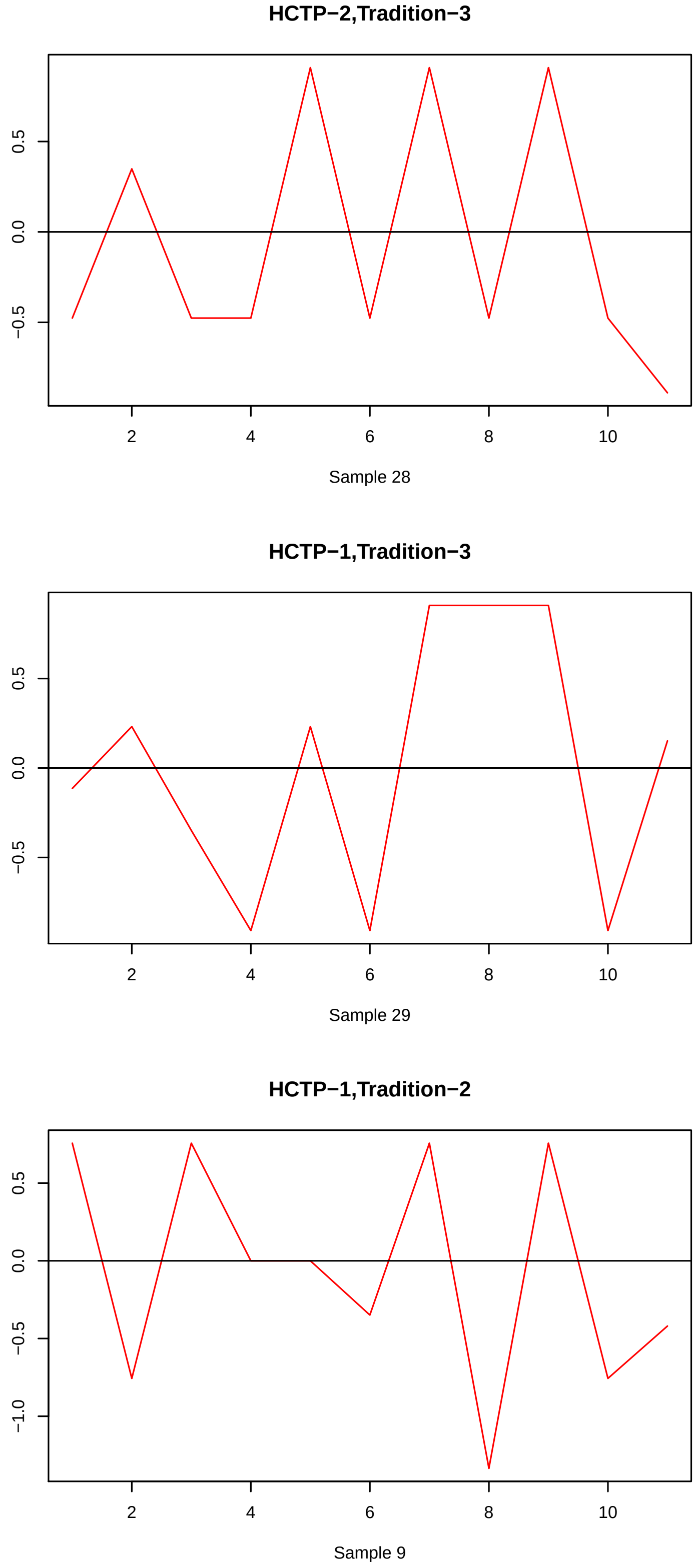


Figure 3.13: Testing observations 2. Show some testing observations which get different prediction result under HCTP and Traditional method. Main title for each sample indicates the prediction cluster result for each method.

Table 3.1: Prediction Table for Both Methods

	Traditional		
HCTP	1	2	3
1	66	21	15
2	7	50	6
3	9	5	31

Table 3.2: Cluster Ratio Table

	1	2	3
Training Set	57.97	18.26	23.77
HCTP	48.57	30	21.43
Tradition	39.05	36.19	24.76

significant treatment effect in the new trial.

Chapter 4

Dose and Cohort Size Adaptive Design

The primary objective of phase I clinical trial is aiming at locating the maximum-tolerated dose (MTD). The Food and Drug Administration published the guidance for industry in 2010 indicating that "an adaptive design clinical study is defined as a study that includes a prospective planned opportunity for modification of one or more specified aspects of the study design and hypotheses based on analysis of data from subjects in the study". Traditional adaptive methods will adapt dose up and down eventually depend on the toxicity from observed data. In this chapter, we are going to introduce a novel dose assignment method called dose and cohort size adaptive design, which will adapt dose and cohort size at the same time, thus able to detect the true MTD with less cohorts while still keep the accuracy.

4.1 Introduction and Background

Early-phase clinical trials are first-in-human studies for new treatment. The primary objective of phase I oncology trial is to define the recommended phase II dose of a new drug, aiming at locating the MTD: the dose with a dose-limiting toxicity (DLT)is closest to a predefined target toxicity rate $\eta(0 < \eta < 1)$. The main outcome for most existing dose-finding designs is toxicity, and escalation action is guided by ethical considerations. It is very important to estimate the MTD as accurate as possible, since it will be further investigated for efficacy in Phase II study. The study will begins at low doses and escalate to higher doses eventually due to the severity of most DLTs. However, we also want the escalation of doses to be as quick as possible since the lower doses are expected to be ineffective in most cases.

A rich literature has been published for dose-finding designs of Phase I trials. The conventional 3+3 design, first introduced in the 1940s, is still the most widely utilized dose-escalation and de-escalation scheme. However, there are some limitations when applying 3+3. Statistical simulations demonstrated that 3+3 design is used to identify the MTD in as few as 30% of trials. Another very popular model-based method is Continual Reassessment Method (CRM) which estimate the MTD based on one-parameter model and eventually updated the estimator every time one cohort completes either by Bayesian methods given by O'Quigley et al. (1990), or maximum likelihood methods given by O'Quigley and Shen (1996).

We developed a new approach called Dose and Size(D&S) adaptive design. Here we are going to introduce algorithms of our approach with comparison to previous designs through a simulation study. Comparisons are focused on accuracy, safety and benefits of the procedures. Results show general advantages of our new method, and also show saving on time and cost, which is the most exciting advantage of our approach.

4.1.1 Case Study

AAB003 STUDY AAB003 from Pfizer is a backup compound of Bapineuzumab. Preclinical evidence suggested that AAB003 may have a reduced risk of VE (vasogenic edema) compared to Bapineuzumab. An AAB003 dose higher than Bapineuzumab dose may result in good efficacy, while maintain the same or lower VE rate($\leq$ 5%)So the First-in-Human (FIH) study of AAB003 in subjects with mild to moderate Alzheimer's disease was conducted, to assess the safety and tolerability of AAB003 at different dose levels (0.5,1,2,4,8 mg/kg).This FIH study was also compared to an alternative, more traditional design of a single ascending dose study (SAD) followed by a multiple ascending dose (MAD) study.

The objective of the trial is to establish as efficiently as possible whether AAB003 has a better safety profile than bapineuzumab(AAB001). The current safety profile of bapineuzumab includes vasogenic edema (VE) of the brain, a radiographic finding that has been reported among some subjects treated with Bapineuzumb. Final dose selection will depend on the full package of preclinical safety data.

DIABETES COMPOUND A multiple ascending dose (MAD) study was conducted for Diabetes Compound project. During the study, a particular adverse event emerged and raised concern. The team planned to narrow down the dose range with a parallel study with AE target rate $\leq 3\%$.

For both projects, the AE rates are very small (3% and 5%). That means the sample size for each dose will be very big: need at least $n = 33(1/3\%)$ and $n = 20(1/5\%)$ to observe one AE in average. So the study will be very big, if go with parallel study, take lot of time and cost.

We are highly motivated to develop a new study to save time and money. Lot of adaptive models are given in different literatures, but no one adapts the cohort size. Our *D&S* design is focusing on cohort size adaptive method and we will show the advantages of this new model.

4.2 Existing Methods for Dose-finding Study

Before going to introduce some well applied existing methods, let's first give the definition of MTD. Rosenberger and Haines (2002) mentioned there are two different definitions of MTD. It could be defined as the dose just below the lowest dose level with unacceptable toxicity rate Γ_U, or can be defined as the dose with the toxicity probability equal to some acceptable toxicity level Γ, where $\Gamma < \Gamma_U$.

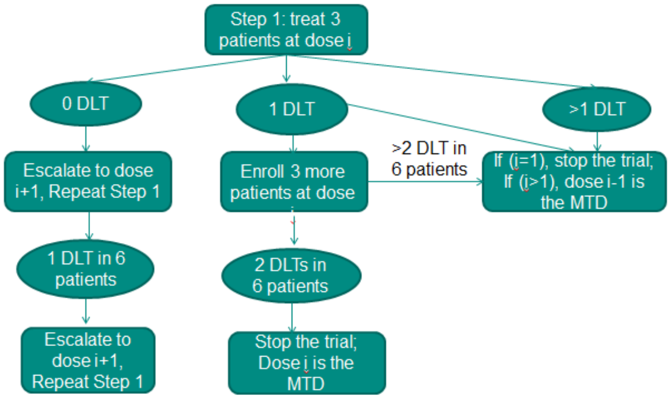

Step 1: treat 3 patients at dose i
0 DLT
1 DLT
>1 DLT
Escalate to dose i+1, Repeat Step 1
Enroll 3 more patients at dose
>2 DLT in 6 patients
If (i=1), stop the trial; If (i>1), dose i-1 is the MTD
1 DLT in 6 patients
2 DLTs in 6 patients
Escalate to dose i+1, Repeat Step 1
Stop the trial; Dose i is the MTD

Figure 4.1: 3+3 Design. One of the most popular dose-escalation and de-escalation scheme.

4.2.1 3+3 Design

The conventional 3+3 design, was originally mentioned in 1940s and re-promoted by Storer (1989). It is among the earliest dose-escalation and de-escalation schemes. We showed the method of 3+3 as in Figure 4.1, which can be more easily to present.

The basic idea of 3+3 is, treat 3 patients per dose level. If one Dose Limiting Toxicity (DLT) occur, dose is escalated for the next cohort of 3 patients. If 1 DLT, put 3 more patients at this levle with dose escalation only if no additional DLTs. If ≥ 2 DLTs, we define prior dose level as MTD. 3+3 is very simple and very easy to implement, however, there are some limitations. First, statistical simulations show that 3+3 is mostly used when MTD is as few as 30% of the trails. Furthermore, this may put very large population being treated at sub-therapeutic doses.

3+3 design has been widely criticized due to its escalation decision is made only based on the most recent recruited patients. Promoted by this disadvantage, O'Quigley et al. (1990) developed a new model-based adaptive design CRM(continual reassessment method) which made decisions based on posterior distributions and likelihoods formed from all accumulated data.

4.2.2 Continuous Reassessment Method

The underlying assumption of CRM is the probability of DLT increases monotonically with re-scaled version of the doses. If define Y_i to be the binary toxicity outcome of patient i: $Y_i = 1$, if DLT occur, $Y_i = 0$ otherwise. The CRM uses a one-parameter hyperbolic-tangent dose-response model as:

$$\phi_i = \pi(d_i; \gamma) = [(tanh(d_i) + 1)/2]^{\gamma} \tag{4.1}$$

where $\gamma \in R$, which will be updated during the trial; d_i is standardized or re-scaled value of the dose assigned to subject i; while π is the monotonic function with range [0,1]. There are two further 1-parameter models given in original paper O'Quigley and Chevret (1991):

1. a logistic model with fixed intercept c (always use 3):

$$\pi(d_i; \gamma) = \frac{exp(c + \gamma d_i)}{1 + exp(c + \gamma d_i)} \tag{4.2}$$

2. a power model given by:

$$\pi(d_i; \gamma) = d_i^{\gamma} \tag{4.3}$$

The exponential prior of γ is $\pi(\gamma) = exp(-\gamma)$ with mean equal to 1. Given the data for doses d_i and outcomes Y_i the likelihood is

$$L(\mathbf{d}|\gamma) = \Pi \pi(d_i; \gamma)^{Y_i}(1 - \pi(d_i; \gamma))^{1-Y_i} \tag{4.4}$$

The posterior is

$$\pi(\gamma|\mathbf{d}) = \frac{L(\mathbf{d}|\gamma)\pi(\gamma)}{\int_0^{\infty} L(\mathbf{d}|\gamma)\pi(\gamma)d\gamma} \tag{4.5}$$

Calculate the posterior mean $\tilde{\pi}_j$ of the DLT rate at dose d_j, then the recommended dose for the next cohort is the one with DLT rate closest to the target η. That is

$$j = arg \min_{j \in (1,\ldots,J)} |\tilde{\pi}_j - \eta| \tag{4.6}$$

The CRM usually does not allow dose skipping during dose escalation. The trial will stop if the total number of subjects N is reached or if a specific stopping rule is satisfied. We will discuss stopping rules later. The MTD is determined at the end of the trial by simply selecting dose j according to function (4.6).

4.2.3 Escalation and Group Designs

Let $D = \{d_1, ...d_D\}$ be the set of ordered dose, and p_i is the corresponding toxicity probability of d_i, where $p_1 < ... < p_k$, n is the cohort size for dose i. The adverse events at each dose $x_i < n$ has a Binomial distribution. The likelihood function is a product of binomial densities,

$$l(\mathbf{p}) \propto \prod_{i=1}^{D} p_i^{x_i}(1-p_i)^{n_i-x_i} \tag{4.7}$$

Escalation Design Subjects are treated in cohort size n starting from lowest dose. Let c_U be an integer that $0 \leq c_U < n$. Assume our current dose is dose d_i, $i = 1, ..., D-1$. Then

1. if $x_i \leq c_U$, the next n cohort subjects will be assigned to dose d_{i+1}.
2. if $x_i > c_U$, the trial is stopped and claim the dose one level below c_U is the MTD.

Group Design Subjects are treated in cohort size n starting from lowest dose. Let c_L, c_U be two integers that $0 \leq c_L < c_U < n$. Assume our current dose is dose d_i, i=1,...D. Then

1. if $x_i \leq c_L$, escalate to dose d_{i+1} and assign the next n cohort subjects.
2. if $c_L < x_i < c_U$, stay in the current dose and assign the next n cohort subjects.
3. if $x_i \geq c_U$, De-escalate to dose d_{i-1} and assign the next n cohort subjects.

4.2.4 Toxicity Probability Intervals Methods

Rosenberger and Haines (2002) noted that "Bayesian methods, such as the CRM, the EWOC, and the decision theoretic approach, are complicated to explain to non-statisticians and computationally challenging to implement." Ji et al. (2007) promoted a simpler Bayesian model which is easier to implement and understand.

Their assignment rule for phase I clinical trials is based on a conjugate Beta-Binomial model. Assume an independent beta prior $B(a, b)$ with mean $a/(a + b), a, b > 0$. With a binomial likelihood, the posterior distribution of toxicity is also a beta. Similar to the idea of group design, Ji et al. (2007) built the probabilistic up-and-down rule based on the posterior distribution at the current tried dose. The idea is to cut the posterior distribution into 3 intervals, each corresponds to de-escalate, stay or escalate dose-assignment action, depends on which interval has the largest posterior probability.

Let p_T be the target AE rate at MTD. The three intervals for current dose d_i are defined as:

- $qE(i) = P(p_i - p_T < -K_1\sigma_i|data)$
- $qS(i) = P(-K_2\sigma_i \leq p_i - p_T \leq K_1\sigma_i|data)$
- $qD(i) = P(p_i - p_T > K_2\sigma_i|data)$

where σ_i is the posterior standard deviation of p_i. K_1 and K_2 are two design parameters and will be adjusted through intensive simulations.

The dose assignment rule can be written as

$$ß = argmax\{qE, qS, qD\} \tag{4.8}$$

where the action corresponding to the interval with the largest posterior probabilities.

4.3 Dose and Size Adaptive Design

Ji's Toxicity Probability Interval(TPI) method is easy to implement and explain compared to biased-coin design(BCD) with isotonic regression estimator by Stylianou and Flournoy (2002), cruve-free method(CFM) by Gasparini and Eisele (2000), and continual reassessment method(CRM) by O'Quigley et al. (1990).

Our D&S method adds more strict rules and is more hesitate to take actions compared to TPI. For example, if current dose is d_i and our action is Stay when applying TPI and the next cohort size is n. Our method will also takes a look at the next lower dose and next higher dose, we will take action depend on all this combined information and will also adapt the cohort size.

For dose escalation rules, $D\&S$ follows the same principles as 3+3, TPI and CRM etc., that will Escalate if current dose has AE rate too high, stay if around the target rate, and De-escalate if is too low. Also, we change the cohort size depending on whether the next dose is likely or unlikely to be the MTD. We will not change cohort size if we are uncertain it is MTD, add more subjects if the dose is likely to be the dose with targeted AE rate, and add much more subjects if the dose is highly likely to be the dose with targeted AE rate.

4.3.1 Stopping Rules

Taking ethical concern of overdosing subjects into consideration, researchers developed lot of stopping rules that assuming all doses under study are too toxic (Korn et al. (1994), O'QUIGLEY and Reiner (1998)). Ji et al. (2007) also mentioned if taking escalation action as long as the current dose is not toxic is incomplete for ethical consideration. However, it is not safe if the next higher dose is highly toxic and usually it is not allowed to expose patients under such high toxic dose. Ji et al. (2007) modified the dose-assignment rule by adding a toxicity exclusion rule based on a random variable

$$\tau_i = 1\{P(p_i > p_T | data) > \xi\} \tag{4.9}$$

where 1 is the indicator function, ξ is a cutoff point that $\xi \in (0, 1)$. For a large value of ξ, e.g., $\xi = 0.95$, $\tau_i = 1$ means dose i is highly toxic and escalate to this dose should never be allowed. So we need to information of current dose and next higher dose to decide the action of "Escalation". Redefine qE as $\hat{qE}$:

$$\hat{qE}(i) = qE(i)(1 - \tau_{i+1}) \tag{4.10}$$

now the assignment rule becomes

$$\hat{\text{ß}} = argmax\{\hat{qE}, qS, qD\} \tag{4.11}$$

Based on random variable τ, we could see if our current dose is d_i and $\tau_{i+1} = 1$, means the next higher dose is too toxic and the probability of $\hat{qE}$ will equal zero and the assignment rule $\hat{\text{ß}}$ will never chose action *Escalation*.

Another question is, if the first dose is already too toxic, what should we do? As we mentioned above, in clinical trial study, the toxic probability is assumed non-decreasing, $p_1 \leq ... \leq p_k$. So if $\tau_1 = 1$, means the first dose is already too toxic, also implicates all the higher doses are highly toxic, so we will terminate the trial immediately. In other words, if the DLT rate of the lowest dose is already higher than the target pre-specified probability η, the trial will terminate immediately and no dose is selected as MTD.

4.3.2 Escalation Tables

We use different escalation rules which are listed from Table 4.1 to table 4.3. They correspond to the design principle tables when current dose has escalation trend, stay trend and de-escalation trend separately. The first column is the trend for current dose in all three tables, second column is the trend for higher dose or lower dose(table 4.3), third column gives the actual action taken and adapted cohort size numbers by our *D&S* adaptive design, and fourth column gives the action and cohort size taken by TPI or other dose-finding studies. The cohort sizes m0, m1, m2, m3, m4 will be specified in our simulation study, also we will give mathematical explanations for strong escalation, weak escalation, stay and de-escalation.

Table 4.1: General Dose Escalation Table1: *Current dose has Escalation trend.*

Current Dose	Higher Dose	Action by D&S	Action by TPI or Others
Strong Esc	Strong De-esc	S;CS=all	E;CS=m0
Strong Esc	Weak De-esc	E;CS=m1	E;CS=m0
Weak Esc	Strong De-esc	S;CS=m4	E;CS=m0
Weak Esc	Weak De-esc	E;CS=m0	E;CS=m0

Table 4.2: General Dose Escalation Table2: *Current dose has Stay trend.*

Current Dose	Higher Dose	Action by D&S	Action by TPI or Others
Stay	Strong De-esc	S;CS=m3	S;CS=m0
Stay	Weak De-esc	S;CS=m2	S;CS=m0

Table 4.3: General Dose Escalation Table3: *Current dose has De-escalation trend.*

Current Dose	Lower Dose	Action by D&S	Action by TPI or Others
Strong De-esc	Strong Esc	D;CS=all	D;CS=m0
Strong De-esc	Weak Esc	D;CS=m4	D;CS=m0
Weak De-esc	Strong Esc	S;CS=m1	D;CS=m0
Weak De-esc	Weak Esc	S;CS=m0	E;CS=m0

We should have noticed that for first dose there is no lower dose, and for last dose there is no higher dose. So the escalation rules are different when the fist dose is $dose1$ or $doseD$. If first dose is $dose1$, our current action can only be Escalation or Stay, then the escalation rules are the same as table 4.1 or table 4.2; but if current action is De-escalation, we take action Stay and don't change the cohort size. If fist dose is *doseD*, our current action is De-escalation, the escalation rules are same as table 4.3; but if current action is Escalation or Stay, see table 4.4 and table 4.5.

Table 4.4: Dose Escalation Table: *d=D, current dose has Escalation trend.*

Current Dose	Action by D&S	Action by TPI or Others
Strong Esc	S;CS=all	S;CS=m0
Weak Esc	S;CS=m4	S;CS=m0

Table 4.5: Dose Escalation Table: *d=D, current dose has Stay trend.*

Current Dose	Lower Dose	Action by D&S	Action by TPI or Others
Stay	Strong Esc	S;CS=m3	S;CS=m0
Stay	Weak Esc	S;CS=m2	S;CS=m0

The mathematical definitions for strong escalation, weak escalation, strong de-escalation and weak de-escalation are listed as:

- Strong Escalation: $qE(i) > \delta$;
- Weak Escalation: $qE(i) < \delta$;
- Strong De-escalation: $qD(i) > \delta$;
- Weak De-escalation: $qD(i) < \delta$.

4.4 Simulation Set-up and Results

4.4.1 Pre-defined Parameters

Similar to most clinical trial studies, we assume the probabilities of toxicity to be correlated and increase with the dose level. We will consider several simulation scenarios to evaluate the performance of the proposed design under starting cohort size of 10 for each pre-defined target toxicity rate η. Here, we use p_T to stand for η. We will look at both $p_T = 0.03$ and $p_T = 0.1$. The simulation results will show comparison between our D&S method and Ji's TPI.

We have 5 active doses $d_1, ..., d_5$ when $p_T = 0.03$, cap subject number $n \leq 100$ for active 5 doses. Each cohort has original 10 subjects on active, but will be adapted eventually during the clinical study. Dose up or down assignment action will be taken depend on the combination information of current dose, lower dose and higher dose. Since our p_T is very small, we need at least $1/0.03 = 33$ subjects to observe 1 AE (Adverse Event). If go with parallel study, it takes lot of time and money. Thirteen scenarios are covered in our simulation study as shown in figure 4.2, we could see these scenarios cover almost all different dose ranges around $p_T = 0.03$.

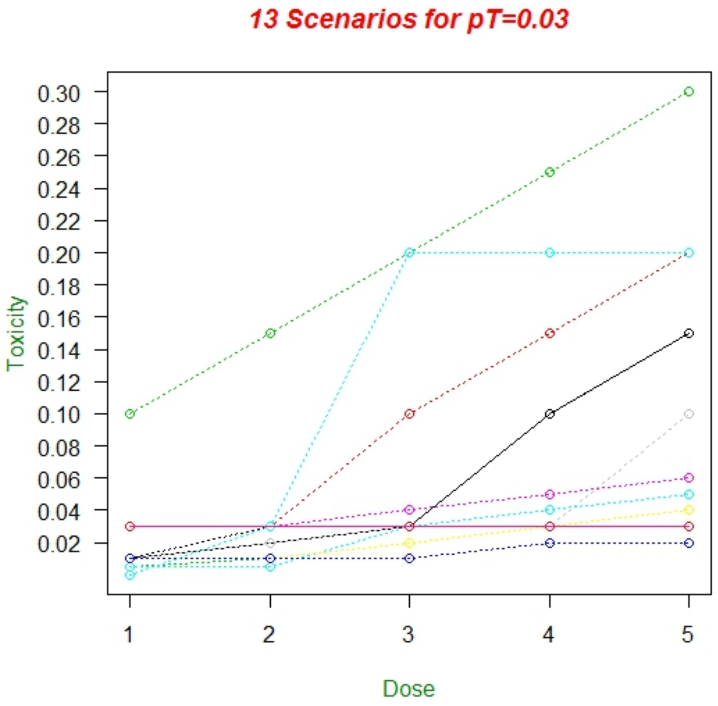


Figure 4.2: 13 Scenarios when $p_T = 0.03$

For $p_T = 0.1$ we have 4 active doses $d_1, ..., d_4$, cap subject number $n \leq 100$ for active 4 doses. Each cohort also has original 10 subjects on active, but will be adapted during the clinical study. When our $p_T = 0.1$, we need around $1/0.1 = 10$ subjects to observe 1 AE (Adverse Event). This is much better than $p_T = 0.03$. Under the situation when $p_T = 0.1$, we will consider 3 different scenarios to show the simulation results in later section.

We assigned $\alpha = 0.003$, $\beta = 0.007$ to the parameters of our prior $Beta(\alpha, \beta)$, given the values of n_i and x_i, our posterior distribution is $Beta(0.003+x_i, 0.007+n_i - x_i)$. Therefore, we can computer decision rule ß for any values of n_i and x_i and tabulate the results.

Also for escalation tables, we set $\delta = 0.8$ in our simulation study.

4.4.2 Model Construction and Variance Transformation

In our study, we will use $Beta(\alpha, \beta)$ distribution as the prior for p, i.e.,

$$\pi(p) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)p^{\alpha-1}(1-p)^{\beta-1}} \tag{4.12}$$

With a binomial likelihood Equation (4.7), the posterior distribution is still a Beta. This makes our model more convenient. Note:

$$\begin{aligned} \pi(p|\mathbf{x}) &\propto f(x_1, x_2, ..., x_n|p)\pi(p) \\ &= p^x(1-p)^{n-x}p^{\alpha-1}(1-p)^{\beta-1} \\ &= p^{\alpha+x-1}(1-p)^{\beta+n-x-1} \end{aligned} \tag{4.13}$$

So in our model, the posterior is $Beta(\alpha+x, \beta+n-x)$, where x stands for adverse event, n is each cohort number. Then the questions is how to pick up the prior properly? Under the posterior distribution, we can see the posterior mean and variance are:

$$E(p|\mathbf{x}) = \frac{\alpha+x}{\alpha+\beta+n} \tag{4.14}$$

$$Var(p|\mathbf{x}) = \frac{(\alpha+x)(\beta+n-x)}{(\alpha+\beta+n)^2(\alpha+\beta+n+1)} \tag{4.15}$$

Zhu and Lu (2004) indicates that, the prior variance is a decreasing function in prior parameters, so the larger the prior variance, the less influential the prior is. Zhu and Lu (2004) also gave other methods on how to select non-informative prior for binomial family.

In our model, we assign $\alpha = 0.003$, $\beta = 0.007$, so the prior doesn't provide much information. But the problem is, with such small prior parameters, if no adverse event occur in the first trial and we put n (n=10 here) subjects into current dose, we could see the posterior mean is close to 0, and posterior variance is also quite close to 0.

Figure 4.3 shows three different prior densities: Black one is Beta(0.003,0.007), which concentrate density around 0 and 1; Red one is Beta(0.003,10.007), also concentrate around 0; and Green one is Beta(1.003,9.007), showing the density when one AE occurs.

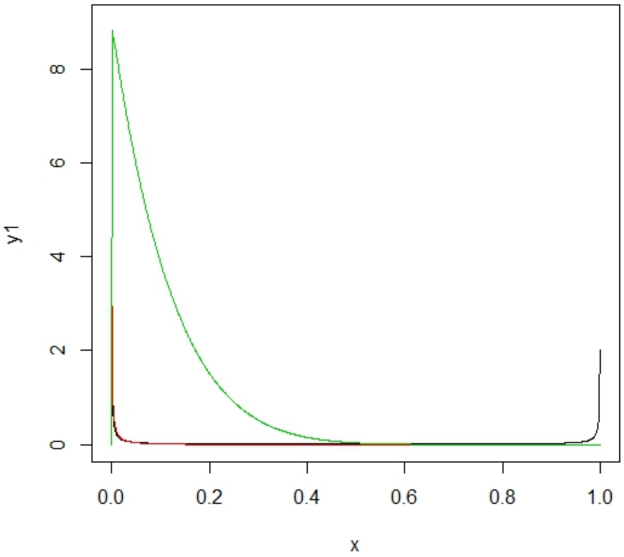


Figure 4.3: Plot for Different Beta Prior densities

We could see from figure 4.3 if no AE occur, the density is too skew. Often, we use a logarithmic transformation $X \rightarrow log(X)$ for analysis (figure 4.4). The logged intensity is popular for number of reasons: taking logs can reduce skewness and improve variance estimation, also the logged intensities variation tends to be less dependent on the magnitude of the values. The raw data are very skew, heavily clumped at low densities and with a long tail. Thus more than 75% of the data lie in the lowest 10% of the intensity. After log transformation, we could see the data are more spread out, and more easily to be examined.

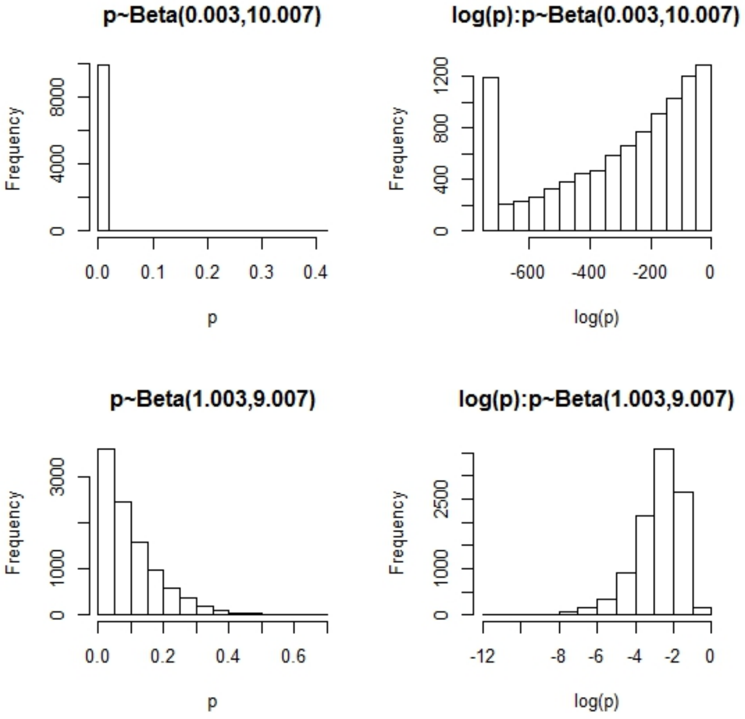


Figure 4.4: Histograms of Log Transformation for Different Beta Prior densities

Sapir and Churchill (2000) also mentioned a more complex transformation, $X \rightarrow log(X+c)$ maybe a better achieve to stabilize the data set. After analyzing real data from experiments with replicate spots, Durbin et al. (2002) discovered that it maybe appropriate to model spot intensity data as

$$X = \alpha + \mu^{\eta} + \epsilon, \tag{4.16}$$

α is the mean background, μ is the true expression level, η and ϵ are error terms with properties η $N(0, \sigma_{\eta}^2)$ and ϵ $N(0, \sigma_{\epsilon}^2)$.

Durbin et al. (2002) showed that after the generalized log transformation,

$$X \rightarrow (X-\alpha) + \sqrt{(X-\alpha)^2 + \frac{\sigma_\epsilon^2}{\sigma_\eta^2}} \tag{4.17}$$

the transformed variance is a constant equal to S_η^2. Durbin et al. (2002) also indicated the loss of convenient interpretation of log ratio by using generalized log transformation can be compromise by the started log transformation, $X \rightarrow log(X+c)$, with $c = \sigma_\epsilon^2/\sigma_\eta^2$.

4.4.3 Isotonic Regression

Isotonic regression is an important form of non-parametric regression. It was first promoted on 1950's and has been widely used in data mining area. Isotonic regression solves the following problem:

$$minimize \sum_i w_i(y_i - \hat{y}_i)^2$$

subject to

$$\hat{y}_{min} = \hat{y}_1 \leq \hat{y}_2 ... \leq y_n = \hat{y}_{max}$$

$w \in R^n$ is the weight vector and every w_i is strictly positive.

One of the biggest advantages of isotonic regression is that it doesn't have any assumption for the target function. Barlow (1972) and Hanson et al. (1973) proposed the *pool adjacent violators algorithm* to solve the above target function, which will find a non-decreasing approximation of the target function. There are lot of existing R packages can implement this PAVA method, so it is quite straightforward to apply.

In our simulation study, when the clinical trial is finished, we will apply PAVA on posterior mean $\hat{p}_i$. Then transformed posterior mean $\hat{p}_i^*$ is non-decreasing, that $\hat{p}_i^* \leq \hat{p}_j^*$ if $i \leq j$. We will select dose i with the smallest value $|\hat{p}_i^* - p_T|$ as our estimated MTD. Suppose there are ties for estimated posterior mean which minimizes the difference, say $p_i = p_{i+1} = ... = p_{i+k} = p^*$, Ji et al. (2007) mentioned that:

- If $p^* < p_T$, we select dose $i + k$ as our estimated MTD;
- If $p^* > p_T$, we select dose i as our estimated MTD.

4.4.4 Simulation Results

We simulated $1,000$ trials. Suppose a toxicity rate of 3% or above is considered unacceptable, 5 doses with different toxicity rates are chosen for the trial. Table 4.6 summarizes the results from 3 common scenarios. Like for scenario1, the first row is the true toxicity probability of each dose (0.01,0.02,0.03,0.1,0.15), from which we generated the trial data, and the target dose is dose3. The first cohort is treated at dose3 with pre-information and the maximum number of patients allowed is 100 in this trial. The next 2 rows are the percentage of times the i^{th} dose was selected as MTD in $1,000$ simulations while applying *D&S* and TPI; 4^{th} and 5^{th} rows are the average number of patients treated at dose i using our *D&S* adaptive design and TPI; last two rows are average adverse events rate for each dose under *D&S* and TPI. The last two columns return the cohort size for each scenario and the average total number of patients used for one trial.

From Table 4.6 we could see our D&S adaptive design could improve the accuracy a little to find the MTD compared to TPI method and also we put more subjects in the MTD. The most exciting improvement of our new design is: generally we can save 3-4 cohorts compared with original 10 cohorts. If each cohort takes 3-4 weeks, our method can save around 3 months for the whole study and keep accuracy at the same time (actually can improve a little bit). Also the AE rates for both methods are quite close to the true toxicity probability.

In Table 4.6, the true toxicity probabilities for each scenario covers our target toxicity probability. We will show some uncommon scenarios in Table 4.7, where the true toxicity probabilities do not cover the target toxicity probability: For scenario11, the starting dose has toxicity probability p=0.1, which is highly toxic while target toxicity $p_T = 0.03$. Both methods stop early, from Table 4.7 can we could see 59.5% times our D&S selects nothing while 74.1% times TPI selects nothing. Also both methods stop without using up 100 subjects.

For scenario12, since the toxicity probabilities for all 5 doses are below target toxicity, both methods select last dose as MTD. In this case, we should conduct further clinical study to test dose with higher toxicity. Our D&S will detect the last dose as MTD more quickly, say takes average 5.5 cohort numbers compared to 10.

Then let's look at scenario13: dose1 has toxicity probability $p = 0.01\%$, which is far lower than 3%, so we want to escalate quickly if we touch dose1; dose3 has toxicity probability $p = 20\%$, which is far higher than 3%, we will de-escalate without hesitation if we touch dose3. In this case, our D&S finds the true MTD with accuracy $p = 92.5\%$ compared to TPI where $p = 87.5\%$, and also we put more subjects in dose2, 74 compared to 58. So under scenario13, it is clear our method is much better than TPI.

Table 4.6: Simulation Results for D&S and TPI Common Scenarios: $p_T = 0.03$, StartDose=3.

		Different Dose levels $p_T = 0.03$						Average Cohort Numbers	Average number of Patients
	Dose	1	2	3	4	5			
Scenario1		1	2	3	10	15	none		
D&S	%MTD	0.3	5.2	84	10.1	0.4	0	5.9	100
	#Pts	0.1	3.5	56.8	32.5	7.1			
	%AE	0	2.9	3	10.2	15.5			
TPI	%MTD	0.7	10.1	77.2	11.9	0	0	10	100
	#Pts	2.23	13.7	50.58	27.81	5.63			
	%AE	1.35	1.9	3.02	9.82	15.28			
Scenario2		1	3	10	15	20	none		
D&S	%MTD	3	81.2	15.5	0.3	0	0	5.9	100
	#Pts	1.5	42.6	44.7	10.2	0.9			
	%AE	0	2.8	10.1	14.7	22.2			
TPI	%MTD	8	77.9	13.3	0.6	0	0	10	100
	#Pts	16.22	39.25	36.21	7.34	0.91			
	%AE	0.86	2.98	9.94	14.99	19.78			
Scenario3		0.5	1	2	3	10	none		
D&S	%MTD	0	2.7	14	75.8	7.5	0	6.4	100
	#Pts	0	2.1	21.1	42.3	34.5			
	%AE	NaN	0	1.9	2.8	10.1			
TPI	%MTD	0.1	2.1	16.4	75.8	7.5	0	10	100
	#Pts	0.5	2.22	30.8	41.27	25.21			
	%AE	0	1.35	1.92	2.86	10			

Table 4.7: Simulation Results for D&S and TPI Uncommon Scenarios: $p_T = 0.03$, StartDose=3.

	Different Dose levels $p_T = 0.03$							Average Cohort Numbers	Average number of Patients
	Dose	1	2	3	4	5			
Scenario11		10	15	20	25	30	none		
D&S	%MTD	28.7	11.6	0.2	0	0	59.5	5.2	82.8
	#Pts	33	25.6	22	2.1	0			
	%AE	10	14.8	20	23.8	NaN			
TPI	%MTD	23	2.8	0.1	0	0	74.1	7.1	71.2
	#Pts	34.73	19.26	15.73	1.35	0.08			
	%AE	9.9	14.75	20.28	25.93	37.5			
Scenario12		1	1	1	2	2	none		
D&S	%MTD	0	0.4	3.2	11.1	85.6	0	5.5	100
	#Pts	0	0.4	15.1	23.8	60.8			
	%AE	NaN	0	0.7	2.1	2			
TPI	%MTD	0	0.5	4	14.7	80.8	0	10	100
	#Pts	0.14	0.89	16.51	20.5	61.96			
	%AE	0	1.12	0.91	1.9	2			
Scenario13		0.01	3	20	20	20	none		
D&S	%MTD	6.7	92.5	0.8	0	0	0	6.4	100
	#Pts	2.8	73.8	20.8	2.3	0.2			
	%AE	0	3	19.7	21.7	0			
TPI	%MTD	12.1	87.5	0.3	0.1	0	0	10	100
	#Pts	23.52	57.97	16.81	1.52	0.81			
	%AE	0	3	19.75	19.74	22.22			

Table 4.8 returns 3 common simulation results for $p_T = 3\%$ but with start dose as Dose1. Table 4.9 is corresponding results for three uncommon scenarios.

Also when $p_T = 10\%$, we conducted simulation studies with 4 active doses $d_1, ..., d_4$, with starting dose as Dose2(Table 4.10). In this case, we get similar result as $p_T = 3\%$. That our method can save 3-4 cohort numbers and improve a little bit of the power compared to TPI.

Table 4.8: Simulation Results for D&S and TPI Common Scenarios: $p_T = 0.03$, StartDose=1.

		Different Dose levels $p_T = 0.03$						Average Cohort Numbers	Average number of Patients
	Dose	1	2	3	4	5			
Scenario1		1	2	3	10	15	none		
D&S	%MTD	3.2	13.8	64.8	16.3	1.4	0	7.6	100
	#Pts	14.7	25.7	33	21.9	4.3			
	%AE	0.7	1.9	2.7	10	14			
TPI	%MTD	6.3	21.1	58.3	12.8	0.8	0	10	100
	#Pts	18.67	28.11	30.39	17.95	4.31			
	%AE	0.96	1.89	2.99	10.14	15.08			
Scenario2		1	3	10	15	20	none		
D&S	%MTD	9	77.1	11.6	2.1	0	0	6.9	100
	#Pts	15.6	49.9	28.5	5.5	0.5			
	%AE	0.6	3	10.2	14.5	20			
TPI	%MTD	15.4	71.4	11.5	0.9	0.1	0	10	100
	#Pts	26.86	44.57	22.19	4.99	0.82			
	%AE	0.93	3.01	10	15.63	17.07			
Scenario3		0.5	1	2	3	10	none		
D&S	%MTD	0.3	4	16.1	69.1	10.4	0	7.7	100
	#Pts	11.7	19.4	23	25.1	20.8			
	%AE	0.9	1	1.7	2.8	10.1			
TPI	%MTD	2.5	5	21.9	61.2	9.3	0	10	100
	#Pts	14.64	15.68	24.45	26.05	19.09			
	%AE	0.55	0.89	1.96	2.99	10.11			

Table 4.9: Simulation Results for D&S and TPI Uncommon Scenarios: $p_T = 0.03$, StartDose=1.

		Different Dose levels $p_T = 0.03$						Average Cohort Numbers	Average number of Patients
	Dose	1	2	3	4	5			
Scenario11		10	15	20	25	30	none		
D&S	%MTD	17.5	0.2	0	0	0	82.3	3.5	56.3
	#Pts	44.4	11.1	0.8	0	0			
	%AE	9.9	15.3	25	NaN	NaN			
TPI	%MTD	13.6	0.4	0	0	0	86	4.7	47.3
	#Pts	38.89	7.28	1.03	0.13	0.01			
	%AE	10.05	15.11	21.36	15.38	100			
Scenario12		1	1	1	2	2	none		
D&S	%MTD	0.8	2.1	6.5	22.4	67.8	0	7.3	99.7
	#Pts	13.5	19.2	17.8	19.5	29.7			
	%AE	0.7	1	1.1	2.1	2			
TPI	%MTD	4.2	4.2	9.1	12.6	69.2	0	9.9	99.4
	#Pts	16.69	15.39	13.93	16.34	37.08			
	%AE	0.96	1.04	0.93	1.9	1.97			
Scenario13		0.01	3	20	20	20	none		
D&S	%MTD	7.8	91.3	0.7	0.2	0	0	5.9	100
	#Pts	12.6	69.7	16.9	0.8	0			
	%AE	0	3	20.1	25	NaN			
TPI	%MTD	11.4	87.4	1.1	0.1	0	0	10	100
	#Pts	26.34	59.29	12.95	1.28	0.14			
	%AE	0	2.92	19.92	21.09	21.43			

Table 4.10: Simulation Results for D&S and TPI: $p_T = 0.1$, StartDose=2.

	Different Dose levels $p_T = 0.1$						Average Cohort Numbers	Average number of Patients
	Dose	1	2	3	4			
Scenario1		10	30	45	55	none		
D&S	%MTD	91.3	0	0	0	0	5.7	96.5
	#Pts	74.3	21.6	0.6	0			
	%AE	10	30.1	50	NaN			
TPI	%MTD	90.5	0	0	0	0	9.6	95.8
	#Pts	77.31	18.04	0.42	0			
	%AE	9.9	29.99	47.62	NaN			
Scenario2		1	10	30	50	none		
D&S	%MTD	9.5	90.4	0.1	0	0	6.6	100
	#Pts	7	78.6	14	0.4			
	%AE	1.4	9.8	30	50			
TPI	%MTD	9.3	90.2	0.5	0	0	10	100
	#Pts	10.37	75.1	14.19	0.34			
	%AE	0.96	9.93	29.67	50			
Scenario3		1	5	10	30	none		
D&S	%MTD	0.2	14.6	84.5	0.7	0	7.1	100
	#Pts	0.3	23.6	60.7	15.4			
	%AE	0	5.1	9.9	29.9			
TPI	%MTD	0.4	15.7	82.6	1.3	0	10	100
	#Pts	1.19	26.5	59.71	12.6			
	%AE	0.84	4.87	10.02	29.68			

4.5 Discussion

We have proposed a new dose-finding algorithm based on conjugate Beta prior and some new rules. Simulation results indicate that with appropriate parameters, *D&S* design performs better at estimating the target dose and at subject assignment of the target dose.

This new method may also appeal to physicians while its implementation and computation are very simple. To implement this new method, we will need to specify the target toxicity probability p_T, the number of doses D and true toxicity probabilities for each dose to start simulation study.

The main distinction of this new proposed method is: it requires all information from current dose, lower dose and higher dose to decide dose assignment action. And we will adapt cohort size at the same time when some specified criteria are satisfied.

APPENDIX 1

Codes for Data Normalization and Fisher-Yates Transformation

Table 11: Description for Function or Package

Name	Description
DNAMR	R package with routines for microarray data analysis.
My.qn(x,y)	Function for Quantile (y missing) and Fisher-Yates normalization.
fyqtest	Simulation function for two-group gene comparative experiments.
FYsim	Simulation function for scoring scale data.

```
    library(DNAMR)
My.qn = function(x, y){
            xm < −apply(x, 2, sort)
            xxm < −if(missing(y)) f.rmedian.na(xm) else
qnorm((1 : nrow(x))/(nrow(x) + 1))
            xr < −c(apply(x, 2, rank))
            array(approx(1 : nrow(x), xxm, xr)$y, dim(x), dimnames(x))
      }
f.rmedian.na < −function(x){
            n < −nrow(x)
            p < −ncol(x)
            xm < −rep(NA, n)
            nax < −is.na(x)
            nai < −c(nax% ∗ %rep(1, p))
            x < −t(x)
            for(i in unique(nai)){
                  j < −nai == i
                  xi < −c(x[, j])
            xi < -array(xi[!is.na(xi)], c(p - i, sum(j)))
            xm[j] < -f.cmedian(xi)
        }
        xm
    }
```

Simulation Function for Two-Group Gene Comparative Experiment

```
fyqtest < -function(n1, s1, s2){
    for(dd in c(6, 10, 20, 50, 100)){
        n = 10000
        c0 = -qt(0.025, dd * 2 - 2)
        set.seed(111)
        k1 = array(rgamma(n * dd, 3), dim = c(n, dd))
        k2 = t(array(c(rgamma(n1 * dd, scale < -rep(c(s1, s2), rep(n1 *
dd/2, 2))), rgamma((n - n1) * dd, 3)), dim = c(dd, n)))
        #k1 = array(rnorm(n * dd), dim = c(n, dd))
        #k2 = t(array(c(rnorm(n1*dd, scale < -rep(c(s1, s2), rep(n1*dd/2, 2))),
        rnorm((n - n1) * dd)), dim = c(dd, n)))
        kfy = My.qn(cbind(k1, k2), 1)
        kqn = My.qn(cbind(k1, k2))
        library(DNAMR)
        pfy = c(f.rtt(kfy[, (1 : dd)], kfy[, -(1 : dd)]))
        pqn = c(f.rtt(kqn[, 1 : dd], kqn[, -(1 : dd)]))
        pt0 = c(f.rtt(k1, k2))
        pp0 = 1 * cbind(abs(pt0) > c0, abs(pfy) > c0, abs(pqn) > c0)
        cat("nsamples = ", dd, "\n")
        print(apply(pp0, 2, function(x, nn)c(mean(x[1 : nn]), mean(x[-(1 :
```

```
nn)])), nn = n1))
      }
  }
```

Simulation Function for Scoring Scale Data

```
  FYsim < -function(sim){
    yy < -rep(0, 4); nsim = rep(0, 4)
    for(k in 1 : sim){
          j = table(sample(3, nn[1], rep = T))
          u1 = c(sample(0 : 10, j[1] ∗ 10, replace = T, prob = vhtop),
          sample(0 : 10, j[2] ∗ 10, replace = T, prob = vhmid),
          sample(0 : 10, j[3] ∗ 10, replace = T, prob = vhbot))
          dim(u1) = c(10, nn[1])
          u1 = t(u1)
          j = table(sample(3, nn[2], rep = T))
          u2 = rbind(array(c(sample(0 : 10, j[1]∗5, replace = T, prob = vhtop),
          sample(0 : 10, j[1] ∗ 5, replace = T, prob = vltop)), dim = c(j[1], 10)),
          array(c(sample(0 : 10, j[2]∗5, replace = T, prob = vhmid), sample(0 :
10, j[2] ∗ 5, replace = T, prob = vlmid)), dim = c(j[2], 10)),
          array(c(sample(0 : 10, j[3] ∗ 5, replace = T, prob = vhbot), sample(0 :
10, j[3] ∗ 5, replace = T, prob = vlbot)), dim = c(j[3], 10)))
          j = table(sample(3, nn[3], rep = T))
          u3 = rbind(array(c(sample(0 : 10, j[1]∗5, replace = T, prob = vltop),
          sample(0 : 10, j[1] ∗ 5, replace = T, prob = vhtop)), dim = c(j[1], 10)),
          array(c(sample(0 : 10, j[2] ∗ 5, replace = T, prob = vlmid), sample(0 :
10, j[2] ∗ 5, replace = T, prob = vhmid)), dim = c(j[2], 10)),
          array(c(sample(0 : 10, j[3] ∗ 5, replace = T, prob = vlbot), sample(0 :
```

```
10, j[3] * 5, replace = T, prob = vhbot)), dim = c(j[3], 10)))
            u = rbind(u1, u2, u3)
            kk = c(apply(u, 1, sd) == 0)
            uu < -array(, dim = c(1200, 10))
            for(i in 1 : 1200){uu[i, ] = u[i, ] - median(u[i, ])}
            qnnorm < -My.qn(t(uu))
            fynorm < -My.qn(t(uu), 1)
            znorm < -t(apply(uu, 1, function(z)(z - mean(z))/sd(z)))
            if(any(kk))znorm[kk, ] = 0
            dn < -hclust(dist(uu))
            dz = hclust(dist(znorm), method = ”ward.D”)
            dqn = hclust(dist(t(qnnorm)), method = ”ward.D”)
            dfy = hclust(dist(t(fynorm)), method = ”ward.D”)
            nna < -sum(diag(comp3(table(cutree(dn, 3), rep(1 : 3, nn)))))
            nz < -sum(diag(comp3(table(cutree(dz, 3), rep(1 : 3, nn)))))
            nq < -sum(diag(comp3(table(cutree(dqn, 3), rep(1 : 3, nn)))))
            nfy < -sum(diag(comp3(table(cutree(dfy, 3), rep(1 : 3, nn)))))
            nsim[1] = nsim[1]+nna; nsim[2] = nsim[2]+nz; nsim[3] = nsim[3]+
nq; nsim[4] = nsim[4] + nfy
            }
      yy < -nsim/sim
      return(yy)
   }
```

APPENDIX 2

Codes for Hierarchical Clustering Tree Prediction

Table 12: Description for Function or Package

Name	Description
ward.pred	Our hierarchical clustering function with “ward” linkage.
Simward	Function for simulation study.

Our prediction function with “ward” linkage. The function for “single” linkage and “complete” linkage etc. is the same just by replacing method=“ward” with method=“single” or method=“complete”.

```
ward.pred = function(z1, z2, nc){
  hq = hclust(dist(z1), method = ”ward”)
  cl1 = cutree(hq, nc)
  nn1 = NULL; nn2 = NULL
  zm = NULL
  for(i in 1 : nc){
        zm = cbind(zm, apply(z1[cl1 == i, ], 2, mean, na.rm = T))
        }
  for(j in 1 : nrow(z2)){
        zz1p = rbind(z1, z2[j, ])
        n1 = nrow(zz1p)
        hq1 = hclust(dist(zz1p), method = ”ward”)
        for(i in 1 : (n1 − nc)){
           cl2 = cutree(hq1, n1 − i)
           jj < −(cl2[n1] == cl2[−n1])
           if(any(jj))break
```

```
          }
          nn1[j] = cl1[jj][1]
          nn2[j] = which.min(apply((zm - unlist(z2[j,]))^2, 2, sum, na.rm = T))
      }
      cbind(nn1, nn2)
  }
```

Function for simulation study

```
Simward < -function(n, Sim){
  r11 < -r12 < -r21 < -r22 < -rep(0, Sim)
  for(tin1 : Sim){
          x = rnorm(2 * n)
          dim(x) = c(n, 2)
          r = apply(x, 1, norm, ”2”)
          i = (r < 1.5&x[, 1] >= 0)|(x[, 1] < 0&abs(x[, 2]) < 1.5)
          ii = (r < 2.25&x[, 1] >= 0)|(x[, 1] < 0&abs(x[, 2]) < 1.5)
          y = rnorm(4 * n) * 3
          dim(y) = c(2 * n, 2)
          y[, 1] = y[, 1] + 1.5
          ry = apply(y, 1, norm, ”2”)
          j = (ry > 3&y[, 1] >= 0&y[, 2] <= 1)
          jj = (ry > 2.25&y[, 1] >= 0&y[, 2] <= 1)
          xx = x[-(1 : 100), ][ii[-(1 : 100)], ]
          yy = y[-(1 : 100), ][jj[-(1 : 100)], ]
          x0 = x[1 : 100, ][i[1 : 100], ]
          y0 = y[1 : 100, ][j[1 : 100], ]
```

```
        $x1 = xx$
        $y1 = yy$
        $u0 < -rbind(x0, y0)$
        $u1 < -rbind(x1, y1)$
        $ward.pred(u0, u1, nc = 2) - > r$
        $r11[t] = 1 - nrow(x1[r[1 : nrow(x1), 1] == 1, ])/nrow(x1)$
        $r12[t] = 1 - nrow(y1[r[(nrow(x1)+1) : nrow(u1), 1] == 2, ])/nrow(y1)$
        $r21[t] = 1 - nrow(x1[r[1 : nrow(x1), 2] == 1, ])/nrow(x1)$
        $r22[t] = 1 - nrow(y1[r[(nrow(x1)+1) : nrow(u1), 2] == 2, ])/nrow(y1)$
    }
    $rr < -rep(0, 6)$
    $rr[1] < -sum(r11)/Sim; rr[2] < -sum(r12)/Sim; rr[3] < -sum(r21)/Sim;$
    $rr[4] < -sum(r22)/Sim$
    $rr[5] < -rr[1] + rr[2]; rr[6] < -rr[3] + rr[4]$
    $return(rr)$
}
```

APPENDIX 3

Simulation Function for Dose and Cohort Size Adaptive Design

$Adpttrs <-function(p, itr)\{$

$nnn <-rep(0, nd); xxx <-rep(0, nd); Nco <-0; rez <-rep(0, itr)$

$N = sub * gp$

$set.seed(123)$

$for(m\ in\ 1 : itr)\{$

$nn <-xx <-x <-sig <-miu <-vary <-qd <-qs <-qe <-qee <-tau <-pa <-pb <-skew <-rep(0, nd)$

$dose = stdose; st <-0; nodose <-0; maxdose <-1; toxdose <-nd + 1; seldose <-0; Nc = 0$

$n <-rep(sub, nd)$

$for(j\ in\ 1 : gp)\{$

$if(st == 0\&sum(nn) < N\&dose >= 1\&dose <= nd)\{$

$Nc = Nc + 1$

$maxdose <-max(maxdose, dose)$

$d = dose$

$x[d] <-rbinom(1, n[d], p[d])$

$nn[d] = nn[d] + n[d]; xx[d] = xx[d] + x[d]$

$mm <-N - sum(nn)$

$m0 = min(mm, sub)$

$m1 = min(mm, sub + aa)$

$m2 = min(mm, sub + 2 * aa)$

$m3 = min(mm, sub + 3 * aa)$

$m4 = min(mm, sub + 4 * aa)$

$pa[d] <-a + xx[d]; pb[d] <-b + nn[d] - xx[d]$

$miu[d] = pa[d]/(pa[d] + pb[d])$

$sig[d] <- sqrt((pa[d] * pb[d])/((pa[d] + pb[d])^2 * (pa[d] + pb[d] + 1)))$

$vary[d] = sig[d]/miu[d]$

$skew[d] = 2 * (pb[d] - pa[d]) * sqrt(pa[d] + pb[d] + 1)/((pa[d] + pb[d] + 2) * sqrt(pa[d] * pb[d]))$

$if(abs(skew[d]) > w)\{$

$qd[d] = pbeta(pt * exp(K2 * vary[d]), pa[d], pb[d], lower.tail = FALSE)$

$qs[d] = pbeta(pt * exp(K2 * vary[d]), pa[d], pb[d]) - pbeta(pt * exp(-K1 * vary[d]),$

$pa[d], pb[d])$

$qe[d] = pbeta(pt * exp(-K1 * vary[d]), pa[d], pb[d])$

$\}$

$else\{$

$qd[d] = pbeta(pt + v2 * sig[d], pa[d], pb[d], lower.tail = FALSE)$

$qs[d] = pbeta(pt + v2 * sig[d], pa[d], pb[d]) - pbeta(pt - v1 * sig[d],$

$pa[d], pb[d])$

$qe[d] = pbeta(pt - v1 * sig[d], pa[d], pb[d])$

$\}$

$if(pbeta(pt, pa[d], pb[d], lower.tail = FALSE) > cut)\{tau[d : nd] = 1\}$

$else\{tau[d] = 0\}$

$if(tau[1] == 1)\{st = 1; nodose <- 1; break\}$

$if(d < nd)\{qee[d] = qe[d] * (1 - tau[d + 1])\}$

$elseqee[nd] = qe[nd]$

$A = max(qd[d], qs[d], qee[d])$

$if((d > 1)\&(d < nd))\{$

$if(qee[d] == A\&qee[d] > u1\&tau[d + 1] == 1)\{toxdose = dose + 1; dose = dose; n[dose] = mm\}$

$if(qee[d] == A\&qee[d] > u1\&tau[d{+}1] == 0\&qd[d{+}1] > 0)\{dose = dose + 1; n[dose] = m1\}$

$if(qee[d] == A\&qee[d] > u1\&tau[d + 1] == 0\&qd[d + 1] == 0)\{dose = dose + 1; n[dose] = m1\}$

$if(qee[d] == A\&qee[d] <= u1\&tau[d + 1] == 1)\{toxdose = dose + 1; dose = dose; n[dose] = m4\}$

$if(qee[d] == A\&qee[d] <= u1\&tau[d + 1] == 0\&qd[d + 1] > 0)\{dose = dose + 1; n[dose] = m0\}$

$if(qee[d] == A\&qee[d] <= u1\&tau[d + 1] == 0\&qd[d + 1] == 0)\{dose = dose + 1; n[dose] = m0\}$

$if(qs[d] == A\&tau[d + 1] == 1)\{toxdose < -dose + 1; dose = dose; n[dose] = m3\}$

$if(qs[d] == A\&tau[d{+}1] == 0\&qd[d{+}1] > 0)\{dose = dose; n[dose] = m2\}$

$if(qs[d] == A\&tau[d{+}1] == 0\&qd[d{+}1] == 0)\{dose = dose; n[dose] = m0\}$

$if(qd[d] == A\&tau[d] == 0\&qee[d{-}1] > u1)\{dose = dose; n[dose] = m1\}$

$if(qd[d] == A\&tau[d] == 0\&qee[d - 1] <= u1\&qee[d - 1] > 0)\{dose = dose; n[dose] = m0\}$

$if(qd[d] == A\&tau[d] == 0\&qee[d - 1] <= u1\&qee[d - 1] == 0)\{dose = dose - 1; n[dose] = m0\}$

$if(tau[d] == 1\&qee[d - 1] > u1)\{toxdose < -dose; dose = dose - 1; n[dose] = mm\}$

$if(tau[d] == 1\&qee[d - 1] <= u1\&qee[d - 1] > 0)\{toxdose < -dose; dose = dose - 1; n[dose] = m4\}$

$if(tau[d] == 1\&qee[d - 1] <= u1\&qee[d - 1] == 0)\{toxdose < -dose; dose = dose - 1; n[dose] = m0\}$

```
}
if(d == 1){
    if(qee[d] == A&qee[d] > u1&tau[d + 1] == 1){toxdose = dose + 1; dose = dose; n[dose] = mm}
    if(qee[d] == A&qee[d] > u1&tau[d+1] == 0&qd[d+1] > 0){dose = dose + 1; n[dose] = m1}
    if(qee[d] == A&qee[d] > u1&tau[d + 1] == 0&qd[d + 1] == 0){dose = dose + 1; n[dose] = m1}
    if(qee[d] == A&qee[d] <= u1&tau[d+1] == 1){dose = dose; n[dose] = m4}
    if(qee[d] == A&qee[d] <= u1&tau[d + 1] == 0&qd[d + 1] > 0){dose = dose + 1; n[dose] = m0}
    if(qee[d] == A&qee[d] <= u1&tau[d + 1] == 0&qd[d + 1] == 0){dose = dose + 1; n[dose] = m1}
    if(qs[d] == A&tau[d + 1] == 1){toxdose < -dose + 1; dose = dose; n[dose] = m3}
    if(qs[d] == A&tau[d+1] == 0&qd[d+1] > 0){dose = dose; n[dose] = m2}
    if(qs[d] == A&tau[d+1] == 0&qd[d+1] == 0){dose = dose; n[dose] = m0}
    if(qd[d] == A&tau[d] == 0){dose = dose; n[dose] = m0}
}
if(d == nd){
    if(qee[d] == A&qee[d] > u1){dose = dose; n[dose] = mm}
    if(qee[d] == A&qee[d] <= u1){dose = dose; n[dose] = m4}
    if(qs[d] == A&qee[d - 1] > u1){dose = dose; n[dose] = m3}
    if(qs[d] == A&qee[d - 1] <= u1){dose = dose; n[dose] = m2}
    if(qd[d] == A&tau[d] == 0&qee[d-1] > u1){dose = dose; n[dose] =
m1}
          if(qd[d] == A&tau[d] == 0&qee[d − 1] <= u1&qee[d − 1] > 0){dose = dose; n[dose] = m0}
          if(qd[d] == A&tau[d] == 0&qee[d − 1] <= u1&qee[d − 1] == 0){dose = dose − 1; n[dose] = m0}
          if(tau[d] == 1&qee[d − 1] > u1){toxdose < −dose; dose = dose − 1; n[dose] = mm}
          if(tau[d] == 1&qee[d − 1] <= u1&qee[d − 1] > 0){toxdose < −dose; dose = dose − 1; n[dose] = m4}
          if(tau[d] == 1&qee[d − 1] <= u1&qee[d − 1] == 0){toxdose < −dose; dose = dose − 1; n[dose] = m0}
        }
      }
    }
    Nco = Nco + Nc
      if(nodose == 0){
        tdose < −min(maxdose, toxdose − 1)
        pp < −rep(−100, tdose)
        pp.var < −rep(0, tdose)
        for(iin1 : tdose){
          if(pa[i]! = 0){
          pp[i] < −pa[i]/(pa[i] + pb[i])
          pp.var[i] < −(pa[i] ∗ pb[i])/((pa[i] + pb[i])^2 ∗ (pa[i] + pb[i] + 1))
        }
      }
      pp < −pava(pp, wt = 1/pp.var)
      for(iin2 : tdose){
        pp[i] < −pp[i] + i ∗ 1E − 10
    }
    seldose < −order(abs(pp − pt))[1]
}
rez[m] < −seldose;
for(i in 1 : nd){
    nnn[i] = nnn[i] + nn[i]
    xxx[i] = xxx[i] + xx[i]}
}
    aaa < −rep(0, nd)
    for(i in 1 : nd){aaa[i] < −sum(rez == i)/itr}
        sbn = round(nnn/itr, 1)
        ae = round(xxx/itr, 1)
        r = round(100 ∗ ae/sbn, 1)
        select = round(100 ∗ aaa, 1)
        NNco < −round(Nco/itr, 1)
        ntotal = round(sum(nnn)/itr, 1)
        ww < −cbind(select, sbn, ae, r)
        result = c(as.vector(ww), NNco, ntotal)
        return(result)
}
```

Bibliography

Amaratunga, D. and Cabrera, J. (2001). Analysis of data from viral dna microchips. *Journal of the American Statistical Association*, 96(456):1161–1170.

Amaratunga, D. and Cabrera, J. (2004). *Exploration and analysis of DNA microarray and protein array data*, volume 446. John Wiley & Sons.

Barlow, R. (1972). Experiments in border disease: Iv. pathological changes in ewes. *Journal of comparative pathology*, 82(2):151–157.

Durbin, B. P., Hardin, J. S., Hawkins, D. M., and Rocke, D. M. (2002). A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics*, 18(suppl 1):S105–S110.

Freeman, R., Baron, R., Bouhassira, D., Cabrera, J., and Emir, B. (2014). Sensory profiles of patients with neuropathic pain based on the neuropathic pain symptoms and signs. *PAIN®*, 155(2):367–376.

Gasparini, M. and Eisele, J. (2000). A curve-free method for phase i clinical trials. *Biometrics*, 56(2):609–615.

Hanson, D. L., Pledger, G., and Wright, F. (1973). On consistency in monotonic regression. *The Annals of Statistics*, pages 401–421.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. (2009). *The elements of statistical learning*, volume 2. Springer.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264.

Ji, Y., Li, Y., and Bekele, B. N. (2007). Dose-finding in phase i clinical trials based on toxicity probability intervals. *Clinical Trials*, 4(3):235–244.

Korn, E. L., Midthune, D., Chen, T. T., Rubinstein, L. V., Christian, M. C., and Simon, R. M. (1994). A comparison of two phase i trial designs. *Statistics in medicine*, 13(18):1799–1806.

Li, C. and Wong, W. H. (2001). Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proceedings of the National Academy of Sciences*, 98(1):31–36.

O'Quigley, J. and Chevret, S. (1991). Methods for dose finding studies in cancer clinical trials: a review and results of a monte carlo study. *Statistics in medicine*, 10(11):1647–1664.

O'Quigley, J., Pepe, M., and Fisher, L. (1990). Continual reassessment method: a practical design for phase 1 clinical trials in cancer. *Biometrics*, pages 33–48.

O'QUIGLEY, J. and Reiner, E. (1998). A stopping rule for the continual reassessment method. *Biometrika*, 85(3):741–748.

O'Quigley, J. and Shen, L. Z. (1996). Continual reassessment method: a likelihood approach. *Biometrics*, pages 673–684.

Rosenberger, W. F. and Haines, L. M. (2002). Competing designs for phase i clinical trials: a review. *Statistics in Medicine*, 21(18):2757–2770.

Sapir, M. and Churchill, G. A. (2000). Estimating the posterior probability of differential gene expression from microarray data.

Schadt, E. E., Li, C., Ellis, B., and Wong, W. H. (2001). Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. *Journal of Cellular Biochemistry*, 84(S37):120–125.

Storer, B. E. (1989). Design and analysis of phase i clinical trials. *Biometrics*, pages 925–937.

Stylianou, M. and Flournoy, N. (2002). Dose finding using the biased coin up-and-down design and isotonic regression. *Biometrics*, 58(1):171–177.

Zhu, M. and Lu, A. Y. (2004). The counter-intuitive non-informative prior for the bernoulli family. *Journal of Statistics Education*, 12(2):1–10.