Virginia Commonwealth University
VCU Scholars Compass

Theses and Dissertations | Graduate School

2017

The Generalized Monotone Incremental Forward Stagewise Method for Modeling Longitudinal, Clustered, and Overdispersed Count Data: Application Predicting Nuclear Bud and Micronuclei Frequencies

Rebecca Lehman

Follow this and additional works at: https://scholarscompass.vcu.edu/etd

 Part of the Biostatistics Commons, Longitudinal Data Analysis and Time Series Commons, and the Microarrays Commons

© The Author

Downloaded from
https://scholarscompass.vcu.edu/etd/4957

This Dissertation is brought to you for free and open access by the Graduate School at VCU Scholars Compass. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of VCU Scholars Compass. For more information, please contact libcompass@vcu.edu.

©Rebecca R. Lehman, June 2017

All Rights Reserved.

THE GENERALIZED MONOTONE INCREMENTAL FORWARD STAGEWISE METHOD FOR MODELING LONGITUDINAL, CLUSTERED, AND OVERDISPERSED COUNT DATA: APPLICATION PREDICTING NUCLEAR BUD AND MICRONUCLEI FREQUENCIES

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

REBECCA R. LEHMAN

B.S. Mathematics, The University of the South, USA, 2012

Director: Kellie J. Archer, Ph.D.,

Chair and Professor, Division of Biostatistics,

College of Public Health at The Ohio State University

Virginia Commonwealth University

Richmond, Virginia

June, 2017

Acknowledgements

First, I would like to thank my adviser, Dr. Kellie J. Archer, for working with me for the past 5 years, even when our work became long distance. Without her patient guidance, I would not have made it this far and grown into an independent researcher. I would also like to thank my dissertation committee members Dr. Nitai Mukhopadhyay, Dr. Roy Sabo, Dr. Colleen Jackson-Cook, and Dr. Nastassja Lewinski for their many recommendations during our frequent committee meetings. Thank you to Russ for recruiting me to the program and for his consistent help and advice every step of the way.

To all of my family members, and especially my parents, thank you for your support for these last 27 years of my life and for knowing my potential even when I couldn't see it. Thank you to my many second mothers, Ann, Deborah, Mrs. Bickley, Lee, Ann, and Mrs. Dunne for always being there for me to offer advice whether it be in career guidance, career dress, or just about life. To Miss Lettie, thank you for your devoted love!

I would like to thank Dr. Chris Parrish who dedicated an entire semester my senior year at Sewanee to an independent study to help me discover Biostatistics. Also, thank you to Dr. Stephen Carl who introduced me to computer science and whose emails containing nerdy jokes and Sewanee gossip always make me smile. To my Sewanee friends, especially Winnie, Blair, Claire, Virginia, Erin, Alex, and Dani, thank you for the many reunions, too much food, and too many drinks. You have put up with listening about biostatistics and have kept me sane these past 5 years!

Thank you Emily for always reminding me there is a light at the end of the tunnel. To Kyle and Amanda, we did it! Without your support this process would have taken much longer. We never turned graduate school into a competition and

have always been there to help each other, even with silly questions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract

THE GENERALIZED MONOTONE INCREMENTAL FORWARD STAGEWISE METHOD FOR MODELING LONGITUDINAL, CLUSTERED, AND OVERDISPERSED COUNT DATA: APPLICATION PREDICTING NUCLEAR BUD AND MICRONUCLEI FREQUENCIES

By Rebecca R. Lehman

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2017.

Director: Kellie J. Archer, Ph.D.,
Chair and Professor, Division of Biostatistics,
College of Public Health at The Ohio State University

With the influx of high-dimensional data there is an immediate need for statistical methods that are able to handle situations when the number of predictors greatly exceeds the number of samples. One such area of growth is in examining how environmental exposures to toxins impact the body long term. The cytokinesis-block micronucleus assay can measure the genotoxic effect of exposure as a count outcome. To investigate potential biomarkers, high-throughput assays that assess gene expression and methylation have been developed. It is of interest to identify biomarkers or molecular features that are associated with elevated micronuclei (MN) or nuclear bud (Nbud) frequency, measures of exposure to environmental toxins.

Given our desire to model a count outcome (MN and Nbud frequency) using high-throughput genomic features as predictors, novel methods that can handle over-

parameterized models need development. Overdispersion, when the variance of a count outcome is larger than its mean, is frequently observed with count response data. For situations where overdispersion is present, the negative binomial distribution is more appropriate. Furthermore, we expand the method to the longitudinal Poisson and longitudinal negative binomial settings for modeling a longitudinal or clustered outcome both when there is equidispersion and overdispersion. The method we have chosen to expand is the Generalized Monotone Incremental Forward Stagewise (GMIFS) method. We extend the GMIFS to the negative binomial distribution so it may be used to analyze a count outcome when both a high-dimensional predictor space and overdispersion are present. Our methods were compared to `glmpath`. We also extend the GMIFS to the longitudinal Poisson and longitudinal negative binomial distribution for analyzing a longitudinal outcome. Our methods were compared to `glmmLasso` and `GLMMLasso`. The developed methods were used to analyze two datasets, one from the Norwegian Mother and Child Cohort study and one from the breast cancer epigenomic study conducted by researchers at Virginia Commonwealth University. In both studies a count outcome measured exposure to potential genotoxins and either gene expression or high-throughput methylation data formed a high dimensional predictor space. Further, the breast cancer study was longitudinal such that outcomes and high-dimensional genomic features were collected at multiple time points during the study for each patient. Our goal is to identify biomarkers that are associated with elevated MN or NBud frequency. From the development of these methods, we hope to make available more comprehensive statistical models for analyzing count outcomes with high dimensional predictor spaces and either cross-sectional or longitudinal study designs.

CHAPTER 1

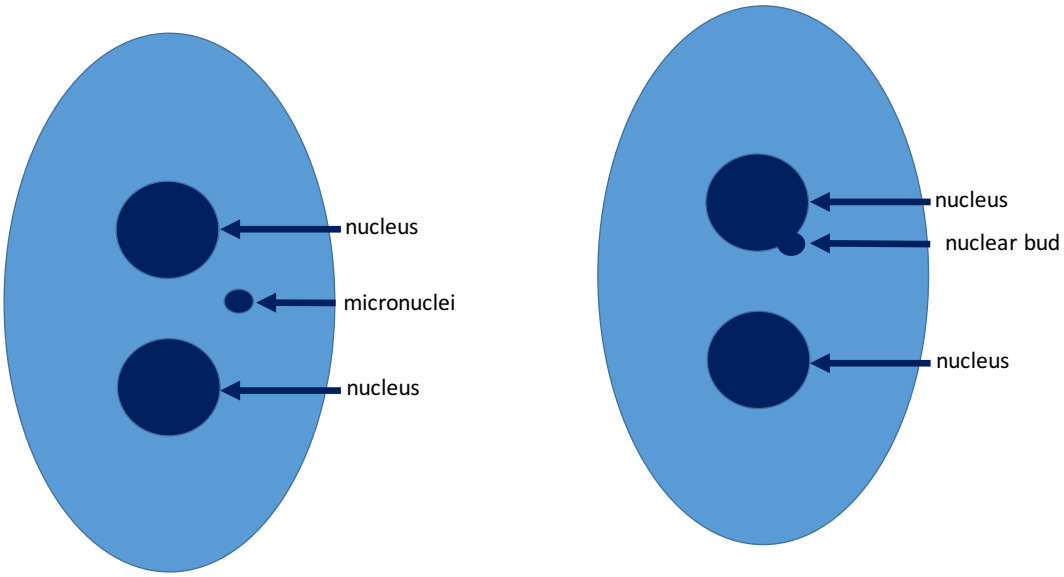
BACKGROUND

1.1 Introduction

1.1.1 Micronuclei and Nuclear Buds

Micronuclei (MN) and Nuclear Buds (NBuds) are nuclear bodies that are formed in cells during the process of cell division in which DNA damage has occurred. Their presence may signify genome damage events and indicate chromosomal instability[11]. MN are formed in dividing cells from fragments or whole chromosomes lagging behind that do not attach to the mitotic spindle prior to cytokinesis[8]. Rather than these fragments or whole chromosomes becoming part of the main nucleus, they are enveloped into an independent, smaller nucleus. Previous research has shown MN to be a reliable and precise method for assessing chromosome damage[8]. There have also been implications that MN formed by mutagens may play a role in carcinogenesis[22]. NBuds form similarly, however, unlike MN, NBuds are still attached to the nucleus by nucleoplasmic material. It is thought NBuds develop from the elimination of nuclear material from the nucleus, the elimination of amplified DNA, or the shrinkage of a broken nucleoplasmic bridge[11]. Figure 1 displays ideograms of a micronuclei and nuclear bud.

Fig. 1: Ideograms of two binucleated cells with the presence of a micronuclei (left) and a nuclear bud (right).



The assay which is used to identify micronuclei and nuclear buds will be described in Section 1.1.2. The process in which these nuclear bodies form will also be depicted.

1.1.2 Cytokinesis-block Micronucleus (CBMN) Assay

MN assays in human lymphocytes have been developed to measure both whole chromosome loss and chromosome breaks[8]. Typically MN are observed in cells that have undergone division after DNA damage has occurred. The DNA damage can be spontaneous or arise from exposure to a genotoxic agent. The original goal of the CBMN assay was to develop a method that may better be able to identify exposure conditions that induce elevated MN counts[10]. The development of an assay relied on the ability to count MN or NBuds after at least one cell division has occurred when cells are in a binucleated state. A binucleated state means the cells must have two nuclei with intact nuclear membranes and situated within the same

cytoplasmic boundary[10]. The initial CBMN assay described uses fresh blood[8]. The cells completed karyokinesis but were stopped from performing cytokinesis by using cytochalasin-B[8,10]. Since the mechanism does not interfere with nuclear division, the binucleated cells may be counted or scored for the presence of at least one MN or NBud. Figure 2 is an ideogram of a cell division when there is no MN formation. Figure 3 is an ideogram of cell division when there is a MN formation. Figure 2 and Figure 3 both show how the cytochalasin-B stops cells from performing cytokinesis thus the final cell is binucleated or contains two main nuclei.

Fig. 2: Ideogram of the CBMN Assay mechanism. The cell has undergone nuclear division and cytochalasin-B has been applied to give rise to a binucleated cell- this binucleated cell does not contain MN.

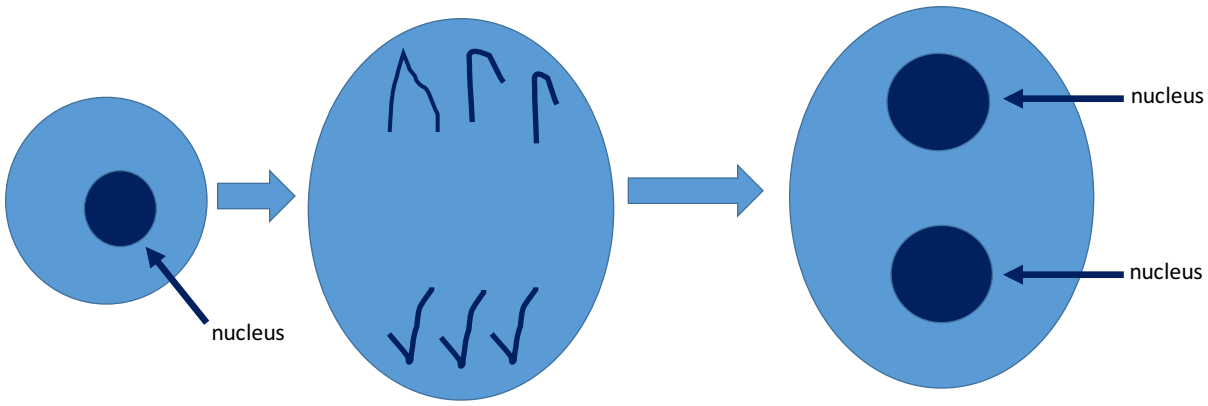
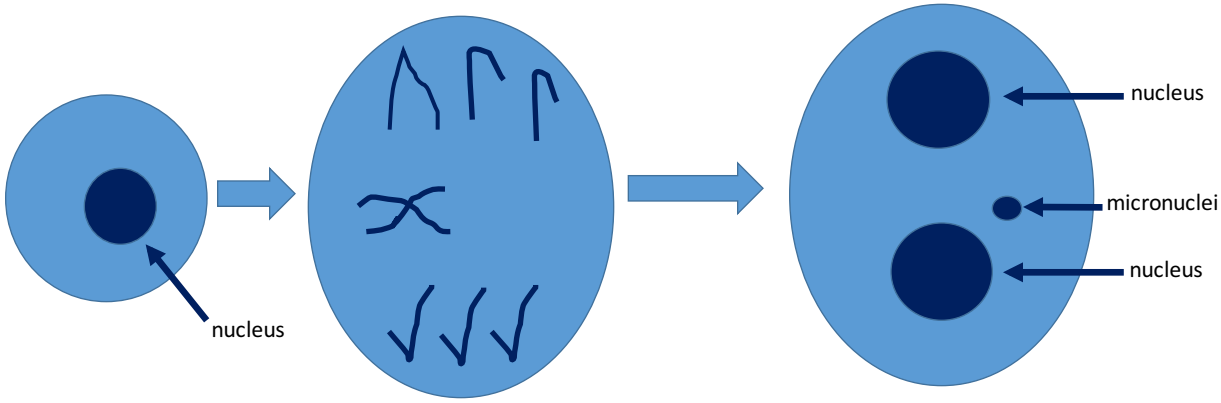


Fig. 3: Ideogram of CBMN Assay mechanism. The cell has undergone nuclear division and cytochalasin-B has been applied; however, a chromosome lags behind and does not attach to the mitotic spindle which gives rise to the MN formation.



It has been recommended to score approximately 2,000 binucleated cells[8,10]. Criteria have been developed for identifying MN including: round or oval in shape, diameter between 1/16 and 1/3 that of the main nuclei, non-refractile, not linked to the main nuclei via a nucleoplasmic bridge, and they may overlap boundaries with the main nuclei[8,10]. NBuds are characterized identically except that they are attached to one of the main nuclei[11].

1.1.3 Previous Research

The majority of previous studies have examined MN as opposed to NBuds. Confounding factors for MN were examined by Fenech et al., 1992 using a dataset consisting of 225 individuals from a South Australian population, which 155 were female and 70 were male[8]. It was determined that confounding factors included age, sex, and smoking status. There was a significant positive correlation between MN frequency and age. Females generally had higher MN frequency than males, and this was statistically significant when controlling for age. For a subset of the patients (N=156) for whom smoking data was collected there was evidence that patients who reported smoking high number of cigarettes per day had elevated MN frequency[8]. Only 29 of the 156 patients were smokers[8]. Fenech et al., 1993[8] described at least four other studies: Au et al., 1991[2], Migliore et al., 1991[26], Tomanin et al., 1991[34], and Yager et al., 1988[37] have shown a statistically significant relationship between MN frequency and age or smoking.

Other previous studies have predominantly focused on identifying exposure conditions such as pesticides, chromium, or organic solvents associated with MN frequency[8]. Population studies have been done where genotoxic chemicals (e.g. styrene, chemicals in tannery industries, paracetmol, etc[8]) are suspects. Various studies have shown that higher MN frequencies result in a higher risk of cancer development. Investigators have also shown that chemotherapy can result in inflated MN frequencies[8]. Prior research has followed testicular carcinoma patients for up to 9 years post-chemotherapy and shown increased MN frequencies[8]. Much of the previous research of MN frequency assumes Gaussian distributed data such as Ban et al., 2004[3], Guiterrez et al., 1997[16], Minozzo et al., 2004[27], and Varga et al., 2006[36]. Varga et al., 2006 found there to be a significant difference in MN frequencies between breast cancer

patients and controls, and also showed that age was a confounding variable[36]. Within breast cancer patients, those treated by irradiation showed greater MN frequencies[36]. Ban et al., 2004 also found elevated MN frequencies in female breast, head, and neck or cervical cancer patients when compared to healthy female subjects[3]. Gutierrez et al., 1997 examined the genetic damage that was attributed to therapeutic exposure to I sodium iodide, a treatment for hyperthyroidism patients[16]. The damage was measured by the presence of MN in the binucleated peripheral blood[16]. It was shown that there is a positive relationship between I dose and MN count[16]. Minozzo et al., 2004 examined workers exposed to lead and found they had significantly higher MN frequencies[27].

While the majority of the previous research focuses on identifying confounding variables or examining relationships between chemical exposures and elevated MN frequencies, we are interested in better understanding the molecular mechanisms that cause MN or NBud formation.

1.1.4 Review of statistical methods used in analyzing MN frequency

Ceppi et al. (2010) examined 63 studies published between January 2000 and August 2008 involving MN for their statistical quality and provided recommendations to improve future analyses[6]. Among those 63 studies, 98.4% considered age as a confounder, 85.7% considered gender as a confounder, and 90.5% considered smoking habit as a confounder. For 77.8% of the studies, non-parametric tests were applied, and Student's t-test was the most commonly applied test[6]. In the 63 studies examined, models were not limited to Poisson and negative binomial, instead many assumed a normal distribution. By choosing an inappropriate probability distribution, costly errors may be produced in the results. Ceppi et al. (2010) reported that better statistical models should be used when analyzing MN data. They concluded

that either Poisson or negative binomial regression would be preferred when modeling a count outcome when more than 2000 cells are scored[6].

1.2 Methods for analyzing MN data

1.2.1 Poisson Regression

Both MN and NBuds are examples of count data, a frequently occurring discrete response. Count outcomes differ from other discrete responses because they cannot be expressed in the form of several proportions, as there is no upper limit to the values they can take[32]. The Poisson distribution is the most frequently used distribution when analyzing a count or rate outcome. Poisson regression methods have been highly developed both in a traditional statistical setting where the number of samples (n) is greater than the number of predictors (p) and many extensions have been made to the high dimensional setting where $n < p$[25,29]. However, the Poisson model is limited in the amount of variability it can account for[1]. The Poisson distribution assumes a mean and variance to be equal to a single parameter, λ_i,

$$\mathbf{E}(y_i) = \text{Var}(y_i) = \lambda_i \tag{1.1}$$

where y_i is the count outcome and i indexes subjects from $i = 1, ..., n$. Assuming the mean and variance to be equal limits the Poisson distribution to only be pertinent in equidispersed settings. The Poisson probability distribution function (PDF) is,

$$f(y_i; \lambda_i) = \frac{e^{-\lambda_i} \lambda_i^{y_i}}{y_i!} \tag{1.2}$$

and the corresponding likelihood is represented by,

$$L(\lambda; \boldsymbol{y}) = \prod_{i=1}^{n} \frac{e^{-\lambda_i} \lambda_i^{y_i}}{y_i!}. \tag{1.3}$$

When using maximum likelihood estimation, it is mathematically easier and equivalent to use the corresponding log-likelihood,

$$l(\lambda; \boldsymbol{y}) = \sum_{i=1}^{n} (y_i \log \lambda_i - \lambda_i - \log(y_i!)). \tag{1.4}$$

When a rate outcome is analyzed, an offset term is incorporated in the distribution. For example, for MN data, when the total number of cells examined varies by subject, an offset should be included in the distribution. Therefore, the expected value is re-written as,

$$\mathbf{E}\left(\frac{y_i}{t_i}\right) = \lambda_i \tag{1.5}$$

where t_i is the offset term. The conditional probability is re-written as,

$$f(y_i; \lambda_i) = \frac{e^{-t_i \lambda_i} (t_i \lambda_i)^{y_i}}{y_i!} \tag{1.6}$$

for each observation i. The likelihood is represented by

$$L(\lambda; \boldsymbol{y}) = \prod_{i=1}^{n} \frac{e^{-t_i \lambda_i} (t_i \lambda_i)^{y_i}}{y_i!}. \tag{1.7}$$

Recall, it is easier to maximize the corresponding log-likelihood,

$$l(\lambda; \boldsymbol{y}) = \sum_{i=1}^{n} (y_i \log t_i \lambda_i - t_i \lambda_i - \log(y_i!)). \tag{1.8}$$

In Poisson regression the model assumes that the mean can be modeled as a linear combination of the predictors through a log link function,

$$\log(\lambda_i) = \log(t_i) + \mathbf{x}_i^\top \boldsymbol{\beta} \tag{1.9}$$

where $\boldsymbol{\beta}$ is a vector of coefficients that correspond with the predictor variables, $\mathbf{x}_i$. The log link function may be rewritten in terms of λ showing how the expected response changes with the predictors. This link function will allow the Poisson log-likelihood to be rewritten in terms of the highly interpretable predictor coefficients

$$\lambda_i = t_i + \exp(\mathbf{x}_i^\top \boldsymbol{\beta}). \tag{1.10}$$

It is often of interest to estimate $\boldsymbol{\beta}$ which may then be exponentiated to determine how the expected response changes with the predictor. In order to estimate $\boldsymbol{\beta}$ using maximum likelihood estimation, first, the log-likelihood must be rewritten in terms of $\boldsymbol{\beta}$ using the link function,

$$l(\lambda; \boldsymbol{y}) = \sum_{i=1}^{n} (y_i(\log t_i + \mathbf{x}_i^\top \boldsymbol{\beta}) - \exp(\log t_i + \mathbf{x}_i^\top \boldsymbol{\beta}) - \log(y_i!)). \tag{1.11}$$

While the Poisson distribution is standard when analyzing a count outcome, it is limited to an equidispersed setting. When there is overdispersion, the negative binomial distribution should be considered.

1.2.2 Negative Binomial Regression

When the data are inherently overdispersed then the negative binomial distribution is more relevant. Overdispersion occurs when the response variance is greater than the mean[18]. There are a number of causes of overdispersion. It often appears when observations are based on time intervals of varying lengths or when data are clustered[32]. The negative binomial distribution allows for a count or rate outcome to be analyzed without the assumption that the mean is equal to the variance. An extra parameter, commonly referred to as the heterogeneity parameter α, is added to the variance function. The constraint on α is that it takes on a positive rational value.

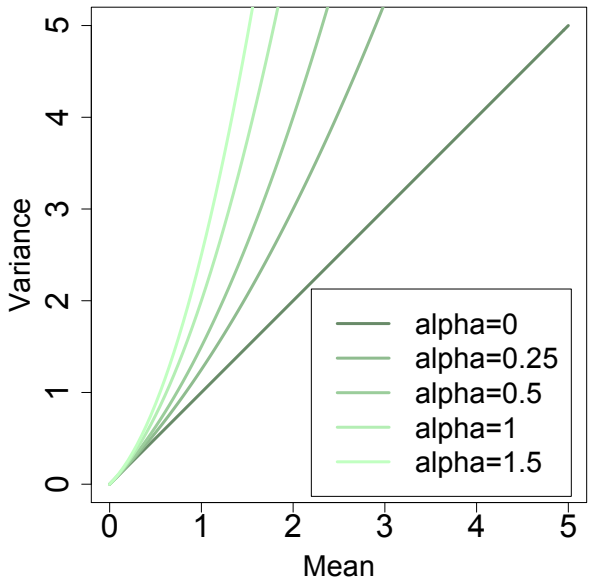
Typically, α is not greater than four[18]. The heterogeneity parameter is inversely related to the dispersion parameter often referred to as ϕ. The negative binomial distribution assumes mean and variance to be given by,

$$\mathbf{E}(y_i) = \mu_i \tag{1.12}$$

$$\text{Var}(y_i) = \mu_i + \alpha\mu_i^2. \tag{1.13}$$

Except when $\alpha = 0$, the variance is larger than the mean in the negative binomial model. Figure 4 demonstrates how the variance and mean change for different values of α.

Fig. 4: Plot of variance by mean for Poisson ($\alpha = 0$) and negative binomial models with different values of the heterogeneity parameter.



When $\alpha = 0$ in a negative binomial model, the mean is equal to the variance which yields the Poisson distribution. Similarly, as α approaches zero, the negative binomial distribution converges to the Poisson distribution[1]. Therefore, the Poisson model is

nested within the negative binomial model given the same set of predictors. The extra parameter, α, accounts for any inherent overdispersion that might exist in count data.

Negative binomial regression methods have been developed in the traditional statistical setting when $n > p$. The negative binomial distribution (NB2 model) is commonly derived as a Poisson-gamma mixture model. The negative binomial PDF is,

$$f(y;\mu,\alpha) = \binom{y_i + \frac{1}{\alpha} - 1}{\frac{1}{\alpha} - 1} \left(\frac{1}{1+\alpha\mu_i}\right)^{\frac{1}{\alpha}} \left(\frac{\alpha\mu_i}{1+\alpha\mu_i}\right)^{y_i} \tag{1.14}$$

where α, the heterogeneity parameter, must be a positive rational value. The estimation of α will be discussed in Section 1.2.3. The likelihood associated with the negative binomial PDF is,

$$\begin{aligned} L(\mu;y,\alpha) = \prod_{i=1}^{n} \exp\Bigg\{ y_i \log\left(\frac{\alpha\mu_i}{1+\alpha\mu_i}\right) - \frac{1}{\alpha}\log\left(1+\alpha\mu_i\right) + \\ \log\Gamma\left(y_i + \frac{1}{\alpha}\right) - \log\Gamma\left(y_i+1\right) - \log\Gamma\left(\frac{1}{\alpha}\right)\Bigg\}. \end{aligned} \tag{1.15}$$

It is more straightforward and equivalent to maximize the corresponding log-likelihood given by,

$$\begin{aligned} l(\mu;y,\alpha) = \sum_{i=1}^{n} y_i \log\left(\frac{\alpha\mu_i}{1+\alpha\mu_i}\right) - \frac{1}{\alpha}\log\left(1+\alpha\mu_i\right) + \\ \log\Gamma\left(y_i + \frac{1}{\alpha}\right) - \log\Gamma\left(y_i+1\right) - \log\Gamma\left(\frac{1}{\alpha}\right). \end{aligned} \tag{1.16}$$

In negative binomial regression the model assumes that the mean can be modeled as a linear combination of the predictors. The log link function is,

$$\log(\mu_i) = \mathbf{x}_i^\top \boldsymbol{\beta} \tag{1.17}$$

where $\mathbf{x}_i$ are the predictor variables and $\boldsymbol{\beta}$ is a vector of their coefficients. The log link function may be rewritten in terms of μ showing how the expected response changes with the predictors. This link function will allow the negative binomial log-likelihood to be rewritten in terms of the predictors

$$\mu_i = \exp(\mathbf{x}_i^\top \boldsymbol{\beta}). \tag{1.18}$$

In order to use maximum likelihood estimation (MLE) the negative binomial log-likelihood must be parametrized in terms of the model coefficients, $\boldsymbol{\beta}$, which can be done using the link function in equation 1.17,

$$\begin{aligned} l(\boldsymbol{\beta}_j; y, \alpha) \;=\; \sum_{i=1}^{n} \; y_i \; \log \left(\frac{\alpha \; \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}{1 + \alpha \; \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)} \right) - \\ \frac{1}{\alpha} \; \log \; \left(1 \; + \; \alpha \; \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right) \right) + \\ \log \; \Gamma \left(y_i \; + \; \frac{1}{\alpha} \right) - \\ \log \; \Gamma \left(y_i + 1 \right) \; - \; \log \; \Gamma \left(\frac{1}{\alpha} \right). \end{aligned} \tag{1.19}$$

As was shown in Section 1.2.1., with Poisson regression, a similar derivation may be shown when the outcome is a rate and an offset term, t_i, is incorporated in the negative binomial regression model. The mean and variance are re-expressed as,

$$\mathbf{E}(y_i) = \mu_i t_i \tag{1.20}$$

$$\mathrm{Var}(y_i) = \mu_i t_i + \alpha (\mu_i t_i)^2. \tag{1.21}$$

The negative binomial PDF with the offset term may be rewritten as,

$$f(y; \mu, \alpha) = \binom{y_i + \frac{1}{\alpha} - 1}{\frac{1}{\alpha} - 1} \left(\frac{1}{1 + \alpha \mu_i t_i} \right)^{\frac{1}{\alpha}} \left(\frac{\alpha \mu_i t_i}{1 + \alpha \mu_i t_i} \right)^{y_i}. \tag{1.22}$$

The associated likelihood is,

$$L(\mu; y, \alpha) = \prod_{i=1}^{n} \exp\left\{ y_i \log\left(\frac{\alpha\mu_i t_i}{1+\alpha\mu_i t_i}\right) - \frac{1}{\alpha}\log\left(1+\alpha\mu_i t_i\right) + \log\Gamma\left(y_i + \frac{1}{\alpha}\right) - \log\Gamma\left(y_i+1\right) - \log\Gamma\left(\frac{1}{\alpha}\right)\right\}. \tag{1.23}$$

The corresponding log-likelihood which is more straightforward to maximize is

$$l(\mu; y, \alpha) = \sum_{i=1}^{n} y_i \log\left(\frac{\alpha\mu_i t_i}{1+\alpha\mu_i t_i}\right) - \frac{1}{\alpha}\log\left(1+\alpha\mu_i t_i\right) + \log\Gamma\left(y_i + \frac{1}{\alpha}\right) - \log\Gamma\left(y_i+1\right) - \log\Gamma\left(\frac{1}{\alpha}\right). \tag{1.24}$$

Again, the negative binomial regression model assumes that the mean can be modeled as linear combination of the predictors. The log link function including the offset term is,

$$\log(\mu_i t_i) = \mathbf{x}_i^\top \boldsymbol{\beta} \tag{1.25}$$

where $\mathbf{x}_i$ are the predictor variables and $\boldsymbol{\beta}$ is a vector of their coefficients. The log link function may be rewritten in terms of μ and t_i showing how the response changes with the predictors. This link function will allow the negative binomial log-likelihood to be rewritten in terms of the predictors

$$\mu_i t_i = \exp(\mathbf{x}_i^\top \boldsymbol{\beta}). \tag{1.26}$$

In order to use MLE the negative binomial log-likelihood must be parametrized in terms of the model coefficients, $\boldsymbol{\beta}$ which can be done using the link function in

equation 1.25,

$$
\begin{aligned}
l(\boldsymbol{\beta}_j; y, \alpha) \;=\; \sum_{i=1}^{n} \; y_i \; \log\left(\frac{\alpha \; \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}{1 + \alpha \; \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}\right) - \\
\frac{1}{\alpha} \; \log \; \left(1 \; + \; \alpha \; \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)\right) \; + \\
\log \; \Gamma \left(y_i \; + \; \frac{1}{\alpha}\right) \; - \\
\log \; \Gamma \left(y_i + 1\right) \; - \; \log \; \Gamma \left(\frac{1}{\alpha}\right).
\end{aligned}
\tag{1.27}
$$

While the negative binomial model is similar to the Poisson model, there is an additional parameter, α. This parameter allows for the negative binomial model to account for overdispersion. In Section 1.2.3 the estimation methods for α will be described.

1.2.3 Hilbe's Method of alpha estimation

The two common methods for estimating the parameter, α, previously described in the negative binomial distribution are MLE and Hilbe's method, a method of moments based estimator. MLE works by finding the estimate that maximizes the likelihood given in Equation 1.22. Equivalently and mathematically more simply, we may find the MLE of the parameter by finding the estimate that maximizes the log-likelihood given in equation 1.23. Hilbe's method for estimating α is to iteratively adjust the value of α so that the deviance-based dispersion approximates one[18]. Hilbe's algorithm is as follows:

1. Estimate μ as the mean of the response.
2. Calculate the chi-square test statistic as $\chi^2 = \sum(y_i - \mu)^2/\mu$.
3. Calculate the degrees of freedom (df) as the number of subjects minus the number of parameters (excluding α) included in the model.

4. The deviance-based dispersion is calculated as $\phi = \chi^2/\text{df}$.

5. Calculate α, the dispersion statistic as $1/\phi$.

6. Set $\phi_{old} = \phi$.

7. Re-estimate μ using the negative binomial model and the estimate of α.

8. Update χ^2 from the negative binomial model as $\sum((y_i - \mu)^2/(\mu + (\alpha * (\mu^2)))$.

9. Re-calculate $\phi = \chi^2/\text{df}$.

10. Re-calculate $\alpha = \phi * \alpha$.

11. Repeat steps 6 to 10 until $|\phi_{old} - \phi|$ is less than some prespecified small tolerance.

Hilbe's function was coded in R and validated by comparing $\hat{\alpha}$ from the R function to $\hat{\alpha}$ from the *theta.mm* function that was passed a *glm.nb* object. The simulation studies were performed at varying levels of α. Secondly, we conducted simulation studies to determine whether Hilbe's method or MLE was more precise at estimating α. Four sets of simulation studies were performed at α levels of 0.1, 0.2, 0.5, and 0.9. For each α level, we simulated 100 independent negative binomial data sets and estimated α using MLE from the `glm.nb` function in R and using Hilbe's function which was coded into R. The simulation studies were as follows:

1. Randomly generate $i = 1, ..., 100$ observations with $P = 500$ variables, $x_{i1}, x_{i2}, ..., x_{iP}$ following a standard normal distribution.

2. Select a subset, P_1, of length 5 of the P variables to be associated with the response. Set the parameter values to $\boldsymbol{\beta} = (0.5, 0.5, -0.5, -0.5, -0.5)$ for these P_1 variables. Also assign α to either $0.1, 0.2, 0.5$, or 0.9. Finally, assign the intercept value, $\gamma_0 = 0.5$.

3. Generate the μ values for the negative binomial distribution using,

$$\mu_i = \exp(\gamma_0 + \sum_{k=1}^{P_1} \beta_k x_{ik}).$$

4. Randomly generate the response, $Y_i \sim$ Negative Binomial(μ_i, α).

5. Estimate α using maximum likelihood estimation in the `glm` package and Hilbe's method using our validated R function.

6. Repeat steps 1-5 times to yield 100 different MLE and Hilbe estimates of α.

As α gets close to 0, neither maximum likelihood estimation nor Hilbe's method can accurately estimate α. When $\alpha = 0.1$, both Hilbe and MLE methods provide estimates that are undefined, which result when trying to divide by zero. Therefore, in situations where alpha is 0.1 or less the Poisson distribution may be more appropriate.

Reported from the simulation studies are histograms (Figures 5, 6, and 7) of the $\hat{\alpha}$ using both Hilbe's method and maximum likelihood estimation when the true α is $0.2, 0.5$, and 0.9. From examination of the figures, we concluded that Hilbe's method outperforms maximum likelihood estimation of α.

Fig. 5: Histograms of $\hat{\alpha}$ from 100 simulations for Hilbe's Method (left) and MLE (right) when the true α is 0.2.

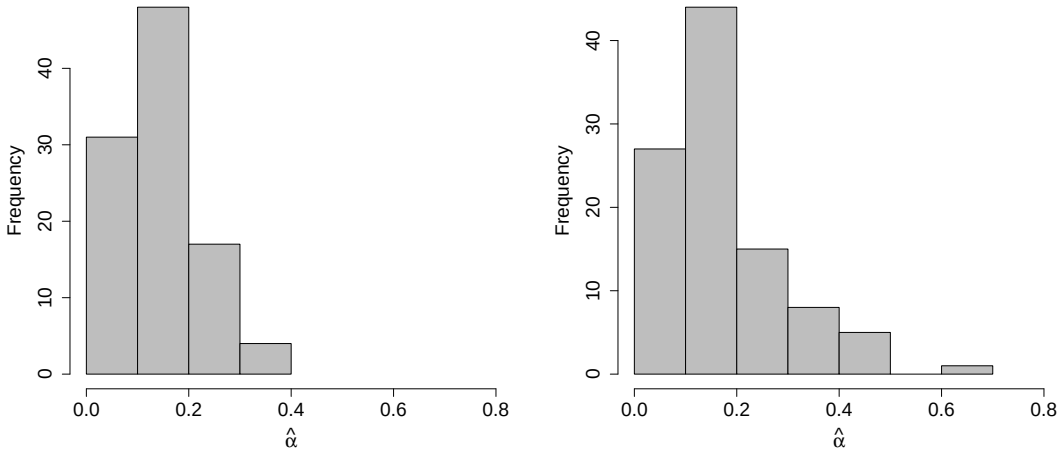


Fig. 6: Histograms of $\hat{\alpha}$ from 100 simulations for Hilbe's Method (left) and MLE (right) when the true α is 0.5.

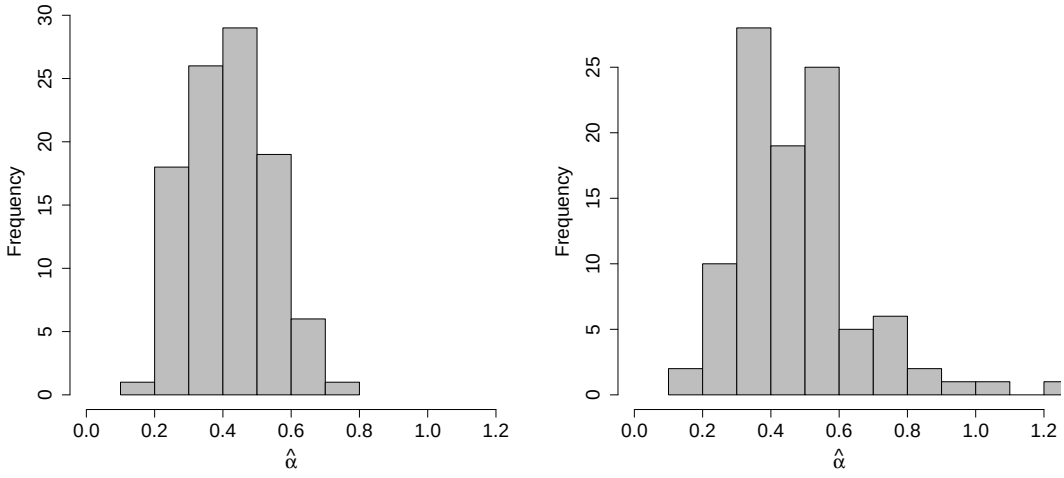
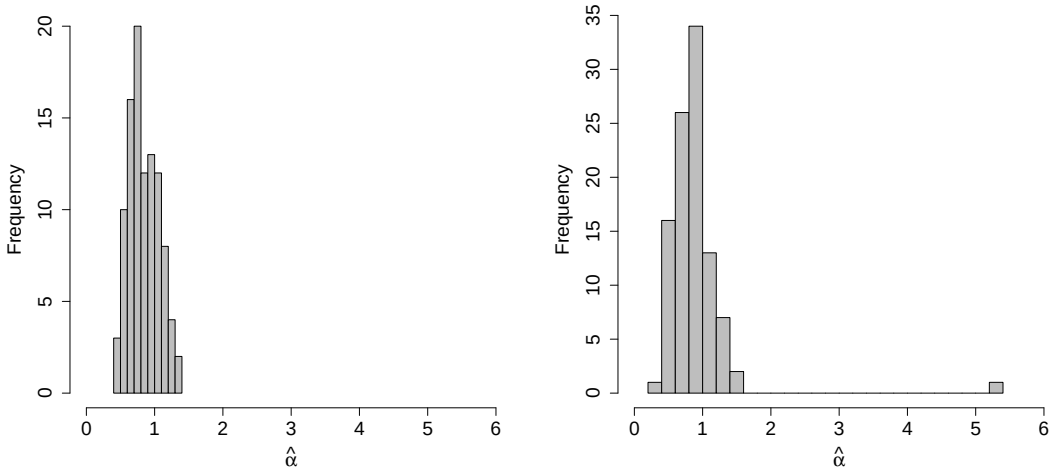


Fig. 7: Histograms of $\hat{\alpha}$ from 100 simulations for Hilbe's Method (left) and MLE (right) when the true α is 0.9.



1.2.4 Extension to High-dimensional Count Methods

Extensive work has been done with the Poisson and negative binomial distribution in the traditional statistical setting. However, there are limited methods for analyzing a count outcome with a high-dimensional predictor space. Development of high-dimensional methods have been restricted to the Poisson distribution[25,29,30,31,35]. Herein we developed three new comprehensive statistical methods for analyzing count data. First, we developed a method that could be used to analyze an overdispersed count outcome when there is a high-dimensional predictor space. Our negative binomial generalized monotone incremental forward stagewise method is described in Chapter 2. Second, we developed a method that could be used to analyze a longitudinal count outcome when there is a high-dimensional predictor space. Our longitudinal Poisson generalized monotone incremental forward stagewise method is described in Chapter 3. Lastly, we developed a method that can be used to analyze an overdis-

persed longitudinal count outcome when there is a high-dimensional predictor space. Our longitudinal negative binomial generalized monotone incremental forward stagewise method is described in Chapter 4.

1.2.5 Discussion

The following chapters will present the three extensions that were made to the generalized monotone incremental forward stagewise method for count data outcomes. For each method we will perform simulation studies to demonstrate how well the new method performs against current methods. It will also be determined when a negative binomial model, which accounts for overdispersion, is superior to a Poisson model. Simulation studies will be performed both with and without an offset, when the outcome of interest is a rate. Each method will be used to analyze a high-dimensional dataset where either MN or NBuds are the outcome of interest. Conclusions will be made about the performance of each method in the simulation studies and when each is the most applicable. Results from application to a real data set will be displayed for each new method.

CHAPTER 2

THE GENERALIZED MONOTONE INCREMENTAL FORWARD STAGEWISE METHOD FOR THE NEGATIVE BINOMIAL DISTRIBUTION

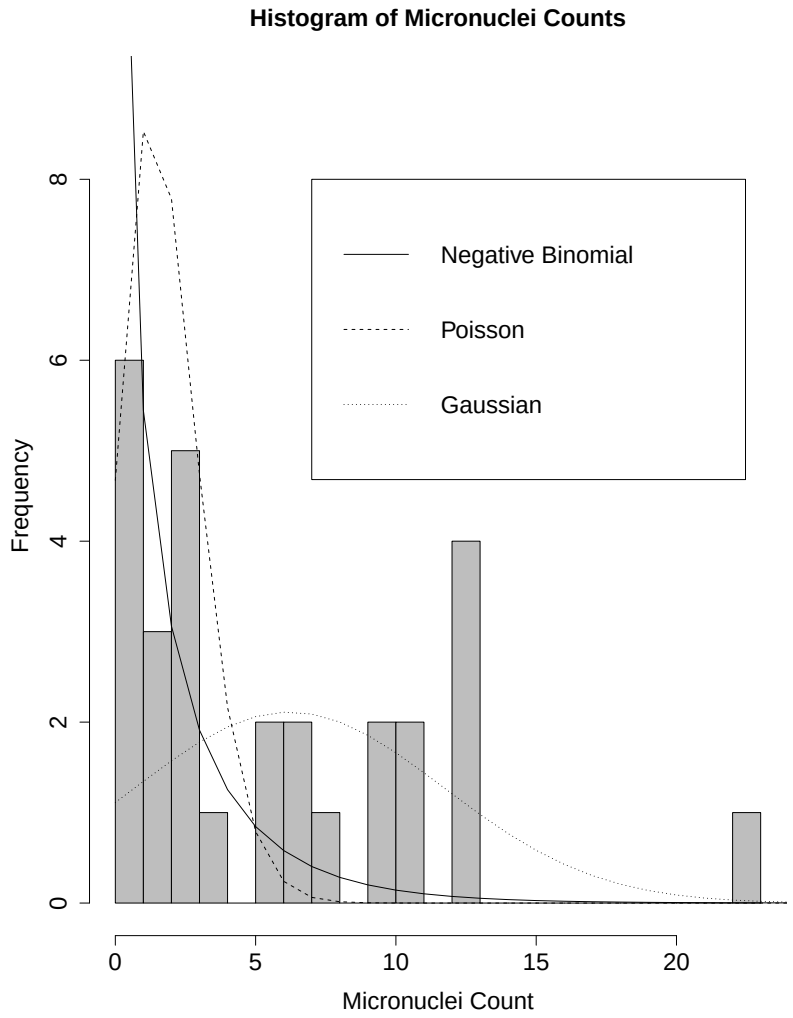
2.1 Negative Binomial Norwegian Data

In the 1990s the Norwegian Mother and Child Cohort Study (MoBa) was designed collaboratively by researchers at the Medical Birth Registry of Norway (MBRN) and by researchers at the National Institute of Public Health[24]. Pregnant women who attended routine ultrasounds in Norway were recruited from 1999 to 2005 from 52 hospitals and maternity units. There was no exclusion criteria, and women who were pregnant more than once in the time period could participate multiple times. The pregnancy was defined as the unit of observation of the study. A total of 150,309 pregnant women were represented in the study with a total of $n = 129,953$ different mothers. Of the invited pregnant mothers, 64,136 decided to participate with, $n = 58,515$ unique mothers. There were 53,060 women who had one pregnancy, 5,290 with two pregnancies, 164 with three pregnancies, and 1 with four pregnancies. Demographic data and other information was collected on all patients through questionnaires, the MBRN, a cancer registry, a prescription database, a cause of death registry, and a vaccination registry[24]. The purpose of the study was to examine the association between exposures, genetic factors, and diseases[24]. From the results there was hope to develop preventions for diseases.

Umbilical cord blood samples were collected immediately after birth in a subset of the babies (n=200). After quality control and other exclusions, 111 samples were

hybridized to Agilent 4x44k human oligonucleotide microarrays to measure gene expression. Sample processing, image analysis, normalization, background correction, and filtering for the gene expression data are described in Hochstenbach et al.[19]. For an even smaller subset (n=29) MN and NBud data were collected and scored using the procedure described by Decordier et al., 2009[7]. Data were downloaded from Gene Expression Omnibus (GSE31836). Before analysis, a Boundary Likelihood Ratio test was performed to determine whether a Poisson or negative binomial model would be more appropriate given the MoBa data[18]. The alternative hypothesis of $\alpha \neq 0$ was tested against a null hypothesis of $\alpha = 0$. The chi-square test results were $\chi_1^2 = 59.8$ with a p-value of $1.04x10^{-14}$. Therefore, we reject the null hypothesis that $\alpha = 0$ implying a negative binomial model is more appropriate given the data. To further support the negative binomial model, a histogram of the micronuclei data with Gaussian, Poisson, and negative binomial overlays is given in Figure 8.

Fig. 8: Histogram of MoBa MN with a Gaussian, Poisson, and negative binomial fit overlays.



From the histogram we can see that the Gaussian model is a poor fit and the negative binomial model is better than the Poisson at estimating the high zero counts. Therefore, this motivates the development of our negative binomial GMIFS model which we expect to be superior to the Poisson GMIFS model and Poisson `glmpath` for the MoBa data analysis.

2.2 Statistical Methods

2.2.1 Current Methods for Analyzing a Count Outcome in a High-dimensional Setting

Few methods have been developed for analyzing a count or rate outcome when there is a high-dimensional predictor space. The methods that have been developed are limited to the Poisson distribution and are in the class of penalized regression models.

Penalized models use a pre-determined penalty function to control the regression coefficients, fit a more appropriate and interpretable model to prevent $p > n$, and to prevent overfitting. The `glmpath` method was developed to be a smoother, less greedy version of forward stepwise selection[29]. It uses a linear combination of the L_1 and L_2 norm penalizations[28,29]. The method developed by Park et al., 2006 is a path-following algorithm that is based on a previous algorithm, least absolute shrinkage and selection operator (LASSO)[28,29]. LASSO is a variable selection and shrinkage method that adds a constraint to the sum of squares[33]. In the linear regression setting, the LASSO is based on minimizing the sum of squares term with the added constraint,

$$\sum_{i=1}^{N}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{\mathbf{j=1}}^{\mathbf{p}} |\beta_{\mathbf{j}}| \tag{2.1}$$

where x_{ij} are the standardized predictors and y_i is the set of centered responses for $i = 1, ..., N$ and $j = 1, ..., p$. The modified version of the LASSO that is used for the `glmpath` algorithm begins with the generalized linear model formula,

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} L(\mathbf{y}; \boldsymbol{\beta}) \tag{2.2}$$

where L represents the likelihood or log-likelihood function. When the number of predictors p exceeds the number of observations n, a penalization may be imposed for an automatic variable selection effect[29]. In the `glmpath` algorithm, a penalization comparable to the LASSO is added to the squared error loss with a regularization,

$$\hat{\boldsymbol{\beta}}(\lambda) = \arg\min_{\boldsymbol{\beta}}(-\log L(\mathbf{y};\boldsymbol{\beta}) + \lambda||\boldsymbol{\beta}||_{\mathbf{1}}) \tag{2.3}$$

where $\lambda > 0$ is the regularization parameter. The initial value of λ is set to ∞. The algorithm computes a series of solution sets with each estimating the coefficients with a smaller λ based on the previous estimate. The three steps of the optimization are: determine the step size in λ, predict the corresponding change in the coefficients, and correct the error in the previous prediction[29].

The coefficient estimates become exceedingly unstable when some of the predictors are correlated[29]. Therefore, a quadratic penalty term is added to control the stability of the fit[17],

$$\boldsymbol{\beta}(\hat{\lambda_1}) = \arg\min_{\boldsymbol{\beta}}(-\log L(\mathbf{y};\boldsymbol{\beta} + \lambda_{\mathbf{1}}||\boldsymbol{\beta}||_{\mathbf{1}} + \frac{\lambda_{\mathbf{2}}}{\mathbf{2}}||\boldsymbol{\beta}||_{\mathbf{2}}^{\mathbf{2}})) \tag{2.4}$$

where $\lambda_1 \in (0, \infty)$ and λ_2 is a fixed small positive constant.

The `glmpath` algorithm has been developed for the following distributions: binomial with a logit link, Poisson with a log link, and Gaussian with an identity link[28]. The algorithm does not accommodate the negative binomial distribution.

Second, `glmnet` will fit a generalized linear model via penalized maximum likelihood[13,14]. The regularization path is computed for the LASSO or elastic net penalty at a grid of values for the regularization parameter[13,14]. The cyclical coordinate descent method is repeated until cycles converge[14]. The cyclical coordinate descent method optimizes the objective function over each parameter while the others are

fixed[14]. The elastic net solves the following problem:

$$\min_{(\beta_0,\beta)\in R^{P+1}}\left[\frac{1}{2N}\sum_{i=1}^{N}(y_i-\beta_0-x_i^\top\beta)^2+\lambda P_\alpha(\beta)\right] \tag{2.5}$$

where

$$\begin{aligned} P_\alpha(\beta) &= (1-\alpha)\frac{1}{2}||\beta||_{l_2}^2+\alpha||\beta||_{l_1} \\ &= \sum_{j=1}^{p}[\frac{1}{2}(1-\alpha)\beta_j^2+\alpha|\beta_j|], \end{aligned} \tag{2.6}$$

N are the number of observations, and x_{ij} are the standardized predictors. P_α is also a compromise between the ridge-regression penalty (L_2 norm) and the LASSO penalty (L_1 norm) so like `glmpath`, `glmnet` reaps the benefits of both methods[13,14]. Ridge regression works to shrink the coefficients of correlated predictors towards each other, therefore allowing them to borrow strength from each other. The LASSO is indifferent towards very correlated predictors and typically picks one and ignores the remaining. The `glmnet` algorithm has been developed for the following situations: linear, logistic, and multinomial, Poisson, and Cox regression models[13]. The `glmnet` package is allegedly highly efficient when examining data where $N << p$.

Last, Makowski and Archer, 2015 recently established the Generalized Monotone Incremental Forward Stagewise (GMIFS) Method for the Poisson regression setting[25]. This was an extension of the GMIFS method that was originally developed by Hastie et al. for the logistic regression model[17]. The Poisson GMIFS method enables modeling a count outcome in a high-dimensional setting or when $n < p$. Recall the log-likelihood for the Poisson distribution defined originally in Section 1.2.1

$$l(\lambda;\boldsymbol{y})=\sum_{i=1}^{n}(y_i(\log t_i+\mathbf{x}_i^\top\boldsymbol{\beta})-\exp(\log t_i+\mathbf{x}_i^\top\boldsymbol{\beta})-\log(y_i!)). \tag{2.7}$$

Often when handling a high-dimensional predictor space, it is of interest to par-

tition it into penalized and unpenalized spaces. Unpenalized predictors are those that will be coerced or forced into the model whereas penalized predictors are those that will be selected by the model via automatic variable selection. The penalized predictors, $\boldsymbol{X}$, are those from a high-throughput genomic experiment and the unpenalized predictors, $\boldsymbol{W}$, are variables that previous research has shown should be forced in the model. The coefficients of the penalized predictors will be defined as $\boldsymbol{\beta}$, and the coefficients of the unpenalized predictors will be defined at $\boldsymbol{\gamma}$. The log-likelihood may be rewritten as,

$$l(\lambda; \boldsymbol{y}) = \sum_{i=1}^{n} (y_i(\log t_i + \mathbf{x}_i^\top \boldsymbol{\beta} + \mathbf{w}_i^\top \boldsymbol{\gamma}) - \exp(\log t_i + \mathbf{x}_i^\top \boldsymbol{\beta} + \mathbf{w}_i^\top \boldsymbol{\gamma}) - \log(y_i!)). \quad (2.8)$$

To simplify calculations in the GMIFS procedure the expanded covariate space as previously described in Hastie et al. was used[17]. By expanding the covariate space, there is no need to take a second derivative to determine the direction of the increment. The Poisson GMIFS method developed by Makowski and Archer, 2015 is as follows[25],

1. Set step $s = 0$ and initialize the components of $\hat{\boldsymbol{\beta}}^s = 0$.

2. Initialize the intercept, γ_0, and the unpenalized coefficients, γ_j, where $j = 1, ..., J$ using the maximization algorithm of the log-likelihood.

3. Treating $\boldsymbol{\gamma}$ and γ_0 as fixed, find the predictor x_m such that $m = \arg\min_k(-\frac{dl}{d\beta_k})$ at the current estimate $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^s$.

4. Update $\hat{\beta}_m^{s+1} = \hat{\beta}_m^s + \epsilon$ to yield a new vector of parameter estimates.

5. Using the new $\boldsymbol{\beta}$ vector from step 4, update $\boldsymbol{\gamma}$ and γ_0 via the maximization algorithm of the log-likelihood. Step is updated to $s = s + 1$.

6. Repeat steps 3-5 until the difference between successive log-likelihoods is less than a pre-specified small tolerance, τ or until $p \geq n$.

The final model may be selected base on Akaike information criterion (AIC) or Bayesian information criterion (BIC).

All currently available methods for analyzing a count or rate outcome when there is a high-dimensional predictor space only consider the Poisson distribution. Therefore, these methods are only appropriate for analyzing equidispersed data. No methods exist for analyzing overdispersed count outcomes when there is a high-dimensional predictor space. An extension of the GMIFS method will be developed for the negative binomial distribution so that a more appropriate analysis may be performed when there is implicit overdispersion in the data.

2.2.2 Extension of the Generalized Monotone Incremental Stagewise Method to the Negative Binomial Distribution

The previously developed Generalized Monotone Incremental Forward Stagewise (GMIFS) method for the Poisson regression setting was described in Section 2.2.1. While the GMIFS method for Poisson regression is applicable for many count datasets, it is limited to handling equidispersed data. There are no statistical methods that can handle overdispersed count outcomes when there is a high dimensional predictor space or when $n < p$. Therefore, we extended the GMIFS method for the negative binomial distribution to handle these situations. Following the GMIFS method developed for the Poisson case, we extended the GMIFS method to the negative binomial case.

Recall that the negative binomial log-likehood (Equation 1.26) may be defined as,

$$
\begin{aligned}
l(\boldsymbol{\beta}_j; y, \alpha) \;=\; \sum_{i=1}^{n} \; y_i \; \log\left(\frac{\alpha \; \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}{1 + \alpha \; \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}\right) - \\
\frac{1}{\alpha} \; \log \; \left(1 \; + \; \alpha \; \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)\right) \; + \\
\log \; \Gamma \left(y_i \; + \; \frac{1}{\alpha}\right) \; - \\
\log \; \Gamma \left(y_i + 1\right) \; - \; \log \; \Gamma \left(\frac{1}{\alpha}\right)
\end{aligned}
\tag{2.9}
$$

where

α: the heterogeneity parameter,

y_i: the count outcome ranging from $i = 1, ..., n$,

$\mathbf{x}_i^\top$: the vector of predictor variables,

$\boldsymbol{\beta}$: the vector of coefficients corresponding to the predictor variables.

By expanding this to the negative binomial setting we added one extra parameter, α, the heterogeneity parameter, which will be estimated iteratively using Hilbe's method previously described in Section 1.2.3[18]. Recall, Hilbe's method for estimating α is to iteratively adjust the value of α so that the deviance-based dispersion approximates one[18].

For the GMIFS method, the predictor space is separated into two components: penalized and unpenalized predictors. First, unpenalized predictors are those that will be forced into the model. Second, penalized predictors are those that will be selected by the negative binomial model via automatic variable selection in the GMIFS procedure. Because we have both penalized and unpenalized predictors, we separate the notation for our parameters and use $\boldsymbol{\beta}$ to represent the parameters that correspond to the penalized predictors ($\boldsymbol{X}$), $\boldsymbol{\gamma}$ to represent the parameters that correspond to the unpenalized predictors ($\boldsymbol{W}$), and γ_0 to represent the intercept. Unpenalized predictors are those which are forced into the model due to already known significance

or prior knowledge. Previous research summarized in Section 1.1.3. has shown that age, gender, and smoking status should be included when modeling MN frequency[6,11]. Penalized predictors are the variables that our model will select. Frequently this will be the data from a high-throughput genomic experiment or some high-dimensional dataset. For the purpose of our research and the MoBa study, the penalized predictors are the gene expression data.

In the GMIFS algorithm, we first set the $\boldsymbol{\beta}$'s, the coefficients of the penalized predictors, to 0 and initialize α using method of moments and estimate $\boldsymbol{\gamma}$ and γ_0 using maximum likelihood estimation. This is fitting a model with no penalized predictors present. The algorithm then iteratively updates the penalized coefficients one at a time by a small value, ϵ, as that having the largest negative gradient followed by re-estimating α, $\boldsymbol{\gamma}$ and γ_0 each time. To determine which coefficient to update, the derivative of the log-likelihood with respect to $\boldsymbol{\beta}$ must be obtained, which is

$$\frac{dl}{d\boldsymbol{\beta}} = \sum_{i=1}^{n} \frac{x_i(y_i - x_i^\top \boldsymbol{\beta})}{1 + \alpha(x_i^\top \boldsymbol{\beta})}. \tag{2.10}$$

The problem arises when we have to determine the direction in which to update the penalized coefficient. To avoid taking the second derivative, Hastie et al. showed that you can expand the covariate space as $[\boldsymbol{X} : -\boldsymbol{X}]$[17]. At each step, only one coefficient in either the positive or negative side of the covariate space is incremented by ϵ. Once the GMIFS algorithm has been applied to the expanded covariate space, the coefficients corresponding to the negative covariate space are subtracted from the coefficients corresponding to the positive covariate space to return to the parameter estimates on the original x scale. Because we are only estimating the coefficients with respect to the penalized covariates, our expanded covariate space is $\boldsymbol{X}^{NEW} = [\boldsymbol{X} : -\boldsymbol{X}]$. The full GMIFS algorithm for the negative binomial model is:

1. Set step $s = 0$ and initialize the components of $\hat{\boldsymbol{\beta}}^s = 0$, initialize $\hat{\alpha}$ using Hilbe's method of moments, and initialize the intercept, $\hat{\gamma_0}$, and the unpenalized coefficients, $\hat{\gamma_j}$, where $j = 1, ..., J$ using the maximization algorithm of the log-likelihood.

2. Treating $\hat{\alpha}$, $\hat{\boldsymbol{\gamma}}$ and $\hat{\gamma_0}$ as fixed, find the predictor x_m such that $m = \arg\min_k(-\frac{dl}{d\beta_k})$.

3. Update $\hat{\beta}_m^{s+1} = \hat{\beta}_m^s + \epsilon$.

4. Using the new $\hat{\boldsymbol{\beta}}$ vector from step 3, update $\hat{\alpha}$ via Hilbe's algorithm and update $\hat{\boldsymbol{\gamma}}$ and $\hat{\gamma_0}$ via the maximization algorithm of the log-likelihood. Step is updated to $s = s + 1$.

5. Repeat steps 2-4 until the difference between successive log-likelihoods is less than a pre-specified small tolerance, τ or until $p \geq n$.

In the implementation of this algorithm, we use $\epsilon = 0.001$ and $\tau = 0.001$. The final model will be selected based on model fitting criteria such as AIC or BIC.

Further, recall that an offset term is often used when the response is a rate as opposed to a count outcome. The GMIFS algorithm accommodates the rate outcome through the link function described in Equation 1.24. The MoBa data does have a consistent number of binucleated cells scored by subject for the MN and NBuds. The recommended number of binucleated cells to be scored is 2,000[6]. However, in other studies a range of cells are scored, typically up to 2,000. The above GMIFS method incorporates an offset term that will account for varying total number of binucleated cells scored by subject.

2.3 Simulation Studies

Simulation studies were performed before application to the MoBa data. Data were simulated from the negative binomial distribution as follows:

1. Randomly generate the predictor set with P different variables, $x_{i1}, x_{i2}, ..., x_{iP}$ where $i = 1, .., n$ from the standard normal distribution.

2. Select a subset, P_1 of length 5 of the P variables to be associated with the response.

3. Coefficient, $\boldsymbol{\beta}$, values were assigned to the P_1 variables to be associated with the response. $\beta_1 = \beta_2 = 0.5$ and $\beta_3 = \beta_4 = \beta_5 = -0.5$. Also assign α, the heterogeneity parameter and the intercept value, $\gamma_0 = 0.5$. For simulations where an offset is used, the offset was generated from a uniform distribution on the interval 1,800 to 2,200.

4. Generate the μ values for the negative binomial distribution using,

$$\mu_i = \exp(\gamma_0 + \sum_{k=1}^{P_1} \beta_k x_{ik}).$$

5. Randomly generate the response, $Y_i \sim$ Negative Binomial(μ_i, α).

6. Fit a Poisson GMIFS model, negative binomial GMIFS model, and Poisson `glmpath` model.

7. Repeat this to simulate r independent data sets.

$r = 100$ negative binomial data sets were simulated. Simulations were performed with and without an offset, for $\alpha = 0.1, 0.5$, and 0.9, and for $n = 100$. Models with and without and offset were examined because often count data must be examined

as a rate. For example, when there are varying numbers of binucleated cells scored per patient it is crucial to incorporate the offset into the model and only analyze the outcome as a rate as opposed to a count. It is of interest to examine varying levels of α as it would be expected for α close to 0, the Poisson model should be similar to the negative binomial model. The number of predictors was set to 500 and the number of predictors associated with the outcome was 5, each being of equal magnitude but with some in the positive direction and some in the negative direction. For all data sets, we fit a Poisson and negative binomial GMIFS model. The `glmpath` model was not fit due to convergence problems that could not be solved. The methods were compared using the following outcomes:

- The number of true predictors that have a non-zero coefficient;
- The number of false predictors that have a non-zero coefficient.

The results for the simulation studies when $\alpha = 0.1, 0.5$, and 0.9 with no offset appear in Tables 1, 2, and 3. The BIC selected models are more parsimonious than the AIC selected models. Overall, the negative binomial models are more parsimonious than the Poisson models. The negative binomial and Poisson model have comparable sensitivity; however, the negative binomial model has more specificity for eliminating false predictors, particularly for BIC selected models.

Table 1.: Results from Simulation Studies when the true α is 0.1: Mean/Median number of true predictors that had a nonzero coefficient estimate in the final model (True Nonzero) and the mean/median number of false predictors that had a nonzero coefficient estimate in the final model (False Nonzero). Oracle number of true nonzero coefficients, $P_1 = 5$. Oracle number of zero coefficients, $P - P_1 = 495$.

	Negative Binomial GMIFS BIC selected model	Poisson GMIFS BIC selected model	Negative Binomial GMIFS AIC selected model	Poisson GMIFS AIC selected model
True Nonzero				
Mean (Standard Deviation)	4.5 (1.09)	4.9 (0.41)	5.0 (0.1)	5.0 (0.1)
True Nonzero				
Median (Range)	5.0 (0.0, 5.0)	5.0 (3.0, 5.0)	5.0 (4.0, 5.0)	5.0 (4.0, 5.0)
False Nonzero				
Mean (Standard Deviation)	5.1 (4.06)	14.1 (5.26)	30.3 (9.22)	30.2 (8.19)
False Nonzero				
Median (Range)	4.0 (0.0, 17.0)	14.0 (3.0, 28.0)	30.0 (6.0, 63.0)	29.0 (13.0, 53.0)

Table 2.: Results from Simulation Studies when the true α is 0.5: Mean/Median number of true predictors that had a nonzero coefficient estimate in the final model (True Nonzero) and the mean/median number of false predictors that had a nonzero coefficient estimate in the final model (False Nonzero). Oracle number of true nonzero coefficients, $P_1 = 5$. Oracle number of zero coefficients, $P - P_1 = 495$.

	Negative Binomial GMIFS BIC selected model	Poisson GMIFS BIC selected model	Negative Binomial GMIFS AIC selected model	Poisson GMIFS AIC selected model
True Nonzero				
Mean (Standard Deviation)	3.0 (1.70)	4.5 (0.69)	4.8 (0.39)	4.8 (0.40)
True Nonzero				
Median (Range)	3.0 (0.0, 5.0)	5.0 (3.0, 5.0)	5.0 (4.0, 5.0)	5.0 (4.0, 5.0)
False Nonzero				
Mean (Standard Deviation)	3.5 (3.78)	23.0 (6.82)	41.7 (11.72)	44.3 (10.6)
False Nonzero				
Median (Range)	2.0 (0.0, 16.0)	23.0 (8.0, 41.0)	42.0 (11.0, 68.0)	42.0 (19.0, 71.0)

Table 3.: Results from Simulation Studies when the true α is 0.9: Mean/Median number of true predictors that had a nonzero coefficient estimate in the final model (True Nonzero) and the mean/median number of false predictors that had a nonzero coefficient estimate in the final model (False Nonzero). Oracle number of true non-zero coefficients, $P_1 = 5$. Oracle number of zero coefficients, $P - P_1 = 495$.

	Negative Binomial GMIFS BIC selected model	Poisson GMIFS BIC selected model	Negative Binomial GMIFS AIC selected model	Poisson GMIFS AIC selected model
True Nonzero				
Mean (Standard Deviation)	1.8 (1.29)	4.0 (0.93)	4.4 (0.87)	4.4 (0.75)
True Nonzero				
Median (Range)	2.0 (0.0, 5.0)	4.0 (1.0, 5.0)	5.0 (1.0, 5.0)	5.0 (2.0, 5.0)
False Nonzero				
Mean (Standard Deviation)	1.9 (2.16)	27.1 (8.34)	45.6 (15.70)	52.8 (10.96)
False Nonzero				
Median (Range)	1.0 (0.0, 12.0)	26.0 (7.0, 57.0)	48.0 (0.0, 71.0)	51.0 (33.0, 77.0)

The α estimates for the BIC selected negative binomial GMIFS models have a mean of 0.3 and standard deviation of 0.27 when the true α value is 0.1, a mean of 1.2 and standard deviation of 0.82 when the true α value is 0.5, and a mean of 1.9 and standard deviation of 0.79 when the true α value is 0.9. The α estimates for the AIC selected negative binomial GMIFS models have a mean of 0.005 and standard deviation of 0.023 when the true α value is 0.1, a mean of 0.06 and standard deviation of 0.13 when the true α value is 0.5, and a mean of 0.14 and a standard deviation of 0.33 when the true α value is 0.9. Perhaps the inclusion of so many extraneous predictors in AIC selected models reduced the overdispersion to underestimate α values. This may indicate the need for an improved information criteria that selects models somewhere between AIC and BIC. Boxplots of the number of true non-zero (Figure 10) and false non-zero coefficients (Figure 9) selected by the models for the

simulated data were examined. Recall that the true number of non-zero coefficients is 5 and that there were 495 extraneous coefficients that truly have a zero coefficient that could be selected by the model as a false non-zero coefficients.

Fig. 9: Boxplot of the False Non-zero Coefficients selected by each of the minimum AIC (left) and minimum BIC (right) models when there was no offset.

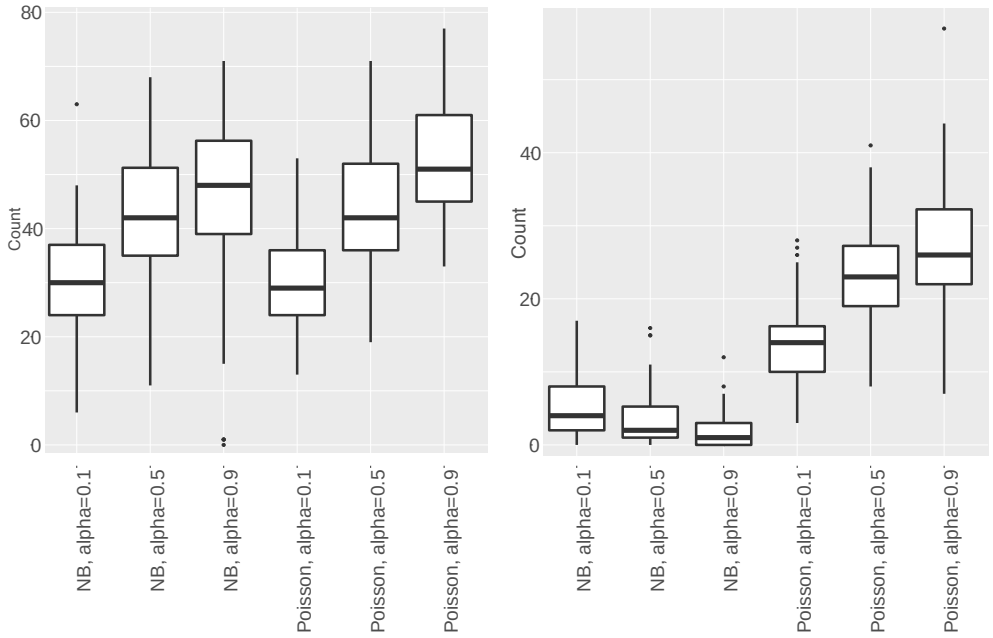
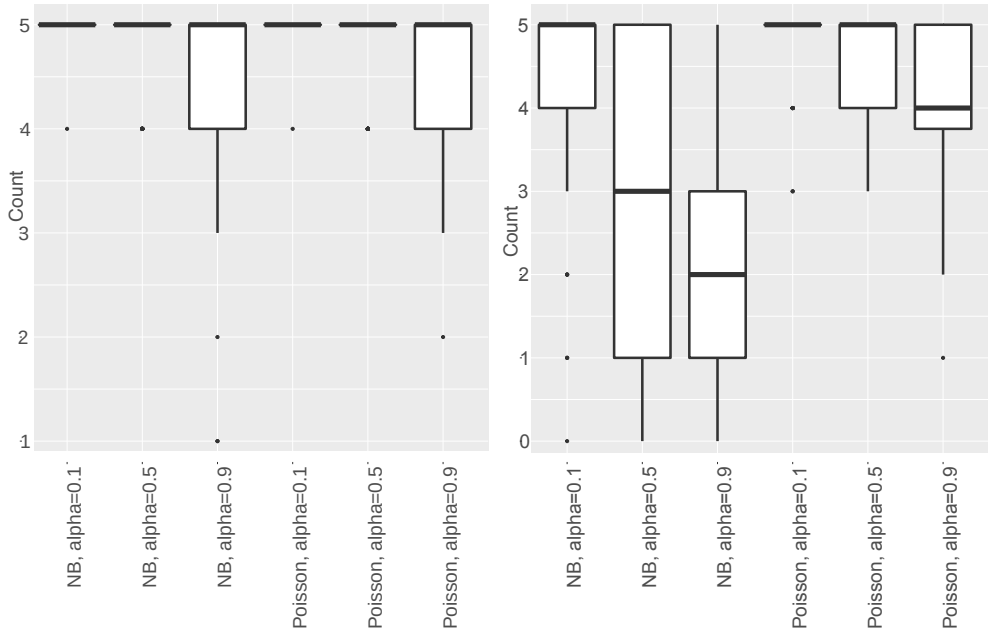


Fig. 10: Boxplot of the True Non-zero Coefficients selected by each of the minimum AIC (left) and minimum BIC (right) models when there was no offset.



The results for the simulation studies when $\alpha = 0.1, 0.5$, and 0.9 with an offset appear in Tables 4, 5, and 6. Similar to the models without an offset, the negative binomial and Poisson model have similar sensitivity, however the negative binomial model has more specificity for weeding out false predictors.

Table 4.: Results from Simulation Studies when the true α is 0.1 and there is an offset term: Mean/Median number of true predictors that had a nonzero coefficient estimate in the final model (True Nonzero) and the mean/median number of false predictors that had a nonzero coefficient estimate in the final model (False Nonzero). Oracle number of true non-zero coefficients, $P_1 = 5$. Oracle number of zero coefficients, $P - P_1 = 495$.

	Negative Binomial GMIFS BIC selected model	Poisson GMIFS BIC selected model	Negative Binomial GMIFS AIC selected model	Poisson GMIFS AIC selected model
True Nonzero				
Mean (Standard Deviation)	5 (0.0)	5.0 (0.0)	5.0 (0.0)	5.0 (0.0)
True Nonzero				
Median (Range)	5.0 (5.0, 5.0)	5.0 (5.0, 5.0)	5.0 (5.0, 5.0)	5.0 (5.0, 5.0)
False Nonzero				
Mean (Standard Deviation)	5.4 (4.53)	92.8 (0.38)	49.7 (17.28)	92.8 (0.37)
False Nonzero				
Median (Range)	4.0 (0.0, 29.0)	93.0 (92.0, 93.0)	51.5 (7.0, 82.0)	93.0 (92.0, 93.0)

Table 5.: Results from Simulation Studies when the true α is 0.5 and there is an offset term: Mean/Median number of true predictors that had a nonzero coefficient estimate in the final model (True Nonzero) and the mean/median number of false predictors that had a nonzero coefficient estimate in the final model (False Nonzero). Oracle number of true non-zero coefficients, $P_1 = 5$. Oracle number of zero coefficients, $P - P_1 = 495$.

	Negative Binomial GMIFS BIC selected model	Poisson GMIFS BIC selected model	Negative Binomial GMIFS AIC selected model	Poisson GMIFS AIC selected model
True Nonzero				
Mean (Standard Deviation)	4.8 (0.65)	5.0 (0.0)	5.0 (0.0)	5.0 (0.0)
True Nonzero				
Median (Range)	5.0 (2.0, 5.0)	5.0 (5.0, 5.0)	5.0 (5.0, 5.0)	5.0 (5.0, 5.0)
False Nonzero				
Mean (Standard Deviation)	7.3 (4.72)	92.8 (0.40)	44.5 (15.69)	92.8 (0.40)
False Nonzero				
Median (Range)	6.5 (0.0, 27.0)	93.0 (92.0, 93.0)	44.0 (12.0, 74.0)	93.0 (92.0, 93.0)

Table 6.: Results from Simulation Studies when the true α is 0.9 and there is an offset term: Mean/Median number of true predictors that had a nonzero coefficient estimate in the final model (True Nonzero) and the mean/median number of false predictors that had a nonzero coefficient estimate in the final model (False Nonzero). Oracle number of true non-zero coefficients, $P_1 = 5$. Oracle number of zero coefficients, $P - P_1 = 495$.

	Negative Binomial GMIFS BIC selected model	Poisson GMIFS BIC selected model	Negative Binomial GMIFS AIC selected model	Poisson GMIFS AIC selected model
True Nonzero				
Mean (Standard Deviation)	3.9 (1.25)	4.9 (0.36)	4.8 (0.49)	4.9 (0.36)
True Nonzero				
Median (Range)	4.0 (1.0, 5.0)	5.0 (3.0, 5.0)	5.0 (2.0, 5.0)	5.0 (3.0, 5.0)
False Nonzero				
Mean (Standard Deviation)	6.4 (4.70)	92.9 (0.57)	38.8 (15.53)	92.9 (0.57)
False Nonzero				
Median (Range)	5.5 (0.0, 20.0)	93.0 (92.0, 95.0)	39.5 (9.0, 71.0)	93.0 (92.0, 95.0)

The α estimates for the BIC selected negative binomial GMIFS models where there is an offset have a mean of 0.1 and standard deviation of 0.03 when the true α value is 0.1, a mean of 0.7 and standard deviation of 0.18 when the true α value is 0.5, and a mean of 1.2 and standard deviation of 0.34 when the true α value is 0.9. The α estimates for the AIC selected negative binomial GMIFS models where there is an offset have a mean of 0.07 and standard deviation of 0.023 when the true α value is 0.1, a mean of 0.3 and standard deviation of 0.09 when the true α value is 0.5, and a mean of 0.6 and a standard deviation of 0.16 when the true α value is 0.9. Figures 11 and 12 graphically display boxplots of the raw counts of true non-zero and false non-zero coefficients selected by the models for the simulated data. Recall that the true number of non-zero coefficients is 5 and 495 extraneous coefficients that

truly have a zero coefficient that could be selected by the model as a false non-zero coefficients.

Fig. 11: Boxplot of the False Non-zero Coefficients selected by each of the minimum AIC (left) and minimum BIC (right) models when there was offset.

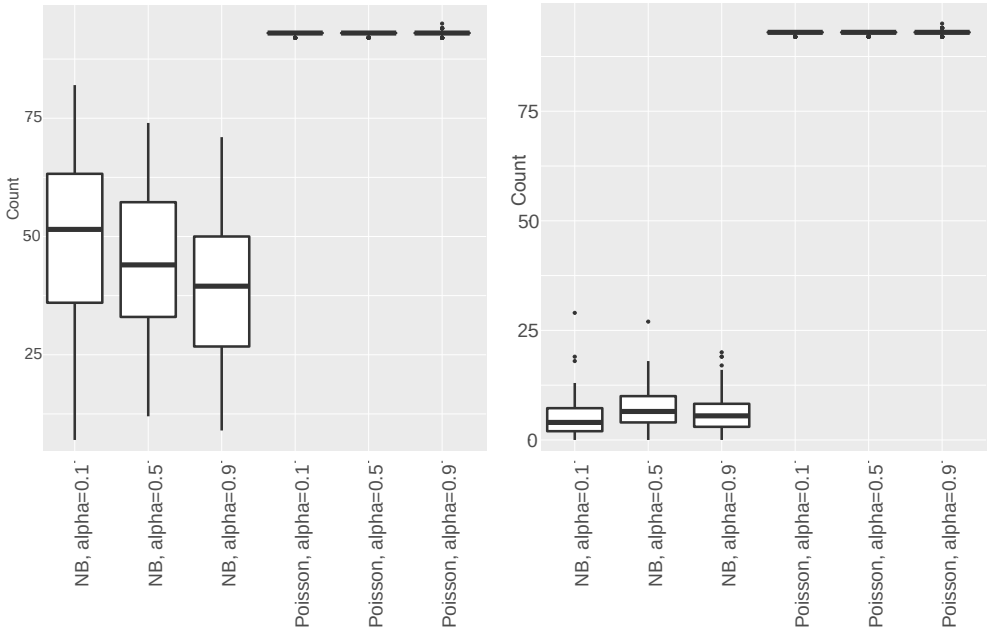
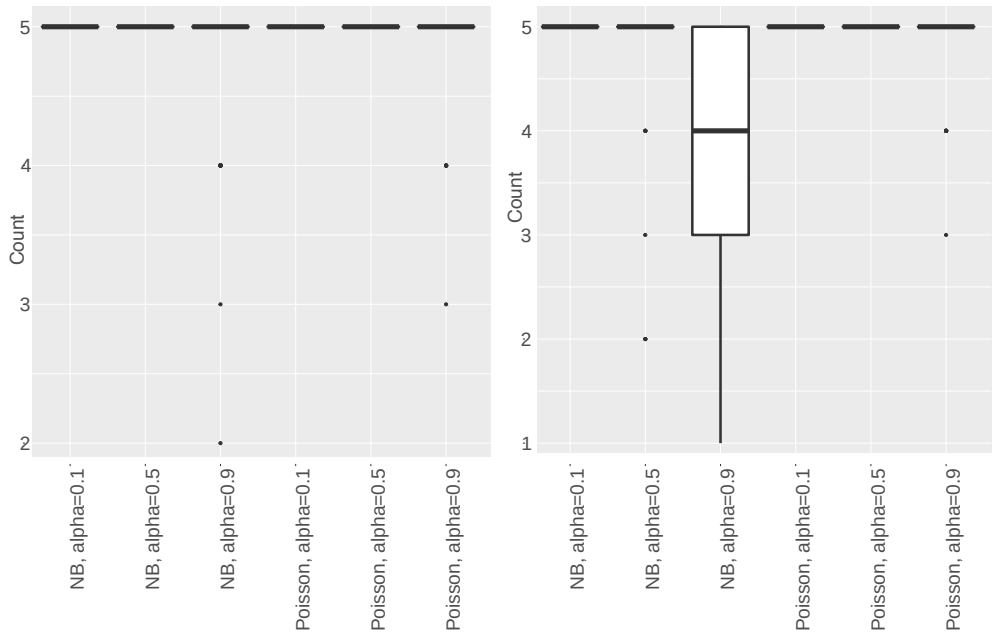


Fig. 12: Boxplot of the True Non-zero Coefficients selected by each of the minimum AIC (left) and minimum BIC (right) models when there was offset.



Overall, for all α levels and for data with and without an offset, if the underlying distribution of the data is negative binomial, the negative binomial GMIFS outperforms the Poisson GMIFS. The negative binomial models are more parsimonious than the Poisson models. While the Poisson and negative binomial models have similar sensitivity, the negative binomial model have better specificity for not selecting false non-zero coefficients.

2.4 Application to MoBa Data

For the MoBa data the outcome analyzed was MN frequency. Though maternal age, gestational age, or maternal smoking status may have been of interest to examine, those data were not available so the only unpenalized predictor was gender. The penalized predictors were the gene expression data. The MoBa data were analyzed

using the negative binomial GMIFS, Poisson GMIFS, and Poisson `glmpath`[25]. Previously Makowski and Archer, 2015 showed that Poisson `glmpath` models overfit both in simulation studies and in the application to real data[25].

For the negative binomial GMIFS model a plot of the negative log-likelihood and how it varies at each step of the GMIFS procedure may be seen in Figure 13, followed by the corresponding AIC and BIC values in Figure 14.

Fig. 13: Log-likelihood Plot for the Negative Binomial GMIFS.

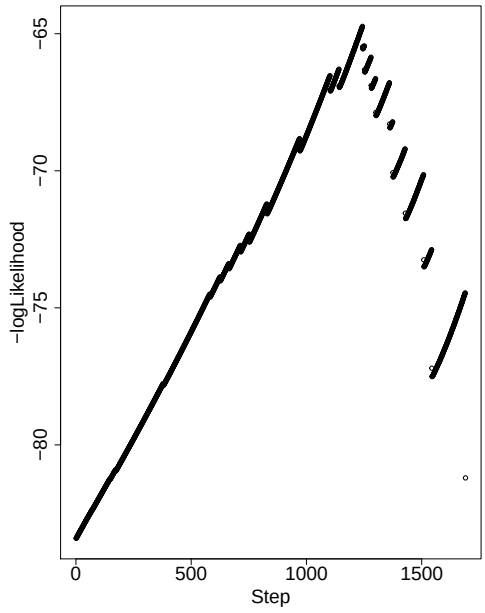
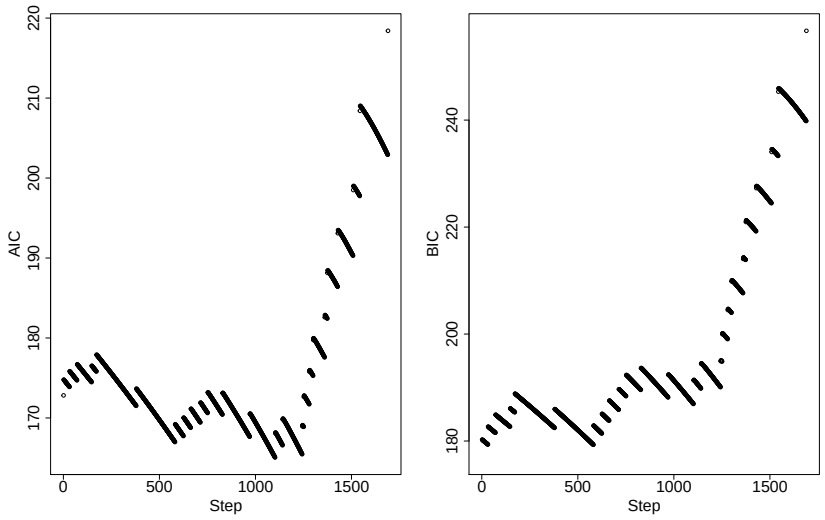
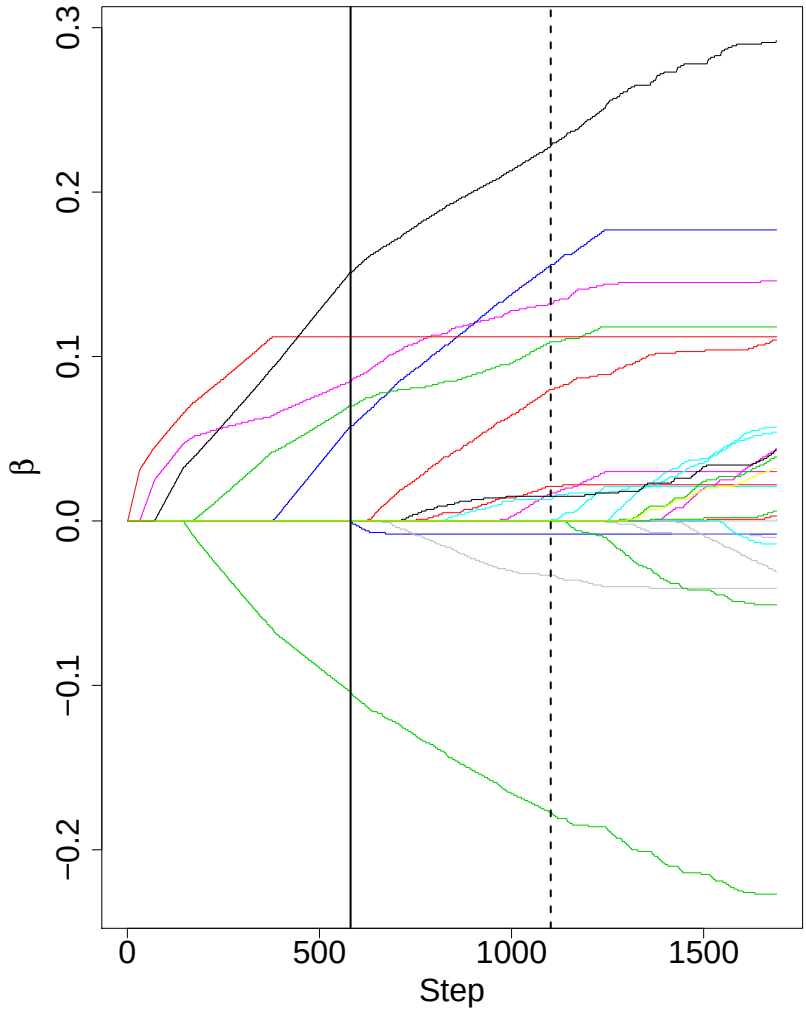


Fig. 14: AIC (left panel) and BIC (right panel) Plot for the Negative Binomial GMIFS.



It can be seen that the minimum BIC occurs right past step 500 and the minimum AIC occurs right past step 1000. The BIC selected model is the more parsimonious model. Figure 15 shows the coefficients paths for the negative binomial GMIFS model. Each coefficient is represented by a different colored line such that you can see when a new coefficient enters the model.

Fig. 15: Plot of coefficient path for the negative binomial GMIFS model with a dotted vertical line representing when the minimum AIC is achieved and a solid line representing where the minimum BIC is achieved.



The negative binomial GMIFS AIC selected model identified 13 genes associated with the MN count or 13 genes with non-zero coefficient estimates. Of those 13, six were also selected by the more parsimonious BIC selected model. Table 7 lists the genes that were selected by the negative binomial GMIFS model using both AIC and BIC for selecting the final model. Also in Table 7 are the corresponding probe

ID, gene name, and whether previous research has shown that this gene is linked to cancer.

Table 7.: Genes Associated with MN Count by the AIC selected and BIC selected Negative Binomial GMIFS Model.

Probe ID	Gene Symbol	Gene Name	Associated with Cancer	AIC selected NB GMIFS	BIC selected NB GMIFS
A_23_P100196	USP10	ubiquitin specific peptidase 10	Glioblastoma multiforme [Grunda et al., 2006]	X	X
A_23_P133424	SKP1	None Found	None Found	X	
A_23_P138967	SDHD	succinate dehydrogenase complex	Tumor Suppressor [King et al., 2006]	X	
A_23_P209394	CFLAR	CASP8 and FADD-like apoptosis regulator	Human cancers [Fulda, 2013]	X	
A_23_P42331	HMGA1	high mobility group AT-hook 1	Pancreatic Adenocarcinoma [Liau et al., 2008]	X	
A_24_P19410	CBX7	chromobox homolog 7	Carcinomas [Federico et al., 2009]	X	X
A_24_P214858	TREML2	triggering receptor expressed on myeloid cells-like 2	Pancreatic [Loos et al., 2009]	X	
A_24_P2463	WHSC1	Wolf-Hirschhorn syndrome candidate 1	Carcinogenesis [Toyokawa et al., 2011]	X	X
A_24_P333019	RNF24	ring finger protein 24	Oral squamous cell carcinoma [Cheong et al., 2009]	X	
A_24_P397584	TBCC	tubulin folding cofactor C	None Found	X	
A_24_P398064	KIAA0258	KIAA0258	Colorectal [Sasaki et al., 2008]	X	X
A_32_P156549	C1ORF144	None Found	None Found	X	X
A_32_P18547	C21ORF57	chromosome 21 open reading frame 57	Breast [Smeets et al., 2011]	X	X

For the Poisson GMIFS model a plot of the negative log-likelihood may be seen in Figure 16, followed by the corresponding AIC and BIC values in Figure 17.

Fig. 16: Log-likelihood Plot for the Poisson GMIFS.

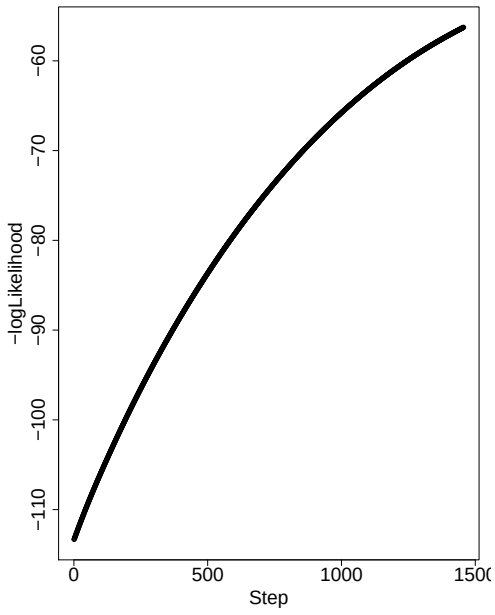
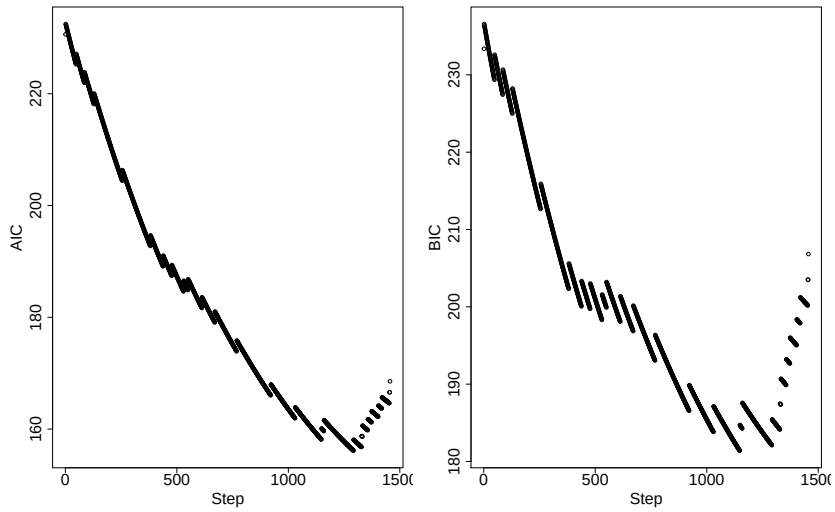
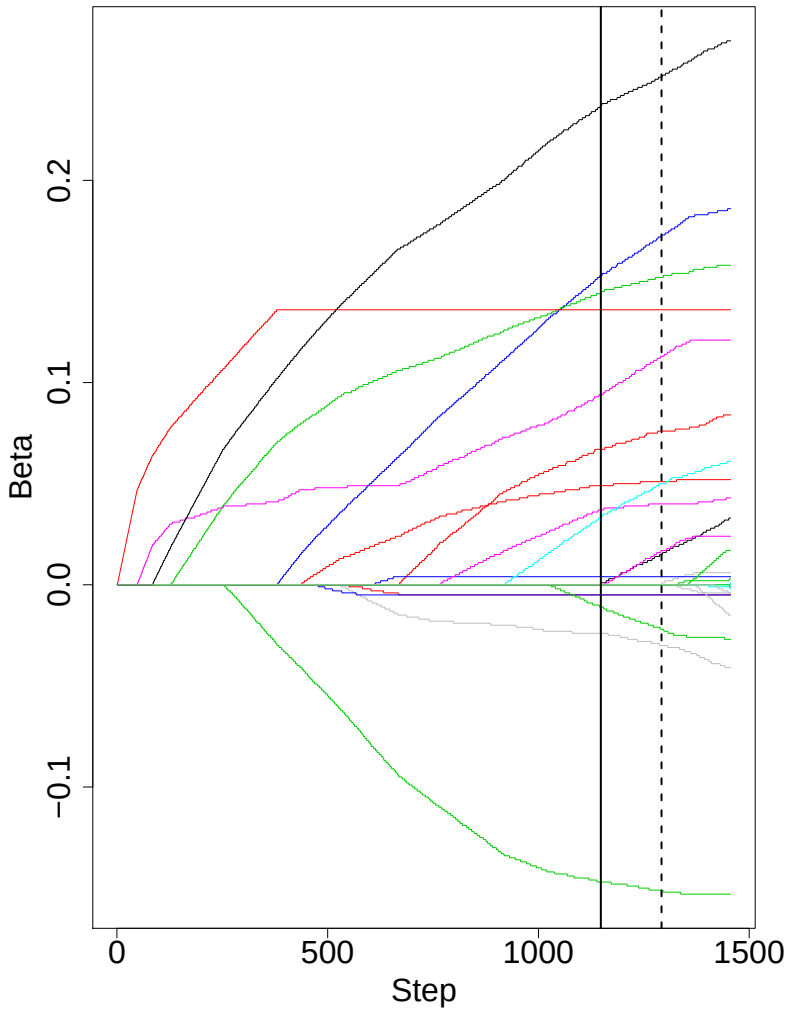


Fig. 17: AIC (left panel) and BIC (right panel) Plot for the Poisson GMIFS.



It can be seen that the minimum BIC occurs around step 1100 and the minimum AIC occurs around step 1300. Figure 18 shows the coefficients paths for the Poisson GMIFS model and indicates when the minimum AIC and minimum BIC occur.

Fig. 18: Plot of coefficient path for the Poisson GMIFS model with a dotted vertical line representing when the minimum AIC is achieved and a solid line representing where the minimum BIC is achieved.



Similarly, the Poisson GMIFS AIC selected model identified 17 genes associated with the MN count or 17 genes with non-zero coefficient estimates. Of those 17, 15 were also selected by the more parsimonious BIC selected model. Eleven of the genes were common across the AIC negative binomial model and AIC Poisson model. Table 8 lists the genes that were selected by the Poisson GMIFS model using both

AIC and BIC for selecting the final model. Also in Table 8 are the corresponding probe ID, gene name, and whether previous research has shown that this gene is linked to cancer.

Table 8.: Genes Associated with MN Count by the AIC selected and BIC selected Poisson GMIFS Model.

Probe ID	Gene Symbol	Gene Name	Associated with Cancer	AIC selected Poisson GMIFS	BIC selected Poisson GMIFS
A_23_P100196	USP10	ubiquitin specific peptidase 10	Glioblastoma multiforme [Grunda et al., 2006]	X	X
A_23_P103824	FAU	Finkel-Biskis-Reilly murine sarcoma virus (FBR-MuSV) ubiquitously expressed	None Found	X	X
A_23_P138967	SDHD	succinate dehydrogenase complex	Tumor Suppressor [King et al., 2006]	X	X
A_23_P209394	CFLAR	CASP8 and FADD-like apoptosis regulator	Human cancers [Fulda, 2013]	X	X
A_23_P42331	HMGA1	high mobility group AT-hook 1	Pancreatic Adenocarcinoma [Liau et al., 2008]	X	X
A_23_P79911	PSMF1	proteasome (prosome, macropain) inhibitor subunit 1 (PI31)	Breast Kuznetsova et al. [2006]	X	X
A_23_P9293	TJP2	tight junction protein 2	Breast [Martin et al., 2004]	X	X
A_24_P19410	CBX7	chromobox homolog 7	Carcinomas [Federico et al., 2009]	X	X
A_24_P202567	ITPKC	inositol 1,4,5-trisphosphate 3-kinase C	Cervical [Yang et al., 2012]	X	X
A_24_P214858	TREML2	triggering receptor expressed on myeloid cells-like 2	Pancreatic [Loos et al., 2009]	X	X
A_24_P2463	WHSC1	Wolf-Hirschhorn syndrome candidate 1	Carcinogenesis [Toyokawa et al., 2011]	X	X
A_24_P31235	EIF5A	eukaryotic translation initiation factor 5A	Chronic myeloid leukemia [Balabanov et al., 2007]	X	X
A_24_P397584	TBCC	tubulin folding cofactor C	None Found	X	
A_24_P398064	KIAA0258	KIAA0258	Colorectal [Sasaki et al., 2008]	X	X
A_24_P405054	C1ORF144	chromosome 1 open reading frame 144	Mantle cell lymphoma [Schraders et al., 2008]	X	
A_32_P156549	C1ORF144	None Found	None Found	X	X
A_32_P18547	C21ORF57	chromosome 21 open reading frame 57	Breast [Smeets et al., 2011]	X	X

The `glmpath` minimum AIC model which occurs at step 66 selected 23 genes and the minimum BIC model occurring at step 37 selected 17 genes to be associated with MN frequency. When plot functions for the `glmpath` model were performed R shut down. Table 9 lists all genes selected by either the AIC or BIC `glmpath` model. `glmpath` selected the most predictors in the final model. As previously reported by Makowski and Archer, 2015 and observed in their simulation studies, the large number of predictors included in `glmpath` Poisson models implies overfitting.

Table 9.: Genes Associated with MN Count by the AIC selected and BIC selected *glmpath* model.

Probe ID	Gene Symbol	Gene Name	Associated with Cancer	AIC selected *glmpath*	BIC selected *glmpath*
A_23_P100196	USP10	ubiquitin specific peptidase 10	Glioblastoma multiforme [Grunda et al., 2006]	X	X
A_23_P118313	GABARAPL2	GABA(A) receptor-associated protein-like 2	Breast Cancer [Hervouet et al., 2015]	X	
A_23_P138967	SDHD	succinate dehydrogenase complex	Tumor Suppressor [King et al., 2006]	X	X
A_23_P143817	MYLK	myosin light chain kinase	Colon Cancer [Stadler et al., 2016]	X	
A_23_P156809	FAM119A	family with sequence similarity 119, member A	None Found	X	
A_23_P394304	PDZK1	interacting protein 1	Renal Cell Carcinoma [Zheng et al., 2016]	X	
A_23_P39665	SLC11A1	solute carrier family 11	Lung Cancer [Zhang et al., 2013]	X	
A_23_P42331	HMGA1	high mobility group AT-hook 1	Pancreatic Adenocarcinoma [Liau et al., 2008]	X	X
A_23_P67529	KCNN4	potassium intermediate	None Found	X	X
A_23_P9293	TMEM169	transmembrane protein 169	None Found	X	
A_24_P19410	CBX7	chromobox homolog 7	Carcinomas [Federico et al., 2009]	X	X
A_24_P214858	TREML2	triggering receptor expressed on myeloid cells-like 2	Pancreatic [Loos et al., 2009]	X	X
A_24_P2463	WHSC1	Wolf-Hirschhorn syndrome candidate 1	Carcinogenesis [Toyokawa et al., 2011]	X	X
A_24_P397584	TBCC	tubulin folding cofactor C	None Found	X	
A_24_P398064	KIAA0258	KIAA0258	Colorectal [Sasaki et al., 2008]	X	X
A_24_P594683	None Found	None Found	None Found	X	X
A_24_P708161	None Found	None Found	None Found	X	
A_24_P98086	GNA12	guanine nucleotide binding protein	Breast Cancer [Mutlu et al., 2016]	X	
A_32_P10067	None Found	None Found	None Found	X	
A_32_P137849	None Found	None Found	None Found	X	
A_32_P169754	YBX1	Y box binding protein 1	Breast Cancer [Lim et al., 2017]	X	
A_32_P18547	C21ORF57	chromosome 21 open reading frame 57	Breast [Smeets et al., 2011]	X	X
A_32_P208078	MTHFR	5,10-methylenetetrahydrofolate reductase	Oral Squamous Cell Cancer [Ferlazzo et al., 2017]		X
A_23_P100196	USP10	ubiquitin specific peptidase 10	Glioblastoma multiforme [Grunda et al., 2006]		X
A_23_P209394	CFLAR	CASP8 and FADD-like apoptosis regulator	Human cancers [Fulda, 2013]		X
A_23_P39665	RPS6KA1	ribosomal protein S6 kinase	None Found		X
A_23_P9293	None Found	None Found	None Found		X
A_24_P227927	IL21R	interleukin 21 receptor	None Found		X
A_24_P31235	EIF5A	eukaryotic translation initiation factor 5A	Chronic myeloid leukemia [Balabanov et al., 2007]		X
A_24_P333019	RNF24	ring finger protein 24	Oral squamous cell carcinoma [Cheong et al., 2009]		X
A_32_P452655	LGALS9C	lectin, galactoside-binding	Pancreatic adenocarcinoma [Dhanraj et al., 2013]		X

Figures 19 to 24 depict the predicted MN count for each selected model versus actual MN count. From the figures it can be seen that the `glmpath` model is acutely overfitting. This was also demonstrated in Table 9 by the substantial number of predictors selected to be included in the final model. Recall the sample size for the Norwegian data was 29 babies. There were 23 genes included in the final model along with an intercept and the unpenalized predictor, gender. Therefore we are estimating 25 coefficients with a sample size of only 29. The AIC selected negative binomial

model appears to have the best fit. Both the AIC and BIC selected negative binomial models are superior to the AIC and BIC selected Poisson models.

Fig. 19: Predicted MN Count vs. Actual MN Count for the Poisson GMIFS Model with the minimum AIC.

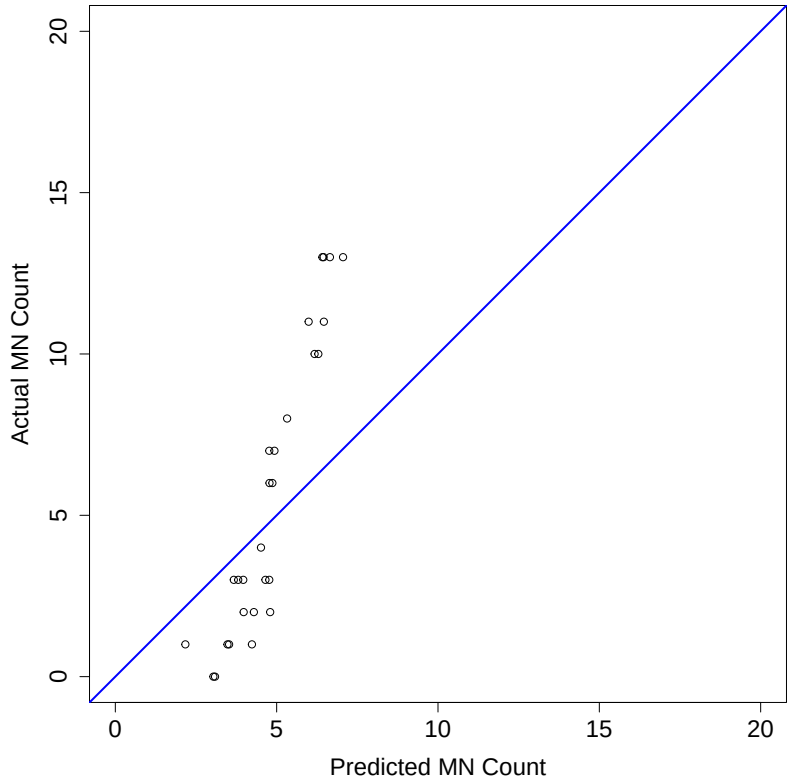


Fig. 20: Predicted MN Count vs. Actual MN Count for the Poisson GMIFS Model with the minimum BIC.

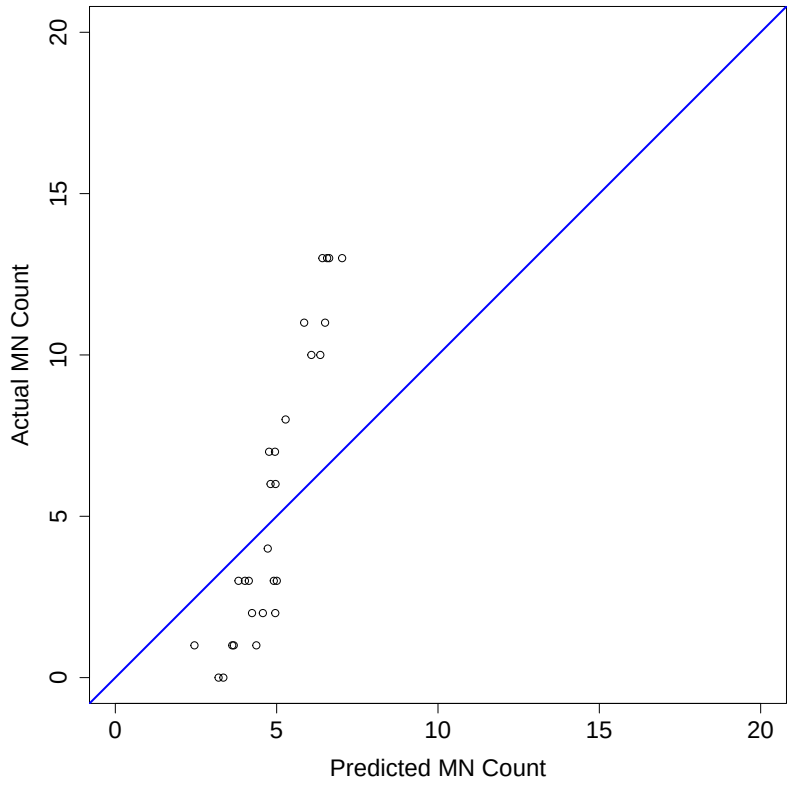


Fig. 21: Predicted MN Count vs. Actual MN Count for the Negative Binomial GMIFS Model with the minimum AIC.

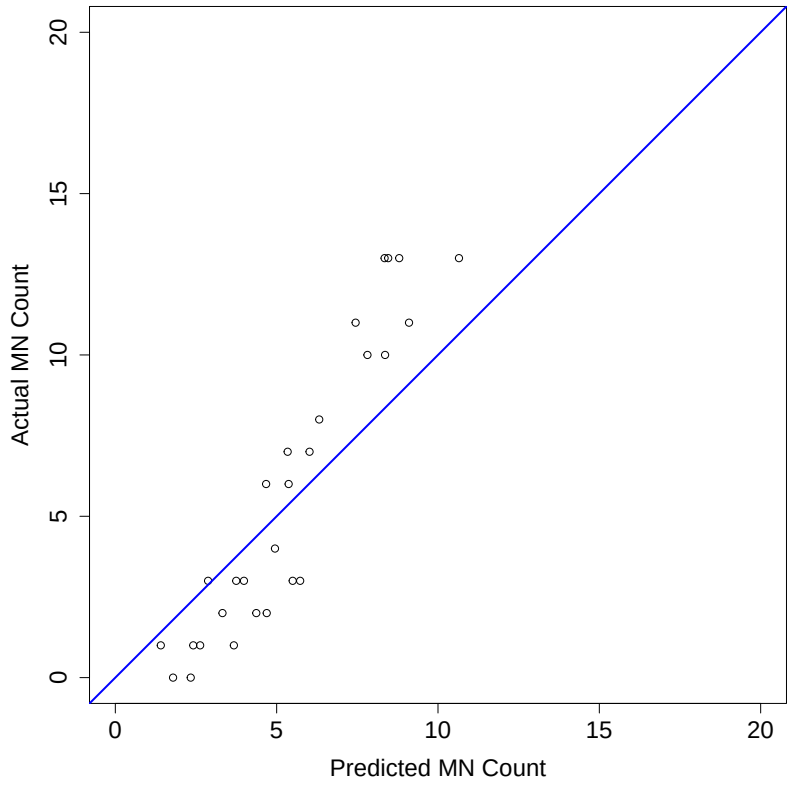


Fig. 22: Predicted MN Count vs. Actual MN Count for the Negative Binomial GMIFS Model with the minimum BIC.

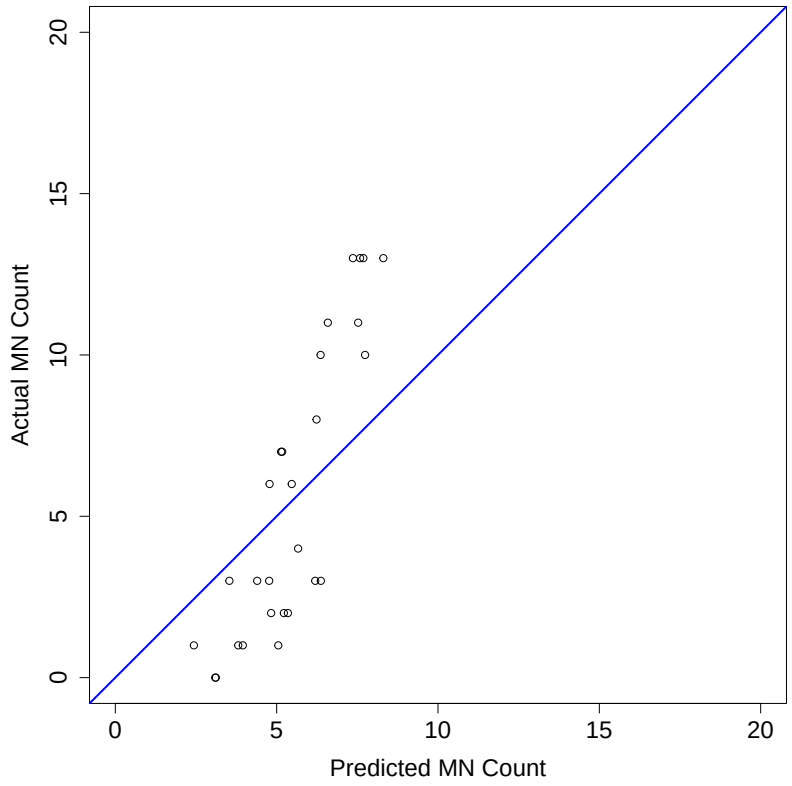


Fig. 23: Predicted MN Count vs. Actual MN Count for the `glmpath` Model with the minimum AIC.

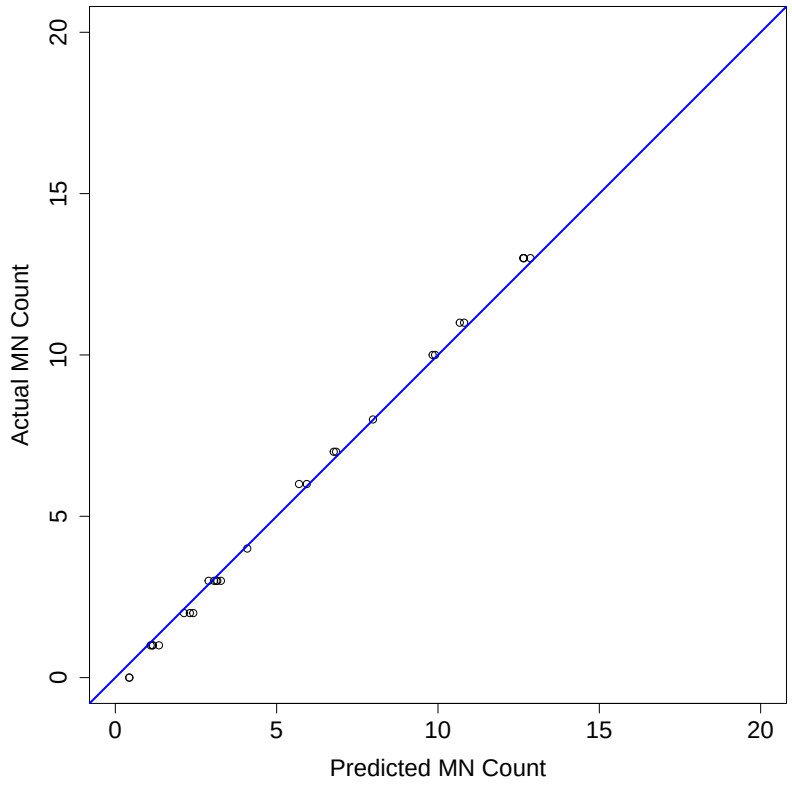
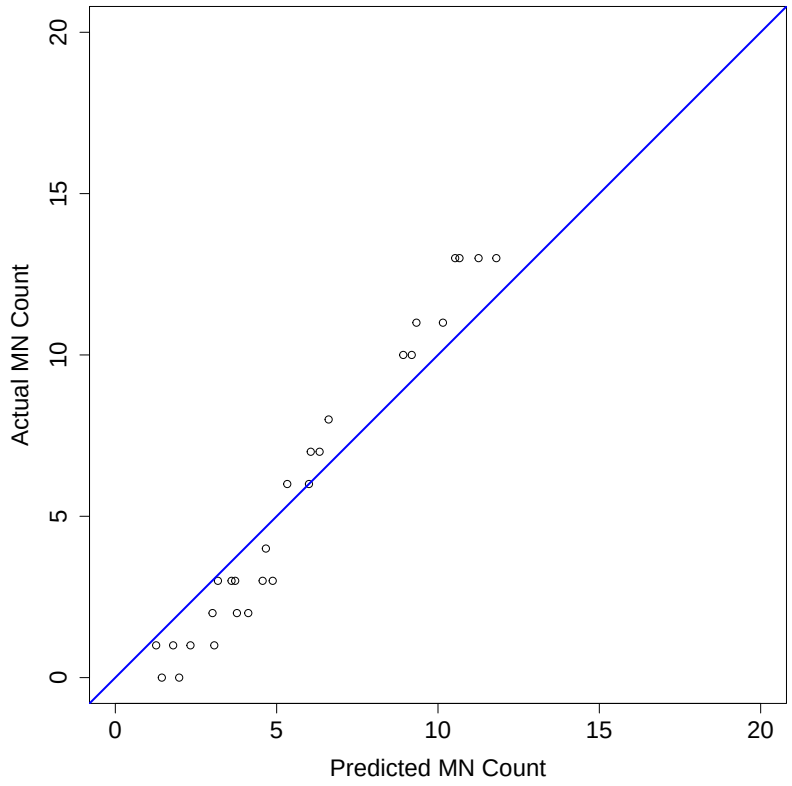


Fig. 24: Predicted MN Count vs. Actual MN Count for the `glmpath` Model with the minimum BIC.



A chi-square goodness of fit test was performed for the final AIC selected Poisson GMIFS model, BIC selected Poisson GMIFS model, AIC selected negative binomial model, and BIC selected negative binomial model. The AIC selected Poisson GMIFS model chi-square test results were $\chi^2_{27} = 1240.9$ with an associated p-value of < 0.001. The BIC selected Poisson GMIFS model chi-square test results were $\chi^2_{27} = 1413.6$ with an associated p-value of < 0.001. The AIC selected negative binomial GMIFS model

chi-square test results were $\chi^2_{26} = 6.5$ with an associated p-value near 1. The BIC selected negative binomial GMIFS model chi-square test results were $\chi^2_{26} = 5.7$ with an associated p-value near 1. The chi-square goodness of fit results conclude that the negative binomial model is a better fit than the Poisson models.

Figure 25 and 26 depict Venn diagrams for the AIC and BIC selected models. Recall that the BIC selected models will lend to more parsimonious models and thus select fewer predictors to be included in the final model. For the AIC selected models, nine of the penalized predictors are consistent across all three models. For the BIC selected models five of the penalized predictors are consistent across all three models. When comparing the AIC selected negative binomial GMIFS model to the AIC selected Poisson GMIFS model, there are 11 common predictors. The Poisson model selected six additional predictors not in the negative binomial model whereas the negative binomial model only selected two additional predictors not in the Poisson model. When comparing the BIC negative binomial GMIFS model to the BIC Poisson GMIFS model there are six common predictors. The Poisson model selected nine additional predictors not in the negative binomial model whereas the negative binomial model does not select any additional predictors. When comparing the AIC negative binomial GMIFS model to the AIC Poisson `glmpath` model, there are nine common predictors. The Poisson model selected 14 additional predictors not in the negative binomial model whereas the negative binomial model only selected four additional predictors not in the Poisson model. When comparing the BIC negative binomial GMIFS model to the BIC Poisson `glmpath` model, there are five common predictors. The Poisson model selected 12 additional predictors not in the negative binomial model whereas the negative binomial model only selected zero additional predictors not in the Poisson model.

Fig. 25: Venn Diagram of the AIC Negative Binomial GMIFS Model, Poisson GMIFS Model, and `glmpath` Model.

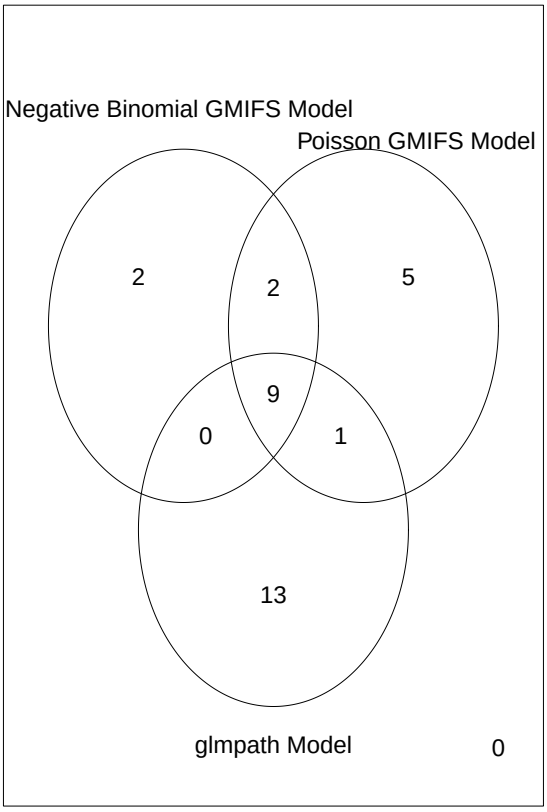
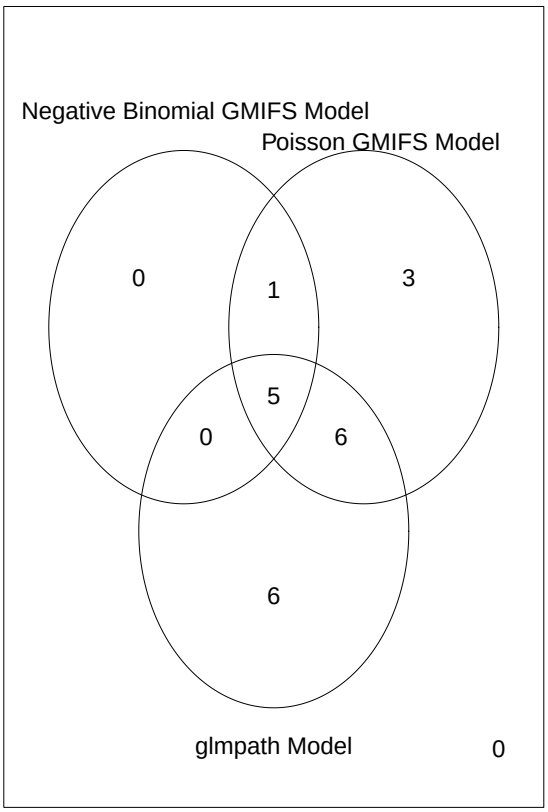


Fig. 26: Venn Diagram of the BIC Negative Binomial GMIFS Model, Poisson GMIFS Model, and `glmpath` Model.



2.5 Discussion

The simulation studies established that when the raw data follows a negative binomial distribution the negative binomial GMIFS outperforms the Poisson GMIFS and Poisson `glmpath`. The `glmpath` package in R suffered from convergence issues when the data were negative binomially distributed. The negative binomial GMIFS model had the same sensitivity as the Poisson GMIFS model, but more specificity for removing false predictors, particularly when an offset was used. Via goodness-of-fit test, it was determined that the MoBa micronuclei counts follow a negative binomial distribution. The `glmpath` Poisson model overfit the data. Promising was

that all three methods selected similar subsets of penalized predictors to be included in the final models selected using AIC or BIC. This may indicate underlying biological relevance of those genes as having an association with MN frequency or formation.

The developed method has accounted for overdispersion in traditional count data models when there is a high-dimensional predictor space. Chapters 3 and 4 will focus on extending the GMIFS method to the longitudinal setting for equidispersed and overdispersed data when there is a high-dimensional predictor space.

CHAPTER 3

THE LONGITUDINAL POISSON GENERALIZED MONOTONE INCREMENTAL FORWARD STAGEWISE METHOD

3.1 Statistical Methods

3.1.1 Current Methods for Analyzing a Count Outcome in a Longitudinal High-dimensional Setting

As previously described in Section 1.2.1 the Poisson distribution, which is frequently used to model count data, assumes that the data are equidispersed, thus the mean and the variance are equal to a single parameter, λ_i,

$$\mathbf{E}(y_i) = \text{Var}(y_i) = \lambda_i. \tag{3.1}$$

The conditional probability of a Poisson distributed random variable is given by

$$f(y_i|\lambda_i) = \frac{\exp(-\lambda_i)\lambda_i^{y_i}}{y_i!}. \tag{3.2}$$

The corresponding likelihood is given by

$$L(\lambda|\boldsymbol{y}) = \prod_{i=1}^{n} \frac{\exp(-\lambda_i)\lambda_i^{y_i}}{y_i!}. \tag{3.3}$$

When written with the optional offset included, the mean and variance are defined in terms of the offset, t_i,

$$\mathbf{E}(y_i) = \text{Var}(y_i) = t_i\lambda_i. \tag{3.4}$$

Recall that an offset is used when the outcome is a rate as opposed to a count. The conditional probability of the Poisson distributed random variable with an offset is

given by

$$f(y_i|\lambda_i) = \frac{\exp(-t_i\lambda_i)(t_i\lambda_i)^{y_i}}{y_i!}, \tag{3.5}$$

and the corresponding likelihood is given by

$$L(\lambda|\boldsymbol{y}) = \prod_{i=1}^{n} \frac{\exp(-t_i\lambda_i)(t_i\lambda_i)^{y_i}}{y_i!}. \tag{3.6}$$

Generalized linear mixed models (GLMMs) are commonly used to model correlated or clustered responses[35]. Let $i = 1, ..., N$ be the number of subjects and $j = 1, ..., n_i$ be the number of observations per subject. Therefore, the total number of observations is given by $n = \sum_{i=1}^{N} n_i$. In the longitudinal setting the observations, y_{ij}, are not assumed to be independent; instead, the observations are assumed to be clustered. Let $\boldsymbol{x}$ be the full design matrix of fixed effects. Let $\mathbf{u}$ be the q-dimensional vector of the coefficients of the random effects, $\boldsymbol{z}$. To specify the GLMM there are three parts:[12,31]

1. In the generalized linear mixed model for count data, the y_{ij} are independent and Poisson distributed and conditioned on a vector of random effects, u_i. It is still true in the Poisson setting that

$$\text{Var}(y_{ij}|\boldsymbol{u}_i) = \mathbf{E}(y_{ij}|\boldsymbol{u}_i). \tag{3.7}$$

2. The conditional mean of y_{ij} depends on fixed and random effects through the linear predictor by a log link function,

$$\eta_i = \log[E(y_{ij}|\boldsymbol{u}_i)] = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \boldsymbol{z}_i^\top \boldsymbol{u}_i. \tag{3.8}$$

3. It is assumed that the random effects are distributed multivariate normal with

mean zero and a covariance matrix, $\boldsymbol{\Sigma}$,

$$\boldsymbol{u}_i \sim N_q(\boldsymbol{0}, \boldsymbol{\Sigma}). \tag{3.9}$$

The pdf associated with the multivariate normal distribution is given by,

$$f(\boldsymbol{z}) = 2\pi^{-q/2}|\boldsymbol{\Sigma}|^{\frac{-1}{2}} \exp\left(\frac{-1}{2}(\boldsymbol{z}_i - \boldsymbol{u}_i)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_i - \boldsymbol{u}_i)\right). \tag{3.10}$$

The marginal likelihood function of the Poisson mixed effects linear model that is conditioned on the normally distributed random effects is given by,

$$\begin{aligned} L(\boldsymbol{\lambda}) &= \int_{R^q} \prod_{i=1}^{N} \prod_{j=1}^{n_i} \left[\frac{\exp(-\lambda_i)\lambda_i^{y_{ij}}}{y_{ij}!}\right] 2\pi^{-q/2}|\Sigma|^{\frac{-1}{2}} \exp\left(\frac{-1}{2}(\boldsymbol{z}_i - \boldsymbol{u}_i)'\Sigma^{-1}(\boldsymbol{z}_i - \boldsymbol{u}_i)\right) d\boldsymbol{u} \\ &= 2\pi^{-q/2}|\Sigma|^{\frac{-1}{2}} \int_{R^q} \prod_{i=1}^{N} \prod_{j=1}^{n_i} \left[\frac{\exp(-\lambda_i)\lambda_i^{y_{ij}}}{y_{ij}!}\right] \exp\left(\frac{-1}{2}(\boldsymbol{z}_i - \boldsymbol{u}_i)'\Sigma^{-1}(\boldsymbol{z}_i - \boldsymbol{u}_i)\right) d\boldsymbol{u}. \end{aligned} \tag{3.11}$$

In the GLMM setting the conditional density is of the exponential family type.

When analyzing a longitudinal count outcome in a high-dimensional setting there are currently two statistical methods that are applicable. Both methods are based on the traditional LASSO that was originally developed by Tibshirani in 1996 described in Section 2.2.1[31,35]. The LASSO is a regression technique that applies an L_1-penalty on the regression coefficients. The resulting effect is that all coefficients are shrunken towards zero and some are set exactly to zero. The LASSO method focuses on achieving sparse estimates. The concept of the original LASSO was to maximize the log-likelihood (l) of the model while constraining the L_1-norm of the parameter vector, thus the LASSO estimate can be obtained using,

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}}(l(\boldsymbol{\beta})) \tag{3.12}$$

subject to

$$||\boldsymbol{\beta}||_1 \leq s \tag{3.13}$$

with $s \geq 0$ and with $||.||_1$ denoting the L_1-norm. It is equivalent to estimating the parameters by solving the optimization problem,

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}}[l(\boldsymbol{\beta}) - \lambda||\boldsymbol{\beta}||_1] \tag{3.14}$$

where $\lambda \geq 0$. Note that s and λ are tuning parameters. These may be selected through cross-validation or by selecting values that minimize AIC or BIC. When in a high-dimensional data setting values for these tuning parameters may be time consuming to obtain. Therefore, efficient algorithms were developed to provide near optimal values[35].

With respect to longitudinal or clustered count outcomes, Groll and Tutz 2011 developed the `glmmLasso` method which is a variable selection technique for generalized linear mixed models that uses L_1-penalization[35] because traditional GLMM methods are limited to few predictors. By applying the LASSO method, the GLMM is expanded in such a way to handle a large numbers of predictors[35]. The L_1-penalty term enforces variable selection and shrinkage simultaneously. A gradient ascent algorithm is used to maximize the penalized log-likelihood producing models with reduced complexity[35]. The `glmmLasso` method is as follows[35],

1. Compute starting values of $\hat{\boldsymbol{\beta}}^{(0)}$, $\hat{\boldsymbol{b}}^{(0)}$, and $\hat{\boldsymbol{\gamma}}^{(0)}$ by fitting a global intercept model using the `glmmPQL` function in R from the `MASS` library.

2. For $l = 1, 2, ...$ until convergence, where convergence is based on changes in linear predictor:

   - Calculate the log-likelihood gradient for the given $\hat{\gamma}^{(l-1)}$

- Calculate the direction of the second dervative
- Determine the optimum of the Taylor approximation
- Update $\hat{\gamma}^{(l)}$

3. Fit a model that includes all variables that have non-zero $\hat{\boldsymbol{\beta}}$ values. Use Fisher scoring to determine the final estimates.

The simulation studies performed were in an overparameterized setting, however, the number of predictors was relatively small and only exceeded the number of samples by 10 with $p = 50$ and $n = 40$. Other simulation studies performed were not in a high-dimensional setting, $p < n$, where $n = 40$ and $p = 3, 5, 10$, and 20. In the simulation studies that are examined in Section 3.2, we examine more extreme cases of $p > n$.

In Groll and Tutz's paper they applied the glmmLASSO to multiple real data sets, two of which examine a Poisson distributed count outcome[35]. The first, The German Bundesliga was a soccer data set that was collected over 3 years for 18 soccer clubs. The outcome is a count based on the number of points scored and the covariates (p=7) include measures for: ball possession, tackle, unfairness, transfer spending, transfer receipts, attendance, and sold out. When the glmmLASSO model was fit three predictors were not found to be significant: unfairness, ball possesion, and tackles. The second data set, CD4 AIDS Study, uses the Multicenter AIDS Cohort Study that collected data on approximately 5,000 infected gay or bisexual men. The outcome of interest was the number of CD4+ cells. CD4+ cells decrease with time from infection and is a measure of AIDS progression[35]. Covariates of interest include time, drugs, partners, packs of cigarettes, mental illness score, and age. When the `glmmLasso` model was fit drugs and age were not found to be significant predictors. Limitations of Groll and Tutz's method include that there is not an option for an

offset term for the model. Recall, that an offset is used when a rate is analyzed as opposed to a count. This function has been implemented in R in the package `glmmLasso`.

Second, Schelldorfer et al. 2012 developed a method referred to as `GLMMLasso`[31]. This is a variable selection method that should be used to select fewer predictors than samples that are then used in fitting a traditional model. Their method relies on the assumption that many of the coefficients of the predictors are truly zero. The objective function considered is,

$$Q_\lambda(\boldsymbol{\beta}, \boldsymbol{\theta}, \phi) = -2 \log L(\boldsymbol{\beta}, \boldsymbol{\theta}, \phi) + \lambda ||\boldsymbol{\beta}||_1. \tag{3.15}$$

where $\lambda \geq 0$ is a regularization parameter. The parameters $\boldsymbol{\beta}, \boldsymbol{\theta}$, and ϕ are estimated by

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}, \hat{\phi}) = \underset{\boldsymbol{\beta}, \boldsymbol{\theta}, \phi}{\arg \min} \, Q_\lambda(\boldsymbol{\beta}, \boldsymbol{\theta}, \phi). \tag{3.16}$$

The `GLMMLasso` algorithm is summarized as[31]

1. Choose starting values for the parameters $\boldsymbol{\beta}^{(0)}, \boldsymbol{\theta}^{(0)}$, and $\phi^{(0)}$
2. Repeat for s=1,2,...
    - Calculate the Laplace approximation

    $$Q_\lambda^{LA}(\boldsymbol{\beta}^{(s,s-1;k)}, \boldsymbol{\theta}^{(s-1)}, \phi^{(s-1)}).$$

    - Quadratic approximation and inexact line search
        - Approximate the second derivative of the pdf.
        - Calculate the descent direction.
        - Choose a step size.

- Optimize the covariance parameter, for $l = 1, ..., d$

$$\theta_l^s = \underset{\theta_t}{\arg\min}\, Q_\lambda^{LA}(\boldsymbol{\beta}^{(s)}, \boldsymbol{\theta}^{(s,s-1,\theta_t)}, \phi^{(s-1)})$$

- Optimize the dispersion parameter

$$\phi^{(s)} = \underset{\phi}{\arg\min}\, Q_\lambda^{LA}(\boldsymbol{\beta}^{(s)}, \boldsymbol{\theta}^{(s)}, \phi)$$

3. Repeat until convergence.

While this is the generic method described by Schelldorfer et al., 2012 an approximation method and hybrid method are also described. This method has been suggested for variable selection followed by re-estimation of the model using traditional statistical methods for when $p < n$[31]. It is described as a highly efficient method when handling high-dimensional data[31].

Simulation studies were performed for the low-dimensional and high-dimensional Poisson mixed model. The following high-dimensional settings were evaluated: $n = 400$ with $p = 500$, $n = 400$ and $p = 1000$, and $n = 300$ and $p = 500$. Comparisons were made between `GLMMLasso` and other R functions that do not take into consideration the grouping structure of the data. The authors concluded that it is crucial to take into account the grouping structure. A limitation of this package is that it reports that there is observed slow convergence rate. No real data applications were performed using the Poisson model.

3.1.2 Proposed Extension of the Generalized Monotone Incremental Stagewise Method to the Longitudinal Poisson Distribution

The GMIFS method developed by Makowski and Archer, 2015 for modeling count data following the Poisson distribution was previously described in Section

2.2.1[25]. Herein we extend this method to accommodate longitudinal and clustered data and allow for an offset. To develop the Poisson GMIFS method for longitudinal data we need to estimate a high-dimensional linear mixed-effects model for count data outcomes, y_{ij}, following the Poisson distribution. Mixed-effects models include a combination of fixed and random effects. The fixed effects include all of the predictors of interest[18]. In contrast, the random effects account for the correlated nature of data arising from the same subject or cluster. Therefore, the variability must be partitioned to within and between cluster[18]. This is equivalent to partitioning longitudinal data that has measurements across time for each subject to between subject and within subject variability.

The log link function is used to re-write the conditional likelihood in terms of the predictors. The link function is given by

$$\log(\lambda_i) = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \boldsymbol{z}_i^\top \boldsymbol{u}_i. \tag{3.17}$$

In the GMIFS method it is necessary to be able to take the derivative of the conditional likelihood in terms of the coefficient $\boldsymbol{\beta}$. This is used to determine which predictor should be incremented at each step of the method. Therefore, the conditional likelihood, written in terms of the coefficients, is given by

$$L(\boldsymbol{\beta}) = 2\pi^{-q/2}|\Sigma|^{\frac{-1}{2}} \int_{R^q} \prod_{i=1}^{N} \prod_{j=1}^{n_i} \left[\frac{\exp(-\exp(\boldsymbol{x}_i^\top \boldsymbol{\beta} + \boldsymbol{z}_i^\top \boldsymbol{u}_i)) \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta} + \boldsymbol{z}_i^\top \boldsymbol{u}_i)^{y_{ij}}}{y_{ij}!} \right] \exp(\frac{-1}{2}(\boldsymbol{z}_i - \boldsymbol{u}_i)^\top \Sigma^{-1}(\boldsymbol{z}_i - \boldsymbol{u}_i)) d\boldsymbol{u} \tag{3.18}$$

Hou and Archer, 2015 showed that to solve for the maximum likelihood, it is only necessary to look at the marginal likelihood[20]. The derivative of the marginal likelihood

with respect to $\boldsymbol{\beta}$ is,

$$\frac{dL}{d\boldsymbol{\beta}} = \boldsymbol{x}_i^\top (y_{ij} - \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta} + \boldsymbol{z}_i^\top \boldsymbol{u}_i)). \tag{3.19}$$

The longitudinal model adds extra terms to the likelihood function for the random effects. It is necessary to estimate these random effects so that they may be integrated out of the likelihood. Then we can estimate the fixed effects terms. The `lme4` package in R can be used to estimate the random effects[4].

Again we have penalized and unpenalized predictors so we divide our parameter space into two components: the parameters ($\boldsymbol{\beta}$) that correspond to the penalized predictors ($\boldsymbol{x}$) and the parameters ($\boldsymbol{\gamma}$) that correspond to the unpenalized predictors ($\boldsymbol{w}$), and γ_0 which is the intercept. As previously described, unpenalized predictors are those which are forced into the model due to known significance or relationship with the outcome. Penalized predictors are those which the model will select for us automatically. The derivative of the marginal likelihood with respect to $\boldsymbol{\beta}$ may be expressed as,

$$\frac{dL}{d\boldsymbol{\beta}} = \boldsymbol{x}_i^\top (y_{ij} - \exp(\gamma_0 + \boldsymbol{w}_i^\top \boldsymbol{\gamma} + \boldsymbol{x}_i^\top \boldsymbol{\beta} + \boldsymbol{z}_i^\top \boldsymbol{u}_i)). \tag{3.20}$$

To simplify the GMIFS method we expand the penalized predictors such that $\boldsymbol{x}^{NEW} = [\boldsymbol{x} : -\boldsymbol{x}]$ to remove the need for a second derivative to determine the direction of the increment. The GMIFS algorithm which selects penalized predictors that should be retained for the longitudinal or clustered Poisson model is:

1. Set the step number, $s = 1$. Initialize the components of $\hat{\boldsymbol{\beta}}^s = 0$. Initialize the random effects, $\hat{\boldsymbol{u}}$, the intercept, $\hat{\gamma_0}$, and the unpenalized coefficients, $\hat{\gamma_j}$, where $j = 1, ..., J$ using the maximization algorithm of the log-likelihood.

2. Treating the fixed effects, $\hat{\boldsymbol{\gamma}}$, and $\hat{\gamma}_0$ and the random effects, $\hat{\boldsymbol{u}}$, as fixed find the predictor x_m such that $m = \arg\min_k(-\frac{dL}{d\beta_k})$.

3. If $\hat{\beta}_m^s = 0$ and $s > 1$, re-estimate $\hat{\boldsymbol{u}}$ using the new $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{\gamma}}$, and $\hat{\gamma}_0$ find the predictor x_m such that $m = \underset{k}{\arg\min}(-\frac{dL}{d\beta_k})$.
   Else, if $\hat{\beta}_m^s \neq 0$ proceed to step 4.

4. Update $\hat{\beta}_m^{s+1} = \hat{\beta}_m^s + \epsilon$.

5. Using the new $\boldsymbol{\beta}$ vector from step 4, update $\boldsymbol{\gamma}$ and γ_0 via the maximization algorithm of the log-likelihood. Step is updated to $s = s + 1$.

6. Repeat steps 2-5 until the difference between successive log-likelihoods is less than a pre-specified small tolerance, or $\delta * p \geq n$.

In the implementation of this algorithm we used `lme4` to estimate the random effects. In our GMIFS algorithm we used $\epsilon = 0.001$, $\tau = 0.001$, and $\delta = 0.10$. δ has been included in the stopping criteria because general sample size rules indicate that the number of predictors should maximally be approximately 10% of the sample size to prevent overparameterization, as noted by Harrell who suggests that if you expect to be able to detect reasonable-size effects with reasonable power, you need 10-20 observations per parameter (covariate) estimated[21]. While this is a general rule, δ is a user defined parameter which can be changed depending on the data and application. The final model will be selected based on some model selection criteria such as AIC or BIC.

Further, recall that an offset term is often used where there is a rate as opposed to a count outcome. The GMIFS algorithm accommodates the rate outcome through the link function described in Equation 1.24.

3.2 Simulation Studies

Simulation studies were conducted to evaluate the performance of our method. The design of the simulation studies is based on the application data set which will

be described in Section 4.1. The data that the method will be applied to is breast cancer data that was collect at five time points during the treatment of cancer. The Poisson data were simulated as follows,

1. For each of the n subjects, randomly generate an intercept from $N(0, 0.25)$.

2. Randomly generate the predictor set with P variables, $x_{i1}, x_{i2}, ..., x_{iP}$ where i= 1 to $N * n_i$ from a standard normal distribution.

3. Select a subset, P_1, of the P variables to be associated with the response. Non-zero coefficients, $\boldsymbol{\beta}$, were assigned to the P_1 of the P variables to be associated with the response. Also assign the intercept value, γ_0 and the coefficient for time, γ_1.

4. Set $\boldsymbol{Z}$ to be the $n_i * N$ x 2 design matrix of the random effects with the first column consisting of 1s that correspond to the random intercept and the second column to be from 0 to 4 consecutively for a random slope. The second column of the random effects is going to be referred to as time.

5. Generate
$$\lambda_{ij} = \exp\left(\gamma_0 + \text{time} \times \gamma_1 + \sum_{k=1}^{P_1} \beta_k x_{ik} + \sum_{l=1}^{2} u_{il} z_{ij}\right)$$

6. Randomly generate the response, $y_{ij} \sim \text{Poisson}(\lambda_{ij})$.

7. Repeat this to simulate r independent data sets.

The number of independent data sets simulated was $r = 100$, the sample size was set to $N = 100$ with each subject having five time points and $P = 200$ predictors plus the fixed effect, time. Of the P predictors, $P_1 = 5$ were selected to be associated with the response. For all data sets we fit a Poisson GMIFS model and the `glmmLasso` model[15,35]. Recall that a limitation of the `glmmLasso` model is that it does not

provide an option for an offset term. Therefore, we only compared models where there is no offset. Simulations were performed for coefficients of equal magnitude, $\boldsymbol{\beta} = (0.5, 0.5, -0.5, -0.5, -0.5)$ with an intercept value and coefficient for time of 0.5; we also varied the magnitude of the coefficients letting $\boldsymbol{\beta} = (0.25, -0.3, -0.4, 0.4, 0.3)$ with an intercept value and coefficient for time of 0.4 The two methods were compared with respect to the following:

- The number of true predictors that had a non-zero coefficient estimate;
- The number of false predictors that had a non-zero coefficient estimate.

The results from the simulation study when $\boldsymbol{\beta} = (0.5, 0.5, -0.5, -0.5, -0.5)$ appear in Table 10. The BIC selected models are more parsimonious than the AIC selected models, including a mean of 1.6 compared to 4.8 true predictors respectively. Overall, the Poisson GMIFS models are more parsimonious than Poisson glmmLASSO models. While the glmmLASSO models have more sensitivity and select more of the true non-zero predictors, the Poisson GMIFS models have more specificity for weeding out false predictors.

Table 10.: Results from Simulation Studies with coefficients of equal magnitude: Mean/Median number of true predictors that had a nonzero coefficient estimate in the final model (True Nonzero) and the mean/median number of false predictors that had a nonzero coefficient estimate in the final model (False Nonzero). Oracle number of true non-zero coefficients, $P_1 = 5$. Oracle number of zero coefficients, $P - P_1 = 195$.

	Poisson GMIFS BIC selected model	Poisson glmmLASSO BIC selected model	Poisson GMIFS AIC selected model	Poisson glmmLASSO AIC selected model
True Nonzero				
Mean (Standard Deviation)	1.6 (1.69)	4.8 (0.40)	1.6 (1.69)	4.8 (0.39)
True Nonzero				
Median (Range)	1.0 (0.0, 5.0)	5.0 (4.0, 5.0)	1.0 (0.0, 5.0)	5.0 (4.0, 5.0)
False Nonzero				
Mean (Standard Deviation)	2.7 (2.43)	150.8 (7.73)	2.9 (2.60)	161.1 (14.35)
False Nonzero				
Median (Range)	3.0 (0.0, 8.0)	151.0 (130.0, 165.0)	4.0 (0.0, 8.0)	157.5 (131.0, 192.0)

The simulation results are graphically displayed using boxplots in Figures 27 and 28. The boxplots depict the minimum, median, 25^{th}, 75^{th}, and maximum of the true non-zero and false non-zero coefficients selected by the models for the simulated data. Recall that the true number of non-zero coefficients is 5 and 195 extraneous coefficients that truly have a zero coefficient that could be selected by the model as a false non-zero coefficients.

Fig. 27: Boxplot of the True Non-zero Coefficients selected by each of the minimum AIC and minimum BIC longitudinal Poisson models when there were true coefficients of equal magnitude for both the GMIFS and glmmLASSO algorithms.

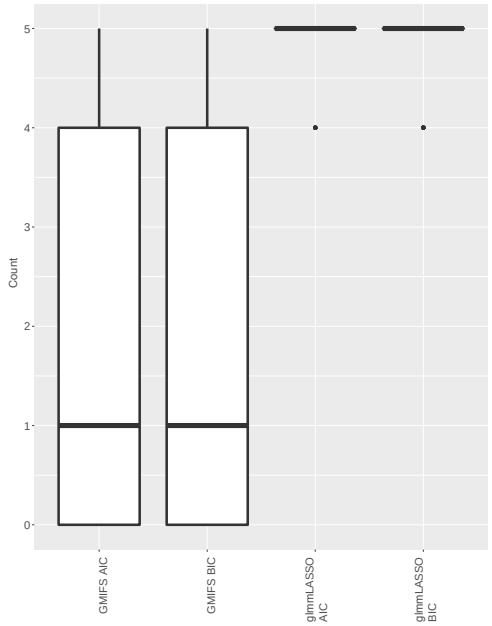
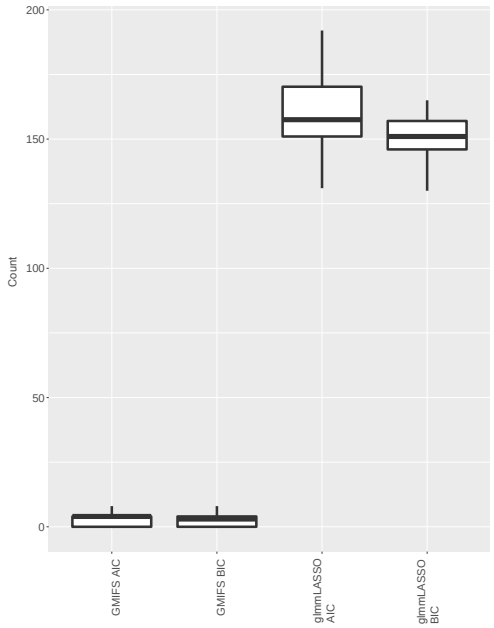


Fig. 28: Boxplot of the False Non-zero Coefficients selected by each of the minimum AIC and minimum BIC longitudinal Poisson models when there were true coefficients of equal magnitude for both the GMIFS and glmmLASSO algorithms.



The results from the simulation studies where the true coefficients had varying magnitudes appear in Table 11. Again, the BIC selected models are more parsimonious than the AIC selected models. As before, the Poisson GMIFS models are more parsimonious than Poisson `glmmLasso` models. While the `glmmLASSO` models have more sensitivity and select more of the true non-zero predictors, the Poisson GMIFS models have improved specificity for weeding out false predictors. Therefore the conclusions are equivalent for the simulation studies that employed either varying or equivalent $\boldsymbol{\beta}$ magnitudes.

Table 11.: Results from Simulation Studies of Poisson longitudinal models with coefficients of varying magnitude: Mean/Median number of true predictors that had a nonzero coefficient estimate in the final model (True Nonzero). Oracle number of true non-zero coefficients, $P_1 = 5$. The mean/median number of false predictors that had a nonzero coefficient estimate in the final model (False Nonzero). Oracle number of zero coefficients, $P - P_1 = 195$.

	Poisson GMIFS BIC selected model	Poisson glmmLASSO BIC selected model	Poisson GMIFS AIC selected model	Poisson glmmLASSO AIC selected model
True Nonzero				
Mean (Standard Deviation)	3.9 (0.64)	4.3 (0.47)	3.9 (0.64)	4.4 (0.48)
True Nonzero				
Median (Range)	4.0 (0.0, 5.0)	5.0 (4.0, 5.0)	4.0 (0.0, 5.0)	4.0 (4.0, 5.0)
False Nonzero				
Mean (Standard Deviation)	2.9 (1.19)	49.6 (33.67)	3.4 (1.00)	62.8 (33.58)
False Nonzero				
Median (Range)	3.0 (1.0, 6.0)	42.0 (4.0, 152.0)	4.0 (1.0, 6.0)	54.0 (11.0, 152.0)

The results from the simulation studies where the coefficients had varying magnitudes are graphically displayed using boxplots in Figures 29 and 30. The boxplots include the minimum, median, 25^{th}, 75^{th}, and maximum of the true non-zero and false non-zero coefficients selected by the models for the simulated data. Recall that the true number of non-zero coefficients is 5 and 195 extraneous coefficients that truly have a zero coefficient that could be selected by the model as a false non-zero coefficients.

Fig. 29: Boxplot of the True Non-zero Coefficients selected by each of the minimum AIC and minimum BIC Poisson longitudinal models for GMIFS and glmmLASSO when there were varying true coefficient values.

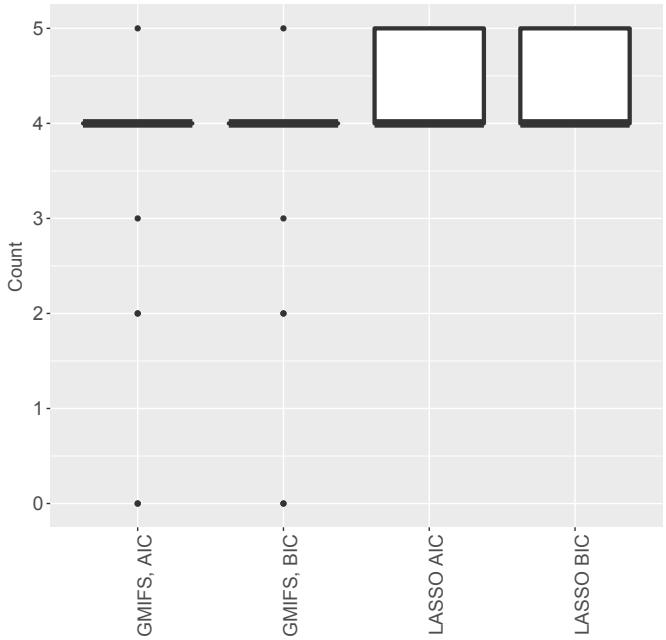
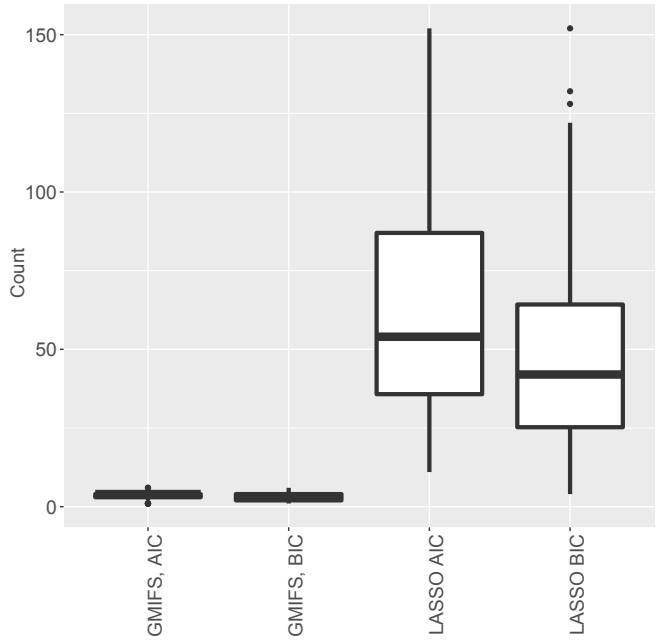


Fig. 30: Boxplot of the False Non-zero Coefficients selected by each of the minimum AIC and minimum BIC longitudinal Poisson models when there were varying true coefficient values for both the GMIFS and glmmLASSO algorithms.



3.3 Discussion

Overall, the simulation studies demonstrated that the GMIFS method is superior to the glmmLasso in weeding out false non-zero predictors. While the glmmLasso selects a larger number of the true non-zero predictors, it also includes a large percentage of the false non-zero predictors as having non-zero coefficient estimates, making the true non-zero predictor selection negligible. These conclusions held true regardless of whether $\boldsymbol{\beta}$ were of equal magnitude or varying magnitude. The GMIFS method has also been developed to handle cases when an offset term must be considered, whereas there is no such implementation in glmmLasso. In the next chapter, this method will be applied to a breast cancer application data set.

CHAPTER 4

THE LONGITUDINAL NEGATIVE BINOMIAL GENERALIZED MONOTONE INCREMENTAL STAGEWISE METHOD

4.1 VCU Health System Breast Cancer Data

Early-stage breast cancer patients (N=76) were followed at VCU Health System during the treatment of breast cancer. The breast cancer data were collected as part of a prospective study titled, "Epigenetics and Psychoneurologic Symptoms in Women with Breast Cancer" (R01NR012667)[25]. The eligibility criteria were as follows: (1) age of 21 years or older; (2) a diagnosis of early-stage breast cancer with a scheduled visit to receive chemotherapy; and (3) female gender (males were excluded because too few male participants were available for study). Exclusion criteria were as follows: (1) a previous history of cancer or chemotherapy; (2) a diagnosis of dementia; (3) active psychosis; or (4) immune-related diagnoses (e.g., multiple sclerosis; systemic lupus erythematosus)[23]. All data were collected at five different time points during the treatment of breast cancer: prior to initiating adjuvant chemotherapy but after surgery, prior to the fourth chemotherapy treatment, and at six, 12, and 24 months after the initiation of chemotherapy[23]. Collected from each patient at each time point were demographic data, symptom questionnaires, performance-based cognitive testing, and blood draws[23]. From the blood draws, methylation data, and MN and NBuds data could be obtained. Methylation data was collected using the Illumina Human-Methylation 450K assay and the cytokinesis-block micronucleus (CBMN) assay was used to score MN and NBuds. The CBMN assay has been verified and a protocol for the scoring of MN and NBuds was developed by the HUman Micronucleus (HUMN)

project[9,10].

To briefly summarize some of the key aspects of the data, the mean age of the study participants was 52 (23,71) years old, 21 women reported smoking, and 12 tumors were HER positive. The break down of stage and grade of cancer may be seen in Table 12.

Table 12.: Table of the stages of cancer and grade of cancer.

Stage of Cancer	I	II	IIIA	IIIB
	21	31	16	8
Grade	1	2	3	
	5	28	43	

DNA methylation is an epigenetic modification in human cells[5]. Research in this specific field is rapidly growing due to the increasing affordability of sequencing-based methylation analysis[5]. A CpG is a cytosine (C) connected by a phosphate (p) backbone to a guanine (G). This is occurs approximately one fifth of the expected frequency[25]. CpG sites exist in two states: methylated or unmethylated. It is known that neighboring CpG sites are correlated with respect to the methylation status, however, the exact structure and a thorough understanding of the correlation is still relatively unknown. The Illumina HumanMethylation 450K assay works by applying bisulfite conversion that converts unmethylated cytosines into uracils and methylated cytosines remain cytosines[25]. This is followed by hybridization of the sample to an array that uses beads with target-specific probes to interrogate CpG sites[5,25]. The Illumina HumanMethylation 450K assay covers 98.9% of the UCSC RefGenes with an average of 17.2 probes per gene[5,25]. When using the Illumina HumanMethylation 450K assay, the quantity computed and commonly analyzed is referred to as a beta-

value. These are defined as

$$beta - value = \frac{M}{M + U + 100} \tag{4.1}$$

where M represents the intensity of methylated alleles

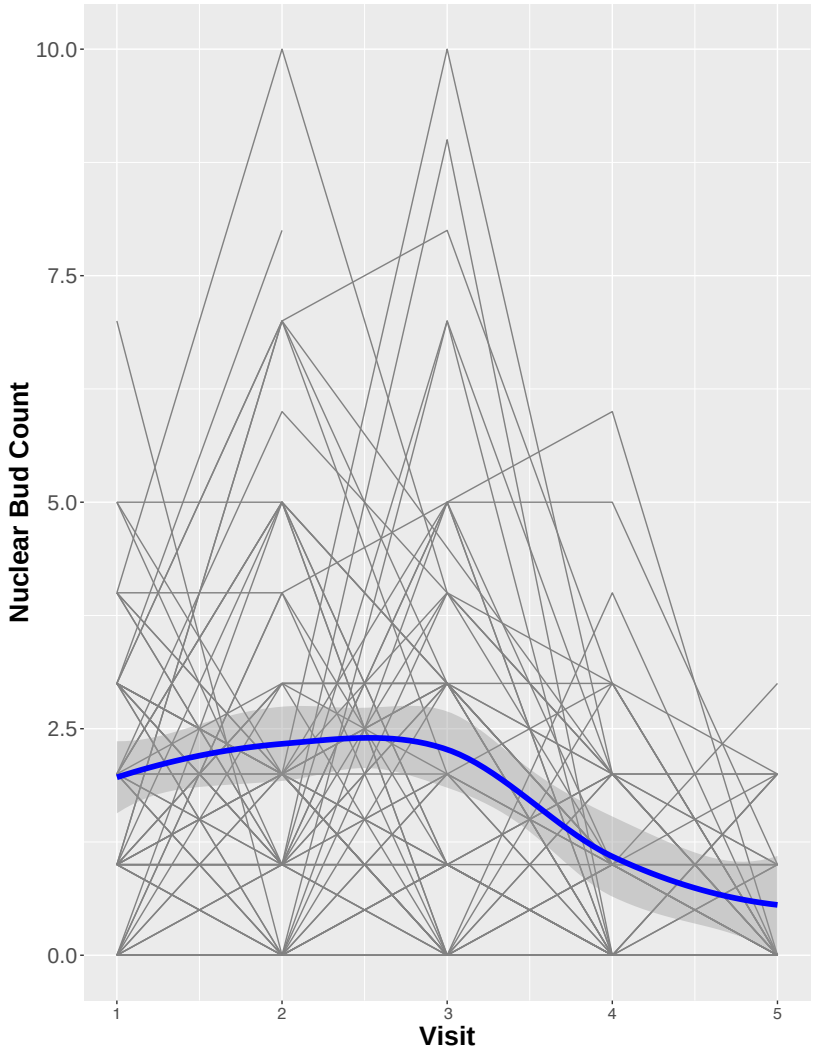
U represents the intensity of unmethylated alleles

100 represents a small positive constant to avoid dividing by 0.

The MN and NBuds data were collected using the CBMN assay previously described in Section 1.1.2. A total of 2,000 binucleated cells were scored for each patient at each time point in the study. Instead of scoring 2,000 cells at once, cells were scored in four groups of 500. The NBud and MN counts were determined by counting the number of binucleated cells with at least one NBud or MN present. The MN data were not analyzed since they follow a normal distribution[25]. For the purpose of our research on count outcomes following the Poisson and negative binomial distribution, we examined the NBud data.

Before analysis, a Boundary Likelihood Ratio test was performed to determine whether a longitudinal Poisson or longitudinal negative binomial model would be more appropriate given the early-stage breast cancer data[18]. The alternative hypothesis of the heterogeneity parameter, $\alpha \neq 0$ was tested against a null hypothesis of $\alpha = 0$. The chi-square test results were $\chi^2_1 = 42.3$ with an associated p-value of $7.97x10^{-11}$. Therefore we reject the null hypothesis that $\alpha = 0$ implying a negative binomial model is more appropriate given the data. In Figure 31 is a profile plot of the nuclear bud counts over time with a lowess fit overlay.

Fig. 31: Profile plot of the raw nuclear bud counts over time and lowess fit in royal blue.



The lowess curve exhibits higher nuclear bud counts in the beginning of the study and lower nuclear bud counts at end of the study. These results motivate the development of our longitudinal negative binomial GMIFS model which we expect to be superior to the longitudinal Poisson GMIFS model and longitudinal Poisson `glmmLasso` for the early-stage breast cancer data analysis. The goal is to better predict NBud frequency using the demographic data and methylation data.

4.2 Statistical Methods

4.2.1 Current Methods for Analyzing a Count Outcome in a Longitudinal High-dimensional Setting

Currently there are no methods that can handle an overdispersed longitudinal count outcome when you have a high-dimensional predictor space. The few methods that are applicable would be those that can handle an equidispersed longitudinal count outcome when you have a high-dimensional predictor space. These methods were described in Section 3.1.1. and 3.1.2. Recall that overdispersion occurs when the count outcome's variance is larger than the mean. It has been implied that for the longitudinal Poisson model, described in Chapter 3, inclusion of random coefficients may induce overdispersion minimally[12].

4.2.2 Proposed Extension of the Generalized Monotone Incremental Forward Stagewise Method to the Longitudinal Negative Binomial Distribution

The first aim which implemented the GMIFS method for the negative binomial model was expanded to allow for longitudinal and clustered negative binomial outcomes when there is a high-dimensional predictor space. By incorporating the ability to analyze longitudinal data, there is an additional dimension of time or clusters to the model.

Let $i = 1, ..., N$ be the number of subjects and $j = 1, ..., n_i$ be the number of observations per subject. Therefore, the total number of observations is given by $n = \sum_{i=1}^{N} n_i$. Recall, in the longitudinal setting the observations, y_{ij}, are not assumed to be independent; instead, the observations are assumed to be grouped. Let $\boldsymbol{x}$ be the full design matrix of fixed effects which are divided into penalized and unpenalized

predictors. Let $\mathbf{u}$ be the q-dimensional vector of the coefficients of the random effects, $\boldsymbol{z}$.

It has been implied that for the longitudinal Poisson model, described in Chapter 3, inclusion of random coefficients may induce overdispersion minimally[12]. Therefore, there is a heightened need for a negative binomial model in the longitudinal setting. The Poisson variability assumption can be made more flexible by adding in a subject specific and time point specific variability or error term, $\boldsymbol{e}_i$[12]. Using the log-link function the model is given by

$$\log E(y_{ij}|\boldsymbol{u}_i, e_{ij}) = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \boldsymbol{z}_i^\top \boldsymbol{u}_i + \boldsymbol{e}_i \tag{4.2}$$

If the distribution of the exponentiated additional error term, $\boldsymbol{e}_i$, is assumed to be gamma with a mean of one and variance of α, then the conditional mean of the count outcome is given by

$$\log E(y_{ij}|\boldsymbol{u}_i) = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \boldsymbol{z}_i^\top \boldsymbol{u}_i \tag{4.3}$$

The corresponding conditional variance is given by

$$\mathrm{Var}(y_{ij}|\boldsymbol{u}_i) = E(y_{ij}|\boldsymbol{u}_i) + \alpha[E(y_{ij}|\boldsymbol{u}_i)]^2 \tag{4.4}$$

Therefore, when compared to the Poisson longitudinal model, the mean is unchanged but the conditional variance is larger than the conditional mean, except when $\alpha = 0$. When approaches zero then the variance becomes equal to the mean therefore the model converges to a Poisson. α accounts for overdispersion and allows the variance to be larger than the mean. For many count outcomes, the assumption that the variance is equal to the mean is invalid. When the error term is assumed to be gamma, then the distribution of the count response is negative binomial[12]. Recall in the traditional model the negative binomial distribution does take into account overdispersion which

the Poisson distribution does not incorporate. The same applies to a longitudinal model.

By assuming a gamma distributed error, the outcome is negative binomial, which makes for easier maximum likelihood calculations based off the distribution[12]. As previously shown in Equation 1.14, the negative binomial PDF is,

$$f(\boldsymbol{y};\boldsymbol{\mu},\alpha) = \binom{y_{ij}+\frac{1}{\alpha}-1}{\frac{1}{\alpha}-1}\left(\frac{1}{1+\alpha\mu_i}\right)^{\frac{1}{\alpha}}\left(\frac{\alpha\mu_i}{1+\alpha\mu_i}\right)^{y_{ij}} \tag{4.5}$$

where α, the heterogeneity parameter, must be a positive rational value. The heterogeneity parameter accounts for the overdispersion and is inversely related to ϕ.

The likelihood function of the negative binomial mixed effects linear model that is conditioned on the normally distributed random effects is given by,

$$\begin{aligned} L(\boldsymbol{\mu},\alpha) = \int_{R^q} \prod_{i=1}^{N}\prod_{j=1}^{n_i} &\left[\binom{y_{ij}+\frac{1}{\alpha}-1}{\frac{1}{\alpha}-1}\left(\frac{1}{1+\alpha\mu_i}\right)^{\frac{1}{\alpha}}\left(\frac{\alpha\mu_i}{1+\alpha\mu_i}\right)^{y_{ij}}\right] \\ &2\pi^{-q/2}|\Sigma|^{\frac{-1}{2}}\exp\left(\frac{-1}{2}(\boldsymbol{z}_i-\boldsymbol{u}_i)^\top\Sigma^{-1}(\boldsymbol{z}_i-\boldsymbol{u}_i)\right)d\boldsymbol{u} \end{aligned} \tag{4.6}$$

$$\begin{aligned} L(\boldsymbol{\mu},\alpha) = 2\pi^{-q/2}|\Sigma|^{\frac{-1}{2}} \int_{R^q} \prod_{i=1}^{N}\prod_{j=1}^{n_i} &\left[\binom{y_{ij}+\frac{1}{\alpha}-1}{\frac{1}{\alpha}-1}\left(\frac{1}{1+\alpha\mu_i}\right)^{\frac{1}{\alpha}}\left(\frac{\alpha\mu_i}{1+\alpha\mu_i}\right)^{y_{ij}}\right] \\ &\exp\left(\frac{-1}{2}(\boldsymbol{z}_i-\boldsymbol{u}_i)^\top\Sigma^{-1}(\boldsymbol{z}_i-\boldsymbol{u}_i)\right)d\boldsymbol{u} \end{aligned} \tag{4.7}$$

The log link function may be used to re-write the conditional likelihood in terms of the predictors. The link function is given by

$$\mu_i = \exp(\boldsymbol{x}_i^\top\boldsymbol{\beta} + \boldsymbol{z}_i^\top\boldsymbol{u}_i) \tag{4.8}$$

In the GMIFS method, it is necessary to be able to take the derivative of the conditional likelihood in terms of the coefficient $\boldsymbol{\beta}$. This is used to determine which predictor should be incremented at each step of the GMIFS method. Therefore, the

marginal likelihood written in terms of the coefficients is given by

$$L(\boldsymbol{\beta}) = 2\pi^{-q/2}|\Sigma|^{\frac{-1}{2}} \int_{R^q} \prod_{i=1}^{N}\prod_{j=1}^{n_i} \left[\binom{y_{ij}+\frac{1}{\alpha}-1}{\frac{1}{\alpha}-1} \left(\frac{1}{1+\alpha\exp(\boldsymbol{x}_i^\top\boldsymbol{\beta}+\boldsymbol{z}_i^\top\boldsymbol{u})}\right)^{\frac{1}{\alpha}} \left(\frac{\alpha\exp(\boldsymbol{x}_i^\top\boldsymbol{\beta}+\boldsymbol{z}_i^\top\boldsymbol{u})}{1+\alpha\exp(\boldsymbol{x}_i^\top\boldsymbol{\beta}+\boldsymbol{z}_i^\top\boldsymbol{u})}\right)^{y_{ij}} \right] \exp\left(\frac{-1}{2}(\boldsymbol{z}_i-\boldsymbol{u}_i)^\top\Sigma^{-1}(\boldsymbol{z}_i-\boldsymbol{u}_i)\right) d\boldsymbol{u} \tag{4.9}$$

Hou and Archer, 2015 showed that it is only necessary to take the derivative of the marginal likelihood with respect to $\boldsymbol{\beta}$[20],

$$\frac{dL}{d\boldsymbol{\beta}} = \sum_{i=1}^{N}\sum_{j=1}^{n_i} \frac{\boldsymbol{x}_i(y_{ij}-\boldsymbol{x}_i^\top\boldsymbol{\beta}_i+\boldsymbol{z}_i^\top\boldsymbol{u}_i)}{1+\alpha(\boldsymbol{x}_i^\top\boldsymbol{\beta}_i+\boldsymbol{z}_i^\top\boldsymbol{u}_i)} \tag{4.10}$$

The `lme4` package in R is used to estimate the random effects[4]. Extracted from the package are the coefficient corresponding to the standard deviation of the random effects.

Again, we have a penalized and unpenalized predictor space. We divide our $\boldsymbol{\beta}$ into a new $\boldsymbol{\beta}$ which are the parameters that correspond to the penalized predictors ($\boldsymbol{x}$), $\boldsymbol{\gamma}$ which are the parameters that correspond to the unpenalized predictors ($\boldsymbol{w}$) and γ_0 which is the intercept. As previously described, unpenalized predictors are those which are forced into the model due to already known significance or knowledge to the outcome. The GMIFS method will be adapted for the longitudinal negative binomial model as follows,

1. Set the step counter, $s = 1$. Initialize the components of $\hat{\boldsymbol{\beta}}^s = 0$. Estimate α using method of moments and the intercept, γ_0, and the unpenalized coefficients, γ_j, where $j = 1, ..., J$ using the maximization algorithm of the log-likelihood. Estimate the random effects, $\hat{u}$.

2. Treating α, $\boldsymbol{\gamma}$, γ_0 and the random effects, $\hat{\boldsymbol{u}}$ as fixed find the predictor x_m such

that $m = \arg\min_k(-\frac{dl}{d\beta_k})$.

3. If $\hat{\beta}_m^s = 0$ and $step > 1$ then re-estimate $\hat{\boldsymbol{u}}$ using the new $\hat{\boldsymbol{u}}$ and α, $\boldsymbol{\gamma}$ and γ_0 find the predictor x_m such that $m = \arg\min_k(-\frac{dl}{d\beta_k})$.
   Else, $\hat{\beta}_m^s \neq 0$ proceed to step 4.

4. Update $\hat{\beta}_m^{s+1} = \hat{\beta}_m^s + \epsilon$.

5. Using the new $\boldsymbol{\beta}$ vector from step 4, update α via Hilbe's algorithm and update $\boldsymbol{\gamma}$ and γ_0 via the maximization algorithm of the log-likelihood. Step is updated to $s = s + 1$.

6. Repeat steps 2-5 until the difference between successive log-likelihoods is less than a pre-specified small tolerance, τ or until $\delta * p \geq n$.

Further, recall that an offset term is often used where there is a rate as opposed to a count outcome. The GMIFS algorithm accommodates the rate outcome through the link function.

In the implementation of this algorithm, we use $\epsilon = 0.001$, $\tau = 0.00001$, and $\delta = 0.10$. In our GMIFS algorithm, we used the `lme4` package in R to estimate the random effects. δ has been included in the stopping criteria because general sample size rules indicate that the number of predictors should maximally be approximately 10% of the sample size to prevent overparameterization, as noted by Harrell who suggested 10-20 observations per parameter (covariate) estimated to be able to detect reasonable size effects with reasonable power[21]. While this is a general rule, δ is a user defined parameter which can be changed depending on the data and application. The final model will be selected based on AIC or BIC.

4.3 Simulation Studies

Simulation studies were conducted to evaluate the performance of our method. The design of the simulation studies is based on the breast cancer application data set, which was described in Section 4.1. The longitudinal negative binomial data were simulated as follows,

1. Set α. For each of the n subjects, randomly generate an intercept from $N(0, 0.25)$.

2. Randomly generate the predictor set with P variables, $x_{i1}, x_{i2}, ..., x_{iP}$ where i= 1 to $N * n_i$ from a standard normal distribution.

3. Select a subset, P_1, of the P variables to be associated with the response. Non-zero coefficients, $\boldsymbol{\beta}$, were assigned to the P_1 of the P variables to be associated with the response. Also assign the intercept value, γ_0 and the coefficient for time, γ_1.

4. Set $\boldsymbol{Z}$ to be the $n_i * N$ x 2 design matrix of the random effects with the first column consisting of 1s and the second column to be from 0 to 4 consecutively. The second column of the random effects is going to be referred to as time.

5. Generate
$$\mu_{ij} = \exp\left(\gamma_0 + \text{time} \times \gamma_1 + \sum_{k=1}^{P_1} \beta_k x_{ik} + \sum_{l=1}^{2} u_{il} z_{ij}\right)$$

6. Randomly generate the response, $y_{ij} \sim$ negative binomial(α, μ_{ij}).

7. Repeat this to simulate r independent data sets.

The number of independent data sets simulated was $r = 100$, the sample size was set to $N = 100$ with each subject having five time points and $P = 200$ predictors plus the fixed effect, time. Of the P predictors, $P_1 = 5$ were selected to be associated with

the response. For all data sets we attempted to fit a Poisson GMIFS model and a negative binomial GMIFS model. Simulations were performed for coefficients of equal magnitude, $\boldsymbol{\beta} = (0.5, 0.5, -0.5, -0.5, -0.5)$ with an intercept value and coefficient for time of 0.5. The two methods were compared with respect to the following:

- The number of true predictors that had a non-zero coefficient estimate;
- The number of false predictors that had a non-zero coefficient estimate.

When attempting to fit the longitudinal Poisson GMIFS model, there were convergence issues when implementing functions in the `lme4` package probably due to the fact that the data are overdispersed and a negative binomial model is more appropriate. The results from the longitudinal negative binomial GMIFS models for the simulation study when $\boldsymbol{\beta} = (0.5, 0.5, -0.5, -0.5, -0.5)$ appear in Table 13 and 14. For the simulation studies when $\alpha = 0.9$, the BIC selected models are slightly more parsimonious than the AIC selected models, including a mean of 1.6 compared to 1.8 true predictors respectively. Similarly, the BIC selected models included a mean of 1.4 false predictors compared to a mean of 2.0 in the AIC selected models. For the simulation studies when $\alpha = 0.5$, the BIC selected models are slightly more parsimonious than the AIC selected models, including a mean of 2.8 compared to 3.2 true predictors respectively. Similarly, the BIC selected models included a mean of 2.1 false predictors compared to a mean of 2.8 in the AIC selected models.

Table 13.: Results from Simulation Studies when $\alpha = 0.9$: Mean/Median number of true predictors that had a nonzero coefficient estimate in the final model (True Nonzero) and the mean/median number of false predictors that had a nonzero coefficient estimate in the final model (False Nonzero). Oracle number of true non-zero coefficients, $P_1 = 5$. Oracle number of zero coefficients, $P - P_1 = 195$.

	Negative Binomial GMIFS BIC selected model	Negative Binomial GMIFS AIC selected model
True Nonzero Mean (Standard Deviation)	1.6 (1.12)	1.8 (1.17)
True Nonzero Median (Range)	1.0 (0.0, 4.0)	2.0 (0.0, 4.0)
False Nonzero Mean (Standard Deviation)	1.4 (1.39)	2.0 (1.61)
False Nonzero Median (Range)	1.0 (0.0, 5.0)	1.0 (0.0, 5.0)

Table 14.: Results from Simulation Studies when $\alpha = 0.5$: Mean/Median number of true predictors that had a nonzero coefficient estimate in the final model (True Nonzero) and the mean/median number of false predictors that had a nonzero coefficient estimate in the final model (False Nonzero). Oracle number of true non-zero coefficients, $P_1 = 5$. Oracle number of zero coefficients, $P - P_1 = 195$.

	Negative Binomial GMIFS BIC selected model	Negative Binomial GMIFS AIC selected model
True Nonzero Mean (Standard Deviation)	2.8 (1.15)	3.2 (0.99)
True Nonzero Median (Range)	3.0 (0.0, 4.0)	3.0 (0.0, 5.0)
False Nonzero Mean (Standard Deviation)	2.1 (1.36)	2.8 (1.18)
False Nonzero Median (Range)	2.0 (0.0, 6.0)	3.0 (0.0, 6.0)

While we were unable to compare the Poisson GMIFS models to the negative binomial GMIFS models, the convergence issues we encountered motivate the need for the negative binomial GMIFS model when data are genuinely overdispersed. Overall, the AIC and BIC selected negative binomial GMIFS models perform well at selecting few true predictors without also selecting out incidental false predictors.

4.4 Results

For the longitudinal breast cancer data the outcome analyzed was NBud frequency. Subjects varied in age as described in Section 4.1, therefore, age was included

in the unpenalized predictor set along with time, visits one to five. The unpenalized predictors are those that will be forced into the final model. All subjects were female so gender was an irrelevant predictor that was not included in the model. The penalized predictors were the high-dimensional methylation data. For select patients, there were records missing for up to three visits. The data were analyzed using the longitudinal negative binomial GMIFS and longitudinal Poisson GMIFS. The data were not analyzed using the `glmmLasso` method since it was shown in Section 3.2 that this method grossly overfits and includes more covariates than there are samples making the results uninterpretable.

Before analysis, the methylation data were filtered. The full data has 485,512 CpG sites. CpG sites for which all samples have beta-values $< 20\%$ are considered completely unmethylated and CpG sites for which all samples have beta-values $> 80\%$ are considered completely methylated and both can be filtered from downstream analysis[38]. After filtering those that are over 80% methylated and under 20% methylated remaining were 356,816 CpG sites.

For the longitudinal negative binomial GMIFS model, a plot of the negative log-likelihood and how it varies at each step of the GMIFS procedure may be seen in Figure 32, followed by the corresponding AIC and BIC values in Figure 33.

Fig. 32: Log-likelihood Plot for the Longitudinal Negative Binomial GMIFS.

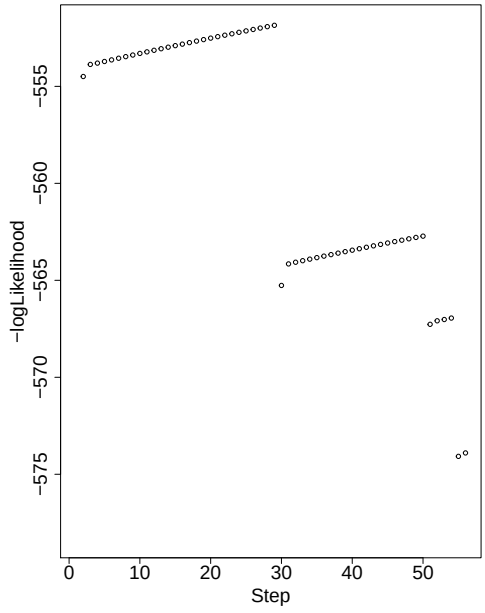
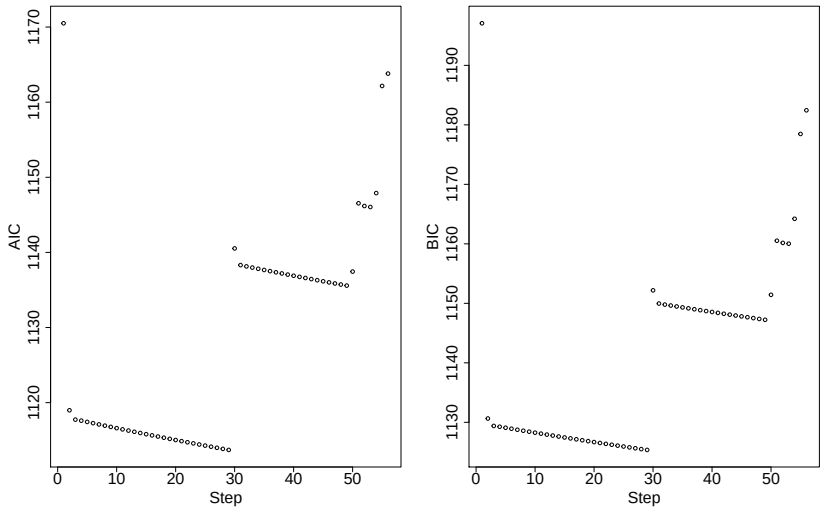


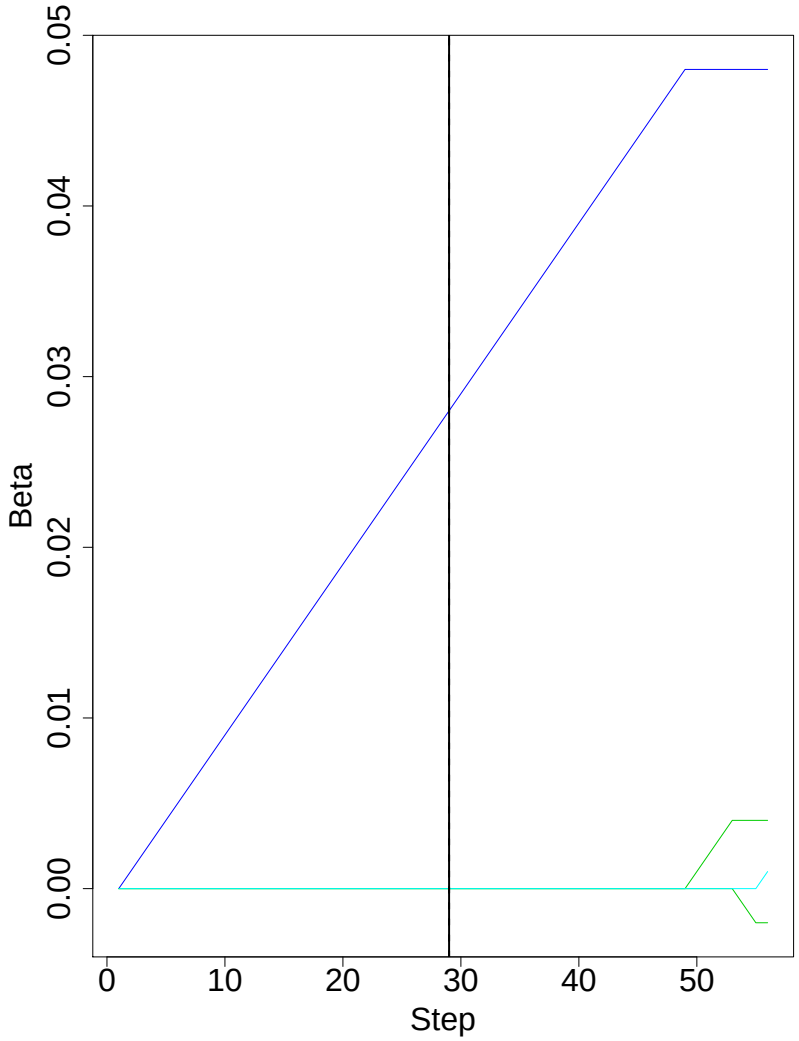
Fig. 33: AIC (left panel) and BIC (right panel) Plot for the Longitudinal Negative Binomial GMIFS.



It can be seen that the minimum BIC occurs right before step 30 as does the minimum AIC. Figure 34 shows the coefficients paths for the longitudinal negative

binomial GMIFS model. Each coefficient is represented by a different colored line such that you can see when a new coefficient enters the model.

Fig. 34: Plot of coefficient path for the longitudinal negative binomial GMIFS model with a vertical line representing when the minimum AIC and BIC is achieved.



The longitudinal negative binomial GMIFS AIC and BIC selected models identified one CpG site associated with the NBud count or one CpG site with a non-zero coefficient estimate. The methylation locus selected was cg20974885. The associated gene is ECE2; ALG3 with corresponding gene names Endothelin Converting Enzyme

2; ALG3. These are protein coding genes. Previous research by Shi et al., 2014 shows that it is associated with cancer. Due to the similarity of the predictors in the negative binomial and Poisson models, we further examined them by re-estimating the final models using traditional methods. Table 15 shows the coefficient estimates, standard error, and p-values for the final model after it was re-estimated using `glmer.nb`. The predictors in the final model included the unpenalized predictors and the one penalized selected by the GMIFS procedure, cg20974885.

Table 15.: Table coefficient estimates for the final model refit using `glmer.nb` with corresponding alpha value of 2.13.

Coefficient	Estimate	Standard Error	P-value
Intercept	0.68	0.241	< 0.001
Slope	-0.25	0.054	< 0.001
Age	0.07	0.068	0.031
cg20974885	2.29	2.119	0.028

Similarly, a longitudinal Poisson GMIFS model was fit for the breast cancer data. A plot of the negative log-likelihood and how it varies at each step of the GMIFS procedure may be seen in Figure 35, followed by the corresponding AIC and BIC values in Figure 36.

Fig. 35: Log-likelihood Plot for the Longitudinal Poisson GMIFS.

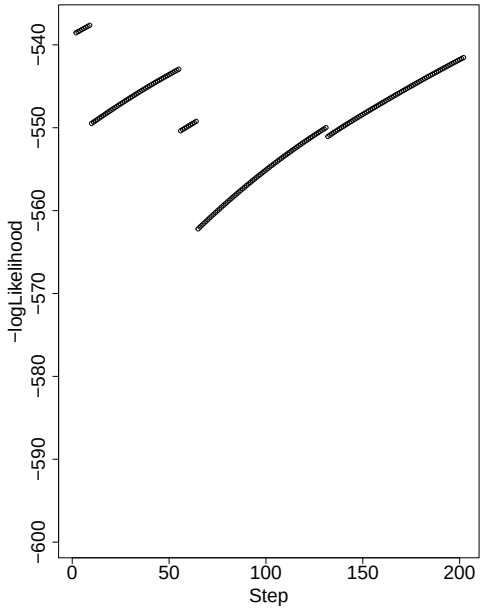
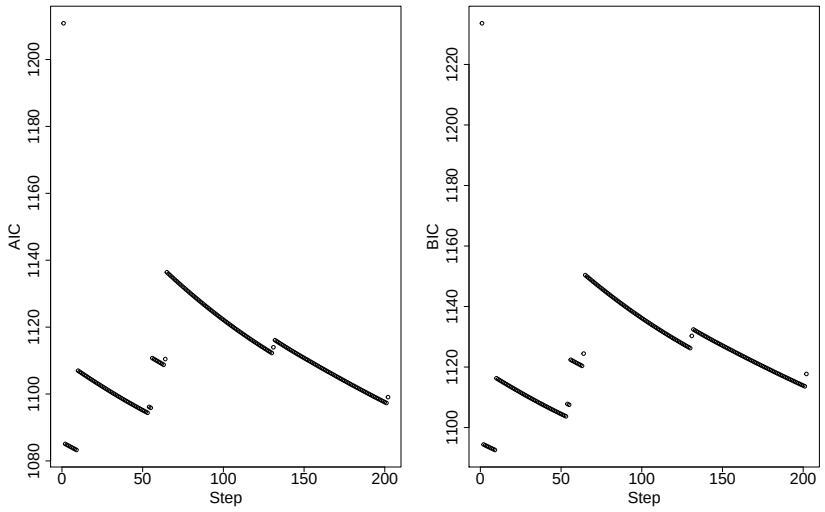


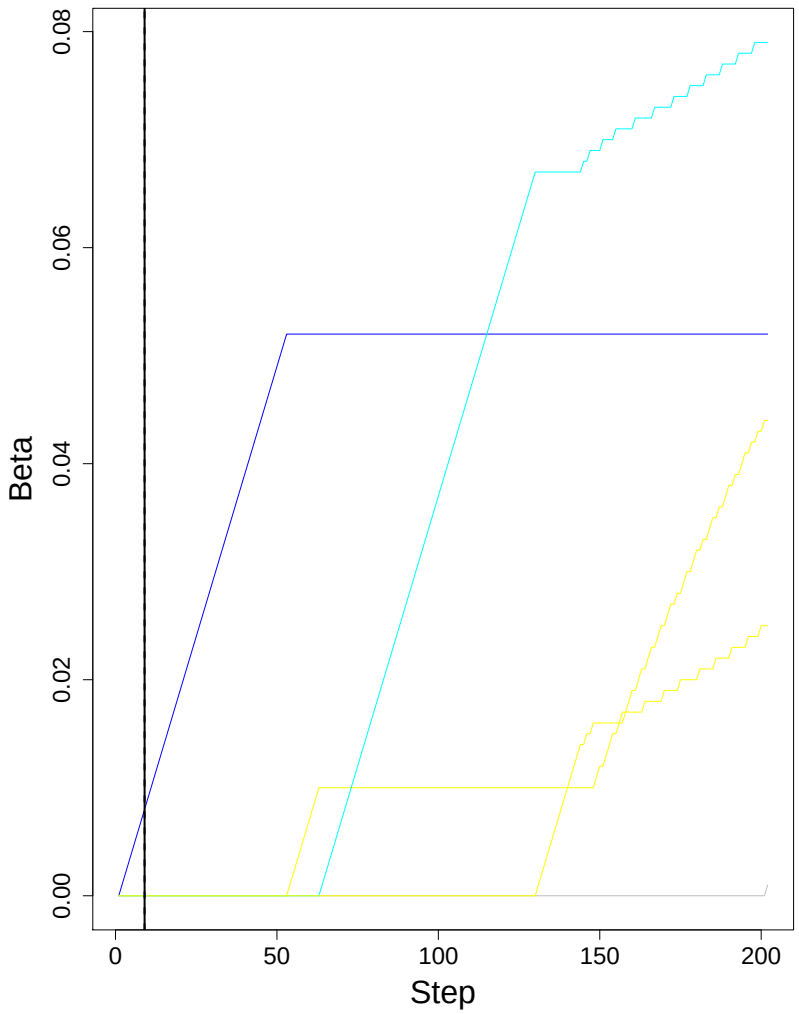
Fig. 36: AIC (left panel) and BIC (right panel) Plot for the Longitudinal Poisson GMIFS.



It can be seen that the minimum BIC occurs right before step 10 as does the minimum AIC. Figure 37 shows the coefficients paths for the longitudinal Poisson

GMIFS model. Each coefficient is represented by a different colored line such that you can see when a new coefficient enters the model.

Fig. 37: Plot of coefficient path for the longitudinal Poisson GMIFS model with a vertical line representing when the minimum AIC and BIC is achieved.



The longitudinal Poisson GMIFS AIC and BIC selected models identified one CpG site associated with the NBud count or one CpG site with a non-zero coefficient estimate. This was the same CpG site selected by the negative binomial model. The methylation locus selected was cg20974885. The associated gene is ECE2; ALG3

with corresponding gene names Endothelin Converting Enzyme 2; ALG3. These are protein coding genes. Previous research by Shi et al., 2014 shows that it associated with cancer. Table 16 shows the coefficient estimates, standard error, and p-values for the final model after it was re-estimated using `glmer`. The final model only included the unpenalized predictors and the one penalized predictor selected by the GMIFS procedure.

Table 16.: Table coefficient estimates for the final model refit using `glmer`

Coefficient	Estimate	Standard Error	P-value
Intercept	0.52	0.187	< 0.001
Slope	-0.22	0.044	< 0.001
Age	0.06	0.070	0.037
cg20974885	2.78	1.53	0.007

4.5 Discussion

We performed simulation studies to compare the longitudinal Poisson GMIFS model to the longitudinal negative binomial GMIFS model. When the simulated data followed the negative binomial distribution, the Poisson GMIFS model failed to converge. While this is a limitation of the Poisson GMIFS model it also shows the need for the negative binomial GMIFS model when data are truly overdispersed. It was concluded that the AIC and BIC selected models from the negative binomial GMIFS performed well by selecting the true predictors without also selecting extraneous false predictors.

When the longitudinal Poisson and longitudinal negative binomial GMIFS methods were applied to the breast cancer data, they selected the same covariates to be

included in the final model. It would be of interest to examine other criteria for assessing when a negative binomial model is more appropriate than a Poisson model. While it is interesting that they selected very similar models, it is also reassuring. Recall that a Poisson model is nested within the negative binomial model. While the boundary likelihood ratio test showed that there was overdispersion and a negative binomial model should be used, there might be a better test for testing this.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions from the Three Extensions to the Generalized Monotone Incremental Forward Stagewise Method

To conclude, the GMIFS method was extended to the negative binomial distribution, the longitudinal Poisson distribution, and the longitudinal negative binomial distribution. The simulation studies for the negative binomial GMIFS demonstrated the importance of accounting for overdispersion when the true underlying distribution is negative binomial. The negative binomial GMIFS had a superior fit to the Poisson GMIFS and Poisson `glmpath` in the simulation studies. The `glmpath` package had convergence issues when analyzing the negative binomial distributed data. The negative binomial GMIFS model had more specificity for removing false predictors or predictors that should not be included in the model. In addition, the negative binomial GMIFS model had analogous sensitivity to the Poisson GMIFS model for selecting true predictors. An application data set, MoBa MN counts with affiliated gene expression, were analyzed and extracted were genomic features found to be associated with elevated MN counts.

When assessing the performance of the longitudinal Poisson GMIFS model it was shown through simulation studies that there were improvements in weeding out false non-zero predictors. The alternative method, `glmmLasso`, selects a larger number of the true non-zero predictor; however, it also includes a substantial percentage of the false non-zero predictor as having non-zero coefficient estimates. Therefore, the true non-zero predictor selection is negligible.

To inspect the longitudinal negative binomial GMIFS model, we simulated longitudinal negative binomial data and attempted to compare the longitudinal Poisson GMIFS model to the longitudinal negative binomial GMIFS model. Encountered were convergence issues with the longitudinal Poisson GMIFS model when the true underlying distribution was negative binomial. Overall, the negative binomial GMIFS models performed well at selecting a large number true predictors and small number of false predictors. When the longitudinal negative binomial GMIFS model and longitudinal Poisson GMIFS model were applied to the breast cancer data set, they fit almost identical models. While the boundary likelihood ratio test suggested that a negative binomial model would be more appropriate given the data, the models were very similar.

To conclude, the developed methods are applicable when analyzing an equidispersed or overdispersed count outcome when there is a high-dimensional predictor space, both in a traditional model and longitudinal model.

5.2 Future Work

When selecting the final model from the GMIFS procedure, we used the traditional AIC and BIC. It would be of interest to investigate other criteria that can be used to select a final model. When examining the simulation study results from the negative binomial GMIFS models, it was clear that the BIC selected model overestimated α while the AIC selected model underestimated α. An alternative model selection criteria that is between AIC and BIC may be more optimal.

While the GMIFS method was predominantly used to select the best predictor set, it would be useful to look at the final predictor space and refit a traditional model for the negative binomial GMIFS method. Comparisons could be made between the biased GMIFS coefficient estimates and the more interpretable traditional model

coefficient estimates. Further extensions should be made to the zero-inflated Poisson model and zero-inflated negative binomial model. It should then be analyzed when each method is most appropriate in a high-dimensional settings.

When in the longitudinal setting, it should be further examined when a negative binomial model is more appropriate than a Poisson model. Based on our application data set, it did not seem pertinent to take into consider the overdispersion even though a boundary likelihood test showed otherwise. Alternatively, in our simulation studies when the true underlying distribution was negative binomial, the longitudinal Poisson GMIFS model would not converge, and we were forced to only examine the longitudinal negative binomial GMIFS model. Different improved methods should be developed for determining what distribution is appropriate in the data are longitudinal and there is a high-dimensional predictor space.

Finally, we plan to develop an extensive R package that can be used for analyzing count outcomes when there is a high dimensional predictor space. A separate R package will be developed for the longitudinal or clustered setting. We plan on making these packages publicly available on the Comprehensive R Archive Network.

CHAPTER 6

R CODE

6.1 Chapter 2: Negative Binomial GMIFS

6.1.1 Negative Binomial Loglikelihood Code

```
library(MASS)
# a is alpha, the overdispersion parameter (1/theta where theta from glm.nb)
# w is the set of unpenalized predictors
# x is the set of penalized predictors
# y is the discrete response
# offset is the offset
# beta are the coefficients for the penalized predictors
# theta are the coefficients for the unpenalized predictors
#################################################################
### Negative Binomial GMIFS Functions
### nb.theta is used to estimate model coefficients for unpenalized subset###
nb.theta<-function (par, a, w, x, y, offset, beta) {
  b<- par
    if (!is.null(offset)) {
      Xb <- cbind(offset, w, x) %*% c(1, b, beta)
    }
    else {
      Xb <- cbind(w, x) %*% c(b, beta)
```

```
    }
  contri.LL<- y*log((a*exp(Xb))/(1+ (a*exp(Xb)))) -
              (1/a)*log(1+ (a*exp(Xb))) +
              lgamma(y+ (1/a)) - lgamma(y+1) - lgamma(1/a)
              # likelihood fxn
  loglik <- sum(contri.LL)
  -loglik
}

```

6.1.2 Hilbe's Methods Code

```
### Hilbe's Algorithm - used to estimate alpha ###
hilbe<- function(w, y, x, theta, beta, offset, delta) {
  mu<- mean(y)                                  # estimate lambda
  chi2<- sum( ((y-mu)^2)/mu)              # Poisson chi2 test
  if(!is.null(x) & !is.null(w)) {
    df<-length(y)- dim(w)[2]-sum(beta!=0)
  }
  else if(is.null(x) & !is.null(w)) {
    df<-length(y)- dim(w)[2]
  }
  else if(!is.null(x) & is.null(w)) {
    df<-length(y)-sum(beta!=0)
  }
  disp<- chi2/df      # Poisson Dispersion
  alpha<- 1/disp     # Inverse of Poisson Dispersion
```

```
16  j<-1
17  delta_disp <- 1.0 # Initiating the change in the dispersion estimate
18  while(abs(delta_disp) >= delta) {
19    old_disp<- disp
20    if (is.null(x)) {
21      if (!is.null(offset)) {
22        Xb <- cbind(offset, w) %*% c(1, theta)
23      } else {
24        Xb <- w %*% theta
25      }
26    } else {
27      if (!is.null(offset)) {
28        Xb <- cbind(offset, w, x) %*% c(1, theta, beta)
29      } else {
30        Xb <- cbind(w, x) %*% c(theta, beta)
31      }
32    }
33    mu<- exp(Xb)
34    chi2<- ((y-mu)^2) / (mu + (alpha*(mu^2)))   # Negative Binomial Chi2 test
35    chi2<- sum ( chi2)
36    disp<- chi2/df
37    alpha<- disp* alpha
38    delta_disp<- disp- old_disp
39    j=j+1
40  }
41  alpha
```

```
}
```

6.1.3 Negative Binomial Generalized Monotone Incremental Forward Stagewise Method Code

```
nb.gmifs<-function (formula, data, x=NULL, offset, subset, epsilon=0.001,
                                    tol=1e-5, scale=TRUE, verbose=FALSE, ...) {
  mf <- match.call(expand.dots = FALSE)
  cl <- match.call()
  m <- match(c("formula", "data", "subset", "offset"), names(mf), 0L)
  mf <- mf[c(1L, m)]
  mf[[1L]] <- as.name("model.frame")
  mf <- eval(mf, parent.frame())
  mt <- attr(mf, "terms")
  y <- model.response(mf)
  w <- model.matrix(mt, mf)
  offset <- model.offset(mf)
  #### Subset code
  if (!is.null(x)) {
    if (missing(subset))
      r <- TRUE
    else {
      e <- substitute(subset)
      r <- eval( e, data)
      if (!is.logical(r))
        stop("'subset' must evaluate to logical" )
      r <- r & !is.na(r)
```

```
23      }
24      if (class(x)=="character") {
25        nl <- as.list( 1:ncol(data))
26        names(nl) <- names( data)
27        vars <- eval(substitute(x), nl, parent.frame())
28        x <- data [r , vars, drop=FALSE ]
29        x <- as.matrix(x )
30      } else if (class(x)== "matrix" || class(x)== "data.frame") {
31        x <- x[r,, drop =FALSE]
32        x <- as.matrix(x)
33      }
34    }
35    #### End subset code
36
37    if(!is.null(offset)){
38      offset<- log(offset)
39    }
40    data <- data.matrix(data)
41    n<- length(y)
42    if (!is.null(x)) {
43      vars <- dim(x)[2]
44              # vars is the number of penalized variables
45      oldx <- x
46      if (scale) {
47        x <- scale(x, center = TRUE, scale = TRUE)
48              # Center and scale the penalized variables
```

```
49      }
50      x_original<-x
51           # Keep the old x, will use in Hilbe's and estimation of theta
52      x <- cbind(x, -1 * x)
53           # x is now the expanded x matrix
54      beta <- rep(0, dim(x)[2])
55           # Beta as a vector of 0's with a length equivalent the the expanded x
56      names(beta) <- dimnames(x)[[2]]
57      step <- 1
58      Estimates <- matrix(0,ncol=vars)
59           # Estimates will be the final collapsed beta values- matrix
60      if(!is.null(offset)){
61                     initialize<-glm.nb(y~w-1 + offset(offset),
62                 control=glm.control(maxit=100))
63          # Starting values theta and (Intercept)
64      }else{
65                   initialize<-glm.nb(y~w-1,control=glm.control(maxit=100))
66            }
67      LL0 <- Likelihood <- logLik(initialize)
68           # Log-likelihood for model with no penalized predictors
69      AIC<-AIC(initialize)
70            # AIC for model with no penalized predictors
71      BIC<-BIC(initialize)
72            # BIC for model with no penalized predictors
73      theta <- coef(initialize)
74            # Unpenalized coefficient estimates for model
```

```
75          # with no penalized predictors
76      theta.update <- matrix(theta, ncol = length(theta))
77      a<- 1/theta.mm(initialize)
78          # Alpha for model with no penalized predictors,
79          # use mm estimate to initialize
80      a.update<- a
81          # a.update will be used to keep track of all alpha estimates
82      repeat {
83                      step <- 1 + step
84          # Xb will be calculated depending on whether offset is present
85          #  and whether there are penalized variables
86        if (!is.null(offset)) {
87          Xb <- cbind(offset, w, x) %*% c(1, theta, beta)
88        }
89        else {
90          Xb <- cbind(w, x) %*% c(theta, beta)
91        }
92
93        u <- t(x) %*% ((y- exp(Xb)) /(1+ (a*exp(Xb))))
94           # Likelihood gradient value- NEGATIVE BINOMIAL Hilbe Page 192
95        update.j <- which.min(-u)
96           # Choose coeffiecient to update
97        if (-u[update.j] < 0) {
98          beta[update.j] <- beta[update.j] + epsilon
99           # Update beta
100       }
```

```
101         Estimates<-rbind(Estimates,beta[1:vars]-beta[(vars+1):length(beta)])
102            # Keep track of beta changes
103         out <- optim(theta, nb.theta, a=a, w=w, x=x_original,
104                      y=y, offset=offset,
105                      beta=beta[1:vars]-beta[(vars+1):length(beta)],
106                      method="BFGS")
107             # Update intercept and non-penalized subset using new beta values
108         theta <- out$par
109         theta.update <- rbind(theta.update, theta)
110             # Keep track of theta values
111         a<- hilbe(w=w,y=y,x=x_original,theta=theta,
112             # Update alpha using Hilbe's algorithm
113                   beta=beta[1:vars]-beta[(vars+1):length(beta)],
114             # Need to use the original x not the expanded,
115             # don't want expanded beta too
116                   offset= offset, delta=1e-5)
117         a.update<- c(a.update,a)
118             # Keep track of the alpha values
119
120         p <- sum(Estimates[step,]!=0) + length(theta)        + 1
121             # Number of predictors in the NB model: nonzero beta + theta + alpha(1)
122         if (!is.null(offset)) {
123             # Calculate Xb to be used to calculate the Likelihood
124           Xb_LL <- cbind(offset, w, x_original) %*%
125                    c(1, theta, beta[1:vars]-beta[(vars+1):length(beta)])
126         }
```

```
127       else {
128         Xb_LL <- cbind(w, x_original) %*%
129                   c(theta,beta[1:vars]-beta[(vars+1):length(beta)])
130       }
131       Likelihood[step]<-LL1<- sum(y*log((a*exp(Xb_LL))/(1+ (a*exp(Xb_LL)))) -
132                         (1/a)*log(1+ (a*exp(Xb_LL))) +
133                          lgamma(y+ (1/a)) - lgamma(y+1) - lgamma(1/a))
134            # likelihood function- NEGATIVE BINOMIAL Hilbe pg 190
135       AIC[step] <- 2*p - 2*Likelihood[step]
136            # AIC - equation 5.16 Hilbe pg 68
137       BIC[step] <- p*log(n) - 2*Likelihood[step]
138            # BIC - equation 5.21 Hilbe pg 71
139       # STOPPING CRITERIA
140      if (step >= 1 && (p>=n-1 )) {
141         break
142       }
143       LL0 <- LL1
144            # Assign the "old" LL value the "new" LL value for the next step
145      }
146     output<-list(a=a.update, beta = Estimates, theta=theta.update, x=oldx,
147                  y=y, scale=scale, Likelihood=Likelihood,
148                  AIC=AIC, BIC=BIC,w=w,offset=offset)
149   } else {
150     out<-glm.nb(y~w-1, offset=offset)
151     output <- list(coef(out), a=1/out$theta)
152   }
```

```
153   class(output) <- "nb.gmifs"
154   output
155 }
156
```

6.1.4 Negative Binomial Generalized Monotone Incremental Forward Stagewise Method Functions

```
1  # ########################################################
2  # Predict function#########
3  # ############################
4  predict.nb.gmifs<- function(fit, newx, model.select=NA) {
5  #browser()
6  y<-fit$y
7  x<-newx
8  w<-fit$w
9  offset<-fit$offset
10   if (is.na(model.select)) {
11     model.select<-dim(fit$beta)[1]
12   }
13   else if (model.select == "AIC"){
14     aic<-  eval(parse(text=paste("fit",model.select,sep="$")))
15     model.select <- which.min(aic[-1])+1
16   }
17   else if (model.select == "BIC"){
18     bic<-  eval(parse(text=paste("fit",model.select,sep="$")))
19     model.select <- which.min(bic[-1])+1
```

```
20  }
21  if (dim(fit$theta)[2]==1) {
22    alpha<-fit$a[model.select]
23    beta<-fit$beta[model.select,]
24    theta<-fit$theta[model.select]
25    offset<-fit$offset
26    if (is.null(offset)) {
27      y.pred <- exp(c(theta,beta) %*% t(cbind(w, x)))
28    }
29    else {
30      offset<-fit$offset
31      y.pred <- exp(c(1,theta,beta) %*% t(cbind(offset, w, x)))
32    }
33  }
34  else {
35    alpha<-fit$a[model.select]
36    beta<-fit$beta[model.select,]
37    theta<-fit$theta[model.select,]
38    if (is.null(offset)) {
39      y.pred <- exp(c(theta,beta) %*% t(cbind(w, x)))
40    }
41    else {
42      offset<-fit$offset
43      y.pred <- exp(c(1,theta,beta) %*% t(cbind(offset, w, x)))
44    }
45  }
```

```
46 output<-list(pred=y.pred,theta=theta,beta=beta,alpha=alpha,offset=offset,w=w,x=x)
47 output
48 }
49
50 # ########################################################
51 # Coefficient function#########
52 # ##############################
53 coef.nb.gmifs<- function(fit, model.select=NA) {
54 #browser()
55 if (is.na(model.select)) {
56   model.select=dim(fit$beta)[1]
57   if (is.null(dim(fit$theta))) {
58     beta<-fit$beta[model.select,]
59     theta<-fit$theta[model.select]
60     alpha<- fit$a[model.select]
61     c.coef<-c(alpha,theta,beta)
62     names(c.coef)<- c("alpha",colnames(fit$w),colnames(fit$x))
63   }
64   else {
65     beta<-fit$beta[model.select,]
66     theta<-fit$theta[model.select,]
67     alpha<- fit$a[model.select]
68     c.coef<-c(alpha,theta,beta)
69     names(c.coef)<- c("alpha",colnames(fit$w),colnames(fit$x))
70   }
71 }
```

```
72 else if (model.select == "AIC") {
73   aic<-  eval(parse(text=paste("fit",model.select,sep="$")))
74   model.select <- which.min(aic[-1])+1
75   if (is.null(dim(fit$theta))) {
76     beta<-fit$beta[model.select,]
77     theta<-fit$theta[model.select]
78     alpha<- fit$a[model.select]
79     c.coef<-c(alpha,theta,beta)
80     names(c.coef)<- c("alpha",colnames(fit$w),colnames(fit$x))
81   }
82   else {
83     beta<-fit$beta[model.select,]
84     theta<-fit$theta[model.select,]
85     alpha<- fit$a[model.select]
86     c.coef<-c(alpha,theta,beta)
87     names(c.coef)<- c("alpha",colnames(fit$w),colnames(fit$x))
88   }
89 }
90 else if (model.select == "BIC") {
91   bic<-  eval(parse(text=paste("fit",model.select,sep="$")))
92   model.select <- which.min(bic[-1])+1
93   if (is.null(dim(fit$theta))) {
94     beta<-fit$beta[model.select,]
95     theta<-fit$theta[model.select]
96     alpha<- fit$a[model.select]
97     c.coef<-c(alpha,theta,beta)
```

```
98      names(c.coef)<- c("alpha",colnames(fit$w),colnames(fit$x))
99    }
100   else {
101     beta<-fit$beta[model.select,]
102     theta<-fit$theta[model.select,]
103     alpha<- fit$a[model.select]
104     c.coef<-c(alpha,theta,beta)
105     names(c.coef)<- c("alpha",colnames(fit$w),colnames(fit$x))
106   }
107 }
108 else if (model.select == "all") {
109   beta<-fit$beta
110   theta<-fit$theta
111   alpha<- fit$alpha
112   c.coef<-cbind(alpha,theta,beta)
113   colnames(c.coef)<- c("alpha",colnames(fit$w),colnames(fit$x))
114   rownames(c.coef)<-as.character(1:dim(beta)[1])
115 }
116
117 else {
118   if (is.null(dim(fit$theta))) {
119     beta<-fit$beta[model.select,]
120     theta<-fit$theta[model.select]
121     alpha<- fit$a[model.select]
122     c.coef<-cbind(alpha,theta,beta)
123     colnames(c.coef)<- c("alpha",colnames(fit$w),colnames(fit$x))
```

```
124     rownames(c.coef)<-as.character(1:dim(beta)[1])
125   }
126   else {
127     beta<-fit$beta[model.select,]
128     theta<-fit$theta[model.select,]
129     alpha<- fit$a[model.select]
130     c.coef<-cbind(alpha,theta,beta)
131     colnames(c.coef)<- c("alpha", colnames(fit$w),colnames(fit$x))
132     rownames(c.coef)<-as.character(1:dim(beta)[1])
133   }
134 }
135 
136 output<-list(coef=c.coef)
137 output
138 }
139 
140 # ##########################################################
141 # #summary function#########
142 # ##############################
143 summary.nb.gmifs<- function(fit, model.select=NA) {
144 #browser()
145   if (is.na(model.select)) {
146     model.select=dim(fit$beta)[1]
147     Likelihood<-fit$Likelihood[model.select]
148     AIC<- fit$AIC[model.select]
149     BIC<- fit$BIC[model.select]
```

```
150     summary<-c(Likelihood,AIC, BIC)
151     names(summary)<- c("Likelihood","AIC", "BIC")
152   }
153   else if (model.select == "AIC") {
154     aic<-  eval(parse(text=paste("fit",model.select,sep="$")))
155     model.select <- which.min(aic[-1])+1
156     Likelihood<-fit$Likelihood[model.select]
157     AIC<- fit$AIC[model.select]
158     BIC<- fit$BIC[model.select]
159     summary<-c(Likelihood,AIC, BIC)
160     names(summary)<- c("Likelihood","AIC", "BIC")
161   }
162   else if (model.select == "BIC") {
163     bic<-  eval(parse(text=paste("fit",model.select,sep="$")))
164     model.select <- which.min(bic[-1])+1
165     Likelihood<-fit$Likelihood[model.select]
166     AIC<- fit$AIC[model.select]
167     BIC<- fit$BIC[model.select]
168     summary<-c(Likelihood, AIC, BIC)
169     names(summary)<- c("Likelihood","AIC", "BIC")
170   }
171   else if (model.select=="all") {
172     Likelihood<-fit$Likelihood
173     AIC<-fit$AIC
174     BIC<- fit$BIC
175     summary<-cbind(Likelihood,AIC, BIC)
```

```
176       colnames(summary)<- c("Likelihood","AIC", "BIC")
177     }
178     else {
179       Likelihood<-fit$Likelihood[model.select]
180       AIC<-fit$AIC[model.select]
181       BIC<- fit$BIC[model.select]
182       summary<-cbind(Likelihood,AIC, BIC)
183       colnames(summary)<- c("Likelihood","AIC", "BIC")
184     }
185   output<-list(summary=summary)
186   output
187   }
188
189   ##########################################################
190   # plot function#########
191   ##############################
192   plot.nb.gmifs<- function(fit, type, main=type, beta="All") {
193     #browser()
194     if (type=="coefficients") {
195       if (beta=="All"){
196         n<-which(fit$beta[dim(fit$beta)[1],] != 0)
197         plot(1:dim(fit$beta)[1],fit$beta[,n[1]],
198           xlab="Step",ylab=expression(beta),main=main,col=500+n[1],type="l",
199           ylim=c(min(fit$beta[dim(fit$beta)[1],]),max(fit$beta[dim(fit$beta)[1],])),
200           cex.lab=2, cex.axis=2, cex.main=2)
201         for (i in 2:length(n)){
```

```
202         lines(fit$beta[,n[i]],col=500+n[i])
203       }
204     }
205     else {
206       n<-beta
207       plot(1:dim(fit$beta)[1],fit$beta[,n[1]],
208         xlab="Step",ylab="Beta",main=main,col=500+n[1],type="l",
209         ylim=c(min(fit$beta[dim(fit$beta)[1],]),max(fit$beta[dim(fit$beta)[1],])),
210         cex.lab=2, cex.axis=2, cex.main=2)
211       for (i in 2:length(n)){
212         lines(fit$beta[,n[i]],col=500+n[i])
213       }
214     }
215   }
216   else if (type == "AIC") {          # This is a plot of the AIC
217     plot(1:length(fit$AIC),fit$AIC,xlab="Step",ylab="AIC",main=main,
218          cex.lab=2, cex.axis=2, cex.main=2)
219   }
220   else if (type == "BIC") {          # This is a plot of the BIC
221     plot(1:length(fit$BIC),fit$BIC,xlab="Step",ylab="BIC",main=main,
222          cex.lab=2, cex.axis=2, cex.main=2)
223   }
224   else {  type = "Likelihood"      # This is a plot of the likelihood
225     plot(1:length(fit$Likelihood),fit$Likelihood,
226          xlab="Step",ylab="-logLikelihood",main=main,
227          cex.lab=2, cex.axis=2, cex.main=2)
```

```
  }
}
```

6.2 Chapter 3: Longitudinal Poisson GMIFS

6.2.1 Longitudinal Poisson Loglikelihood Code

```
# w is the set of unpenalized predictors
# x is the set of penalized predictors
# y is the discrete response
# z is the set of random effects
# offset is the offset
# beta are the coefficients for the penalized predictors
# theta are the coefficients for the unpenalized predictors
# u are the coefficents for the random effects
##################################################
### poisson.theta is used to estimate model coefficients for
### unpenalized subset###
### Poisson GMIFS Functions ###
poisson.theta<-function (par, w, x, y, offset, beta, zu) {
  if (!is.null(offset)) {
    Xb <- cbind(offset, w, x, zu) %*% c(1, par, beta, 1)
  }
  else {
    Xb <- cbind(w, x, zu) %*% c(par, beta, 1)
  }
  contri.LL<-y*Xb-exp(Xb)-lgamma(y+1)
  # likelihood function Poisson
```

```
  loglik <- sum(contri.LL)
  -loglik
}

```

6.2.2 Longitudinal Poisson Generalized Monotone Incremental Forward Stagewise Method Code

```
poisson.long.gmifs<-function (formula, id, slope, data,
        x=NULL, offset, subset, epsilon=0.001, tol=1e-5,
        tau=0.1,scale=TRUE, verbose=FALSE, ...) {
  mf <- match.call(expand.dots = FALSE)
  cl <- match.call()
  m <- match(c("formula", "data", "subset", "offset"), names(mf), 0L)
  mf <- mf[c(1L, m)]
  mf[[1L]] <- as.name("model.frame")
  mf <- eval(mf, parent.frame())
  mt <- attr(mf, "terms")
  y <- model.response(mf)
  w <- model.matrix(mt, mf)
  offset <- model.offset(mf)
  n<- length(unique(id))

  if (!is.null(x)) {                        # Subset code
    if (missing(subset))
      r <- TRUE
    else {
```

```
20        e <- substitute( subset)
21        r <- eval( e, data)
22        if (!is.logical(r))
23          stop("'subset' must evaluate to logical" )
24        r <- r & !is.na(r)
25      }
26      if (class(x)=="character") {
27        nl <- as.list( 1:ncol(data))
28        names(nl) <- names( data)
29        vars <- eval(substitute(x), nl, parent.frame())
30        x <- data [r , vars, drop=FALSE ]
31        x <- as.matrix(x )
32      } else if (class(x)== "matrix" || class(x)== "data.frame") {
33        x <- x[r,, drop =FALSE]
34        x <- as.matrix(x)
35      }
36    }                                              # End subset code
37    if(!is.null(offset)){
38      offset<- log(offset)
39    }
40    data <- data.matrix(data)
41    if (!is.null(x)) {
42      vars <- dim(x)[2]
43      oldx <- x
44      if (scale) {
45        x <- scale(x, center = TRUE, scale = TRUE)
```

```
46      }
47      x <- cbind(x, -1 * x)
48      beta <- rep(0, dim(x)[2])
49      names(beta) <- dimnames(x)[[2]]
50      step <- 1
51      Estimates <- matrix(0,ncol=vars)
52      beta_all<- matrix(0,ncol=2*vars)
53      initialize<-glmer(y~w-1 + (slope|id), offset=offset, family="poisson",
54                                        control=glmerControl(optimizer="bobyqa"))
55      # Initialize values
56      theta <- fixef(initialize)
57      # theta values or the unpenalized predictors
58      Likelihood <- LL0 <- logLik(initialize)[1]
59      # First log likelihood value
60      AIC <- AIC(initialize)
61      # First AIC value
62      BIC <- BIC(initialize)
63      # First BIC value
64      u<- c(rbind(ranef(initialize)$id[,1],
65                  ranef(initialize)$id[,2]))
66      i<- unique(id)
67      freq<- melt(table(id))$value
68      mat<-lapply(i, function(i) matrix(c(rep(1,freq[i]), seq(1,freq[i])),
69                                        nrow=freq[i], ncol=2))
70      z<-bdiag(mat)
71      zu<- z %*% u
```

```
72      # zu are the random effects times their coefficients, this is changing
73      theta.update <- matrix(theta, ncol = length(theta))
74      repeat {
75        step<- step+1
76        if (!is.null(offset)) {
77          Xb <- cbind(offset, w, x, zu) %*% c(1, theta, beta, 1)
78          # Added in the random effects to the log link fxn
79        }
80        else {
81          x[is.na(x)]<-0
82          Xb <- cbind(w, x, zu) %*% c(theta, beta, 1)
83          # Added in the random effects to this too
84        }
85        grad <- t(x)%*%(y-exp(Xb))
86        # Likelihood gradient value
87        update.j <- which.min(as.vector(-grad))
88        # Choose coeffiecient to update
89
90        if((step>2)&&(beta_all[step-1, update.j] - beta_all[step-2,update.j] ==0)){
91        # if this is a new beta entering the model then
92          assign("last.warning", NULL, envir = baseenv())
93          b.test<- beta[1:vars]-beta[(vars+1):length(beta)]
94          if (is.null(warnings())) {
95            initialize<-glmer(y~oldx[,b.test!=0] + w-1 + (slope|id),
96                              offset=offset, family="poisson",
97                              control=glmerControl(optimizer="bobyqa",
```

```
98                                                    optCtrl=list(maxfun=2e4)))
99          # update the random effects
100       }
101       u<- c(rbind(ranef(initialize)$id[,1],
102                   ranef(initialize)$id[,2]))
103       zu<- z %*% u
104       # zu are the random effects times their coefficients, this is changing
105       if (!is.null(offset)) {
106         Xb <- cbind(offset, w, x, zu) %*% c(1, theta, beta, 1)
107         # Added in the random effects to the log link fxn
108       }
109       else {
110         x[is.na(x)]<-0
111         Xb <- cbind(w, x, zu) %*% c(theta, beta, 1)
112         # Added in the random effects to this too
113       }
114       grad <- t(x)%*%(y-exp(Xb))
115       update.j <- which.min(as.vector(-grad))
116       # Choose coeffiecient to update
117     }
118
119     if (-grad[update.j] < 0) {
120       beta[update.j] <- beta[update.j] + epsilon
121       # Update beta
122     }
123     beta_all<- rbind(beta_all,beta)
```

```
124       Estimates<-rbind(Estimates,beta[1:vars]-beta[(vars+1):length(beta)])
125       # Keep track of beta changes
126       out <- optim(theta, poisson.theta, w=w, x=x, y=y,
127                    offset=offset, zu=zu, beta=beta,method="BFGS")
128                    # Update intercept and non-penalized subset
129       theta <- out$par
130       theta.update <- rbind(theta.update, theta)
131       # Keep track of new theta values
132       p <- sum(Estimates[step,]!=0) + length(theta)
133       Likelihood[step]<- LL1<- -out$value
134       AIC[step]<- 2*p-2*Likelihood[step]
135       BIC[step] <- p*log(n) - 2*Likelihood[step]
136       # BIC - equation 5.21 Hilbe pg 71
137       if (p>tau*n ){
138         break}
139       LL0 <- LL1
140       }
141       output<-list(beta = Estimates, theta=theta.update, x=oldx, y=y,
142                       scale=scale, Likelihood=Likelihood, AIC=AIC, BIC=BIC,
143                  w=w,offset=offset, id=id, slope=slope, u=u, z=z)
144     class(output) <- "poisson.long.gmifs"
145   } else {
146     output<-glmer(y~w-1 + (slope|id), offset=offset, family="poisson",
147                                       control= glmerControl(optimizer="bobyqa"))
148   }
149   output
```

```
}

```

6.2.3 Longitudinal Poisson Generalized Monotone Incremental Forward Stagewise Method Functions

```
predict.long.poisson.gmifs<- function(fit, newx, model.select=NA) {
#browser()
y<-fit$y
x<-fit$x
w<-fit$w
z<- fit$z
u<- fit$u
offset<-fit$offset
  if (is.na(model.select)) {
    model.select=dim(fit$beta)[1]
  }
  else if(model.select == "AIC"){
    aic<- eval(parse(text=paste("fit", model.select, sep="$")))
    model.select<- which.min(aic)
  }
  else if(model.select == "BIC"){
    bic<- eval(parse(text=paste("fit", model.select, sep="$")))
    model.select<- which.min(bic)
  }

  if (is.null(dim(fit$theta))) {
```

```
22      beta<-fit$beta[model.select,]
23      theta<-fit$theta[model.select,]
24      offset<-fit$offset
25      if (is.null(offset)) {
26        y.pred <- exp(c(theta,beta,1) %*% t(cbind(w, x, (z %*% u))))
27      }
28      else {
29        offset<-fit$offset
30        y.pred <- exp(c(1,theta,beta,1) %*% t(cbind(offset, w, x, (z %*% u))))
31      }
32    }
33    else {
34      beta<-fit$beta[model.select,]
35      theta<-fit$theta[model.select,]
36      if (is.null(offset)) {
37        y.pred <- exp(c(theta,beta,1) %*% t(cbind(w, x, (z %*% u))))
38      }
39      else {
40        offset<-fit$offset
41        y.pred <- exp(c(1,theta,beta, 1) %*% t(cbind(offset, w, x, (z %*% u))))
42      }
43    }
44  output<-list(pred=y.pred,theta=theta,beta=beta,offset=offset,w=w,x=x,z=z,u=u)
45  output
46  }
47
```

```
48
49 ##########################################################
50 #Coefficient function#########
51 ##############################
52 #NEED TO ADD ABILITY TO CHOOSE SUBSET OF BETAS
53 coef.long.poisson.gmifs<- function(fit, model.select=NA) {
54 #browser()
55 if (is.na(model.select)) {
56   model.select=dim(fit$beta)[1]
57   if (is.null(dim(fit$theta))) {
58     beta<-fit$beta[model.select,]
59     theta<-fit$theta[model.select]
60     c.coef<-c(theta,beta)
61     names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
62   }else {
63     beta<-fit$beta[model.select,]
64     theta<-fit$theta[model.select,]
65     c.coef<-c(theta,beta)
66     names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
67   }
68 }else if (model.select == "AIC") {
69   aic<-  eval(parse(text=paste("fit",model.select,sep="$")))
70   model.select <- which.min(aic)
71   if (is.null(dim(fit$theta))) {
72     beta<-fit$beta[model.select,]
73     theta<-fit$theta[model.select]
```

```
74      c.coef<-c(theta,beta)
75      names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
76    }else {
77      beta<-fit$beta[model.select,]
78      theta<-fit$theta[model.select,]
79      c.coef<-c(theta,beta)
80      names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
81    }
82  }else if (model.select == "BIC") {
83    bic<-  eval(parse(text=paste("fit",model.select,sep="$")))
84    model.select <- which.min(bic)
85    if (is.null(dim(fit$theta))) {
86      beta<-fit$beta[model.select,]
87      theta<-fit$theta[model.select]
88      c.coef<-c(theta,beta)
89      names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
90    }else {
91      beta<-fit$beta[model.select,]
92      theta<-fit$theta[model.select,]
93      c.coef<-c(theta,beta)
94      names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
95    }
96  }else if (model.select == "all") {
97    beta<-fit$beta
98    theta<-fit$theta
99    c.coef<-cbind(theta,beta)
```

```
100   colnames(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
101   rownames(c.coef)<-as.character(1:dim(beta)[1])
102 }else {
103   if (is.null(dim(fit$theta))) {
104    beta<-fit$beta[model.select,]
105   theta<-fit$theta[model.select]
106   c.coef<-c(theta,beta)
107   names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
108   }else {
109   beta<-fit$beta[model.select,]
110    theta<-fit$theta[model.select,]
111   c.coef<-c(theta,beta)
112   names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
113   }
114 }
115
116 output<-list(coef=c.coef)
117 output
118 }
119
120 ###########################################################
121 #summary function#########
122 ##############################
123
124 summary.long.poisson.gmifs<- function(fit, model.select=NA) {
125 #browser()
```

```
126 if (is.na(model.select)) {
127   model.select=dim(fit$beta)[1]
128   Likelihood<-fit$Likelihood[model.select]
129   AIC<-fit$AIC[model.select]
130   BIC<-fit$BIC[model.select]
131   summary<-c(Likelihood,AIC,BIC)
132   names(summary)<- c("Likelihood","AIC", "BIC")
133 }
134 else if (model.select == "AIC") {
135   aic<-  eval(parse(text=paste("fit",model.select,sep="$")))
136   model.select <- which.min(aic)
137   Likelihood<-fit$Likelihood[model.select]
138   AIC<-fit$AIC[model.select]
139   BIC<-fit$BIC[model.select]
140   summary<-c(Likelihood,AIC,BIC)
141   names(summary)<- c("Likelihood","AIC", "BIC")
142 }
143 else if (model.select == "BIC") {
144   bic<-  eval(parse(text=paste("fit",model.select,sep="$")))
145   model.select <- which.min(bic)
146   Likelihood<-fit$Likelihood[model.select]
147   AIC<- fit$AIC[model.select]
148   BIC<- fit$BIC[model.select]
149   summary<-c(Likelihood, AIC, BIC)
150   names(summary)<- c("Likelihood","AIC", "BIC")
151 }
```

```
152 else if (model.select=="all") {
153   Likelihood<-fit$Likelihood
154   AIC<-fit$AIC
155   BIC<- fit$BIC
156   summary<-cbind(Likelihood,AIC, BIC)
157   colnames(summary)<- c("Likelihood","AIC", "BIC")
158 }
159 else {
160   Likelihood<-fit$Likelihood[model.select]
161   AIC<-fit$AIC[model.select]
162   BIC<- fit$BIC[model.select]
163   summary<-cbind(Likelihood,AIC, BIC)
164   colnames(summary)<- c("Likelihood","AIC", "BIC")
165 }
166 output<-list(summary=summary)
167 output
168 }
169
170 ##########################################################
171 #plot function#########
172 ##############################
173
174 plot.long.poisson.gmifs<- function(fit, type, main=main,xlim=xlim,beta="All") {
175   if (type=="coefficients") {
176     if (beta=="All"){
177       n<-which(fit$beta[dim(fit$beta)[1],] != 0)
```

```
178       plot(1:dim(fit$beta)[1],fit$beta[,n[1]],xlab="Step",ylab="Beta",main=main,
179           col=500+n[1],type="l",ylim=c(min(fit$beta[dim(fit$beta)[1],]),
180           max(fit$beta[dim(fit$beta)[1],])),
181           cex=2, cex.main=2, cex.lab=2, cex.axis=1.5)
182       for (i in 2:length(n)){
183         lines(fit$beta[,n[i]],col=500+n[i])
184       }
185     }
186     else {
187       n<-beta
188       plot(1:dim(fit$beta)[1],fit$beta[,n[1]],xlab="Step",ylab="Beta",
189            main=main,col=500+n[1],type="l",
190            ylim=c(min(fit$beta[dim(fit$beta)[1],]),
191                           max(fit$beta[dim(fit$beta)[1],])),
192            cex.lab=1.5, cex.axis=1.5, cex.main=1.5)
193       for (i in 2:length(n)){
194         lines(fit$beta[,n[i]],col=500+n[i])
195       }
196     }
197   }
198   else if (type == "AIC") {
199     plot(1:length(fit$AIC),fit$AIC,xlab="Step",ylab="AIC",main=main,
200          cex.lab=2, cex.axis=2, cex.main=2)
201   }
202   else if (type == "BIC") {
203     plot(1:length(fit$BIC),fit$BIC,xlab="Step",ylab="BIC",main=main,
```

```
204           cex.lab=2, cex.axis=2, cex.main=2)
205   }
206   else {  type = "Likelihood"
207     plot(1:length(fit$Likelihood),fit$Likelihood,xlab="Step",
208           ylab="-logLikelihood",main=main,
209           cex.lab=2, cex.axis=2, cex.main=2)
210   }
211 }
```

6.3 Chapter 4: Longitudinal Negative Binomial GMIFS

6.3.1 Longitudinal Negative Binomial Loglikelihood Code

```
1  # w is the set of unpenalized predictors
2  # x is the set of penalized predictors
3  # y is the discrete response
4  # z is the set of random effects
5  # offset is the offset
6  # beta are the coefficients for the penalized predictors
7  # theta are the coefficients for the unpenalized predictors
8  # u are the coefficents for the random effects
9  # a is the alpha or the overdispersion parameter
10 ##########################################################
11 ### nb.theta is used to estimate model
12 ### coefficients for unpenalized subset###
13 ### This FXN has been edited from Matt's code to i
14 ### nclude the random effects- we will pass zu
15 ### to the function
```

```
### NB GMIFS Functions ###
nb.theta<-function (par, a, w, x, y, offset, beta, zu) {
  if (!is.null(offset)) {
    Xb <- cbind(offset, w, x, zu) %*% c(1, par, beta, 1)
  }
  else {
    Xb <- cbind(w, x, zu) %*% c(par, beta, 1)
  }
  contri.LL<- y*log((a*exp(Xb))/(1+ (a*exp(Xb)))) -
    (1/a)*log(1+ (a*exp(Xb))) +
    lgamma(y+ (1/a)) - lgamma(y+1) - lgamma(1/a)
    # likelihood function- NEGATIVE BINOMIAL Hilbe pg 190
  loglik <- sum(contri.LL)
  -loglik
}
```

6.3.2 Hilbe's Method Code

```
### Hilbe's Algorithm - used to estimate alpha ###
hilbe<- function(w, y, x, theta, beta, zu, offset, delta=.001) {
  #browser()
  x[is.na(x)]<-0
  mu<- mean(y)                          # estimate lambda
   chi2<- sum(((y-mu)^2)/mu)    # Poisson chi2 test
  if(!is.null(x) & !is.null(w)) {
    df<-length(y)- dim(w)[2]-sum(beta!=0)
  }
```

```
10  if(is.null(x) & !is.null(w)) {
11    df<-length(y)- dim(w)[2]
12  }
13  if(!is.null(x) & is.null(w)) {
14    df<-length(y)-sum(beta!=0)
15  }
16  disp<- chi2/df              # Poisson Dispersion
17  alpha<- 1/disp             # Inverse of Poisson Dispersion
18  j<-1
19  delta_disp <- 1.0
20  # Initiating the change in the dispersion estimate
21   while(abs(delta_disp) >= delta) {
22    old_disp<- disp
23     if (is.null(x)) {
24       if (!is.null(offset)) {
25         Xb <- cbind(offset, w, zu) %*% c(1, theta, 1)
26       } else {
27         Xb <- w %*% theta
28       }
29     } else {
30       if (!is.null(offset)) {
31         Xb <- cbind(offset, w, x, zu) %*% c(1, theta, beta, 1)
32       }
33       else {
34         Xb <- cbind(w, x, zu) %*% c(theta, beta, 1)
35       }
```

```
    }
    mu<- exp(Xb)
    chi2<- ((y-mu)^2) / (mu + (alpha*(mu^2)))
    # Negative Binomial Chi2 test
    chi2<- sum(chi2)
    disp<- chi2/df
    alpha<- disp*alpha
    delta_disp<- disp - old_disp
    j=j+1
  }
  alpha
}

```

6.3.3 Longitudinal Negative Binomial Generalized Monotone Incremental Forward Stagewise Method Code

```
nb.long.gmifs<-function (formula, id, slope, data,
                         x=NULL, offset, subset, epsilon=0.001,
                         tol=1e-5, tau=0.1,scale=TRUE,
                         verbose=FALSE, ...) {
  mf <- match.call(expand.dots = FALSE)
  cl <- match.call()
  m <- match(c("formula", "data", "subset", "offset"), names(mf), 0L)
  mf <- mf[c(1L, m)]
  mf[[1L]] <- as.name("model.frame")
  mf <- eval(mf, parent.frame())
```

```
11  mt <- attr(mf, "terms")
12  y <- model.response(mf)
13  w <- model.matrix(mt, mf)
14  #print(head(w))
15  offset <- model.offset(mf)
16  n<- length(unique(id))
17
18  if (!is.null(x)) { ############## Subset code
19    if (missing(subset))
20      r <- TRUE
21    else {
22      e <- substitute( subset)
23      r <- eval( e, data)
24      if (!is.logical(r))
25        stop("'subset' must evaluate to logical" )
26      r <- r & !is.na(r)
27    }
28    if (class(x)=="character") {
29      nl <- as.list( 1:ncol(data))
30      names(nl) <- names( data)
31      vars <- eval(substitute(x), nl, parent.frame())
32      x <- data [r , vars, drop=FALSE ]
33      x <- as.matrix(x )
34    } else if (class(x)== "matrix" || class(x)== "data.frame") {
35      x <- x[r,, drop =FALSE]
36      x <- as.matrix(x)
```

```
37      }
38    }               ################ End subset code
39    if(!is.null(offset)){
40      offset<- log(offset)
41    }
42    data <- data.matrix(data)
43    if (!is.null(x)) {
44      vars <- dim(x)[2]
45      oldx <- x
46      if (scale) {
47        x <- scale(x, center = TRUE, scale = TRUE)
48      }
49      x[is.na(x)] <- 0
50      x_original<-x
51      x <- cbind(x, -1 * x)
52      beta <- rep(0, dim(x)[2])
53      names(beta) <- dimnames(x)[[2]]
54      step <- 1
55      Estimates <- matrix(0,ncol=vars)
56      beta_all<- matrix(0,ncol=2*vars)
57      initialize<-glmer.nb(y~w-1 + (slope|id), data=as.data.frame(data),
58                           offset=offset, # want an intercept
59                         control=glmerControl(optimizer="bobyqa",
60                         optCtrl=list(maxfun=2e5)))
61
62      BAD<- warnings()
```

```
63      if(!is.null(BAD)) stop(paste0("Warnings at step = ", step))
64
65      theta <- fixef(initialize)
66      # theta values or the unpenalized predictors
67      Likelihood <- LL0 <- logLik(initialize)[1]
68      # First log likelihood value for the model with random effects
69      # and only penalized
70      AIC <- AIC(initialize)
71      # First AIC value for the model with random effects
72      # and only unpenalized
73      BIC <- BIC(initialize)
74      # First BIC value for the model with random effects and
75      # only unpenalized
76      a<- 1/getME(initialize, "glmer.nb.theta")
77      # Alpha for model with no penalized predictors,
78      # use mm estimate to initialize
79      a.update<- a
80      # a.update will be used to keep track of all alpha estimates
81      u<- c(rbind(ranef(initialize)$id[,1],
82                  ranef(initialize)$id[,2]))
83      i<- unique(id)
84      freq<- melt(table(id))$value
85      mat<-lapply(i, function(i) matrix(c(rep(1,freq[i]), seq(1,freq[i])),
86                      nrow=freq[i], ncol=2))
87      z<-bdiag(mat)
88      zu<- z %*% u
```

```
89      # zu are the random effects times their coefficients, this is changing
90      theta.update <- matrix(theta, ncol = length(theta))
91      repeat {
92        step<- step+1
93        if (!is.null(offset)) {
94          Xb <- cbind(offset, w, x, zu) %*% c(1, theta, beta, c(1,1))
95          # Added in the random effects to the log link fxn
96        }
97        if(is.null(offset)) {
98          Xb <- cbind(w, x, zu) %*% c(theta, beta, 1)
99          # Added in the random effects to this too
100       }
101       grad <- t(x)%*%(y-exp(Xb))
102       # Likelihood gradient value
103       update.j <- which.min(as.vector(-grad))
104       # Choose coeffiecient to update
105
106       if((step>2)&&(beta_all[step-1, update.j] - beta_all[step-2,update.j] ==0)){
107       # if this is a new beta entering the model then
108         assign("last.warning", NULL, envir = baseenv())
109         b.test<- beta[1:vars]-beta[(vars+1):length(beta)]
110         if (is.null(warnings())) {
111           initialize<-glmer.nb(y~x_original[,b.test!=0] + w-1 + (slope|id),
112                                offset=offset, data=as.data.frame(data),
113                                control=glmerControl(optimizer="bobyqa",
114                                                     optCtrl=list(maxfun=2e4)))
```

```
115            # update the random effects
116
117          BAD<- warnings()
118              if(!is.null(BAD))stop(paste0("Warnings at Re-estimating step = ",step))
119          }
120          u<- c(rbind(ranef(initialize)$id[,1],
121                      ranef(initialize)$id[,2]))
122          zu<- z %*% u
123          # zu are the random effects times their coefficients, this is changing
124          if (!is.null(offset)) {
125            Xb <- cbind(offset, w, x, zu) %*% c(1, theta, beta, c(1,1))
126            # Added in the random effects to the log link fxn
127          } else {
128            Xb <- cbind(w, x, zu) %*% c(theta, beta, 1)
129            # Added in the random effects to this too
130          }
131          grad <- t(x)%*%(y-exp(Xb))
132          # Likelihood gradient value
133          update.j <- which.min(as.vector(-grad))
134          # Choose coeffiecient to update
135        }
136
137        if (-grad[update.j] < 0) {
138          beta[update.j] <- beta[update.j] + epsilon
139          # Update beta
140        }
```

```
141         beta_all<- rbind(beta_all,beta)
142         Estimates<-rbind(Estimates,beta[1:vars]-beta[(vars+1):length(beta)])
143         # Keep track of beta changes
144         b.test<- beta[1:vars]-beta[(vars+1):length(beta)]
145         out <- optim(theta, nb.theta,a=a, w=w, x=x_original, y=y,
146                      offset=offset, zu=zu, beta=b.test, method="BFGS")
147          # Update intercept and non-penalized subset using new beta values
148
149         BAD<- warnings()
150         if(!is.null(BAD)) stop(paste0("Warnings at Optim FXN step = ", step))
151
152
153
154         theta <- out$par
155         theta.update <- rbind(theta.update, theta)
156         # Keep track of new theta values
157         a<- hilbe(w=w,y=y,x=x_original,theta=theta, zu=zu,
158         # Update alpha using Hilbe's algorithm
159                   beta=beta[1:vars]-beta[(vars+1):length(beta)],
160                   # Need to use the original x not the expanded
161                   offset= offset, delta=.001)
162         a.update<- c(a.update,a)
163         # Keep track of the alpha values
164         p <- sum(Estimates[step,]!=0) + length(theta) +1
165         # penalized, unpenalized, alpha
166         Likelihood[step]<- LL1<- -out$value
```

```
167       AIC[step]<- 2*p-2*Likelihood[step]
168       BIC[step] <- p*log(n) - 2*Likelihood[step]
169       # BIC - equation 5.21 Hilbe pg 71
170       if (p>floor(tau*n)){
171         break
172       }
173       LL0 <- LL1
174     }
175     output<-list(beta = Estimates, theta=theta.update,a=a.update,
176                  x=oldx, y=y, scale=scale, Likelihood=Likelihood, AIC=AIC, BIC=BIC,
177                  w=w,offset=offset, id=id, slope=slope, u=u, z=z)
178     class(output) <- "nb.long.gmifs"
179   } else {
180     output<-glmer.nb(y~w-1 + (slope|id), offset=offset,
181                   control= glmerControl(optimizer="bobyqa"))
182   }
183   output
184 }
```

6.3.4 Longitudinal Negative Binomail Generalized Monotone Incremental Forward Stagewise Method Functions

```
1 predict.long.nb.gmifs<- function(fit, newx, model.select=NA) {
2   #browser()
3   y<-fit$y
4   x<-fit$x
5   w<-fit$w
```

```
z<- fit$z
u<- fit$u
offset<-fit$offset
if (is.na(model.select)) {
  model.select=dim(fit$beta)[1]
}
else if(model.select == "AIC"){
  aic<- eval(parse(text=paste("fit", model.select, sep="$")))
  model.select<- which.min(aic)
}
else if(model.select == "BIC"){
  bic<- eval(parse(text=paste("fit", model.select, sep="$")))
  model.select<- which.min(bic)
}

if (is.null(dim(fit$theta))) {
  beta<-fit$beta[model.select,]
  theta<-fit$theta[model.select,]
  offset<-fit$offset
  if (is.null(offset)) {
    y.pred <- exp(c(theta,beta,1) %*% t(cbind(w, x, (z %*% u))))
  }
  else {
    offset<-fit$offset
    y.pred <- exp(c(1,theta,beta,1) %*% t(cbind(offset, w, x, (z %*% u))))
  }
```

```
  }
  else {
    beta<-fit$beta[model.select,]
    theta<-fit$theta[model.select,]
    if (is.null(offset)) {
      y.pred <- exp(c(theta,beta,1) %*% t(cbind(w, x, (z %*% u))))
    }
    else {
      offset<-fit$offset
      y.pred <- exp(c(1,theta,beta, 1) %*% t(cbind(offset, w, x, (z %*% u))))
    }
  }
  output<-list(pred=y.pred,theta=theta,beta=beta,offset=offset,w=w,x=x,z=z,u=u)
  output
}


############################################################
#Coefficient function#########
##############################
#NEED TO ADD ABILITY TO CHOOSE SUBSET OF BETAS
coef.long.nb.gmifs<- function(fit, model.select=NA) {
  #browser()
  if (is.na(model.select)) {
    model.select=dim(fit$beta)[1]
    if (is.null(dim(fit$theta))) {
```

```
58        beta<-fit$beta[model.select,]
59        theta<-fit$theta[model.select]
60        c.coef<-c(theta,beta)
61        names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
62      }
63      else {
64        beta<-fit$beta[model.select,]
65        theta<-fit$theta[model.select,]
66        c.coef<-c(theta,beta)
67        names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
68      }
69    }
70
71    else if (model.select == "AIC") {
72      aic<-  eval(parse(text=paste("fit",model.select,sep="$")))
73      model.select <- which.min(aic)
74      if (is.null(dim(fit$theta))) {
75        beta<-fit$beta[model.select,]
76        theta<-fit$theta[model.select]
77        c.coef<-c(theta,beta)
78        names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
79      }
80      else {
81        beta<-fit$beta[model.select,]
82        theta<-fit$theta[model.select,]
83        c.coef<-c(theta,beta)
```

```
84        names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
85      }
86    }
87
88    else if (model.select == "BIC") {
89      bic<-  eval(parse(text=paste("fit",model.select,sep="$")))
90      model.select <- which.min(bic)
91      if (is.null(dim(fit$theta))) {
92        beta<-fit$beta[model.select,]
93        theta<-fit$theta[model.select]
94        c.coef<-c(theta,beta)
95        names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
96      }
97      else {
98        beta<-fit$beta[model.select,]
99        theta<-fit$theta[model.select,]
100       c.coef<-c(theta,beta)
101       names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
102     }
103   }
104   else if (model.select == "all") {
105     beta<-fit$beta
106     theta<-fit$theta
107     c.coef<-cbind(theta,beta)
108     colnames(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
109     rownames(c.coef)<-as.character(1:dim(beta)[1])
```

```
110   }
111
112   else {
113     if (is.null(dim(fit$theta))) {
114       beta<-fit$beta[model.select,]
115       theta<-fit$theta[model.select]
116       c.coef<-c(theta,beta)
117       names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
118     }
119     else {
120       beta<-fit$beta[model.select,]
121       theta<-fit$theta[model.select,]
122       c.coef<-c(theta,beta)
123       names(c.coef)<- c("intercept",colnames(fit$w)[-1],colnames(fit$x))
124     }
125   }
126
127   output<-list(coef=c.coef)
128   output
129 }
130
131 ############################################################
132 #summary function#########
133 ##############################
134
135 summary.long.nb.gmifs<- function(fit, model.select=NA) {
```

```
136   #browser()
137   if (is.na(model.select)) {
138     model.select=dim(fit$beta)[1]
139     Likelihood<-fit$Likelihood[model.select]
140     AIC<-fit$AIC[model.select]
141     BIC<-fit$BIC[model.select]
142     summary<-c(Likelihood,AIC,BIC)
143     names(summary)<- c("Likelihood","AIC", "BIC")
144   }
145   else if (model.select == "AIC") {
146     aic<-  eval(parse(text=paste("fit",model.select,sep="$")))
147     model.select <- which.min(aic)
148     Likelihood<-fit$Likelihood[model.select]
149     AIC<-fit$AIC[model.select]
150     BIC<-fit$BIC[model.select]
151     summary<-c(Likelihood,AIC,BIC)
152     names(summary)<- c("Likelihood","AIC", "BIC")
153   }
154   else if (model.select == "BIC") {                                    #
155     bic<-  eval(parse(text=paste("fit",model.select,sep="$")))
156     model.select <- which.min(bic)
157     Likelihood<-fit$Likelihood[model.select]
158     AIC<- fit$AIC[model.select]
159     BIC<- fit$BIC[model.select]
160     summary<-c(Likelihood, AIC, BIC)
161     names(summary)<- c("Likelihood","AIC", "BIC")
```

```
162   }
163   else if (model.select=="all") {
164     Likelihood<-fit$Likelihood
165     AIC<-fit$AIC
166     BIC<- fit$BIC
167     summary<-cbind(Likelihood,AIC, BIC)
168     colnames(summary)<- c("Likelihood","AIC", "BIC")
169   }
170   else {
171     Likelihood<-fit$Likelihood[model.select]
172     AIC<-fit$AIC[model.select]
173     BIC<- fit$BIC[model.select]
174     summary<-cbind(Likelihood,AIC, BIC)
175     colnames(summary)<- c("Likelihood","AIC", "BIC")
176   }
177   output<-list(summary=summary)
178   output
179 }
180
181 ##########################################################
182 #plot function#########
183 ##############################
184
185 plot.long.nb.gmifs<- function(fit, type, main=main,xlim=xlim,beta="All") {
186   if (type=="coefficients") {
187     if (beta=="All"){
```

```
188       n<-which(fit$beta[dim(fit$beta)[1],] != 0)
189       plot(1:dim(fit$beta)[1],fit$beta[,n[1]],xlab="Step",ylab="Beta",main=main,
190            col=500+n[1],type="l",ylim=c(min(fit$beta[dim(fit$beta)[1],]),
191                                         max(fit$beta[dim(fit$beta)[1],])),
192            cex=1.5, cex.main=2, cex.lab=2, cex.axis=2)
193       for (i in 2:length(n)){
194         lines(fit$beta[,n[i]],col=500+n[i])
195       }
196     }
197     else {
198       n<-beta
199       plot(1:dim(fit$beta)[1],fit$beta[,n[1]],xlab="Step",ylab="Beta",
200            main=main,col=500+n[1],type="l",
201            ylim=c(min(fit$beta[dim(fit$beta)[1],]),
202                            max(fit$beta[dim(fit$beta)[1],])),
203            cex.lab=2, cex.axis=2, cex.main=2)
204       for (i in 2:length(n)){
205         lines(fit$beta[,n[i]],col=500+n[i])
206       }
207     }
208   }
209   else if (type == "AIC") {
210     plot(1:length(fit$AIC),fit$AIC,xlab="Step",ylab="AIC",main=main,
211          cex.lab=2, cex.axis=2, cex.main=2)
212   }
213   else if (type == "BIC") {
```

```
214       plot(1:length(fit$BIC),fit$BIC,xlab="Step",ylab="BIC",main=main,
215            cex.lab=2, cex.axis=2, cex.main=2)
216     }
217     else {  type = "Likelihood"
218     plot(1:length(fit$Likelihood),fit$Likelihood,xlab="Step",
219          ylab="-logLikelihood",main=main,
220          cex.lab=2, cex.axis=2, cex.main=2)
221     }
222 }
```

Appendix A

ABBREVIATIONS

VCU Virginia Commonwealth University

RVA Richmond Virginia

Appendix B

OTHER

Bibliography

[1] Alan Agresti. *Categorical Data Analysis*. Wiley, 3 edition, 2013.

[2] W.W. Au, D.M. Walker, J.B. Ward, EB. Whorton, M.S. Legator, and V. Singh. Factors contributing to chromosome damage in lymphocytes of cigarette smokers. *Mutation Research*, 1, 1991.

[3] Sadyuki Ban, Chika Konomi, Mayumi Iwakawa, Shigeru Yamada, Tatsuya Ohno, Hiroshi Tsuji, Syuhei Noda, Yoshifumi Matui, Yoshinobu Harada, John B. Cologne, and Takashi Imai. Radiosensitivity of peripheral blood lymphocytes obtained from patients with cancers of the breast, head and neck or cervix as determined with a micronucleus assay. *Journal of Radiation Research*, 45, 2004.

[4] Douglas Bates, Martin Maechler, Ben Bolker, Steven Walker, Rune Haubo Bojesen Christensen, Henrik Singmann, Bin Dai, Gabor Grothendieck, and Peter Green. *lme4: Linear Mixed-Effects Models using 'Eigen' and S4*. lme4 R package vignette, available at `https://cran.r-project.org/web/packages/glmmLasso/glmmLasso.pdf`.

[5] Marina Bibikova, Bret Barnes, Chan Tsan, Vincent Ho, Brandy Klotzle, Jennie M Le, David Delano, Lu Zhang, Gary P Schroth, Kevin L Gunderson, and et al. High density DNA methylation array with single CpG site resolution. *Genomics*, 98, 2011.

[6] Marcello Ceppi, Barbara Biasotti, Michael Fenech, and Stefano Bonassi. Human population studies with the exfoliated buccal micronucleus assay: statisti-

cal and epidemiological issues. *Cancer Epidemiology Biomarkers and Prevention*, 705(1):11–19, 2010.

[7] Ilse Decordier, Alexander Papine, Gina Plas, Sam Roesems, Kim Vande Loock, Jennifer Moreno-Palomo, Eduardo Cemeli, Diana Anderson, Aleksandra Fucic, Ricardo Marcos, Francoise Soussaline, and Micheline Kirsch-Volders. Automated image analysis of cytokinesis-block micronuclei: an adapted protocol and validated scoring procedure for biomonitoring. *Mutagenesis*, 24, 2009.

[8] Michael Fenech. The cytokinesis-block micronucleus technique: a detailed description of the method and its application to genotoxicity studies in human populations. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 285(1):35–44, 1993.

[9] Michael Fenech, Wushou P. Chang, Micheline Kirch-Volders, Nina Holland, Stefano Bonassi, and Errol Zeiger. HUMN project: detailed description of the scoring criteria for the cytokinesis-block micronucleus assay using isolated human lymphocyte cultures. *Mutation Research/Genetic Toxicology and Environmental Mutagenesis*, 534(1):65–75, 2003.

[10] Michael Fenech, Nina Holland, Wushou P. Chang, Errol Zeiger, and Stefano Bonassi. The HUman Micronucleus Project- an international collaborative study on the use of the micronucleus technique for measuring DNA damage in humans. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 428(1):271–283, 1999.

[11] Michael Fenech, Micheline Kirsch-Volders, H Natarajan, J Surralles, Crott, Parry, Norppa, Eastmond, Tucker, and Thomas. Molecular mechanisms of mi-

cronucleus, nucleoplasmic bridge and nuclear bud formation in mammalian and human cells. *Mutagenesis*, 26(1):125–132, 2011.

[12] Garrett M. Fitzmaurice, Nan M. Laird, and James H. Ware. *Applied Longitudinal Analysis*. John Wiley Sons, Inc., 2 edition, 2011.

[13] Jerome Friedman, Trevor Hastie, Noah Simon, and Rob Tibshirani. *glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models*. glmnet R package vignette, available at `https://cran.r-project.org/web/packages/glmnet/index.html`.

[14] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, (33(1)):1–22, 2010.

[15] Andreas Groll. *glmmLasso: Variable Selection for Generalized Linear Mixed Models by L1-Penalized Estimation*. glmmLasso R package vignette, available at `https://cran.r-project.org/web/packages/glmmLasso/glmmLasso.pdf`.

[16] S Gutirreza, E Carbonella, P Galofrb, A Creusa, and R Marcos. Micronuclei induction by I exposure: Study in hyperthyroidism patients. *Mutation Research*, 373, 1997.

[17] Trevor Hastie, Jonathan Taylor, Robert Tibshirani, and Guenther Walther. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1:1–29, 2007.

[18] Joseph M. Hilbe. *Negative Binomial Regression*. Cambridge University Press, 2 edition, 2010.

[19] Kevin Hochstenbach, Danitsja M. van Leeuwen, Hans Gmuender, Ralf W. Gottschalk, Martinus Lovik, Berit Granum, Unni Nygaard, Ellen Namork, Micheline Kirsch-Volders, Ilse Decordier, Kim Vande Loock, Harrie Besselink, Margareta Tornqvist, Hans von Stedingk, Per Rydberg, Jos C.S. Kleinjans, Hek van Loveren, and Joost H.M. van Delft. Global gene expression analysis in cord blood reveals gender specific differences in response to carcinogenic exposure in utero. *Mutation Research/Reviews in Mutation Research*, 21(10):1756–1767, 2012.

[20] Jiayi Hou and Kellie J. Archer. Regularization method for predicting an ordinal response using longitudinal high-dimensional genomic data. *Statistical Applications in Genetics and Molecular Biology*, 14(1):93–111, 2015.

[21] Frank E. Harrell Jr. *Regression Modeling Strategies With Applications to Linear Models, Logistic Regression, and Survival Analysis*. Springer, 1 edition, 2001.

[22] Micheline Kirsch-Volders, Gina Plas, Azeddine Elhajouji, Magdalena Lukamowicz, Laetitia Gonzalez, Kim Vande Loock, and Ilse Decordier. The in vitro MN assay in 2011: origin and fate, biological significance, protocols, high throughput methodologies and toxicological relevance. *Archives of Toxicology*, (85(8)):873–899, 2011.

[23] Debra E. Lyon, Ronald Cohen, Huaihou Chen, Debra L. Kelly, Angela Starkweather, Hyo-Chol Ahn, and Colleen K. Jackson-Cook. The relationship of cognitive performance to concurrent symptoms, cancer- and cancer-treatment-related variables in women with early-stage breast cancer: a 2-year longitudinal study. *Journal of Cancer Research and Clinical Oncology*, 142, 2016.

[24] Per Magnus, Lorentz M Irgens, Kjell Haug, Wenche Nystad, Rolv Skjaerven,

Camilla Stoltenberg, and The Moba Study Group. Cohort profile: The Norwegian Mother and Child Cohort Study (MoBa). *International Journal of Epidemiology*, 35, 2006.

[25] Mateusz Makowski and Kellie J Archer. Generalized monotone incremental forward stagewise method for modeling count data: Application predicting micronuclei frequency. *Cancer Informatics*, 14(2):97–105, 2015.

[26] L. Migliore, M. Parrini, I. Sbrana, C. Biagini, A. Battaglia, and N. Loprieno. Micronucleated lymphocytes in people occupationally exposed to potential environmental contaminants: the age effect. *Mutation Research*, 1, 1991.

[27] Renato Minozzo, Luiz Irineu Deimling, Luciana Petrucci Gigante, and Renato Santos-Mello. Micronuclei in peripheral blood lymphocytes of workers exposed to lead. *Mutation Research*, 565, 2004.

[28] Mee Young Park and Trevor Hastie. *glmpath: L1 Regularization Path for Generalized Linear Models and Cox Proportional Hazards Model*. glmpath R package vignette, available at `https://cran.r-project.org/web/packages/glmpath/glmpath.pdf`.

[29] Mee Young Park and Trevor Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007.

[30] Jurg Schelldorfer, Peter Buhlmann, and Sara van de Geer. Estimation for high-dimensional linear mixed-effects models using l1-penalization. *Scandinavian Journal of Statistics*, 38(2):197–214, 2011.

[31] Jurg Schelldorfer, Lukas Meier, and Peter Buhlmann. GLMMLasso: An

algorithm for high-dimensional generalized linear mixed models using L1-penalization. *Journal of Computational and Graphical Statistics*, 23(2):460–477, 2014.

[32] Wan Tang, Hua He, and Xin M. Tu. *Applied Categorical and Count Data Analysis*. Taylor Francis Group, LLC, 1 edition, 2012.

[33] Robert Tibshirani. Regression shrinkage and selection via lasso: a retrospective. *Journal of the Royal Statistical Society:Series B (Statistical Methodology)*, 73, 2011.

[34] R. Tomanin, C. Ballarin, B. Nardini, G. Mastrangelo, and F. Sarto. Influence of smoking habit on the frequency of micronuclei in human lymphocytes by the cytokinesis-block method. *Mutagenesis*, 6, 1991.

[35] Gerhard Tutz and Andreas Groll. Variable selection for generalized linear mixed models by L1-penalized estimation. *Statistics and Computing*, 2014.

[36] Dominic Varga, Josef Hoegel, Christiane Maier, Silke Jainta, Maren Hoehne, Brenda Patino-Garcia, Isabell Michel, Ulrike Schwarz-Boeger, Marion Kiechle, Rolf Kreienberg, and Walther Vogel. On the difference of micronucleus frequencies in peripheral blood lymphocytes between breast cancer patients and controls. *Mutagenesis*, 21, 2006.

[37] Janice W. Yager, Marja Sorsa, and Steve Selvin. Micronuclei in cytokinesis-blocked lymphocytes as an index of occupational exposure to alkylating cytostatic drugs. *IARC Scientific Publications*, 1, 1988.

[38] James Zou, Christoph Lippert, David Heckerman, Martin Aryee, and Jennifer

Listgarten. Epigenome-wide association studies without the need for cell-type composition. *Nature Methods*, 2014.

VITA

Rebecca Ruffin Lehman was born on May 24, 1990 in Richmond, VA. She received her Bachelor of Science degree in May of 2012 from The University of the South in Sewanee, TN. She began work towards a PhD in Biostatistics at Virginia Commonwealth University in August of 2012. While pursuing her PhD, she has worked as a Teaching Assistant for the Department of Biostatistics, a Statistical Consultant for Technology Services, a Graduate Research Assistant in the Department of Family Medicine, and a Trainee on the NIH Training Grant 5T32ES007334-13.