Signal Detection of Adverse Drug Reaction using the Adverse Event Reporting System: Literature Review and Novel Methods

by

Minh H. Pham

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Arts in Statistics
Department of Mathematics and Statistics
College of Arts and Sciences
University of South Florida

Major Professor: Kandethody M. Ramachandran, Ph.D.
Feng Cheng, Ph.D.
Chris P. Tsokos, Ph.D.

Date of Approval:

Keywords: Signal Detection, Association Study,
Pharmacovigilance, Statistical Algorithms, Big Data.

DEDICATION

To my parents: for their unconditional love, support, and sacrifices. You taught me well the value of discipline and hard work.

To my sister: for your encouragement and support. You have always encouraged me to pursue what I love.

To Won Yi and Malko Cajuste: for your guidance and care. You have made me a more well-rounded person.

ACKNOLEDGEMENT

I would like to gratefully acknowledge grant 7AZ23 from the Ed and Ethel Moore Alzheimer's Disease Research Program, Florida Department of Health. The grant allowed us to combine all the FAERS data submissions and store in a local database at the University of South Florida. The database was used extensively in this thesis.

I owe my deepest gratitude to my thesis advisor Dr. Kandethody Ramachandran for accepting to chair my thesis. Thank you for advising my work, opening my mind, and challenging me to do better.

I would like to thank Dr. Feng Cheng for suggesting the problem and serving on my thesis committee. Thank you forgiving insights and directions to my thesis work.

I would like to thank Dr. Chris Tsokos for serving on my thesis committee. Thank you for your time and valuable feedback on my thesis work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

One of the objectives of the U.S. Food and Drug Administration is to protect the public health through post-marketing drug safety surveillance, also known as Pharmacovigilance. An inexpensive and efficient method to inspect post-marketing drug safety is to use data mining algorithms on electronic health records to discover associations between drugs and adverse events.

The purpose of this study is two-fold. First, we review the methods and algorithms proposed in the literature for identifying association drug interactions to an adverse event and discuss their advantages and drawbacks. Second, we attempt to adapt some novel methods that have been used in comparable problems such as the genome-wide association studies and the market-basket problems. Most of the common methods in the drug-adverse event problem have univariate structure and thus are vulnerable to give false positive when certain drugs are usually co-prescribed. Therefore, we will study applicability of multivariate methods in the literature such as Logistic Regression and Regression-adjusted Gamma-Poisson Shrinkage Model for the association studies. We also adopted Random Forest and Monte Carlo Logic Regression from the genome-wide association study to our problem because of their ability to detect inherent interactions. We have built a computer program for the Regression-adjusted Gamma Poisson Shrinkage model, which was proposed by DuMouchel in 2013 but has not been made available in any public software package. A comparison study between popular methods and the proposed new methods is presented in this study.

CHAPTER I: INTRODUCTION AND PROBLEM STATEMENT

In order to monitor adverse events of drugs that have been approved for marketing, the US Food and Drug Administration (FDA) has organized the FDA Adverse Event Reporting System (FAERS) since 1968 [1]. FAERS is a rich data source for the study and identification of adverse reactions to regulated drugs in the US. This database contains over 2 million voluntary reports of pharmaceutical products in the world and increases by more than 300,000 reports each year [2, 3]. For the past four decades, the FAERS database have played a major role in signaling known and unknown adverse events that are associated with single or interacted drugs. If a potential safety concern is discovered through FAERS, the FDA then performs other evaluations and might take regulatory actions to protect the public health, such as restricting the signaled drug, updating information labels, or removing the product from the market [1].

Despite its critical role in Pharmacovigilance, the FAERS database has limitations and presents challenging problems to data scientists in designing statistical processes and algorithm to detect safety signals. First, safety signals, even correct and significant signals, do not always present cause-effect relationship between drugs and adverse events because according to the FDA's requirements for data collection, the relationship between reported adverse event and drug are not necessarily proven to be causal-effect. Second, since patients and their service providers may independently report the same adverse events to the database, duplicated reports are possible and is in fact a well-known problem for the FAERS database [1]. In order to tackle the duplicated report problem, researchers usually take into account the case versions and discrepancies between FAERS and the FDA's legacy data [34]. Finally, the gigantic and rapidly increasing size of FAERS (more than 1 million records of prescribed drugs added every quarter [1]) creates challenges in computational statistics, resolving event and drug dictionary problems and data miscoding [18].

The study of drug-adverse event association problem is a fairly new problem in the literature. The first systematic studies that addressed this specific problem were carried out in the early 2000s [2 – 4, 11, 12]. However, the literature has progressed quickly because of its similarities with other problems such as the market basket problem [6-10] and the genome-wide association problem [4, 5]. In the market basket problem, researchers attempt to identify patterns of the type “A customer purchasing item A is likely to purchase item B”. In the genome-wide association problem, we find associations between genomic patterns and diseases or traits.

The drug-adverse event problem could be mathematically stated as follows. Given a set of drugs $X_1, X_2, \dots, X_p$ and a set of adverse events $Y_1, Y_2, \dots, Y_q$, the objective is to find the set of drugs that associates with a specific adverse event $Y_h, 1 \leq h \leq q$. Mathematically speaking, we would like to generate all sets that contain one or more drugs and one adverse event, $(X_i, X_j, \dots X_k, Y_h), 1 \leq i, j, k \leq p$, that have significant association measures between event Y_h and drug(s) $X_i, X_j, \dots, X_k$. Various measures of association have been proposed by researchers in the literature such as Proportional Reporting Ratio [11], Reporting Odds Ratio [13], Relative Risk [20], and Information Component [32]. If a set has only one X, the drug is called associated with event Y_h. Two or more X’s indicate drug interactions that created the adverse event.

The remainder of this thesis is organized in 4 chapters. Chapter 2 presents the notable statistical tests and algorithms and the survey of research related to the problem being addressed. In Chapter 3, we discuss the Random Forest algorithm and Monte Carlo Logic Regression that we introduced for drug association studies because they have interesting properties that might tackle the challenges. In Chapter 4, we perform a comparison study between the commonly used data mining methods and the novel methods using the Observational Medical Outcomes Partnership's Gold Standard as a testing bed. The concluding remarks and the suggested future work are presented in Chapter 5.

CHAPTER 2: LITERATURE REVIEW

The following notations are used to describe the methodologies. Suppose our data has n rows corresponding to n cases. Variables $X_1, X_2, \dots, X_p$ indicate use of drugs (1 means used, 0 means not used). Variables $Y_1, Y_2, \dots, Y_q$ indicate presence of q adverse events. Variables $Z_1, Z_2, \dots, Z_r$ contain demographic information, such as age and gender. The data's dimension is n*(p + q + r).

For the remainder of this thesis, we use $X_{i,j}$ to denote the ith column and jth row entry of matrix X. Therefore, $X_{i,l}, Y_{j,l}, Z_{k,l}$ denote the values of each variable at the *l*th case in the data where $X_{i,l}$ and $Y_{j,l}$ take value of 0 or 1 for all *i, j, l*. For instance, $X_{3,10} = 1$ means that the patient in the 10th row took drug X_3, $Y_{5,20} = 1$ means that the patient in the 20th row observed adverse event Y_5.

2.1 Association Rules

Association Rules was introduced for the market basket problem by Agrawal et al. in 1993, [6].

Let $N_i = \sum_{l=1}^{n} X_{il}$ be the count of rows that observe the use of drug X_i (*1* $\leq i \leq p$ and *1* $\leq l \leq n$ is the index for cases), $N_{ij} = \sum_{l=1}^{n} X_{il} Y_{jl}$ be count of rows that observe both drug $X_i = 1$ and adverse event $Y_j = 1$ (*1* $\leq j \leq q$). Association Rules uses *confidence* as a measure of interestingness, which is the probability of observing adverse event Y_j given X_i is present $P(Y_j = 1 | X_i = 1) = \frac{N_{ij}}{N_i}$. The method is conducted through 2 steps:

- *Support* is the proportion of data that observe both $X_i = 1$ and $Y_j = 1$. This proportion is $\frac{N_{ij}}{n}$. Select all sets of X_i and Y_j that have *support* higher than an arbitrary threshold: $\frac{N_{ij}}{n} \geq S_0$
- From the sets found in the previous step, identify the sets that have *confidence* higher than an arbitrary threshold: $P(Y_j = 1|X_i = 1) = \frac{N_{ij}}{N_i} \geq C_0$

There is no definitive way to determine the thresholds S_0 and C_0. The choice of thresholds is subject to the context of the data set and how interesting the associations are [33].

Finding association between three or more items is done in similar fashion, where *support* is the proportion of records that observe all of X_i, $X_{i'}$, and Y_j in the data and *confidence* is the probability of observing event Y_j given both $X_i$and $X_{i'}$ is present. More specifically, the two steps Association Rules are now:

- Select all sets of X_i and Y_j that have *support*, which is the percentage of observing all of X_i, $X_{i'}$, and Y_j in the data $\frac{N_{ii'j}}{n}$ where $N_{ii'j} = \sum_{l=1}^{n} X_{il} X_{i'l} Y_{jl}$, higher than an arbitrary threshold: $\frac{N_{ii'j}}{n} \geq S_0$
- From the sets found in the previous step, identify the sets that have *confidence* higher than an arbitrary threshold: $P(Y_j = 1|X_i = 1 \,\&\, X_{i'} = 1) = \frac{N_{ii'j}}{N_i} \geq C_0$

In real-world practice, it is common that the number of drugs *p* and the number of events *q* are so large that we cannot consider all combinations of drug-adverse event because generating and evaluating all combinations is computationally intensive. Only considering combinations of one drug and one event, the total number of combinations we need to consider is $p \times q$, which can be immensely big if the dataset has thousands of drugs and events. Algorithms such as Apriori [7] or FP-growth [8] are designed to finish the first step efficiently by reducing the number of item sets that we must consider. Apriori algorithm does this by eliminating an item set if any of its subset does not have enough support. FP-tree compresses data into a tree structure where frequent item sets lay on top of the tree and can easily be found.

Advantages of Association Rules:

Being one of the first methods to be proposed in the association study literature, Association Rule is intuitive and easy to implement. This method is also computationally less intensive than the later ones because all computational operations include only summing and logical comparisons.

Drawbacks of Association Rules:

The simple operation does not make statistical soundness in many cases because it does not adjust for the popularity of individual drug or correlation. Brin & Motwani [9] gives the following example to illustrate its weakness. Consider drug X_1 and adverse event Y_2 with the total number of records n = 100, N_1 = 25 records have $X_1 = 1$, N_2 = 90 records have $Y_2 = 1$, N_{12} = 20 records have $X_1 = 1 \; and \; Y_2 = 1$, and 5 records have $X_1 = 0$ and $Y_2 = 0$.

The percentage of records having $X_1 = 1 \; and \; Y_2 = 1$ is:

$$support = \frac{20}{100} = 0.2, or \; 20\%$$

The percentage of records having $Y_2 = 1$, given $X_1 = 1$ is:

$$confidence = \frac{20}{25} = 0.8, or \; 80\%$$

Suppose a researcher sets the threshold $S_0 = 10\%$ for support and $C_0 = 70\%$ for confidence, Association Rules will determine that the association between X_1 and Y_2 as significant. However, considering that adverse event Y_2 is very popular (90%), the use of drug X_1 actually decreases the adverse event rate from 90% to 80%. Because of situations like this, Association Rules is well-known for detecting false associations, also known as spurious associations.

Another weakness shows up when we apply this method to data sets with huge number of items (big p). The data may be so big that most item sets have tiny support and hence cannot pass the support threshold S_0. For instance, in a database with a total of 20 million records, there are 200 records with $X_1 = 1\ and\ Y_2 = 1$, then $support = \frac{200}{20{,}000{,}000} = 0.00001, or$ 0.001%. This can easily fail any arbitrary support threshold S_0. This is the case for our FDA data where we have over 17 million records of drug and over 14 million records of adverse events.

In order to tackle the spurious association problem, other methods such as Gamma-Poisson Shrinkage Model, Proportional Reporting Ratio, and Reporting Odds Ratio were proposed.

2.2 Collective Strength

As an attempt to solve the spurious association problem of Association Rules that was discussed in section 2.1, Aggarwal and Yu proposed a new measure of association, Collective Strength [10].

Let I be an item set of drug(s) and an adverse events, $I = (X_i, X_j, \dots X_k, Y_h), 1 \le i, j, k \le p$. Aggarwal and Yu defined *violation* v(I) of an item set I as the sets containing some but not all items of I. Suppose we are evaluating drug X_i and adverse event Y_j, $I = (X_i, Y_j)$ is the event of using drug X_i and observing adverse event Y_j. The violation $v(I)$ is the event of observing either $X_1 = 1$ or $Y_1 = 1$, but not both: $v(I) = \left(X_i = 1 \; and \; Y_j = 0\right) \; or \; \left(X_i = 0 \; and \; Y_j = 1\right)$.

We can then estimate the probability of violation event from the data: $P\left(v(I)\right) = \frac{\sum_{l=1}^{n} I\left(X_{il}=1 \; and \; Y_{jl}=0\right) \; or \; \left(X_{il}=0 \; and \; Y_{jl}=1\right)}{n}$.

Collective Strength is then defined as: $C(I) = \frac{1-P(v(I))}{1-E(P(v(I)))} * \frac{E(P(v(I)))}{P(v(I))}$, $0 \leq C(I) \leq \infty$, where $E(P\left(v(I)\right))$ is calculated by assuming the independence of items and using raw probabilities of individual items. In our notations, $E(P(v(I))) = 1 - P(X_i = 1)P\left(Y_j = 1\right) - P(X_i = 0)P\left(Y_j = 0\right)$, where $P(X_i = 1), \; P\left(Y_j = 1\right), P(X_i = 0), P\left(Y_j = 0\right)$ are estimated from the data as follows.

$$P(X_i = 1) = \frac{\sum_{l=1}^{n} X_{il}}{n}$$

$$P(Y_i = 1) = \frac{\sum_{l=1}^{n} Y_{jl}}{n}$$

$$P(X_i = 0) = 1 - P(X_i = 1)$$

$$P(Y_i = 0) = 1 - P(Y_i = 1)$$

Collective Strength $C(I)$ can take any value from 0 to infinity. A value of 0 indicates perfectly negative correlation between X_i and Y_j, i.e. $Y_j = 0$ when $X_i = 1$ and vice versa. $C(I) = 1$ indicates no association between X_i and Y_j . The more $C(I)$ exceeds 1, the stronger the association between X_i and Y_j.

Advantages of Collective Strength:

The authors proved that Collective Strength does not suffer from detecting false positive because it considers the presence/absence of individual items. In addition, it has nice computational properties that allow the setup of algorithms that works as efficiently as Association Rules for large number of items.

Drawbacks of Collective Strength:

The convenient computational properties come with the price of loss of interpretability as a measure of association, since the formula of Collective Strength does not suggest any useful meaning. Compared to other measures of association described later such as Relative Reporting Rate, Proportional Reporting Rate, or Reporting Odds Ratio, Collective Strength is a lot less intuitive.

To illustrate this weakness, let's consider an item set I = $\{X_1, Y_1\}$, where the probability of observing each item is 0.1: $P(X_1 = 1) = P(Y_1 = 1) = 0.1$. Under independence assumption (no association), the expectation of observing both X_1 and Y_1 is $0.1^2 = 0.01$. Suppose we observe from the data that the probability of observing both items is 0.05.

Using the formulae above, we can obtain the Collective Strength value $C(I) = 1.09$. This is somewhat close to 1, which shows the weakness of the method because we cannot interpret how strong an association with C(I) = 1.09 is. However, if we compare the expected and observed frequency of I, we can see that observed frequency is 5 times higher than expectation (0.05/0.01), which should indicate a strong association. This measurement of 5 times higher than expectation is called Relative Report Rate and is utilized in the Gamma-Poisson Shrinkage model below.

All methods described later in this thesis are based on statistical development and their measures of association are more meaningful and statistically grounded than Collective Strength and thus will be better alternatives than Collective Strength in evaluating associations.

2.3 Proportional Reporting Ratio & Reporting Odds Ratio

Proportional Reporting Ratio (PRR) and Reporting Odds Ratio (ROR) are both meaningful and popular measures of association [11-13] that can test the association between one drug X_i and one event Y_j. To calculate both PRR and ROR, we first calculate the four counting values:

$$a = \sum_{l=1}^{n} I_{X_{i,l}=1 \text{ } and \text{ } Y_{j,l}=1}$$

$$b = \sum_{l=1}^{n} I_{X_{i,l}=0 \text{ } and \text{ } Y_{j,l}=1}$$

$$c = \sum_{l=1}^{n} I_{X_{i,l}=1 \text{ } and \text{ } Y_{j,l}=0}$$

$$d = \sum_{l=1}^{n} I_{X_{i,l}=0 \text{ } and \text{ } Y_{j,l}=0}$$

Simply put, *a* is the count of cases where both X_i and Y_j are observed, *b* is the count of cases where X_i is not observed but Y_j is, *c* is the count of cases where X_i is observed but Y_j is not, and *d* is the count of cases where neither X_i nor Y_j is observed. We can construct the following contingency table:

	Drug X_i	Other drugs
Effect Y_j	a	b
Other effects	c	d

Table 1: Contingency Table for PRR and ROR

PRR and ROR can then be calculated as:

$$PRR = \frac{a/(a+c)}{b/(b+d)}$$

$$ROR = \frac{a/c}{b/d}$$

PRR is the ratio between having side effect using drug A over having side effect using all other drugs. ROR measures the ratio between the odds ratio of side effect using drug A and the odds ratio of side effect using all other drugs. They both approach to 1 if there is no association between Drug A and Effect B and are bigger than 1 if the association is significant. Each measure was proven superior in certain scenarios [13].

We can construct confidence intervals for PRR and ROR as follows. PRR and ROR have skew distributions, since they are lower bounded by zero but have no upper bound. However, the logarithm of PRR and ROR can take any value and are approximately Normal distributed when *a, b, c, d* are sufficiently large [43]. Therefore, the confidence interval of PRR can be calculated as $\left(\frac{PRR}{\exp(z_\alpha s)}, PRR * \exp(z_\alpha s)\right)$ where z_α is the critical value from the Standard Normal Distribution and $s = \sqrt{\frac{1}{a} + \frac{1}{c} - \frac{1}{a+b} - \frac{1}{c+d}}$. The confidence interval for ROR is calculated as $e^{\log(ROR) \pm (z_\alpha * s)}$ where $s = \sqrt{\frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}}$. These calculations are subject to the assumption of

Normality. A pair of drug and adverse event is determined to have significant association if the lower bound of the confidence interval of PRR or ROR is larger than 1.

Advantages of PRR and ROR:

These two measures are simple to implement and both have meaningful interpretations. PRR and ROR measure how often an adverse event is reported for individuals taking a drug, compared to the frequency that the same adverse event is reported for patients taking other drugs.

Drawbacks of PRR and ROR:

There are three major issues if ROR and PRR are applied to our problem. First, since PRR and ROR compare the frequencies of an adverse event between taking a particular drug and taking other drugs, they use data of other drugs as benchmarks. If many drugs in the data are associated with the adverse event, comparison between the benchmarks and a drug that has true positive but not as frequent association will return a weak signal. Second, these methods require specification of drug X_i and side effect Y_j. In a large database such as FAERS, there are thousands of drugs and side effects and hence testing every pair of drug and effect is computationally inefficient. Finally, these methods cannot test more than one drug at a time and hence cannot be used to detect drug-drug interactions to create adverse events.

2.4 Dependence Rules (Chi-Squared Test)

Silverstein et al. also attempted to find an alternative to Association Rules using Chi-squared Test of Independence [14].

Using the Contingency table in Table 1, we calculate the expected count of each cell under the null hypothesis of independence as:

$$E_{11} = \frac{(a+b)(a+c)}{n}$$

$$E_{12} = \frac{(a+b)(b+d)}{n}$$

$$E_{21} = \frac{(a+c)(c+d)}{n}$$

$$E_{11} = \frac{(b+d)(c+d)}{n}$$

The Chi-squared test statistic is:

$$\chi^2 = \sum_{i=1}^{2}\sum_{j=1}^{2}(O_{ij} - E_{ij})$$

where O_{ij} is the observed count of cell (i, j) (a, b, c, or d). The test statistic has 3 degrees of freedom.

Chi-squared test is robust and is solidly grounded in statistical theory, but it suffers from two major weaknesses. First, it is sensitive to samples of size if any expected frequency is less than 5. Second, regular Chi-squared test of independence can only be applied to two variables. In our drug-effect problem, it can be used to test for independence between one drug and one association, but is useless with testing for drug interaction where we have more than 2 drugs and an effect.

To overcome the second problem, Silverstein et al. provides a framework for Chi-squared test of independence for more than 3 variables. The process is very similar to the 2-variable Chi-squared test. Suppose we have two drugs X_1, X_2 and an adverse event Y_3 as defined in the introduction. We would like to test the null hypothesis that they are pairwise independent as follows.

First, we construct a three-way contingency table:

		$X_1 = 1$	$X_1 = 0$
$Y_3 = 1$	$X_2 = 1$	$O_{1,1,1}$	$O_{0,1,1}$
	$X_2 = 0$	$O_{1,0,1}$	$N_{0,0,1}$
		$X_1 = 1$	$X_1 = 0$
$Y_3 = 0$	$X_2 = 1$	$O_{1,1,0}$	$O_{0,1,0}$
	$X_2 = 0$	$O_{1,0,0}$	$O_{0,0,0}$

Table 2: Three-way contingency table for Chi-squared test

where $O_{i,j,k} = \sum_{l=1}^{n} I_{X_{1,l}=i \text{ and } X_{2,l}=j \text{ and } Y_{3,l}=k}$ is the observed count of each cell. The expected counts under the null hypothesis is:

$$E_{i,j,k} = \frac{\sum_{l=1}^{n}\left(I_{X_{1,l}=i}\right)}{n} * \frac{\sum_{l=1}^{n}\left(I_{X_{2,l}=j}\right)}{n} * \frac{\sum_{l=1}^{n}\left(I_{Y_{3,l}=k}\right)}{n} * n$$

$$= \frac{\sum_{l=1}^{n}(I_{X_{1,l}=i}) * \sum_{l=1}^{n}\left(I_{X_{2,l}=j}\right) * \sum_{l=1}^{n}(I_{Y_{3,l}=k})}{n^2}$$

Then the Chi-squared statistic is $\chi^2 = \sum\left(O_{i,j,k} - E_{i,j,k}\right)^2/E_{i,j,k}$ with 4 degree of freedom.

There is a flaw if we want to apply this approach to our drug-effect problem. The Chi-Squared test also considers the dependency between X_1 and X_2 that we are not interested. We are only interested in the correlation of (X_1& Y_3), (X_2& Y_3), or (X_1& X_2 & Y_3).

One way to overcome this problem is to combine X_1, X_2 into a new variable with 4 categories, namely (00,01,10,11), and then apply the 2-variable Chi-squared test. Nevertheless, the test will not tell us whether X_1, X_2, or combination of X_1X_2 is accountable for significant side effect.

The problem with small sample remains unsolved for Chi-squared test. Chi-squared test, PRR, and ROR are all better alternatives than Association Rules and Collective Strength in evaluating drug-event association because they are built upon statistical theories. However, they all have drawbacks when it comes to testing small samples. This problem is well known for Chi-squared test [15, 16]. PRR and ROR's confidence interval are constructed using standard normal distribution [17], which is also problematic for small samples. The two methods Gamma-Poisson Shrinkage Model and Information Component both attempt to overcome this issue by assuming parametric distributions on their measures of association and finding Bayesian posterior distributions. The Bayesian methods have more complicated calculations, but they are both more conservative when sample size gets smaller.

2.5 Gamma-Poisson Shrinkage Model (aka Empirical Bayes Geometric Mean)

The Gamma-Poisson Shrinkage Model (GPS) was first developed to detect associations of international calls at AT&T, but the FDA adopted the method to their own database and found about 40,000 drug-event signals [23].

We use the same notations. Let $N_i = \sum_{l=1}^{n} X_{il}$ be the number of occurrence of drug X_i ($1 \leq i \leq p$ and $1 \leq l \leq n$ is the index for cases), $N_{ij} = \sum_{l=1}^{n} X_{il} Y_{jl}$ be the number of occurrence of both X_i and Y_j ($1 \leq j \leq q$).

A measurement of association that makes logical soundness is Relative Reporting Rate:

$$RR_{ij} = \frac{N_{ij}}{E(N_{ij})} = \frac{N_{ij}}{E_{ij}}$$

where $E_{ij} = P(X_i = 1) * P(Y_j = 1) * N = N_i * N_j/N$ is the expected count of observing both X_i and Y_j under the null hypothesis that X_i and Y_j are independent.

If $RR_{ij} \gg 1$, which means the count of $(X_i = 1\ and\ Y_j = 1)$ is much larger than its expectation under the independence hypothesis, an association between X_i and X_j is likely. DuMouchel developed the Gamma-Poisson Shrinkage Model (GPS) to test for the significance of this measurement with the Bayesian approach [19, 20]. The test is carried out as follow:

Assume that $N_{ij} \sim Poisson(\lambda_{ij} * E_{ij}), 1 \leq i \leq p, 1 \leq j \leq q$ where all the $\lambda_{ij}'s$ is drawn from a common prior distribution, which is assumed to be a mixture of two Gamma distributions. The parameters of the prior distribution is estimated from the raw data of $\lambda_{ij} = \frac{N_{ij}}{E_{ij}}$. We are interested in calculating $P(\lambda_{ij} > 1)$, since $\lambda_{ij} > 1$ means the adverse event happens more frequent expected and thus signals drug-adverse event association. The author chose a mixture of 2 Gamma distributions as prior to exploit the conjugate prior property so that the posterior distribution has a closed form. He first used a single Gamma Distribution as prior to utilize the Gamma-Poisson conjugate property, but then needed a more flexible prior distribution because he estimated the prior distribution from a whole data set. Therefore, a Gamma mixture was chosen to preserve the availability of closed-form solution and to increase the goodness-of-fit. According to the conjugate property, the unconditional distribution of each N_{ij} is a mixture of 2 negative binomial distributions [22]. The probability density function of the parameter λ_{ij} is given as:

$$\pi(\lambda; \alpha_1, \beta_1, \alpha_2, \beta_2, p) = pg(\lambda; \alpha_1, \beta_1) + (1-p)g(\lambda; \alpha_2, \beta_2), \;\; \alpha_1, \beta_1, \alpha_2, \beta_2 > 0, 0 \leq p \leq 1$$

where $g(\lambda; \alpha_1, \beta_1)$ and $g(\lambda; \alpha_2, \beta_2)$ are the probability density functions of the Gamma Distribution with shape parameters α_1, α_2 and scale parameters β_1, β_2, and p is the weight of the first distribution. The probability density function of the Gamma Distribution [21] is given by:

$$f(x, \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$$

Let $\theta = (\alpha_1, \beta_1, \alpha_2, \beta_2, p)'$. To estimate how much λ_{ij} exceeds 1 from the data, the author applied the Empirical Bayesian approach with the following steps:

- The unconditional distribution of each N_{ij} is a mixture of 2 negative binomial distributions with parameter θ. We can calculate Maximum Likelihood estimates of θ based on data of $N_{ij}'s$ and E_{ij}'s as follows.

  The Log-Likelihood function is:

  $$l(\lambda; \alpha_1, \beta_1, \alpha_2, \beta_2, p) = \sum_{i=1}^{n} \ln\left(p \frac{\beta_1^{\alpha_1}}{\Gamma(\alpha_1)} x_i^{\alpha_1 - 1} e^{-\beta_1 x_i} + (1-p) \frac{\beta_2^{\alpha_2}}{\Gamma(\alpha_2)} x_i^{\alpha_2 - 1} e^{-\beta_2 x_i}\right)$$

  We would like to find $\theta = (\alpha_1, \beta_1, \alpha_2, \beta_2, p)'$ such that $\frac{\partial l(\lambda;\theta)}{\partial \theta} = 0$. Obviously, a close-form solution is not available. Therefore, we need to use Newton-type numerical methods to estimate the solution of $\frac{\partial l(\lambda;\theta)}{\partial \theta} = 0$ [57, 58].

- For each N_{ij}, we compute the posterior distribution of λ_{ij} as $Poi(N_{ij}|\lambda_{ij} * E_{ij})\pi(\lambda_{ij}|\theta) / \int Poi(N_{ij}|\lambda_{ij} * E_{ij})\pi(\lambda_{ij}|\theta)\, d\lambda$, where $Poi(X|\lambda)$ is the Poisson probability mass function with mean λ.

- For each cell (i, j), obtain the 5th percentile of the posterior distribution $\lambda_{0.05}$. In other words, $\lambda_{0.05}$ is the lower 95% confidence bound of λ. We can then make a decision rule that, if $\lambda_{0.05} > 1$, the association of (i, j) item is significant. Since $\lambda > 1$ means a significant association, this decision rule will put the probability of false positive, which is $P(\lambda > 1|\lambda_{0.05} > 1)$, lower than 0.05.

The model is named Shrinkage because $\lambda_{0.05}$ gets smaller if N_{ij} is smaller, thus makes the test more conservative when observed size is small. The prior distribution is not pre-specified but estimated from the data. Therefore, this method follows the Empirical Bayes approach.

Advantages of GPS:

This method fixes all weaknesses of Association Rules, Collective Strength, and Chi-squared test: it has good interpretability of the measurement of association, statistical soundness, and applicability to small samples. Since the method uses the Empirical Bayes approach by estimating the prior distribution from the data, it provides inferences that are conditional on the data and are not reliant on asymptotic approximation. Therefore, we can expect this method to outperform the frequentist methods such as ROR, PRR, and Chi-squared when small samples are considered.

Drawbacks of GPS:

There are three main problems with this method. First, it cannot easily take into account the effect of demographic variables in our data (age and gender). In order to do this, the DuMouchel et al. had to stratify the data based on these covariates and repeat the same process [7]. This is computationally intensive especially when we have many stratums. Second, this method is not applicable to test more than one drug at once, which means that we cannot test for drug-drug interactions to create adverse event. Finally, the choice of mixture of Gamma Distribution as the prior distribution should be used with caution since the Bayesian approach might produce posterior distribution that are heavily influenced by the prior distribution [24].

2.6 Bayesian Confidence Propagation Neural Network (aka Information Component)

In 1996, Lansner and Holst studied the training and inference of Neural Network using the Bayesian training rule, which they called Bayesian Confidence Propagation Neural Network (BCPNN) [35]. When Bate et al. [36] applied the method to the drug-effect problem, he used a simple neural network with one input layer as drugs and one output layers as adverse events:

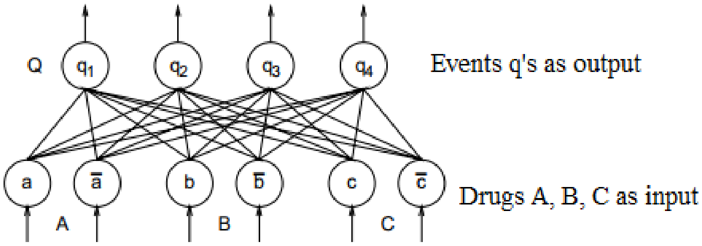

Figure 1:Neural Network by bate et al.

The expectation of the weight between input x_i and output q_i was found to be:

$$w = \log_2(\frac{P(q_i, x_i)}{P(q_i)P(x_i)})$$

which is called the Information Component (IC), and is also the log of Relative Reporting Rate in GPS method. As Bate et al. developed the method for the drug association problem, he moved away from the neural network and focused more on the estimation of IC. Therefore, even though the method inherits the name “Bayesian Confidence Propagation Neural Network”, it is in fact univariate and we do not actually interpret the results with the neural network.

Noren et al. described the Baysian estimates of IC as follows. We would like to estimate the distributions of $P(q_i, x_i), P(q_i), P(x_i)$. Using the same set up as in PRR and ROR, we consider the contingency table that is calculated from the data:

	Drug X_i	Other Drugs
Event Y_j	n_{11}	n_{10}
Other events	n_{01}	n_{00}

Table 3: Contingency Table for Information Component

where

$$n_{11} = \sum_{l=1}^{n} I_{Y_{j,l}=1 \ and \ X_{i,l}=1}$$

$$n_{10} = \sum_{l=1}^{n} I_{Y_{j,l}=1 \ and \ X_{i,l}=0}$$

$$n_{01} = \sum_{l=1}^{n} I_{Y_{j,l}=0 \ and \ X_{i,l}=1}$$

$$n_{00} = \sum_{l=1}^{n} I_{Y_{j,l}=0 \ and \ X_{i,l}=0}$$

$$n_{..} = n_{11} + n_{10} + n_{01} + n_{00} = n$$

We assume that $(n_{11}, n_{10}, n_{01}, n_{00})$ follows the Multinomial distribution with Probability Mass Function:

$$P(n_{11}, n_{10}, n_{01}, n_{00} | n, p_{11}, p_{10}, p_{01}, p_{00}) = \frac{n!}{n_{11}!\, n_{10}!\, n_{01}!\, n_{00}!} p_{11}^{n_{11}} p_{10}^{n_{10}} p_{01}^{n_{01}} p_{00}^{n_{00}}$$

where $n = n_{11} + \ n_{10} + \ n_{01} + n_{00}$ and $(p_{11}, p_{10}, p_{01}, p_{00})$ are parameters. These parameters are assumed to follow the Dirichlet distribution Dir$(\alpha_{11}, \alpha_{10}, \alpha_{01}, \alpha_{00})$ as the prior distribution. The probability density function of the prior distribution is:

$$f(p_{11}, p_{10}, p_{01}, p_{00} | \alpha_{11}, \alpha_{10}, \alpha_{01}, \alpha_{00}) = \frac{1}{\mathrm{B}(\alpha_{11}, \alpha_{10}, \alpha_{01}, \alpha_{00})} p_{11}^{\alpha_{11}} p_{10}^{\alpha_{10}} p_{01}^{\alpha_{01}} p_{00}^{\alpha_{00}}$$

where B$(\alpha_{11}, \alpha_{10}, \alpha_{01}, \alpha_{00})$ is the the multivariate Beta function. The prior parameters are calculated according to the assumption of independence between the drug and the adverse event:

$$\alpha_{11} = q_{1.} q_{.1} \alpha_{..}$$

$$\alpha_{10} = q_{1.} q_{.0} \alpha_{..}$$

$$\alpha_{01} = q_{10} q_{.1} \alpha_{..}$$

$$\alpha_{00} = q_{0.} q_{.0} \alpha_{..}$$

where

$$\alpha_{..} = \frac{0.5}{q_{1.} q_{.1}}$$

$$q_{1.} = \frac{n_{1.} + 0.5}{n_{..} + 1}$$

$$q_{0.} = \frac{n_{0.} + 0.5}{n_{..} + 1}$$

$$q_{.1} = \frac{n_{.1} + 0.5}{n_{..} + 1}$$

$$q_{.0} = \frac{n_{.0} + 0.5}{n_{..} + 1}$$

The conjugate prior property makes the posterior distribution Dirichlet with parameters $(\gamma_{11}, \gamma_{10}, \gamma_{01}, \gamma_{00})$ where $\gamma_{ij} = \alpha_{ij} + n_{ij},\ i, j \in \{0,1\}$.

Knowing the posterior distribution for $(p_{11}, p_{10}, p_{01}, p_{00})$, we can calculate the expectation of IC as:

$$E(IC) = \log_2(\frac{E(p_{11})}{E(p_{1.})E(p_{.1})})$$

Obviously, the closed form of distribution of IC is unknown, we need to estimate the lower 95% confidence bound by Monte Carlo Simulation or Normal Approximation.

If the lower 95% bound is larger than 0, a signal is determined.

The Bayesian approaches, GPS and IC, were proven to have better performance than PRR, ROR, and Chi-squared with higher area under the Receiver Operating Characteristic (ROC) curve [28]. With modern computer's strength, performing complex Bayesian calculation is not too intensive and therefore, GPS and IC should be a superior choice over PRR, ROR, or Chi-squared.

2.7 Logistic Regression

PRR, ROR, GPS, and BCPNN are called Disproportionality methods. They all have two drawbacks. First, they cannot easily consider demographic variables such as age and gender. Second, they are vulnerable to raise false positive for co-prescribed drugs. For example, drug A and drug B are often prescribed together but only drug A causes a side effect. Disproportionality methods, even the Bayesian ones, will likely find drug B associated with the side effect because the two drugs are not considered simultaneously. Logistic Regression (LR) was first applied to this type of problem by DuMouchel (2004) [25]. An advantage of Logistic Regression over all the previous methods is that it considers all variables at once and hence is less vulnerable to the co-prescribed drugs situation.

The logic is straight forward: we consider each adverse event Y_j $(1 \leq j \leq q)$ as a binary response variable and all drugs $X_1, X_2, \ldots, X_p$ as explanatory variables. The logistic regression has the form:

$$logit\left(P\left(Y_j = 1\right)\right) = \log\left(\frac{P\left(Y_j = 1\right)}{1 - P\left(Y_j = 1\right)}\right) = \sum \beta_i X_i$$

We can also add demographic information Z_i as covariates:

$$logit\left(p(Y_h)\right) = \log\left(\frac{P\left(Y_j = 1\right)}{1 - P\left(Y_j = 1\right)}\right) = \sum \beta_i X_i + \sum \alpha_i Z_i$$

We are interested in the significance of β_i's in this regression using the usual t-test.

Interestingly, a recent study that compared the methods using FDA data shows that Logistic Regression family performs better than GPS and generally has higher specificity at a given level of sensitivity [27]

To investigate drug-drug interaction, we just need to add the interaction terms to the model:

$$logit\left(p(A)\right) = \sum \beta_i D_i + \sum \alpha_i X_i + \sum \gamma_i D_i D_j$$

However, this will increase the number of parameters quickly. 1,000 drugs will yield 500,000 interaction terms, which can easily exceed the amount of data to fit. An alternative is to include only the drug combinations that are observed in the data more than an arbitrary threshold. For example, we may only include in the model the pairs of drugs that are co-prescribed more than 5 times in the data.

Another drawback of logistic regression is that it requires a large amount of data to obtain a stable model. A recent study shows that a 20:1 ratio between numbers of observations and parameters are needed [26]. Nevertheless, this is not our issue since we are currently dealing with rather large FAERS database.

2.8 Regression - Adjusted Gamma-Poisson Shrinkage Model

DuMouchel's GPS method was found to perform worse than Logistic Regression [27].However, the use of t-test in Logistic Regression is vulnerable to small samples, which was one of the reasons why GPS was introduced [19]. In 2012, DuMouchel combined GPS and LR into a hybrid method that has strengths of both [28]. The main idea is to replace the t-test of coefficient significance in LR by GPS instead of the t-test. First we select a subset of p drugs to fit the Logistic Regression model. Suppose the subset of predicting drugs is $S \subset \{1, 2, \dots, p\}$. The Logistic Regression model is:

$$logit\left(P\left(Y_j = 1\right)\right) = \sum_{i \subset S} \beta_i X_i$$

In the publication, DuMouchel selects the predicting drugs based on their event rates. We can rewrite this equation to include all drugs $X_i, 1 \le i \le p$, but set $\beta_i = 0$ if $i \not\subset S$:

$$logit\left(P\left(Y_j = 1\right)\right) = \sum_{i=1}^{p} \beta_i X_i\,, \qquad where\ \beta_i = 0\ if\ i \not\subset S$$

Unlike the regular Logistic Regression, we do not use the t-test for significance of β_i's as the final decision. Instead, DuMouchel proposed to calculate the expected count of observing both $X_k, 1 \leq k \leq p,$ and Y_j under the null hypothesis that drug X_k has no effect on event Y_j to be used for the rest of the GPS process. The null hypothesis is equivalent to $\beta_k = 0$. Therefore, the expected probability of event $Y_j = 1$ is calculated as:

$$E(logit\left(P\left(Y_j = 1\right)\right)) = \sum_{i=1}^{p} \beta_i X_i - \beta_k X_k$$

therefore,

$$E\left(P\left(Y_j = 1\right)\right) = 1/(1 + \sum_{i=1}^{p} \beta_i X_i - \beta_k X_k)$$

We apply this formula to each row of the data (each patient in the data) to calculate each of their expected count of observing event Y_j. Then, the expected count of event Y_j under the null hypothesis $\beta_k = 0$ is the sum of $E\left(P\left(Y_j = 1\right)\right)$ across all data records (again, rows are indexed with $1 \leq l \leq n$):

$$E_{kj} = \sum_{l=1}^{n} (\frac{1}{1 + \sum_{i=1}^{p} \beta_i X_{il} - \beta_k X_{kl}})$$

This process is repeated for each of the drugs $X_k, 1 \leq k \leq p$. As a result, we get an array of expected counts E_{kj} of observing both drug $X_k, 1 \leq k \leq p,$ and adverse event $Y_j, 1 \leq j \leq q$.

In the original GPS method, this is calculated based on raw data: $E_{kj} = P(X_k = 1) * P\left(Y_j = 1\right)/N = N_k * N_j/N$.

GPS method is then continued as in section 2.6 with this new expected count $E_{kj} = \sum_{l=1}^{n}(\frac{1}{1+\sum_{i=1}^{p}\beta_i X_{il} - \beta_k X_{kl}})$

Regression-adjusted GPS was proven in the same study to have better performance than both LR and GPS [28]. This is intuitive because it combines the sample size-sensitive Bayesian method and the multivariate method of calculating expected count.

Since RGPS is not available in any public software package, we attempted to write the program according to DuMouchel's description. We made a slight adjustment to the algorithm however. We do not select the predicting variables based on their event rates but using a forward step-wise algorithm with Akaike information criterion [55].

CHAPTER 3: THE NOVEL METHODS

All the methods discussed in Chapter 2 suffer from a common problem. They do not automatically evaluate interactions between drugs unless we clearly state the specific interactions in the model (only for Chi-squared Test and Logistic Regression). Specifying interactions might be arduous or even impossible when the number of drugs p and the number of adverse events q get large. Therefore, we attempt to apply two algorithms, Random Forests and Monte Carlo Logic Regression, to this drug association problem. These two algorithms can detect interactions between input variables along with the main effects without specifying the interactions. They were both successfully applied in genome-wide association studies to detect both the main effects and interactions [37 - 42].

For both methods, we consider a specific adverse event $Y_j, 1 \leq j \leq q$ (output variable) and all drugs in the data $X_1, X_2, \ldots, X_p$ (input variables). Both methods attempt to predict the value of Y_j using the given values of $X_1, X_2, \ldots, X_p$ and evaluate the significance of each of the input variables and their interactions in the process.

3.1 Tree-Based Methods

Random Forests is a non-parametric method for regression and classification and requires no assumption about the data [44, 45]. To describe Random Forests, we first need to introduce Decision Trees, which is a simpler method for regression and classification.

3.1.1 Decision Tree

Decision Tree consists of many levels of decision nodes, each splits the one of the input variables into two categories. Therefore, a Decision Tree partitions the input variables' domain, and the bottom branches of a Decision Tree show the predicted values for each partition. Figure 2 shows an example of a Decision Tree using notations from our problem. The ending boxes to the far right of the tree, labeled either 0 or 1, indicates the best prediction value of Y_j for that partition. For example, the top branch of the tree means that when $X_1 = 1 \; and \; X_2 = 1$ then the best prediction for Y_j is 1.

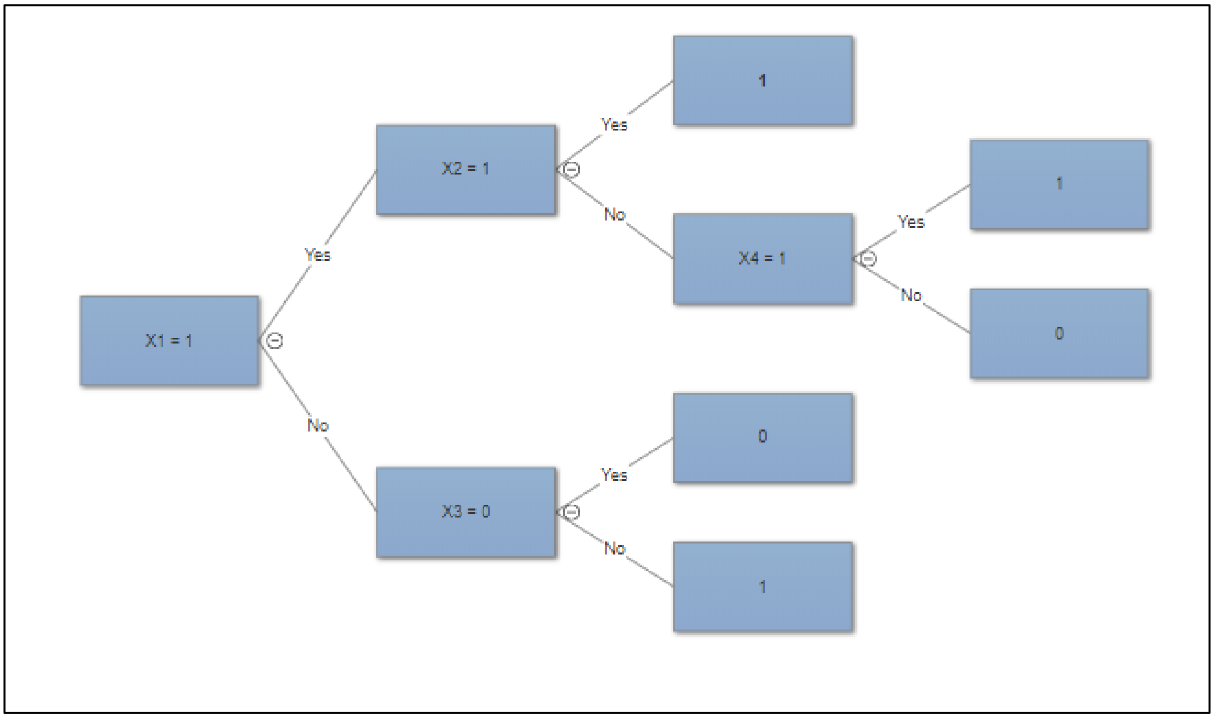

Figure 2: An example of Decision Tree

We now discuss the process of building an optimized Decision Tree. The goal is to divide the predictor space, which is the set of all possible values of $X_1, X_2, \dots, X_p$, into J distinct and non-overlapping regions $R_1, R_2, \dots, R_J$ with $n_1, n_2, \dots, n_J$ observations respectively. For each region $R_m, 1 \leq m \leq J$, the predicted value is the most common class in that region. The classification error rate in region R_m is the proportion of observations not equal to predictions:

$$1 - \hat{p}_m = 1 - \frac{1}{n_m} \max(\sum_{X_1, X_2, \ldots, X_p \in R_m} I\big(Y_j = 0\big), \sum_{X_1, X_2, \ldots, X_p \in R_m} I\big(Y_j = 1\big))$$

Then the classification error rate for the whole tree is: $1 - \hat{p} = \frac{1}{n} \sum_{m=1}^{J} n_m (1 - \hat{p}_m)$

Gini Index is another measure of region purity. Since our classification problem only has two classes 0 and 1, the Gini Index formula [45] is reduced to:

$$G = 2\hat{p}_m (1 - \hat{p}_m)$$

The goal is to construct a decision tree with the highest measure of purity. Breiman [46] described the process of finding the best decision tree using a greedy algorithm as follows.

- Starting with all the data, for each input variable $X_i, 1 \leq i \leq p$, we split the input space into two half-planes: $R_1(i) = \{X \big| X_j = 0\big\}$ and $R_2(i) = \{X | X_j = 1\}$. Then we calculate the misclassification rate $1 - \hat{p}^{(i)} = \frac{1}{n} \sum_{m=1}^{2} n_m (1 - \hat{p}_m)$.
- Select the input variable X_i that has the lowest misclassification rate $1 - \hat{p}^{(i)}$.
- Having found the best splitting variable, we partition the data into two sub-regions R_1 and R_2.
- Repeat this process on each sub-region until the misclassification rate stops decreasing.

How many times should we split the data, or how large should we grow the tree? A common strategy is to grow a very large tree, called tree T_0, until the sample sizes $n_j, 1 \leq j \leq J$ reach a pre-determined number (usually 5). Then this large tree is simplified by cost-complexity pruning as follows.

We define a subtree T of T_0 to be any tree that can be obtained by removing a number of T_0's non-terminal nodes. Let $|T|$ denote the number of terminal nodes in T. The false classification rate in region R_m of tree T is:

$$1 - \hat{p}_m(T) = \frac{1}{n_m} \max(\sum_{X_1, X_2, \ldots, X_p \in R_m} I(Y_j = 0), \sum_{X_1, X_2, \ldots, X_p \in R_m} I(Y_j = 1))$$

The cost-complexity criterion is define by

$$C_\alpha(T) = \sum_{m=1}^{|T|} n_m(1 - \hat{p}_m(T)) + \alpha|T|$$

where α is the penalizing parameter for the tree size, which can be determined by cross-validation [45]. For each value of α, there is only a finite number of sub-trees T and we find the sub-tree that produces the lowest $C_\alpha(T)$.

3.1.2 Random Forests

Decision Tree suffers from high variance, which means that a slight change in the data can yield a significantly different tree and prediction. Random Forests is a popular way to reduce variance and increase prediction power [46]. Random Forests makes two improvements on Decision Tree:

First, we bootstrap the data by taking repeated *B* samples from the training data set, generally by repeatedly sampling 2/3 of the data. We then train Decision Tree on each of the *B* bootstrapped samples and average all the predictions. Suppose we have *B* Decision Trees $T_b, 1 \leq b \leq B$ corresponding to *B* bootstrapped samples, the prediction for an input vector x is:

$$T(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

Second, when building decision trees, each time a split is performed, a random sample of *m* out of *p* predictors is chosen as split candidates instead of all the *p* predictors. The rationale is that, suppose that there are some very strong predictors in the data set, then most trees will use these strong predictors in the top splits. Therefore, many of the trees will have similar structure and hence will be highly correlated. By sampling the predictors, we reduce the correlation between trees and hence making the average of trees more reliable [46]. A popular choice of m is $\sqrt{p}$.

3.1.3 Variable Importance

The ultimate purpose of our study is to determine how important each input variable X_i is in predicting Y_j. At each split in each tree, the reduction in false classification rate of the whole tree $1 - \hat{p}$ or the Gini Index $G = 2\hat{p}_m(1 - \hat{p}_m)$ is attributed to the splitting variable, and is accumulated over all trees in the forest for each variable. For each tree T_b:

- If X_i is not used in the tree, its variable importance for tree *b* is $VI_b(X_i) = 0$
- If X_i is used in the tree, the variable importance for tree *b* is the reduction in false classification rate or Gini Index before and after the split. Suppose the false classification rate before the split is $1 - \hat{p}_{(before)}$ and the false classification rate after the split is $1 - \hat{p}_{(after)}$, then the variable importance for tree *b* is $VI_b(X_i) = \hat{p}_{(after)} - \hat{p}_{(before)}$. Suppose the Gini Index before the split is $G_{(before)}$ and the Gini Index after the split is $G_{(after)}$, then the variable importance for tree *b* is $VI_b(X_i) = G_{(before)} - G_{(after)}$

The total variable importance of $\hat{p}_{(after)}$ is then $VI(X_i) = \sum_{b=1}^{B} VI_b(X_i)$. Since $VI(X_i)$ is dependent on the number of tree B, there is no accurate cut-off point to determine whether $VI(X_i)$ is significant or not. Instead, we rank all the $VI(X_i)$ from largest to smallest and only consider several largest $VI(X_i)$ to be significant. Significant $VI(X_i)$ also means that there is significant association between drug X_i and adverse event Y_j.

An important reason why we proposed Random Forests is its inherent ability to detect interacting variables without specifying them in a model [39, 47, 49]. The regular variable importance, however, does not provide us with a convenient way to measure the interactions from a Random Forests. This could be done using the idea of Maximal Subtrees [50]. For a decision Tree T, Ishwaran et al. defined a X_v-subtree T_v as a part of T that has the top node split by variable X_v. T_v is called a maximal X_v-subtree if T_v is not a subtree of a larger X_v-subtree. Let D_v denote the distance from the root of T to the root of a maximal X_v-subtree, which is the number of nodes between the root of T_v and the root of T plus one. We further define second-order maximal (X_i, X_j)-subtree as the maximal X_j-subtree within a maximal X_j-subtree. The minimal depth of a second-order maximal (X_i, X_j)-subtree is the distance from the root of (X_i, X_j)-subtree to the root of X_i-subtree. The minimal depth of a second-order maximal (X_i, X_j)-subtree is a measurement of interaction between X_i and X_j. For a Random Forests, we average the minimal depths of (X_i, X_j)-subtree and (X_j, X_i)-subtree across all decision trees to compute the joint importance of X_i and X_j. All the joint importance for a Random Forests can then be ranked to determine the most significant interactions.

3.2 Monte Carlo Logic Regression

Logic Regression was developed for genomic association studies to relate single nucleotide polymorphisms (SNPs) to disease outcomes [40, 41]. It was designed for situations where most predictors are binary (taking value 0 or 1) and the goal is to find Boolean combinations of these predictors that are associated with an outcome variable. Our drug association study is one such situation where most predictors (drugs) are binary and we are interested in finding interactions between drugs to create an adverse event. Therefore, it would be interesting to see how this method fit in to our problem.

3.2.1 Logic Regression

We first simplify our notations for convenience. We denote the use of drug X_i as X_i instead of $X_i = 1$, and not using drug X_i as X_i^c instead of $X_i = 0$. Similarly, we denote an observation of event Y_j as Y_j instead of $Y_j = 1$ and no observation of Y_j as Y_j^c. Let $X_1 \wedge X_2$ denote the event of observing both X_1 and X_2, and $X_1 \vee X_2$ denote the event of observing either X_1 or X_2. For example, the notation $X_1^c \wedge (X_2 \vee X_3)$ means not observing X_1 and observing (X_2 or X_3). Such a combination is called a Logic Tree and can be presented in a tree as in figure 3.

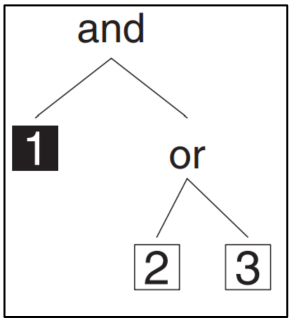

Figure 3: An example of Logic Tree

In figure 3, the numbers are the subscriptions of variables. For instance, number 1 in the figure represents X_1. The black color indicates compliment of that variable. Therefore, the black number one in the figure represents X_1^c. For any row in a data set, the tree takes value of 1 if its expression is true in that row and 0 otherwise.

The Logic Regression model has the form

$$g[E(Y|X)] = \beta_0 + \sum_{i=1}^{K} \beta_i L_i$$

where g is a link function, $\beta_0, \beta_1, \dots, \beta_K$ are parameters, and L_i are the Logic Trees on the input variables $X_1, \dots, X_p$. For any link function, we define a score function that reflects the quality of the model. For instance, an identity link function (linear regression) may have the sum of squares the score function, a logit link function (Logistic Regression) may have Deviance as the score function. The number of parameters in this model is always $K + 1$ and does not depend on how many input variables are in the model. The challenge is how to form the Logic Trees and how many trees we should use.

We first discuss how to form the Logic Trees. We start with *K* number of trees, each tree is $L = 0$. We iteratively grow the trees. At each iteration, a tree is selected at random and modified using one of the six moves:

- *Alternate a leaf*: we pick a leaf and replace it with another leaf
- *Alternate operators*: replace $\wedge$ by $\vee$ and vice versa
- *Grow Branch:* for any knot that is not a leaf, we add a new branch by moving the current subtree below to the right and add another branch to the left, connecting by either $\wedge$ or $\vee$
- *Prune Branch:* for any knot that is not a leaf, we remove one side and shift up the other.
- *Split Leaf*: Add one leaf to the position of an existing leaf, connecting the two by either $\wedge$ or $\vee$

- *Delete Leaf*: Remove one leaf in a pair of leaves.

These six moves are demonstrated in figure 4, which was taken from [41].

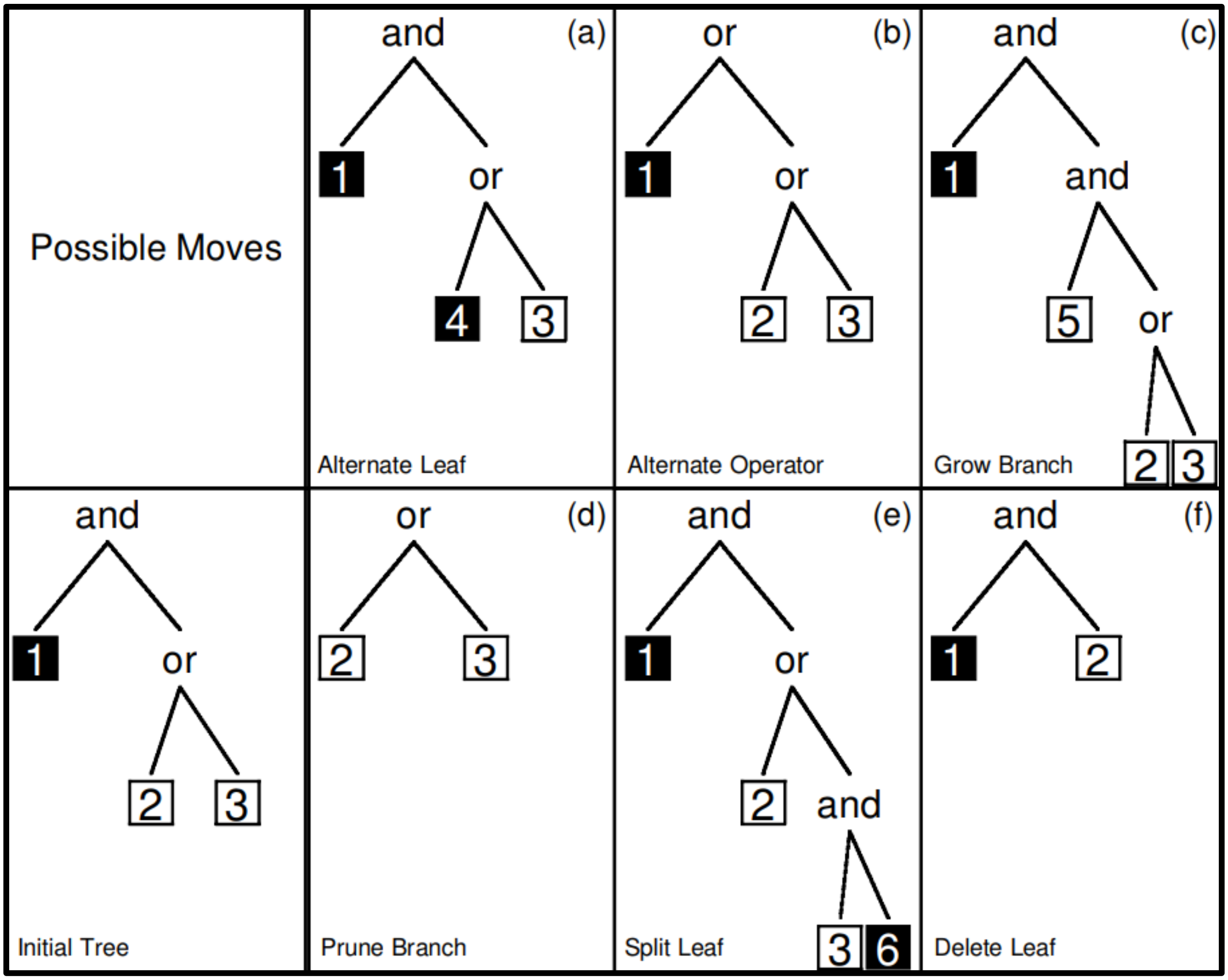

Figure 4: Demonstration of the six moves to modify Logic Tree

Then with the new tree in the model, we estimate the parameters $\beta_0, \beta_1, \dots, \beta_K$ and calculate the score function. If the new tree improves the score function of the model, it is accepted and replaces the old tree. Otherwise, it is accepted with a probability that depends on the difference between the old and the new scores. The higher iteration, the lower this probability of acceptance will be.

Next, we discuss how to choose the best number of tree K. We can do this by cross-validation. The data is repeatedly split into a training set and a test set. Logic Regression models with different K are fitted on the training data. The K that has the best score is selected, and a model of that size is computed on the complete data.

3.2.2 Monte Carlo Logic Regression

Since the process of Logic Regression is random, we might obtain a different model at each run. Our result therefore will be highly variated. As we are not interested in the coefficients $\beta_0, \beta_1, \dots, \beta_K$ but in the Logic Trees L_1 in the model, running the regression model multiple times and summarizing the information in trees L_1 will serve our purpose better than a single Logic Regression model. Therefore, the goal of Monte Carlo Logic Regression is to identify all models and combinations of input variables that are associated with the outcome.

Kooperberg and Ruczinski used the Markov chain Monte Carlo (MCMC) to explore a large number of good-fitting models [40]. They implemented the reversible jump MCMC algorithm of Green [48]. They first select a geometric prior on the model size, which is the total number of leaves on all of the Logic Trees. For instance, the model $\beta_0 + \beta_1(X_1 \vee (X_8 \wedge X_9)) + \beta_2 X_{10}$ has size 4. For each model size, they calculated the total number possible logic regression models and assume uniform prior distribution on all logic regression of a particular size. Iteratively, a model is then selected at random and the likelihood ratio, the prior ratio, and the posterior ratio are computed [48]. More details of the algorithm can be found in [41].

After the MCMC simulation, we obtain a large number of Logic Regression models. The importance of input variables and interactions can be calculated and ranked as follows.

- We calculate the fraction p_i of models that contain the input variable X_i. An input variable that appear in multiple places in different Logic Trees in the same model is only counted as one appearance. This fraction p_i is an indicator of how important variable X_i is for predicting the outcome rather than its own association with the outcome. To obtain the direct association between X_i and the outcome, we subtract second-order and higher fractions (described below) from p_i.
- We calculate the fraction p_{ij} of models that contain both X_i and X_j in the same logic tree. This indicates whether an interaction between X_i and X_j may be associated with the outcome. Similarly, we can count how often triplets, quadruplets of input variables occur together in models.
- The fractions are ranked to determine the most significant variables and interactions in predicting the outcome.

Monte Carlo Logic Regression is a very powerful tool to detect interactions between binary input variables. As described by Witte and Fijal, this method was the only out of ten approaches that identified all correct associations between genetic sequences and a disease, including the interactions between genetic sequences [56].

CHAPTER 4: COMPARISON STUDY

4.1 The Gold Standard for Testing

The Observational Medical Outcomes Partnership (OMOP)'s aim is to evaluate methods for analyzing data in electronic medical records. It has developed a reference set of drug–event pairs that are classified as positive or negative controls, called Gold Standard, which consists of drug–event pairs that the OMOP proposes would return positive or negative results from a perfect test, designed to serve as a test bed for quantitative techniques [29]. Though imperfect, the Gold Standard has been described as the best available benchmark. An early test set constructed by OMOP consisted of 53 drug–event pairs, nine positive controls (drug-event association exists) and 44 as negative controls [30]. Positive control was determined based on listing of the event in the product label along with prior observational database research suggesting an association, followed by expert panel consensus. Negative control assignment was determined based on absence of the association in the product label and published literature, followed by endorsement by an expert panel. Subsequently, a larger test set consisting of 398 test cases (165 positive controls and 233 negative controls) was published using related but distinct criteria [31].

The full OMOP's list of drug and adverse event and counts of their occurrences in the FAERS database are presented in Appendix A. Out of the 398 pairs of drug-event, only four distinct adverse events exist, namely Acute Kidney Injury (AKI), Acute Liver Injury (ALI), Acute Myocardial Infarction (AMI), and Gastrointestinal Bleed (GIB).

4.2 The FAERS Database

The FAERS quarterly datafiles since the second quarter of 2014 [51] were combined into a local database at the University of South Florida. The database has 7 tables, all named the same as the 7 tables in the FAERS quarterly data files. In this study, we primarily used table *Drug*, which contains more than 17 million records of drugs taken, and table *Reaction*, which contains more than 14 million records of adverse event observed.

Using Structured Query Language (SQL), we first transformed the data into a format that can be analyzed in R. We joined table *Drug* and table *Reaction* on the field *primaryid*, which is the code that identifies individuals taking drugs (if the same person takes drugs at two different times, the two *primaryid's* are different). Then for each *primaryid*, we concatenate all the taken drugs' active ingredients (AI) into one field named *prod_ai*. Since the OMOP Gold Standard has only four types of adverse event, we created four column named AMI, AKI, GIB, and ALI to denote existence (1) or absence (0) of each event in each case. The top ten rows of the resulted table are shown in table 3.

primaryid	**prod_ai**	**AMI**	**AKI**	**GIB**	**ALI**
101329582	ASPIRIN,LISINOPRIL,METFORMIN HYDROCHLORIDE\ROSIGLITAZONE MALEATE,ROSIGLITAZONE MALEATE	0	0	0	0
107006552	ACETAMINOPHEN\CODEINE PHOSPHATE,ALPRAZOLAM,BISOPROLOL,CLINDAMYCIN\CLINDAMYCIN PHOSPHATE,DESLORATADINE,ERTAPENEM SODIUM,FUROSEMIDE,GABAPENTIN,HYDROCHLOROTHIAZIDE,OFLOXACIN,PANTOPRAZOLE SODIUM,SERTRALINE HYDROCHLORIDE,TELMISARTAN,VANCOMYCIN,VORICONAZOLE	0	0	0	0
109070881	ACETAMINOPHEN\HYDROCODONE BITARTRATE,ALBUTEROL,ALBUTEROL SULFATE\IPRATROPIUM BROMIDE,CARVEDILOL,DIGOXIN,FUROSEMIDE,GEMFIBROZIL,IBUPROFEN,INSULIN DETEMIR,LISINOPRIL,METFORMIN HYDROCHLORIDE	0	0	0	0
105140671	OLMESARTAN MEDOXOMIL,PREGABALIN	0	0	0	0
106082232	ASPIRIN,BISOPROLOL,CALCIUM CARBONATE,CHOLECALCIFEROL,CYCLOSPORINE,DILTIAZEM,EVEROLIMUS,EZETI	0	0	0	0

	MIBE,INSULIN NOS,PANTOPRAZOLE SODIUM,PRAVASTATIN\PRAVASTATIN SODIUM,TELMISARTAN,ZOLEDRONIC ACID				
106518922	ALISKIREN HEMIFUMARATE,AMLODIPINE BESYLATE,CARVEDILOL,DICLOFENAC SODIUM,FLUTICASONE\FLUTICASONE PROPIONATE,HYDROCHLOROTHIAZIDE,HYDROCODONE,INFLUENZA VIRUS VACCINE,LEVOTHYROXINE SODIUM,MELOXICAM,OXYCODONE,PREDNISONE,TRAMADOL HYDROCHLORIDE,VALSARTAN	0	1	0	0
113035051	CETIRIZINE HYDROCHLORIDE,DILTIAZEM HYDROCHLORIDE,LISINOPRIL	0	0	0	0
115184541	LORATADINE	0	0	0	0
114311161	ESTRADIOL,THYROID, PORCINE\THYROID, UNSPECIFIED	0	0	0	0
114781811	ACETAMINOPHEN\HYDROCODONE BITARTRATE,ALBUTEROL SULFATE,BUDESONIDE\FORMOTEROL FUMARATE DIHYDRATE,CALCIUM CARBONATE,CETIRIZINE HYDROCHLORIDE,CHOLECALCIFEROL,CROMOLYN,CYANOCOBALAMIN,FLUNISOLIDE,GABAPENTIN,HYDROCHLOROTHIAZIDE,HYDROCORTISONE BUTYRATE,MELOXICAM,METHYLPHENIDATE,OMEPRAZOLE,PANTOPRAZOLE SODIUM,PLANTAGO SEED,SODIUM OXYBATE,TERAZOSIN\TERAZOSIN HYDROCHLORIDE,TESTOSTERONE CYPIONATE,WARFARIN SODIUM	0	0	0	0

Table 4: Merging and Transforming Drug Table and Reaction Table

There are 183 distinct drugs in the OMOP Gold Standard. Therefore, we create 183 indicator variables corresponding to each drug, taking value 1 if the drug exists in *prod_ai* and 0 otherwise. For example, the variable HYDROCHLOROTHIAZIDE has value 1 on the second row of Table 3 because *prod_ai* in this row contains the string “HYDROCHLOROTHIAZIDE”.

This data table is then transferred to R to be perform 8 of the methods mentioned in chapter 2 and 3, namely Proportional Reporting Ratio (PRR), Reporting Odds Ratio (ROR), Gamma-Poisson Shrinkage Model (GPS), Bayesian Confidence Propagation Neural Network (BCPNN), Logistic Regression (Logistic Reg), Regression-adjusted GPS (RGPS), Random Forests (R Forests), and Monte Carlo Logic Regression (MC Logic Reg)

4.3 Computational details

After processing the data using SQL, the data table with 187 binary columns are transferred to R to perform the 8 methods. PRR, ROR, GPS, and BCPNN are all available in the

R package "*PhViD*" [52]. Logistic Regression exists within the base function *glm* in R. Random Forests is available in the package "*RandomForests*" [53]. For each adverse event, we grew 100 decision trees. Monte Carlo Logic Regression is available in the package "*LogicReg*" [54]. For each adverse event, we choose the logit link function (logistic regression) and 25,000 iterations of MCMC. Detailed description of RGPS was published in 2012 [28] but does not exist in any public software package and hence we needed to compile the program. As discussed in section 2.8, we select the predicting variables using a forward step-wise algorithm instead of using drugs' event rates. The R code for RGPS is presented in Appendix B.

We then compared the outputs of all methods by plotting the Receiver Operating Characteristic (ROC) curves, calculating the Area under the ROC curves, and recording computing time.

4.4 Results of Performance Testing

The Receiver Operating Characteristic curves of all methods are presented in figure 1:

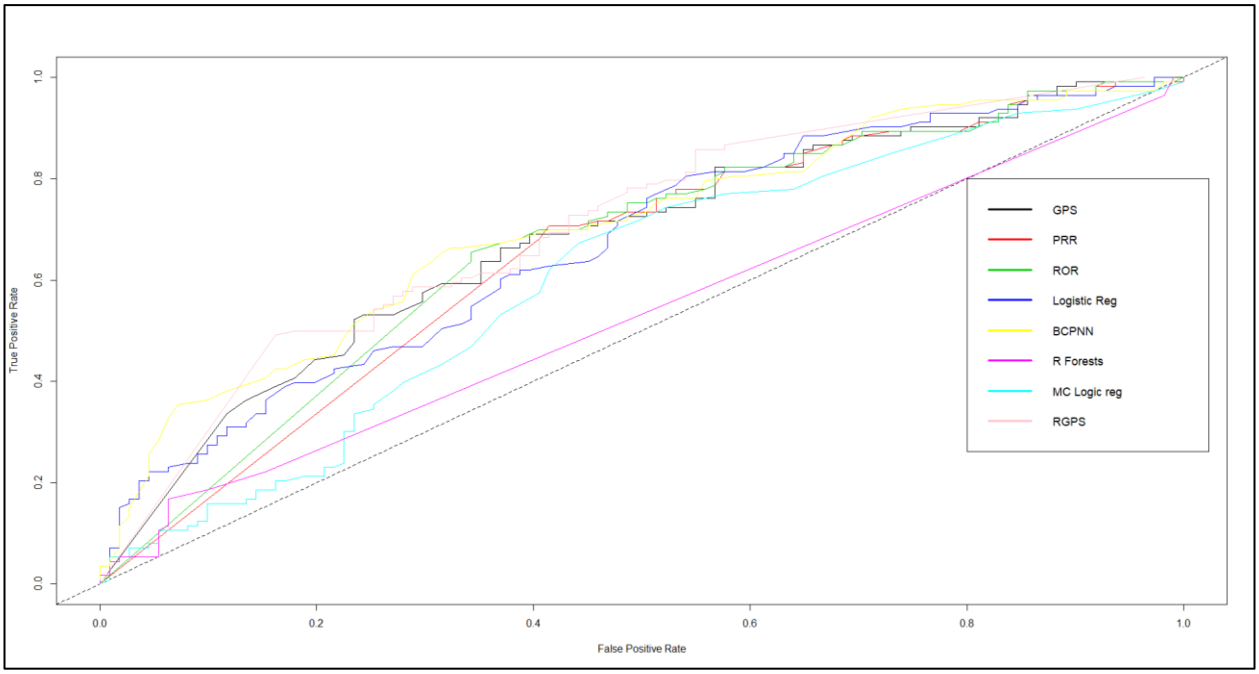

Figure 5: ROC curve of 7 different methods

Areas under the curves are presented in table 3, ordered from largest to smallest.

Method	**Area Under Curve**
RGPS	0.7091224
BCPNN	0.693893
GPS	0.6803396
ROR	0.6653113
Logistic Reg	0.6604082
PRR	0.6513593
MC Logic Reg	0.6050785
Random Forests	0.5208084

Table 5: Areas Under Curve

The total computing time to scan the database for each method is given in table 5, ordered from shortest to longest:

Method	Computing Time
Logistic Reg	8.04 minutes
PRR	12.73 minutes
ROR	12.73 minutes
GPS	12.73 minutes
BCPNN	12.81 minutes
MC Logic Reg	14.21 minutes
Random Forests	8.17 hours
RGPS	19.83 hours

Table 6: Computing Time

Regarding performance, RGPS has the best correct classification rate, followed closely by BCPNN and GPS. This is consistent with the results from DuMouchel and Harpaz [28]. The two novel methods Random Forests and Monte Carlo Logic Regression perform the worst. Random Forests is only slightly better than random guess (50% chance).

One possible explanation for this situation is the sparseness of the data. AMI occurs in 0.66 % of the records, AKI occurs in 1.88% of the records, GIB occurs in 0.016% of the records, and ALI occurs in 0.4% of the records. Since Random Forests creates bootstrapped samples from the data, a lot of the bootstrapped samples will contain no observation of the adverse event. Similarly, the drugs are also sparse. When bootstrapped samples with no observation are used to construct Decision Trees, no association can be measured.

Monte Carlo Logic Regression does not perform as bad as Random Forests because it does not use bootstrapped samples. However, there is an issue in applying Monte Carlo Logic

Regression to our problem. The compliment logics on input variables does not make sense in the context of drugs and adverse events. For example, association between X_1^c and Y_1 means that not taking drug X_1 will result in adverse event Y_1. This interpretation is not meaningful in the context of our problem. Since Monte Carlo Logic Regression was designed for genetic and genomic association study, the compliment logics was implemented to explain relationships such as not having genetic sequence X_1 will result in disease Y_1. Therefore, we expect the removal of the compliment logics to boost the performance of Monte Carlo Logic Regression in our problem.

Regarding computing time, Random Forests and RGPS take significantly longer time than the other methods. The long time taken in RGPS can be attributed to our modification on the algorithm using the step-wise selection method.

CHAPTER 5: CONCLUSION AND DISCUSSION OF FUTURE WORK

The purpose of this thesis was to introduce the drug – adverse event association study problem, review the literature, and perform a comparison study. The findings of this study lead to the following discussions and conclusions.

Several methods have been proposed in the literature for the drug – adverse event association study. Proportional Reporting Ration (PRR) and Reporting Odds Ratio (ROR), which follow the frequentist approach, are the most commonly used methods. However, our comparison study pointed out that the best-performing approaches are the Bayesian approaches, namely Gamma-Poisson Shrinkage model (GPS) and Bayesian Confidence Propagation Neural Network (BCPNN). These two Bayesian approaches have advantages over other the frequentist methods for their statistical soundness and robustness against small samples. Despite the strengths of GPS and BCPNN, we would like to contribute to the drug – adverse event association study by addressing two issues. First, we are interested in multivariate method that can resolve confounding factors such as commonly co-prescribed drugs. In the literature, only Logistic Regression and Regression-Adjusted Gamma Poisson Shrinkage model (RGPS) addressed this issue. In addition, the description of RGPS was published in 2012 but is currently not available in any public software package. Second, we would like to find a method that can test for all interactions without having to specify the interactions in a model, because the number of interaction terms can be too large to specify. None of the current approaches can do this.

The drug – adverse event association study shares many similarities with the market-basket problem and the genetic and genomic association study. Therefore, the drug – adverse event association study may benefit from the vast variety of approaches from these two problems. In the literature of the market-basket problem and the genetic and genomic association study, we have identified two approaches that have the properties in being multivariate and automatically considering interactions. Random Forests was introduced by Breiman in 2001 and is applicable and popular in a wide range of problems. It is non-parametric, non-linear, and inherently measures interactions between input variables. Monte Carlo Logic Regression was introduced by Kooperberg and Ruczinski in 2004 to deal with a large number of binary input variables in genetic and genomic association problem. The approach also helps us evaluate second-order and higher interactions without having to specify the interaction terms in the model. Therefore, Monte Carlo Logic Regression fit into our problem perfectly.

Nevertheless, our comparison study shows that the drug-adverse event problem has special issues that require modifications of the two novel methods. The sparseness in data makes Random Forests fail to perform properly because many of the bootstrapped samples it creates may contain no information. Long computing time is also an issue to this method. Monte Carlo Logic Regression has a decent performance but suffers from the compliment logics not being applicable in the context of the problem. Performance of the other methods are found to be consistent with DuMouchel and Harpaz's study [28].

We can suggest two items for future works. First, we suggest modifications on the Monte Carlo Logic Regression method to remove the compliment logics. Since the compliment logics was created in the basis of Logic Tree, which is the foundation for Monte Carlo Logic Regression, removing the compliment logics will require modification of all the codes in the R package "*LogicReg*". This is going to be an arduous work since the program was built on several years of work. Second, we are interested in looking at the drug – adverse event association study as a time-series problem and evaluating the trends in association signals over time. In the study discussed in this thesis, we combined all submissions of FAERS into one large database and hence ignored the dynamics of signals over time. We believe that the association between drugs and adverse events might not be stationary over time because some drugs are prescribed more often during some time periods. Therefore, it will be interesting to observe the dynamics of associations over time. The measures of association signals such as Proportional Reporting Ratio, Reporting Odds Ratio, and Relative Risk can be calculated for each of FAERS submissions and the resulted time series can be tested for trend and seasonality. There has been no study in the literature that looked at the problem from this point of view.

REFERENCES

[1] U.S Food and Drugs Administration. (2017, November 14). Questions and Answers on FDA's Adverse Event Reporting System (FAERS). Retrieved from http://www.fda.gov/cder/aers/default.htm

[2] Szarfman, A., Machado, S. G., & O'neill, R. T. (2002). Use of screening algorithms and computer systems to efficiently signal higher-than-expected combinations of drugs and events in the US FDA's spontaneous reports database. Drug Safety, 25(6), 381-392.

[3] Wilson, A. M., Thabane, L., & Holbrook, A. (2004). Application of data mining techniques in pharmacovigilance. British journal of clinical pharmacology, 57(2), 127-134.

[4] Fram, D. M., Almenoff, J. S., & DuMouchel, W. (2003, August). Empirical Bayesian data mining for discovering patterns in post-marketing drug safety. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 359-368). ACM.

[5] Daly, A. K. (2013). Pharmacogenomics of adverse drug reactions. Genome medicine, 5(1), 5.

[6] Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining association rules between sets of items in large databases. In Acm sigmod record (Vol. 22, No. 2, pp. 207-216). ACM.

[7] Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).

[8] Han, J., Pei, J., & Yin, Y. (2000, May). Mining frequent patterns without candidate generation. In ACM Sigmod Record (Vol. 29, No. 2, pp. 1-12). ACM.

[9] Silverstein, C., Brin, S., & Motwani, R. (1998). Beyond market baskets: Generalizing association rules to dependence rules. Data mining and knowledge discovery, 2(1), 39-68.

[10] Aggarwal, C. C., & Yu, P. S. (1998, May). A new framework for itemset generation. In Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (pp. 18-24). ACM.

[11] Evans, S. J. W., Waller, P. C., & Davis, S. (2001). Use of proportional reporting ratios (PRRs) for signal generation from spontaneous adverse drug reaction reports. Pharmacoepidemiology and drug safety, 10(6), 483-486.

[12] Rothman, K. J., Lanes, S., & Sacks, S. T. (2004). The reporting odds ratio and its advantages over the proportional reporting ratio. Pharmacoepidemiology and drug safety, 13(8), 519-523.

[13] Waller, P., Van Puijenbroek, E. P., Egberts, A. C. G., & Evans, S. (2004). The reporting odds ratio versus the proportional reporting ratio:'deuce'. Pharmacoepidemiology and drug safety, 13(8), 525-526.

[14] Silverstein, C., Brin, S., & Motwani, R. (1998). Beyond market baskets: Generalizing association rules to dependence rules. Data mining and knowledge discovery, 2(1), 39-68.

[15] Stock, J. H., & Yogo, M. (2002). Testing for weak instruments in linear IV regression.

[16] Durbin, J. (1973). Weak convergence of the sample distribution function when parameters are estimated. The Annals of Statistics, 279-290.

[17] Zorych, I., Madigan, D., Ryan, P., & Bate, A. (2013). Disproportionality methods for pharmacovigilance in longitudinal observational databases. Statistical methods in medical research, 22(1), 39-56.

[18] Lu, Z. (2009). Information technology in pharmacovigilance: Benefits, challenges, and future directions from industry perspectives. Drug, healthcare and patient safety, 1, 35.

[19] DuMouchel, W., & Pregibon, D. (2001, August). Empirical bayes screening for multi-item associations. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 67-76). ACM.

[20] DuMouchel, W. (1999). Bayesian data mining in large frequency tables, with an application to the FDA spontaneous reporting system. The American Statistician, 53(3), 177-190.

[21] Dubey, S. D. (1970). Compound gamma, beta and F distributions. Metrika, 16(1), 27-31.

[22] Gopalan, P., Hofman, J. M., & Blei, D. M. (2013). Scalable recommendation with poisson factorization. arXiv preprint arXiv:1311.1704.

[23] Szarfman, A., Machado, S. G., & O'neill, R. T. (2002). Use of screening algorithms and computer systems to efficiently signal higher-than-expected combinations of drugs and events in the US FDA's spontaneous reports database. Drug Safety, 25(6), 381-392.

[24] Wasserman, L. A. (2010). Bayesian Inference. In All of statistics: A concise course in statistical inference. New-York, NY: Springer.

[25] DuMouchel, W., Fram, D., Yang, X., Mahmoud, R. A., Grogg, A. L., Engelhart, L., & Ramaswamy, K. (2008). Antipsychotics, glycemic disorders, and life-threatening diabetic

events: a Bayesian data-mining analysis of the FDA adverse event reporting system (1968–2004). Annals of Clinical Psychiatry, 20(1), 21-31.

[26] van der Ploeg, T., Austin, P. C., & Steyerberg, E. W. (2014). Modern modelling techniques are data hungry: a simulation study for predicting dichotomous endpoints. BMC medical research methodology, 14(1), 137.

[27] Harpaz, R., DuMouchel, W., LePendu, P., Bauer-Mehren, A., Ryan, P., & Shah, N. H. (2013). Performance of Pharmacovigilance Signal-Detection Algorithms for the FDA Adverse Event Reporting System. Clinical Pharmacology & Therapeutics, 93(6), 539-546.

[28] DuMouchel, W., & Harpaz, R. (2012). Regression-adjusted GPS algorithm (RGPS). An Oracle White Paper November.

[29] Stang, P. E., Ryan, P. B., Racoosin, J. A., Overhage, J. M., Hartzema, A. G., Reich, C., ... & Woodcock, J. (2010). Advancing the science for active surveillance: rationale and design for the Observational Medical Outcomes Partnership. Annals of internal medicine, 153(9), 600-606.

[30] Ryan, P. B., Madigan, D., Stang, P. E., Marc Overhage, J., Racoosin, J. A., & Hartzema, A. G. (2012). Empirical assessment of methods for risk identification in healthcare data: results from the experiments of the Observational Medical Outcomes Partnership. Statistics in medicine, 31(30), 4401-4415.

[31] Ryan, P. B., Schuemie, M. J., Welebob, E., Duke, J., Valentine, S., & Hartzema, A. G. (2013). Defining a reference set to support methodological research in drug safety. Drug safety, 36(1), 33-47.

[32] Bate, A., Lindquist, M., Edwards, I. R., Olsson, S., Orre, R., Lansner, A., & De Freitas, R. M. (1998). A Bayesian neural network method for adverse drug reaction signal generation. European journal of clinical pharmacology, 54(4), 315-321.

[33] Hipp, J., Güntzer, U., & Nakhaeizadeh, G. (2000). Algorithms for association rule mining—a general survey and comparison. ACM sigkdd explorations newsletter, 2(1), 58-64.

[34] Banda, J. M., Evans, L., Vanguri, R. S., Tatonetti, N. P., Ryan, P. B., & Shah, N. H. (2016). A curated and standardized adverse drug event resource to accelerate drug safety research. Scientific data, 3, 160026.

[35] Lansner, A., & Holst, A. (1996). A higher order Bayesian neural network with spiking units. International Journal of Neural Systems, 7(02), 115-128.

[36] Bate, A., Lindquist, M., Edwards, I. R., Olsson, S., Orre, R., Lansner, A., & De Freitas, R. M. (1998). A Bayesian neural network method for adverse drug reaction signal generation. European journal of clinical pharmacology, 54(4), 315-321.

[37] Szymczak, S., Holzinger, E., Dasgupta, A., Malley, J. D., Molloy, A. M., Mills, J. L., ... & Bailey-Wilson, J. E. (2016). r2VIM: A new variable selection method for random forests in genome-wide association studies. BioData mining, 9(1), 7.

[38] Botta, V., Louppe, G., Geurts, P., & Wehenkel, L. (2014). Exploiting SNP correlations within random forest for genome-wide association studies. PloS one, 9(4), e93379.

[39] Goldstein, B. A., Polley, E. C., & Briggs, F. B. (2011). Random forests for genetic association studies. Statistical applications in genetics and molecular biology, 10(1).

[40] Kooperberg, C., & Ruczinski, I. (2005). Identifying interacting SNPs using Monte Carlo logic regression. Genetic epidemiology, 28(2), 157-170.

[41] Ruczinski, I., Kooperberg, C., & LeBlanc, M. L. (2004). Exploring interactions in high-dimensional genomic data: an overview of logic regression, with applications. Journal of Multivariate Analysis, 90(1), 178-195.

[42] Dinu, I., Mahasirimongkol, S., Liu, Q., Yanai, H., Eldin, N. S., Kreiter, E., ... & Yasui, Y. (2012). SNP-SNP interactions discovered by logic regression explain Crohn's disease genetics. PloS one, 7(10), e43035.

[43] Bland, J. M., & Altman, D. G. (2000). The odds ratio. Bmj, 320(7247), 1468.

[44] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). New York: springer.

[45] Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, pp. 337-387). New York: Springer series in statistics.

[46] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[47] McKinney, B. A., Reif, D. M., Ritchie, M. D., & Moore, J. H. (2006). Machine learning for detecting gene-gene interactions. Applied bioinformatics, 5(2), 77-88.

[48] Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika, 82(4), 711-732.

[49] Liu, C., Ackerman, H. H., & Carulli, J. P. (2011). A genome-wide screen of gene–gene interactions for rheumatoid arthritis susceptibility. Human genetics, 129(5), 473-485.

[50] Ishwaran, H., Kogalur, U. B., Gorodeski, E. Z., Minn, A. J., & Lauer, M. S. (2010). High-dimensional variable selection for survival data. Journal of the American Statistical Association, 105(489), 205-217.

[51] US Food & Drug Administration. (2018, February 5). FDA Adverse Event Reporting System (FAERS): Latest Quarterly Data Files. Retrieved from https://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/ucm082193.htm

[52] Ahmed, I., & Poncet, A. (2013). PhViD: an R package for pharmacovigilance signal detection. R package version, 1(6), 2014.

[53] A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18--22.

[54] Kooperberg, C., & Ruczinski, I. (2011). LogicReg: Logic Regression. R package version, 1(10).

[55] Jennrich, R. I., & Sampson, P. F. (1968). Application of stepwise regression to non-linear estimation. Technometrics, 10(1), 63-72.

[56] Witte, J. S., & Fijal, B. A. (2001). Introduction: analysis of sequence data and population structure. Genetic Epidemiology, 21(S1).

[57] Schnabel, R. B., Koonatz, J. E., & Weiss, B. E. (1985). A modular system of algorithms for unconstrained minimization. ACM Transactions on Mathematical Software (TOMS), 11(4), 419-440.

[58] Dennis, J. E. (1983). RB Schnabel Numerical Methods for Unconstrained Optimization and Nonlinear Equations Prentice-Hall. New Jersey.

Appendix A: OMOP Gold Standard List

Drug	Adverse Event	Classify	Count in Data
acyclovir	Acute Kidney Injury	Positive	74
hydrochlorothiazide	Acute Kidney Injury	Positive	129
ibuprofen	Acute Kidney Injury	Positive	212
lisinopril	Acute Kidney Injury	Positive	102
meloxicam	Acute Kidney Injury	Positive	54
naproxen	Acute Kidney Injury	Positive	54
olmesartan medoxomil	Acute Kidney Injury	Positive	38
allopurinol	Acute Kidney Injury	Positive	147
candesartan	Acute Kidney Injury	Positive	27
capreomycin	Acute Kidney Injury	Positive	9
captopril	Acute Kidney Injury	Positive	3
chlorothiazide	Acute Kidney Injury	Positive	129
cyclosporine	Acute Kidney Injury	Positive	62
diflunisal	Acute Kidney Injury	Positive	0
enalaprilat	Acute Kidney Injury	Positive	0
etodolac	Acute Kidney Injury	Positive	3
fenoprofen	Acute Kidney Injury	Positive	0
ketoprofen	Acute Kidney Injury	Positive	16
ketorolac	Acute Kidney Injury	Positive	3
mefenamate	Acute Kidney Injury	Positive	0
moexipril	Acute Kidney Injury	Positive	0
oxaprozin	Acute Kidney Injury	Positive	7
piroxicam	Acute Kidney Injury	Positive	5
Telmisartan	Acute Kidney Injury	Positive	40
Benzonatate	Acute Kidney Injury	Negative	4
ketoconazole	Acute Kidney Injury	Negative	21
loratadine	Acute Kidney Injury	Negative	42
metaxalone	Acute Kidney Injury	Negative	2
temazepam	Acute Kidney Injury	Negative	31
acarbose	Acute Kidney Injury	Negative	11
adenosine	Acute Kidney Injury	Negative	0
almotriptan	Acute Kidney Injury	Negative	0
amylases	Acute Kidney Injury	Negative	0
benzocaine	Acute Kidney Injury	Negative	0
bromfenac	Acute Kidney Injury	Negative	0
chlorambucil	Acute Kidney Injury	Negative	2

chlorazepate	Acute Kidney Injury	Negative	0
clozapine	Acute Kidney Injury	Negative	37
cosyntropin	Acute Kidney Injury	Negative	0
dacarbazine	Acute Kidney Injury	Negative	2
darbepoetin alfa	Acute Kidney Injury	Negative	6
darifenacin	Acute Kidney Injury	Negative	1
darunavir	Acute Kidney Injury	Negative	29
dicyclomine	Acute Kidney Injury	Negative	4
disulfiram	Acute Kidney Injury	Negative	12
eletriptan	Acute Kidney Injury	Negative	1
endopeptidases	Acute Kidney Injury	Negative	0
entecavir	Acute Kidney Injury	Negative	14
ergotamine	Acute Kidney Injury	Negative	0
ferrous gluconate	Acute Kidney Injury	Negative	3
flavoxate	Acute Kidney Injury	Negative	2
flutamide	Acute Kidney Injury	Negative	2
frovatriptan	Acute Kidney Injury	Negative	5
gatifloxacin	Acute Kidney Injury	Negative	0
griseofulvin	Acute Kidney Injury	Negative	0
hyoscyamine	Acute Kidney Injury	Negative	0
imipramine	Acute Kidney Injury	Negative	2
infliximab	Acute Kidney Injury	Negative	80
ketotifen	Acute Kidney Injury	Negative	0
lactulose	Acute Kidney Injury	Negative	41
lipase	Acute Kidney Injury	Negative	2
mebendazole	Acute Kidney Injury	Negative	1
methenamine	Acute Kidney Injury	Negative	0
methocarbamol	Acute Kidney Injury	Negative	3
miconazole	Acute Kidney Injury	Negative	1
nelfinavir	Acute Kidney Injury	Negative	6
neostigmine	Acute Kidney Injury	Negative	0
nortriptyline	Acute Kidney Injury	Negative	11
orlistat	Acute Kidney Injury	Negative	11
paromomycin	Acute Kidney Injury	Negative	0
penicillin V	Acute Kidney Injury	Negative	8
phentermine	Acute Kidney Injury	Negative	4
phentolamine	Acute Kidney Injury	Negative	0
prilocaine	Acute Kidney Injury	Negative	4
primidone	Acute Kidney Injury	Negative	2

prochlorperazine	Acute Kidney Injury	Negative	7
ramelteon	Acute Kidney Injury	Negative	6
rizatriptan	Acute Kidney Injury	Negative	0
scopolamine	Acute Kidney Injury	Negative	11
simethicone	Acute Kidney Injury	Negative	1
sodium phosphate, monobasic	Acute Kidney Injury	Negative	2
tetrahydrocannabinol	Acute Kidney Injury	Negative	0
thiabendazole	Acute Kidney Injury	Negative	3
thiothixene	Acute Kidney Injury	Negative	0
tinidazole	Acute Kidney Injury	Negative	0
urea	Acute Kidney Injury	Negative	4
vitamin A	Acute Kidney Injury	Negative	4
zafirlukast	Acute Kidney Injury	Negative	0
allopurinol	Acute Liver Injury	Positive	147
carbamazepine	Acute Liver Injury	Positive	22
celecoxib	Acute Liver Injury	Positive	35
ciprofloxacin	Acute Liver Injury	Positive	32
cyclosporine	Acute Liver Injury	Positive	62
diltiazem	Acute Liver Injury	Positive	15
erythromycin	Acute Liver Injury	Positive	5
etodolac	Acute Liver Injury	Positive	3
fluconazole	Acute Liver Injury	Positive	30
ibuprofen	Acute Liver Injury	Positive	212
indomethacin	Acute Liver Injury	Positive	16
ketorolac	Acute Liver Injury	Positive	3
lamotriGastrointestinalne	Acute Liver Injury	Positive	16
levofloxacin	Acute Liver Injury	Positive	50
lisinopril	Acute Liver Injury	Positive	102
methotrexate	Acute Liver Injury	Positive	78
naproxen	Acute Liver Injury	Positive	54
niacin	Acute Liver Injury	Positive	8
nifedipine	Acute Liver Injury	Positive	35
nitrofurantoin	Acute Liver Injury	Positive	19
nortriptyline	Acute Liver Injury	Positive	11
ofloxacin	Acute Liver Injury	Positive	82
oxaprozin	Acute Liver Injury	Positive	7
pioglitazone	Acute Liver Injury	Positive	7
piroxicam	Acute Liver Injury	Positive	5
quinapril	Acute Liver Injury	Positive	1

ramipril	Acute Liver Injury	Positive	31
sulindac	Acute Liver Injury	Positive	1
tamoxifen	Acute Liver Injury	Positive	1
terbinafine	Acute Liver Injury	Positive	10
trandolapril	Acute Liver Injury	Positive	0
valproate	Acute Liver Injury	Positive	14
acetazolamide	Acute Liver Injury	Positive	1
abacavir	Acute Liver Injury	Positive	19
alatrofloxacin	Acute Liver Injury	Positive	0
bortezomib	Acute Liver Injury	Positive	2
bosentan	Acute Liver Injury	Positive	10
busulfan	Acute Liver Injury	Positive	1
captopril	Acute Liver Injury	Positive	3
caspofunGastrointestinaln	Acute Liver Injury	Positive	24
clozapine	Acute Liver Injury	Positive	37
dacarbazine	Acute Liver Injury	Positive	2
darunavir	Acute Liver Injury	Positive	29
didanosine	Acute Liver Injury	Positive	11
disulfiram	Acute Liver Injury	Positive	12
efavirenz	Acute Liver Injury	Positive	4
enalaprilat	Acute Liver Injury	Positive	0
felbamate	Acute Liver Injury	Positive	0
flutamide	Acute Liver Injury	Positive	2
gemcitabine	Acute Liver Injury	Positive	0
gemifloxacin	Acute Liver Injury	Positive	0
imatinib	Acute Liver Injury	Positive	13
infliximab	Acute Liver Injury	Positive	80
interferon beta-1a	Acute Liver Injury	Positive	7
isoniazid	Acute Liver Injury	Positive	28
itraconazole	Acute Liver Injury	Positive	3
lamivudine	Acute Liver Injury	Positive	38
methimazole	Acute Liver Injury	Positive	15
methyldopa	Acute Liver Injury	Positive	3
moexipril	Acute Liver Injury	Positive	0
nefazodone	Acute Liver Injury	Positive	0
nevirapine	Acute Liver Injury	Positive	37
norfloxacin	Acute Liver Injury	Positive	1
orlistat	Acute Liver Injury	Positive	11
penicillamine	Acute Liver Injury	Positive	0

posaconazole	Acute Liver Injury	Positive	12
propylthiouracil	Acute Liver Injury	Positive	6
rifampin	Acute Liver Injury	Positive	13
stavudine	Acute Liver Injury	Positive	2
sulfisoxazole	Acute Liver Injury	Positive	0
tenofovir	Acute Liver Injury	Positive	38
thiabendazole	Acute Liver Injury	Positive	3
thioguanine	Acute Liver Injury	Positive	0
tipranavir	Acute Liver Injury	Positive	12
tolcapone	Acute Liver Injury	Positive	0
tolmetin	Acute Liver Injury	Positive	0
trovafloxacin	Acute Liver Injury	Positive	0
voriconazole	Acute Liver Injury	Positive	14
zafirlukast	Acute Liver Injury	Positive	0
zalcitabine	Acute Liver Injury	Positive	0
zidovudine	Acute Liver Injury	Positive	42
adenosine	Acute Liver Injury	Negative	0
benzocaine	Acute Liver Injury	Negative	0
benzonatate	Acute Liver Injury	Negative	4
dicyclomine	Acute Liver Injury	Negative	4
fluticasone	Acute Liver Injury	Negative	35
gatifloxacin	Acute Liver Injury	Negative	0
griseofulvin	Acute Liver Injury	Negative	0
hyoscyamine	Acute Liver Injury	Negative	0
lactulose	Acute Liver Injury	Negative	41
miconazole	Acute Liver Injury	Negative	1
oxybutynin	Acute Liver Injury	Negative	3
penicillin V	Acute Liver Injury	Negative	8
salmeterol	Acute Liver Injury	Negative	10
scopolamine	Acute Liver Injury	Negative	11
sitagliptin	Acute Liver Injury	Negative	22
Sucralfate	Acute Liver Injury	Negative	7
almotriptan	Acute Liver Injury	Negative	0
amylases	Acute Liver Injury	Negative	0
cosyntropin	Acute Liver Injury	Negative	0
droperidol	Acute Liver Injury	Negative	0
endopeptidases	Acute Liver Injury	Negative	0
ergotamine	Acute Liver Injury	Negative	0
ferrous gluconate	Acute Liver Injury	Negative	3

flavoxate	Acute Liver Injury	Negative	2
ketotifen	Acute Liver Injury	Negative	0
lipase	Acute Liver Injury	Negative	2
lithium citrate	Acute Liver Injury	Negative	0
Methenamine	Acute Liver Injury	Negative	0
Neostigmine	Acute Liver Injury	Negative	0
Paromomycin	Acute Liver Injury	Negative	0
Phentermine	Acute Liver Injury	Negative	4
Phentolamine	Acute Liver Injury	Negative	0
Primidone	Acute Liver Injury	Negative	2
Propantheline	Acute Liver Injury	Negative	0
Sodium Phosphate, Monobasic	Acute Liver Injury	Negative	2
Tetrahydrocannabinol	Acute Liver Injury	Negative	0
Tinidazole	Acute Liver Injury	Negative	0
amlodipine	Acute Myocardial Infarction	Positive	222
darbepoetin alfa	Acute Myocardial Infarction	Positive	31
dipyridamole	Acute Myocardial Infarction	Positive	8
epoetin Alfa	Acute Myocardial Infarction	Positive	33
estradiol	Acute Myocardial Infarction	Positive	18
estrogens, conjugated	Acute Myocardial Infarction	Positive	5
etodolac	Acute Myocardial Infarction	Positive	3
indomethacin	Acute Myocardial Infarction	Positive	3
ketorolac	Acute Myocardial Infarction	Positive	3
nabumetone	Acute Myocardial Infarction	Positive	3
nifedipine	Acute Myocardial Infarction	Positive	81
nortriptyline	Acute Myocardial Infarction	Positive	7
oxaprozin	Acute Myocardial Infarction	Positive	0
piroxicam	Acute Myocardial Infarction	Positive	8
sulindac	Acute Myocardial Infarction	Positive	3
sumatriptan	Acute Myocardial Infarction	Positive	18
almotriptan	Acute Myocardial Infarction	Positive	0
amoxapine	Acute Myocardial Infarction	Positive	0
bromocriptine	Acute Myocardial Infarction	Positive	4
desipramine	Acute Myocardial Infarction	Positive	0
diflunisal	Acute Myocardial Infarction	Positive	0
eletriptan	Acute Myocardial Infarction	Positive	2
enalaprilat	Acute Myocardial Infarction	Positive	0
estropipate	Acute Myocardial Infarction	Positive	0
factor VIIa	Acute Myocardial Infarction	Positive	0

fenoprofen	Acute Myocardial Infarction	Positive	0
flurbiprofen	Acute Myocardial Infarction	Positive	1
frovatriptan	Acute Myocardial Infarction	Positive	1
imipramine	Acute Myocardial Infarction	Positive	1
ketoprofen	Acute Myocardial Infarction	Positive	11
moexipril	Acute Myocardial Infarction	Positive	0
naratriptan	Acute Myocardial Infarction	Positive	0
rizatriptan	Acute Myocardial Infarction	Positive	1
salsalate	Acute Myocardial Infarction	Positive	0
tolmetin	Acute Myocardial Infarction	Positive	0
zolmitriptan	Acute Myocardial Infarction	Positive	3
benzonatate	Acute Myocardial Infarction	Negative	8
clindamycin	Acute Myocardial Infarction	Negative	5
dicyclomine	Acute Myocardial Infarction	Negative	2
fluticasone	Acute Myocardial Infarction	Negative	28
gatifloxacin	Acute Myocardial Infarction	Negative	4
hyoscyamine	Acute Myocardial Infarction	Negative	1
ketoconazole	Acute Myocardial Infarction	Negative	2
lactulose	Acute Myocardial Infarction	Negative	12
loratadine	Acute Myocardial Infarction	Negative	43
metaxalone	Acute Myocardial Infarction	Negative	3
methocarbamol	Acute Myocardial Infarction	Negative	6
penicillin V	Acute Myocardial Infarction	Negative	2
prochlorperazine	Acute Myocardial Infarction	Negative	19
oxybutynin	Acute Myocardial Infarction	Negative	8
ramelteon	Acute Myocardial Infarction	Negative	5
salmeterol	Acute Myocardial Infarction	Negative	17
scopolamine	Acute Myocardial Infarction	Negative	1
sitagliptin	Acute Myocardial Infarction	Negative	40
sucralfate	Acute Myocardial Infarction	Negative	12
temazepam	Acute Myocardial Infarction	Negative	13
terbinafine	Acute Myocardial Infarction	Negative	8
urea	Acute Myocardial Infarction	Negative	7
acarbose	Acute Myocardial Infarction	Negative	3
acetazolamide	Acute Myocardial Infarction	Negative	1
amylases	Acute Myocardial Infarction	Negative	0
bromfenac	Acute Myocardial Infarction	Negative	0
chlorambucil	Acute Myocardial Infarction	Negative	6
chlorazepate	Acute Myocardial Infarction	Negative	0

chlorothiazide	Acute Myocardial Infarction	Negative	196
cosyntropin	Acute Myocardial Infarction	Negative	0
darifenacin	Acute Myocardial Infarction	Negative	2
didanosine	Acute Myocardial Infarction	Negative	0
droperidol	Acute Myocardial Infarction	Negative	0
endopeptidases	Acute Myocardial Infarction	Negative	0
entecavir	Acute Myocardial Infarction	Negative	2
ferrous gluconate	Acute Myocardial Infarction	Negative	11
flavoxate	Acute Myocardial Infarction	Negative	0
flutamide	Acute Myocardial Infarction	Negative	0
ketotifen	Acute Myocardial Infarction	Negative	0
lipase	Acute Myocardial Infarction	Negative	1
lithium citrate	Acute Myocardial Infarction	Negative	0
mebendazole	Acute Myocardial Infarction	Negative	0
methenamine	Acute Myocardial Infarction	Negative	0
methimazole	Acute Myocardial Infarction	Negative	4
miconazole	Acute Myocardial Infarction	Negative	2
nelfinavir	Acute Myocardial Infarction	Negative	0
nevirapine	Acute Myocardial Infarction	Negative	7
paromomycin	Acute Myocardial Infarction	Negative	0
pemoline	Acute Myocardial Infarction	Negative	0
penicillamine	Acute Myocardial Infarction	Negative	1
posaconazole	Acute Myocardial Infarction	Negative	4
prilocaine	Acute Myocardial Infarction	Negative	4
primidone	Acute Myocardial Infarction	Negative	3
propantheline	Acute Myocardial Infarction	Negative	0
simethicone	Acute Myocardial Infarction	Negative	1
sodiumphosphate, monobasic	Acute Myocardial Infarction	Negative	0
stavudine	Acute Myocardial Infarction	Negative	1
sulfasalazine	Acute Myocardial Infarction	Negative	15
sulfisoxazole	Acute Myocardial Infarction	Negative	0
tetrahydrocannabinol	Acute Myocardial Infarction	Negative	0
thiabendazole	Acute Myocardial Infarction	Negative	0
thiothixene	Acute Myocardial Infarction	Negative	1
tinidazole	Acute Myocardial Infarction	Negative	0
tipranavir	Acute Myocardial Infarction	Negative	1
vitamin A	Acute Myocardial Infarction	Negative	2
zafirlukast	Acute Myocardial Infarction	Negative	0
citalopram	Gastrointestinal Bleed	Positive	4

clindamycin	Gastrointestinal Bleed	Positive	0
clopidogrel	Gastrointestinal Bleed	Positive	1
escitalopram	Gastrointestinal Bleed	Positive	0
etodolac	Gastrointestinal Bleed	Positive	0
fluoxetine	Gastrointestinal Bleed	Positive	1
ibuprofen	Gastrointestinal Bleed	Positive	1
indomethacin	Gastrointestinal Bleed	Positive	0
ketorolac	Gastrointestinal Bleed	Positive	0
meloxicam	Gastrointestinal Bleed	Positive	0
nabumetone	Gastrointestinal Bleed	Positive	0
naproxen	Gastrointestinal Bleed	Positive	5
piroxicam	Gastrointestinal Bleed	Positive	0
potassium Chloride	Gastrointestinal Bleed	Positive	4
sertraline	Gastrointestinal Bleed	Positive	0
oxaprozin	Gastrointestinal Bleed	Positive	0
diflunisal	Gastrointestinal Bleed	Positive	0
fenoprofen	Gastrointestinal Bleed	Positive	0
flurbiprofen	Gastrointestinal Bleed	Positive	0
ketoprofen	Gastrointestinal Bleed	Positive	0
mefenamate	Gastrointestinal Bleed	Positive	0
sulindac	Gastrointestinal Bleed	Positive	0
tolmetin	Gastrointestinal Bleed	Positive	0
valdecoxib	Gastrointestinal Bleed	Positive	0
adenosine	Gastrointestinal Bleed	Negative	1
benzonatate	Gastrointestinal Bleed	Negative	1
dicyclomine	Gastrointestinal Bleed	Negative	0
epoetin alfa	Gastrointestinal Bleed	Negative	0
fluticasone	Gastrointestinal Bleed	Negative	3
hyoscyamine	Gastrointestinal Bleed	Negative	0
ketoconazole	Gastrointestinal Bleed	Negative	0
lactulose	Gastrointestinal Bleed	Negative	1
loratadine	Gastrointestinal Bleed	Negative	0
metaxalone	Gastrointestinal Bleed	Negative	0
methocarbamol	Gastrointestinal Bleed	Negative	0
nitrofurantoin	Gastrointestinal Bleed	Negative	0
oxybutynin	Gastrointestinal Bleed	Negative	0
penicillin V	Gastrointestinal Bleed	Negative	1
pioglitazone	Gastrointestinal Bleed	Negative	3
prochlorperazine	Gastrointestinal Bleed	Negative	0

rosiglitazone	Gastrointestinal Bleed	Negative	0
salmeterol	Gastrointestinal Bleed	Negative	2
scopolamine	Gastrointestinal Bleed	Negative	0
sitagliptin	Gastrointestinal Bleed	Negative	0
sucralfate	Gastrointestinal Bleed	Negative	1
temazepam	Gastrointestinal Bleed	Negative	0
terbinafine	Gastrointestinal Bleed	Negative	0
urea	Gastrointestinal Bleed	Negative	0
abacavir	Gastrointestinal Bleed	Negative	0
acarbose	Gastrointestinal Bleed	Negative	0
amylases	Gastrointestinal Bleed	Negative	0
benzocaine	Gastrointestinal Bleed	Negative	0
bromfenac	Gastrointestinal Bleed	Negative	0
chlorambucil	Gastrointestinal Bleed	Negative	0
chlorazepate	Gastrointestinal Bleed	Negative	0
cosyntropin	Gastrointestinal Bleed	Negative	0
dacarbazine	Gastrointestinal Bleed	Negative	0
darifenacin	Gastrointestinal Bleed	Negative	0
disulfiram	Gastrointestinal Bleed	Negative	0
droperidol	Gastrointestinal Bleed	Negative	0
endopeptidases	Gastrointestinal Bleed	Negative	0
entecavir	Gastrointestinal Bleed	Negative	0
ergotamine	Gastrointestinal Bleed	Negative	0
ferrous gluconate	Gastrointestinal Bleed	Negative	0
griseofulvin	Gastrointestinal Bleed	Negative	0
itraconazole	Gastrointestinal Bleed	Negative	0
ketotifen	Gastrointestinal Bleed	Negative	0
lamivudine	Gastrointestinal Bleed	Negative	0
lipase	Gastrointestinal Bleed	Negative	0
lithium citrate	Gastrointestinal Bleed	Negative	0
mebendazole	Gastrointestinal Bleed	Negative	0
miconazole	Gastrointestinal Bleed	Negative	0
moexipril	Gastrointestinal Bleed	Negative	0
neostigmine	Gastrointestinal Bleed	Negative	0
nevirapine	Gastrointestinal Bleed	Negative	0
orlistat	Gastrointestinal Bleed	Negative	3
paromomycin	Gastrointestinal Bleed	Negative	0
pemoline	Gastrointestinal Bleed	Negative	0
phentermine	Gastrointestinal Bleed	Negative	0

phentolamine	Gastrointestinal Bleed	Negative	0
prilocaine	Gastrointestinal Bleed	Negative	0
propantheline	Gastrointestinal Bleed	Negative	0
simethicone	Gastrointestinal Bleed	Negative	0
stavudine	Gastrointestinal Bleed	Negative	0
tetrahydrocannabinol	Gastrointestinal Bleed	Negative	0
thiabendazole	Gastrointestinal Bleed	Negative	0
thiothixene	Gastrointestinal Bleed	Negative	0
tinidazole	Gastrointestinal Bleed	Negative	0
vitamin A	Gastrointestinal Bleed	Negative	0
zidovudine	Gastrointestinal Bleed	Negative	0

Appendix B: RGPS Code

This code is maintained and updated at https://github.com/minh2182000/RGPS.

```
library(PhViD)
RGPS =
function (formula, data,
          RR0 = 1, MIN.n11 = 1, DECISION = 1, DECISION.THRES = 0.05,
          RANKSTAT = 1, TRONC = FALSE, TRONC.THRES = 1, PRIOR.INIT = c(alpha1
= 0.2, beta1 = 0.06, alpha2 = 1.4, beta2 = 1.8, w = 0.1), PRIOR.PARAM = NULL)
{
  # - stepwise logistic reg
  formula = formula(lm(formula, data = data[1,]))
  logmodel = step(glm(as.formula(paste(all.vars(formula)[1], " ~ 1")),
                      family = binomial, data),
                  scope = formula, direction = "forward", trace = 0)
  chosen_vars = all.vars(formula(logmodel)[-1])
  beta = rep(NA, length(all.vars(formula)[-1])); names(beta) =
all.vars(formula)[-1]
  beta[chosen_vars] = coef(logmodel)[chosen_vars]
  beta[is.na(beta)] = 0

  # - calculate expectations -----
  E = rep(NA, length(all.vars(formula)[-1])); names(E) = all.vars(formula)[-
1]
  X = as.matrix(data[all.vars(formula)[-1]])
  for (j in 1:length(E)){
    var_name = names(E)[j]
    Xj = data[var_name]
    betaj = as.matrix(beta, ncol = 1); betaj[j] = 0
    mu = coef(logmodel)[1] + X%*%betaj
    E[j] = sum(Xj / (1 + exp(-mu)))
  }

  # --------- recreate DATABASE -------
  count_table = data.frame(drug = all.vars(formula)[-1], AE =
all.vars(formula)[1], count = NA)
  for (i in 1:nrow(count_table)){
    count_table$count[i] = sum(data[as.character(count_table$drug[i])] *
data[as.character(count_table$AE[i])])
  }
  DATABASE = as.PhViD(count_table)
  #---------------GPS---------------------

  DATA <- DATABASE$data
  E = E[DATABASE$L$AE]
  N <- DATABASE$N
  L <- DATABASE$L
  n11 <- DATA[, 1]
  n1. <- DATA[, 2]
  n.1 <- DATA[, 3]

  P_OUT <- TRUE
  if (is.null(PRIOR.PARAM)) {
    P_OUT <- FALSE
    if (TRONC == FALSE) {
      data_cont <- xtabs(DATA[, 1] ~ L[, 1] + L[, 2])
      n1._mat <- apply(data_cont, 1, sum)
      n.1_mat <- apply(data_cont, 2, sum)
      n1._c <- rep(n1._mat, times = length(n.1_mat))
      n.1_c <- rep(n.1_mat, each = length(n1._mat))
      E_c <- E
      n11_c <- as.vector(data_cont)
      p_out <- suppressWarnings(nlminb(start = PRIOR.INIT, .lik2NB, n11 =
n11_c, E = E_c,
                                       control = list(iter.max = 500), lower
= c(0,0,0,0,0), upper = c(Inf,Inf,Inf,Inf,1)))
    }
    if (TRONC == TRUE) {
      tronc <- TRONC.THRES - 1
      p_out <- suppressWarnings(nlm(.likTronc2NB, p = PRIOR.INIT,
                                    n11 = n11[n11 >= TRONC.THRES], E = E[n11
>=

TRONC.THRES], tronc, iterlim = 500))
    }
    PRIOR.PARAM <- p_out$par
    code.convergence <- p_out$convergence
  }
  if (MIN.n11 > 1) {
    E <- E[n11 >= MIN.n11]
    n1. <- n1.[n11 >= MIN.n11]
    n.1 <- n.1[n11 >= MIN.n11]
    LL <- data.frame(drugs = L[, 1], events = L[, 2], n11)
    LL1 <- LL[, 1][n11 >= MIN.n11]
    LL2 <- LL[, 2][n11 >= MIN.n11]
    rm(list = "L")
    L <- data.frame(LL1, LL2)
    n11 <- n11[n11 >= MIN.n11]
  }
  Nb.Cell <- length(n11)
  post.H0 <- vector(length = Nb.Cell)
  Q <- PRIOR.PARAM[5] * dnbinom(n11, size = PRIOR.PARAM[1],
                                prob = PRIOR.PARAM[2]/(PRIOR.PARAM[2] +
E))/(PRIOR.PARAM[5] *

dnbinom(n11, size = PRIOR.PARAM[1], prob = PRIOR.PARAM[2]/(PRIOR.PARAM[2] +

E)) + (1 - PRIOR.PARAM[5]) * dnbinom(n11, size = PRIOR.PARAM[3],

prob = PRIOR.PARAM[4]/(PRIOR.PARAM[4] + E)))
  post.H0 <- Q * pgamma(RR0, PRIOR.PARAM[1] + n11, PRIOR.PARAM[2] +
                          E) + (1 - Q) * pgamma(RR0, PRIOR.PARAM[3] + n11,
PRIOR.PARAM[4] +
                                                  E)
  postE <- log(2)^(-1) * (Q * (digamma(PRIOR.PARAM[1] + n11) -
                                 log(PRIOR.PARAM[2] + E)) + (1 - Q) *
(digamma(PRIOR.PARAM[3] +

n11) - log(PRIOR.PARAM[4] + E)))
  LB <- .QuantileDuMouchel(0.05, Q, PRIOR.PARAM[1] + n11,
                           PRIOR.PARAM[2] + E, PRIOR.PARAM[3] + n11,
PRIOR.PARAM[4] +
                             E)
  if (RANKSTAT == 1)
    RankStat <- post.H0
  if (RANKSTAT == 2)
    RankStat <- LB
  if (RANKSTAT == 3)
    RankStat <- postE
  if (RANKSTAT == 1) {
    FDR <- (cumsum(post.H0[order(RankStat)])/(1:length(post.H0)))
    FNR <- rev(cumsum((1 - post.H0)[order(1 - RankStat)]))/(Nb.Cell -

1:length(post.H0))
    Se <- cumsum((1 - post.H0)[order(RankStat)])/(sum(1 -
                                                        post.H0))
    Sp <- rev(cumsum(post.H0[order(1 - RankStat)]))/(Nb.Cell -
                                                       sum(1 - post.H0))
  }
  if (RANKSTAT == 2 | RANKSTAT == 3) {
    FDR <- (cumsum(post.H0[order(RankStat, decreasing =
TRUE)])/(1:length(post.H0)))
    FNR <- rev(cumsum((1 - post.H0)[order(1 - RankStat,
                                          decreasing = TRUE)]))/(Nb.Cell -
1:length(post.H0))
    Se <- cumsum((1 - post.H0)[order(RankStat, decreasing = TRUE)])/(sum(1 -

post.H0))
    Sp <- rev(cumsum(post.H0[order(1 - RankStat, decreasing =
TRUE)]))/(Nb.Cell -

sum(1 - post.H0))
  }
  if (DECISION == 1)
    Nb.signaux <- sum(FDR <= DECISION.THRES)
  if (DECISION == 2)
    Nb.signaux <- min(DECISION.THRES, Nb.Cell)
  if (DECISION == 3) {
    if (RANKSTAT == 1)
      Nb.signaux <- sum(RankStat <= DECISION.THRES, na.rm = TRUE)
    if (RANKSTAT == 2 | RANKSTAT == 3)
      Nb.signaux <- sum(RankStat >= DECISION.THRES, na.rm = TRUE)
  }
  RES <- vector(mode = "list")
  RES$INPUT.PARAM <- data.frame(RR0, MIN.n11, DECISION, DECISION.THRES,
                                RANKSTAT, TRONC, TRONC.THRES)
  RES$PARAM <- vector(mode = "list")
  if (P_OUT == TRUE)
    RES$PARAM$PRIOR.PARAM <- data.frame(PRIOR.PARAM)
  if (P_OUT == FALSE) {
    RES$PARAM$PRIOR.INIT <- data.frame(PRIOR.INIT)
    RES$PARAM$PRIOR.PARAM <- PRIOR.PARAM
    RES$PARAM$CONVERGENCE <- code.convergence
  }
  if (RANKSTAT == 1) {
    RES$ALLSIGNALS <- data.frame(L[, 1][order(RankStat)],
                                 L[, 2][order(RankStat)],
n11[order(RankStat)], E[order(RankStat)],
                                 RankStat[order(RankStat)],
(n11/E)[order(RankStat)],
                                 n1.[order(RankStat)], n.1[order(RankStat)],
FDR,
                                 FNR, Se, Sp)
    colnames(RES$ALLSIGNALS) <- c("drug", "event", "count",
                                  "expected count", "postH0", "n11/E", "drug
margin",
                                  "event margin", "FDR", "FNR", "Se", "Sp")
  }
  if (RANKSTAT == 2 | RANKSTAT == 3) {
    RES$ALLSIGNALS <- data.frame(L[, 1][order(RankStat,
                                              decreasing = TRUE)], L[,
2][order(RankStat, decreasing = TRUE)],
                                 n11[order(RankStat, decreasing = TRUE)],
E[order(RankStat,

decreasing = TRUE)], RankStat[order(RankStat,

decreasing = TRUE)], (n11/E)[order(RankStat,

decreasing = TRUE)], n1.[order(RankStat, decreasing = TRUE)],
                                 n.1[order(RankStat, decreasing = TRUE)],
FDR, FNR,
                                 Se, Sp, post.H0[order(RankStat, decreasing =
TRUE)])
    if (RANKSTAT == 2)
      colnames(RES$ALLSIGNALS) <- c("drug", "event", "count",
                                    "expected count", "Q_0.05(lambda)",
"n11/E",
                                    "drug margin", "event margin", "FDR",
"FNR",
                                    "Se", "Sp", "postH0")
    if (RANKSTAT == 3)
      colnames(RES$ALLSIGNALS) <- c("drug", "event", "count",
                                    "expected count", "post E(Lambda)",
"n11/E",
                                    "drug margin", "event margin", "FDR",
"FNR",
                                    "Se", "Sp", "postH0")
  }
  RES$SIGNALS <- RES$ALLSIGNALS[1:Nb.signaux, ]
  RES$NB.SIGNALS <- Nb.signaux
  RES
}
```