KNOWLEDGE DISCOVERY WITH BAYESIAN RULE LEARNING METHODS FOR ACTIONABLE BIOMEDICINE

by

Jeya Balaji Balasubramanian

MS, Carnegie Mellon University, 2010

Submitted to the Graduate Faculty of

the School of Computing and Information in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2019

UNIVERSITY OF PITTSBURGH

SCHOOL OF COMPUTING AND INFORMATION

This dissertation was presented

by

Jeya Balaji Balasubramanian

It was defended on

August 27th 2019

and approved by

Dr. Vanathi Gopalakrishnan, Intelligent Systems Program, University of Pittsburgh

Dr. Gregory F. Cooper, Intelligent Systems Program, University of Pittsburgh

Dr. Shyam Visweswaran, Intelligent Systems Program, University of Pittsburgh

Dr. Steven E. Reis, Associate Vice Chancellor for Clinical Research, University of Pittsburgh

Dissertation Director: Dr. Vanathi Gopalakrishnan, Intelligent Systems Program, University of Pittsburgh

KNOWLEDGE DISCOVERY
WITH BAYESIAN RULE LEARNING METHODS
FOR ACTIONABLE BIOMEDICINE

Jeya Balaji Balasubramanian, PhD

University of Pittsburgh, 2019

Discovery of precise biomarkers are crucial for improved clinical diagnostic, prognostic, and therapeutic decision-making. They help improve our understanding of the underlying physiological (and pathophysiological processes) within an individual. To discover precise biomarkers, we must take a personalized medical approach that accounts for an individual's unique clinical, genetic, omic, and environmental information. The molecular-level omic information provides an opportunity to understand complex physiological processes at an unprecedented resolution. The reducing costs and improvements in high-throughput technologies, which collect omic data from an individual, has now made it feasible to include a person's omic information as a standard component to their medical record. This information can only be clinically actionable if it is understandable to a clinician and applicable in the correct medical context. Biomarker discovery from omic data is challenging because they are 1) *high-dimensional*, which increases the chance of false positive discoveries from traditional data mining methods; 2) most diseases are *multifactorial*, where many factors influence the disease outcome, making it challenging to be modeled by most data mining algorithms while keeping the model understandable to a clinician; and 3) traditional data mining methods discover only statistically significant biomarkers but do not account for *clinical relevance*, therefore they do not translate well in clinical practice.

In this dissertation, I formulate the problem of learning both statistically significant and clinically relevant biomarkers as a knowledge discovery problem. In computer science,

knowledge discovery in databases is "*a non-trivial process of the extraction of valid, novel, potentially useful, and ultimately understandable patterns in data*". Clinical practice guidelines in decision support systems are often presented as explicit propositional logic rules because they are easy for a clinician to understand and are often actionable instructions themselves. Bayesian rule learning (BRL) is a rule-learning classifier that learns patterns as a set of probabilistic classification rules. I develop BRL search to efficiently learn from high-dimensional data. I study different BRL model representations to help obtain a robust set of rules that can encode context-specific independencies found in the data. To help efficiently model multifactorial diseases, I study various ensemble methods with BRL, collectively called Ensemble Bayesian Rule Learning (EBRL). I also develop a novel ensemble model visualization method called Bayesian Rule Ensemble Visualization tool (BREVity) to make EBRL more human-readable for a researcher or a clinician. I develop BRL with informative priors (BRL_p) to enable BRL to incorporate prior domain knowledge into the model learning process, thereby further reducing the chance of discovering false positives. Finally, I develop BRL for knowledge discovery (BRL-KD) that can incorporate a clinical utility function to learn models that are clinically more relevant. Collectively, I use these BRL methods, developed for the task of biomarker discovery, as the knowledge engine of an intelligent clinical decision support system called Bayesian Rules for Actionable Informed Decisions or BRAID, a concept framework that can be deployed in clinical practice.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

Medical practice depends upon precise biomarkers to help better understand the underlying physiological (and pathophysiological) processes within an individual to make correct diagnostic, prognostic, and therapeutic decisions. A *biomarker* is "a characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacologic responses to a therapeutic intervention" [Group et al., 2001]. Biomarkers commonly used in medical practice include macroscopic factors such as— age, smoking, blood pressure, gender, family history of disease, etc. Biomarkers can also include microscopic, biomolecular factors, such as— cholesterol, blood glucose, harmful mutations in *BRCA* genes is a diagnostic biomarker because it substantially increases the risk of developing breast or ovarian cancer [Levy-Lahad and Friedman, 2007], a mutation in the *EGFR* gene can help predict if tumor cells will respond to therapeutic treatment using tyrosine kinase inhibitors [da Cunha Santos et al., 2011], an increase in *prostate-specific antigen* in blood serum of men helps predict an increased risk developing a prostatic disease [Catalona et al., 1991] etc. *Biomarker discovery* is the process of discovering novel biomarkers to help improve our understanding of the biological process under study.

Based on their functional relationship to the clinical outcome of interest, biomarkers can be classified into two types, namely— *predictive* and *mechanistic* biomarkers [Shortliffe and Cimino, 2013]. *Predictive biomarkers* are correlated to the clinical outcome. They may or may not be causal of the outcome. They are useful for decision support (e.g., finding a subpopulation at a high risk of developing a disease or behaving differently to standard therapy) and suggest focus for research (e.g., study its role in the biological system). *Mechanistic biomarkers*, on the other hand, are causes of the pathological condition, disease progression, or sensitivity to a given drug. They are potential candidates for interventions to either

activate or inhibit pathways relevant to the disease mechanism. In this dissertation, when I mention biomarkers, I am referring to predictive biomarkers.

Precise descriptions of biological processes can only be achieved with a *personalized medical* approach, which takes into account an individual's unique clinical, genetic, omic, and environmental information [Ginsburg and Willard, 2009]. The omic information includes information from the whole molecular mechanism within an individual, from the genome to its derivatives (RNA, proteins, and metabolites). These molecular fingerprints help identify and distinguish each individual's unique physiology. Omic information promises to improve our understanding of biological systems at an unprecedented resolution. Biomarkers derived from omic datasets can help realize the goals of personalized medicine.

High-throughput technologies are methods of automation for performing a large number of experiments in parallel. They can measure omics information from a large number of individuals (experiments). Since the completion of the Human Genome Project [Collins et al., 2003] in 2003, high-throughput sequencing technologies have undergone remarkable developments in terms of speed, resolution, and cost-efficiency [Mardis, 2011]. These developments have also made it feasible to include omic information as a standard component of an individual's medical record, thereby moving a step closer towards personalized medicine. To discover biomarkers associated with the clinical outcome of interest, from these complex omic datasets, we need data mining tools to efficiently discover patterns of such associations from the data. Particularly, *machine learning* methods offer promising solutions to learn models from these complex and voluminous datasets [Olson et al., 2017]. Omic datasets are high-dimensional, noisy, and usually collected from complex multifactorial disease processes making them challenging for analysis using traditional data mining methods. I describe these challenges and their effect on traditional data mining methods in Section 1.1.

Information learned from data analysis of omic datasets can only be clinically useful if it is ultimately understandable to a clinician and can be applied in a specific medical context. *Translational medicine* is a field that deals with the translation of biological knowledge discovered from basic scientific research (bench-side) to clinical practice (bed-side), which can eventually help improve population health. [Lenfant, 2003, Sung et al., 2003] identify two main barriers, T1 barrier and T2 barrier, which must be overcome, to successfully

translate newly discovered biological knowledge into clinical practice. I depict these barriers by modifying a figure from [Shortliffe and Cimino, 2013], as shown in Figure 1.

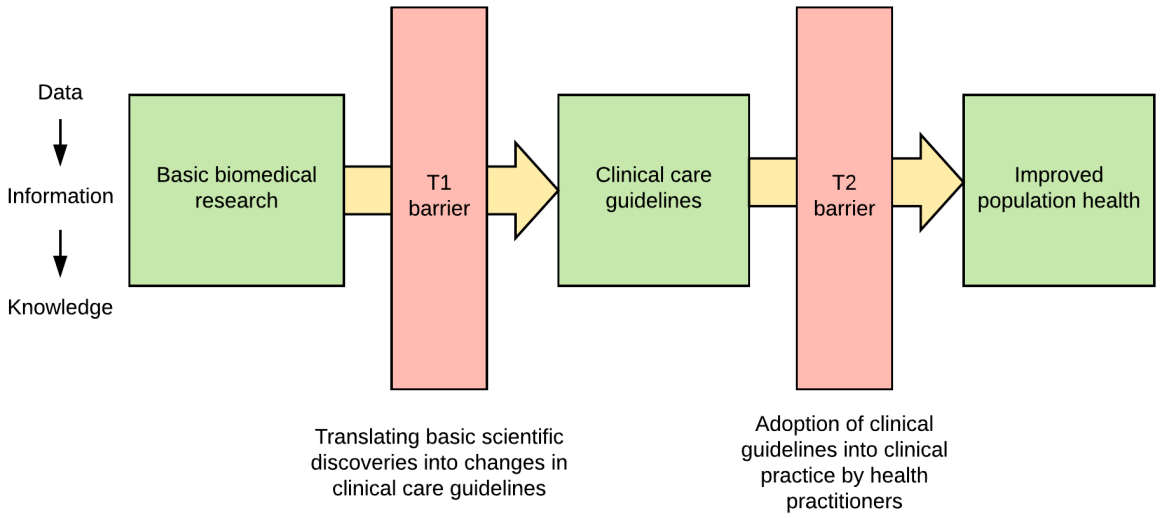


Figure 1: Translational medicine roadblocks. *Figure was obtained and modified from Figure 25.3 in* [Shortliffe and Cimino, 2013].

The T1 translational barrier comes from the hesitation in translating knowledge from new discoveries in biological research to changing clinical care guidelines. The T2 translational barrier comes from the hesitation in adopting the changed clinical guidelines into regular clinical practice by health practitioners. Currently, these barriers are long and can take about 10 to 20 years to overcome the T1 barrier alone. [Burke, 2016] reported in 2016, that there were 768,000 papers indexed in PubMed about biomarkers. Yet, despite all the technological advances in omics research, we are still very far from widespread clinical use of these omic biomarkers. Currently, there are only a few dozen clinically relevant cancer biomarkers [Selleck et al., 2017].

In data-driven sciences, there are three levels of informatics— *data*, *information*, and *knowledge*. *Data* is simply the raw observations recorded in a database. *Information* is giving meaning to the data by discovering patterns via data analysis methods. *Knowledge* is to interpret the information in a specific clinical context and to use that information to

guide actions. To help motivate crossing the T1 and T2 barriers, we must seek knowledge and not just information from our omic data analysis.

Clinical context comes from the validation of utility in the intended clinical application of the the discovered biomarkers. Traditional data analysis methods do not account for clinical utility. They seek information and not knowledge. Their focus is to find patterns with high statistical significance i.e., the certainty that the discovered pattern (of association between biomarker and the clinical outcome) is correct. Statistical significance is not clinical relevance. Clinical relevance is a measure of how useful the discovered pattern is in guiding clinical care. Clinical relevance, in addition to statistical significance, also accounts for clinical utility such as invasiveness, efficacy, safety, and cost of the discovered biomarkers [Shortliffe and Cimino, 2013, Selleck et al., 2017]. [Selleck et al., 2017] emphasize that it is not enough to identify variants but to identify *actionable* variants that have the potential to revolutionize healthcare. Traditional data analysis methods don't account for clinical relevance because it is hard to quantify a metric for clinical relevance. To reduce the translational medicine barriers, we need data analysis methods that not only discover patterns with high statistical significance but can also find those with high clinical relevance.

The data mining model that learns statistically significant and clinically relevant association patterns between the clinical outcome of interest and the biomarkers, must also ultimately be understandable to a clinician for it to be actionable in practice. A minimum requirement for the biomarker model to be routinely applied in medical practice is for it to be deployed using clinical decision support systems [Selleck et al., 2017]. A *clinical decision support system* (CDS) is a health information technology that assists physicians and other medical practitioners with clinical decision support i.e. it provides relevant knowledge and patient-specific information, intelligently filtered or presented at appropriate times, to enhance patient health and healthcare [Osheroff et al., 2007]. A popular approach to modeling CDS is using rule-based models that use simple propositional logic statements of the form *IF-<condition>-THEN-<consequent>* (e.g., MYCIN [Shortliffe, 1977], Leeds Abdominal Pain System [De Dombal et al., 1972], HELP system [Kuperman et al., 1991], etc., each described further in Section 2.3). Rule-based CDS are popular because they are human-readable and are often actionable instructions themselves.

In this dissertation, I formulate the problem of discovering statistically significant and clinically relevant biomarkers as a knowledge discovery problem. In computer science, knowledge discovery in databases (KDD) is the process of discovering knowledge from data. Formally, [Fayyad et al., 1996b] define the KDD process as "*the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in the data*". In this definition, knowledge *patterns* refer to associations between biomarkers and clinical outcome of interest. They must be *understandable* to a clinician for it to be deployed in a CDS and for the medical practitioner to be able to act on this knowledge. *Valid* patterns generalize to unseen patient population. We can achieve this with models that contain patterns that are statistically significant. *Novel* biomarkers are useful in research, whereas *useful* patterns can mean non-invasive, cost-effective, high efficacy, and safe biomarkers. So, to solve the problem being tackled in this dissertation, we need to solve the problem of knowledge discovery.

Real-world application:

After developing the data mining method that solves the knowledge discovery problem, I will demonstrate its use on a real-world problem of discovering biomarkers associated with cardiovascular diseases. Cardiovascular disease (CVD) is a group of disorders of the heart, vascular diseases of the brain, and diseases of blood vessels [Mendis et al., 2011]. CVDs includes a range of serious medical conditions including heart attacks, some types of stroke, and heart failure. In 2015, collectively CVDs were responsible for over 15 million deaths in the world [Organization et al., 2017]. In the last 15 years, CVDs have been the leading global cause of deaths. In the clinical management of CVD, risk assessment scores are used to identify high-risk individuals to help target preventive therapies to those with high-risk of developing CVD. Heart SCORE (**S**trategies **C**oncentrating **o**n **R**isk **E**valuation) project, is a longitudinal prospective study undertaken to— help identify racial differences in the development of cardiovascular diseases, to evaluate mechanisms for these differences, and to improve risk stratification. The project studies a cohort based on the greater Pittsburgh metropolitan area. The Heart SCORE project collects a myriad of data from the cohort including demographics, medical history, laboratory assessment, lifestyle variables, etc. The project also collects the metabolomic profile of the cohort. High-throughput technologies for metabolomics provide a holistic view to study small-molecule metabolites created as the

end products of specific cellular processes [Shortliffe and Cimino, 2013]. [Shah and Newgard, 2015] calls for an integrated metabolomics and genomics approach in order to decipher the mechanism of CVDs. In this dissertation, I develop the data mining method for knowledge discovery method and apply it in the real-world problem of discovering the clinical and metabolic biomarkers associated with CVD.

1.1 PROBLEM DESCRIPTION

To implement efficient statistical methods for biomarker discovery, with the ultimate goal of being used in clinical practice, the statistical method must overcome some of the challenges posed by these, otherwise promising, omic datasets. Particularly, high degree of uncertainty in modeling these datasets are attributable to their high-dimensionality, noise, and class imbalance. The statistical method should be able to model multifactorial diseases, the most prevalent and complex type of diseases, which these omic datasets can help better understand. With the ultimate goal of being clinically useful, the statistical method should be able to learn models that are not only statistically valid but also clinically useful. Finally, for knowledge derived from statistically valid and clinically useful models to be actionable in medical practice, it is important for the clinician to be able to interpret the reasoning behind the predictions made by the statistical model.

We further discuss each of these challenges as follows—

1. **High-dimensionality:** Omic datasets are *high-dimensional*. High-dimensional datasets can often have several thousands of candidate variable measurements (e.g., genes, SNPs, or metabolites) that can potentially explain an outcome variable of interest (e.g., phenotype or disease states) but they have only a few samples (e.g., number of patients in study cohort) as evidence to support any patterns inferred from such datasets. These large numbers of candidate variables generate a model search space that is too large for most data mining algorithms to explore efficiently, and having only a few samples as evidence generates uncertainty for the algorithm to determine the statistical correctness of any proposed hypothesis [Ein-Dor et al., 2006]. In such a model search space, data

mining algorithms can easily get stuck in local optima.

High-dimensional data also have an increased risk of supporting false positives. With such a large number of variables and little evidence, the algorithms can easily infer associations— between spurious variables and the outcome variable— by chance [Hastie et al., 2005].

2. **Data noise:** Many biomedical datasets are generated from noisy and unreliable data measurement methods [Shortliffe and Cimino, 2013]. Faulty and miscalibrated equipment can lead to erroneous measurements. Poor record-keeping and incorrect reporting is also an important source of noise. As an example from omics, gene-expression data measurements are known to be noisy. As a result gene-signatures discovered from such datasets are often unreliable and non-reproducible [Koscielny, 2010]. Non-reproducibility of biomarkers is one of the most important reasons why biomarkers, discovered from data analytical methods, do not eventually get used in clinical practice.

   Many powerful statistical techniques, including boosting classifiers, are known to struggle with modeling noisy data as they rely on the correctness of each data sample [Freund et al., 1996].

3. **Class imbalance:** A very common source of uncertainty from biomedical datasets is from class imbalance. This means that out of the samples collected in a study, there are much fewer positive samples than negative samples. Since most diseases have fewer cases than normals in the general population, often even careful selection of at-risk patients may lead up to only a few individuals in the cohort that eventually develop the disease. Many powerful statistical methods in machine learning— including C4.5 and C5.0 decision trees, logistic regression, neural networks, and support vector machines— perform poorly on class imbalanced data [Japkowicz and Stephen, 2002]. The main problem is that their heuristic score (e.g., loss functions like least-squared error) used to evaluate the quality of the model rewards accuracy (i.e., getting as many correct predictions as possible) instead of class discrimination (e.g., quantitatively separate the positive samples from the negative samples). As a result, in their over-eagerness to get as many correct predictions as possible, these methods end up predicting all samples as negative (i.e., the majority class). Such statistical models have very limited clinical value.

4. **Multifactorial diseases:** Most diseases are multifactorial in nature. Single factor diseases are those that have only one causal factor that leads to the disease state. Examples of single-factor disease are monogenic disorders (or Mendelian disorders), where a single gene is responsible for the pathological condition. Examples of single-factor diseases include cystic fibrosis (caused from variants of the *CTFR* gene) [Riordan et al., 1989] and Huntington's disease (caused from variations in *HTT* gene) [Vonsattel and DiFiglia, 1998]. These diseases and their causes are important to study, not only from the disease and treatment standpoint but also from the perspective of functional genomics. Single factor diseases are, however, much rarer compared to the far more prevalent, multifactorial diseases [Antonarakis and Beckmann, 2006]. Multifactorial diseases are diseases, where many common variants, each with a small effect, collectively increase the disease risk. Commonly occurring diseases like type II diabetes [Fuchsberger et al., 2016] and coronary heart disease [Poulter, 1999] are known to be multifactorial. An example of a clinically used multifactorial biomarker model is— a 21-gene score that helps predict the likelihood of distant recurrence in node-negative and estrogen-receptorpositive breast cancer patients treated with tamoxifen [Paik et al., 2004].

   Popular data mining methods like decision trees struggle with modeling such diseases [Seni and Elder, 2010]. These tree-based methods use nodes and edges in the tree to specify a variable-value condition. The samples in the dataset are split by being assigned to a tree leaf, if they match the variable-value conditions specified by the path from the root to that leaf. As a result of this algorithmic nature, they suffer from *data fragmentation* i.e., with each specialization of the tree, the depth of the tree increases and the dataset is further split into fewer subsets in form of tree leaves. The algorithm is now left with fewer examples to help validate any new specialization to its hypothesis. Hypotheses inferred from a small number of samples don't generalize well to unseen data and therefore have poor *validity*.

5. **Clinical utility:** Statistical significance is not clinical relevance [Shortliffe and Cimino, 2013]. Statistical significance is the certainty that the test will give the correct answer. Clinical relevance is a measure of how valuable this information is in guiding clinical care. Clinical relevance includes specificity, efficacy, safety, non-invasiveness, and cost-

effectiveness associated with the answer. If a test significantly lowers the quality of life, then the test, while statistically significant, is not clinically relevant. Ultimately, newly discovered biomarkers must prove to be better in some capacity from the current clinical standard. One way to do this is to show general improvement of predictive performance. In case of personalized medicine approach, this can come from an improvement in prediction for a specific molecular subtype. Clinical utility can also come from similarly good predictive models but with biomarkers at a cheaper cost. For example, both finger stick glucose test and venipuncture can be used to measure blood glucose as they measure this fairly accurately for a certain range. If the goal was to just measure blood glucose for that range and nothing else, finger stick testing is clinically more relevant because they are cheaper, faster, and less-invasive than venipuncture, all the while offering the same accuracy.

To the best of my knowledge, there is no data mining method that allows us to define clinical utility and seek clinically more relevant statistical models. Often in the biomarker development process, establishing clinical utility is left to a later stage of the process using decision theoretic and econometric methods. Such an approach does not include the dataset used for biomarker discovery and as a result loses valuable information from the data about alternative models with varying trade-offs between statistical significance and clinical relevance.

6. **Model interpretability:** Interpretable models offer human-readable explanations for their predictions. Interpretable machine learning models provide knowledge that can be more actionable in practice. Interpretable models provide both the context and reasoning for their predictions to the medical practitioner to assist in their decision making. A *context* is a description of a sub-population that behave differently in terms of their clinical outcome when compared to the general population. Models that learn symbolic representations of data—for example, Bayesian networks, decision trees, and rule-based classifiers— readily provide these human-readable explanations for their predictions and also describe the contexts for the different clinical outcomes.

Rule-based models and decision trees are among the most frequently used statistical models in biomedicine due to their simplicity and interpretability [Esfandiari et al., 2014].

Rule learning is one of the oldest, intensively developed, and widely applicable fields in machine learning. Rule learning methods are particularly useful for knowledge discovery tasks, like biomarker discovery, where model parsimony (succinct representations of patterns in the data), interpretability (human readable explanations for predictions), and actionability (use the model inferred knowledge in some way, like biomarker validation). [Fürnkranz et al., 2012] present a comprehensive overview of the wide range of modern rule learning methods, including relational and propositional rule learning, under a unified framework. Rule models offer both descriptive and predictive explanations of the data. Descriptive learning in statistics is a metric meant to group a sub-population based on their unique characteristics. For example— individuals over the age of 60, with low cholesterol, or containing a particular molecular subtype. Being able to identify sub-populations with a significant statistical support can help assist in personalized medicine if this subpopulation behaves differently than the general population. Predictive rules are actionable instructions themselves with an IF-THEN statement. If the condition part of the rule is true, then the consequent must also be true. Then there is an associated confidence value for such a claim.

Two rule learning classifiers that have achieved considerable success in modeling high-dimensional omic datasets are— Rule Learning (RL) [Clearwater and Provost, 1990, Ganchev et al., 2011, Ogoe et al., 2015] and Bayesian Rule Learning (BRL) [Gopalakrishnan et al., 2010]. In [Gopalakrishnan et al., 2006] modified RL algorithm was used for the analysis of high-dimensional proteomic mass spectrometry data to identify protein biomarkers for early detection of Amyotropic Lateral Sclerosis (ALS) a chronic neurodegenerative disease. The discovered markers were validated with immunoblot and immunohistochemistry using commercially available antibodies [Ranganathan et al., 2005]. RL was used to develop a panel of 10 serum biomarkers to identify indviduals at risk of developing lung cancer using data generated from Luminex xMAP (Luminex Corporation) multiplexed immunoassays [Bigbee et al., 2012]. BRL was shown to achieve significantly better predictive performance when compared to other popular decision tree and rule-based classifiers, on average, over 24 high-dimensional genomic and proteomic datasets [Gopalakrishnan et al., 2010]. A 4-protein serum biomarker panel was designed using BRL for the detection of esophageal adenocarcinoma from mass spectrometrybased spectral count data [Zaidi et al., 2014]. There have

been many other successful applications of RL and BRL for the task of biomarker discovery using high-dimensional datasets [Gopalakrishnan et al., 2004, Gopalakrishnan et al., 2006, Ranganathan et al., 2005, Ryberg et al., 2010, Zeng et al., 2011].

As a result of its past success, BRL classification models are suitable candidates to help overcome the challenges listed in this section and develop it further into a valuable tool for biomarker discovery. In the next section, I outline the plan for its development.

1.2 THE APPROACH

In this dissertation, I design a decision support system, powered by a suite of Bayesian Rule Learning-based methods that tries to overcome the challenges posed by omic data analysis, as discussed in the previous section, and develop it into a valuable tool for biomarker discovery. Specifically, this dissertation proposes to overcome the challanges with the following four developments to BRL, namely— 1) Bayesian Rule Learning methods to represent context-specific independencies (BRL.G, BRL.DT, and BRL.DG), 2) Ensemble methods in Bayesian Rule Learning (EBRL), 3) Bayesian Rule Learning with informative priors (BRL_p), and 4) Bayesian Rule Learning for Knowledge Discovery (BRL-KD). The four methods collectively solve the problem of knowledge discovery and are part of the Bayesian Rule Learning System suite of algorithms. These algorithms are part of a larger framework of a decision support system that provides actionable decision support to a medical practitioner. This decision support system is called Bayesian Rules for Actionable Informed Decisions or BRAID. The BRAID framework is illustrated in Figure 2.

The cloud component of the BRAID framework can be run on a cloud-based server and involves the computational work of developing the predictive model from BRL. An expert or a team of experts iteratively run BRL, with feedback, to learn rule models that are clinically relevant at the point-of-care. The computational work includes running the machine learning algorithms in the BRL system, validating the knowledge discovered by BRL, and deploying the validated knowledge as a decision support system. The first component of this is a data repository. BRAID uses BRL to discover knowledge relevant to the clinical problem.

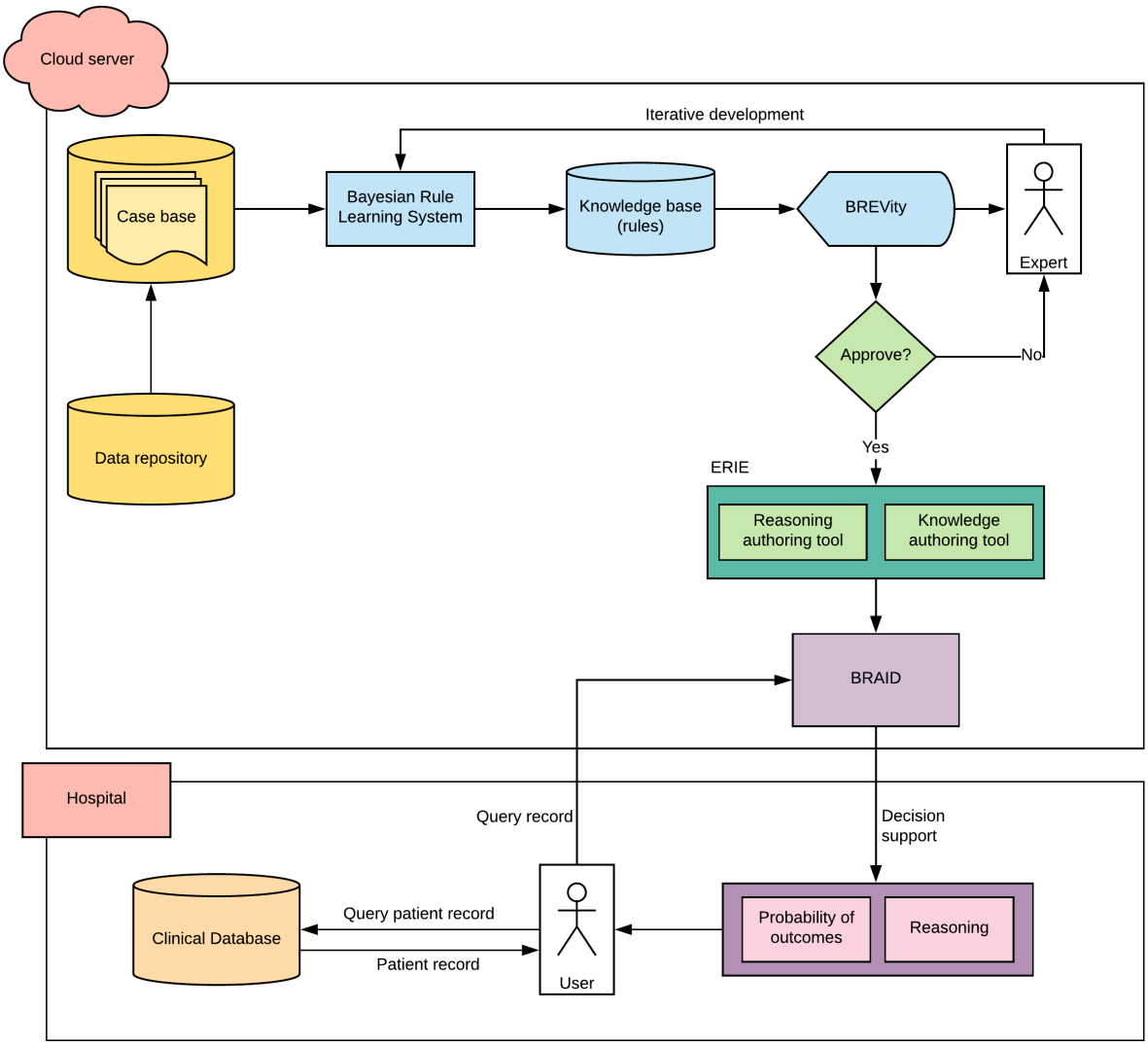


Figure 2: Bayesian Rules for Actionable Informed Decisions (BRAID): An overview of the conceptual cloud-based clinical decision support system that uses the Bayesian Rule Learning (BRL) system as the knowledge engine. The user is a medical practitioner, who queries a patient record to BRAID, which will in turn return the probability of the different clinical outcomes for the patient, and also offer human-readable explanation of factors driving the predicted outcome. The expert helps develop the knowledge base, iteratively generated from BRL system, to develop a rule model that is clinically relevant at the point-of-care.

This data source can be historical electronic health records or a record of a clinical study. We must specify the candidate biomarkers and an outcome variable of interest. This is fed into the BRL suite of algorithms. Using the various methods developed in this dissertation, BRL learns clinically relevant knowledge from the data with the help of the expert. The expert iteratively develops the BRL models leveraging its various functionalities to help improve the clinical relevance of the model. Once a rule is validated, this knowledge is entered into an editor that maps the BRL rule base into clinical standardized terminologies. This module is called Ensemble of Rules Integrated Expert (ERIE). This module allows the domain expert to accept, edit, or validate the rules. This editor can also be used by an expert to add new knowledge or integrate knowledge from other sources, including existing clinical practice guidelines. The expert can also delete or modify rules from this module. Finally, the approved model is deployed by BRAID as a clinical decision support system for a physician.

In the hospital component of the BRAID framework, the medical practitioner is the user. The user can query a patient record to BRAID. BRAID identifies the variables from the patient record that are relevant to its model. It returns the probability of the outcome variable of interest to the user. Additionally, it offers human-readable reasoning behind the prediction.

To realize this BRAID framework, I formulate BRL's role as a knowledge discovery algorithm. In computer science, knowledge discovery in databases (KDD) is the process of discovering knowledge from data. Formally, [Fayyad et al., 1996b] define the KDD process as "*the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in the data*". In this dissertation, the patterns learned by BRL are explicit propositional rules, which are *understandable* because rules are human-readable. To learn *valid* rules, they must generalize well to unseen data by overcoming the challenges described in 1.1. *Novel* and *useful* rules account for clinical relevance. Having framed the problem in such a way, I describe how the four algorithms within the BRL system, together help discover knowledge—

1. **BRL.G, BRL.DT, BRL.DG:** Three representations of BRL were studied to improve the *understandability* of the BRL rule model. Specifically, BRL was extended to represent

all possible context-specific independencies. The BRL model represents the data using a *full decision tree* i.e., the paths from the root node to each leaf node are all of the same length. Collectively, these paths represent all possible variable-value (biomarker-value) combinations using the variables (biomarkers) selected by the model. BRL translates each unique path from the root to leaf as an *IF-THEN* rule. Using the rules to exactly represent a full-decision tree limits the use of the expressive power of rules. Each of these rules describe a context, using variable-value assignments, to uniquely identify a sub-population in the data. Context-specific independence means that this subpopulation does not depend upon a variable that the general population depends upon. Full decision trees cannot represent context-specific independencies. Rules are more flexible representations of these context-specific independences. Explicitly encoding these independences as rules is beneficial to biomarker discovery. Such rules are more compact because there are fewer rules needed to describe the entire study population. This makes the rule model more readable. These representations also reduces the number of biomarkers that need to be validated for the subpopulation (described by the rule) by removing biomarkers that this subpopulation does not depend upon.

In this dissertation, I study three representations— 1) Bayesian Rule Learning with global constraints (BRL.G), 2) Bayesian Rule Learning with local constraints represented as a decision tree (BRL.DT), and 3) Bayesian Rule Learning with local constraints represented as a decision graph (BRL.DG). Each of these methods represent different degrees of context-specific independences.

2. **EBRL:** BRL was extended to better model multifactorial diseases to improve its *validity* on such problems. Multifactorial diseases are challenging to model for tree-based (or graph-based) algorithms because of data fragmentation as described earlier (see point 4 in Section 1.1). BRL is also a tree-based algorithm and as a result isn't efficient in modeling multifactorial diseases. One approach to overcome this challenge is to use ensemble methods to combine predictions from several models, each focusing on different aspects of the prediction problem.

   An unwanted consequence of using ensemble methods is the loss of interpretability that was the main motivation of using rule-based methods in BRAID. To help alleviate this

problem, we developed an ensemble model visualization method called Bayesian Rule Ensemble Visualization tool (or BREVity) that helps interpret the reasoning behind the predictions made by EBRL.

3. **BRL_p:** BRL was extended to BRL_p that can include prior domain knowledge into the model learning process to help further improve its *validity*. [Fayyad et al., 1996a] emphasized the importance of prior domain knowledge in all steps of the KDD process. In biomedicine, often in addition to the dataset, we have some prior domain knowledge about the dataset. This domain knowledge can help guide the data mining algorithm to focus on regions in the model search space that are either objectively more promising for a given problem. The prior knowledge can come from domain literature (e.g., searching through PubMed), a domain expert (e.g., a physician), domain knowledge-bases (e.g., bioinformatics databases or ontologies like Gene Ontology) or from other related datasets (e.g., from public data repositories like Gene Expression Omnibus). It is imperative to develop data mining methods that can leverage domain knowledge to assist with the data mining process.

   I develop BRL_p by leveraging the Bayesian learning framework within BRL to incorporate prior domain knowledge into the model learning process by BRL. The benefit of this capability can be seen for problems involving high-dimensional datasets. When there are considerably large number of variables and only a few examples in the data, associations between spurious variables in the data can occur by chance (false positive discovery). By incorporating reliable prior knowledge from other sources, we can help the search algorithm focus on model subspace with known and reliable patterns. This can help reduce false positive discoveries.

4. **BRL-KD:** All the methods described so far attempt to improve the *validity* of the rule patterns. This method is developed to improve the clinical relevance of the discovered knowledge by also accounting for *novelty* and *usefulness*. BRL-KD can incorporate a clinical utility function into the model learning process. An example of that is to learn cost-efficient biomarkers. By incorporating the costs associated with each biomarker, BRL-KD is able to offer a set of BRL models, each offering a different trade-off between cost and validity. The choice of the ideal trade-off depends upon the clinical point-of-care

and the current clinical standard in use.

With the help of all these four methods, BRL can perform knowledge discovery that finds *valid*, *novel*, *useful*, and *understandable* patterns from the data.

1.2.1 Thesis

The main hypothesis of this dissertation is that—

> *the designed BRAID system powered by the BRL, EBRL, BRL$_p$, and BRL-KD methods developed herein, learn classifiers for biomarker discovery that are, on average, statistically more significant compared to traditional supervised learning methods, while also being able to find clinically more relevant classifiers.*

To test this hypothesis, I pursue two main aims— 1) To design and develop the four algorithms BRL, EBRL, BRL$_\text{p}$, and BRL-KD, that collectively help learn clinically relevant knowledge, and 2) To evaluate these developed methods, using 25 publicly available gene-expression diagnostic datasets over 10-fold cross-validation design, in terms of its prediction and calibration performance, and compared to state-of-the-art traditional supervised methods in machine learning. For BRL-KD, I will also evaluate for clinical utility using an objectively defined utility function and evaluate the clinical utility on a held-out test set.

1.3 SIGNIFICANCE

The contributions of this dissertation include the following—

1. To my knowledge, this is the first work to learn and represent context-specific independencies existing in the data, using rule-based classifiers. This is important because rule-based models are widely applicable in biomedicine (including in the design of clinical decision support systems). Explicitly representing context-specific independencies in rules make them succinct because each rule describes a subpopulation with a unique outcome distribution. Such rules are also more robust because they exclude any variables that the population as a whole may depend upon but the subpopulation does not depend upon. This is especially important in biomarker development process during the validation of the discovered biomarkers.

2. This would be the first work to combine BRL models in an ensemble learning framework to study the properties of the different strategies of model generation and model aggregation, in the context of omic data analysis.
3. Bayesian Rule Ensemble Visualization (BREVity) tool in the BRAID system would be a novel model visualization tool to help visualize variable relationships in an ensemble of rule-based classifiers. BREVity could prove to be an important tool in the biomarker development process.
4. BRL_p would be a novel rule-based method capable of allowing the user to incorporate prior domain knowledge into the rule learning process. There is a wealth of knowledge in various bioinformatics resources that can potentially assist in learning more accurate classifiers. BRL_p would enable the user to use this knowledge in model learning.
5. BRL-KD would be a novel algorithm that can incorporate clinical utility into the model learning process, in order to learn clinically more relevant classifiers. This can be important in both machine learning and medicine. In machine learning, to my knowledge, there exists no model learning method, without manual intervention by experts, that can learn models that are clinically more useful. From the medical perspective, this tool can be important to improve the chances of clinical adoption of the learned models.
6. The BRAID system is a novel clinical decision support system design that uses machine learning algorithms to learn statistically significant and clinically relevant classifiers that offers decision support to a physician, while providing readable explanations for its predictions in form of rules or the BREVity model visualization tool.

1.4 DISSERTATION OVERVIEW

The dissertation is organized as follows— in chapter 2, I provide an overview of the biomarker development process, the wealth of knowledge available from the disciplines of bioinformatics and translational informatics, description of clinical decision support systems, and a conceptual background on knowledge discovery in databases. I also provide the reader with a review on the Bayesian approach to probability and statistics, which then sets up understanding

a popular and robust statistical model called Bayesian networks. We will then review rule learning methods, one of the oldest and still widely used statistical models. I then introduce the reader to Bayesian Rule Learning, a rule learning algorithm that leverages the benefits of the Bayesian inference framework. Chapter 3 contains the motivation and design of each of the Bayesian Rule Learning methods developed in this dissertation study. Chapter 4 describes the experimental design, evaluation metrics, and results from the evaluation of the BRL algorithms in terms of their predictive and calibration performance on real-world gene-expression datasets. Here, BRL performance is compared with other popular supervised learning methods. Chapter 5 applies the developed Bayesian Rule Learning methods in the BRAID system to a real-world biomarker discovery problem for predicting cardiovascular disease outcomes. Finally, in Chapter 6 we summarize the dissertation and consider directions for future work.

2.0 SIGNIFICANCE AND BACKGROUND

In this chapter, I provide some necessary background and useful references, to the reader, to help understand the ideas explored and developed in this dissertation. In section 2.1, I introduce biomarkers and walk through the process of biomarker development in biomedical research. In this process, we see the statistical challenge of analyzing biomedical datasets and also the challenge of identifying clinically relevant biomarkers. Section 2.2 describes the fields of bioinformatics and translational bioinformatics. We take a look at the wealth of resources available from tools in these fields that can help bridge the gap between knowledge discovered from basic biomedical research into its adoption in clinical practice. Section 2.3 describes clinical decision support systems, a commonly used application of biomarkers in clinical practice. Section 2.4 defines some important concepts, from computer science, pertaining to a field knowledge discovery from databases. As seen in the introduction chapter, we formulate the problem of finding statistically significant and clinically relevant biomarkers as a knowledge discovery problem. Section 2.5 introduces the Bayesian approach to probability and statistics. Here, we see that Bayesian methods are ideally suited to help solve the problem of finding subjectively interesting knowledge from data. We also take a look at Bayesian networks, a powerful statistical model to represent and learn knowledge from data. Section 2.6 provides the background for rule learning, one of the oldest, intensively studied, well-developed, and widely deployed models in machine learning. Rules are popularly used in clinical decision support systems. Section 2.7 describes Bayesian Rule Learning, a rule learning method that infers rule models from Bayesian networks, thereby leveraging the benefits of both Bayesian methods and rule learning methods.

2.1 BIOMARKERS

A *biomarker* is a characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacologic responses to a therapeutic intervention [Group et al., 2001]. Biomarkers help better understand the underlying physiological and pathophysiological processes within an individual. Scientific investigators discover novel biomarkers for a wide range of purposes including improved clinical decision making and guiding research directions for biomedicine.

Sources of biomarkers can be macroscopic clinical variables like gender, age, cholesterol, family history of a disease, environmental exposure, etc. They can also come from omics data that contain molecular measurements from the DNA, RNA (and its derivatives), proteins, metabolites, and epigenomics. Biomarkers can be measured from blood, serum, lymph, tissues, or body fluids like urine and saliva. Biomarkers help influence biomedical research directions, clinical practice, and public health practice. An example of a biomarker commonly applied in public health practice is weight loss, a lifestyle marker associated with many substantial health benefits like improved glycemic control, reduced blood pressure, and reduced cholesterol levels [Goldstein, 1992]. In medicine, biomarkers are used for risk stratification, screening, diagnosis, prognosis, to predict response to therapy, and to predict clinical outcomes.

Based on the common clinical applications of biomarkers, they can be broadly classified into three types— *diagnostic*, *prognostic*, and *effect modifiers* [Micheel et al., 2012]. *Diagnostic markers* for a specific disease help differentiate an individual with the disease from a healthy individual (e.g., certain harmful mutations in *BRCA* genes can lead to substantial increase in risk of developing breast or ovarian cancer [Levy-Lahad and Friedman, 2007]; prostate-specific antigen is increased in serum of men with prostatic disease, including prostate cancer [Catalona et al., 1991]). *Prognostic markers* help predict survival or disease progression (e.g., a 21-gene score that helps predict the likelihood of distant recurrence in node-negative and estrogen-receptorpositive breast cancer patients treated with tamoxifen [Paik et al., 2004]). *Effect modifiers* predict response to some therapeutic intervention (e.g., for individuals with abnormal expression of *ALK* gene, a molecular driver for

non-small-cell lung cancer, crizotinib is a more effective treatment than standard chemotherapy [Shaw et al., 2013]; an *EGFR* mutation can help predict if tumor cells will respond to treatment using tyrosine kinase inhibitors [da Cunha Santos et al., 2011]).

Based on their biological function in relation to the outcome of interest, biomarkers can be classified as either predictive or mechanistic. Predictive biomarkers are correlative to the outcome event of interest. They may or may not be causal of the outcome. They merely help predict a clinical outcome of interest. Mechanistic biomarkers, on the other hand, are causes of pathology, disease progression, or sensitivity to a given drug. They suggest interventions, for example, therapeutic targets to potentially alter the clinical outcome.

2.1.1 Biomarker development

The biomarker development process consists of four stages [Goossens et al., 2015]— 1) biomarker discovery, 2) analytical validation, 3) clinical utility validation, and 4) clinical implementation. The process has been summarized in Figure 3.

Biomarker discovery: In the biomarker discovery stage, new biomarkers are proposed often using data analysis methods performed on a training dataset containing a cohort of relevant patients. The proposed biomarkers are then validated either on an independent test cohort or using a cross-validation study design. The research question and the purpose of the biomarker needs to be articulated before the relevant data is even collected. The clinical endpoints, which are the clinical outcomes of interest, should be clearly defined. We also need to determine if we seek predictive or mechanistic biomarkers. These decisions will help design the statistical data analysis and knowledge discovery tasks. Leaving this decision to the end of the data analysis step can lead to many false positive predictions that will not translate well into clinical practice. Decision on the patient cohort description and distribution, and the number of samples necessary (using power analysis) for a required effect size for the biomarker to be clinically useful, has to be made before the data is collected. The source for the biospecimen and the technology used to quantify the candidate biomarkers also have to be decided early on with respect to the clinical application.

Analytical validation: In the analytical validation stage, the proposed biomarkers from

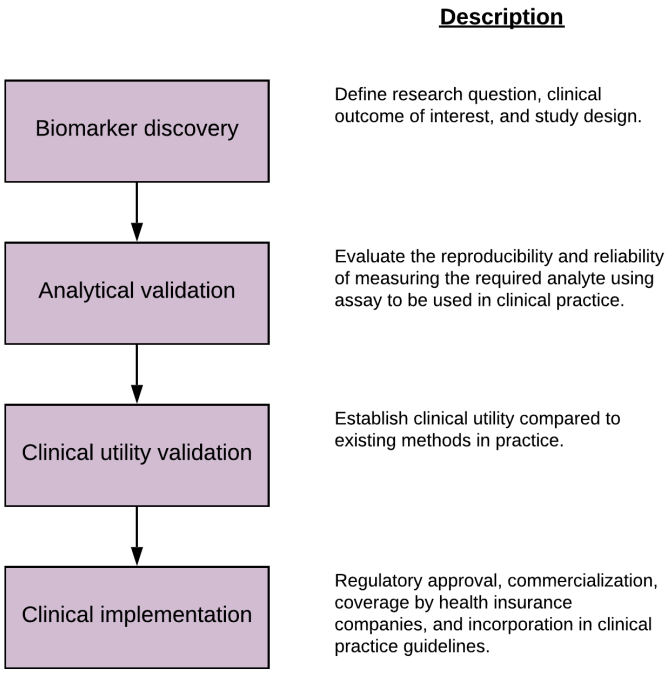


Figure 3: The four stages of the biomarker development process.

the biomarker discovery stage are evaluated for how reliably the analytes can be measured in the patient biospecimen. Often the original samples from the biomarker discovery stage are assayed again to test for the robustness and reproducibility of the measurements made in the discovery stage. The assay technology used for this validation is the same one that will be clinically deployed. For example, poor reproducibility of measurements is an important reason why gene-expression signatures are not used in clinical practice [Koscielny, 2010]. [Goossens et al., 2015] discuss the promise and limitations of various emerging technologies to help improve reproducibility of analyte measurements. They include direct digital counting of transcripts without target amplification, clinical gene sequencing, and sensitive assays like single cell profiling.

Clinical utility validation: Once we establish that the analytes can be reliably reproduced in the clinical setting, we establish the clinical utility of the biomarker. One aspect is

to validate if the proposed biomarkers indeed reliably predict the clinical outcome of interest, as predicted in the discovery stage. Ideally, this is done on well-designed prospective study with a large enough sample cohort. This stage is constrained by time, financial resources, and the availability of patient cohorts fitting the requirements. Additionally, clinical utility also includes establishing clinically meaningful benefit of the new test. [Parkinson et al., 2014] outline key steps to evaluate clinical utility of any new proposed test. An important first step is to determine what benefit the new test brings in comparison to existing methods either in terms of sensitivity/specificity, non-invasiveness, efficacy, and cost-effectiveness. Some of the other steps outlined by [Parkinson et al., 2014] are covered in the other stages of biomarker development described in this section.

Clinical implementation: Before implementing the new test clinically, we must consider regulatory approval, commercialization, coverage by health insurance companies, and incorporation in clinical practice guidelines. Regulatory approval strategies can be either to deploy the test by obtaining a Food and Drug Administration (FDA) approved in vitro diagnostic device (IVD) test or as a laboratory developed test (LDT) with clearance from Clinical Laboratory Improvement Amendments (CLIA) with no need for approval from the FDA [Parkinson et al., 2014]. Cost-effectiveness and improvement in performance when compared to standard procedures can help determine commercial success. Coverage by health insurance companies is necessary for the physicians to be able to order these tests. By establishing the benefits of the new tests and observe success in clinical practice, we can change clinical practice guidelines with the discovered biomarkers.

2.1.2 The promise and challenges from omic data

Biomarkers derived from high-throughput measurements of molecular information from the DNA (genomics), RNA and its derivatives (transcriptomics), proteins (proteomics), metabolites (metabolomics), and epigenetics (epigenomics), are collectively called omics datasets. Omics data usually measures a comprehensive molecular profile of an individual. This dataset offers us an opportunity to understand biological mechanisms at an unprecedented resolution. With the ever-reducing costs and improvements in the accuracies of these high-throughput

technologies, it is now feasible to include omics data as a standard component to a person's health record. This will allow us to discover novel biomarkers that improve clinical diagnostic, prognostic, and therapeutic decision making thereby eventually improving population health.

Statistical modeling is necessary for biomarker discovery from omic datasets. But omic datasets are often marred with many sources of uncertainties. Uncertainty can stem from data scarcity (number of samples or patients in the training data). This can happen if, for example, the occurrence of the disease being studied is rare or it is too expensive to add individuals to the study. This is very common in many biomedical studies. Uncertainty can also come from noisy or unreliable data measurements. Faulty/miscalibrated equipments can give erroneous measurements. As we saw earlier, gene-expression measurements can also be unreliable [Koscielny, 2010].

Omic datasets are high-dimensional datasets. They are challenging to data mining algorithms because typically several thousands of candidate variables (e.g. gene expressions or SNPs) can potentially explain an outcome variable of interest (e.g. phenotypes or disease states) but have only a few instances as evidence to support an explanation. These large numbers of candidate variables generate a model search space that is very large for data mining algorithms to explore efficiently, and having only a few instances generates uncertainty for the algorithm to determine the correctness of any candidate model. In such model search spaces, statistical modeling methods can easily get stuck in local optima or they may infer associations between spurious variables and the outcome variable, by chance (high false positives).

Omic datasets are often used to differentiate disease phenotypes. A lot of diseases are multifactorial in nature. Single factor diseases, for example, monogenic disorders (or Mendelian disorders), while important to study, are relatively rare compared to the more common, multifactorial diseases [Antonarakis and Beckmann, 2006]. Examples of single factor diseases include cystic fibrosis (caused from variants of the *CTFR* gene) [Riordan et al., 1989] and Huntington's disease (caused from variations in *HTT* gene) [Vonsattel and DiFiglia, 1998]. Whereas the more common diseases like type II diabetes [Fuchsberger et al., 2016] and coronary heart disease [Poulter, 1999] are multifactorial, where many common

genetic variants, each with a small effect, collectively increase the disease risk.

All these sources of uncertainties make omic data analysis a challenging task for statistical modeling. *We need statistical models for biomarker discovery that can efficiently learn models from high-dimensional datasets for multifactorial diseases.*

2.1.3 Clinically relevant biomarkers

From the biomarker development process, we can see that in addition to a good study design, we may still not pass the clinical utility validation step if our proposed biomarkers are not clinically meaningful. The new proposed test should improve upon the existing clinical standard either in terms of sensitivity, specificity, efficacy, non-invasiveness, and cost-effectiveness.

[Selleck et al., 2017] emphasizes that it is not enough to identify variants but to identify *actionable* variants that has the potential to revolutionize healthcare. [Burke, 2016] reported in 2016, that there were 768,000 papers indexed in PubMed about biomarkers. Yet, despite all the technological advances in omics research and bioinformatics methods, we are still very far from widespread clinical use of these omic biomarkers. Currently there are only a few dozen clinically relevant cancer biomarkers. There is also a general lack of support from clinical practice guidelines. The European Society of Medical Oncology (ESMO) clinical practice guidelines for lung, breast, colon, and prostate cancers give only a weak recommendation for the use of about 20 omic biomarkers.

The process of translating new biological research knowledge into clinical practice is slow. However, there are other factors that also determine the potential for success of biomarker discovery projects. It is to establish clinical relevance.

For example, specificity of the biomarkers is an important aspect of clinical relevance. Some biomarkers may serve as a test for many pathological conditions and are not specific to the disease it is being developed as a test for. For example, cancer antigen 19-9 (CA19-9) is a biomarker for pancreatic cancer with high statistical significance [Steinberg, 1990]. However, this antigen is elevated in many other pathological conditions, such as biliary obstruction, that often co-exists with pancreatic cancer.

[Selleck et al., 2017] suggest a 4 step evaluation process for newly discovered biomarkers to improve their chances of being clinically useful. Their evaluation is directly inspired by the biomarker development process described earlier. They are— 1) Analytical validity, the test should be reproducible; 2) Clinical validity, the biomarker should be able to distinguish one group from another in a meaningful way (for example, we must either show improvement in predictive performance in general or for a defined molecular subpopulation, for personalized medicine); 3) clinical utility, will the result of the new test change clinical outcomes?, and 4) cost-effectiveness, psychological and ethical implications.

2.2 BIOINFORMATICS AND TRANSLATIONAL INFORMATICS

In this section, we take a look at the disciplines of bioinformatics and translational bioinformatics. We account for the wealth of information available from the various public resources available in bioinformatics. We also take a look at the challenges faced by and resources available in the field of translational bioinformatics.

2.2.1 Bioinformatics

Bioinformatics is the study of the storage and analysis of information in basic biological systems. The field mainly deals with molecular-level information [Lesk, 2019]. From omics data generated from high-throughput technologies, various bioinformatics methods can help interpret and integrate various knowledge resources in biological research, to provide researchers with consolidated scientific resources to improve our understanding of complex biological systems. Currently, statistical methods do not try to leverage this wealth of information. Finding efficient ways to incorporate this knowledge can be crucial to help learn clinically more relevant biomarkers.

There are five major research areas in bioinformatics, they are— 1) DNA and protein sequence analysis, 2) macromolecular structure-function analysis, 3) gene expression analysis, 4) proteomics, and 5) system biology.

Sequencing data analysis can help identify genetic variants associated with phenotypes of interest like disease outcomes, predict gene function, and provide evolutionary information by comparing sequences from other species. Metagenomics is a new subdomain that studies microorganism ecosystems using DNA sequencing from, for example, the human gut flora. This can open up promising new resources for biomarker discovery in the future. National Center for Biotechnology Information (NCBI) GenBank [Benson et al., 2018] is a publicly available comprehensive database of biologically annotated with literature references for nucleotide sequences of more than $400,000$ formally described species. The European Bioinformatics Institute (EBI), Swiss Institute of Bioinformatics (SIB), and Protein Information Resource (PIR) collectively maintain a comprehensive publicly available resource of annotated protein sequences called UniProt [Consortium et al., 2018].

Predicting macromolecular function from 3D structures of macromolecules is a challenging but worthwhile effort. They can help infer biological functions, identify therapeutic targets, and predict drug efficacy. Protein Data Bank [Burley et al., 2018] is a publicly available data repository of 3D structures of proteins and nucleic acids obtained from X-ray diffraction, nuclear magnetic resonance, and electron microscopy experiments. As of 4^{th} July 2019, Protein Data Bank contains 3D structures of $153,601$ macromolecular structures.

In gene expression analysis, the transcriptomes (mRNA or cDNA copied from a template RNA) are quantified from two groups of cells that are exposed to different physical conditions (e.g., disease vs. healthy cell, cells with drug treatment vs. cells without treatment, etc.). Using DNA microarrays or RNAseq, the transcriptome in different conditions are quantified. Using these measurements, genes that are differentially expressed can be identified. This information can provide insights into the collective gene function. NCBI's Gene Expression Omnibus (GEO) [Edgar et al., 2002] is a publicly available repository of gene expression experiments. The database also provides metadata from the conducted experiments that can be very useful for data analysis.

Proteomics, in addition to sequence analysis also deals with determining their structure and post-translational modifications. UniProt and Protein Data Bank additionally provide such comprehensive information about proteins.

Systems biology is the study the entire biological system including all molecular infor-

mation from genomics to metabolomics. It includes the study of relationships between the different genes, transcripts, proteins, and drugs, in the orchestration of a biological process. This field is still young but promises high-impact scientific discoveries. A popular public database that looks at system level information is Kyoto Encyclopedia of Genes and Genomes (KEGG) [Kanehisa et al., 2011], which is a collection of databases including genomes, pathways, diseases, drugs, and metabolites.

In addition to those major bioinformatics resources, biomedical literature is an invaluable knowledge resource that can also help with data analysis. The U.S. National Library of Medicine's MEDLINE is a bibliographic database for life sciences and medicine. PubMed is a search engine that helps efficiently search this database.

2.2.2 Translational bioinformatics

Translational medicine is the translation of biological knowledge discovered from basic scientific research (bench-side) to clinical practice (bed-side), which can eventually help improve population health. *Translational Bioinformatics* (TBI) is a sub-field in bioinformatics that enables translational medicine by applying the concepts and methods developed in bioinformatics to human healthcare [Butte and Chen, 2006]. TBI can help realize the goal of personalized medicine by developing methods to discover and translate knowledge pertinent to clinical care, which are contained in the voluminous omics datasets. Traditional clinical guidelines were based on macroscopic symptoms and physiological observations made in a course of a physical exam or what is reported by the patient. In a personalized medicine approach, we must consider omic biomarkers that provide us with the unique molecular fingerprint of the individual. Such an approach to personalized medicine can lead to improved clinical care. TBI needs to develop methods for standardized data storage and retrieval, novel methods for biomedical data analysis and interpretation, and provide decision support to clinicians.

As illustrated in Figure 1 in the introduction section, there is a translational barrier in translating discoveries made in basic biomedical research to its adoption in clinical practice. There are two such barriers— 1) T1 barrier, which is the translation of scientific discoveries

into changes in clinical guidelines [Lenfant, 2003], and 2) T2 barrier, the adoption of clinical guidelines into medical practice by health practitioners [Sung et al., 2003].

There are three levels of informatics— data, information, and knowledge. Data is the raw recorded observations in the database. Example of raw data is proteomic assay from a cohort of patients and a label for each patient indicating whether or not they have a phenotype of interest, say, prostate cancer = {*case*, *normal*}. Information is giving meaning to the data using data analysis methods. For example, if the data analysis finds an association between the phenotype and a protein, say, *prostate-specific antigen*. Knowledge is to interpret the information in a specific context. In translational medicine, this is the clinical context. For example, knowing that a combination of ultrasonography and elevated levels of prostate-specific antigen, has a higher precision in detecting prostate cancer than ultrasonography alone. So, for patients over the age of 50, a combination of the two tests can be routinely conducted to detect prostate cancer with improved clinical precision. However, as we saw in section 2.1, precision is only one aspect of clinical utility. Efficacy, cost-effectiveness, and non-invasiveness are other aspects of clinical utility that can better define clinical context. In section 2.4, we will see how we can formulate the statistical data analysis task to seek such knowledge instead of just information.

2.3 CLINICAL DECISION SUPPORT SYSTEMS (CDS)

A *clinical decision support system* (CDS) is a health information technology that assists physicians and other medical practitioners with clinical decision support i.e. it provides relevant knowledge and patient-specific information, intelligently filtered or presented at appropriate times, to enhance patient health and healthcare [Shortliffe and Cimino, 2013, Osheroff et al., 2007]. They are mainly of three types— 1) infobuttons that retrieves relevant documents for a clinical context, 2) provide patient-specific or situation-based alerts or reminders, physician order sets, etc., and 3) present information in a way as to facilitate decision making.

A classic example of a CDS is de Dombal's Leeds Abdominal Pain System [De Dombal

et al., 1972]. They use Bayesian reasoning from high-quality data to calculate the probability of seven possible explanations for acute abdominal pain. For 304 patient cases, CDS's overall diagnostic accuracy of 91.8% was found to be significantly higher than that of the most senior member of the clinical team, who obtained an accuracy of only 79.6%. However, this system did not generalize well in practice. One possible explanation is that different subpopulations may exhibit different probabilistic dependencies in the domain. Another explanation is inconsistency in agreement between physicians, while reporting the results of a physical examination.

MYCIN is a famous rule-based CDS for diagnosing and managing infections [Shortliffe, 1977]. The rules chained together, where the output of a rule that fires for a case is taken as an input for another rule in the system. The sequence of fired rules offered human-readable explanations for MYCIN's decisions. Extending MYCIN was easy, it involved simply adding, altering, or removing rules from the system. MYCIN was evaluated for therapy selection for patients suffering from blood-borne bacterial infections and meningitis. For the evaluation with meningitis patients, MYCIN performed better than experts. While MYCIN was never used clinically, it offers an excellent, flexible framework to develop future CDS.

The HELP system is another famous CDS, which was integrated into the LDS Hospital system at Salt Lake City [Kuperman et al., 1991]. HELP is a patient record monitoring system that generated alerts to medical practitioners when there was an aberration detected in a patient's medical record. These alerts were stored as simple decision logic rules, called the Arden Syntax. HELP communicated to medical practitioners via the hospital information system's workstation or using written reports.

CDS typically have three parts— 1) a knowledge base, 2) an inference engine, and 3) a communication system [Soufi et al., 2018]. The knowledge base has domain knowledge encoded, usually in form of IF-THEN rules. The inference engine links a specific patient to the set of relevant rules from the knowledge base. The communication system allows the user to query the system and conveys the relevant information to the user.

Musen et al. [Shortliffe and Cimino, 2013] observe that with the increased adoption of health information technology in medical practice, investigators are considering large-scale data-mining methods to design CDS for a wide range of applications including population

monitoring, public health surveillance, and offer patient-specific recommendations based on cohort data when there is no information available to guide therapeutic decisions.

2.4 KNOWLEDGE DISCOVERY IN DATABASES

Knowledge Discovery in Databases (KDD) in computer science, is an important process of discovering useful knowledge from data. Formally, [Fayyad et al., 1996b] define the KDD process as follows—

> *Knowledge discovery in databases* is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in the data.

Data, D, is a collection of observations, called *instances*. Each instance contains values for a set of recorded variables. For example, data collected to study the influence of smoking on lung cancer would attempt to collect information from several individuals in a population, i.e, instances, and record two variables of interest from them— 1) their smoking history, and 2) whether or not they have lung cancer. A *pattern*, E, is an expression in some language, say L, that uniquely identifies a subpopulation of instances from data D. An example of a language is propositional logic, which expresses a pattern in form of IF-THEN statements. For example, the pattern "IF the individual does not smoke THEN they do not have lung cancer", uniquely identifies a subpopulation in D, containing individuals who do not smoke and do not have lung cancer.

Valid patterns generalize well to unseen test instances. If our previous example pattern were to be valid, then for any new individual not represented in D, who happens to not smoke, would also not have lung cancer. The confidence that the pattern would map correctly to an unseen test instance is uncertain. A mapping function, c, can be used to assign a uncertainty in a pattern's validity. This can be any mapping function, $c = C(E, D)$, that takes as input, the data D and pattern E and maps it to a metric space M_C.

Novel patterns either deviate from what is already known in the domain by stating a contradiction to what is known or enhance the current knowledge of the domain. In our

example, the pattern that relates smoking to lung cancer can be trivial as it is common knowledge. However, a novel pattern would be something that either finds a previously unknown variable now found to be associated with lung cancer or an unexpected contradiction, like a pattern that links smoking to lung cancer but with an exception. Let us hypothesize, like in validity, it is possible to measure the novelty of a pattern using a mapping function $n = N(E, D)$ into a metric space, M_N.

Useful patterns are actionable in the domain, both in terms of feasibility and domain impact. In the lung cancer example, if we found a pattern that associates a variable with lung cancer that is very expensive to evaluate for new test instances or impossible to modify (e.g. age) then they aren't quite useful to a user. Again, let us assume it is possible to specify a utility function $u = U(E, D)$ that maps the pattern into a metric space, M_U.

Understandable patterns are those that are comprehensible to the user. Examples of understandable patterns are pattern languages that are interpretable by a user for example, rules, Bayesian networks, and decision trees. Examples of patterns relatively not interpretable include support vector machines, naïve Bayes, and neural networks. It is hard to measure pattern understandability but a helpful substitute is measuring the semantic complexity of the pattern. For example, if the pattern language is a rule then a more understandable pattern would have fewer disjuncts or clauses. Say, we can measure this with a mapping function for semantic readability, $s = S(E, D)$.

Further, [Fayyad et al., 1996b] define *interestingness* as a measure that distinguishes patterns that are "valid, novel, potentially useful, and ultimately understandable" from those patterns that are not. Let us assume that we can quantify the interestingness, i, of a pattern using the previously defined metrics for validity (C), novelty (N), usefulness (U), and understandability (S). This mapping function $i = I(E, D, C, N, U, S)$ would map the pattern into a metric space, M_I. Mapping functions that map a pattern into M_I are called subjective interestingness measures because while validity is typically an objective metric with respect to the data, novelty, usefulness, and understandability are subjective measures with respect to a user.

To summarize, knowledge that is subjectively interesting to a user is a pattern that for a user specified threshold $i \in M_I$ achieves an interestingness measure of $I(E, D, C, N, U, S) > i$.

2.4.1 Subjective interestingness measures

[Silberschatz and Tuzhilin, 1996] point out that objective evaluation of interestingness from the data alone do not represent the most important patterns present in the data, with respect to a user. Important patterns should be subjectively interesting to the user. We need subjective interestingness measures to evaluate such patterns. In the following subsection, we will review current literature in quantifying subjective interestingness.

There is a large body of work done on subjective interestingness measures, especially in descriptive rule induction approaches like association rule mining and relational rule mining. [Geng and Hamilton, 2006] and [McGarry, 2005] provide an extensive review of the approaches to subjective interestingness in KDD. We summarize some of those works here. [Silberschatz and Tuzhilin, 1996] points out that objective evaluation of interestingness from the data alone do not represent the most important patterns present in the data for a user. The patterns most important to a user are those that are unexpected and actionable. *Unexpected* patterns add to the user's knowledge, and *actionable* patterns are potentially useful to the user. As an example, consider the rule pattern, *IF(PERSON_PREGNANT = TRUE) THEN (PERSON = FEMALE)*. While this association pattern is likely to get a lot of support in the dataset, this isn't interesting because it is obvious and this knowledge isn't useful. However, consider two rule patterns learned from a dataset studying accidents— 1) *IF(SEAT_BELT = TRUE) THEN (INJURY = FALSE)*, and 2) *IF((SEAT_BELT = TRUE) AND (PASSENGER = CHILD)) THEN (INJURY = TRUE)*. These two associations are interesting and *unexpected* because they contradict each other. This makes us wonder what happens in the instances where the second rule fires. It turns out that children are too short for the chest portion of the seat belt and only wear the lap portion of the seat belt. This is a problem because during an accident, their heads are projected in front towards some object causing head injuries. This is due to the lack of restraint on the chest by the seat belt. This discovery is *actionable* because we know that we can act upon it. So, there is now a law by National Highway Traffic Safety Administration in the USA that requires children under a certain height to have child restraint in the car.

An approach to quantifying subjective interestingness is to represent the entire known

knowledge about the domain. [Silberschatz and Tuzhilin, 1996] describe a probabilistic way to represent prior beliefs, which is then updated using Bayesian inference. [Liu et al., 1999] represent known knowledge using simple propositional rule patterns. They quantify interestingness using semantic distance between the pattern learned from the data to the specified prior knowledge. The main limitation of this approach of prior knowledge coding is that it is infeasible to specify all of the known knowledge in a domain. This is especially true for biomedicine. [Fabris and Freitas, 2000] propose a subjective metric for rule learning that intentionally looks for a Simpsons's paradox. Since these paradoxes are usually unexpected, they quantify unexpectedness from the number of paradoxes. This method is useful if we feel such paradoxes are useful. [Piatetsky-Shapiro and Matheus, 1994] quantify actionability in term of profits made by the organization. The limitation here is the need to specify a mapping between the space of all models to some utility like profits. This may not be feasible for most problems. [Sahar, 2002] incorporate subjectiveness into the association rule mining process of APRIORI algorithm by creating constraints in the search space of association rules. In each iteration, the user provides feedback to the rule mining process. This is very applicable because it does not require the user to specify all known knowledge at once.

2.5 BAYESIAN PROBABILITY AND STATISTICS

Bayesian methods give us a systematic way to include subjective knowledge (subjective interestingness). [Heckerman, 2008] provide a detailed introduction to learning with Bayesian networks including a sound background on Bayesian approach to probability and statistics.

A Bayesian probability refers to a person's degree of belief in an event. It is the *subjective* property of an individual who assigns the probability. Instead, classical probability refers to the physical property of the system generating the event. It is the *objective* property of the world.

The Bayesian approach to probability and statistics is philosophically different from the classical (or frequentist) approach. In the classical approach, it is believed that our observation is variable but the system that generates these observations is deterministic. On

the other hand, the Bayesian approach posits that our observations are deterministic but the system that generates these observations is a variable. As an example, take the coin-toss problem. The outcome of a coin toss is either *heads* or *tails*. We want to predict the outcome of the next coin toss i.e, the probability of heads. In the classical approach, the probability of heads is deterministic albeit unknown. The observation of toss outcomes differ in each trial. In theory, if we could find the relative frequency of heads over infinite series of identical coin tosses, where each outcome differs but the probability of heads remains deterministic, we can calculate the true probability of heads. Alternatively, in the Bayesian approach, the toss outcomes is deterministic but the probability of heads is a variable. The probability of heads is influenced by a number of initial conditions of the system that generates a coin toss, for example— the force applied to the coin, the effects of atmosphere on the toss outcome, obstructions to the coin, etc. If all these initial conditions of the system is mimicked identically then the toss outcome would be deterministic. In theory, if we knew all the variables affecting the outcome, we do not need infinite trials of data to predict the outcome, in fact, we only need one observation for a particular set of initial conditions. There is a joke comparing the classical (or frequentist) approach to the Bayesian approach that does a good job of summarizing the differences— "the frequentists believe there is infinite data, while the Bayesians believe there is infinite initial conditions". Of course, for most practical purposes, the frequentists don't need infinite data and the Bayesians do not have to identify all the finitely large initial conditions.

The Bayesian approach is a more natural view of reality that makes it more applicable in practice. For example, say we want to know— what is the probability that the Pittsburgh Steelers will win the next football game against Baltimore Ravens? In the classical approach, we want a dataset of an infinite series of identical games between the Steelers and the Ravens. Note that the games should be identical, with the same set of players, in the same physical condition, and the same environmental conditions. Such a dataset does not exist and so the application of the correct classical approach is not feasible. However, the Bayesian approach is still applicable because here we just want to assess an individual's belief in Steelers winning the game. There have been many methods developed for accurate and precise Bayesian *probability assessment* from an individual [Heckerman, 2008].

2.5.1 Learning from the data

Let us identify the elements of the classical and the Bayesian approach to solving the coin toss problem by formalizing the problem. We start by defining some notations. A random variable in the system is denoted by an upper-case letter, e.g., X, Y, Θ. Lower-case letters denote the state or the value for the variables denoted by their upper-case letters, e.g., x, y, θ. Bold font is used to indicate a set of variables, e.g., $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{\Theta}$, or a set of variable value assignments, e.g., $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{\theta}$. A set of variable value assignments $\boldsymbol{x}$ is also called as a *configuration* of $\boldsymbol{X}$. The outcome variable of interest, that which we wish to predict, will be represented by Y. In the coin toss problem Y is the outcome of a coin toss. The outcome value is represented by $Y = y$ or simply y. The probability of the outcome y is represented as $p(Y = y)$ or simply $p(y)$. We use the variable ξ to represent an individual's subjective belief. The classical physical probability of heads from a coin toss is $p(Y = heads)$. The Bayesian probability of heads is $p(Y = heads|\xi)$. $p(Y = heads|\xi)$ can refer to any of Bayesian probability, probability distribution, or a probability density. Its meaning will be clear from the context in which it is used. A single observation of a random variable (or a set of variables) contains an assignment (or a configuration). A collection of observations into a dataset is represented by D. We index the dataset to refer to the i-th observation in D using a superscript for the variable (or set of variables) observed, e.g., X^i.

In the classical approach, say we collect a large number of observations of coin toss outcomes, Y and store it in D. We make a total of m such observations. We need to predict the probability that the next toss outcome would be heads given this data, i.e., $p(Y = heads|D)$.

$$\begin{aligned} p(Y^{m+1} = heads|D, \xi) &= \int_\theta p(Y^{m+1} = heads|\theta, \xi) \cdot p(\theta|D, \xi) \cdot d\theta \\ &= \int_\theta \theta \cdot p(\theta|D, \xi) \cdot d\theta \\ &= \mathrm{E}_{p(\theta|D,\xi)}[\theta] \end{aligned} \tag{2.1}$$

To compute $p(\theta|D,\xi)$, we use the Bayes' theorem, as shown below.

$$p(\theta|D,\xi) = \frac{p(\theta|\xi) \cdot p(D|\theta,\xi)}{p(D|\xi)} \tag{2.2}$$

Here, $p(D|\theta,\xi) = \int_\theta p(\theta|\xi) \cdot p(D|\theta,\xi) \cdot d\theta$. The term $p(D|\theta,\xi)$ is the likelihood function. This term is similar for both classical and Bayesian probability. For binomial outcomes we can use the binomial distribution to compute this, i.e., $\theta^h \cdot (1-\theta)^t$, where h is the number of heads and t is the number of tails. All we need to do is compute the total number of heads h and tails t from our training data to compute this term.

Bayesian probability has this additional prior probability term, $p(\theta|\xi)$. We can encode this using a conjugate prior distribution to the binomial distribution, for example, the Beta distribution. Having a conjugate prior ensures we can compute the integral in the denominator in closed form. We can now compute the prior term using the beta distribution—$p(\theta|\xi) = Beta(\theta|\alpha_h, \alpha_t)$. So, all we need is to estimate, before seeing the data, of all the coin tosses we have experienced relevant to the current situation, how many were heads α_h and how many were tails α_t. There are many probability assessment methods in literature to assess these values from an individual [Heckerman, 2008].

From the objective count of heads and tails from the data, and the subjective count assessments of prior heads and tails from an individual, we now can compute the probability that the next coin toss would be a heads using Equation 2.1. Unlike Bayesian probability of this outcome, classical probability does not include subjective counts and only uses the data to infer the probability. So, using classical probability, the probability of heads is simply the likelihood function without the prior term, $p(D|\theta,\xi)$. We computed this using binomial distribution using only the counts of heads and tails from the data.

2.5.2 Bayesian networks

A Bayesian belief network or Bayesian network (BN) is a probabilistic graphical model, a type of statistical model, represented by nodes and edges, where the nodes represent variables and the edges encode the probabilistic relationships between those variables.

There are many possible ways to represent knowledge in Artificial Intelligence. They

include rules, decision trees, linear models, artificial neural networks, etc. However, BNs have several advantages over other methods [Heckerman, 2008] including—

1. Since BNs encode the probabilistic relationships between all input variables, they are well suited to handle missing values. If two input variables are strongly correlated, if one of the variables has a missing value, the encoded relationship between the two variables can help infer the missing value from the known value of the correlated variable. Most other representations do not encode these relationships.
2. BNs can be used to learn causal relationships [Pearl, 2009]. This is useful when we want to make causal inferences from the model. For example, if we plan to intervene on the cause variable to alter the outcome of the effect variable.
3. Since BNs have both causal and probabilistic semantics, they are suited for encoding prior domain knowledge (often specified in causal form) and combine with the knowledge induced from the data.
4. Bayesian methods offer a principled approach for avoiding overfitting to the data. So, all training data can be used for modeling without the need for a hold out test set. This is particularly attractive for domains with scarce data.

2.5.2.1 Bayesian network representation We begin this section with some notations, which will be used throughout this dissertation. A random variable in the domain is represented with an upper-cased letter, e.g., X or Y. Assume, the domain is composed of n discrete-valued variables. The i-th variable in the domain is represented with a subscript, X_i, where $i = 1, \cdots, n$. Such a domain is said to have n *dimensions* or an n-dimensional dataset. A set of variables is represented with a bold upper-cased letter, e.g., $\boldsymbol{X}$. The domain is therefore $\boldsymbol{X} = \{X_1, \cdots, X_n\} = \{X_{i=1:n}\}$. The discrete value or state taken by a random variable is represented with a lower-cased letter of the variable name, e.g., $X = x$ or simply x. The values or states taken by a set of variables is represented with a bold lower-cased letter, e.g., $\boldsymbol{X} = \boldsymbol{x}$ or simply $\boldsymbol{x}$. These values given to each of the variables in the set is called an *assignment* or a *configuration*.

A Bayesian network (BN) is a probabilistic graphical model that uses directed acyclic graphs (DAGs) to represent the joint probability distribution of the problem domain [Pearl,

2014]. Say, the domain is $\boldsymbol{X} = \{X_{i=1:n}\}$. A BN uses DAGs to encode the joint probability distribution of the domain variables, $p(\boldsymbol{X})$. A BN is represented as a tuple $B = (B_S, \Theta)$, where B_S is the DAG network structure and Θ is a set of numerical parameters encoded in the network. The DAG, B_S, is composed of nodes and directed edges, where a node represents a variable, $X_i \in \boldsymbol{X}$, and directed edges represent probabilistic dependencies between the variables. A missing edge between nodes indicates conditional independencies between the nodes. In this paper, we use the terms node and variable interchangeably. The numerical parameters, Θ, is a set of conditional probability distributions associated with each node in the network. A node X_j is said to be a parent of another node X_i, when there is a directed edge from $X_j \rightarrow X_i$. A set of all nodes that are the parents of X_i is represented as Π_{X_i}. Let us assume that we have a variable ordering, such that, the parents of any variable can only come from variables earlier in the ordering. Then, using the chain rule of probability, we can get the expression for the full joint distribution of the domain from Equation 2.3.

$$p(X) = \prod_{i=1}^{n} p(x_i | x_1, x_2, \cdots, x_{i-1}) \tag{2.3}$$

A variable is independent of its non-descendents, given its parents [Pearl, 2014], i.e., $p(x_i|x_1, x_2, \cdots, x_{i-1}) = p(x_i|\boldsymbol{\Pi_{x_i}})$, where $\boldsymbol{\Pi_{x_i}} \subseteq \{x_1, x_2, \cdots, x_{i-1}\}$. From the BN structure B_S, we can identify the parents of each node. And so, the complex Equation 2.3 can be reduced to Equation 2.4. Such functions that can be factored into a product of the node and its parents are called *node decomposable* functions. Parameter set Θ, associated with the network B is nothing but a set of these conditional distributions associated with each variable in the network. These probabilities are also code local probability distributions.

$$p(X) = \prod_{i=1}^{n} p(x_i | \boldsymbol{\Pi_{x_i}}) \tag{2.4}$$

2.5.2.2 Learning Bayesian networks We now look at learning a BN from data for a classification problem. The discussion in this dissertation is limited to learning of the Bayesian network structure and local conditional probabilities associated with only one variable as shown in Equation 2.4. The variable there would be a target variable of interest,

for example disease outcome. For the purpose of understanding the work done in this dissertation, this knowledge about BNs would suffice. The discussions here will not extend to generalized Bayesian networks constructed over all the variables in the dataset and making probabilistic inference from such a generalized network. For a detailed discussion on those topics, please refer to [Koller and Friedman, 2009].

Assume, we have some data D containing a set of m examples or instances. Each instance here is an independent and identically distributed sample from some true joint distribution over the variables in D. Let's say the variables in $D = \{X_{1:n}, Y\} = \{\boldsymbol{X}, Y\}$, where Y is some outcome variable of interest, for example, disease phenotype like $Y \in \{Case, Normal\}$. This variable is also known as target variable. In a classification problem, this is the variable that we want to predict. The remaining variables in D, i.e., $X_{1:n}$ are possible candidate variables that may help us predict the target variable. We refer to the variable values for the i-th instance in the dataset using a super-script, $\{\boldsymbol{X^i}, Y^i\}$. For example, in gene expression datasets being used to learn differentially expressed genes between samples with abnormal and normal phenotype, the target variable is the phenotype $(normal, abnormal)$ and the candidate variables are the various gene expressions. We assume that all these variables occur earlier in the variable ordering to Y. Under these assumptions, from Equation 2.4 we know that we need only learn the parents of Y to help predict Y, shown in Equation 2.5.

$$p(X) = p(Y|\boldsymbol{\Pi_Y}) \tag{2.5}$$

Of the candidate BN structures (B_S), BRL attempts to find the BN that maximizes the posterior probability of the structure given the observed data, $p(B_S|D)$. From the definition of conditional probability, we can write the expression for posterior probability of the BN structure with Equation 2.6.

$$p(B_S|D) = \frac{p(B_S, D)}{p(D)} \tag{2.6}$$

During the search we compare candidate BN models, say B_{S_1} and B_{S_2}, to evaluate which one is better, and so, the denominator $p(D)$ does not help make this decision. So, we only need to compute the odds of the joint probability of the BN structure and the data, in order

to compare them as shown in Equation 2.7.

$$\frac{p(B_{S_1}|D)}{p(B_{S_2}|D)} = \frac{\frac{p(B_{S_1},D)}{p(D)}}{\frac{p(B_{S_2},D)}{p(D)}} = \frac{p(B_{S_1},D)}{p(B_{S_2},D)} \tag{2.7}$$

In other words, the posterior probability of the BN is proportional to the joint probability of the BN and data as seen in Equation 2.8.

$$p(B_S|D) \propto p(B_S,D) \tag{2.8}$$

From Bayes theorem, the joint probability of the BN and the data can be expressed using Equation 2.9. Here, the joint distribution of the networks and data is expressed as a product of the prior distribution over networks and the likelihood function of the data being generated by the network.

$$p(B_S,D) = p(B_S) \cdot p(D|B_S) \tag{2.9}$$

Buntine [Buntine, 1991], under certain assumptions, proposed a heuristic score called the BDeu (Bayesian Dirichlet equivalence uniform) score to compute the joint probability of the networks and the data. This score is shown in Equation 2.10.

$$p(B_S,D;\alpha) = p(B_S) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\frac{\alpha}{q_i})}{\Gamma(N_{ij}+\frac{\alpha}{q_i})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk}+\frac{\alpha}{r_i q_i})}{\Gamma(\frac{\alpha}{r_i q_i})} \tag{2.10}$$

Here, $p(B_S)$ is the prior distribution over the possible network structures. Index i iterates through each of the n nodes in the network structure. Index j iterates though each of the q_i possible variable-value assignments (or configuration) of the parents of node i. Index k iterates through each value taken by node i, with r_i being the number of values it can take on. Hyperparameter α, also called the *prior equivalent sample size*, expresses the strength of our belief in the prior distribution over the networks. The expression $\frac{\alpha}{r_i q_i}$ assigns a uniform prior probability to each parental configuration. $\Gamma(\cdot)$ is the gamma function, where $\Gamma(x+1) = x!$. N_{ijk} is the number of instances in D, where node i takes the value k, while its parents take the configuration j. And, $N_{ij} = \sum_k N_{ijk}$.

2.6 RULE LEARNING

Machine learning is a sub-domain in computer science concerned with building algorithms for the discovery of mathematical models, patterns, and other regularities in the data [Mitchell et al., 1997]. There are primarily two kinds of machine learning methods— 1) symbolic methods, and 2) statistical methods. Symbolic methods perform inductive learning of human-readable symbolic descriptions, like rules and decision trees, from the data (example algorithms include C4.5 [Quinlan, 2014], PART [Frank and Witten, 1998], RIPPER [Cohen, 1995]). Statistical methods infer parameters of statistical models from the data. These include human-readable models like— generalized Bayesian networks [Pearl, 2014] and unreadable models like— logistic regression [le Cessie and van Houwelingen, 1992], support vector machines [Platt, 1999], and ensemble methods [Breiman, 1996, Freund et al., 1996, Seni and Elder, 2010].

Given a suitable choice of machine learning method, the data analysis problem itself can be categorized under different categories based on their properties. The primary two types of problems are— 1) supervised learning and 2) unsupervised learning. Supervised learning involves models learned with respect to a target variable of interest. Unsupervised learning involves general pattern discovery from the data, without focusing on a particular variable. A *model* usually refers to a global explanation of the entire input training dataset. *Pattern* is a local hypothesis that only explains a subset of the training data. In unsupervised learning, patterns can help describe a subpopulation that is over-represented in a dataset. In supervised learning, patterns are useful to help describe a subpopulation that behaves differently with respect to a target variable. For example, in biomedicine, we can discover patterns that describe a subpopulation that responds differently to a treatment. Further study of this subpopulation may help discover what is different about this subpopulation that leads to a different physiological process from the general population.

Rules are an effective representation of patterns. Classification rules are induced by supervised rule learning algorithms from the data, such that they can describe a subpopulation in association with a target variable. For example, the rules learned by most rule learning algorithms are in form of explicit propositional logic statements i.e., IF $\langle antecedent \rangle$-

THEN$\langle$*consequent*$\rangle$. The rule antecedent is a sequence of predictor variable-value pairs joined by the logical-AND. This antecedent provides a subpopulation description, in form of a condition using a logical series of variable-value pairs, which if all true, implies that the rule consequent is also likely to be true. The consequent part of the rule provides a variable value of the target variable of interest. For example, if the predictor variables can be a set of genes (say, *Gene1* and *Gene2*) in gene-expression data. The target variable of interest can be a disease phenotype (say, disease outcome represented by T). To represent a rule that indicates that if an individual has up-regulated *Gene1* and down-regulated *Gene2*, then as a consequence, the individual is at risk of developing the disease T. This sentence is represented using propositional logic as follows— IF $\langle(Gene1 = UP)\text{AND}(Gene2 = DOWN)\rangle$ THEN $\langle T = true\rangle$.

Therefore, by representing in form of IF $\langle$*antecedent*$\rangle$-THEN$\langle$*consequent*$\rangle$, rules of this nature provide both descriptive statistics, to help describe a subpopulation, and predictive statistics, that describes how the subpopulation differs within the general population with respect to a target variable of interest. Due to these attractive properties, rule models are very popular in biomedicine, including for clinical decision support systems.

Propositional rule learning is a rule learning method that learns patterns and/or models, expressed in form of propositional logic, from a dataset [Fürnkranz et al., 2012]. Another popular form of supervised rule learning is relational learning. Relational learning involves inducing patterns/models expressed using relational formalism of first-order logic [Lavrac and Dzeroski, 2001].

A rule model is a collection of predictive rule patterns, together explain the training dataset, and ideally the domain as a whole. There are two major approaches in rule learning for inducing a complete rule model [Fürnkranz et al., 2012]— 1) *decision tree induction*, and 2) *rule set induction*.

A *decision tree* consists of a tree structure composed of *nodes* and *edges*, where the nodes represent variables and edges represent values taken by the variable. The top-most node of the model is called the *root* of the tree. The set of nodes in the bottom of the tree, which themselves do not have edges are called *leaves*. In supervised learning the leaves of the decision tree represent the target variable-value distributions. The path from the root to the

leaf is a sequence of node-edge relationships that describes a subpopulation in the dataset. The leaf typically encodes the target variable-value distribution. As it can be clearly seen, each path in this tree can be represented as a rule, where the path from root to leaf is the rule antecedent, and the target variable-value distribution determines the rule consequent.

The goal of the tree learning algorithm to have leaves with examples primarily of one of the values of the target variable. Having such a tree model helps in differentiating the various target values via different branches representing varying paths. For example, it will help us identify patterns that differ between cases and controls for a target variable representing disease outcome. During learning, the tree is specialized by adding variable-values to a leaf, thereby describing a smaller subpopulation. By so, doing the leaf has fewer examples from the original dataset. So, any target value distribution would have fewer evidence to support the pattern described by the pattern from the root to leaf. This problem is called *data fragmentation*. This is due to the restriction imposed by the tree data structure. One positive reason to use decision tree to model your data is because they are exclusive and exhaustive explanation of your entire dataset. So, for an unseen test example, there will be a prediction made by the decision tree, and only one rule will fire. A popular example of a decision tree is the C4.5 tree induction algorithm [Quinlan, 2014]. It uses a concept of information entropy to learn decision tree with leaves each representing examples primarily belonging to one of the values of the target variable. Bayesian Rule Learning is another example of a rule learning method that uses decision tree for learning. This method is explained in better detail in the next section (see 2.7).

The other type of rule model learning is *rule set induction*. Unlike decision trees, rule set induction methods do not attempt to explain the entire training dataset. So, while decision trees contain mutually exclusive rules, with rule sets, we may have overlaps. When multiple rules explain a single example, we need a way to resolve the rule conflict to decide on the rule consequent. An advantage of this approach is if we face data scarcity, for example in case of high-dimensional datasets, we can still learn local patterns without having to specify a global hypothesis that explains the entire training dataset. On the other hand conflict resolution of multiple rules firing for the same example, and sometimes no rules explaining a given example, make this model unfavorable in certain applications.

Examples of rule set induction algorithms include— Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [Cohen, 1995], PART (short for partial decision trees) [Frank and Witten, 1998], and Rule Learning (RL) [Clearwater and Provost, 1990]. RIPPER splits the training data into growth data and prune data. It greedily learns a rule for one target-value from the growth data. Then based on a heuristic evaluated over prune data, it generalizes the greedily learned rule from the growth data. One rule for each target value is learned this way. Then in the optimization step, the rule with the largest heuristic score on the prune data is selected. The examples covered by this rule is removed from the data, and the process is repeated. PART iteratively learns a partial C4.5 tree, and selects the leaf with the most coverage, turns it into a rule and adds it to a rule set, removes the examples covered by the rule, and re-iterates the process until no examples are left. PART does not optimize the model on a global metric and yet was found to be better than RIPPER and similar to C4.5 [Frank and Witten, 1998]. RL [Clearwater and Provost, 1990] uses breadth-first marker propagation to specialize the rules. It uses inductive strengthening to select newly learned rules, such that the new rule must cover a certain number of new examples from the training dataset.

For a thorough overview of rule learning methods, please refer to [Fürnkranz et al., 2012].

2.7 BAYESIAN RULE LEARNING

Bayesian Rule Learning (BRL) [Gopalakrishnan et al., 2010] is a rule-based classifier that takes as input, a dataset and returns a rule set model. A rule model is a set of mutually exclusive and exhaustive rules that can be applied to new data to predict a target variable of interest. Mutually exclusive rules mean that only one rule fires for a unique instance representing an individual observation in the dataset. Exhaustive rules mean that at least one rule fires for a given individual. Unlike traditional rule learning methods, BRL quantifies the uncertainty of the validity of the rule model using a Bayesian score. It uses this score for model selection.

Let the dataset D be an observed instantiation of a system with a probability distribution over a set of n random variables and a target random variable of interest, $D = \{X_i, T; i \in 1 \cdots n\}$. Here, T is the target variable of interest, which is the dependent variable for the prediction task. Each of the other variable, X_i in D is an independent random variable that may help predict T. There are a total of m instances in D. In the classification problem, our task is to accurately predict the value of the target variable.

The BRL search algorithm explores a space of probabilistic graphical models called Bayesian-belief networks (BNs), learned from observed dataset D and returns the most optimal BN found during the search. A BN is a tuple, $B = (B_S, \Theta)$, where B_S is the network structure, and Θ is the network parameters. The network structure consists of a directed acyclic graph (DAG). The nodes represent variables, and variables are related to each other by directed arcs that do not form any directed cycles. When there is a directed arc from node A to node B, node B is said to be the *child node*, and node A is said to be the *parent node*. The network parameters Θ are a conditional probability distributions over each node in the network. A probability distribution is associated with each node, X, in the graphical structure given the state of its parent nodes, $\theta_X = p(X|\Pi(X))$, where $\Pi(X)$ represents the different discrete value assignments of the parents of node X. A constrained BN, that BRL learns, is the network structure of T and its parents, $\Pi(T)$. This probability distribution is generally called a conditional probability distribution (CPD). For discrete-valued random variables, the CPD can be represented in form of a table called conditional probability table (CPT) [Koller and Friedman, 2009, Chickering et al., 1997]. Furthermore, any CPT can be represented as a rule base. Here, we consider only the CPT for the target variable. Each possible value assignments of the parents represent a different rule in the rule base. The evidence in form of the distribution of instances, for each target value, in the training data helps infer the rule consequent. The resulting rule base consists of rules that are mutually exclusive and exhaustive. In other words, at least one rule from the rule base matches a given instance and only one rule matches that instance.

The BRL rules are represented in the form of explicit propositional logic, IF $\langle antecedent \rangle$-THEN$\langle consequent \rangle$, as described in section 2.6. The rule antecedent is the condition made up of conjunctions of the independent random variable-value pairs, which when matched to

a test instance, implies the rule consequent composed of the dependent target variable-value.

The search algorithm searches to find the BN model that maximizes the search heuristic of Bayesian score, using the K2 score [Cooper and Herskovits, 1992]. K2 is a popular Bayesian Score from the Bayesian Dirichlet Score family used to compute the joint probability $P(B_S, D)$. It makes the following four assumptions—

1. The database variables are discrete.
2. The instances in the dataset are independent and identically distributed.
3. There are no missing values.
4. A uniform prior probability distribution of the network parameters, given its structure.

The K2 score (used by BRL) is shown in Equation 2.11.

$$P(B_S, D) = P(B_S) \prod_{j=1}^{q} \frac{(r-1)!}{(N_j + r - 1)!} \prod_{k=1}^{r} N_{jk}! \tag{2.11}$$

Here, $P(B_S)$ is the prior probability of the BN being the data generating model. [Gopalakrishnan et al., 2010], set this to be 1, which indicates an uninformative prior where *a prior* no BN is more likely to be correct than another. The rest of the term is the likelihood function that computes the probability that the data D was generated by the BN, B_S. Variable j iterates through all q different variable-value instantiations of the parents of the target variable. The number of values the target variable takes, r. N_j is the number of instances in D that take the j-th instantiation of predictor variables. Variable k iterates through the different values of the target variable. N_{jk} is the number of instances in the dataset with j-th instantiation of the predictor variables and the target variable takes value k.

Figure 4 shows an example of a BN structure learned from BRL. Panel (a) displays a constrained BN structure (B_S) with two predictive variables, $Gene1$ and $Gene2$, as parents of the target variable T. The two predictive variables are binary with values of UP and $DOWN$. The target variable is binary having values *Case* and *Control*. Figure 4(b) shows the parameters for the target node as a complete decision tree. The interior nodes of the tree are the predictive variables (represented by ellipses) and the leaf nodes (represented by rectangles) show the probability distribution over T. Figure 4(c) shows the rule set inferred from the decision tree by BRL. Each rule antecedent is a path from a leaf to the root node.

The consequent is the probability distribution of T. The following parentheses show the number of $Case$ instances and the number of $Control$ instances that match the antecedent, respectively.

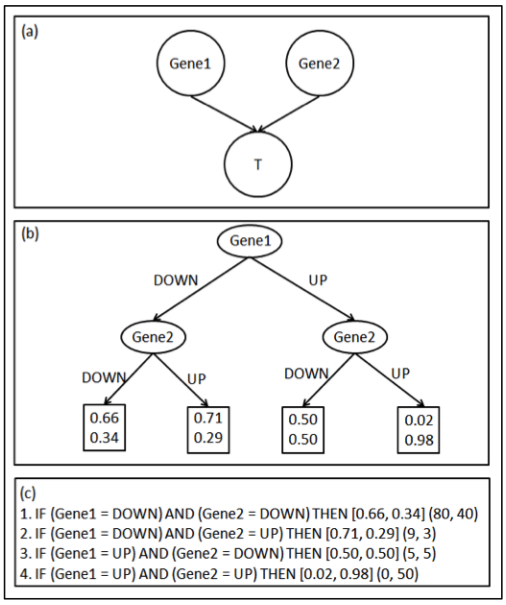


Figure 4: Bayesian Rule Learning (BRL): (a) Bayesian network (BN) learned by BRL. (b) decision tree inferred from the BN. (c) a set of mutually exclusive and exhaustive rules inferred from the decision tree.

[Gopalakrishnan et al., 2010] show a greedy best-first search, and a beam search to search over the space of BNs. The search takes as input— 1) the training dataset, D, 2) MAX_CONJ, the maximum number of predictor variables that can be included into BRL. They compared BRL with greedy-best-first and beam search algorithms to 3 popularly used rule- or decision tree-based classifiers— Conjunctive Rule Learning, RIPPER [Cohen, 1995], and C4.5[Quinlan, 2014]. They evaluated on balanced accuracy (average of sensitivity and specificity) and relative classifier information (RCI). This comparison was done on 24 publicly available high-dimensional datasets. Over the 24 datasets they show that BRL performs statistically significantly better than Conjunctive Rule Learning, RIPPER, and C4.5 both in terms of balanced accuracy and RCI. BRL was also shown to require significantly fewer variables for prediction than C4.5. Fewer variables mean fewer biomarkers for subsequent validation, which can be important for biomarker discovery.

3.0 BAYESIAN RULE LEARNING METHODS DEVELOPMENT

In this chapter, I describe the Bayesian Rule Learning (BRL) methods developed in the Bayesian Rules for Actionable Informed Decisions (BRAID) system to help solve the Knowledge Discovery in Databases (KDD) problem (as defined and described in Section 2.4). Section 3.1 describes the BRL search algorithm. Here, I explore three model representations of BRL that may help improve the *understandability* of BRL rule model (as explained with the KDD definition in Section 2.4). I also describe the model search algorithm and the heuristic score used to evaluate the quality of the models encountered during the search. To improve the *validity* of the BRL model (per the KDD definition), I study two methods—an ensemble method and a method to incorporate prior domain knowledge into the model learning process. Specifically, Section 3.2 describes the Ensemble Bayesian Rule Learning (EBRL) methods that explore different ensemble techniques applied to BRL models. Ensemble methods are a popular approach in statistics known to efficiently improve the predictive performance of a classifier. An important drawback of ensemble methods is the loss of *understandability* of the model. To help overcome this, I describe a novel visualization method to help make the EBRL model more understandable to the user. Section 3.3 describes the second method to help improve model validity called BRL with informative priors (BRL_p). BRL_p enables BRL to incorporate prior domain knowledge into the model learning process. Finally, section 3.4 describes a BRL method designed to search *novel* and *useful* patterns (per the KDD definition). The method is called BRL for knowledge discovery (BRL-KD). It enables BRL to incorporate information about the relative clinical relevance of each variable and uses this information to search for models that are clinically more relevant.

3.1 BAYESIAN RULE LEARNING (BRL)

The Bayesian Rule Learning (BRL) [Gopalakrishnan et al., 2010] search algorithm searches over a hypothesis space of Bayesian networks (BN) to identify the BN most likely to have generated the observations in a given dataset, D. BRL then infers a set of rules from the parameters of this BN. Sub-section 3.1.2 describes different representations of the parameters of the BN that were explored in this study. Sub-section 3.1.3 explains the heuristic score used to evaluate the quality of a BN in terms of explaining the observations in D. Finally, sub-section 3.1.4 describes a search algorithm to help find the BN that maximizes the heuristic score.

3.1.1 Background and motivation

The joint probability distribution over all variables of a domain provides useful predictive and descriptive insights into the modeled system. Computing this joint probability distribution is expensive for even moderately-sized problems because the number of parameters of the joint distribution grows exponentially with the number of variables in the domain. The graphical structure of a Bayesian network (BN) represents dependence relationships between variables in the domain. Particularly, the absence of an edge between two nodes (or variables) indicates independence. It is these absence of edges that help BNs decompose the joint probability distribution into a product of modularized smaller distributions, thereby significantly reducing the number of parameters in the joint probability distribution of the domain [Pearl, 2014]. The conditional independence in a BN is defined below.

Definition 2.1.1: *Conditional independence*— Let $\boldsymbol{U} = \{\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}$ be a domain of three disjoint subsets of variables ($\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$). To claim that $\boldsymbol{X}$ is independent of $\boldsymbol{Y}$, given $\boldsymbol{Z}$ i.e., $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y}|\boldsymbol{Z}$ implies that Equation 3.1 is true.

$$p(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{Z}) = p(\boldsymbol{X}|\boldsymbol{Z}) \cdot p(\boldsymbol{Y}|\boldsymbol{Z}) \tag{3.1}$$

Conditional independence is a numerical property of the domain variables. BN structures explicitly encode these independencies using a graphical structure (described in detail in

Section 3.1.2). Conditional independencies over the variables of the domain are also called *global independencies*.

In addition to the graphical structure, BNs also encode parameters, which are probability distributions associated with each variable of the domain. They are probability distributions conditioned on the parents of the variable in the BN. They are typically encoded as tables and are called conditional probability tables (CPTs) [Koller and Friedman, 2009]. CPTs with respect to a variable X represents the probability distribution over its values given all the possible assignments of its parent values. CPTs are typically represented as a complete decision tree with all leaves at the same depth. The nodes of this tree are the parent variables of the node. The different edges from a node represents the different values assigned to the variable in the node. The depth of the complete tree is the same as the total number of parents. BRL parses this complete tree from root to each leaf and translates it into a propositional logic rule.

However, there exist certain regularities in the probability distribution in the rows of a CPT. With a table or a complete tree representation, these regularities cannot be expressed explicitly. These regularities represent context-specific independence (CSI) [Koller and Friedman, 2009, Boutilier et al., 1996] as defined below.

Definition 2.1.2: *Context-specific independence*— Let $\boldsymbol{U} = \{\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\}$ be a domain of three disjoint subsets of variables ($\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$). Let $\boldsymbol{C}$ include overlapping variables from the set $\{\boldsymbol{X} \cup \boldsymbol{Y} \cup \boldsymbol{Z}\}$. A context, $\boldsymbol{C} = \boldsymbol{c}$ is a configuration of $\boldsymbol{C}$. To claim that $\boldsymbol{X}$ is contextually independent given $\boldsymbol{Z}$ and context $\boldsymbol{c}$ i.e., $\boldsymbol{X} \perp\!\!\!\perp_{\boldsymbol{c}} \boldsymbol{Y} | \boldsymbol{Z}, \boldsymbol{c}$ implies that Equation 3.2 is true.

$$p(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{c}) = p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{c}) \tag{3.2}$$

The regularities of form that indicate CSIs are quite common occurence in data [Boutilier et al., 1996]. CSIs are also a numerical property like conditional independences but they are not represented in the BN graphical structure. They also cannot be represented using a complete decision tree representation of the parameters of the BN. We need special representations to encode these CSIs.

The main advantage of explicitly encoding these CSIs is that there are much fewer number of rules. Rule models with fewer rules are more efficiently represented and therefore more

understandable. Fewer rules mean there are fewer parameters of the BN model to either learn from the data or to elicit priors for them from an expert.

Like Bayesian networks, rule models (e.g. BRL) are both— a descriptive statistical model and a predictive statistical model. By enabling BRL to represent CSIs we have a unique model that has an additional benefit over plain BNs. As a descriptive model, unlike BNs, BRL can capture both global condition independences from the BN structure and also CSIs from the propositional rules. To make a prediction, BRL uses the explicit propositional rules containing a rule consequent. If the condition part of the propositional rule is satisfied, it implies that the outcome described by the rule consequent, is more likely. For biomarker discovery it can be beneficial to know the CSIs of a subpopulation. In personalized medicine, this could mean a molecular subtype whose condition does not depend upon certain variable that is globally picked by the model. To learn such independencies, we modified BRL to include representations that can encode CSIs.

Previous works to encode CSIs largely pertain to extending BNs to represent CSIs. [Heckerman, 1990] used probabilistic similarity networks, which are a type of influence diagrams that can represent CSIs. [Friedman and Goldszmidt, 1998] and [Boutilier et al., 1996] use decision trees to represent CSIs. Decision trees can capture many types of regularities in the CPT but not all regularities. [Chickering et al., 1997] proposed the use of decision graphs to represent all regularities in CPTs. [Jabbari et al., 2018] show that an instance-specific learning approach to finding BNs, represented using a decision tree [Boutilier et al., 1996], can retrieve CSIs with high precision.

There has also been an earlier work in using rule set representation of BN that can explicitly encode CSIs. This was done using a language called probabilistic Horn abduction [Poole, 1993]. Such a representation can easily be coded into the Prolog programming language. The work in this dissertation differs in many important ways. Firstly, Bayesian score is used to search for the BN and the parameters. Therefore, we have a measure of the uncertainty in the validity of the model from which the rules were generated. This will be helpful in the implementation in Section 3.2 of this dissertation. Secondly, the BN parameters are encoded using propositional logic, which are simple IF-THEN rules.

The decision graphs proposed by [Chickering et al., 1997] is the basis of the approach

used in this dissertation to learn rules for BRL. These decision graphs can represent all possible CSIs. However, decision graphs can become fairly complex with more and more variables. In the following sections, I parse these learned decision graphs to be represented as explicit propositional logic rules in disjunctive normal form to help make these decision graphs more readable.

3.1.2 Model representation

The conditional probability table of the constrained BN (see Equation 2.5) of the BN is represented here in form of a decision tree. For example, see Figure 4a. The constrained BN contains one target node Y with two parents X_1 and X_2. The conditional probability distribution is represented in the table next to the BN. Note that when $X_1 = 1$, regardless of the value X_2 takes, the distribution over the values of Y does not change. In this context, where $X_1 = 1$, Y is independent of X_2. So, in the learned decision tree by BRL (see Figure 4b) the leaf containing X_1 is not specialized further with X_2. Each path from the root to the leaf of this tree is a rule as seen in Figure 4c. In the rules, TP corresponds to the number of true positives for the rule i.e., out of the number of instances in the training dataset that matches the left hand side (LHS) of the rule, the number of instances that were labeled with the positive class, say *Case*, is 50 for rule 1. The number that were labeled with the negative class, say *Normal*, is 5 for rule 1. How we compute the probability distribution values will be clear in the next subsection (with Equation 3.6 to be specific).

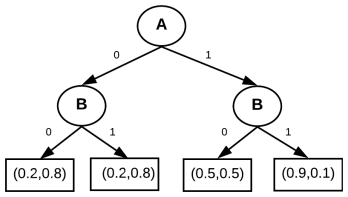
3.1.2.1 Bayesian Rule Learning— Global Structure Search with Complete Decision Trees (BRL.G)

The BRL with global structure uses a complete decision tree to represent the CPT, as shown in Figure 5a.

Figure 5a depicts a possible conditional probability distribution table. The first row of the table says that if variable A takes the value 0, and variable B takes the value 0, then according to this probability distribution, the probability of such an instance having the variable T value as 1 is with a probability 0.8. Each row is a probability distribution conditioned on the values taken by A and B. So, the probability distribution of each row

A	B	p(T | A, B)	
		T = 0	T = 1
0	0	0.2	0.8
0	1	0.2	0.8
1	0	0.5	0.5
1	1	0.9	0.1

(a) CPT



(b) CPT represented as a complete tree.

Figure 5: Model representation in BRL.G

sums to 1.

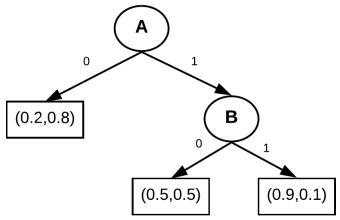
The complete decision tree in Figure 5b represents each row of the CPT with a leaf. BRL parses each path from root to leaf of this tree. Each path is then translated into an IF-THEN rule.

3.1.2.2 Bayesian Rule Learning— Local Structure Search with Decision Trees (BRL.DT) We notice in BRL.G representation that the first two rows in the CPT (see Figure 5a) have the same distribution. Yet BRL.G represents them as separate rules. This is an example of context specific independence. The value distribution over T only depends upon the value taken by A. In fact these instances are independent of the value of B. BRL.DT representation (see Figure 6) helps overcome this problem. in BRL.DT, the first two rows with the same distribution removes the dependency for those rule with variable B. This can be beneficial to biomarker validation as for this subpopulation, there was no need to validate variable B.

This representation alone was studied and evaluated in my earlier work [Lustgarten et al., 2017]. In this dissertation study, I will also include the decision graph representation as described in the next section.

A	B	p(T | A, B)	
		T = 0	T = 1
0	0	0.2	0.8
0	1	0.2	0.8
1	0	0.5	0.5
1	1	0.9	0.1

(a) CPT



(b) CPT represented as a generalized decision tree.

Figure 6: Types of context-specific independencies represented in BRL.DT

3.1.2.3 Bayesian Rule Learning— Local Structure Search with Decision Graphs (BRL.DG)

There are certain kinds of independencies that BRL.DT cannot represent. An example of that is shown in Figure 7.

Here, regardless of the value take by variable A, if variable B takes the value 0, then the distribution over T remains the same. This means that, this subpopulation is independent of the root of the tree. We represent that with a decision graph. Here, we can see a benefit of representing the CPT as a decision graph that enables us to find a regularity that could not be discovered using the decision tree structure. A being the root node of a decision tree would be incorrectly inferred as a variable that all subpopulations depend upon. However, we see in this example that this is not necessarily true.

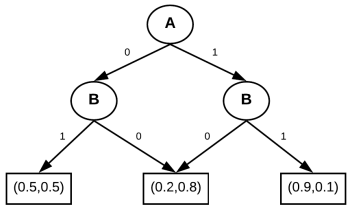
To translate decision graph into rules, we use a disjunctive normal form. The logical OR helps account for multiple paths from the root to leaf of a decision graph.

3.1.3 Heuristic score

We use the same heuristic score as in Section 2.5.2.2 called the BDeu score [Buntine, 1991]. The equation for the joint probability of the BN structure and data using BDeu score is

A	B	p(T | A, B)	
		T = 0	T = 1
0	0	0.2	0.8
0	1	0.5	0.5
1	0	0.2	0.8
1	1	0.9	0.1

(a) CPT



(b) CPT represented as a generalized decision graph.

Figure 7: Types of context-specific independencies represented in BRL.DG

shown below.

$$p(B_S, D; \alpha) = p(B_S) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\frac{\alpha}{q_i})}{\Gamma(N_{ij} + \frac{\alpha}{q_i})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \frac{\alpha}{r_i q_i})}{\Gamma(\frac{\alpha}{r_i q_i})} \tag{3.3}$$

Here, $p(B_S)$ is the prior distribution over the possible network structures. Index i iterates through each of the n nodes in the network structure. Index j iterates though each of the q_i possible variable-value assignments (or configuration) of the parents of node i. Index k iterates through each value taken by node i. Hyperparameter α, also called the *prior equivalent sample size*, expresses the strength of our belief in the prior distribution over the networks. The expression $\frac{\alpha}{r_i q_i}$ assigns a uniform prior probability to each parental configuration. $\Gamma(\cdot)$ is the gamma function, where $\Gamma(x+1) = x!$. N_{ijk} is the number of instances in D, where node i takes the value k, while its parents take the configuration j. And, $N_{ij} = \sum_k N_{ijk}$.

While modeling high-dimensional data, it is preferable to have models with fewer variables needed to predict the target variable. So, sparser models are preferred. One way to achieve this is to create a penalty on the number of parents added to the target variable. We can use the prior distribution over networks, $p(B_S)$, to favor models that are sparser. Koller et al. [Koller and Friedman, 2009], describes a node decomposable prior distribution score to help favor models with fewer parents added to any given node in the network. This prior

term is shown in Equation 3.4.

$$p(B_S;\kappa) = \prod_{i=1}^{n} \kappa^{|\Pi_i|} \tag{3.4}$$

Here, the power term $|\Pi_i|$ is a count of the total number of parents that node i has in structure B_S. Hyperparameter κ is some value $0 < \kappa \leq 1$. A value of $\kappa = 1$ represents a uniform prior over all network structures. Smaller values of κ generates a distribution that prefers structures with nodes having fewer parents. Larger values for κ gives a distribution that allows more parents.

For a constrained Bayesian network, we only need to compute the score for one node, the target variable Y. Substituting Equation 3.4 into Equation 2.10 and reducing the terms to just one node Y, we get Equation 3.5.

$$P(B_S, D; \alpha, \kappa) = \kappa^{|\Pi_i|} \cdot \prod_{j=1}^{q_Y} \frac{\Gamma(\frac{\alpha}{q_Y})}{\Gamma(N_j + \frac{\alpha}{q_Y})} \prod_{k=1}^{r_Y} \frac{\Gamma(N_{jk} + \frac{\alpha}{r_Y q_Y})}{\Gamma(\frac{\alpha}{r_Y q_Y})} \tag{3.5}$$

BRL search algorithm searches over a space of constrained Bayesian networks to find the model that maximizes the heuristic score shown in Equation 3.5. Ultimately, the goal of the model is to make a prediction on the probability that a new test instance, t, belongs to a particular class k i.e., $p(Y^t = k|\boldsymbol{X^t} = \boldsymbol{j}, B_S, D)$ or $\theta_{jk} \in \Theta$. These probabilities can be computed using the expectation of the model parameter for node Y taking class value k and its parent configuration taking state j that corresponds to test instance $\boldsymbol{X^t}$. This is computed using Equation 3.6.

$$p(y^t = k|\boldsymbol{x^t} = \boldsymbol{j}, B_S, D) = \mathbb{E}[\theta_{jk}|B_S, D] = \frac{N_{jk} + \frac{\alpha}{r_Y q_Y}}{N_j + \frac{\alpha}{q_Y}} \tag{3.6}$$

To get BRL to make a prediction, we need to choose a cut-off. If the predicted probability for a class, say c, exceeds this cut-off value, then the classifier decides to label the test instance with class c. The choice of the cut-off depends upon the application domain and the cost of false positives and false negatives. Assuming no such information is available, by default, BRL simply classifies the instance with the class value $c \in C$ with the highest predicted class

probability. Here, C is a set of all r_Y classes. This is shown in Equation 3.7.

$$\hat{y} = \arg\max_{c \in C} p(y^t = c | \boldsymbol{x^t} = \boldsymbol{j}, B_S, D) \tag{3.7}$$

3.1.4 Search algorithm

BRL uses a greedy best-first search algorithm to find the model that best explains the observed dataset, D. This BRL algorithm pseudocode is presented in Algorithm 1. The algorithm takes a training data, D as an input. The user specifies the variable in the dataset that is the target variable of interest (or dependent variable) Y. All other variables are treated as candidate predictive or independent variables in the system. The algorithm outputs the BN represented as a decision tree that maximizes the Bayesian score in Equation 3.5. For this paper, we fix the value of $\alpha = 1.0$ and $\kappa = 0.01$. In practice, values of these hyperparameters can be explored to further optimize the learned model. BRL parses the resulting decision tree into a set of rules, one for each path from root to leaf as shown in Figure 4c.

There are three main model specialization operators used in the algorithm, namely— 1) *complete split*, 2) *binary split* and 3) *merge*. For a predictive variable, X_i, with s values, the complete split operator splits the data at the leaf of the decision tree into s branches, one for each possible value of X_i. The resulting s leaves only contains examples from D that take the values specified by the branch. Binary split operator splits the data into 2 branches. If the number of possible values, $s > 2$, for a variable X_i, the binary split operator merges branches from a complete split to find all possible binary splits. For example, if the set of values that X_i takes is $\{0, 1, 2\}$, the binary split operators splits them as $\{\{0, 1\}, \{2\}\}$, $\{\{0, 2\}, \{1\}\}$, and $\{\{1, 2\}, \{0\}\}$. All these resulting trees are generated from this operator in the algorithm. The merge operator merges any two leaves from the tree.

In steps 1-3 of the greedy best-first search algorithm, we create singleton BNs. Singleton BNs have two nodes— target variable Y and a parent selected from the set of candidate predictive variables $X_{i:n}$. In the decision tree, the root of the tree undergoes both complete split and binary split from a candidate variable and the tree with the best Bayesian score

Algorithm 1: BRL GREEDY BEST-FIRST SEARCH

Input : Data ($D = \{X_{i=1:n}, Y\}$), where $X_{i=1:n}$ are discrete-valued attributes and Y is the user-specified target variable. BRL $algorithm = \{G, DT, DG\}$, where G is *complete tree*, DT is *decision tree*, and DG is *decision graph*.

Output: Optimal model M_{best} found using greedy best-first search.

Create Priority Queue *beam* that sorts models contained in them by their Bayesian score in descending order;
foreach *Attribute* $X_i \in D$ **do**
    Create a constrained BN with one parent $\Pi(Y) = X_i$ to target Y;
$sortedAttributes \leftarrow$ add attributes $X_i \in D$ in order in which they appear in *beam* models each with one attribute as parent;
$bestModel \leftarrow beam.poll()$;
$iterationImprovedModel \leftarrow true$;
while $iterationImprovedModel$ **do**
    $beam \leftarrow \emptyset$;
    $iterationImprovedModel \leftarrow false$;
    foreach *Attribute* $X_i \in sortedAttributes$ **do**
        if $algorithm = DG$ **then**
            foreach *Leaf* $l \in bestModel$ **do**
                $beam \leftarrow$ add `Complete-Split` ($bestModel$, X_i, l);
        if $algorithm = DT\ OR\ DG$ **then**
            $beam \leftarrow$ add `Complete-Split` ($bestModel$, X_i);
            $beam \leftarrow$ add `Binary-Split` ($bestModel$, X_i);
        if $algorithm = DG$ **then**
            $beam \leftarrow$ add `Merge` ($bestModel$, X_i);
    $bestModelInIteration \leftarrow beam.poll()$;
    if $bestModelInIteration.score() > bestModel.score()$ **then**
        $bestModel \leftarrow bestModelInIteration$;
return $bestModel$;

is selected to represent the singleton BN for that candidate variable. We create a singleton model for each of the candidate variables in the dataset. These BNs are added to a priority queue that sorts the models in descending order of their Bayesian scores. So, the top of the queue contains the model with the highest Bayesian score. Step 4 selects the BN with the highest Bayesian score. This model is the best one seen so far and undergoes further specialization. The specialization happens in the loop from steps 6-14. At step 7, the priority queue is cleared. Steps 9-11, applies the complete split and the binary split operator once for each leaf of the decision tree using each variable in the dataset. The resulting decision trees are added to the priority queue. When the split operator is being applied to a leaf of the decision tree, if the path from the root to that leaf already contains a specific variable, then that variable isn't used to split the leaf again. After all the specializations are done, we pick the model on top of the priority queue. We compare the Bayesian score of this model to the one that was being specialized at the beginning of the loop. If the score of the specialized model was higher, then the iteration appears to have helped improve the score. So, this model, with the highest score seen so far, is put into another specialization loop from steps 6-14. The loop breaks if the iteration does not help improve the model score.

Finally, the model with the highest Bayesian score seen so far is returned as the best model. BRL parses each path from root to leaf and translates them into IF-THEN rules as shown in Figure 4c.

3.2 ENSEMBLE BAYESIAN RULE LEARNING (EBRL)

In this section, I describe the methods I developed to enable BRL to model multifactorial diseases. I start with the motivation and some background, in subsection 3.2.1, explaining why this problem is currently difficult for BRL to solve due to data fragmentation. In subsection 3.2.2, I propose and implement ensemble methods in BRL to help model multifactorial diseases. Subsection 3.2.3 shows how we can calculate variable importance from these ensemble models. In subsection 3.2.4, I propose a novel method to visualize the ensemble model.

3.2.1 Background and motivation

As described earlier, omic datasets are challenging for data analysis due to their high-dimensionality. In addition to that, they are often collected from samples with complex, multifactorial diseases. Single factor diseases only have one biomarker associated with the outcome of interest. Single factor diseases include monogenic disorders (or Mendelian disorders), where variation in one gene is the cause of the disease. Examples of single factor diseases include cystic fibrosis (caused from variants of the *CTFR* gene) [Riordan et al., 1989] and Huntington's disease (caused from variations in *HTT* gene) [Vonsattel and DiFiglia, 1998]. Single factor diseases, while important to study themselves, are relatively rare compared to the more prevalent, multifactorial diseases [Antonarakis and Beckmann, 2006]. The more common diseases like type II diabetes [Fuchsberger et al., 2016] and coronary heart disease [Poulter, 1999] are known to be multifactorial, where many common genetic variants, each with a small effect, collectively increase the disease risk. We need data mining methods for biomarker discovery that can efficiently learn models from high-dimensional datasets for multifactorial diseases.

BRL is very adept for biomarker discovery from high-dimensional datasets and has been shown to perform better than state-of-the-art classifiers typically used in such applications [Gopalakrishnan et al., 2010, Lustgarten et al., 2017]. A single BRL model alone is not sufficient to model multifactorial diseases because it selects a small set of variables to learn the model. Similar to learning decision trees, BRL also suffers from *data fragmentation*, where model specialization by addition of more variables in the model leads to the training data being split into smaller subsets. As a result, to test the association between a new variable and the target, after each specialization, we are left with fewer examples to evaluate the relationship from. This leads to a loss of statistical power to validate any new associations.

One potential solution to this problem comes from ensemble methods. In statistics and machine learning, ensemble models are predictive models built by integrating multiple models (called base models) and they often achieve predictive performances better than any of its constituent models [Polikar, 2006]. *Bootstrap aggregating* or *bagging* [Breiman, 1996] is one

such approach that learns a set of base models each from a different bootstrap sample of the original training dataset. To predict a class in bagging, the base model predictions are aggregated either by averaging or majority voting. Another popular ensemble method is *boosting* [Freund et al., 1999] where, base models learned in each subsequent iteration of the algorithm tries to focus more on the examples from the training data that were harder to predict in the previous iteration. The ensemble predictions are then made using a weighted sum of predictions from the base models. Ensemble methods have achieved a lot of success in applications where predictive performance is critical [Seni and Elder, 2010]. Ensemble methods have also been successfully used in bioinformatics including for biomarker discovery tasks [Yang et al., 2010, Günther et al., 2012].

The theoretical Bayes optimal classifier [Mitchell et al., 1997] is an ensemble that makes predictions from a weighted combination of all possible models in the model space. On average, no other model can outperform this classifier using the same model space and prior knowledge. An explanation for its success comes from Bayesian learning theory [Bernardo and Smith, 2009] which posits that a single model ignores the uncertainty associated with the correctness of the model as a result of limited data and noise. Ensembles, on the other hand, combine predictions from several models weighed by their uncertainty of being correct. By doing so, they have a mechanism to account for model uncertainty. In Bayesian methods, Bayesian model averaging [Hoeting et al., 1999] is the standard way to handle uncertainty in model correctness. In BRL, we implemented this approach and found that it indeed helps improve the predictive performance of BRL in biomarker discovery tasks [Balasubramanian et al., 2014].

Domingos showed that simple ensemble methods like bagging easily outperform model averaging [Domingos, 2000]. Minka suggests that model averaging is not model combination [Minka, 2000] because model averaging still works on the premise that only one of the base models is correct. This does not enable us to benefit from the enriched space of models that ensemble methods provide. Unlike model averaging, ensemble methods work on the premise that a combination of models from the model space is the correct model. Monteith et al. [Monteith et al., 2011], further showed that by accounting for the uncertainty in the correctness of the combination of models, using Bayesian model combination, we can further

improve the performance of ensemble methods like bagging.

BRL has been shown to be successful in modeling from high-dimensional datasets and has this useful utility for biomarker discovery tasks of being able to incorporate prior domain knowledge (see section 3.3), which makes it a good candidate to be used as base classifiers for ensemble models. BRL has been shown to consistently perform better than state-of-the-art classifiers including C4.5 decision trees [Quinlan, 2014] on high-dimensional datasets [Gopalakrishnan et al., 2010, Lustgarten et al., 2017]. We would like to see if this advantage also translates into an ensemble methods. Specifically, we want to test if a bagged BRL model achieves a better predictive performance than BRL alone, C4.5, and bagged or boosted C4.5 trees. We would also like to do the same comparison using boosted BRL.

Additionally, we would like to evaluate if there is an added benefit over an ensemble BRL model if we account for the uncertainty in the correctness of model combination of BRL models by using Bayesian model combination [Monteith et al., 2011]. We would also like to compare its predictive performance to our previous work on model averaged BRL model. With respect to the observations by Domingos [Domingos, 2000], Minka [Minka, 2000], and Monteith et al. [Monteith et al., 2011], we expect both bagged (or boosted) BRL and BRL with model combination to perform better than model averaged BRL. While being a misnomer because model averaging isn't model combination (an ensemble method), we will still collectively call the algorithms (bagged BRL, boosted BRL, model combination of BRL classifiers generated from either bagging or boosting, and model averaging of BRL classifiers generated from bagging or boosting) as Ensemble Bayesian Rule Learning (EBRL).

An important advantage of BRL is its interpretability i.e., the model predictions are supported by human-readable explanations in rule forms. These explanations can be very helpful in the clinical validation step of the biomarker development process [Goossens et al., 2015] following the biomarker discovery stage. However, these rule models become less interpretable when combined into an ensemble model. So, we also propose a novel method to visualize an ensemble of BRL called Bayesian Rule Ensemble Visualizing tool (or BREVity) to help interpret an ensemble of BRL models.

To summarize, we implement and evaluate the various methods in EBRL and present a novel visualization method to help interpret these models to assist in the biomarker devel-

opment process.

3.2.2 Ensemble Bayesian Rule Learning (EBRL) algorithms

In this section, I describe the implementation of ensemble methods using BRL to help improve the predictive performance using BRL classifiers. Collectively, we call these methods as Ensemble Bayesian Rule Learning or EBRL. We start by explaining the theoretical concept of Bayes optimal classifier and its attractive properties.

Most model search algorithms like the one shown in Algorithm 1 attempt to find the most probable hypothesis given the training data i.e., $\max_{h \in H} p(h|D)$. The hypothesis, h, here is a BRL model but can be any classifier. The hypothesis space H is the space of all possible BRL classifiers. The ultimate goal of a classifier is to provide accurate class probabilities for a queried test instance. Mitchell et al. [Mitchell et al., 1997] point out that it is possible to do better than to just use the class probabilities from the most probable hypothesis for prediction. The most probable class probabilities, for a test instance, can be obtained by combining the predicted class probabilities from all hypotheses in the hypothesis space $h \in H$, weighed by their posterior probability, $p(h|D)$. Such a classifier is called the *Bayes optimal classifier*. On average, no other classification approach can outperform this method using the same hypothesis space and prior knowledge. The Bayes optimal classifier computes the class probabilities as shown in Equation 3.8.

$$p(y^t|\boldsymbol{x^t}, H, D) = \sum_{h \in H} p(y^t|\boldsymbol{x^t}, h) \cdot p(h|D) \tag{3.8}$$

By using default cut-offs, we can assign a class to the test instance with the class that had the highest class probability. This is shown in Equation 3.9.

$$\hat{y} = \arg\max_{c \in C} p(y^t|\boldsymbol{x^t}, H, D) \tag{3.9}$$

Ideally, to employ the Bayes optimal classifier in Equation 3.8, we should have access to the space of all possible hypotheses in the hypothesis space. The hypothesis space grows exponentially to the number of variables in the dataset. So, it is computationally prohibitive for even problems with a moderate number of variables. In biomedicine, we often deal with

problems with several tens of thousands of variables. In such cases, it is not feasible to compute the entire hypothesis space. Instead, we can sample a diverse set of classifiers that focus on different aspects of the classification problem. *Bootstrap* sampling and *boosting* are two popular methods to do this. These approaches are described in the next subsection.

3.2.2.1 Model generation We implement two approaches to generate models to represent the hypothesis space in Equation 3.8. The first approach is bagging, where different datasets are generated by bootstrap sampling. The other approach is boosting, where iteratively, we focus more on instances that were misclassified in the previous iterations.

Bagging In statistics, bootstrap sampling [Efron, 1992] is a popular data sampling technique. For a dataset with m instances, $D = \{X^i, Y^i\}_{i=1}^m$, a bootstrap sample, D_j, is a dataset with m instances that uniformly samples with replacement, instances from the original dataset D. There are $|H|$ such bootstrap samples generated, $D_{j=1:|H|}$. Each bootstrap sample is expected to have $\approx 63.2\%$ of the unique instances in D, while the rest are duplicates as a result of sampling with replacement [Aslam et al., 2007]. A BRL classifier, h is learned on each of the $|H|$ bootstrap samples and added to the set of H hypotheses.

Boosting AdaBoost [Freund et al., 1999], short for Adaptive Boosting, is a popular machine learning meta-learning method that iteratively tries to focus more on the misclassified instances from the previous iteration of the algorithm. AdaBoost with decision trees is often considered as the best off-the-shelf classifier in machine learning [Kégl, 2013].

AdaBoost was written for binary classifiers. Stagewise Additive Modeling using Multiclass Exponential loss function or SAMME is a powerful extension of AdaBoost meant to handle multi-class problems. The pseudocode for boosting with SAMME using BRL classifiers is shown in Algorithm 2.

The input to the SAMME is the training data $D = \{X^i, Y^i\}_{i=1}^m$ with m instances. We also specify the number of models to generate for the hypothesis space H. The output of the algorithm is a set of models H to combine into an ensemble classifier using Equation 3.8. SAMME also gives a model weight distribution α, which can be substituted into $p(h|D)$ term in Equation 3.8 to make a prediction. Step 1 of the algorithm creates an empty set of H and α. Step 2 initializes instance weights to uniform distribution. Step 3 iterates

Algorithm 2: BOOSTING

Input : Data ($D = \{X_{i=1:n}, Y\}$) containing m instances and $|H|$, the number of boosting models.

Output: A set of models H and associated model weight distribution α.

1 Initialize $H \leftarrow \emptyset$ and $\alpha \leftarrow \emptyset$;
2 Initialize uniform weights, $w^j = \frac{1}{m}$, for each instance $\{X^j_{i=1:n}, Y^i\}$ in D;
3 **foreach** $t = 1$ *to* $|H|$ *models* **do**
4 Learn with weighted instances, $h^t \leftarrow$ `BRL-Search`$(D(w^j))$;
5 Compute error, $\epsilon^t = \sum_{j=1:m} w^t \mathbb{1}_{\{Y^j \neq h^t(X^j)\}} / \sum_{j=1:m} w^j$;
6 Compute model weight, $\alpha^t = \log \frac{1-\epsilon^t}{\epsilon^t} + \log(r-1)$;
7 Add model, $H \leftarrow h^t$;
8 Add associated model weight, $\alpha^t \leftarrow \alpha^t$;
9 Update, $w^j = w^j \cdot exp(\alpha^t \mathbb{1}_{\{Y^j \neq h^t(X^j)\}}); j = 1 : m$;
10 Normalize, w to sum to 1;
11 **return** H and α;

over $|H|$ times to focus more on instances misclassified in the previous iteration. Step 4 learns the BRL model using the greedy best first search in Algorithm 1. To enable BRL to calculate from weighted instances, the N_{jk} and N_j terms in the Bayesian score shown in Equation 3.5 are modified to take in sum of weights of instances instead of counts. For example, if two instances have the configuration j and belong to class k, $N_{jk} = 2$. Say, from boosting, their weights of the two instances were 0.02 and 0.01, then the sum $N_{jk} = 0.03$. Step 5 calculates the total error made by BRL, by summing the normalized weights of all the instances misclassified by the BRL classifier, learned in this iteration, when used to classify each training instance. Step 6 computes the model weight, α, with the log odds of the error and an additive weight where, r is the number of classes in Y. The models from the iteration and the learned weight is added to H and α, respectively. The weights of instances are updated for the next iteration in steps 9 and 10 to focus on misclassified instances and is weighted by the α weight.

The next step in designing the ensemble model is to aggregate the predictions from classifiers in H for making a prediction using the ensemble model.

3.2.2.2 Model aggregation We will describe three different approaches to combining the BRL classifiers in H approximating the same general equation for the Bayes optimal classifier in Equation 3.8. The first approach is linear combination of model weights (Bagging-

BRL-LC and Boosted-BRL-LC). It is a special case of model averaging where all models weights are uniform in bagging or weighted by the model weights detemined by the SAMME algorithm in the previous section. The second approach of model combination is the more general case of weighting each predicted classifier probabilities with the likelihood of the data having been generated by the model. This is called Bayesian model averaging (Bagging-BRL-BMA and Boosted-BRL-BMA). The final approach we use to aggregate the models is Bayesian model combination (Bagging-BRL-BMC and Boosted-BRL-BMC), which accounts for the uncertainty in the correctness of model combination. Here, we sample different model weight distribution to aggregate the ensemble over and weigh each ensemble with a posterior probability of the correctness of that ensemble.

Linear combination (Bagging-BRL-LC and Boosted-BRL-LC): This is the classic model combination strategy used in bagging and boosting. We approximate the Bayes optimal classifier in Equation 3.8 by obtaining the predicted class probabilities from each of the base BRL classifiers in H and weigh its prediction with a weight $p(h|D)$. In bagging this weight is uniform $p(h|D) = \frac{1}{|H|}$. Note that the sum of $\sum_{h \in H} p(h|D) = 1$. This EBRL classifier places equal importance on each of the models learned from the bootstrap samples, for the prediction task. In boosting, we use the α weight given by the SAMME algorithm for each model in H. This weight is also normalized to sum to 1. In short, Bagged-BRL-LC and Boosted-BRL-LC methods refer to classic bagging and boosting using BRL classifiers as base models.

Bayesian model averaging (Bagged-BRL-BMA and Boosted-BRL-BMA): A more general case of model aggregation is Bayesian model averaging (BMA). BMA tries to integrate out the uncertainty about which of the models in the ensemble is correct. This is similar to the model we implemented and evaluated in our previous work in [Balasubramanian et al., 2014], with some important changes. The model generation step is different. Here, we generate the different models learned on bootstrap samples. Here, again, we do not average over the space of all possible models, which is computationally unfeasible. So, this approach is also selective model averaging, to be precise, as it was in our previous work.

Unlike linear combination approach that weighs all classifiers equally, the Bayesian model averaging weighs them with the likelihood that the original training data was generated by

that particular classifier. This is shown in Equation 3.10.

$$p(h|D) = \frac{p(D|h) \cdot p(h)}{\sum_{h \in H} p(D|h) \cdot p(h)} \tag{3.10}$$

For EBRL, the term $p(D|h) \cdot p(h)$ is directly obtained from the Bayesian score in Equation 3.5. The denominator term normalizes the Bayesian score for only the models in H. This way, the sum of $\sum_{h \in H} p(h|D) = 1$. The only distinction between Bagged-BRL-BMA and Boosted-BRL-BMA is that the hypothesis space is generated using bagging and boosting, respectively.

Bayesian model combination (Bagged-BRL-BMC and Boosted-BRL-BMC): Monteith et al., [Monteith et al., 2011] show how to modify BMA to Bayesian model combination (BMC). Following the implementation in that paper, we implement the BMC approach for BRL. To modify BMA to BMC, Equation 3.8 is modified to Equation 3.11.

$$p(y^t|\boldsymbol{x^t}, H, E, D) = \sum_{e \in E} p(y^t|\boldsymbol{x^t}, H, e, D) \cdot p(e|D) \cdot p(e) \tag{3.11}$$

Here, e represents a model combination strategy i.e., a specific distribution of model weights $p(h|e, D)$ to be used in Equation 3.8 to obtain a prediction from the Bayes optimal classifier. Given the model weights specified by e, we can now perform the ensemble prediction using Equation 3.8— $p(y^t|\boldsymbol{x^t}, H, e, D) = \sum_{h \in H} p(y^t|\boldsymbol{x^t}, h, D) \cdot p(h|e, D)$. However, to implement BMC, we need a way to generate a set of possible model weight distribution, E, and also to compute the posterior probability of that ensemble, $p(e|D)$, and finally the prior probability of the ensemble, $p(e)$. We first describe how to compute the terms $p(e|D)$ and $p(e)$ before explaining the method we used to generate the set of model weight distributions.

We compute the posterior of the ensemble combination method using Bayes theorem in Equation 3.12 that assumes all instances in the training dataset, $x^i; i = 1, \cdots, m$, are independent.

$$p(e|D) = \frac{p(e)}{p(D)} \cdot \prod_{i=1}^{m} p(y^i|e) \tag{3.12}$$

Here, in the Bayes theorem expression, the value of the likelihood is set to $p(D|e) = \prod_{i=1}^{m} p(y^i|e)$. Assuming uniform class noise model [Domingos, 1997], which posits that each example in the training dataset is corrupted with a probability ϵ. Then the probability of

making a correct prediction is $1-\epsilon$; and the probability of making an incorrect prediction is ϵ. If there are r correct predictions made over m instances by the ensemble, Equation 3.12 shows how to compute the posterior of the ensemble.

$$p(e|D) \propto p(e) \cdot (1-\epsilon)^r(\epsilon)^{m-r} \tag{3.13}$$

One consequence of using Equation 3.13 to compute $p(e|D)$ is that it prefers model weight distribution that leads to predictions with higher accuracies. If the goal was instead to optimize on some other metric, we should alter the computation of $p(e|D)$ by using the other metric.

We set the ensemble prior to follow a uniform distribution $p(e) = \frac{1}{|E|}$. Here, $|E|$ is the total number of ensembles we generate for Equation 3.11. Finally, we normalize to ensure that $\sum_{e \in E} p(e|D) \cdot p(e) = 1$.

Finally, we show how we generate a set of possible model weight distributions. Even for a moderate number of base models in H, we would need to sample over a very large number of possible model weight distributions. So, we need an informative way of sampling over these model weights. We do this using the Dirichlet distribution as shown in [Monteith et al., 2011]. We initialize the Dirichlet distribution for $|H|$ categories and set the initial hyperparameters all to 1.0, i.e., $\alpha_{i=1:|H|} = 1.0$. Please distinguish this α from the one used for the Bayesian score. This generates a normalized weight, one for each model in the ensemble. Then, we sample U times using the Dirichlet distribution with the same α hyperparameter. We refer to this value U as *number of Dirichlet samples*. We evaluate each model weight distribution using the ensemble posterior probability in Equation 3.12. The weight distribution that achieved the highest posterior probability is summed to the current values of the Dirichlet hyperparameters, and the Dirichlet distribution with updated parameters is used to sample the model weights distribution in the next iteration. This iteration is repeated $|E|$ times, to generate a set of $|E|$ model weight distribution. We now have everything we need to compute Equation 3.11.

While BMA tries to integrate out the uncertainty about which of the models in the ensemble is correct, BMC integrates out the uncertainty about which of the ensemble combination methods is correct. As Minka [Minka, 2000] pointed out, such a combination no

longer assumes that only one of the BRL model is correct, instead assumes that only one of the ensemble combination of BRL models is correct.

Again, the only distinction between Bagged-BRL-BMC and Boosted-BRL-BMC is that the hypothesis space is generated using bagging and boosting, respectively.

3.2.3 Variable importance

In biomarker discovery, it is helpful to have a ranked list of the most important biomarkers according to the ensemble. For Bagged-BRL-LC, Boosted-BRL-LC, Bagged-BRL-BMA, and Boosted-BRL-BMA, we compute variable importance for X_i in the ensemble of H models using Equation 3.14.

$$p(X_i|H, D) = \sum_{h \in H} \mathbb{1}_{X_i \in h} p(h|D) \tag{3.14}$$

This is nothing but the sum of the weights of all BRL models in the ensemble that selected the variable X_i. This is the same approach used by Yeung et al. [Yeung et al., 2005], to evaluate gene relevance using their approach to Bayesian model averaging of logistic regression models. Since we ensured that $\sum_{h \in H} p(h|D) = 1$, the variable importance $p(X_i|H, D)$ can simply be interpreted as the percent of weight carried by models containing variable X_i.

To compute the variable importance using Bagged-BRL-BMC and Boosted-BRL-BMC, we use Equation 3.15.

$$p(X_i|H, E, D) = \sum_{h \in H} \sum_{e \in E} \mathbb{1}_{X_i \in h} p(h|e, D) \cdot p(e|D) \cdot p(e) \tag{3.15}$$

Again, we had normalized the ensemble weights to ensure that $\sum_{e \in E} p(e|D) \cdot p(e) = 1$. So, the variable importance $p(X_i|H, E, D)$ can be interpreted as the percent of the ensemble-weighted average of model-weight carried by models containing variable X_i.

3.2.4 Bayesian Rule Ensemble Visualizing tool (BREVity)

An advantage of using BRL for biomarker discovery tasks was its interpretability. BRL offered IF-THEN rules explanations for each prediction it made. This could particularly prove useful for biomarker validation step of the biomarker development process. Ensemble

predictions made by EBRL are less interpretable. We develop a novel method to visualize an ensemble of BRL models and we call this tool— Bayesian Rule Ensemble Visualizing tool or BREVity. BREVity is a tree-graph, where paths from root to nodes in the tree represent rule patterns in EBRL. The edge weights in BREVity is the relative importance of the rule patterns in EBRL. We will clarify BREVity with the help of Figure 8.

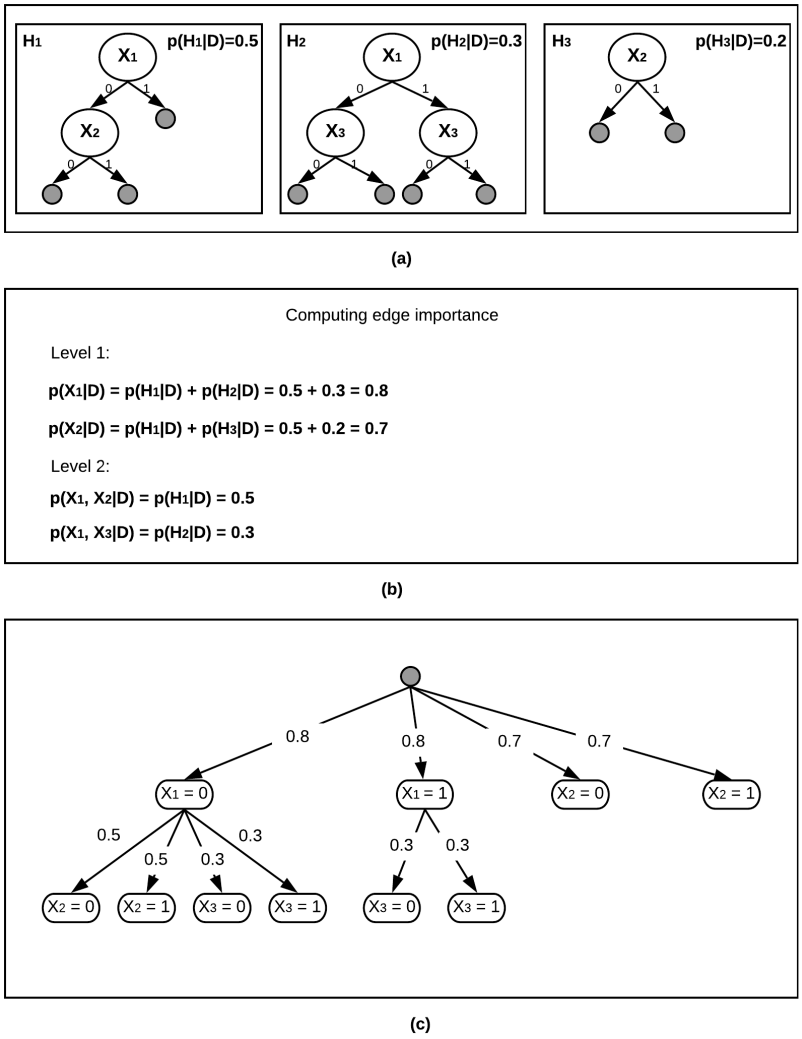


Figure 8: Bayesian Rule Ensemble Visualizing tool (BREVity)— (a) A set of 3 models in EBRL with their posterior probabilities in the top-right corners; (b) edge weights using set variable importance computed for all variable combinations in EBRL; and (c) the BREVity graph of EBRL.

Assume, we learned either of the three types of EBRL model using 3 bootstrap samples.

After running BRL greedy best first search on each of the three bootstrap samples, we obtain 3 models— $\{H_1, H_2, H_3\}$. The learned models and the posterior probabilities of each model is shown in Figure 8a. Note that the posterior probabilities sum to 1 since they were normalized in each implementation of EBRL. Also, if the EBRL model being used is Bagged-BRL-BMC, $p(H|D) = \sum_{e \in E} p(H, e|D) \cdot p(e|D) \cdot p(e)$.

Next, we compute the importance of each variable in EBRL just as we did in section 3.2.3. We extend the equations in section 3.2.3 to compute the variable importance for a set of variables in EBRL. For Bagged-BRL-LC and Bagged-BRL-BMA, we extend Equation 3.14 for a set of variables with Equation 3.16.

$$p(\boldsymbol{X}|H, D) = \sum_{h \in H} \mathbb{1}_{X_i \in \boldsymbol{X}, X_i \in h}\ p(h|D)$$
$$\text{where, } \boldsymbol{X} \subset X_{1:n} \tag{3.16}$$

Similarly, we extend Equation 3.15 for a set of variables with Equation 3.16.

$$p(\boldsymbol{X}|H, E, D) = \sum_{h \in H} \sum_{e \in E} \mathbb{1}_{X_i \in \boldsymbol{X}, X_i \in h} p(h|e, D) \cdot p(e|D) \cdot p(e)$$
$$\text{where, } \boldsymbol{X} \subset X_{1:n} \tag{3.17}$$

With these new extended equations to compute the variable importance to a set of variables importance, we can now compute a pattern importance. A pattern importance is the variable set importance for all variables in the pattern. Simply put, pattern importance is the sum of the normalized model weights of each model that contains the variable. In Figure 8, there are cumulatively 4 variable set patterns in all the models in EBRL— 1) only X_1, 2) only X_2, 3) both X_1 and X_2, and 4) both X_1 and X_2. We compute the pattern importance for these sets using Equations 3.16 and 3.17. This computation is shown in Figure 8b.

Finally, we construct the BREVity tree visualization as shown in Figure 8c. Each path from the root to a node can represent a possible rule in EBRL. For example, the rule pattern from model H_1, 'IF $(X_1 = 0)$ AND $(X_2 = 0)$' is the left-most path in the BREVity tree. The variable importance of X_1 was computed to be 0.8. The pattern importance of set $\{X_1, X_2\}$,

was computed to be 0.5. So, the edge weight of the BREVity tree from root to X_1 is 0.8, and the weight of edge to root to X_2, while crossing X_1, is 0.5.

This visualization has a useful utility in having the left-most part of the tree in each level of the tree to be the most important pattern in EBRL. For example, only look until depth 1 of the tree. The left-most pattern is X_1, which is also the most important pattern of length 1 in EBRL. given that we chose to look at variables associated with X_1, at depth two, we notice that pattern $\{X_1, X_2\}$ has higher importance than pattern $\{X_1, X_3\}$. In the real-world application with many trees, this tree is likely to be very dense, breadth-wise. Such a pattern importance sorting helps the user focus only on the most important patterns in EBRL. So, the left-most patterns are the best candidates for validation according to EBRL.

We implemented BREVity as a javascript web application that is deployed within an open-source Eclipse Jetty Web server/servlet container. The BREVity tree is constructed as a Java object by EBRL. This Java object is translated into a JSON formatted file. Given a JSON input file, the javascript code parses the file and creates a hierarchical tree structure that is passed into a d3 visualization tree layout and displayed in a web browser. It includes the ability for the user to expand nodes from root to leaf, and to filter tree nodes based on tree edge weights.

A useful function of BREVity visualization is that the weights in the edges of the tree are simply the strength of influence of the pattern in the EBRL prediction. By setting a filter, to say, $\theta > 0.5$, we only focus on patterns that contributed at least 50% to the total EBRL prediction.

While this collection of patterns is more complex to interpret than the patterns in a single BRL rule model, BREVity helps the user focus on interpretable patterns of the most influence for prediction made by the EBRL model. In multifactorial diseases, many variables must interact in different ways with other variables. This visualization offers us to have an interpretable access to the understanding of a complex model that is very adept in explaining multifactorial disease processes.

3.3 BAYESIAN RULE LEARNING WITH INFORMATIVE PRIORS ($\mathbf{BRL_P}$)

In this section, I describe the implementation of BRL_p, an extension to the BRL algorithm to enable it to incorporate prior domain knowledge. This is a published work [Balasubramanian and Gopalakrishnan, 2018], described here in detail to show how it contributes to solving the KDD problem 2.4. We saw in the background section (see section 2.2), that there are plenty of sources of knowledge in biomedicine available in the various bioinformatics resources including the literature. When presented with such a challenging problem of high-dimensionality, it may help us reduce the chances of making false positive predictions, if we focused more on promising regions of the model space. In subsection 3.3.1, I motivate the problem further and in section 3.3.2, I show an implementation of the BRL_p algorithm.

3.3.1 Background and motivation

Omic datasets are high-dimensional. The large numbers of candidate variables generate a model search space that is very large for data mining algorithms to explore efficiently, and having only a few instances generates uncertainty for the algorithm to determine the correctness of any candidate model. In such model search spaces, data mining algorithms can easily get stuck in local optima or they may infer associations between spurious variables and the outcome variable, by chance (false positive).

Fayyad et al. [Fayyad et al., 1996b], emphasized the importance of domain prior knowledge in all steps of the Knowledge Discovery in Databases (KDD) process. In biomedicine, often in addition to the training dataset, we have some prior domain knowledge. This domain knowledge can help guide the data mining algorithm to focus on regions in the model search space that are either objectively more promising for a given problem or subjectively more interesting to a user. The prior knowledge can come from domain literature (e.g. searching through PubMed), a domain expert (e.g. a physician), domain knowledge-bases (e.g. Gene Ontology) or from other related datasets (e.g. from public data repositories like Gene Expression Omnibus). It is imperative to develop data mining methods that can leverage

domain knowledge to assist with the data mining process.

BRL takes a dataset as input and searches over a space of Bayesian belief-networks (BN) to identify the BN that best explains the input dataset. BRL then infers a rule model from this BN. BRL uses the Bayesian score as a heuristic to evaluate a BN during search. The score allows the user to specify a prior belief distribution over the space of BNs that encodes our prior beliefs about which models are more likely to be correct than others with respect to our domain knowledge. Typically, in literature and in the BRL methods discussed so far, uninformative priors are used, which means that we claim that *a priori* all models are equally likely to be correct. As said earlier, often along with the dataset, additional domain knowledge is available that can assist with the data mining process. These sources lead us to believe that some models are more likely to be correct than others even before we see the dataset. We can specify this belief using informative priors. Two approaches to using informative priors in literature have shown promise [Castelo and Siebes, 2000, Mukherjee and Speed, 2008]. In the next subsection, we discuss each of the two approaches and describe ways to extend BRL to specify such informative priors that can incorporate domain knowledge.

3.3.2 BRL$_\mathbf{p}$ algorithm

The Bayesian score, the heuristic score used by BRL from Equation 3.5 is reproduced below for easy reference—

$$P(B_S, D; \alpha, \kappa) = \kappa^{|\Pi_i|} \cdot \prod_{j=1}^{q_Y} \frac{\Gamma(\frac{\alpha}{q_Y})}{\Gamma(N_j + \frac{\alpha}{q_Y})} \prod_{k=1}^{r_Y} \frac{\Gamma(N_{jk} + \frac{\alpha}{r_Y q_Y})}{\Gamma(\frac{\alpha}{r_Y q_Y})}$$

Here, the structure prior term is $p(B_S) = \kappa^{|\Pi_i|}$, which helps us prefer models with fewer variables. This structure prior represents the prior distribution over all network structures. Here, we can specify our prior bias of certain network structure over others to skew the BRL search to focus on certain network structures more than others. Typically, in literature uninformative priors are used, i.e., they set $p(B_S = 1)$. This means that *a priori* we claim that we do not have any preference of network structures over the others. BRL in this case lets the data alone decide the final learned model. The challenge of specifying these priors is that the total number of network structures grows super-exponentially with the number

of variables n [Harary and Palmer, 2014]. It often becomes unfeasible to specify structure priors for each of these network structures for even moderately sized datasets.

Castelo and Siebes [Castelo and Siebes, 2000] describe a promising approach to elicit structure priors by specifying the probability of the presence or absence of each edge in the network structure. The user only needs to specify the probability of a subset of edges in the network structure. The probabilities for all the remaining edges are assigned a discrete uniform distribution value. A challenge using this approach is to specify the values of these probabilities. In our experiments with BRL using these priors, we observed that the likelihood term in Equation 3.5 always dominates the structure prior term. It would help us if we could control the influence of structure priors over the likelihood term using a scaling factor. As we described earlier in the introduction section, the background knowledge, we specify, itself has uncertainty associated with it. A scaling factor would help us control the influence of data and our prior knowledge.

Mukherjee and Speed [Mukherjee and Speed, 2008] propose an informative prior that uses a log-linear combination of weighted real-valued function of the network structure, $f_i(B_S)$. This function is called the *concordance function*. It can be any function that monotonically increases with the increase in agreement between the learned network structure and the prior beliefs of the user. This is shown in Equation 3.18.

$$p(B_S) \propto exp\left[\lambda \cdot \left(\sum_i w_i \cdot f_i(B_S)\right)\right] \tag{3.18}$$

The hyperparameter w_i are the positive weights that represent the relative importance of each function. The hyperparameter λ is a scaling factor that helps to control the overall influence of the structure prior. This will help us quantify the uncertainty in the validity of our prior knowledge.

The structure prior we used for BRL_p comes from an instantiation of the general form of this prior, shown in Equation 3.18, as described by Mukherjee and Speed [Mukherjee and Speed, 2008]. It allows the user to specify their prior beliefs about the presence and absence

of the edges in the network structure. This instantiation is shown in Equation 3.19.

$$p(B_S) \propto exp\left[\lambda \cdot \Big(|E(B_S \cap E_+)| - |E(B_S \cap E_-)|\Big)\right] \tag{3.19}$$

Here, set E_+ (positive edge-set) represents the set of edges the user believes should be present in the model, and set E_- (negative edge set) represents the set of edges the user believes should be absent from the model. So, the concordance function in this instantiation simply gives a positive count for if the candidate graph contains an edge from the positive edge-set, and a negative count (penalty) when it contains an edge from the negative edge-set. In this instantiation, the weights hyperparameter is set to 1, since our counts are all valued 1. We need to learn the value of the hyperparameter λ. The range of values it can take depends upon the well-known Jeffreys scale [Jeffreys, 1998]. When $\lambda = 0$, the whole exponent becomes 0, and $p(B_S) = exp(0) = 1$, which is the uninformative prior. In other words, when $\lambda = 0$, BRL_p should have no effect of structure prior and so would behave the same as the baseline model, BRL. As we increase the value of λ, the effect of the structure prior would have an increased influence over the likelihood term in Equation 3.5.

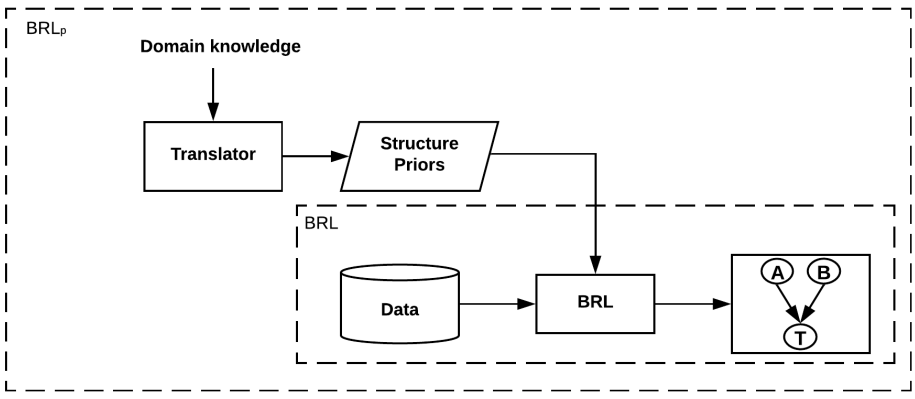


Figure 9: BRL_p framework.

To summarize, BRL_p uses a heuristic score called the BDeu score, shown in Equation 3.5, and encodes the structure prior in that score using Equation 3.19. The BRL_p framework is shown in Figure 9. The inner dotted box, labeled BRL, is the classic BRL without prior knowledge, which takes in an input dataset, uses BRL algorithm to learn and output a model.

The outer dotted box is our extension, BRL_p that can incorporate domain knowledge. The translator process, currently done manually, converts knowledge from various sources to input into Equation 3.19.

3.4 BAYESIAN RULE LEARNING FOR KNOWLEDGE DISCOVERY (BRL-KD)

In this section, I describe the method I developed to not only seem models that are statistically significant, but also clinically relevant. This work is currently available in bioRxiv, an open access preprint repository, here [Balasubramanian et al., 2019].

One important conflict to resolve before designing a solution to this problem was to determine— is finding novel/actionable biomarkers a statistical problem or a decision theoretic problem? If it is a statistical problem, then we must learn a model that is capable of picking biomarkers that are novel/actionable. We can do so by either making changes to learning representations, scoring functions (loss functions), or optimization methods. However, if it was a decision theoretic problem, then the task of statistics would just be to provide the most accurate model possible to explain the observed dataset. Then the decision theoretic approach would help us pick biomarkers that were novel/actionable. My choice boiled down to one important observation of reality summarized succinctly by George E. P. Box— *"All models are wrong, but some are useful"* [Box, 1976]. The choice of rule learning for deploying clinical decision support was one such choice. Perhaps neural networks were more correct in the scenario but rule models were more useful. Similarly, I argue that there are applications where certain models, perhaps sub-optimal, may be more useful. In this case, they are useful because they are more novel/actionable. For example, if the application is developing models for medical screening. Models selecting cheaper biomarkers are more useful, within a constraint of reasonable model performance, because they are easier to produce for mass screening. Similarly, if the application is research, models picking biomarkers that are more novel are more useful. The definition of clinical relevance depends upon the eventual application.

As a result, I developed BRL-KD, a statistical approach to discover clinically relevant models. In subsection 3.4.1, I provide the background and motivation for BRL-KD. Subsection 3.4.2 describes the BRL-KD algorithm.

3.4.1 Background and motivation

[Selleck et al., 2017] emphasize that it is not enough to identify variants but to identify *actionable* variants that has the potential to revolutionize healthcare. [Burke, 2016] reported in 2016, that there were 768,000 papers indexed in PubMed about biomarkers. Yet, despite all the technological advances in omics research and bioinformatics methods, we are still very far from widespread clinical use of these omic biomarkers. Currently, there are only a few dozen clinically relevant cancer biomarkers. There is also a general lack of support from clinical practice guidelines. The European Society of Medical Oncology (ESMO) clinical practice guidelines for lung, breast, colon, and prostate cancers give only a weak recommendation for the use of about 20 omic biomarkers.

To improve the translation of biomarker discovery projects into clinical practice, we must incorporate clinical relevance into the model development process. Clinically relevant biomarkers are not only statistically significant but account to many practical aspects of biomarker including its specificity to the condition, efficacy, cost-effectiveness, and non-invasiveness. For example, specificity of the biomarkers is an important aspect of clinical relevance. Some biomarkers may serve as a test for many pathological conditions and are not specific to the disease it the being developed as a test for. For example, cancer antigen 19-9 (CA19-9) is a biomarker for pancreatic cancer with high statistical signficance [Steinberg, 1990]. However, this antigen is elevated in many other pathological conditions, such as biliary obstruction, that often co-exists with pancreatic cancer.

For example, there are two prominent ways to measure blood glucose— 1) finger stick tests and venipuncture. Both are known to have similar accuracy in quantifying blood glucose. However, finger stick tests are cheaper and less invasive than venipuncture. So, in this scenario where we want to measure a patient's blood glucose alone, finger stick method is clinically more relevant.

Various bioinformatics resources can help us quantify these subjective metrics of clinical relevance. For example, efficacy can be quantified by PharmGKB [Hewett et al., 2002] drug entries for each molecule. To quantify specificity, we can look at gene ontology [Ashburner et al., 2000] that include known associated diseases. If a gene is known to be associated with many diseases, the gene is not specific enough and is a poor candidate to be a biomarker for a disease.

One way to solve this is using a classical constrained optimization search. We specify a utility function to determine the clinical relevance of the model. Say, our goal is to develop a model that is cost-effective, then the utility function can be a simple sum of the cost of including all the biomarkers (present in the model) in clinical practice. We assume that we have some pre-specified cost constraint, exceeding which, we lost the clinical relevance by being more expensive than the current methods in practice. In a classical constrained optimization search using BRL, while we search for BRL models that optimize the Bayesian score, we can add a search constraint, such that, if the sum of the cost of the biomarkers in the current model being specialized exceeds the user-defined cost constraint, then we simply do not accept the model. Using the classical constrained search, we can find a model from the search, which is the most optimal model seen during the search that did not exceed the user-specified cost constraint. This approach is not ideal for two reasons— 1) the limitations of greedy search, and 2) the uncertainty in the specification of the utility function.

In the greedy search such as the one we developed in section 3.1, the search may pick a marker that is very expensive in the first step because of it being the most optimal single marker. However, the cost constraint may not allow us to add any more markers to the model. There may exist a model without this expensive first marker, that combines to give a better predictive performance with a combined cost less than the constraint. We will not be able to find such a model using greedy search and the classical constrained optimization procedure.

There is uncertainty associated with the specification of the utility function. For example, in practice specifying the cost-constraint is difficult. The cost depends upon many factors including medicare and the type of insurance the individual is on. Perhaps the patient being evaluated cannot afford the cost constraint set during the classical constrained search. Or

perhaps, the patient is found to be at high-risk of developing the disease and we need to use the model to help get a more precise evaluation in exchange of an increased cost. In such scenarios, the user would need to have run BRL specifying a wide range of cost-constraints to accommodate the different scenarios in the point of care. It is also hard to quantify such a constraint for utility functions like novelty or efficacy, which can be hard to interpret.

In these above two scenarios, classical constrained optimization search would not suffice. Instead, Bayesian methods offer an elegant way to specify this useful information of the uncertainty of our own knowledge about the utility function. In this study, we formulated the problem of discovering clinically relevant knowledge as a knowledge discovery problem. We extended the heuristic score used by BRL to include clinical relevance to help search for clinically more relevant models.

3.4.2 BRL-KD algorithm

We formulate the problem of identifying clinically relevant models as a knowledge discovery problem. Knowledge Discovery in Databases (KDD) is an important process of discovering useful knowledge from data. KDD is the non-trivial process of "identifying valid, novel, potentially useful, and ultimately understandable patterns in the data"[Fayyad et al., 1996b]. According to this definition, we want BRL to discover knowledge. The current BRL algorithm is designed to mine *valid* and *understandable* patterns from the data. The rule patterns generated by BRL are human-readable and such IF-THEN statements are easy to understand. The BRL search finds valid patterns by trying to optimize its heuristic score. In this work, we modified the heuristic score to help BRL search for *valid*, *novel*, *potentially useful*, and *understandable* patterns from the data. We define *novel* and *potentially useful* patterns together as clinically relevant patterns. Based on the clinical application, novel patterns are desired, for example, in biomedical research to help improve our understanding of a physiological process. In such a case, *novel* patterns are useful. But if the clinical application is to develop medical screening methods, many factors— novelty, cost-efficiency, biomarker specificity, efficacy, and non-invasiveness— together define *useful* patterns. So, in this study, we combine the terms *novel* and *potentially useful* from the KDD definition and

together call them *useful* patterns.

The heuristic score used by BRL search algorithm (from Equation 3.5) is shown below.

$$P(B_S, D; \alpha, \kappa) = \kappa^{|\Pi_i|} \cdot \prod_{j=1}^{q_Y} \frac{\Gamma(\frac{\alpha}{q_Y})}{\Gamma(N_j + \frac{\alpha}{q_Y})} \prod_{k=1}^{r_Y} \frac{\Gamma(N_{jk} + \frac{\alpha}{r_Y q_Y})}{\Gamma(\frac{\alpha}{r_Y q_Y})}$$

Here, the structure prior term is $p(B_S) = \kappa^{|\Pi_i|}$, is the prior distribution that represents our belief in which of the models in the hypothesis space is likely to be correct *a priori*. This structure prior represents the prior distribution over all BN structures. We encode it to $\kappa^{|\Pi_i|}$ to prefer BNs with fewer variables. In BRL_p, we used the structure prior term, $p(B_S)$, to incorporate prior domain knowledge to create a bias in the search to prefer network sub-structures that have been shown to be promising from previous works. The rest of the term in the equation is called the likelihood function that encodes the likelihood that the observed training data was generated by a given BN model. The likelihood function gives us a measure of how well the model fits the data. Generally speaking, the better the model fits the data, the more likely it is to generalize to unseen test data. The prior term encodes the prior probability of which of the BNs is the correct data-generating model. Together, the likelihood function and the prior term assist the search algorithm in identifying promising candidate BNs that are most likely to have generated the training dataset. In KDD definition, these two terms help the search algorithm find *valid* patterns from data.

To include assistance to find *useful* patterns, we modify the heuristic score to Equation 3.20.

$$P(B_S, \Psi, D; \kappa, \alpha) = p(B_S) \cdot p(\Psi|B_S) \cdot \prod_{j=1}^{q_Y} \frac{\Gamma(\frac{\alpha}{q_Y})}{\Gamma(N_j + \frac{\alpha}{q_Y})} \prod_{k=1}^{r_Y} \frac{\Gamma(N_{jk} + \frac{\alpha}{r_Y q_Y})}{\Gamma(\frac{\alpha}{r_Y q_Y})} \tag{3.20}$$

This equation encodes the joint probability of the BN structure (B_S), the data (D), and the clinical relevance as encoded by the utility function Ψ. The utility function $p(\Psi|B_S)$ takes as input the BN structure, and outputs the clinical relevance of the model. The term $p(\Psi|B_S)$ encodes a probability distribution that represents our belief about which of the models in the hypothesis space is more clinically relevant.

We encode the utility function similar to how we encoded the prior distribution over models in BRL_p. We use the informative prior as specified by [Mukherjee and Speed, 2008].

The utility function is a log-linear combination of weighted real-valued function (concordance function) of the network structure, B_S. The utility function monotonically increases with the increase in clinical relevance of the model. The utility function is shown in Equation 3.21.

$$p(\Psi|B_S;\lambda,w) \propto exp\left[\lambda \cdot \left(\sum_{t\in T} w_t \cdot f_t(B_S)\right)\right] \tag{3.21}$$

As an example, if we want more cost-effective models, the utility function simply prefers cheaper models. One way to compute that is the sum of costs associated with each biomarker in the model. To enable BRL-KD to look for cost-effective models, the set T in Equation 3.21 iterates through each biomarker in the dataset. Weight w_t is the weight of the biomarker t. For cost-effectiveness we set the weight to the cost of the biomarker. The concordance function here can be an indicator function that returns 1, when the variable t exists in the BN structure, B_S. Of course, we maximize the heuristic score in BRL search. However, we have encoded the current utility function as a minimization problem. To convert this into a maximization problem, we simply encode the negative value of cost, instead of the cost. Now, the utility function needs to be maximized in order to minimize the overall cost. Another important consideration is the value of the weights. By encoding the cost, some markers may cost a few US dollars. Some other may cost thousands of dollars. This will lead to this term being either too large or too small. To avoid this, we can perform min-max scaling for the values to range between 0 and 1.

Similarly, we can encode marker specificity by looking at gene-disease ontologies and encoding the weights with the reciprocal of the number of known diseases associated with the gene. By maximizing such a function, BRL-KD would prefer biomarkers with fewer known disease associations. Similar approaches can be done to encode efficacy, invasiveness, or a combination of multiple utilities.

In Equation 3.21, $\lambda = 0$ implies no confidence in the specified values of each biomarker. $\lambda \geq 25$ asks the search algorithm to prioritize cost before looking at the likelihood function. The user may search over a range of values for λ in between. The set of models generated by varying λ generate a pareto set of solutions, all accommodating our constraints while incorporating our specified knowledge at varying degrees. We can leave the decision of which model to use based on the circumstances at the point-of-care. One way to cut-down

on the number of models to examine further is by specifying a constraint like the required AUROC (or cost per life years gained) achieved by each λ over 10-fold cross-validation.

Now, as before with BRL, BRL-KD maximizes the heuristic function in Equation 3.20 to identify patterns that are *valid*, *novel*, *potentially useful*, and *understandable*, from the data.

It is easy to see how BRL-KD can be extended to EBRL methods. Consider the outcome for one individual being predicted by EBRL. A set of rules would fire, aggregated in a specific way depending upon the EBRL method used. For that individual, a set of biomarkers are found to be relevant to help predict their outcome. Using BRL-KD specified priors over each base model, for the same individual, the original set of biomarkers would ideally be replaced by clinically more relevant biomarkers.

4.0 EXPERIMENTS AND RESULTS

In this chapter, the methods developed in this dissertation are evaluated and compared to other state-of-the-art methods commonly used in data mining. To evaluate each developed method, an experiment is designed, results are presented and discussed. The goal of this dissertation was to develop algorithms that learn classifiers, which are actionable in the field of biomedicine. As discussed in the Significance and Background section, one of the major challenges in biomedical data analysis comes from high-dimensional datasets generated from high-throughput technologies. An example of such a dataset is gene expression data. Section 4.1 introduces the problem and importance of finding differentially expressed genes from gene expression data. Section 4.2 outlines the general experimental design used to evaluate the performances, of the developed methods, on publicly available gene expression datasets. This section includes the description of— gene-expression data collection from public data repositories, data pre-processing, metrics used to evaluate the performance of each classifier, and finally the statistical methods used to establish significance of the results from the analysis. Sections 4.3, 4.4, 4.5, and 4.6 show the experiment design, results, and discussions after evaluating BRL (see 3.1), EBRL (see 3.2), BRL_p (see 3.3), and BRL-KD (see 3.4), respectively.

4.1 PROBLEM DESCRIPTION: DISCOVERING DIFFERENTIALLY EXPRESSED GENES

Broadly speaking, the DNA present inside cells in the human body contain genes that encode proteins. The proteins, in turn, helps the cell perform biological functions necessary for life.

Gene expression is a multi-step process of the genes synthesizing proteins. The first step, called *transcription*, involves the information in the DNA being passed into mRNA molecules. The second step, called *translation*, involves the information from mRNA molecules being passed into proteins. Quantitatively measuring the mRNA levels is cheaper than measuring proteins. The abundance of the different mRNAs can give us a good estimate of the abundance of related proteins. Each cell can have a different composition of mRNA levels based on its state. A cell state can be described by its tissues of origin, the stage of cell development cycle, environmental response, and disease states. The functional differences between states can be measured by the difference in the composition of the cell mRNA levels called *transcriptome profile*. Difference in the transcriptome profiles can also provide molecular fingerprints of normal and aberrant (disease state) tissue behavior. Each of these mRNAs can be mapped to genes that they are transcribed from. The differentially expressed mRNAs therefore correspond to differential expression of genes. These differentially expressed genes (DEGs) between the cell states are useful to not only understand the function of genes but also to understand the overall mechanism of a biological process. As a result, analyzing gene expression data and identifying DEGs is an important knowledge source for functional genomics, molecular medicine, and pharmacogenomics. This knowledge improves our understanding of the disease, suggests potential therapeutic target, and provides venues for personalized treatments.

The mRNA levels are usually quantified either using high-throughput technologies like RNA sequencing or hybridization microarrays. These are called platforms. RNA sequencing (or RNA-seq) is currently the more popular approach. It uses next generation sequencing (NGS) to quantify the amount of each type of transcripts present in a sample at any given time. Hybridization microarrays, on the other hand, is a microarray chip composed of a glass slide with an array containing single-stranded DNA molecules, called probes, embedded in fixed locations. The probe molecule may be RNA, cDNA or oligonucleotides depending upon the microarray platform technology being used. There are tens of thousands such probes. The mRNAs are then extracted from the tissue samples collected for the experiment. They then hybridize with the probes in the chip. Raw microarray data are images, which are then transformed into numerical gene expression matrices. Regardless of the platform we

use, we obtain a numerical matrix, where the rows are genes and the columns are samples. These samples represent different entities or experimental conditions. These can be different individuals with different phenotype (e.g., disease state and healthy) or two tissue samples from the same individual but with different phenotype of the tissue (e.g., tumor cells and normal cells). For a detailed introduction to genomics, the various genomic technologies, and the importance of genomic data analysis, please refer to [Lesk, 2017].

Recent efforts have been made to make the data, from high-throughput gene expression profiling studies, publicly available. Popular public data repositories for gene expression datasets include The Cancer Genome Atlas (TCGA) [Edgar et al., 2002] and Gene Expression Omnibus (GEO) [Weinstein et al., 2013].

The gene expression datasets are high-dimensional with tens of thousands of candidate variables to help explain a few tens or hundreds of samples (e.g., individuals in the cohort). Such datasets are challenging for data mining methods. This dissertation focused on developing BRL methods to handle such data. In the next section, I explain how publicly available gene expression data was collected from the GEO data repository, pre-processed to prepare them for data analysis, modeled using BRL methods to find differentially expressed genes, and the experimental design and metrics used to evaluate the performance of BRL methods on the gene-expression datasets. These metrics will help compare BRL methods to other state-of-the-art data mining methods.

4.2 EXPERIMENTAL DESIGN

Experiments were conducted to evaluate each of the developed methods and were compared against state-of-the-art methods. Sub-section 4.2.1, gives the details about how these publicly available gene-expression datasets were collected from the GEO data repository for these experiments.

4.2.1 Data collection

A total of 25 real-world gene expression datasets were collected to evaluate the developed BRL methods. These were downloaded from the Gene Expression Omnibus (GEO) [Barrett et al., 2012], a public gene-expression data repository. The list of datasets downloaded are listed in the Table 1.

Data ID	GEO ID	Disease name	Platform name	Year submitted	Year updated
1	GSE66360	Acute Myocardial Infarction (AMI)	Affymetrix Human Genome U133 Plus 2.0 Array	2015	2019
2	GSE62646	Acute Myocardial Infarction (AMI)	Affymetrix Human Gene 1.0 ST Array	2014	2018
3	GSE41861	Asthma	Affymetrix Human Genome U133 Plus 2.0 Array	2012	2019
4	GSE20881	Inflammatory Bowel Disease (IBD)	Agilent Whole Human Genome Oligo Microarray	2010	2017
5	GSE3365	Inflammatory Bowel Disease (IBD)	Affymetrix Human Genome U133A Array	2006	2018
6	GSE16879	Inflammatory Bowel Disease (IBD)	Affymetrix Human Genome U133 Plus 2.0 Array	2009	2019
7	GSE15245	Multiple Sclerosis (MS)	Affymetrix Human Genome U133A Array	2009	2018
8	GSE6613	Parkinson's Disease	Affymetrix Human Genome U133A Array	2007	2018
9	GSE20295	Parkinson's Disease	Affymetrix Human Genome U133A Array	2005	2018
10	GSE30999	Psoriasis	Affymetrix Human Genome U133 Plus 2.0 Array	2011	2019
11	GSE55447	Systemic Lupus Erythematosus (SLE)	Illumina HumanHT-12 V4.0 expression beadchip	2014	2018
12	GSE19429	Myelodysplastic syndrome (MDS)	Affymetrix Human Genome U133 Plus 2.0 Array	2009	2019
13	GSE9006	Diabetes	Affymetrix Human Genome U133A Array	2007	2018
14	GSE48350	Alzheimer's disease	Affymetrix Human Genome U133 Plus 2.0 Array	2013	2019
15	GSE5281	Alzheimer's disease	Affymetrix Human Genome U133 Plus 2.0 Array	2006	2019
16	GSE35978	Schizophrenia, Bipolar, Depression	Affymetrix Human Gene 1.0 ST Array	2012	2019
17	GSE53987	Schizophrenia, Bipolar, MDD	Affymetrix Human Genome U133 Plus 2.0 Array	2014	2019
18	GSE12288	CAD	Affymetrix Human Genome U133A Array	2008	2018
19	GSE15852	Breast Cancer	Affymetrix Human Genome U133A Array	2009	2018
20	GSE42568	Breast Cancer	Affymetrix Human Genome U133 Plus 2.0 Array	2012	2019
21	GSE29431	Breast Cancer	Affymetrix Human Genome U133 Plus 2.0 Array	2011	2019
22	GSE18520	Ovarian Cancer	Affymetrix Human Genome U133 Plus 2.0 Array	2009	2019
23	GSE19804	Lung Cancer	Affymetrix Human Genome U133 Plus 2.0 Array	2010	2019
24	GSE10072	Lung Cancer	Affymetrix Human Genome U133A Array	2008	2018
25	GSE68571	Lung Cancer	Affymetrix Human Full Length HuGeneFL Array	2015	2016

Table 1: Datasets collected from GEO data repository.

4.2.1.1 Data pre-processing The probes are mapped to the genes they represent. Multiple probes can map to a single gene. In the final dataset, only one random variable is intended to represent a unique gene. Among the multiple probes that map to one gene, the probe with the largest inter-quantile range was chosen to represent the gene. This process is

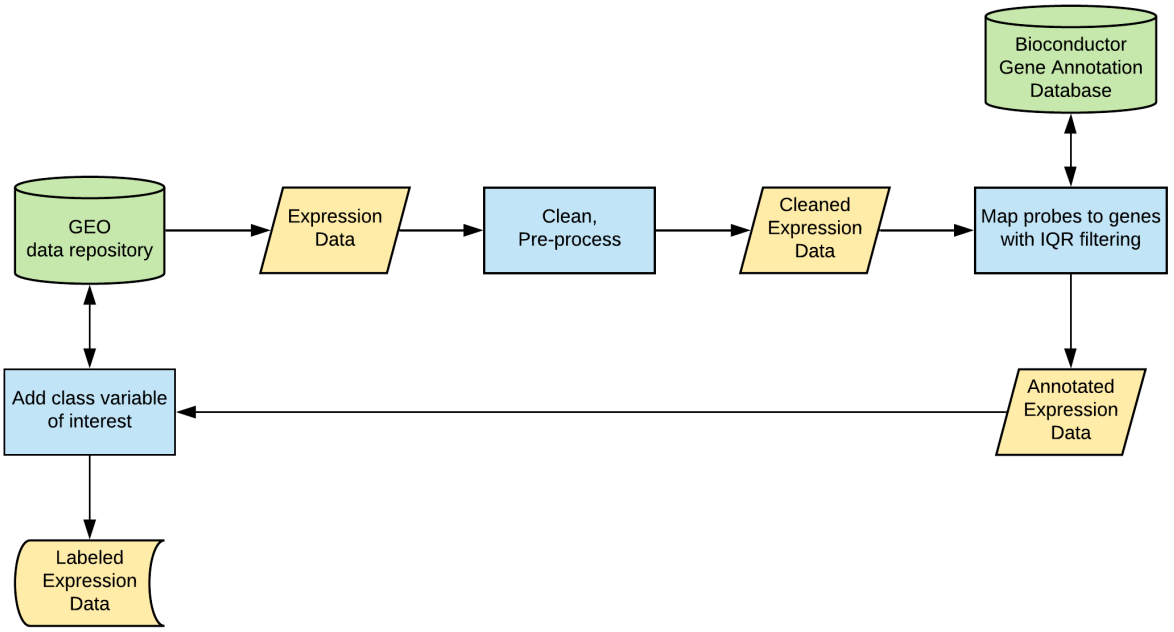


Figure 10: Data pre-processing

called inter-quantile range (IQR) filtering. We also extracted the tissue phenotype (tumor or normal) for each sample and add to this dataset as labels for classification.

4.2.1.2 Cross-validation design The experiment design is setup to perform a 10-fold stratified cross-validation. Each collected gene expression dataset is split into 10-folds. In each fold, the entire dataset is randomly split into 90% training data (on which the classifier is learned) and 10% test data (on which the learned classifier is evaluated). Over the 10 folds, the test datasets from each of the folds together compose mutually exclusive and exhaustive set of instances from the dataset. Stratified cross-validation ensures that the class distribution between the train and test datasets is similar.

4.2.1.3 Variable discretization Many data mining methods, for example— rule learning methods, Bayesian networks, decision tree learning algorithms, etc., cannot handle

Data ID	GEO ID	# instances	Class distribution (Case/Normal)	# variables	# variables mapped to genes
1	GSE66360	99	49/50	54675	20192
2	GSE62646	42	28/14	33297	18842
3	GSE41861	138	91/47	30427	14255
4	GSE20881	172	99/73	44290	17625
5	GSE3365	127	85/42	22283	12236
6	GSE16879	73	61/12	54675	19700
7	GSE15245	65	51/14	22215	12403
8	GSE6613	105	50/55	22283	12267
9	GSE20295	93	40/53	22283	12403
10	GSE30999	170	85/85	54675	20192
11	GSE55447	52	42/10	48107	18513
12	GSE19429	200	183/17	54675	19876
13	GSE9006	77	53/24	22283	11304
14	GSE48350	68	25/43	54675	20192
15	GSE5281	161	87/74	54675	20010
16	GSE35978	305	206/100	33297	18842
17	GSE53987	205	150/55	54675	19836
18	GSE12288	222	110/112	22283	12157
19	GSE15852	86	43/43	22283	11288
20	GSE42568	121	104/17	54675	20192
21	GSE29431	66	54/12	54675	20192
22	GSE18520	63	53/10	54675	19850
23	GSE19804	120	60/60	54675	20192
24	GSE10072	107	58/49	22283	12403
25	GSE68571	96	86/10	7129	4896

Table 2: Data properties

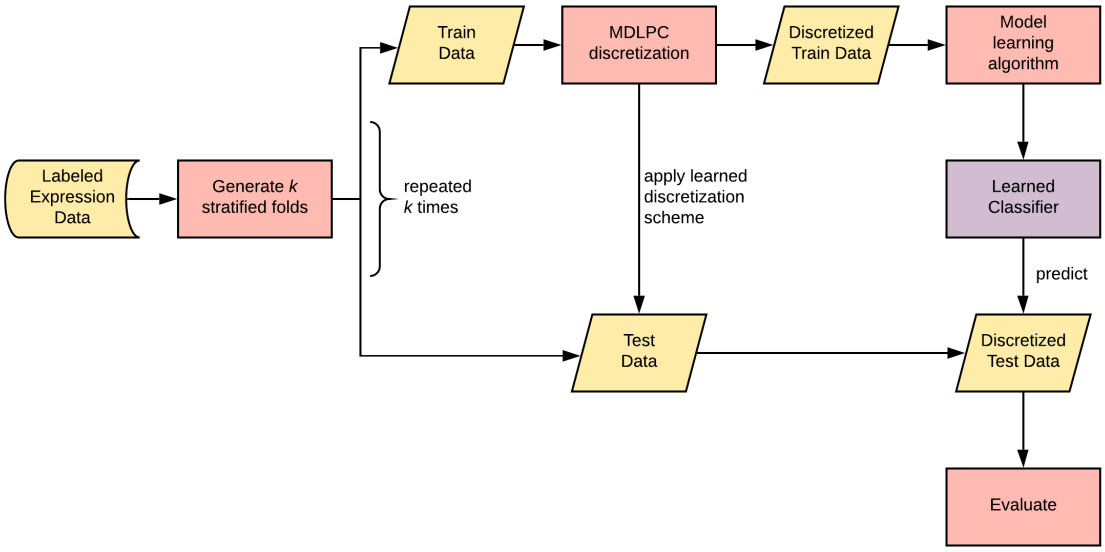


Figure 11: k-fold cross-validation study design

continuous-valued variables. In such cases, these variables need to be converted into discrete-valued variables before data analysis. Gene expression data contain variables that are raw microarray images that are transformed into a numerical matrix. So, all variables are continuous-valued.

Discretization is the process of transforming a continuous-valued variable into a discrete-valued variable. Minimum Description Length Principle Criterion method [Fayyad et al., 1993] (or MDLPC) is a popular univariate, supervised discretization method. Univariate discretization methods discretize one variable at a time (as opposed to multi-variate discretization approaches). Supervised discretization methods use the class variable-values to assist in finding the best discretization for a variable. Supervised discretization have even been shown to help improve the predictive performance of classifiers such as Support Vector Machines and Random Forests [Lustgarten et al., 2008]. This is because supervised discretization also acts as a feature selector that eliminates variables without any signal to help predict the class. Such a variable cannot be discretized using a supervised method. There can be many noisy variables or variables not associated with the class variable. Supervised

discretization can help remove some of these variables from the model learning process.

MDLPC is a greedy algorithm uses a decision tree to split a continuous-valued variable into bins divided at a specific value (called the *cut-point*). The algorithm evaluates all possible cut-points and computes the entropy (lower is better) of the resulting split. The split that resulted in the lowest entropy is chosen and the gain in entropy is computed. Finally, a decision criterion based on Minimum Description Length [Rissanen, 1978] is used to evaluate whether or not we accept the chosen split. If accepted, the value ranges in the bins, as defined by the winning cut-point, is used to give a discrete label to the continuous values in the bin.

For each fold, the training data is used to perform supervised discretization using MDLPC (as described above). The learned cut-points are used to discretize the test dataset in the same fold. This is repeated for each of the 10 folds. This is done so that MDLPC does not look at the class values of the test dataset to learn the cut-points. Otherwise, the results would be biased when evaluating the data mining methods as they would have used the discretized variables that had cut-points learned using the information from the test dataset.

4.2.2 Evaluation metrics

All evaluated classifiers were evaluated using the same implementation of the predictive metric computation. This was done to avoid biased estimates computed in Weka, for certain evaluation metrics, across cross-validation as pointed out by [Forman and Scholz, 2010].

4.2.2.1 Predictive metrics In a classification problem, predictive metrics measure the discrimination ability of a classifier in distinguishing the different classes in the problem domain that the instance may belong to. Six predictive metrics are measured in the experiments here, they are—

(i) Accuracy

(ii) Precision

(iii) Recall

(iv) F-measure

(v) Area under receiver operator characteristic curve (AUROC)

(vi) Area under precision recall gain curve (AUPRG)

(i) Accuracy Every classification task can be simplified as a binary classification problem by iteratively labeling each class as class *positive* and labeling every other class as class *negative*. For a binary classification task, when a classifier makes a prediction on the likely class for a given instance, the prediction can belong to one of the four possible scenarios— 1) *True positive* or TP: the instance belongs to the positive class and the classifier also predicts positive. 2) *False positive* or FP: the instance belongs to the positive class but the classifier incorrectly predicts class negative. This is also known as *Type I error*. 3) *False negative* or FN: the instance belongs to class negative but the classifier incorrectly predicts as class positive. This is also known as *Type II error*. 4) *True negative* or TN: the instance belongs to class negative and the classifier correctly predicts as class negative.

Accuracy is a measure of the total number of instances whose class the classifier predicts correctly, divided by the total number of instances predicted. Accuracy is computed with Equation 4.1.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.1}$$

Accuracy ranges from 0.0% to 100.0%. Higher values generally indicate better predictive classifiers. For a k-fold cross-validation evaluation design, for a given dataset, TP, FP, FN, and TN are summed across all of the k folds and then the accuracy is computed using Equation 4.1. A drawback of using accuracy to evaluate the classifiers is that they sensitive to class imbalance. In those cases, higher values of class accuracy does not necessarily mean that the classifier is good at discriminating the two classes.

(ii) Precision Precision (or *positive predictive value*) is the fraction of correctly predicted instances among all instances predicted to belong to the positive class by the classifier. Precision is an important metric to monitor when it is expensive to make an incorrect positive prediction. For example, in fraud detection, it may be expensive for a bank to

falsely accuse an innocent user of fraud. Precision is computed by Equation 4.2.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Precision ranges from 0.0 to 1.0. Higher values generally indicate better predictive classifiers. For a k-fold cross-validation evaluation design, for a given dataset, precision is computed for each fold and then averaged across the k folds. For a fold, where TP and FP are both 0, the precision is set to 0, in accordance with [Forman and Scholz, 2010].

(iii) Recall Recall (or *sensitivity*) is the fraction of positive class instances that are correctly predicted to belong to the positive class by the classifier. Recall is a useful metric to monitor in applications, where it is expensive to miss predicting a positive instance. For example, it may be very expensive to miss identifying a patient who may develop a certain disease. Recall is computed by Equation 4.3.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

Recall ranges from 0.0 to 1.0. Higher values generally indicate better predictive classifiers. For a k-fold cross-validation evaluation design, for a given dataset, recall is computed for each fold and then averaged across the k folds. For a fold, where both TP and FN are 0, then recall is set to 0, in accordance with [Forman and Scholz, 2010].

(iv) F-measure F-measure (or *F1 score*) is a metric that computes the harmonic mean between precision and recall. This metric is insensitive to class imbalance and particularly interesting when the data suffers from class imbalance as is typically the case in gene expression data and many other biomedical datasets. F-measure is given by Equation 4.4.

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4.4}$$

The F-measure ranges from 0.0 to 1.0. Higher values generally indicate better predictive classifiers. As described in [Forman and Scholz, 2010], for k-fold cross-validation evaluation design, for a given dataset, unbiased estimate for F-measure is computed by first summing

TP, FP, and FN across all k folds and then calculated using Equation 4.5.

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{4.5}$$

If all of TP, FP, and FN are equal to 0, the F-measure is set to 0.

(v) Area under receiver operator characteristic curve (AUROC) Most classifiers produce a continuous-valued score to predict the probability that the queried instance belongs to the positive class. A receiver operating characteristic curve (or ROC curve) [Fawcett, 2006] is a graphical plot that represents the trade-off between true positive rates and false positive rates for different threshold values of the classifier scores. For a given instance and a specific threshold value for the classifier score, if the classifier evaluates a score for the instance that is higher than or equal to the threshold, the classifier predicts the instance to belong to the positive class. If the score is less than the threshold, then the classifier predicts the instance to belong to the negative class. For each threshold value, we can assign class memberships to the instances in the test dataset and calculate the true positive rate and false positive rate for that threshold. The true positive rate (or *recall*, which we saw earlier) is the fraction of individuals correctly predicted positive by the model, among all the positive instances i.e., True positive rate $= \frac{TP}{TP+FN}$. The false positive rate (or *false alarm rate*) is the fraction of instances correctly predicted negative by the classifier, among all negative instances i.e., False positive rate $= \frac{TN}{TN+FP}$. Once this curve is plotted, the *area under the ROC curve* ($AUROC$) is calculated. This area corresponds to the probability that the classifier score will rank a randomly chosen positive instance higher than a randomly chosen negative instance. The AUROC provides a convenient one-dimensional metric to evaluate the ability of the classifier to discriminate the positive and the negative class.

The AUROC is closely related to the Wilcoxon signed-ranks test or the Mann-Whitney U test [Hanley and McNeil, 1982]. The AUROC can be calculated using the U statistic as follows— Each test instance is assigned a score corresponding to the probability of it belonging to the positive class according to the classifier being evaluated. After all instances are scored, they are sorted in ascending order of scores. The instance with the lowest score is assigned a rank of 1, second lowest gets the score of 2, and so on. Instances with tied scores have their ranks averaged. After each instance has been given a rank this way, we

look at only the instances that actually belonged to the positive class. We sum the ranks obtained by the positive instances and refer to this number as R_p. U statistic is calculated using Equation 4.6.

$$U_p = R_p - \frac{n_p(n_p + 1)}{2} \tag{4.6}$$

R_p is the average rank achieved by the positive instances and n_p is the total number of positive instances in the test dataset. The AUROC for the positive class is calculated using Equation 4.7.

$$AUROC_p = \frac{U_p}{n_p n_{\neg p}} \tag{4.7}$$

Here, $n_{\neg p}$ is the total number of instances in the test dataset that do not belong tot he positive class. To compute the AUROC for multi-class problem, the AUROC can be averaged using Equation 4.8.

$$AUROC = \frac{1}{|C|} \sum_{c \in C} AUROC_c \tag{4.8}$$

Here, C is the set of all classes, $|C|$ is the total number of classes, c is the c-th class, and $AUROC_c$ is the AUROC computed by Equation 4.7, where c-th class is considered positive and all other classes are considered negative class.

The AUROC ranges from 0.0 to 1.0. Higher values generally indicate better predictive classifiers. As described in [Forman and Scholz, 2010], for k-fold cross-validation evaluation design, for a given dataset, unbiased estimate for AUROC is computed by averaging AUROC across the k folds.

(vi) Area under precision recall gain curve (AUPRG) In some binary classification problem, true negative predictions do not help evaluate the classifier predictive performance. This is especially true in domains where data is skewed with significantly more negatives examples than positive examples. In such cases, often precision-recall curves are used. [Flach and Kull, 2015] argue that this is a source of many biases and propose the area under the precision-recall gain curve. Just as AUROC generalizes the accuracy metric by accounting for all possible threshold values, AUPRG generalizes the F-measure to account for all possible

β values in the more generalized F-beta score. The F-beta score is computed using Equation 4.9.

$$f_\beta = (1+\beta)\frac{precision \cdot recall}{\beta \cdot precision + recall} \tag{4.9}$$

The AUPRG ranges from -1.0 to 1.0. Higher values generally indicate better predictive classifiers. For k-fold cross-validation evaluation design, for a given dataset, AUPRG is computed by averaging AUPRG across the k folds.

4.2.2.2 Calibration metrics In a classification problem, classifiers typically compute the probability that a new test instance belongs to a specific class, called class membership probability, using a score. Calibration refers to the transformation of classifier scores for class membership into probabilities. In prediction and forecasting, calibration metrics is used to evaluate the prediction accuracy of class membership probabilities assigned by a classifier, when compared to the actual frequency of the observed outcomes. A well-calibrated classifier predicts the probability of an outcome as p, when the outcome does occur p fraction of the times, for all values of p [DeGroot and Fienberg, 1983].

I evaluate the calibration of the classifiers studied here using the following three metrics—

(i) Brier score
(ii) Expected Calibration Error (ECE)
(iii) Maximum Calibration Error (MCE)

(i) Brier score Brier score [Brier, 1950] is a popular calibration metric to compute the accuracy of the classifier predicted probabilities. It is computed for each instance predicted by the classifier, $i \in 1 \cdots m$, for m total predictions. For a specific positive class, Brier score is computed as shown in Equation 4.10.

$$\text{Brier score} = \frac{1}{m}\sum_{i=1}^{m}(f_i - o_i)^2 \tag{4.10}$$

Here, f_i is the probability that the instance belongs to the positive class, as predicted by the classifier. o_i is the actual outcome for the i^{th} instance. For our classification problem, this function returns value 1 if the instance belongs to the positive class, or value 0 otherwise.

Error is computed by subtracting the predicted probability of the outcome and the actual outcome. The error is squared then summed across all predicted instances. We then compute the mean of the sum of the squared error. In effect, the Brier score is the same as mean squared error.

Brier score ranges from 0.0 to 1.0. It is the expected squared calibration error for an instance. Classifiers with lower Brier score are better calibrated.

(ii) Expected Calibration Error (ECE) [Naeini et al., 2015] define an intuitive calibration metric, meant for classifers, called Expected Calibration Error (ECE). The classifier probability predictions of m test instances are binned into B equal-sized bins. B is a user-defined parameter set to $B = 10$, like the authors of the paper. ECE is computed as shown in Equation 4.11.

$$\text{ECE} = \sum_{i=1}^{B} P(i) \cdot |o_i - f_i| \tag{4.11}$$

Here, $P(i)$ is the proportion of predicted instances contained in bin i. So, total number of instances in bin i divided by the total number of instances; o_i is the observed outcomes, i.e., the proportion of positive instances in the i^{th} bin; f_i is the classifier predicted probability of belonging to the positive class, averaged for instances in the i^{th} bin. We take the absolute error between the predicted outcome and actual outcome, weighted by the proportion of instances.

ECE ranges from 0.0 to 1.0. It is the expected calibration error for an instance. Classifiers with lower ECE are better calibrated.

(iii) Maximum Calibration Error (MCE) [Naeini et al., 2015] define another calibration metric, for classifiers, called Maximum Calibration Error (MCE). Similar to ECE, MCE takes a user-defined parameter of the number of bins, B, set $B = 10$ for the experiments here. MCE is computed as shown in Equation 4.12.

$$\text{MCE} = \max_{i=1}^{B}(|o_i - f_i|) \tag{4.12}$$

Here, o_i is the proportion of positive instances in the i^{th} bin. f_i is the average classifier predicted probability of belonging to the positive class, for the i^{th} bin. We compute the absolute error and return the maximum error over B bins.

MCE ranges from 0.0 to 1.0. It measures the largest calibration error for an instance made by the classifier. Classifiers with lower MCE are better calibrated.

4.2.2.3 Semantic complexity metrics

Semantic complexity metrics help assess the readability of models. In the following experiments, we evaluate semantic complexity in terms of the Number of rules (NR) and Number of variables (NV).

Number of rules (NR) Number of rules (NR) is the total number of rules used by the rule learning model. Generally, rule models with fewer rules are considered to be more readable.

Number of variables (NV) Number of variables (NV) is the total number of unique variables selected by the rule learning model to build the entire set of rules within the classifier. Generally, rule models with fewer variables are considered more readable. They are also more efficient since in practice, fewer variables need to be validated to evaluate the proposed model.

Both these metrics are only relevant to the rule-based classifiers used in the experiments, including C4.5, RIPPER, PART, and the BRL methods. A method to return this metric was implemented in BRL API. However, Weka API does not provide this metric, we wrote a parser in BRL that parses the Weka model output and computes this metric.

4.2.3 Decision theory: choosing an optimal threshold

Typically, in a classification task, statistics helps us learn a model that assigns optimally calibrated probabilities for an instance belonging to the positive class. This is also known as the *prediction problem*. The *classification problem*, on the other hand, is using the model predicted probabilities to decide the class of the instance. To that end, the reliable metrics for evaluating model predictive performance are— area under the ROC curve (AUROC), area under the Precision-Recall gain curves (AUPRG), Brier score, expected calibration error (ECE), and maximum calibration error (MCE). The unreliable metrics are— Accuracy, Precision, Recall, and F-measure. The reason those metrics are unreliable is because they are meant to evaluate classification performance. We often choose an arbitrary threshold

for classification. For example, choosing a probability cut-off of 0.5, such that if the model evaluates the probability of the instance belonging to the positive class is greater than 0.5 then we classify the instance as positive. This choice of cut-off is arbitrary. The evaluation section in this chapter is also done using this arbitrary cut-off of 0.5, which means the metrics Accuracy, Precision, Recall, and F-measure will not be used for judging the predictive performance of the classifiers.

The classification problem requires a decision theoretic approach. Prematurely assuming that the cost of false positive and false negative is equal, thereby using a cut-off value of 0.5, incorrectly makes a decision ahead of the decision maker. Instead, the choice of cut-off should be made at the point-of-care and is not a part of data analysis. For optimum decision making, the decision maker needs all the available data, reliable model predictions, and a utility function. The utility function could be to minimize the expected loss, maximize some utility function (e.g., F1-score, Recall), etc.

4.2.4 Significance testing

For each dataset, an evaluation metric is computed across 10-folds as described above. Then the metric is averaged across the 25 datasets. We now want to know if one or more algorithms is statistically significantly different than other algorithms, for a given metric, evaluated over the 25 datasets.

4.2.4.1 Parametric or non-parametric method The first question in significance testing is whether to use parametric or non-parametric methods. Parametric methods are more powerful but only when the assumptions hold. Specifically, the assumptions of normality and homoscedasticity must hold. Normally distributed data follow the Gaussian distribution. They are unimodal and symmetric. Homoscedastic data have variables with the same finite variance. These assumptions can be tested using methods like Shapiro-Wilk test for normality and the Bartlett's test for homoscedasticity. However, both these methods are not powerful and are affected by small sample sizes, which make them uneffective. Alternatively, we can visually check for normality and homoscedasticity by simply plotting the

data. Also, quantile-quantile plots can help us visually test for normality. However, [Demšar, 2006] argues that machine learning experiment analysis and evaluation of optimization algorithms do not generate data that hold these assumptions and so, no parametric tests are suitable in these scenarios. For all the hypotheses testing in this chapter, I performed the visual tests of normality. All of them were found to be non-normal. So, non-parametric tests were used for significance testing.

4.2.4.2 Global test for significance The second question in significance testing is to check whether one (or more) of the algorithms, being compared, behave significantly different from others. The null hypothesis here is that all classifiers perform the same and the observed differences are merely random. The alternate hypothesis is that all classifiers do not perform the same. If the normality and homoscedasticity assumptions were to hold, we can test this using the parametric repeated-measures ANOVA test. If these assumptions don't hold, we must use non-parametric tests for significance.

Wilcoxon signed-ranks test: For comparing only two algorithms, [Demšar, 2006] recommends using Wilcoxon signed-ranks test[Wilcoxon, 1992]. It is the non-parametric equivalent of paired Student's t-test. Here, the null hypothesis is that the median difference between the performances of the two classifiers is zero. The alternate hypothesis is that the median difference between the two classifier performance is not zero.

We first list the classifier performance for each of the n datasets. The difference between the classifier performance values is computed, then its absolute values are computed. The absolute differences are then ranked in ascending order. The ranks for tied ranks are averaged. If the second classifier outperformed the first classifier, their ranks are summed to R_+. If the first classifier outperforms the first, their ranks are summed to R_-. Absolute differences of 0 are split evenly among the sums and if there are odd numbers of them, one is dropped.

To compute the Wilcoxon statistic, we simply take the minimum of the two sums i.e., $T = \min(R_+, R_-)$. We then compare this statistic to the critical value T_c from the Wilcoxon distribution for a specified confidence level of α and n degrees of freedom (number of datasets compared). If $T < T_c$, we reject the null hypothesis that the median difference between the

performances of the two classifiers is zero.

Friedman test: For comparing more than two algorithms, [Demšar, 2006] recommends using the Friedman test [Friedman, 1937]. It is the non-parametric equivalent of repeated-measures ANOVA test. For each dataset being evaluated, this test ranks the classifiers in order of their performance. Rank 1 being the best performing classifier, rank 2 being second best, and so on. In case of ties, the ranks are averaged. Say, we want to compare k algorithms over n datasets. Let r_i^j be the rank of the j-th algorithm on the i-th dataset. Then Freidman test compares the average rank of classifiers across the n datasets i.e., $R_j = (\frac{1}{n}) \sum_{i=1}^{n} r_i^j$. Under the null hypothesis that the average rank of the classifiers are equal, the Friedman statistic is given by Equation 4.13.

$$\chi_F^2 = \frac{12 \cdot n}{k(k+1)} \left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4} \right] \tag{4.13}$$

Under the null hypothesis, the statistic is distributed according to χ_F^2 with $(k-1)$ degrees of freedom.

Friedman test with Iman-Davenport correction: [Iman and Davenport, 1980] show that the Friedman statistic is overly conservative and suggested an F-statistic with a correction, as shown in Equation 4.14.

$$F_F = \frac{(n-1) \cdot \chi_F^2}{n \cdot (k-1) - \chi_F^2} \tag{4.14}$$

Under the null hypothesis, this statistic is distributed according to the F-distribution with $(k-1)$ and $(k-1)(n-1)$ degrees of freedom.

Friedman aligned rank test: Friedman test depends upon n sets of ranks, one for each dataset, containing the classifier rankings based on their performance.

$$\chi_{F-align}^2 = \frac{(k-1) \cdot \left[\sum_{j=1}^{k} R_j^2 - (kn^2/4)(kn+1)^2 \right]}{\{[kn(kn+1)(2kn+1)]/6\} - (1/k) \sum_{i=1}^{n} R_i^2} \tag{4.15}$$

Under the null hypothesis, this statistic is distributed according to $\chi_{F-align}^2$ with $(k-1)$ degrees of freedom.

Quade test: The Friedman test assumes all datasets to be equally important. However, sometimes, some datasets are harder than others. There are datasets where the classifier

performances differ a lot. The Quade test computes rankings for each dataset, scales it based on the differences in the classifier performances on that dataset, and conducts a weighted ranking analysis of the results.

When comparing multiple classifier performances, if the null hypothesis is rejected by any of these global methods, we need to perform post-hoc tests to detect which classifiers actually differ.

4.2.4.3 Post-hoc test for significance

Given that we rejected the null hypothesis that all the compared classifiers perform the same, post-hoc tests further help us identify which classifiers perform differently. There are two main approaches to performing post-hoc tests— 1) Finding pairwise differences and 2) Comparing with a control classifier.

For pairwise differences, we used Bergmann and Hommel's method to correct the p-values generated by the global test. For comparison with control, we used the following four methods to correct the p-values— Holland, Finner, Rom, and Li's method [García et al., 2010].

4.3 EXPERIMENT 1: EVALUATING BRL METHODS

In experiment 1, we will test 3 hypotheses, as follows—

1. Experiment 1a (see subsection 4.3.1): Does BRL.G achieve better predictive and calibration performance when compared to other popular state-of-the-art (SOTA) rule learning classifiers (C4.5, RIPPER, and PART)? BRL.G is also expected to require significantly fewer variables to model the data than the compared SOTA classifiers.
2. Experiment 1b (see subsection 4.3.2): BRL.G, BRL.DT, and BRL.DG are expected to have similar predictive and calibration performance. BRL.DG is expected to achieve significantly higher Bayesian scores and fewer rules than BRL.DT. In turn, BRL.DT is expected to achieve significantly higher Bayesian scores and require fewer rules than BRL.G.

3. Experiment 1c (see subsection 4.3.3): Experiment 1b is repeated with an improved search space using beam search instead of greedy best-first search. Since BRL.DT and BRL.DG search over a much denser space than BRL.G, we expect the improved search to benefit the performances of BRL.DT and BRL.DG.

Subsection 4.3.4 compares the AUROC, AUPRG, and Brier scores of the best performing BRL algorithm (from the experiments), to other popular state-of-the-art algorithms (not just limited to rule learning classifiers). Subsection 4.3.5 summarizes the results from experiment 1.

4.3.1 Experiment 1a: BRL.G compared to state-of-the-art rule learning classifiers

This experiment was conducted to test if BRL.G achieves a better predictive performance than other state-of-the-art rule learning classifiers. The representative examples for state-of-the-art rule learning classifiers compared here are— C4.5 [Quinlan, 2014], RIPPER [Cohen, 1995], and PART [Frank and Witten, 1998].

Accuracy, Precision, Recall, and F-measures achieved by the classifiers over the 25 datasets are shown in Appendix C (see 9.1). They do not help us determine which of the compared models have a better predictive performances. So, no significance testing was done with these metrics.

To compare the model predictive performance and test if BRL.G performs better than the other algorithms; AUROC, AUPRG, Brier score, ECE, and MCE performances are compared. Significance testing is only performed on these predictive performance metrics.

AUROCs achieved by the different classifiers were first compared against BRL.G. The results are shown in Table 3. Friedman's test (p-value = 0.001586), Friedman's test with Iman-Davenport correction (p-value = 0.0008853), Friedman's aligned ranks test (p-value = 0.01641), and Quade test (p-value = 0.005958) all suggest that we must reject the null hypothesis suggesting that all classifiers have the same AUROC. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were adjusted using Holland's method to find that the AUROC achieved by

BRL.G is statistically significantly different from C4.5 (p-value = 0.002050848), RIPPER (p-value = 0.006169899), and PART (p-value = 0.002050848). Similar conclusions were reached using Finner, Rom, and Li methods. These results suggest that BRL.G generally achieves a statistically significantly higher AUROC than C4.5, RIPPER, and PART.

AUPRGs achieved by the different classifiers were first compared against BRL.G. The results are shown in Table 4. Friedman's test (p-value = 0.07718), Friedman's test with Iman-Davenport correction (p-value = 0.07408), Friedman's aligned ranks test (p-value = 0.1227), and Quade test (p-value = 0.1531) all suggest that we cannot reject the null hypothesis suggesting that all classifiers have the same AUPRGs. However, the average AUPRG achieved by BRL.G (0.5939 ± 0.0644) is much higher than C4.5 (0.5473 ± 0.00638), RIPPER (0.5382 ± 0.0656), and PART (0.5475 ± 0.0644). These results suggest that perhaps more datasets need to be evaluated to establish a more definitive conclusion from this metric.

The next three metrics— Brier score, ECE, and MCE— help evaluate the calibration performance.

Brier scores achieved by the different classifiers were first compared against BRL.G. The results are shown in Table 5. Friedman's test (p-value = 0.001508), Friedman's test with Iman-Davenport correction (p-value = 0.0008321), Friedman's aligned ranks test (p-value = 0.001865), and Quade test (p-value = 0.0005386) all suggest that we must reject the null hypothesis suggesting that all classifiers have the same Brier scores. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were adjusted using Holland's method to find that the Brier score achieved by BRL.G is statistically significantly different from C4.5 (p-value = 0.001676764), RIPPER (p-value = 0.02845974), and PART (p-value = 0.001676764). Similar conclusions were reached using Finner, Rom, and Li methods. These results suggest that BRL.G generally achieves a statistically significantly better Brier score than C4.5, RIPPER, and PART.

ECEs achieved by the different classifiers were first compared against BRL.G. The results are shown in Table 6. Friedman's test (p-value = 0.04123) and Friedman's test with Iman-Davenport correction (p-value = 0.03772) suggest that we must reject the null hypothesis suggesting that all classifiers have the same ECE. However, Friedman's aligned ranks test (p-value = 0.1309), and Quade test (p-value = 0.2444) suggest that we cannot reject the null

hypothesis. Continuing from the results from Friedman's test, post-hoc test is conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were adjusted using Holland's method to find that the ECE achieved by BRL.G is statistically significantly different from C4.5 (p-value = 0.02983163) and PART (p-value = 0.04883991) but not different from RIPPER (p-value = 0.2733217). Similar conclusions were reached using Finner, Rom, and Li methods. These results suggest that BRL.G generally achieves a statistically significantly worse ECE than C4.5 and PART.

MCEs achieved by the different classifiers were first compared against BRL.G. The results are shown in Table 7. Friedman's test (p-value = 0.5351), Friedman's test with Iman-Davenport correction (p-value = 0.5434), Friedman's aligned ranks test (p-value = 0.7471), and Quade test (p-value = 0.9455) all suggest that we cannot reject the null hypothesis suggesting that all classifiers have the same MCE.

The next two metrics— average number of rules (NR) and average number of variables (NV)— help evaluate the model parsimony.

Average NR needed by the different classifiers were first compared against BRL.G. The results are shown in Table 8. Friedman's test (p-value = $5.597e-12$), Friedman's test with Iman-Davenport correction (p-value < $2.2e-16$), Friedman's aligned ranks test (p-value = $7.603e-11$), and Quade test (p-value = $2.22e-16$) strongly suggest that we reject the null hypothesis suggesting all classifiers use the same number of rules. Post-hoc test is conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were adjusted using Holland's method to find that the NR needed by BRL.G is statistically significantly different from C4.5 (p-value = 0.0001956923), RIPPER (p-value = $1.280736e-08$), and PART (p-value = $1.54583e-11$). Similar conclusions were reached using Finner, Rom, and Li methods. These results suggest that BRL.G on average needs statistically significantly more rules than C4.5, RIPPER, and PART.

Average NV needed needed by the different classifiers were first compared against BRL.G. The results are shown in Table 9. Friedman's test (p-value = 0.01922), Friedman's test with Iman-Davenport correction (p-value = 0.01626), Friedman's aligned ranks test (p-value = 0.0001692), and Quade test (p-value = $5.741e-06$) strongly suggest that we reject the null hypothesis suggesting all classifiers use the same number of rules. Post-hoc test is conducted

to determine which of the classifiers are different from others. The p-values generated from Friedman's test did not identify significantly different performing classifiers using any of Holland, Finner, Rom and Li methods. Instead, the p-values generated by Friedman's aligned test were used. They were adjusted using Holland's method to find that the NV needed by BRL.G is statistically significantly different from C4.5 (p-value = 0.001833508) and PART (p-value = $6.749709e-05$) but not RIPPER (p-value = 0.6173192). Similar conclusions were reached using Finner, Rom, and Li methods. These results suggest that BRL.G on average needs statistically significantly fewer variables than C4.5 and PART.

4.3.1.1 Experiment 1a: Conclusion

Experiment 1a shows that BRL.G achieves a statistically significantly better predictive performance than C4.5, RIPPER, and PART in terms of AUROC and Brier score. But BRL.G had statistically significantly worse ECE than C4.5 and PART. However, in terms of AUPRG and MCE the classifiers were statistically indistinguishable. In terms of model parsimony, BRL.G requires significantly more rules but requires significantly fewer variables than C4.5 and PART.

To summarize, BRL.G presents an alternative to C4.5, RIPPER, and PART that is likely to attain better predictive and require fewer variables to achieve this performance. These results indicate that BRL.G is a suitable candidate for biomarker discovery, when it is expensive to validate the biomarkers. By choosing fewer and better predicting biomarkers, BRL.G is an important model to consider for biomarker discovery tasks.

4.3.2 Experiment 1b: Comparing BRL.G, BRL.DT, and BRL.DG

Experiment 1b compares the performances of BRL.G, BRL.DT, and BRL.DG. The tables showing the Accuracy, Precision, Recall, and F-measures achieved by the three classifiers over the 25 datasets is moved to Appendix C (see 9.2).

To compare the model predictive performance of the different BRL algorithms, the metrics Bayesian score, AUROC, AUPRG, Brier score, ECE, and MCE are compared. Significance testing is only performed on these predictive performance metrics.

Bayesian scores achieved by the three classifiers are shown in Table 10. Friedman's

Data	C4.5	RIPPER	PART	BRL.G
GSE66360	0.6940	0.6995	0.7040	0.8640
GSE62646	0.9083	0.9083	0.9083	0.9083
GSE41861	0.7164	0.7531	0.7267	0.7401
GSE20881	0.7518	0.8246	0.7182	0.8117
GSE3365	0.7994	0.7931	0.7940	0.9311
GSE16879	0.9845	0.8917	0.9845	0.9845
GSE15245	0.6083	0.4550	0.6233	0.6950
GSE6613	0.5847	0.5887	0.5590	0.4563
GSE20295	0.6413	0.5775	0.6650	0.5750
GSE30999	0.9458	0.9646	0.9396	0.9722
GSE55447	0.6125	0.4825	0.5525	0.6175
GSE19429	0.6254	0.7003	0.5838	0.9157
GSE9006	0.7383	0.7483	0.7783	0.8458
GSE48350	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8263	0.8540	0.8170	0.8436
GSE35978	0.5129	0.5960	0.5917	0.6013
GSE53987	0.5502	0.5029	0.5041	0.5581
GSE12288	0.5357	0.5996	0.5207	0.5669
GSE15852	0.7631	0.7888	0.7831	0.8569
GSE42568	0.8955	0.8455	0.8955	0.8109
GSE29431	0.9417	0.9317	0.9417	0.9417
GSE18520	0.9900	0.9900	0.9900	0.9900
GSE19804	0.8806	0.8931	0.8889	0.9153
GSE10072	0.9425	0.9258	0.9425	0.9425
GSE68571	0.9938	0.9938	0.9938	0.9938
Average ± SEM	0.7777 ± 0.0326	0.7723 ± 0.0341	0.7762 ± 0.0334	0.8135 ± 0.0329

Table 3: Experiment 1a: Area under the ROC cruves (AUROCs) for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of AUROCs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	C4.5	RIPPER	PART	BRL.G
GSE66360	0.4213	0.4575	0.4313	0.7558
GSE62646	0.7833	0.7833	0.7833	0.7833
GSE41861	0.4223	0.4780	0.4263	0.3882
GSE20881	0.5146	0.6616	0.4456	0.6381
GSE3365	0.5734	0.5187	0.5600	0.8003
GSE16879	0.8845	0.7417	0.8845	0.8845
GSE15245	0.1481	-0.0308	0.1882	0.3119
GSE6613	0.1942	0.2067	0.1447	-0.0736
GSE20295	0.3233	0.1661	0.3819	0.2157
GSE30999	0.8988	0.9404	0.8863	0.9548
GSE55447	0.2500	-0.0075	0.1400	0.1175
GSE19429	0.1514	0.2630	0.1069	0.4804
GSE9006	0.4379	0.4405	0.5267	0.6300
GSE48350	1.0000	1.0000	1.0000	1.0000
GSE5281	0.6788	0.7374	0.6619	0.6881
GSE35978	0.0644	0.1473	0.2064	0.1944
GSE53987	0.0854	0.0417	0.0286	0.1127
GSE12288	0.0855	0.2562	0.0591	0.1829
GSE15852	0.5590	0.6043	0.6103	0.7579
GSE42568	0.7955	0.6955	0.7955	0.5631
GSE29431	0.8417	0.7817	0.8417	0.8417
GSE18520	0.9400	0.9400	0.9400	0.9400
GSE19804	0.7937	0.8244	0.8020	0.8435
GSE10072	0.8920	0.8645	0.8920	0.8920
GSE68571	0.9438	0.9438	0.9438	0.9438
Average $\pm$ SEM	0.5473 ± 0.0638	0.5382 ± 0.0656	0.5475 ± 0.0644	0.5939 ± 0.0644

Table 4: Experiment 1a: Area under precision-recall gain curves (AUPRGs) for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of AUPRGs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	C4.5	RIPPER	PART	BRL.G
GSE66360	0.2889	0.2475	0.2886	0.2169
GSE62646	0.0700	0.0701	0.0700	0.0684
GSE41861	0.2482	0.2254	0.2470	0.2339
GSE20881	0.2442	0.1724	0.2687	0.2007
GSE3365	0.1884	0.1590	0.1887	0.1039
GSE16879	0.0268	0.0539	0.0268	0.0257
GSE15245	0.3128	0.3079	0.2984	0.2235
GSE6613	0.4097	0.3620	0.4243	0.4136
GSE20295	0.3467	0.3915	0.3228	0.3568
GSE30999	0.0530	0.0354	0.0587	0.0497
GSE55447	0.3194	0.3348	0.3402	0.2805
GSE19429	0.1436	0.0850	0.1302	0.0824
GSE9006	0.2049	0.2162	0.1753	0.2125
GSE48350	0.0000	0.0000	0.0000	0.0001
GSE5281	0.1804	0.1510	0.1847	0.1621
GSE35978	0.3918	0.3057	0.3383	0.2577
GSE53987	0.2996	0.3101	0.3549	0.2401
GSE12288	0.4541	0.3469	0.4587	0.3149
GSE15852	0.2271	0.1998	0.2045	0.1668
GSE42568	0.0251	0.0417	0.0251	0.0806
GSE29431	0.0286	0.0429	0.0286	0.0279
GSE18520	0.0167	0.0167	0.0167	0.0159
GSE19804	0.1125	0.0980	0.1122	0.1092
GSE10072	0.0556	0.0743	0.0556	0.0552
GSE68571	0.0111	0.0111	0.0111	0.0106
Average ± SEM	0.1864 ± 0.0283	0.1704 ± 0.0254	0.1852 ± 0.0284	0.1564 ± 0.0234

Table 5: Experiment 1a: Brier scores for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with lower values of Brier score are better calibrated for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	C4.5	RIPPER	PART	BRL.G
GSE66360	0.0309	0.0220	0.0309	0.0265
GSE62646	0.0250	0.0250	0.0250	0.0289
GSE41861	0.0000	0.0250	0.0005	0.0281
GSE20881	0.0005	0.0115	0.0061	0.0119
GSE3365	0.0243	0.0161	0.0244	0.0130
GSE16879	0.0000	0.0302	0.0000	0.0030
GSE15245	0.0455	0.0572	0.0444	0.0535
GSE6613	0.0456	0.0201	0.0558	0.0644
GSE20295	0.0433	0.0341	0.0535	0.0507
GSE30999	0.0059	0.0117	0.0060	0.0060
GSE55447	0.0210	0.0572	0.0210	0.0408
GSE19429	0.0599	0.0529	0.0603	0.0468
GSE9006	0.0404	0.0672	0.0397	0.0798
GSE48350	0.0000	0.0000	0.0000	0.0016
GSE5281	0.0000	0.0083	0.0002	0.0064
GSE35978	0.0385	0.0370	0.0360	0.0433
GSE53987	0.0564	0.0642	0.0508	0.0481
GSE12288	0.0279	0.0165	0.0402	0.0459
GSE15852	0.0600	0.0484	0.0486	0.0379
GSE42568	0.0167	0.0259	0.0167	0.0268
GSE29431	0.0000	0.0000	0.0000	0.0008
GSE18520	0.0000	0.0000	0.0000	0.0042
GSE19804	0.0173	0.0255	0.0173	0.0131
GSE10072	0.0000	0.0004	0.0000	0.0004
GSE68571	0.0000	0.0000	0.0000	0.0027
Average $\pm$ SEM	0.0224 ± 0.0043	0.0263 ± 0.0043	0.0231 ± 0.0043	0.0274 ± 0.0045

Table 6: Experiment 1a: Expected calibration errors (ECEs) for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with lower values of ECEs are better calibrated for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	C4.5	RIPPER	PART	BRL.G
GSE66360	1.0000	0.9877	0.9977	0.9953
GSE62646	0.3000	0.3037	0.3000	0.3096
GSE41861	0.9988	0.9813	0.9940	0.9926
GSE20881	0.9978	0.9835	0.9974	0.9988
GSE3365	0.8919	0.8985	0.8935	0.7581
GSE16879	0.2000	0.3250	0.2000	0.2135
GSE15245	0.9042	0.9934	0.9042	0.7487
GSE6613	1.0000	0.9894	0.9977	0.9456
GSE20295	1.0000	0.9937	0.9917	0.9990
GSE30999	0.6026	0.3166	0.7013	0.4987
GSE55447	0.8400	0.9692	0.8900	0.9743
GSE19429	0.6452	0.6006	0.6481	0.4968
GSE9006	0.8053	0.9736	0.7955	0.8316
GSE48350	0.0000	0.0000	0.0000	0.0109
GSE5281	0.9667	0.8897	0.9970	0.9982
GSE35978	0.8729	0.8425	0.8041	0.7997
GSE53987	0.9783	0.9650	0.9213	0.8011
GSE12288	0.9763	0.8948	0.9750	0.7927
GSE15852	0.9004	0.8914	0.8978	0.9423
GSE42568	0.2084	0.4032	0.2084	0.7815
GSE29431	0.2000	0.3008	0.2000	0.2151
GSE18520	0.1000	0.1000	0.1000	0.1211
GSE19804	0.9001	0.7001	0.8982	0.9866
GSE10072	0.4113	0.5063	0.4113	0.4289
GSE68571	0.1000	0.1000	0.1000	0.1211
Average ± SEM	0.6720 ± 0.0716	0.6764 ± 0.0686	0.6730 ± 0.0712	0.6705 ± 0.0668

Table 7: Experiment 1a: Maximum calibration errors (MCEs) for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with lower values of MCEs are better calibrated for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	C4.5	RIPPER	PART	BRL.G
GSE66360	4.3	3.0	2.4	10.4
GSE62646	2.0	2.0	2.0	2.0
GSE41861	7.8	4.1	2.2	37.6
GSE20881	10.0	4.8	2.4	86.4
GSE3365	4.8	3.1	2.3	14.4
GSE16879	2.0	2.0	2.0	2.0
GSE15245	4.5	3.8	2.2	10.2
GSE6613	9.5	5.0	2.7	13.2
GSE20295	9.8	4.9	3.7	103.0
GSE30999	3.5	2.2	3.2	3.6
GSE55447	3.9	2.3	2.4	5.7
GSE19429	5.3	2.4	2.3	12.4
GSE9006	3.8	2.8	3.0	8.4
GSE48350	2.0	2.0	2.0	2.0
GSE5281	6.1	4.5	2.3	25.6
GSE35978	19.8	7.1	8.0	31.2
GSE53987	11.6	4.0	4.7	27.2
GSE12288	17.4	8.8	6.9	27.2
GSE15852	4.9	3.3	3.1	8.3
GSE42568	2.0	2.2	2.0	3.0
GSE29431	2.0	2.0	2.0	2.0
GSE18520	2.0	2.0	2.0	2.0
GSE19804	3.0	2.7	2.1	6.2
GSE10072	2.0	2.2	2.0	3.2
GSE68571	2.0	2.0	2.0	2.0
Average $\pm$ SEM	5.84 ± 0.96	3.41 ± 0.35	2.88 ± 0.31	17.97 ± 5.09

Table 8: Experiment 1a: Average number of rules (NRs) for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with lower values of NRs are more succinct, and perhaps more readable for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	C4.5	RIPPER	PART	BRL.G
GSE66360	3.3	3.2	3.3	3.1
GSE62646	1.0	1.0	1.0	1.0
GSE41861	6.8	5.2	7.0	4.3
GSE20881	8.8	6.9	9.0	5.4
GSE3365	3.8	3.5	4.0	3.2
GSE16879	1.0	1.0	1.0	1.0
GSE15245	2.4	3.1	2.5	2.4
GSE6613	8.4	6.6	9.4	3.3
GSE20295	6.7	5.6	7.3	3.5
GSE30999	2.5	1.2	2.5	1.5
GSE55447	2.6	1.8	3.0	1.8
GSE19429	4.3	2.9	4.6	3.4
GSE9006	2.8	2.7	2.9	3.0
GSE48350	1.0	1.0	1.0	1.0
GSE5281	5.1	4.7	5.1	3.8
GSE35978	17.9	15.0	32.8	4.1
GSE53987	10.3	12.2	15.0	4.3
GSE12288	16.3	13.3	29.8	4.3
GSE15852	3.9	3.5	4.0	2.7
GSE42568	1.0	1.2	1.0	1.4
GSE29431	1.0	1.0	1.0	1.0
GSE18520	1.0	1.0	1.0	1.0
GSE19804	1.9	1.7	1.9	2.4
GSE10072	1.0	1.3	1.0	1.6
GSE68571	1.0	1.0	1.0	1.0
Average $\pm$ SEM	4.63 ± 0.93	4.06 ± 0.80	6.08 ± 1.67	2.62 ± 0.27

Table 9: Experiment 1a: Average number of variables (NVs) for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with lower values of NVs are more parsimonious, and perhaps easier to validate for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

test (p-value = $2.087e-06$), Friedman's test with Iman-Davenport correction (p-value = $1.906e{-}08$), Friedman's aligned ranks test (p-value = $2.761e{-}07$), and Quade test (p-value = $1.412e-10$) all strongly suggest that we must reject the null hypothesis suggesting that all classifiers have the same Bayesian scores. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were corrected using Bergmann and Hommel's method. Each of the pairwise corrected p-values are shown in Figure 12a. Each of BRL.G, BRL.DT, and BRL.DG were found to have statistically significantly different Bayesian scores from each other. BRL.DG, on average, achieves the highest Bayesian score, followed by BRL.DT and then BRL.G.

AUROCs achieved by the three classifiers are shown in Table 11. Friedman's test (p-value = 0.01317), Friedman's test with Iman-Davenport correction (p-value = 0.01041), Friedman's aligned ranks test (p-value = 0.005371), and Quade test (p-value = 0.001472) all suggest that we must reject the null hypothesis suggesting that all classifiers have the same AUROCs. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were corrected using Bergmann and Hommel's method. Each of the pairwise corrected p-values are shown in Figure 12b. BRL.DG is found to have significantly worse AUROC when compared to BRL.G and BRL.DT.

AUPRGs achieved by the three classifiers are shown in Table 12. Friedman's test (p-value = 0.01317), Friedman's test with Iman-Davenport correction (p-value = 0.01041), Friedman's aligned ranks test (p-value = 0.006131), and Quade test (p-value = 0.001409) all suggest that we must reject the null hypothesis suggesting that all classifiers have the same AUPRGs. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were corrected using Bergmann and Hommel's method. Each of the pairwise corrected p-values are shown in Figure 12c. BRL.DG is found to have significantly worse AUPRG when compared to BRL.G and BRL.DT.

The next three metrics— Brier score, ECE, and MCE— help evaluate the calibration performance.

Brier achieved by the three classifiers are shown in Table 13. Friedman's test (p-value = 0.2894), Friedman's test with Iman-Davenport correction (p-value = 0.295), Friedman's

aligned ranks test (p-value = 0.2222), and Quade test (p-value = 0.1193) all suggest that we cannot reject the null hypothesis suggesting that all classifiers have the same Brier scores. BRL.G, BRL.DT, and BRL.DG were all found to have comparable Brier scores.

ECEs achieved by the three classifiers are shown in Table 14. Friedman's test (p-value = 0.2894), Friedman's test with Iman-Davenport correction (p-value = 0.295), Friedman's aligned ranks test (p-value = 0.3333), and Quade test (p-value = 0.2748) all suggest that we cannot reject the null hypothesis suggesting that all classifiers have the same ECEs. BRL.G, BRL.DT, and BRL.DG were all found to have comparable ECEs.

MCEs achieved by the three classifiers are shown in Table 15. Friedman's test (p-value = 0.6977), Friedman's test with Iman-Davenport correction (p-value = 0.706), Friedman's aligned ranks test (p-value = 0.4795), and Quade test (p-value = 0.4639) all suggest that we cannot reject the null hypothesis suggesting that all classifiers have the same MCEs. BRL.G, BRL.DT, and BRL.DG were all found to have comparable MCEs.

The next two metrics— average number of rules (NR) and average number of variables (NV)— help evaluate model parsimony.

Average NR needed by the different classifiers is shown in Table 16. Friedman's test (p-value = $1.53e-06$), Friedman's test with Iman-Davenport correction (p-value = $1.013e-08$), Friedman's aligned ranks test (p-value = $1.121e-07$), and Quade test (p-value = $1.751e-11$) all strongly suggest that we must reject the null hypothesis suggesting that all classifiers need the same number of rules. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were corrected using Bergmann and Hommel's method. Each of the pairwise corrected p-values are shown in Figure 12d. Each of BRL.G, BRL.DT, and BRL.DG were found to learn statistically significantly different number of rules from each other. BRL.DG requires the fewest rules, followed by BRL.DT and then BRL.G.

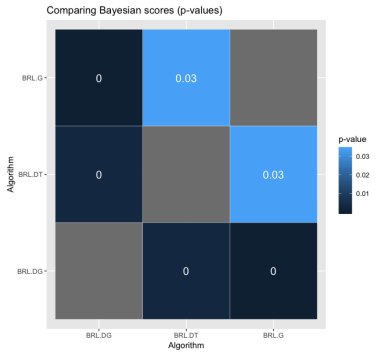
Average NV needed by the different classifiers is shown in Table 17. Friedman's test (p-value = 0.0008169), Friedman's test with Iman-Davenport correction (p-value = 0.0003252), Friedman's aligned ranks test (p-value = $3.054e-06$), and Quade test (p-value = $2.993e-06$) all strongly suggest that we must reject the null hypothesis suggesting that all classifiers need the same number of variables. Post-hoc test is now conducted to determine

which of the classifiers are different from others. The p-values generated by Friedman's test were corrected using Bergmann and Hommel's method. Each of the pairwise corrected p-values are shown in Figure 12e. BRL.DT and BRL.DG learn similar number of variables but BRL.G requires significantly fewer number of variables.

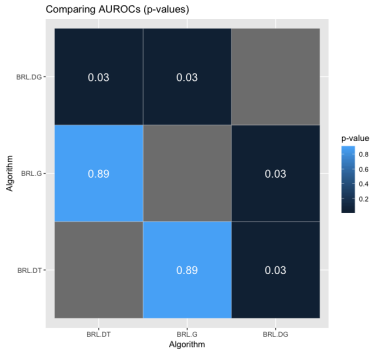
4.3.2.1 Experiment 1b: Conclusion Experiment 1b shows that BRL.G and BRL.DT achieves a statistically significantly better predictive performance than BRL.DG in terms of AUROC and AUPRG. BRL.G and BRL.DT themselves were indistinguishable. Calibration performance using Brier score, ECE, and MCE showed that all BRL methods have statistically indistinguishable performance. In terms of model parsimony, BRL.G requires significantly more rules but requires significantly fewer variables than BRL.DT and BRL.DG. BRL.DG required significantly the fewest number of rules.

To summarize, BRL.DT presents an efficient alternative to BRL.G by achieving similar predictive and calibration performance as BRL.G but requiring much fewer rules. However, BRL.G still requires much fewer variables. If the application requires fewer rules for readability/validation, I recommend BRL.DT. If the application requires fewer variables, I recommend BRL.G.

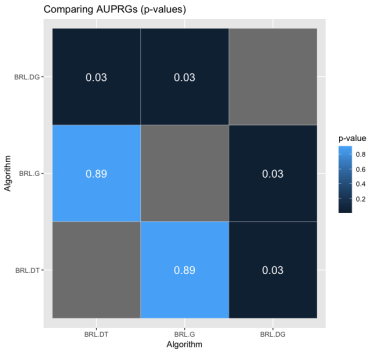
BRL.DG using greedy best-first search does not give a good predictive performance when compared to BRL.G and BRL.DT. Although, it requires the fewest rules. BRL.DG has important benefits. Rules there represent context-specific independences. Such rules are parsimonious in the sense the sub-population represented by the leaf does not depend on all variables in the path from root to leaf. It should also be noted that BRL.DG searches over a much larger model search space than BRL.DT, which itself searches over a much larger space than BRL.G. Perhaps greedy best-first search is very limiting to BRL.DG and warrants a better search algorithm. In the next experiment, I expand the greedy best-first search to beam search and hope to improve the predictive performance of BRL.DG.



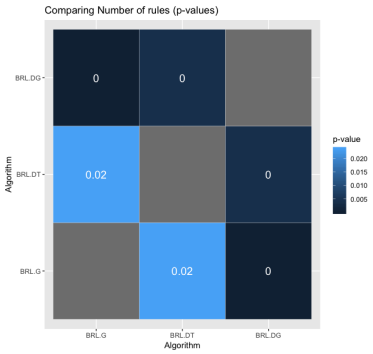
(a) Comparing Bayesian scores



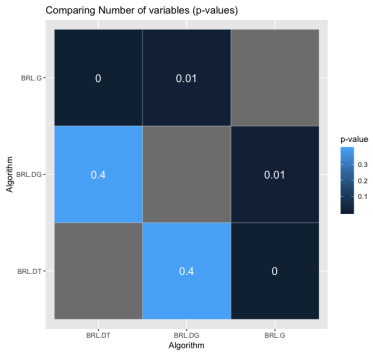
(b) Comparing AUROCs



(c) Comparing AUPRGs



(d) Comparing number of rules



(e) Comparing number of variables

Figure 12: Experiment 1b: Corrected p-values using Bergmann and Hommel's method while comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first approach.

4.3.3 Experiment 1c: Comparing BRL classifiers using beam search

In Experiment 1c, I extend greedy best-first search to beam search that keeps memory of 100 best models seen in each iteration. Greedy best-first search only keeps track of 1 best model in each iteration. This should help the BRL algorithms to search over a much larger space that may lead to BRL.DG learning models with better predictive performance than when using greedy best-first search. To distinguish tthe two search procedures, I add a suffix "-

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	-23.01	-22.97	-21.59
GSE62646	-7.47	-7.47	-7.47
GSE41861	-47.33	-39.18	-33.39
GSE20881	-57.76	-49.92	-42.62
GSE3365	-26.41	-26.76	-22.72
GSE16879	-7.63	-7.63	-7.63
GSE15245	-19.75	-19.03	-18.04
GSE6613	-55.32	-50.46	-43.99
GSE20295	-49.42	-39.15	-32.22
GSE30999	-15.97	-15.05	-14.96
GSE55447	-14.59	-14.27	-13.65
GSE19429	-24.58	-22.12	-20.61
GSE9006	-20.72	-20.87	-19.54
GSE48350	-7.73	-7.73	-7.73
GSE5281	-40.47	-33.09	-29.32
GSE35978	-147.63	-135.53	-123.93
GSE53987	-95.87	-89.36	-84.09
GSE12288	-123.56	-115.81	-106.12
GSE15852	-24.85	-23.89	-21.92
GSE42568	-11.31	-10.99	-10.72
GSE29431	-7.60	-7.60	-7.60
GSE18520	-7.55	-7.55	-7.55
GSE19804	-17.17	-16.68	-16.08
GSE10072	-11.00	-10.59	-10.37
GSE68571	-7.67	-7.67	-7.67
Average $\pm$ SEM	-34.89 ± 7.43	-32.05 ± 6.79	-29.26 ± 6.18

Table 10: Experiment 1b: Log of Bayesian score for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Higher Bayesian scores indicate more optimal models in the model search space. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	0.8640	0.8490	0.8435
GSE62646	0.9083	0.9083	0.9083
GSE41861	0.7401	0.7717	0.7300
GSE20881	0.8117	0.8649	0.7895
GSE3365	0.9311	0.9174	0.8193
GSE16879	0.9845	0.9845	0.9845
GSE15245	0.6950	0.7150	0.7150
GSE6613	0.4563	0.4697	0.4987
GSE20295	0.5750	0.6221	0.6338
GSE30999	0.9722	0.9764	0.9535
GSE55447	0.6175	0.5325	0.4825
GSE19429	0.9157	0.9001	0.7529
GSE9006	0.8458	0.8658	0.7883
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.8436	0.8801	0.8136
GSE35978	0.6013	0.5436	0.5295
GSE53987	0.5581	0.5564	0.5483
GSE12288	0.5669	0.5723	0.5519
GSE15852	0.8569	0.8194	0.8075
GSE42568	0.8109	0.8109	0.8109
GSE29431	0.9417	0.9417	0.9417
GSE18520	0.9900	0.9900	0.9900
GSE19804	0.9153	0.9083	0.9083
GSE10072	0.9425	0.9425	0.9225
GSE68571	0.9938	0.9938	0.9938
Average $\pm$ SEM	0.8135 ± 0.0329	0.8135 ± 0.0336	0.7887 ± 0.0332

Table 11: Experiment 1b: Area under the ROC cruves (AUROCs) for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Classifier with higher values of AUROCs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	0.7558	0.7234	0.7167
GSE62646	0.7833	0.7833	0.7833
GSE41861	0.3882	0.4126	0.3982
GSE20881	0.6381	0.7311	0.5785
GSE3365	0.8003	0.7627	0.6226
GSE16879	0.8845	0.8845	0.8845
GSE15245	0.3119	0.3070	0.3070
GSE6613	-0.0736	-0.0205	0.0118
GSE20295	0.2157	0.3450	0.3519
GSE30999	0.9548	0.9599	0.9141
GSE55447	0.1175	-0.0075	-0.0200
GSE19429	0.4804	0.6025	0.3636
GSE9006	0.6300	0.6413	0.5035
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.6881	0.7763	0.6360
GSE35978	0.1944	0.0948	0.0835
GSE53987	0.1127	0.0819	0.0661
GSE12288	0.1829	0.1860	0.1459
GSE15852	0.7579	0.6777	0.6526
GSE42568	0.5631	0.5631	0.5631
GSE29431	0.8417	0.8417	0.8417
GSE18520	0.9400	0.9400	0.9400
GSE19804	0.8435	0.8420	0.8420
GSE10072	0.8920	0.8920	0.8520
GSE68571	0.9438	0.9438	0.9438
Average $\pm$ SEM	0.5939 ± 0.0644	0.5986 ± 0.0655	0.5593 ± 0.0646

Table 12: Experiment 1b: Area under precision-recall gain curves (AUPRGs) for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Classifier with higher values of AUPRGs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	0.2169	0.1675	0.1686
GSE62646	0.0684	0.0684	0.0684
GSE41861	0.2339	0.2492	0.2538
GSE20881	0.2007	0.2059	0.2154
GSE3365	0.1039	0.1642	0.1482
GSE16879	0.0257	0.0257	0.0257
GSE15245	0.2235	0.1910	0.1922
GSE6613	0.4136	0.4566	0.4503
GSE20295	0.3568	0.3234	0.3287
GSE30999	0.0497	0.0552	0.0576
GSE55447	0.2805	0.2943	0.2973
GSE19429	0.0824	0.0708	0.0717
GSE9006	0.2125	0.2111	0.2085
GSE48350	0.0001	0.0001	0.0001
GSE5281	0.1621	0.1717	0.1731
GSE35978	0.2577	0.3237	0.3309
GSE53987	0.2401	0.2711	0.2759
GSE12288	0.3149	0.3319	0.3566
GSE15852	0.1668	0.1929	0.1885
GSE42568	0.0806	0.0650	0.0649
GSE29431	0.0279	0.0279	0.0279
GSE18520	0.0159	0.0159	0.0159
GSE19804	0.1092	0.0992	0.0984
GSE10072	0.0552	0.0726	0.0761
GSE68571	0.0106	0.0106	0.0106
Average $\pm$ SEM	0.1564 ± 0.0234	0.1626 ± 0.0248	0.1642 ± 0.0252

Table 13: Experiment 1b: Brier scores for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Classifier with lower values of Brier score are better calibrated for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	0.0265	0.0168	0.0169
GSE62646	0.0289	0.0289	0.0289
GSE41861	0.0281	0.0302	0.0284
GSE20881	0.0119	0.0173	0.0173
GSE3365	0.0130	0.0241	0.0244
GSE16879	0.0030	0.0030	0.0030
GSE15245	0.0535	0.0289	0.0288
GSE6613	0.0644	0.0608	0.0533
GSE20295	0.0507	0.0225	0.0224
GSE30999	0.0060	0.0060	0.0060
GSE55447	0.0408	0.0188	0.0188
GSE19429	0.0468	0.0354	0.0365
GSE9006	0.0798	0.0819	0.0700
GSE48350	0.0016	0.0016	0.0016
GSE5281	0.0064	0.0064	0.0064
GSE35978	0.0433	0.0419	0.0421
GSE53987	0.0481	0.0585	0.0590
GSE12288	0.0459	0.0272	0.0347
GSE15852	0.0379	0.0368	0.0422
GSE42568	0.0268	0.0260	0.0261
GSE29431	0.0008	0.0008	0.0008
GSE18520	0.0042	0.0042	0.0042
GSE19804	0.0131	0.0172	0.0168
GSE10072	0.0004	0.0130	0.0140
GSE68571	0.0027	0.0027	0.0027
Average $\pm$ SEM	0.0274 ± 0.0045	0.0244 ± 0.0041	0.0242 ± 0.0037

Table 14: Experiment 1b: Expected calibration errors (ECEs) for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Classifier with lower values of ECEs are better calibrated for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	0.9953	0.8840	0.8796
GSE62646	0.3096	0.3096	0.3096
GSE41861	0.9926	0.9979	0.9973
GSE20881	0.9988	0.9986	0.9977
GSE3365	0.7581	0.8828	0.8982
GSE16879	0.2135	0.2135	0.2135
GSE15245	0.7487	0.7891	0.7947
GSE6613	0.9456	0.9920	0.9958
GSE20295	0.9990	0.9976	0.9982
GSE30999	0.4987	0.6697	0.6862
GSE55447	0.9743	0.9854	0.9886
GSE19429	0.4968	0.4447	0.4525
GSE9006	0.8316	0.9418	0.8600
GSE48350	0.0109	0.0109	0.0109
GSE5281	0.9982	0.9969	0.9978
GSE35978	0.7997	0.8063	0.8212
GSE53987	0.8011	0.8838	0.8742
GSE12288	0.7927	0.8926	0.8910
GSE15852	0.9423	0.9032	0.8929
GSE42568	0.7815	0.5946	0.5951
GSE29431	0.2151	0.2151	0.2151
GSE18520	0.1211	0.1211	0.1211
GSE19804	0.9866	0.8899	0.8850
GSE10072	0.4289	0.6289	0.6485
GSE68571	0.1211	0.1211	0.1211
Average $\pm$ SEM	0.6705 ± 0.0668	0.6868 ± 0.0666	0.6858 ± 0.0661

Table 15: Experiment 1b: Maximum calibration errors (MCEs) for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Classifier with lower values of MCEs are better calibrated for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	10.4	5.4	3.2
GSE62646	2.0	2.0	2.0
GSE41861	37.6	8.2	2.7
GSE20881	86.4	10.7	2.8
GSE3365	14.4	6.9	2.7
GSE16879	2.0	2.0	2.0
GSE15245	10.2	4.8	3.2
GSE6613	13.2	7.8	2.9
GSE20295	103.0	14.2	7.5
GSE30999	3.6	3.3	3.0
GSE55447	5.7	3.9	2.9
GSE19429	12.4	5.7	2.7
GSE9006	8.4	4.3	2.4
GSE48350	2.0	2.0	2.0
GSE5281	25.6	9.1	3.3
GSE35978	31.2	11.5	4.2
GSE53987	27.2	8.4	2.9
GSE12288	27.2	8.9	3.4
GSE15852	8.3	5.2	2.3
GSE42568	3.0	3.2	2.6
GSE29431	2.0	2.0	2.0
GSE18520	2.0	2.0	2.0
GSE19804	6.2	4.1	2.8
GSE10072	3.2	3.2	2.6
GSE68571	2.0	2.0	2.0
Average $\pm$ SEM	17.97 ± 5.09	5.63 ± 0.68	2.88 ± 0.22

Table 16: Experiment 1b: Average number of rules (NRs) for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Classifier with lower values of NRs are more succinct, and perhaps more readable for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	3.1	3.4	3.4
GSE62646	1.0	1.0	1.0
GSE41861	4.3	5.9	6.0
GSE20881	5.4	8.2	8.1
GSE3365	3.2	4.3	4.2
GSE16879	1.0	1.0	1.0
GSE15245	2.4	2.4	2.4
GSE6613	3.3	5.9	6.5
GSE20295	3.5	5.4	5.2
GSE30999	1.5	1.9	1.9
GSE55447	1.8	1.7	1.7
GSE19429	3.4	3.7	3.6
GSE9006	3.0	3.2	3.1
GSE48350	1.0	1.0	1.0
GSE5281	3.8	5.6	5.5
GSE35978	4.1	9.2	9.0
GSE53987	4.3	7.1	6.9
GSE12288	4.3	7.4	7.8
GSE15852	2.7	3.8	3.7
GSE42568	1.4	1.5	1.5
GSE29431	1.0	1.0	1.0
GSE18520	1.0	1.0	1.0
GSE19804	2.4	2.5	2.5
GSE10072	1.6	1.6	1.6
GSE68571	1.0	1.0	1.0
Average $\pm$ SEM	2.62 ± 0.27	3.63 ± 0.51	3.62 ± 0.51

Table 17: Experiment 1b: Average number of variables (NVs) for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Classifier with lower values of NVs are more parsimonious, and perhaps easier to validate for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Beam“ to the algorithm names. While BRL.G, BRL.DT, and BRL.DG are optimized using greedy best-first search; BRL.G-Beam, BRL.DT-Beam, and BRL.DG-Beam are optimized using beam search.

To see the Accuracy, Precision, Recall, and F-measures achieved by the classifiers over the 25 datasets, see Appendix C (9.3).

To compare the model predictive performance of the different BRL algorithms, the metrics Bayesian score, AUROC, AUPRG, Brier score, ECE, and MCE are compared. Significance testing is only performed on these predictive performance metrics.

Bayesian scores achieved by the three classifiers are shown in Table 18. Using Wilcoxon signed ranks test, we can see that the Bayesian scores achieved by— BRL.G-Beam is better than BRL.G (p-value $= 1.83e-05$), BRL.DT-Beam is better than BRL.DT (p-value $= 1.234e-05$), and BRL.DG-Beam is better than BRL.DG (p-value $= 0.0003291$). It is clear that each of the BRL methods benefitted from the expanded search space. Friedman's test (p-value $= 4.644e-06$), Friedman's test with Iman-Davenport correction (p-value $= 9.06e-08$), Friedman's aligned ranks test (p-value $= 3.597e-06$), and Quade test (p-value $= 1.913e-08$) all strongly suggest that we must reject the null hypothesis suggesting that all classifiers have the same Bayesian scores. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were corrected using Bergmann and Hommel‘s method. Each of the pairwise corrected p-values are shown in Figure 13a. Each of BRL.G-Beam, BRL.DT-Beam, and BRL.DG-Beam were found to have statistically significantly different Bayesian scores from each other. BRL.DG-Beam, on average, achieves the highest Bayesian score, followed by BRL.DT-Beam and then BRL.G-Beam.

AUROCs achieved by the three classifiers are shown in Table 19. Using Wilcoxon signed ranks test, we can see that the AUROCs achieved by— BRL.G-Beam is similar to BRL.G (p-value $= 0.2156$), BRL.DT-Beam is similar to BRL.DT (p-value $= 0.9143$), and BRL.DG-Beam is similar to BRL.DG (p-value $= 0.3065$). While not significantly different from best-first search, beam search slightly deteriorates the performance of BRL.G but slightly improves the performance of BRL.DT and BRL.DG. Friedman's test (p-value $= 0.06522$), Friedman's test with Iman-Davenport correction (p-value $= 0.06233$), Friedman's aligned

ranks test (p-value = 0.0146), and Quade test (p-value = 0.07982) all but Friedman's aligned ranks test suggest that we must not reject the null hypothesis suggesting that all classifiers have the same AUROCs. Continuing with the result from Friedman's aligned ranks test, post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were corrected using Bergmann and Hommel's method. Each of the pairwise corrected p-values are shown in Figure 13b. BRL.DG-Beam is found to have significantly better AUROC when compared to BRL.G-Beam. BRL.DT-Beam is found to have significantly better AUROC when compared to both BRL.G-Beam and BRL.DG-Beam.

AUPRGs achieved by the three classifiers are shown in Table 20. Using Wilcoxon signed ranks test, we can see that the AUPRGs achieved by— BRL.G-Beam is similar to BRL.G (p-value = 0.3696), BRL.DT-Beam is similar to BRL.DT (p-value = 0.6158), but BRL.DG-Beam is significantly better than BRL.DG (p-value = 0.03453). It appears like BRL-DG benefited a lot in terms of AUPRG using beam search. Friedman's test (p-value = 0.1791), Friedman's test with Iman-Davenport correction (p-value = 0.1807), Friedman's aligned ranks test (p-value = 0.07778), and Quade test (p-value = 0.1544) all suggest that we cannot reject the null hypothesis suggesting that all classifiers have the same AUPRGs. All of BRL.G-Beam, BRL.DT-Beam, and BRT.DG-Beam appear to now have comparable AUPRG. Again, the beam search has helped improve the performance of BRL.DG.

The next three metrics— Brier score, ECE, and MCE— help evaluate the calibration performance.

Brier achieved by the three classifiers are shown in Table 21. Using Wilcoxon signed ranks test, we can see that the Brier scores achieved by— BRL.G-Beam is similar to BRL.G (p-value = 0.3696), BRL.DT-Beam is similar to BRL.DT (p-value = 0.6158), and BRL.DG-Beam is significantly better than BRL.DG (p-value = 0.03453). Friedman's test (p-value = 0.1409), Friedman's test with Iman-Davenport correction (p-value = 0.1409), Friedman's aligned ranks test (p-value = 0.08571), and Quade test (p-value = 0.06874) all suggest that we cannot reject the null hypothesis suggesting that all classifiers have the same Brier scores. BRL.DG significantly benefits from beam search, in terms of the Brier score.

ECEs achieved by the three classifiers are shown in Table 22. Using Wilcoxon signed

ranks test, we can see that the ECEs achieved by— BRL.G-Beam is similar to BRL.G (p-value = 0.1135), BRL.DT-Beam is similar to BRL.DT (p-value = 0.5965), and BRL.DG-Beam is similar to BRL.DG (p-value = 0.1645). Friedman's test (p-value = 0.8869), Friedman's test with Iman-Davenport correction (p-value = 0.8909), Friedman's aligned ranks test (p-value = 0.6445), and Quade test (p-value = 0.9771) all suggest that we cannot reject the null hypothesis suggesting that all classifiers have the same ECEs.

MCEs achieved by the three classifiers are shown in Table 23. Using Wilcoxon signed ranks test, we can see that the MCEs achieved by— BRL.G-Beam is similar to BRL.G (p-value = 0.9785), BRL.DT-Beam is similar to BRL.DT (p-value = 0.1965), and BRL.DG-Beam is similar to BRL.DG (p-value = 0.1965). Friedman's test (p-value = 0.6188), Friedman's test with Iman-Davenport correction (p-value = 0.628), Friedman's aligned ranks test (p-value = 0.651), and Quade test (p-value = 0.4916) all suggest that we cannot reject the null hypothesis suggesting that all classifiers have the same MCEs.

The next two metrics— average number of rules (NR) and average number of variables (NV)— help evaluate model parsimony.

Average NR needed by the different classifiers is shown in Table 24. Using Wilcoxon signed ranks test, we can see that the number of rules needed by— BRL.G-Beam is similar to BRL.G (p-value = 0.5523), BRL.DT-Beam requires fewer rules than BRL.DT (p-value = 0.03796), and BRL.DG-Beam is similar to BRL.DG (p-value = 0.05691). Friedman's test (p-value = $1.625e{-}06$), Friedman's test with Iman-Davenport correction (p-value = $1.146e{-}08$), Friedman's aligned ranks test (p-value = $8.296e-08$), and Quade test (p-value = $1.591e-11$) all strongly suggest that we must reject the null hypothesis suggesting that all classifiers need the same number of rules. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were corrected using Bergmann and Hommel's method. Each of the pairwise corrected p-values are shown in Figure 13c. Each of BRL.G-Beam, BRL.DT-Beam, and BRL.DG-Beam were found to learn statistically significantly different number of rules from each other. BRL.DG-Beam requires the fewest rules, followed by BRL.DT-Beam and then BRL.G-Beam.

Average NV needed by the different classifiers is shown in Table 25. Using Wilcoxon signed ranks test, we can see that the number of variables needed by— BRL.G-Beam is sim-

ilar to BRL.G (p-value = 0.1431), BRL.DT-Beam is similar to BRL.DT (p-value = 0.1655), while BRL.DG-Beam requires seignificantly fewer variables than BRL.DG (p-value = 0.007125). Friedman's test (p-value = 0.08982) and Friedman's test with Iman-Davenport correction (p-value = 0.08779) do not recommend rejecting the null hypothesis. Whereas, Friedman's aligned ranks test (p-value = 0.02686), and Quade test (p-value = 0.009083) suggest that we reject the null hypothesis suggesting that all classifiers need the same number of variables. Continuing with the results from Friedman's aligned ranks test and Quade test, post-hoc test is conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's aligned ranks test were corrected using Bergmann and Hommel's method. Each of the pairwise corrected p-values are shown in Figure 13d. BRL.G-Beam requires significantly fewer variables than BRL.DT but similar to BRL.DG.

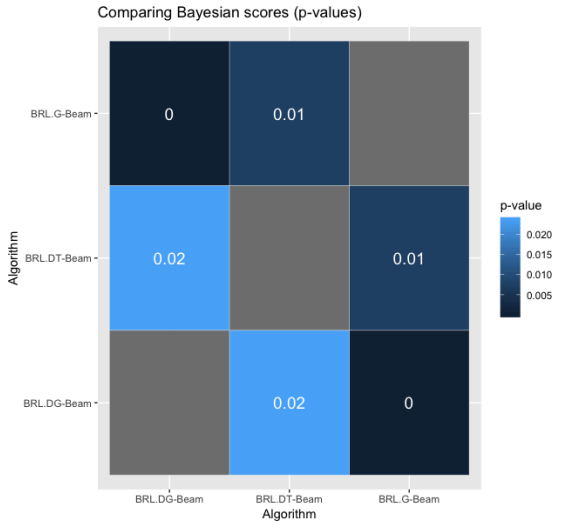
4.3.3.1 Experiment 1c: Conclusion

Experiment 1c shows how expanding the search space using beam search has significant benefits, especially for BRL.DG which had a much larger search space than BRL.G. It benefits in terms of predictive performance, calibration, and model parsimony. Using beam search, it appears like BRL.G-Beam, BRL.DT-Beam, and BRL.DG-Beam all have similar predictive performances. BRL.DG requiring much fewer rules than others and requires similar number of variables as BRL.G.

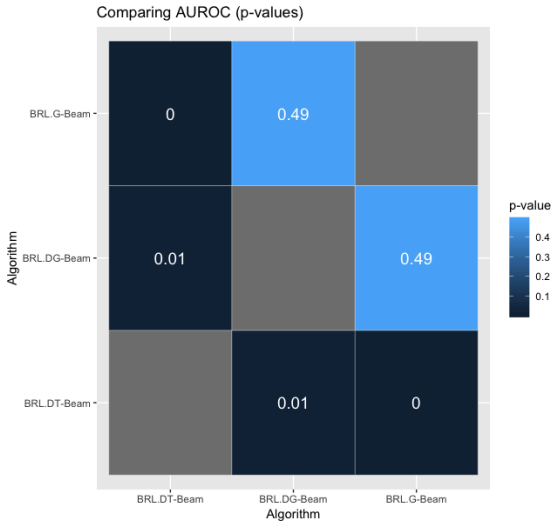
To summarize, BRL.DG-Beam has now emerged as a competent alternative to BRL.G with similar predictive performance, calibration, and the number of variables, while requiring significantly fewer rules. With an added benefit of the rules capturing all types of context-specific independence, BRL.DG-Beam is now an attractive model for biomarker discovery.

4.3.4 BRL compared to other state-of-the-art classifiers

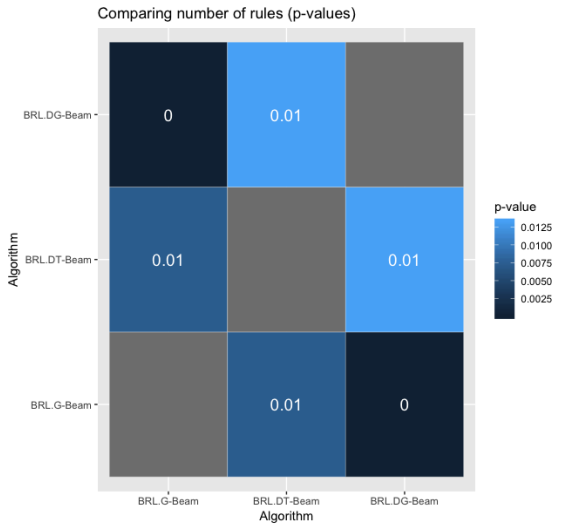
In this section, I compare the AUROC, AUPRG, and Brier scores achieved by state-of-the-art classifiers and the best performing BRL mode, BRL.DT and BRL.DT-Beam. The complete table containing the metrics for each dataset is shown in Appendix C (see 9.4). Instead, in this section, only the average (and standard error) of the metric values, across the 25 datasets, is shown in a bar plot in Figure 14.



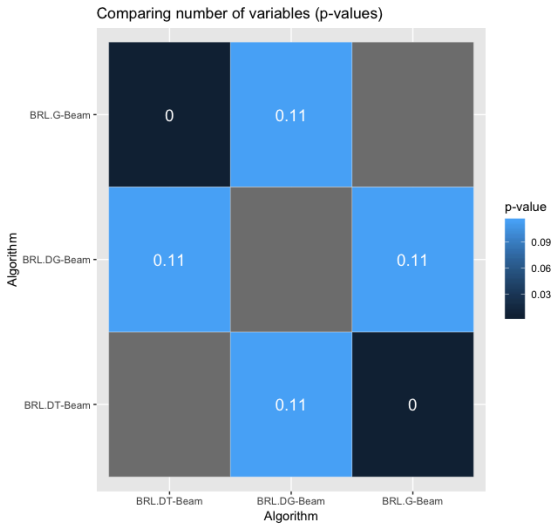
(a) Comparing Bayesian scores



(b) Comparing AUROCs



(c) Comparing number of rules



(d) Comparing number of variables

Figure 13: Experiment 1c: Corrected p-values using Bergmann and Hommel's method while comparing BRL.G-Beam, BRL.DT-Beam, and BRL.DG-Beam that use greedy beam search.

The AUROCs (see Figure 14a) and AUPRGs (see Figure 14b) by BRL.DT and BRL.DT-Beam is better than rule learning SOTA classifiers (C4.5, RIPPER, and PART). However,

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	-23.01	-19.76	-22.97	-18.08	-21.59	-17.02
GSE62646	-7.47	-7.47	-7.47	-7.47	-7.47	-7.47
GSE41861	-47.33	-37.28	-39.18	-31.86	-33.39	-28.96
GSE20881	-57.76	-48.16	-49.92	-38.17	-42.62	-33.93
GSE3365	-26.41	-20.66	-26.76	-18.97	-22.72	-17.48
GSE16879	-7.63	-7.63	-7.63	-7.63	-7.63	-7.63
GSE15245	-19.75	-18.03	-19.03	-17.23	-18.04	-16.18
GSE6613	-55.32	-49.90	-50.46	-40.74	-43.99	-38.90
GSE20295	-49.42	-38.72	-39.15	-35.33	-32.22	-58.14
GSE30999	-15.97	-15.97	-15.05	-15.05	-14.96	-14.64
GSE55447	-14.59	-12.89	-14.27	-12.29	-13.65	-11.79
GSE19429	-24.58	-20.50	-22.12	-18.02	-20.61	-16.83
GSE9006	-20.72	-20.11	-20.87	-18.95	-19.54	-18.01
GSE48350	-7.73	-7.73	-7.73	-7.73	-7.73	-7.73
GSE5281	-40.47	-30.85	-33.09	-25.70	-29.32	-23.90
GSE35978	-147.63	-143.51	-135.53	-119.99	-123.93	-116.04
GSE53987	-95.87	-93.89	-89.36	-86.17	-84.09	-81.89
GSE12288	-123.56	-120.58	-115.81	-107.22	-106.12	-101.46
GSE15852	-24.85	-20.40	-23.89	-18.85	-21.92	-17.82
GSE42568	-11.31	-11.11	-10.99	-10.91	-10.72	-10.70
GSE29431	-7.60	-7.60	-7.60	-7.60	-7.60	-7.60
GSE18520	-7.55	-7.55	-7.55	-7.55	-7.55	-7.55
GSE19804	-17.17	-15.00	-16.68	-14.21	-16.08	-13.90
GSE10072	-11.00	-11.00	-10.59	-10.59	-10.37	-10.37
GSE68571	-7.67	-7.67	-7.67	-7.67	-7.67	-7.67
Average $\pm$ SEM	-34.89 ± 7.43	-31.76 ± 7.19	-32.05 ± 6.79	-28.16 ± 6.15	-29.26 ± 6.18	-27.74 ± 6.00

Table 18: Experiment 1c: Log of Bayesian score for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	0.8640	0.8700	0.8490	0.9135	0.8435	0.8760
GSE62646	0.9083	0.9083	0.9083	0.9083	0.9083	0.9083
GSE41861	0.7401	0.6451	0.7717	0.7464	0.7300	0.7392
GSE20881	0.8117	0.7897	0.8649	0.8629	0.7895	0.8189
GSE3365	0.9311	0.8723	0.9174	0.9021	0.8193	0.8522
GSE16879	0.9845	0.9845	0.9845	0.9845	0.9845	0.9845
GSE15245	0.6950	0.5450	0.7150	0.6400	0.7150	0.6400
GSE6613	0.4563	0.5410	0.4697	0.4610	0.4987	0.5303
GSE20295	0.5750	0.5638	0.6221	0.6254	0.6338	0.6292
GSE30999	0.9722	0.9722	0.9764	0.9764	0.9535	0.9479
GSE55447	0.6175	0.8000	0.5325	0.8375	0.4825	0.8000
GSE19429	0.9157	0.8646	0.9001	0.8174	0.7529	0.7432
GSE9006	0.8458	0.8975	0.8658	0.8750	0.7883	0.7667
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8436	0.8463	0.8801	0.9232	0.8136	0.8547
GSE35978	0.6013	0.5763	0.5436	0.5851	0.5295	0.5735
GSE53987	0.5581	0.4409	0.5564	0.4847	0.5483	0.4820
GSE12288	0.5669	0.4526	0.5723	0.5394	0.5519	0.5741
GSE15852	0.8569	0.8431	0.8194	0.8644	0.8075	0.8600
GSE42568	0.8109	0.8159	0.8109	0.8159	0.8109	0.8405
GSE29431	0.9417	0.9417	0.9417	0.9417	0.9417	0.9417
GSE18520	0.9900	0.9900	0.9900	0.9900	0.9900	0.9900
GSE19804	0.9153	0.9264	0.9083	0.9250	0.9083	0.9042
GSE10072	0.9425	0.9425	0.9425	0.9425	0.9225	0.9025
GSE68571	0.9938	0.9937	0.9938	0.9937	0.9938	0.9937
Average ± SEM	0.8135 ± 0.0329	0.8009 ± 0.0362	0.8135 ± 0.0336	0.8222 ± 0.0336	0.7887 ± 0.0332	0.8061 ± 0.0310

Table 19: Experiment 1c: Area under the ROC cruves (AUROCs) for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. Classifier with higher values of AUROCs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	0.7558	0.7582	0.7234	0.8392	0.7167	0.7767
GSE62646	0.7833	0.7833	0.7833	0.7833	0.7833	0.7833
GSE41861	0.3882	0.3340	0.4126	0.4384	0.3982	0.4515
GSE20881	0.6381	0.6270	0.7311	0.7340	0.5785	0.6533
GSE3365	0.8003	0.6762	0.7627	0.7401	0.6226	0.6857
GSE16879	0.8845	0.8845	0.8845	0.8845	0.8845	0.8845
GSE15245	0.3119	0.1052	0.3070	0.2362	0.3070	0.2070
GSE6613	-0.0736	0.1284	-0.0205	-0.0488	0.0118	0.0927
GSE20295	0.2157	0.1528	0.3450	0.2800	0.3519	0.2993
GSE30999	0.9548	0.9548	0.9599	0.9599	0.9141	0.9030
GSE55447	0.1175	0.4500	-0.0075	0.5000	-0.0200	0.4625
GSE19429	0.4804	0.5775	0.6025	0.4856	0.3636	0.4680
GSE9006	0.6300	0.7073	0.6413	0.6607	0.5035	0.4991
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.6881	0.6688	0.7763	0.8603	0.6360	0.7325
GSE35978	0.1944	0.1513	0.0948	0.1380	0.0835	0.1404
GSE53987	0.1127	0.0069	0.0819	0.0124	0.0661	0.0144
GSE12288	0.1829	-0.0448	0.1860	0.1700	0.1459	0.2073
GSE15852	0.7579	0.7095	0.6777	0.7399	0.6526	0.7474
GSE42568	0.5631	0.5909	0.5631	0.5909	0.5631	0.6677
GSE29431	0.8417	0.8417	0.8417	0.8417	0.8417	0.8417
GSE18520	0.9400	0.9400	0.9400	0.9400	0.9400	0.9400
GSE19804	0.8435	0.8731	0.8420	0.8770	0.8420	0.8443
GSE10072	0.8920	0.8920	0.8920	0.8920	0.8520	0.8120
GSE68571	0.9438	0.9438	0.9438	0.9438	0.9438	0.9438
Average $\pm$ SEM	0.5939 ± 0.0644	0.5885 ± 0.0663	0.5986 ± 0.0655	0.6200 ± 0.0643	0.5593 ± 0.0646	0.6023 ± 0.0595

Table 20: Experiment 1c: Area under precision-recall gain curves (AUPRGs) for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. Classifier with higher values of AUPRGs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	0.2169	0.2064	0.1675	0.1409	0.1686	0.1224
GSE62646	0.0684	0.0684	0.0684	0.0684	0.0684	0.0684
GSE41861	0.2339	0.2503	0.2492	0.2294	0.2538	0.2274
GSE20881	0.2007	0.1964	0.2059	0.1786	0.2154	0.1681
GSE3365	0.1039	0.1359	0.1642	0.1200	0.1482	0.1280
GSE16879	0.0257	0.0257	0.0257	0.0257	0.0257	0.0257
GSE15245	0.2235	0.2885	0.1910	0.2308	0.1922	0.2318
GSE6613	0.4136	0.4198	0.4566	0.5334	0.4503	0.4620
GSE20295	0.3568	0.3813	0.3234	0.3707	0.3287	0.2503
GSE30999	0.0497	0.0497	0.0552	0.0552	0.0576	0.0571
GSE55447	0.2805	0.1939	0.2943	0.1737	0.2973	0.1358
GSE19429	0.0824	0.0542	0.0708	0.0499	0.0717	0.0448
GSE9006	0.2125	0.2143	0.2111	0.1981	0.2085	0.1888
GSE48350	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
GSE5281	0.1621	0.1557	0.1717	0.1074	0.1731	0.1499
GSE35978	0.2577	0.2853	0.3237	0.3570	0.3309	0.3106
GSE53987	0.2401	0.2947	0.2711	0.3082	0.2759	0.3038
GSE12288	0.3149	0.4010	0.3319	0.3687	0.3566	0.3458
GSE15852	0.1668	0.2022	0.1929	0.1479	0.1885	0.1305
GSE42568	0.0806	0.0587	0.0650	0.0568	0.0649	0.0494
GSE29431	0.0279	0.0279	0.0279	0.0279	0.0279	0.0279
GSE18520	0.0159	0.0159	0.0159	0.0159	0.0159	0.0159
GSE19804	0.1092	0.0747	0.0992	0.0828	0.0984	0.1062
GSE10072	0.0552	0.0549	0.0726	0.0719	0.0761	0.0942
GSE68571	0.0106	0.0106	0.0106	0.0106	0.0106	0.0106
Average $\pm$ SEM	0.1564 ± 0.0234	0.1627 ± 0.0261	0.1626 ± 0.0248	0.1572 ± 0.0278	0.1642 ± 0.0252	0.1462 ± 0.0239

Table 21: Experiment 1c: Brier scores for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. Classifier with lower values of Brier score are better calibrated for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	0.0265	0.0102	0.0168	0.0103	0.0169	0.0005
GSE62646	0.0289	0.0289	0.0289	0.0289	0.0289	0.0289
GSE41861	0.0281	0.0328	0.0302	0.0370	0.0284	0.0370
GSE20881	0.0119	0.0029	0.0173	0.0118	0.0173	0.0118
GSE3365	0.0130	0.0164	0.0241	0.0158	0.0244	0.0238
GSE16879	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030
GSE15245	0.0535	0.0441	0.0289	0.0445	0.0288	0.0449
GSE6613	0.0644	0.0539	0.0608	0.0532	0.0533	0.0383
GSE20295	0.0507	0.0259	0.0225	0.0003	0.0224	0.0443
GSE30999	0.0060	0.0060	0.0060	0.0060	0.0060	0.0060
GSE55447	0.0408	0.0408	0.0188	0.0405	0.0188	0.0206
GSE19429	0.0468	0.0400	0.0354	0.0471	0.0365	0.0430
GSE9006	0.0798	0.0700	0.0819	0.0710	0.0700	0.0674
GSE48350	0.0016	0.0016	0.0016	0.0016	0.0016	0.0016
GSE5281	0.0064	0.0065	0.0064	0.0065	0.0064	0.0002
GSE35978	0.0433	0.0420	0.0419	0.0424	0.0421	0.0367
GSE53987	0.0481	0.0559	0.0585	0.0522	0.0590	0.0585
GSE12288	0.0459	0.0493	0.0272	0.0403	0.0347	0.0336
GSE15852	0.0379	0.0009	0.0368	0.0006	0.0422	0.0117
GSE42568	0.0268	0.0297	0.0260	0.0260	0.0261	0.0256
GSE29431	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008
GSE18520	0.0042	0.0042	0.0042	0.0042	0.0042	0.0042
GSE19804	0.0131	0.0090	0.0172	0.0168	0.0168	0.0168
GSE10072	0.0004	0.0004	0.0130	0.0085	0.0140	0.0004
GSE68571	0.0027	0.0027	0.0027	0.0027	0.0027	0.0027
Average $\pm$ SEM	0.0274 ± 0.0045	0.0231 ± 0.0043	0.0244 ± 0.0041	0.0229 ± 0.0042	0.0242 ± 0.0037	0.0225 ± 0.0040

Table 22: Experiment 1c: Expected calibration errors (ECEs) for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. Classifier with lower values of ECEs are better calibrated for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	0.9953	0.9899	0.8840	0.8948	0.8796	0.7967
GSE62646	0.3096	0.3096	0.3096	0.3096	0.3096	0.3096
GSE41861	0.9926	0.9496	0.9979	0.9973	0.9973	0.9975
GSE20881	0.9988	0.9997	0.9986	0.9979	0.9977	0.9978
GSE3365	0.7581	0.8980	0.8828	0.7009	0.8982	0.6998
GSE16879	0.2135	0.2135	0.2135	0.2135	0.2135	0.2135
GSE15245	0.7487	0.9969	0.7891	0.8967	0.7947	0.8965
GSE6613	0.9456	0.9489	0.9920	0.9975	0.9958	0.9959
GSE20295	0.9990	0.9997	0.9976	0.7996	0.9982	0.8302
GSE30999	0.4987	0.4987	0.6697	0.6697	0.6862	0.5495
GSE55447	0.9743	0.6944	0.9854	0.6936	0.9886	0.6021
GSE19429	0.4968	0.4997	0.4447	0.4705	0.4525	0.4295
GSE9006	0.8316	0.8880	0.9418	0.8558	0.8600	0.8649
GSE48350	0.0109	0.0109	0.0109	0.0109	0.0109	0.0109
GSE5281	0.9982	0.9117	0.9969	0.8975	0.9978	0.9756
GSE35978	0.7997	0.8183	0.8063	0.8381	0.8212	0.8455
GSE53987	0.8011	0.8568	0.8838	0.9386	0.8742	0.9581
GSE12288	0.7927	0.8818	0.8926	0.9446	0.8910	0.8697
GSE15852	0.9423	0.8941	0.9032	0.6986	0.8929	0.6978
GSE42568	0.7815	0.5450	0.5946	0.4963	0.5951	0.5034
GSE29431	0.2151	0.2151	0.2151	0.2151	0.2151	0.2151
GSE18520	0.1211	0.1211	0.1211	0.1211	0.1211	0.1211
GSE19804	0.9866	0.7186	0.8899	0.7176	0.8850	0.7975
GSE10072	0.4289	0.3998	0.6289	0.5791	0.6485	0.7975
GSE68571	0.1211	0.1211	0.1211	0.1211	0.1211	0.1211
Average $\pm$ SEM	0.6705 ± 0.0668	0.6552 ± 0.0661	0.6868 ± 0.0666	0.6430 ± 0.0626	0.6858 ± 0.0661	0.6439 ± 0.0633

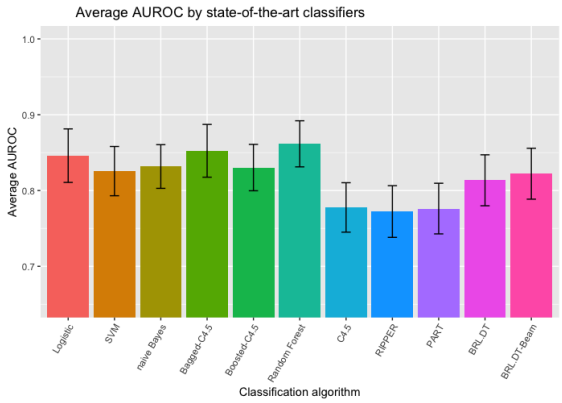
Table 23: Experiment 1c: Maximum calibration errors (MCEs) for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. Classifier with lower values of MCEs are better calibrated for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	10.4	8.0	5.4	4.2	3.2	2.2
GSE62646	2.0	2.0	2.0	2.0	2.0	2.0
GSE41861	37.6	54.8	8.2	6.8	2.7	2.8
GSE20881	86.4	120.0	10.7	9.1	2.8	2.7
GSE3365	14.4	9.2	6.9	4.9	2.7	2.4
GSE16879	2.0	2.0	2.0	2.0	2.0	2.0
GSE15245	10.2	9.6	4.8	5.2	3.2	3.5
GSE6613	13.2	51.2	7.8	8.0	2.9	2.8
GSE20295	103.0	143.0	14.2	12.7	7.5	4.7
GSE30999	3.6	3.6	3.3	3.3	3.0	3.4
GSE55447	5.7	4.3	3.9	3.2	2.9	2.3
GSE19429	12.4	8.0	5.7	4.6	2.7	2.6
GSE9006	8.4	8.0	4.3	4.1	2.4	2.4
GSE48350	2.0	2.0	2.0	2.0	2.0	2.0
GSE5281	25.6	21.6	9.1	6.0	3.3	2.6
GSE35978	31.2	98.4	11.5	15.0	4.2	3.9
GSE53987	27.2	32.8	8.4	8.7	2.9	3.0
GSE12288	27.2	120.8	8.9	10.8	3.4	3.7
GSE15852	8.3	8.8	5.2	4.1	2.3	2.2
GSE42568	3.0	3.9	3.2	3.1	2.6	2.8
GSE29431	2.0	2.0	2.0	2.0	2.0	2.0
GSE18520	2.0	2.0	2.0	2.0	2.0	2.0
GSE19804	6.2	4.0	4.1	3.1	2.8	2.5
GSE10072	3.2	3.2	3.2	3.2	2.6	2.6
GSE68571	2.0	2.0	2.0	2.0	2.0	2.0
Average $\pm$ SEM	17.97 ± 5.09	29.01 ± 8.74	5.63 ± 0.68	5.28 ± 0.72	2.88 ± 0.22	2.68 ± 0.14

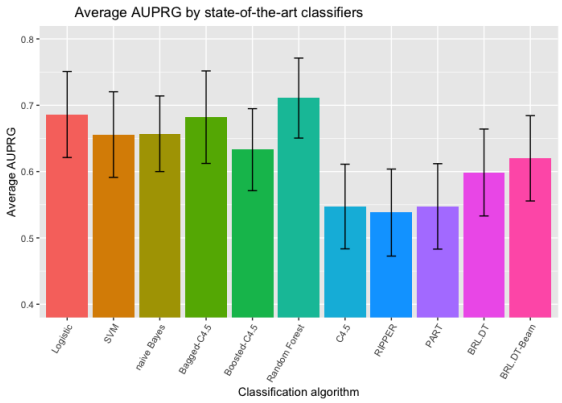
Table 24: Experiment 1c: Average number of rules (NRs) for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. Classifier with lower values of NRs are more succinct, and perhaps more readable for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	3.1	3.0	3.4	3.0	3.4	3.0
GSE62646	1.0	1.0	1.0	1.0	1.0	1.0
GSE41861	4.3	5.3	5.9	5.2	6.0	5.0
GSE20881	5.4	6.2	8.2	6.6	8.1	6.4
GSE3365	3.2	3.0	4.3	3.0	4.2	3.0
GSE16879	1.0	1.0	1.0	1.0	1.0	1.0
GSE15245	2.4	2.4	2.4	2.5	2.4	2.5
GSE6613	3.3	5.2	5.9	6.7	6.5	6.0
GSE20295	3.5	4.7	5.4	5.3	5.2	1.0
GSE30999	1.5	1.5	1.9	1.9	1.9	2.4
GSE55447	1.8	1.9	1.7	1.9	1.7	1.9
GSE19429	3.4	3.0	3.7	3.0	3.6	3.0
GSE9006	3.0	3.0	3.2	3.0	3.1	3.0
GSE48350	1.0	1.0	1.0	1.0	1.0	1.0
GSE5281	3.8	4.1	5.6	4.2	5.5	4.1
GSE35978	4.1	6.0	9.2	12.4	9.0	9.1
GSE53987	4.3	4.7	7.1	7.4	6.9	6.8
GSE12288	4.3	6.1	7.4	9.4	7.8	7.7
GSE15852	2.7	2.9	3.8	3.0	3.7	3.0
GSE42568	1.4	1.5	1.5	1.5	1.5	1.7
GSE29431	1.0	1.0	1.0	1.0	1.0	1.0
GSE18520	1.0	1.0	1.0	1.0	1.0	1.0
GSE19804	2.4	2.0	2.5	2.0	2.5	2.0
GSE10072	1.6	1.6	1.6	1.6	1.6	1.6
GSE68571	1.0	1.0	1.0	1.0	1.0	1.0
Average $\pm$ SEM	2.62 ± 0.27	2.96 ± 0.36	3.63 ± 0.51	3.58 ± 0.59	3.62 ± 0.51	3.17 ± 0.47

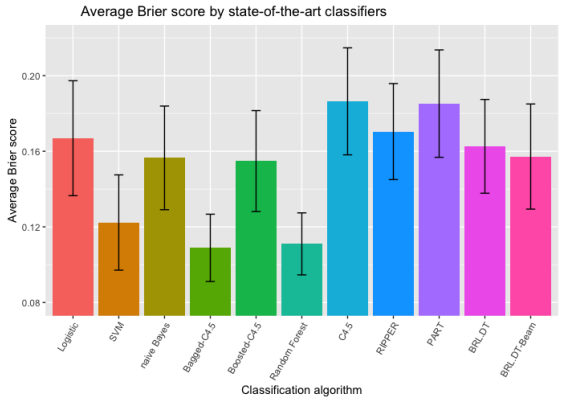
Table 25: Experiment 1c: Average number of variables (NVs) for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. Classifier with lower values of NVs are more parsimonious, and perhaps easier to validate for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.



(a) Comparing AUROCs



(b) Comparing AUPRGs



(c) Comparing Brier scores

Figure 14: Experiment 1: Average AUROCs, AUPRGs, and Brier scores by state-of-the-art classifiers, BRL.DT, and BRL.DT-Beam.

complex models like multivariate Logistic regression, SVM, Bagged and Boosted C4.5, and Random Forest outperform BRL.DT and BRL.DG. Same conclusion can be reached from Brier scores (see Figure 14c) where lower values indicate better calibrated classifiers.

4.3.5 Experiment 1: Conclusion

If the application were to require the user to use BRL off-the-shelf, I recommend BRL-DT starting with a default κ value of 0.01. These applications include exploratory data analysis for identifying potential hypotheses suggested by the data. The user may then optimize on different κ values, smaller κ values for learning models with fewer rules, and larger κ values for learning models with more rules. The user can perform optimization on this parameter using either grid search or more sophisticated methods like distributed asynchronized hyper-parameter optimization [Bergstra et al., 2015] or Bayesian optimization [Martinez-Cantin, 2014].

While BRL-DT performs similar to BRL-G, BRL-DT has a more parsimonious representation i.e., uses fewer rules and variables. Fewer rules means there are generally more instances covered by any given rule in the model. This offers more confidence in the predictions made by the rule. Fewer variables provides models that are cheaper to validate in practice.

4.4 EXPERIMENT 2: EVALUATING EBRL METHODS

All experiments in experiment 2 were performed using BRL.DT. And so, I use the terms BRL and BRL.DT interchangeably in this section. We had conducted the experiments with BRL.G and BRL.DG as well. The conclusions reached from each representation were not different. And so, to minimize the required number of tables, we only show results from BRL.DT.

In experiment 2, we will test 4 hypotheses. They are follows—

1. Experiment 2a (see subsection 4.4.1): We compare Bagged-BRL-LC to BRL, C4.5, Bagged-C4.5, and Boosted-C4.5. Since BRL has was previously shown to be better at modeling high-dimensional data, we expect it to be better at leveraging the enriched search space from bagging. As a result, we expect Bagged-BRL-LC to have better predictive and calibration performance than the other compared classifiers

2. Experiment 2b (see subsection 4.4.2): We compare the three EBRL models that use different strategies to combine a set of base classifiers. All base classifiers in this experiment are generated from bootstrap sampling. The three combination methods are— linear combination (Bagged-BRL-LC), Bayesian model averaging (Bagged-BRL-BMA), and Bayesian model combination (Bagged-BRL-BMC).
3. Experiment 2c (see subsection 4.4.3): Experiment 2a is repeated but this time we combine BRL models using Boosting as opposed to Bagging.
4. Experiment 2d (see subsection 4.4.4): Experiment 2b is repeated but by comparing the EBRL models that had base classifiers generated from Boosting.

We primarily focus on three metrics— predictive performance evaluated using AUROC and AUPRG, and calibration performance evaluated using Brier score.

In subsection 4.4.5, we take a closer look at Bayesian model combination method and outline situations where this aggregation method is superior to traditional bagging. In subsection 4.4.6, we compare the best performing EBRL method with the state-of-the-art classifiers in machine learning. We summarize the observations from this experiment in subsection 4.4.8.

4.4.1 Experiment 2a: Comparing Bagged-BRL-LC to BRL, C4.5, Bagged-C4.5, and Boosted-C4.5

In this experiment, we compare Bagged-BRL-LC (classic bagging with BRL) to BRL, C4.5, Bagged-C4.5, and Boosted-C4.5. We hypothesized Bagged-BRL-LC to take advantage of both bagging procedure and BRL's abilities in modeling high-dimensional datasets to achieve better predictive performance than BRL, C4.5, Bagged-C4.5, and Boosted-C4.5. The accuracy, precision, recall, and F-measure do not help us test the hypothesis, so those results were moved to Appendix D (see 10.1). Tables 26, 27, and 28 show the AUROCs, AUPRGs, and Brier scores attained by C4.5, Bagged-C4.5, Boosted C4.5, BRL, and Bagged-BRL-LC, respectively. The last row of the table shows the average of the metrics across the 25 datasets and the standard error of mean.

AUROCs achieved by the different classifiers were first compared against Bagged-BRL-

LC. The results are shown in Table 26. Friedman's test (p-value = $3.037E-08$), Friedman's test with Iman-Davenport correction (p-value = $2.536E-10$), Friedman's aligned ranks test (p-value = $4.859E-08$), and Quade test (p-value = $6.534E-10$) all strongly suggest that we must reject the null hypothesis suggesting that all classifiers have the same AUROC. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were adjusted using Holland's method to find that the AUROC achieved by Bagged-BRL-LC is statistically significantly different from BRL (p-value = 0.0005836114), C4.5 (p-value = $1.086147E-08$), Bagged-C4.5 (p-value = 0.03966867), and Boosted-C4.5 (p-value = $6.45522E-05$). Similar conclusions were reached using Finner, Rom, and Li methods. These results suggest that Bagged-BRL-LC generally achieves a statistically significantly higher AUROC than BRL, C4.5, Bagged-C4.5, and Boosted-C4.5.

AUPRGs achieved by the different classifiers were first compared against Bagged-BRL-LC. The results are shown in Table 27. Friedman's test (p-value = $5.937E-08$), Friedman's test with Iman-Davenport correction (p-value = $7.571E-10$), Friedman's aligned ranks test (p-value = $6.503E-09$), and Quade test (p-value = $2.156E-10$) all strongly suggest that we must reject the null hypothesis suggesting that all classifiers have the same AUPRG. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were adjusted using Holland's method to find that the AUPRG achieved by Bagged-BRL-LC is statistically significantly different from BRL (p-value = 0.0001377093), C4.5 (p-value = $9.073899E-08$), and Boosted-C4.5 (p-value = $9.585383E-05$) but not Bagged-C4.5 (p-value = 0.08924165). Similar conclusions were reached using Finner, Rom, and Li methods. These results suggest that Bagged-BRL-LC generally achieves a statistically significantly higher AUPRG than BRL, C4.5, and Boosted-C4.5. The gain of AUPRG over Bagged-C4.5, however, is not statistically significant.

Brier scores achieved by the different classifiers were first compared against Bagged-BRL-LC. The results are shown in Table 28. Friedman's test (p-value = $9.266E-08$), Friedman's test with Iman-Davenport correction (p-value = $1.543E-09$), Friedman's aligned ranks test (p-value = $1.071E-11$), and Quade test (p-value = $4.619E-14$) all strongly suggest that we must reject the null hypothesis suggesting that all classifiers have the same Brier

score. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were adjusted using Holland's method to find that the Brier score achieved by Bagged-BRL-LC is statistically significantly different from BRL (p-value = 0.001039498), C4.5 (p-value = $3.210044E-07$), and Boosted-C4.5 (p-value = 0.001039498) but not Bagged-C4.5 (p-value = 0.3710934). Similar conclusions were reached using Finner, Rom, and Li methods. These results suggest that Bagged-BRL-LC generally achieves a statistically significantly higher Brier scores than BRL, C4.5, and Boosted-C4.5. The gain of Brier score over Bagged-C4.5, however, is not statistically significant.

4.4.1.1 Experiment 2a: Conclusion

The results from experiment 2a shows that on an ensemble using BRL as base classifiers (Bagged-BRL-LC) gives better AUROC than an ensemble using C4.5 as base classifiers (Bagged-C4.5 and Boosted-C4.5). Bagged-BRL-LC also has better AUROC than BRL, showing that the ensemble model better represents multifactorial diseases than single model.

4.4.2 Experiment 2b: Comparing Bagged-BRL-LC , Bagged-BRL-BMA, and Bagged-BRL-BMC

In this experiment, we compare the three different model combination strategies using BRL models generated from bootstrap samples. The three models are Bagged-BRL-LC, Bagged-BRL-BMA, and Bagged-BRL-BMC. We hypothesized Bagged-BRL-BMC would have the better predictive performance by accounting for the uncertainty in the correctness of model combination. The accuracy, precision, recall, and F-measure do not help us test the hypothesis, so those results were moved to Appendix D (see 10.2). Tables 29, 30, and 31 show the AUROCs, AUPRGs, and Brier scores attained by Bagged-BRL-LC, Bagged-BRL-BMA, and Bagged-BRL-BMC, respectively. The last row of the table shows the average of the metrics across the 25 datasets and the standard error of mean.

AUROCs achieved by the different classifiers were first compared against Bagged-BRL-LC. The results are shown in Table 29. Friedman's test (p-value = 0.03239), Friedman's

Data	Bagged-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	0.9630	0.8490	0.6940	0.9500	0.8320
GSE62646	1.0000	0.9083	0.9083	1.0000	0.9083
GSE41861	0.9242	0.7717	0.7164	0.8539	0.8961
GSE20881	0.9153	0.8649	0.7518	0.9118	0.8342
GSE3365	0.9680	0.9174	0.7994	0.9661	0.9875
GSE16879	1.0000	0.9845	0.9845	1.0000	0.9845
GSE15245	0.7000	0.7150	0.6083	0.6850	0.5800
GSE6613	0.6260	0.4697	0.5847	0.5817	0.6130
GSE20295	0.7292	0.6221	0.6413	0.7729	0.7046
GSE30999	0.9812	0.9764	0.9458	0.9826	0.9458
GSE55447	0.6450	0.5325	0.6125	0.5000	0.6375
GSE19429	0.9642	0.9001	0.6254	0.7019	0.8828
GSE9006	0.9200	0.8658	0.7383	0.8967	0.8150
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.9626	0.8801	0.8263	0.9560	0.8808
GSE35978	0.7444	0.5436	0.5129	0.6642	0.6435
GSE53987	0.4917	0.5564	0.5502	0.5291	0.5493
GSE12288	0.6003	0.5723	0.5357	0.5361	0.5339
GSE15852	0.9487	0.8194	0.7631	0.8988	0.8475
GSE42568	0.9614	0.8109	0.8955	0.9682	0.8705
GSE29431	1.0000	0.9417	0.9417	0.9917	0.9417
GSE18520	1.0000	0.9900	0.9900	0.9900	0.9900
GSE19804	0.9639	0.9083	0.8806	0.9861	0.9431
GSE10072	0.9867	0.9425	0.9425	0.9933	0.9425
GSE68571	1.0000	0.9938	0.9937	0.9937	0.9937
Average $\pm$ SEM	0.8798 ± 0.0311	0.8135 ± 0.0336	0.7777 ± 0.0326	0.8524 ± 0.0348	0.8303 ± 0.0306

Table 26: Experiment 2a: AUROCs for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of AUROCs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	Bagged-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	0.9357	0.7234	0.4212	0.9104	0.6697
GSE62646	1.0000	0.7833	0.7833	1.0000	0.7833
GSE41861	0.8179	0.4126	0.4223	0.6425	0.7755
GSE20881	0.8139	0.7311	0.5146	0.8094	0.6763
GSE3365	0.9078	0.7627	0.5734	0.9063	0.9611
GSE16879	1.0000	0.8845	0.8845	1.0000	0.8845
GSE15245	0.5400	0.3070	0.1481	0.3100	0.1195
GSE6613	0.2774	-0.0205	0.1942	0.2267	0.2677
GSE20295	0.4984	0.3450	0.3233	0.5638	0.4296
GSE30999	0.9747	0.9599	0.8988	0.9768	0.8988
GSE55447	0.2150	-0.0075	0.2500	0.1025	0.2000
GSE19429	0.6556	0.6025	0.1514	0.1807	0.4480
GSE9006	0.7486	0.6413	0.4379	0.6576	0.5752
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.9291	0.7763	0.6788	0.9009	0.7618
GSE35978	0.4257	0.0948	0.0644	0.2591	0.2303
GSE53987	0.0139	0.0819	0.0854	0.0633	0.0597
GSE12288	0.1119	0.1860	0.0855	0.0440	0.0883
GSE15852	0.9038	0.6777	0.5590	0.8100	0.7237
GSE42568	0.8455	0.5631	0.7955	0.8955	0.7455
GSE29431	1.0000	0.8417	0.8417	0.9417	0.8417
GSE18520	1.0000	0.9400	0.9400	0.9400	0.9400
GSE19804	0.9358	0.8420	0.7937	0.9793	0.9083
GSE10072	0.9700	0.8920	0.8920	0.9850	0.8920
GSE68571	1.0000	0.9438	0.9438	0.9438	0.9438
Average $\pm$ SEM	0.7408 ± 0.0622	0.5986 ± 0.0655	0.5473 ± 0.0638	0.6820 ± 0.0697	0.6330 ± 0.0618

Table 27: Experiment 2a: AUPRGs for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of AUPRGs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	Bagged-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	0.1009	0.1675	0.2889	0.1012	0.2044
GSE62646	0.0355	0.0684	0.0700	0.0466	0.0700
GSE41861	0.1107	0.2492	0.2482	0.1499	0.1414
GSE20881	0.1201	0.2059	0.2442	0.1313	0.2096
GSE3365	0.0672	0.1642	0.1884	0.0831	0.0371
GSE16879	0.0304	0.0257	0.0268	0.0265	0.0268
GSE15245	0.1604	0.1910	0.3128	0.1461	0.2312
GSE6613	0.2485	0.4566	0.4097	0.2656	0.4011
GSE20295	0.1942	0.3234	0.3467	0.1955	0.3384
GSE30999	0.0323	0.0552	0.0530	0.0298	0.0531
GSE55447	0.1731	0.2943	0.3194	0.1875	0.3071
GSE19429	0.0506	0.0708	0.1436	0.0799	0.0721
GSE9006	0.1130	0.2111	0.2049	0.1348	0.1826
GSE48350	0.0010	0.0001	0.0000	0.0006	0.0000
GSE5281	0.0901	0.1717	0.1804	0.0898	0.1517
GSE35978	0.1860	0.3237	0.3918	0.2427	0.3300
GSE53987	0.2414	0.2711	0.2996	0.2445	0.3003
GSE12288	0.2590	0.3319	0.4541	0.3047	0.4413
GSE15852	0.1023	0.1929	0.2271	0.1086	0.1490
GSE42568	0.0308	0.0650	0.0251	0.0239	0.0379
GSE29431	0.0138	0.0279	0.0286	0.0248	0.0286
GSE18520	0.0111	0.0159	0.0167	0.0176	0.0167
GSE19804	0.0664	0.0992	0.1125	0.0464	0.0743
GSE10072	0.0306	0.0726	0.0556	0.0258	0.0555
GSE68571	0.0143	0.0106	0.0111	0.0140	0.0111
Average $\pm$ SEM	0.0993 ± 0.0160	0.1626 ± 0.0248	0.1864 ± 0.0283	0.1089 ± 0.0178	0.1548 ± 0.0267

Table 28: Experiment 2a: Brier scores for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with lower values of Brier scores are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

test with Iman-Davenport correction (p-value = 0.02896), Friedman's aligned ranks test (p-value = 0.0007198), and Quade test (p-value = 0.0007889) all suggest that we must reject the null hypothesis suggesting that all classifiers have the same AUROC. We perform a post-hoc test to see which of the classifiers behave differently from one another. We do this using a pairwise comparison by correcting the p-values obtained by Friedman's test with Bergmann and Hommel's method. Bagged-BRL-BMA has statistically significantly weaker predictive performance than Bagged-BRL-LC (p-*value* = 0.04) and Bagged-BRL-BMC (p-*value* = 0.05). However, Bagged-BRL-LC is similar to Bagged-BRL-BMC (p-*value* = 0.62).

AUPRGs achieved by the different classifiers were first compared against Bagged-BRL-LC. The results are shown in Table 30. Friedman's test (p-value = 0.05448) and Friedman's test with Iman-Davenport correction (p-value = 0.0513) do not recommend rejecting the null hypothesis suggesting that all classifiers have the same AUPRG. However, Friedman's aligned ranks test (p-value = 0.003475) and Quade test (p-value = 0.002091) both support rejecting the null hypothesis. We proceed with the results from Friedman's aligned ranks test. We perform a post-hoc test to see which of the classifiers behave differently from one another. We do this using a pairwise comparison by correcting the p-values obtained by Friedman's aligned ranks test with Bergmann and Hommel's method. Bagged-BRL-BMA has statistically significantly weaker predictive performance than Bagged-BRL-LC (p-*value* < 0.01) and Bagged-BRL-BMC (p-*value* < 0.01). However, Bagged-BRL-LC is similar to Bagged-BRL-BMC (p-*value* = 0.95).

Brier scores achieved by the different classifiers were first compared against Bagged-BRL-LC. The results are shown in Table 31. Friedman's test (p-value = $3.167E-05$), Friedman's test with Iman-Davenport correction (p-value = $2.645E-06$), Friedman's aligned ranks test (p-value = $9.428E-07$), and Quade test (p-value = $5.073E-07$) all strongly suggest that we must reject the null hypothesis suggesting that all classifiers have the same Brier scores. We perform a post-hoc test to see which of the classifiers behave differently from one another. We do this using a pairwise comparison by correcting the p-values obtained by Friedman's test with Bergmann and Hommel's method. Bagged-BRL-BMA has statistically significantly weaker calibration performance than Bagged-BRL-LC (p-*value* < 0.01) and Bagged-BRL-BMC (p-*value* < 0.01). However, Bagged-BRL-LC is similar to Bagged-BRL-

BMC ($p\text{-}value = 0.26$).

4.4.2.1 Experiment 2b: Conclusion The results from experiment 2b shows that Bagged-BRL-LC and Bagged-BRL-BMC have similar predictive and calibration performance. But they are significantly better than Bagged-BRL-BMA. These results are consistent with the observations of [Minka, 2000]. However, we expected Bagged-BRL-BMC to outperform Bagged-BRL-LC. We explore this result further in subsection 4.4.5.

4.4.3 Experiment 2c: Comparing Boosted-BRL-LC to BRL, C4.5, Bagged-C4.5, and Boosted-C4.5

In this experiment, we compare Boosted-BRL-LC (classic bagging with BRL) to BRL, C4.5, Bagged-C4.5, and Boosted-C4.5. We hypothesized Boosted-BRL-LC to take advantage of both bagging procedure and BRL's abilities in modeling high-dimensional datasets to achieve better predictive performance than BRL, C4.5, Bagged-C4.5, and Boosted-C4.5. The accuracy, precision, recall, and F-measure do not help us test the hypothesis, so those results were moved to Appendix D (see 10.3). Tables 32, 33, and 34 show the AUROCs, AUPRGs, and Brier scores attained by C4.5, Bagged-C4.5, Boosted C4.5, BRL, and Boosted-BRL-LC, respectively. The last row of the table shows the average of the metrics across the 25 datasets and the standard error of mean.

AUROCs achieved by the different classifiers were first compared against Boosted-BRL-LC. The results are shown in Table 32. Friedman's test (p-value = 0.0003144), Friedman's test with Iman-Davenport correction (p-value = 0.0001339), Friedman's aligned ranks test (p-value = 0.0001994), and Quade test (p-value = 0.0005432) all suggest that we must reject the null hypothesis suggesting that all classifiers have the same AUROC. Post-hoc test is now conducted to determine which of the classifiers are different from others. The p-values generated by Friedman's test were adjusted using Holland's method to find that the AUROC achieved by Boosted-BRL-LC is statistically significantly different from any of the classifiers. Similar conclusions were reached using Finner, Rom, and Li methods. The results indicate that the source of difference between the classifiers does not come from Boosted-BRL-LC.

Data	Bagged-BRL.DT-LC	Bagged-BRL.DT-BMA	Bagged-BRL.DT-BMC
GSE66360	0.9630	0.9300	0.9630
GSE62646	1.0000	1.0000	1.0000
GSE41861	0.9242	0.8811	0.9358
GSE20881	0.9153	0.8731	0.9158
GSE3365	0.9680	0.9693	0.9764
GSE16879	1.0000	0.9833	0.9833
GSE15245	0.7000	0.6800	0.6800
GSE6613	0.6260	0.6013	0.6247
GSE20295	0.7292	0.7183	0.7533
GSE30999	0.9812	0.9812	0.9799
GSE55447	0.6450	0.5250	0.6900
GSE19429	0.9642	0.9642	0.9670
GSE9006	0.9200	0.9117	0.9400
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.9626	0.9172	0.9597
GSE35978	0.7444	0.6593	0.7338
GSE53987	0.4917	0.4941	0.4859
GSE12288	0.6003	0.5619	0.5630
GSE15852	0.9487	0.9175	0.9325
GSE42568	0.9614	0.9614	0.9432
GSE29431	1.0000	1.0000	1.0000
GSE18520	1.0000	1.0000	1.0000
GSE19804	0.9639	0.9639	0.9583
GSE10072	0.9867	0.9867	0.9900
GSE68571	1.0000	1.0000	1.0000
Average $\pm$ SEM	0.8798 ± 0.0311	0.8592 ± 0.0341	0.8790 ± 0.0314

Table 29: Experiment 2b: AUROCs for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of AUROCs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	Bagged-BRL.DT-LC	Bagged-BRL.DT-BMA	Bagged-BRL.DT-BMC
GSE66360	0.9357	0.8835	0.9357
GSE62646	1.0000	1.0000	1.0000
GSE41861	0.8179	0.6648	0.8471
GSE20881	0.8139	0.7336	0.8131
GSE3365	0.9078	0.9185	0.9354
GSE16879	1.0000	0.8917	0.8917
GSE15245	0.5400	0.3700	0.4800
GSE6613	0.2774	0.2232	0.2767
GSE20295	0.4984	0.4754	0.5506
GSE30999	0.9747	0.9747	0.9731
GSE55447	0.2150	0.1025	0.3275
GSE19429	0.6556	0.6556	0.6828
GSE9006	0.7486	0.7319	0.8207
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.9291	0.8242	0.9255
GSE35978	0.4257	0.2191	0.3733
GSE53987	0.0139	0.0349	0.0022
GSE12288	0.1119	0.1210	0.0724
GSE15852	0.9038	0.8353	0.8494
GSE42568	0.8455	0.8455	0.8455
GSE29431	1.0000	1.0000	1.0000
GSE18520	1.0000	1.0000	1.0000
GSE19804	0.9358	0.9358	0.9254
GSE10072	0.9700	0.9700	0.9797
GSE68571	1.0000	1.0000	1.0000
Average $\pm$ SEM	0.7408 ± 0.0622	0.6964 ± 0.0656	0.7403 ± 0.0616

Table 30: Experiment 2b: AUPRGs for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of AUPRGs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	Bagged-BRL.DT-LC	Bagged-BRL.DT-BMA	Bagged-BRL.DT-BMC
GSE66360	0.1009	0.1394	0.0985
GSE62646	0.0355	0.0363	0.0377
GSE41861	0.1107	0.1646	0.1071
GSE20881	0.1201	0.1675	0.1209
GSE3365	0.0672	0.0993	0.0649
GSE16879	0.0304	0.0312	0.0310
GSE15245	0.1604	0.1590	0.1615
GSE6613	0.2485	0.3194	0.2495
GSE20295	0.1942	0.2297	0.1929
GSE30999	0.0323	0.0374	0.0347
GSE55447	0.1731	0.2322	0.1640
GSE19429	0.0506	0.0571	0.0498
GSE9006	0.1130	0.1867	0.1167
GSE48350	0.0010	0.0021	0.0011
GSE5281	0.0901	0.1389	0.0882
GSE35978	0.1860	0.2969	0.1929
GSE53987	0.2414	0.3225	0.2509
GSE12288	0.2590	0.3914	0.2720
GSE15852	0.1023	0.1267	0.1060
GSE42568	0.0308	0.0292	0.0319
GSE29431	0.0138	0.0167	0.0161
GSE18520	0.0111	0.0116	0.0123
GSE19804	0.0664	0.0696	0.0693
GSE10072	0.0306	0.0306	0.0300
GSE68571	0.0143	0.0148	0.0142
Average $\pm$ SEM	0.0993 ± 0.0160	0.1324 ± 0.0227	0.1006 ± 0.0163

Table 31: Experiment 2b: Brier scores for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with lower values of Brier scores are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

The difference likely stems from Bagged-C4.5 since Bagging appears to do better in these datasets than Boosting.

Similar results as AUROC were obtained using AUPRG and Brier score.

4.4.3.1 Experiment 2c: Conclusion

The results from experiment 2c shows that Boosting method generally performs poorly on these datasets. There is a gain in performance when compared to single BRL models but the gain isn't statistically significant. We tried tuning some parameters of boosting with little improvement on these datasets.

One possible explanation comes from [Freund et al., 1996]. Boosting methods generally perform poorly on noisy datasets. This is because noisy instances are likely to be misclassified by a good classifier. boosting iteratively focuses on misclassified instances. This forces the classifiers to try and fit a model with more emphasis on noisy instances. This leads to learning poor classifiers.

4.4.4 Experiment 2d: Comparing Boosted-BRL-LC , Boosted-BRL-BMA, and Boosted-BRL-BMC

In this experiment, we repeat the experiment 2b (see 4.4.2) but we now test the combination of base classifiers generated from boosting. The accuracy, precision, recall, and F-measure do not help us test the hypothesis, so those results were moved to Appendix D (see 10.4). Tables 35, 36, and 37 show the AUROCs, AUPRGs, and Brier scores attained by Boosted-BRL-LC, Boosted-BRL-BMA, and Boosted-BRL-BMC, respectively. The last row of the table shows the average of the metrics across the 25 datasets and the standard error of mean.

AUROCs achieved by the different classifiers were first compared against Boosted-BRL-LC. The results are shown in Table 35. Friedman's test (p-value = 0.0455), Friedman's test with Iman-Davenport correction (p-value = 0.04216), Friedman's aligned ranks test (p-value = 0.00205), and Quade test (p-value = 0.002149) all suggest that we must reject the null hypothesis suggesting that all classifiers have the same AUROC. We perform a post-hoc test to see which of the classifiers behave differently from one another. We do this using a pairwise comparison by correcting the p-values obtained by Friedman's aligned

Data	Boosted-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	0.9110	0.8490	0.6940	0.9500	0.8320
GSE62646	0.9083	0.9083	0.9083	1.0000	0.9083
GSE41861	0.7618	0.7717	0.7164	0.8539	0.8961
GSE20881	0.8938	0.8649	0.7518	0.9118	0.8342
GSE3365	0.9118	0.9174	0.7994	0.9661	0.9875
GSE16879	0.9845	0.9845	0.9845	1.0000	0.9845
GSE15245	0.6750	0.7150	0.6083	0.6850	0.5800
GSE6613	0.4433	0.4697	0.5847	0.5817	0.6130
GSE20295	0.6458	0.6221	0.6413	0.7729	0.7046
GSE30999	0.9764	0.9764	0.9458	0.9826	0.9458
GSE55447	0.7525	0.5325	0.6125	0.5000	0.6375
GSE19429	0.7895	0.9001	0.6254	0.7019	0.8828
GSE9006	0.9325	0.8658	0.7383	0.8967	0.8150
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8940	0.8801	0.8263	0.9560	0.8808
GSE35978	0.5642	0.5436	0.5129	0.6642	0.6435
GSE53987	0.4844	0.5564	0.5502	0.5291	0.5493
GSE12288	0.5732	0.5723	0.5357	0.5361	0.5339
GSE15852	0.8550	0.8194	0.7631	0.8988	0.8475
GSE42568	0.8705	0.8109	0.8955	0.9682	0.8705
GSE29431	0.9417	0.9417	0.9417	0.9917	0.9417
GSE18520	0.9900	0.9900	0.9900	0.9900	0.9900
GSE19804	0.9486	0.9083	0.8806	0.9861	0.9431
GSE10072	0.9425	0.9425	0.9425	0.9933	0.9425
GSE68571	0.9937	0.9938	0.9937	0.9937	0.9937
Average ± SEM	0.8258 ± 0.0337	0.8135 ± 0.0336	0.7777 ± 0.0326	0.8524 ± 0.0348	0.8303 ± 0.0306

Table 32: Experiment 2c: AUROCs for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of AUROCs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	Boosted-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	0.8339	0.7234	0.4212	0.9104	0.6697
GSE62646	0.7833	0.7833	0.7833	1.0000	0.7833
GSE41861	0.4558	0.4126	0.4223	0.6425	0.7755
GSE20881	0.7685	0.7311	0.5146	0.8094	0.6763
GSE3365	0.7534	0.7627	0.5734	0.9063	0.9611
GSE16879	0.8845	0.8845	0.8845	1.0000	0.8845
GSE15245	0.1786	0.3070	0.1481	0.3100	0.1195
GSE6613	-0.0974	-0.0205	0.1942	0.2267	0.2677
GSE20295	0.3391	0.3450	0.3233	0.5638	0.4296
GSE30999	0.9599	0.9599	0.8988	0.9768	0.8988
GSE55447	0.3400	-0.0075	0.2500	0.1025	0.2000
GSE19429	0.4636	0.6025	0.1514	0.1807	0.4480
GSE9006	0.8004	0.6413	0.4379	0.6576	0.5752
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.7847	0.7763	0.6788	0.9009	0.7618
GSE35978	0.1322	0.0948	0.0644	0.2591	0.2303
GSE53987	0.0194	0.0819	0.0854	0.0633	0.0597
GSE12288	0.1237	0.1860	0.0855	0.0440	0.0883
GSE15852	0.7421	0.6777	0.5590	0.8100	0.7237
GSE42568	0.7455	0.5631	0.7955	0.8955	0.7455
GSE29431	0.8417	0.8417	0.8417	0.9417	0.8417
GSE18520	0.9400	0.9400	0.9400	0.9400	0.9400
GSE19804	0.9170	0.8420	0.7937	0.9793	0.9083
GSE10072	0.8920	0.8920	0.8920	0.9850	0.8920
GSE68571	0.9438	0.9438	0.9438	0.9438	0.9438
Average $\pm$ SEM	0.6218 ± 0.0673	0.5986 ± 0.0655	0.5473 ± 0.0638	0.6820 ± 0.0697	0.6330 ± 0.0618

Table 33: Experiment 2c: AUPRGs for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of AUPRGs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	Boosted-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	0.1195	0.1675	0.2889	0.1012	0.2044
GSE62646	0.0658	0.0684	0.0700	0.0466	0.0700
GSE41861	0.1896	0.2492	0.2482	0.1499	0.1414
GSE20881	0.1720	0.2059	0.2442	0.1313	0.2096
GSE3365	0.1168	0.1642	0.1884	0.0831	0.0371
GSE16879	0.0314	0.0257	0.0268	0.0265	0.0268
GSE15245	0.1798	0.1910	0.3128	0.1461	0.2312
GSE6613	0.2629	0.4566	0.4097	0.2656	0.4011
GSE20295	0.2293	0.3234	0.3467	0.1955	0.3384
GSE30999	0.0402	0.0552	0.0530	0.0298	0.0531
GSE55447	0.1857	0.2943	0.3194	0.1875	0.3071
GSE19429	0.0592	0.0708	0.1436	0.0799	0.0721
GSE9006	0.1434	0.2111	0.2049	0.1348	0.1826
GSE48350	0.0046	0.0001	0.0000	0.0006	0.0000
GSE5281	0.1410	0.1717	0.1804	0.0898	0.1517
GSE35978	0.2364	0.3237	0.3918	0.2427	0.3300
GSE53987	0.2255	0.2711	0.2996	0.2445	0.3003
GSE12288	0.2475	0.3319	0.4541	0.3047	0.4413
GSE15852	0.1489	0.1929	0.2271	0.1086	0.1490
GSE42568	0.0388	0.0650	0.0251	0.0239	0.0379
GSE29431	0.0346	0.0279	0.0286	0.0248	0.0286
GSE18520	0.0256	0.0159	0.0167	0.0176	0.0167
GSE19804	0.0858	0.0992	0.1125	0.0464	0.0743
GSE10072	0.0598	0.0726	0.0556	0.0258	0.0555
GSE68571	0.0211	0.0106	0.0111	0.0140	0.0111
Average $\pm$ SEM	0.1226 ± 0.0164	0.1626 ± 0.0248	0.1864 ± 0.0283	0.1089 ± 0.0178	0.1548 ± 0.0267

Table 34: Experiment 2c: Brier scores for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with lower values of Brier scores are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

ranks test with Bergmann and Hommel's method. Boosted-BRL-BMA has statistically significantly weaker predictive performance than Boosted-BRL-LC ($p\text{-}value < 0.01$) and Boosted-BRL-BMC ($p\text{-}value < 0.01$). However, Boosted-BRL-LC is similar to Boosted-BRL-BMC ($p\text{-}value = 0.97$).

AUPRGs achieved by the different classifiers were first compared against Boosted-BRL-LC. The results are shown in Table 36. Friedman's test (p-value = 0.02548), Friedman's test with Iman-Davenport correction (p-value = 0.02214), Friedman's aligned ranks test (p-value = 0.0007343) and Quade test (p-value = 0.0009154) both support rejecting the null hypothesis suggesting that all classifiers have the same AUPRG. We perform a post-hoc test to see which of the classifiers behave differently from one another. We do this using a pairwise comparison by correcting the p-values obtained by Friedman's test with Bergmann and Hommel's method. Boosted-BRL-BMA has statistically significantly weaker predictive performance than Boosted-BRL-LC ($p\text{-}value = 0.04$) and Boosted-BRL-BMC ($p\text{-}value = 0.04$). However, Boosted-BRL-LC is similar to Boosted-BRL-BMC ($p\text{-}value = 0.78$).

Brier scores achieved by the different classifiers were first compared against Boosted-BRL-LC. The results are shown in Table 37. Friedman's test (p-value = $6.007E-05$), Friedman's test with Iman-Davenport correction (p-value = $7.386E-06$), Friedman's aligned ranks test (p-value = $2.592E-07$), and Quade test (p-value = $2.355E-07$) all strongly suggest that we must reject the null hypothesis suggesting that all classifiers have the same Brier scores. We perform a post-hoc test to see which of the classifiers behave differently from one another. We do this using a pairwise comparison by correcting the p-values obtained by Friedman's test with Bergmann and Hommel's method. Boosted-BRL-BMA has statistically significantly weaker calibration performance than Boosted-BRL-LC ($p\text{-}value < 0.01$) and Boosted-BRL-BMC ($p\text{-}value < 0.01$). However, Boosted-BRL-LC is similar to Boosted-BRL-BMC ($p\text{-}value = 1.00$).

4.4.4.1 Experiment 2d: Conclusion

The results from experiment 2d are largely consistent with the results of experiment 2b. This shows that the superior performance of the true ensemble methods Boosted-BRL-LC and Boosted-BRL-BMC, when compared to Boosted-BRL-BMA, was not dependent upon the method of generation of base models (Bag-

ging and Boosting).

Data	Boosted-BRL.DT-LC	Boosted-BRL.DT-BMA	Boosted-BRL.DT-BMC
GSE66360	0.9110	0.9020	0.9060
GSE62646	0.9083	0.9083	0.9083
GSE41861	0.7618	0.7163	0.7801
GSE20881	0.8938	0.7593	0.9186
GSE3365	0.9118	0.8620	0.9168
GSE16879	0.9845	0.9845	0.9845
GSE15245	0.6750	0.5950	0.6550
GSE6613	0.4433	0.4333	0.4433
GSE20295	0.6458	0.5900	0.6750
GSE30999	0.9764	0.9764	0.9764
GSE55447	0.7525	0.7525	0.7525
GSE19429	0.7895	0.7923	0.7895
GSE9006	0.9325	0.9325	0.9325
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.8940	0.8386	0.8936
GSE35978	0.5642	0.5283	0.5586
GSE53987	0.4844	0.5083	0.4856
GSE12288	0.5732	0.5180	0.5840
GSE15852	0.8550	0.8488	0.8613
GSE42568	0.8705	0.8705	0.8705
GSE29431	0.9417	0.9417	0.9417
GSE18520	0.9900	0.9900	0.9900
GSE19804	0.9486	0.9458	0.9486
GSE10072	0.9425	0.9425	0.9425
GSE68571	0.9937	0.9937	0.9937
Average ± SEM	0.8258 ± 0.0337	0.8052 ± 0.0356	0.8283 ± 0.0336

Table 35: Experiment 2d: AUROCs for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of AUROCs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	Boosted-BRL.DT-LC	Boosted-BRL.DT-BMA	Boosted-BRL.DT-BMC
GSE66360	0.8339	0.8221	0.8277
GSE62646	0.7833	0.7833	0.7833
GSE41861	0.4558	0.3624	0.4848
GSE20881	0.7685	0.5198	0.8256
GSE3365	0.7534	0.6254	0.7691
GSE16879	0.8845	0.8845	0.8845
GSE15245	0.1786	0.1786	0.1786
GSE6613	-0.0974	-0.1307	-0.1058
GSE20295	0.3391	0.2038	0.4012
GSE30999	0.9599	0.9599	0.9599
GSE55447	0.3400	0.3400	0.3400
GSE19429	0.4636	0.4891	0.4636
GSE9006	0.8004	0.8004	0.8004
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.7847	0.6434	0.7998
GSE35978	0.1322	0.0252	0.1159
GSE53987	0.0194	0.0052	0.0194
GSE12288	0.1237	0.0587	0.0903
GSE15852	0.7421	0.7324	0.7518
GSE42568	0.7455	0.7455	0.7455
GSE29431	0.8417	0.8417	0.8417
GSE18520	0.9400	0.9400	0.9400
GSE19804	0.9170	0.9133	0.9170
GSE10072	0.8920	0.8920	0.8920
GSE68571	0.9438	0.9438	0.9438
Average $\pm$ SEM	0.6218 ± 0.0673	0.5832 ± 0.0706	0.6268 ± 0.0679

Table 36: Experiment 2d: AUPRGs for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of AUPRGs are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

Data	Boosted-BRL.DT-LC	Boosted-BRL.DT-BMA	Boosted-BRL.DT-BMC
GSE66360	0.1195	0.2496	0.1197
GSE62646	0.0658	0.2290	0.0967
GSE41861	0.1896	0.2309	0.1761
GSE20881	0.1720	0.2457	0.1631
GSE3365	0.1168	0.2283	0.1141
GSE16879	0.0314	0.1647	0.0459
GSE15245	0.1798	0.1855	0.1812
GSE6613	0.2629	0.2497	0.2696
GSE20295	0.2293	0.2472	0.2256
GSE30999	0.0402	0.2492	0.0604
GSE55447	0.1857	0.1783	0.1750
GSE19429	0.0592	0.1209	0.0621
GSE9006	0.1434	0.2236	0.1413
GSE48350	0.0046	0.2364	0.0401
GSE5281	0.1410	0.2488	0.1370
GSE35978	0.2364	0.2278	0.2328
GSE53987	0.2255	0.2096	0.2364
GSE12288	0.2475	0.2500	0.2473
GSE15852	0.1489	0.2496	0.1443
GSE42568	0.0388	0.1515	0.0386
GSE29431	0.0346	0.1737	0.0550
GSE18520	0.0256	0.1625	0.0359
GSE19804	0.0858	0.2491	0.0847
GSE10072	0.0598	0.2454	0.0666
GSE68571	0.0211	0.1323	0.0344
Average $\pm$ SEM	0.1226 ± 0.0164	0.2136 ± 0.0083	0.1273 ± 0.0151

Table 37: Experiment 2d: Brier scores for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with lower values of Brier scores are better performing for a given dataset. The last row calculates the average for each classifier across 25 datasets and also reports the standard error of mean.

4.4.5 Experiment 2: Bagged-BRL-LC and Bagged-BRL-BMC compared with unreliable base classifiers

Bagged-BRL-LC and Bagged-BRL-BMC do appear to benefit from the enriched ensemble space that Bagged-BRL-BMA does not take advantage of. However, based on these results alone, it would appear that Bagged-BRL-BMC has an additional overhead of computation to Bagged-BRL-LC, which results in no apparent gain in predictive performance. One explanation for this could be because BMC attempts to evaluate the uncertainty in the correctness of model combination and it could be that bagging (LC) approach already had the most likely correct way of combining the models given the way that we generate the ensemble of base models. From bagged samples followed by greedy best-first BRL search it is possible that we learn models that are more or less equally correct. So, BMC ends up learning that the uniform model weight distribution used by Bagged-BRL-LC is most likely the correct weight distribution.

We conducted an additional experiment to test how LC and BMC behave, when the base classifiers are not all, more or less, equally correct. To test this, we conducted an experiment wherein we deliberately added substandard models to the hypothesis space and let LC and BMC learn models from this hypothesis space. So, instead of generating base models using the bagging approach, we instead create an alternate approach to generating base classifiers for the ensemble to combine. We call this alternate approach— 1N9R (short for 1 normal and 9 random classifiers). 1N9R generates the first base classifier, learned using the BRL greedy-best first search on the original data. Here, the first model is a reliable base classifier for the ensemble. The next 9 iterations learn random BRL classifiers. A random BRL classifier randomly guesses the label of a queried test instance, with a toss of a coin. The probability associated with the random prediction itself is randomly sampled uniformly from 0.0 to 1.0. To summarize, the first base model is reliable, generated from BRL's greedy best-first search, and the remaining 9 models are unreliable random classifiers. We expect that BMC would learn a model weight distribution, where the first model gets weighed significantly more than the remaining nine models during model combination. LC, on the other hand, would be forced to treat all models equally during combination from the

equal model weight distribution that it assigns.

Data	1N9R-BRL-LC	1N9R-BRL-BMC
GSE66360	0.6730	0.8420
GSE62646	0.7000	0.9500
GSE41861	0.6392	0.7664
GSE20881	0.7087	0.7759
GSE3365	0.6622	0.8648
GSE16879	0.6750	0.9690
GSE15245	0.6600	0.7400
GSE6613	0.4600	0.4720
GSE20295	0.6992	0.7158
GSE30999	0.7014	0.9542
GSE55447	0.6000	0.4450
GSE19429	0.5241	0.7228
GSE9006	0.7617	0.7733
GSE48350	0.6667	1.0000
GSE5281	0.6558	0.8047
GSE35978	0.4776	0.5639
GSE53987	0.5200	0.5344
GSE12288	0.5205	0.5796
GSE15852	0.7787	0.7912
GSE42568	0.6609	0.8377
GSE29431	0.6533	1.0000
GSE18520	0.7767	0.9800
GSE19804	0.7028	0.8944
GSE10072	0.8148	0.9032
GSE68571	0.7444	1.0000
Average ± SEM	0.6575 ± 0.0189	0.7952 ± 0.0337

Table 38: AUROCs of 1N9R-BRL-LC and 1N9R-BRL-BMC. The modified base model generation method generates the first model using greedy best-first BRL search on the original training data and the subsequent 9 models are random classifiers.

As theorized, the average AUROC obtained by 1N9R-BRL-LC (*average* $AUROC = 0.6575$) is now much lesser than 1N9R-BRL-BMC (*average* $AUROC = 0.7952$). Wilcoxon

signed ranks test (W-statistic = 13 is lesser than W-critical-value = 76) suggests us to reject the null hypothesis that the two classifiers have the same AUROC. 1N9R-BRL-BMC achieves a much better predictive performance than 1N9R-BRL-LC.

4.4.6 Bagged-BRL.DT-BMC compared to other state-of-the-art classifiers

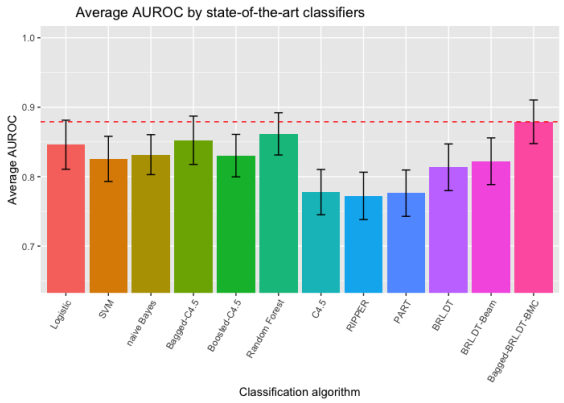
In this subsection, we compare Bagged-BRL-BMC with other state-of-the-art classifiers in machine learning. The AUROC, AUPRG and the Brier scores are plotted in Figure 15a, 15b, and 15c, respectively.

We see the impressive improvements offered by Bagged-BRL-BMC. On average, Bagged-BRL-BMC achieves the best AUROC, AUPRG, and Brier scores. Note that the only choice we made on EBRL was whether or not we use Bootstrapped samples or use Boosting procedure. EBRL models were not cherry-picked to find the model that performs optimally. In fact, the best performing model happens to be Bagged-BRL.DG-LC. So, this improvement in performance is a significant result. It is consistent with our hypotheses and is unlikely to have happened by chance.

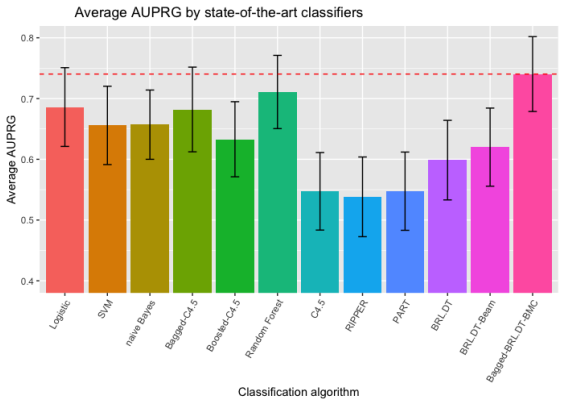
4.4.7 Experiment 2: Model visualization

The purpose of this case study is to observe the models generated by BRL and Bagged-BRL-BMC methods. Among the datasets in Table 1, GEO ID GSE19429 appears to have benefited the most from BRL methods, when compared to other classifiers. In this case study, we take a closer look at the model learned by the BRL and related methods. GSE19429 corresponds to a study on myelodysplastic Syndromes (MDS), a group of bone marrow disorders that results in a scarcity in the production of healthy blood cells in afflicted individuals. The instances are individuals with their global gene expression profiles done on the hematopoietic stem cells (HSC). The target variable of interest for this data analysis was whether or not the individual has MDS. A total of 183 instances were cases (MDS patients) and the remaining 17 individuals were normal (do not have MDS).

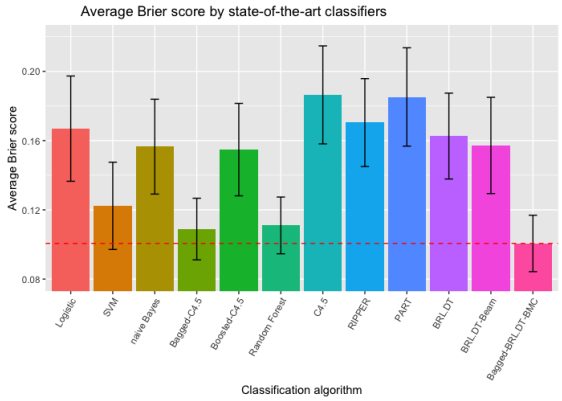
The BRL model achieves an average 10-fold cross-validation AUROC of 0.9001. The learned BRL model is shown in Figure 16.



(a) Comparing AUROCs



(b) Comparing AUPRGs



(c) Comparing Brier scores

Figure 15: Experiment 2: Average AUROCs, AUPRGs, and Brier scores by state-of-the-art classifiers, BRL.DT, BRL.DT-Beam, and Bagged-BRL.DT-BMC.

The BRL model contains 6 rules composed of 4 genes (*OR7A5*, *SORT1*, *ASB7*, and *SCAMP1-AS1*). We now explain how to read the rule model using the first rule of the model as an example. The first rule states that if the expression of gene *OR7A5* is less than or equal to 44.06 and the expression of gene *SCAMP1-AS1* is less than or equal to 72.65, then the individual is most likely normal. *Confidence* is the posterior probability of this rule. *TP* is the number of true positives, i.e., the number of instances in the dataset that agrees to both the left-hand and the right-hand side of the rule. *FP* is the number of false positives,

1. **IF** ((*OR7A5* ≤ 44.06)(*SCAMP1-AS1* ≤ 72.65)) **THEN** (Class = *Normal*)
   Confidence = 0.9286, TP = 1, FP = 0

2. **IF** ((*OR7A5* ≤ 44.06)(*SCAMP1-AS1* > 72.65)) **THEN** (Class = *Case*)
   Confidence = 0.9995, TP = 171, FP = 0

3. **IF** ((*OR7A5* > 44.06)(*SORT1* ≤ 107.06)) **THEN** (Class = *Case*)
   Confidence = 0.9737, TP = 3, FP = 0

4. **IF** ((*OR7A5* > 44.06)(107.06 < *SORT1* ≤ 143.56)) **THEN** (Class = *Normal*)
   Confidence = 0.9937, TP = 13, FP = 0

5. **IF** ((*OR7A5* > 44.06)(*SORT1* > 143.56)(*ASB7* ≤ 79.72)) **THEN** (Class = *Normal*)
   Confidence = 0.9737, TP = 3, FP = 0

6. **IF** ((*OR7A5* > 44.06)(*SORT1* > 143.56)(*ASB7* > 79.72)) **THEN** (Class = *Case*)
   Confidence = 0.9909, TP = 9, FP = 0

Variables used (4): *OR7A5*, *SORT1*, *ASB7*, *SCAMP1-AS1*.

Figure 16: BRL model for GSE19429.

i.e., the number of instances that agree with the left-hand side of the rule but disagree with the right-hand side of the rule.

The EBRL model of Bagged-BRL-BMC achieves an average 10-fold cross-validation AUROC of 0.9670. The relative variable importance computed for each variable selected by Bagged-BRL-BMC is shown in Table 39.

A part of the learned Bagged-BRL-BMC model is visualized by running BREVity on a web browser as shown in Figure 17.

The figure shows the pattern *(OR7A5 = -inf to 44.06)* is the most important pattern for the predictions made by Bagged-BRL-BMC. Indeed, this pattern covers 171 MDS patients and misclassifies 1 normal patient in the data. The edge weight of 0.42 shows the importance of the pattern, computed by simply summing up the model weights containing this pattern. We explore this important pattern further to see that some models contain the specialized pattern *(OR7A5 = -inf to 44.06) AND (KRT73 = -inf to 57.38)*. Indeed, this specialization results in covering 171 MDS patients and 0 normal patients.

With this demonstration, we hope to show the utility of BREVity in helping explain

Rank	Variable	Importance	Rank	Variable	Importance	Rank	Variable	Importance
1	OR7A5	0.4246	11	EVC2	0.1143	21	RANBP17	0.0691
2	KRT73	0.2449	12	FAM104B	0.1124	22	ANKRD36BP2	0.0691
3	HSPA2	0.1326	13	MUSK	0.1093	23	STAT1	0.067
4	FCRL1	0.1326	14	GPR176	0.1093			
5	WFS1	0.1238	15	OGFRL1	0.1011			
6	SETBP1	0.1238	16	ABCF3	0.1011			
7	PER3	0.1238	17	NCKIPSD	0.0955			
8	MMP12	0.1238	18	MME	0.0955			
9	SAMD4B	0.1143	19	MAP3K11	0.0955			
10	IGHV5-78	0.1143	20	IFIT1	0.075			

Table 39: Variable importance of the Bagged-BRL-BMC model on GSE19429 dataset.

EBRL model predictions. Typically, ensemble methods do not offer any interpretation for their predictions. With BREVity, we can interpret EBRL models as a complex decision tree. Specifically, BREVity highlights the important relationships between variables in the model of the domain. Such interpretations may help in biomarker validation.

4.4.8 Experiment 2: Conclusion

In general Bagging methods performed better than Boosting methods. One explanation is that Boosting methods are sensitive to noise from the data [Freund et al., 1996]. Gene expression datasets are notoriously noisy. This should explain why Boosting methods, regardless of the base classifiers used, performed poorly on these datasets.

Ensemble methods Bagged-BRL-LC and Bagged-BRL-BMC appear to do better than Bagged-BRL-BMA, which does not take advantage of the enriched hypothesis space offered by the ensemble. Bagged-BRL-LC and Bagged BRL-BMC perform similarly. However, when we deliberately created a scenario where sub-standard models were added to the ensemble hypothesis space, BMC performed better than LC. This is because BMC has the ability to

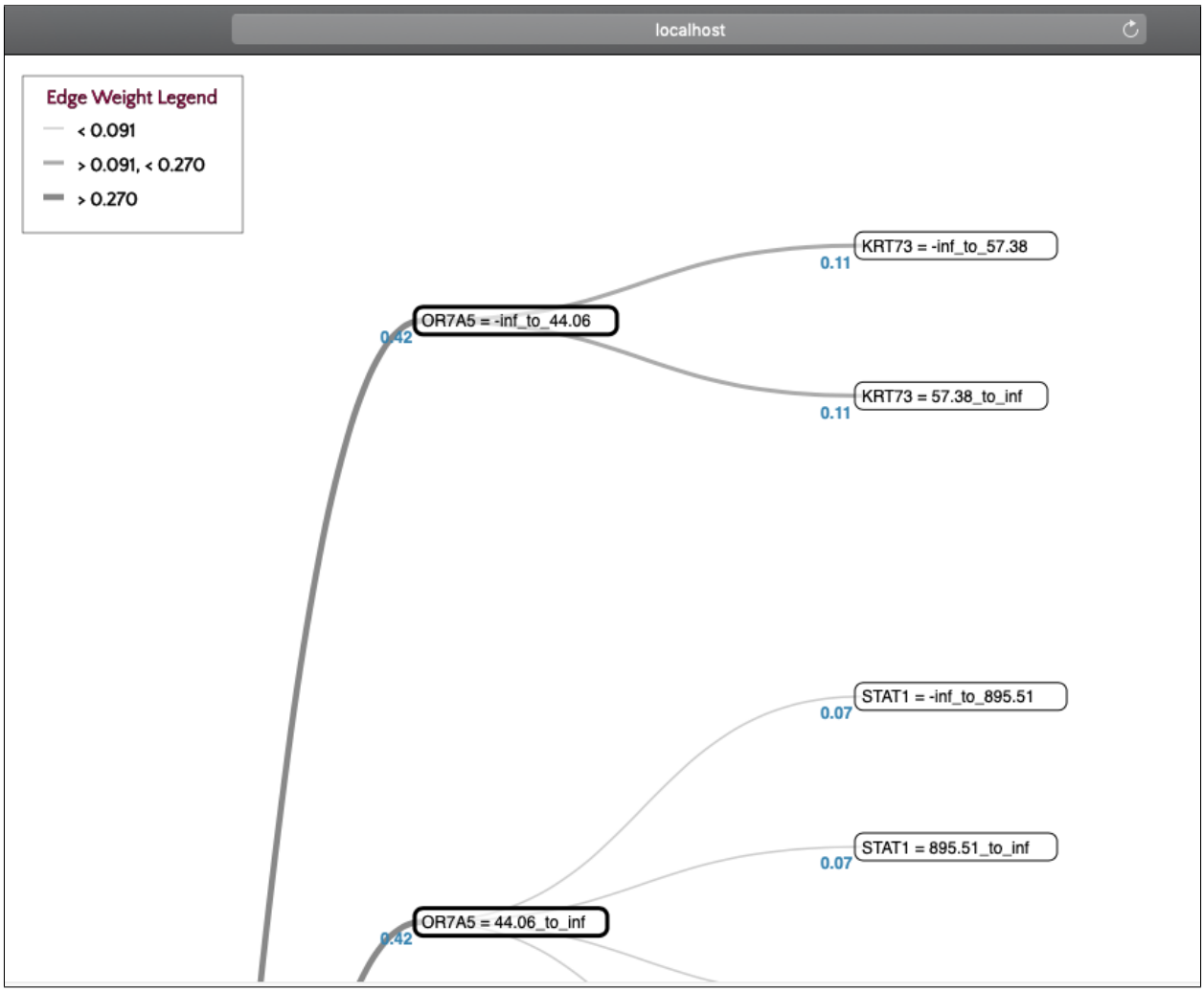


Figure 17: Bagged-BRL-BMC model for GSE19429.

learn the uncertainty in the correctness of model combination. BMC managed to learn the model weights so as to weigh the predictions from the superior model much higher than the inferior models in the ensemble. So, if we are dealing with performance sensitive applications, we recommend using Bagged-BRL-BMC as the preferred EBRL method. This is because there may be situations, where the base classifiers (hypothesis space) generation method may have included substandard models for the ensemble to combine. Instead of blindly accepting predictions from each base classifier, BMC weighs the decisions made by the reliable models higher than the less reliable models. In the worst case, where all models are more or less equally reliable, BMC would perform similar to LC (bagging).

4.5 EXPERIMENT 3: EVALUATING $\mathbf{BRL_P}$

We evaluated BRL_p using two experiments— 1) with a simulated data and 2) with a real-world lung cancer prognostic dataset. Subsection 4.5.1 shows the data generation/collection, pre-processing, evaluation metrics, and methods compared for the two experiments. Subsection 4.5.2 summarizes the results from the experiments.

4.5.1 Experiments

We were interested in the ability of BRL_p to incorporate the supplied prior domain knowledge with respect to the structure prior hyperparameter λ. Additionally, we also monitored the changes in the predictive power of the learned model resulting from the influence of the supplied prior domain knowledge. We studied the functionality of BRL_p on a simulated dataset, and then on a real-world dataset. Each is described, in detail, in the following subsections.

4.5.1.1 Simulated data study : The simulated dataset was generated from the graph in Figure 18. It has 1000 binary variables and one target variable T. Only one variable, R_{1000}, is relevant and 999 irrelevant variables, $\{I_{1\cdots999}\}$. The conditional distributions in the

graph are Bernoulli with the success parameter p depending upon the value instantiation of their parent variables. The irrelevant and relevant variable values are randomly sampled with $p = 0.5$. The T variable value is sampled with $p = 0.9$ if its parent, R_{1000}, takes the value 1, and $p = 0.1$ otherwise. We specify the edges in Figure 18 as the structure prior. We calculated the Graph Edit Distance [Riesen, 2015] (described in the next paragraph) between the model learned by BRL_p and the true data-generating model. We also measure AUROC. We evaluate these metrics over 5 runs of 10-fold cross-validation.

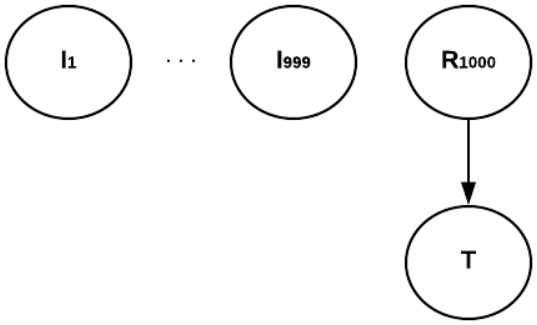


Figure 18: The data-generating graph for the simulated data.

Graph Edit Distance (GED) [Cortés et al., 2016] is a metric of similarity between two graphs. In the experiment with the simulated data, it is used to compare two constrained BNs. Specifically, it is used to measure how closely the BN learned by BRL_p, i.e., $\widehat{B_S}$ (learned by BRL_p) resembled the true BN, BS, which generated the simulated dataset (Figure 18). This was used to estimate the value of adding structure prior knowledge for model learning when the true model is available for comparison. We computed this metric using Equation 4.16.

$$d_{v_{\min}(B_S,\widehat{B_S})} = \min_{\upsilon \in \Upsilon(B_S,\widehat{B_S})} \sum_{e_i \in \upsilon} c(e_i) \tag{4.16}$$

Here, $d_{v_{\min}(B_S,\widehat{B_S})}$ is a function that returns the GED between the two BNs. A specific e_i is an edit operation to transform one graph into another. For the constrained BN we have two available edit operations— delete edge and insert edge. There is a cost $c(e_i)$ associated

with each edit operation. We set $c(e_i) = -1$, for both the edit operations. A υ is an edit path containing a sequence of edit operations to transform graph B_S into $\widehat{B_S}$. The set $\Upsilon(B_S, \widehat{B_S})$ is a set of all possible edit paths. To compute the graph edit distance, we find the edit path, υ, that minimizes the overall cost and then return this minimum cost value indicating the minimum number of operations needed to transform one graph to another. Therefore, an edit distance of 0 indicates that the predicted graph is identical to the true graph. Since the maximum parents resulted from BRL is constrained to 8 from the user parameter, the worst possible model contains all 8 irrelevant variables. So, we get $d_{\upsilon_{\min}} = 9$ (8 edge deletion operations from irrelevant variables, 1 insert edge operation to the relevant variables).

4.5.1.2 Real-world lung cancer prognostic data study : We extract a real world lung cancer prognostic dataset from Gene Expression Omnibus [Barrett et al., 2012]. We extract the dataset from a study [Lu et al., 2010] that collected both tumor and normal tissue samples from 60 non-smoking female, non-small cell lung cancer (NSCLC) patients in Taiwan. The GEO accession ID for this study is GSE19804. The data was prepared by processing it with the affy package in Bioconductor in R. We normalize the data using Robust Multichip Analysis (RMA) for background correction, quantile normalization, and probe summarization. Multiple probes can map to a single gene. In the final dataset, we would like to have just one random variable representing a unique gene. Among the multiple probes that map to a single gene, we chose the probe with the largest inter-quantile range to represent the gene. We also extracted the tissue phenotype (*tumor* or *normal*) for each sample and add to this dataset. The outcome variable of interest was this tissue phenotype. After this pre-processing step, we were left with 16382 genes. So, the final dataset for our analysis had 16382 variables and 120 instances.

To specify a structure prior, we look into the literature for relevant, known prognostic markers for our study population. Epidermal growth factor receptor (EGFR), a receptor tyrosine kinase is prognostic marker known to be frequently over-expressed in NSCLC [Bethune et al., 2010]. In NSCLC patients, [Shigematsu et al., 2005] observed that EGFR domain mutations are statistically significantly more frequent in women than men (42% versus 14%), in adenocarcinomas than other histologies (40% verses 3%), in non-smokers than smokers (51%

verses 10%), and in East Asians than other ethnicities (30% verses 8%); all with a p-value of < 0.001. This description is very similar to the subjects in the dataset we are studying. So, we specify EGFR in the positive edge set of the structure prior of BRL_p.

We evaluate on two metrics— 1) Prior Frequency (PF), the fraction of models that contains EGFR, we need the true data generating graph to calculate Graph Edit Distance, which is not feasible for real-world problems; and 2) AUROC. We calculate them over 5 runs of 10-fold cross-validation. We study the effect of the hyperparameter $\lambda = \{0, 1, 2, 4, 6, 8, 10, 20\}$. Finally, we compare their performance to some state-of-the-art classifiers.

4.5.1.3 Methods compared : We again evaluated BRL_p here. We set its of maximum conjuncts to 8. We evaluated the effect of the hyperparameter λ by assigning it values of $\lambda = \{0, 1, 2, 4, 6, 8, 10, 20\}$. The value $\lambda = 0$ represents the baseline model of BRL with no structure priors. We included $\lambda = 20$, to study the scenario where the structure priors overwhelmingly dominates the likelihood score. Additionally, we compared these models with some state-of-the-art classifiers including three rule learning classifiers namely— C4.5 [Quinlan, 2014], RIPPER [Cohen, 1995], and PART [Frank and Witten, 1998]; and three other popular SOTA classifiers namely— Random Forests [Breiman, 2001], naïve Bayes [John and Langley, 1995], and Support Vector Machines [Platt, 1999]. C4.5 is a popular decision tree learning algorithm, where each path of the decision tree can be interpreted as rules. RIPPER (Repeated Incremental Pruning to Produce Error Reduction) is a propositional rule learning algorithm that uses a divide-and-conquer strategy during model training. PART is a rule learning method that combines the approaches of both C4.5 and RIPPER by building partial decision trees, inferring rules from the trees, and using a divide-and-conquer strategy to build the rule model. Random Forest is an ensemble learning method that learns a number of decision trees during training, and combines predictions from them during inference. The naïve Bayes classifier is a simple probabilistic classifier that learns a network with strong independence assumption between the variables, and uses the Bayes theorem for inference from the learned network. Support Vector Machines is an algorithm that learns a hyperplane function to differentiate the classes in the problem space. We ran these classifiers from the Weka [Frank et al., 2016] workbench (version 3.8.1) using the default parameters for each

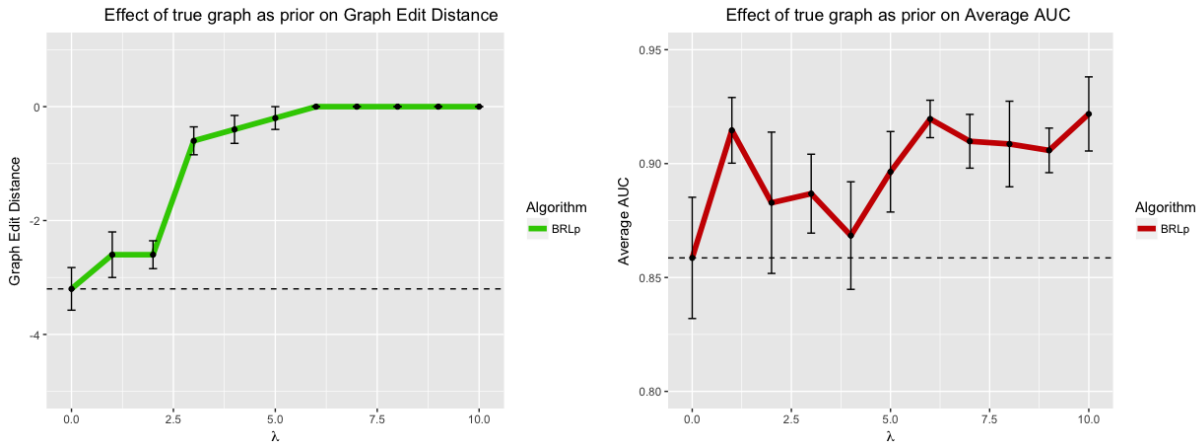
classifier.

4.5.2 Results

We take a look at the results from our experiments to learn the relation between the λ hyperparameter and the degree of incorporation of prior knowledge in BRL_p's model learning process.

4.5.2.1 Simulated data study results : The results from the 5 runs of 10-fold cross-validation are summarized in Figure 19. In Figure 19a, the x-axis shows the various values of the hyperparameter λ and the y-axis shows the average GED. This average is obtained across the 10-folds of each run, and then averaged across the 5 runs. Each data-point in the graph is this average, and the error bars represent the standard error of mean. The dotted line shows the value of BRL_p with $\lambda = 0$, which as we mentioned earlier is the same as BRL, where we use uninformative priors. We see that even with $\lambda = 1$, we see the effect of the structure priors in bringing the learned model closer to the data-generating model. We see a sharp gain of GED from $\lambda = 2$ to 3. For $\lambda \geq 6$, BRL_p returns the true data-generating model specified by the structure priors. This shows that BRL_p effectively and correctly incorporates the specified domain knowledge. The degree of incorporation is controlled by λ.

Figure 19b displays the average AUROC. The overall trend is a gain in AUROC but the trend is noisy, especially with low λ values when the $GED > 0$. This region indicates models that pick up irrelevant variables, which are spurious and are associated with T, by chance. Their AUROC fluctuate a lot because random associations are found. When $\lambda \geq 6$, where the GED reaches the perfect 0, we see a rise in AUROC. The noise reduces in this region of the graph. Random samplings from our simulation generate slightly different values of the parameters, which are reflected in the fluctuations here. So, from the AUROC graph we see a gradual gain in predictive performance with the incorporation of prior knowledge of the truth.



(a) Different values for structure prior hyperparameter λ compared to the Graph Edit Distance between the true data-generating graph and the BRL_p model.

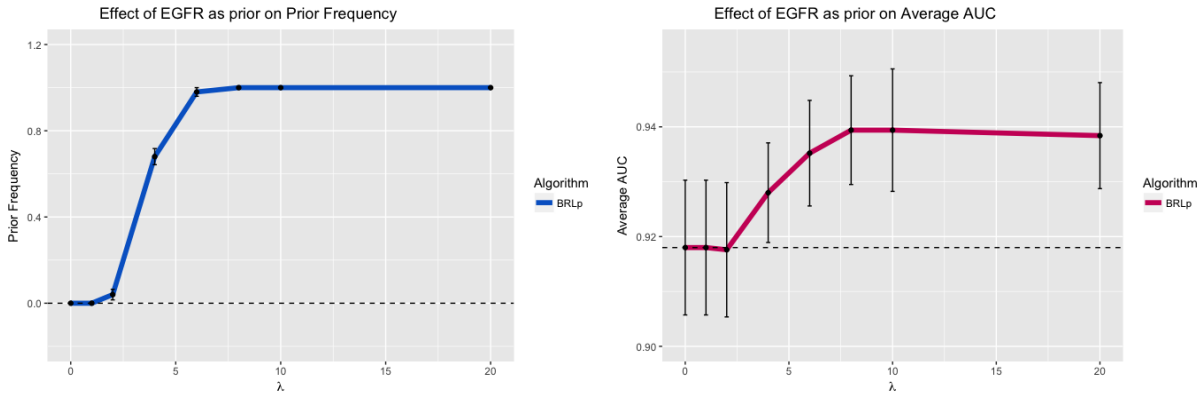
(b) Different values for structure prior hyperparameter λ compared to the AUROC of BRL_p model.

Figure 19: Simulated data analysis by BRL_p.

4.5.2.2 Real-world lung cancer prognostic data study results : The results from the 5 runs of 10-fold cross-validation on the real-world lung cancer prognostic dataset are summarized in Figure 20. We specified the structure prior of an edge between EGFR and the outcome Class variable to be present. We alter the values of λ and observe its effect on the learned model. Figure 20a, shows the effect of the different values of λ on PF, the fraction of models that contain EGFR. From $\lambda = 2$ to 6, we see a steep gain in PF. For $\lambda \geq 8$, EGFR is present in every learned model. This again shows that BRL_p effectively incorporates the specified prior knowledge and the λ hyperparameter allows the user to determine the degree of incorporation of this knowledge by BRL_p.

Figure 5b, shows the gain of average AUROC across 5 runs of 10-fold cross-validation. We observe a steady gain of AUROC for $\lambda > 2$. For $\lambda \geq 8$, the AUROC gain tapers off. The results show that the EGFR prior knowledge helped improve the AUROC of BRLp.

Finally, we compare two BRL_p models with state-of-the-art classifiers using average AUROC achieved across 5 runs of 10-fold cross-validation. The two BRL_p models are— 1) with



(a) Different values for structure prior hyperparameter λ compared to Prior Frequency (fraction of BRL_p models with EGFR.)

(b) Different values for structure prior hyperparameter λ compared to the AUROC of BRL_p model.

Figure 20: Real-world lung cancer prognostic data analysis by BRL_p.

$\lambda = 0$, which represents the baseline BRL model with uninformative priors, and 2) with $\lambda = 8$ incorporating EGFR into the structure prior, which achieved the highest average AUROC of 0.935. The state-of-the-art classifiers compared are C4.5, RIPPER, PART, Random Forests, naïve Bayes, and Support Vector Machines. This comparison is done in Figure 21.

The first two bars in Figure 21 are BRL_p algorithms, BRL_p with $\lambda = 0$ is indicated as BRL, and then BRL_p with $\lambda = 8$. We see a gain in performance from incorporating EGFR as structure priors. The next three bars— C4.5, RIPPER, and PART are rule learning models, which are human readable. C4.5 is a decision tree learning algorithm. RIPPER and PART are rule learning algorithms. We notice that these three algorithms perform worse than both BRL_p algorithms in this dataset. The last three bars in Figure 21 are— Random Forest, naïve Bayes, and Support Vector Machines. These are examples of complex models that use all variables in the dataset to generate a classifier. It is not easy to explain the reasoning behind their predictions. But all three algorithms here outperform BRL_p on this dataset. This comparison shows the trade-off of predictive performance and interpretability. On this dataset, BRL_p offers an rule learning model that outperforms other popular rule learning

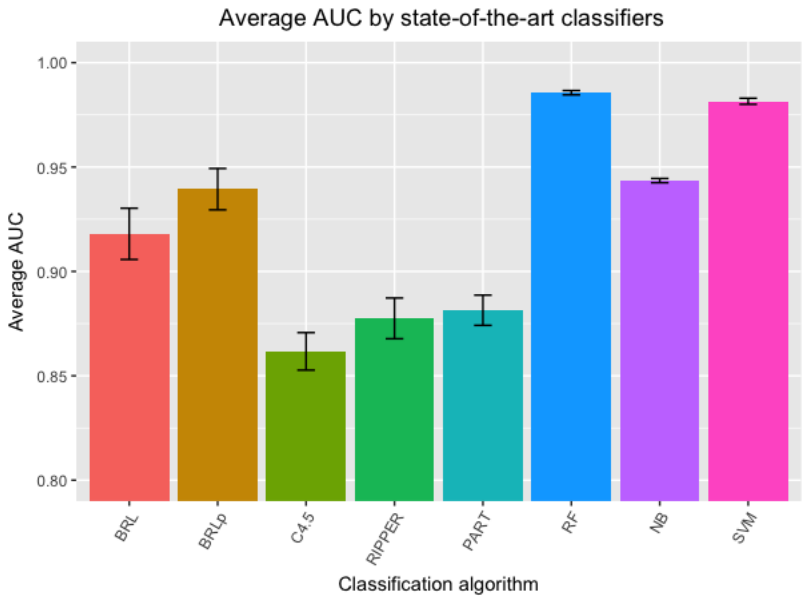


Figure 21: Comparison of AUROC achieved by BRL_p with state-of-the-art classifiers.

models but does not perform as well as the other, less-interpretable SOTA models.

4.5.3 Experiment 3: Conclusion

We demonstrated the ability of BRL_p to incorporate this knowledge on simulated data and a real-world lung cancer prognostic dataset. We observed that the λ hyperparameter allows us to control the degree of incorporation of prior knowledge. This parameter can be helpful if we are uncertain about the specified prior knowledge. We also observed that dataset relevant prior knowledge could sometimes help improve the predictive performance of BRL_p.

4.6 EXPERIMENT 4: EVALUATING BRL-KD

We evaluate BRL-KD using a real-world dataset collected to study the features that can help discriminate individuals at risk of developing cardiovascular diseases from those who will not develop the disease. In this section, we do not analyze this data in depth. Instead, the focus

will be to study the functionality of BRL-KD algorithm. As a result, we do not make any biological inferences from the models learned in this section. For an in-depth analysis of this cardiovascular disease dataset, including modeling with BRL, and inferring biological significance of the induced model, please refer to Chapter 5.

Subsection 4.6.1 contains the description of the experimental design for studying BRL-KD, including— data collection, pre-processing, evaluation metrics used, and methods compared. Subsection 4.6.2 summarizes the results from the experiments with BRL-KD.

4.6.1 Experiment design

The goal of the experiment here was to study the changes in behavior of models learned from BRL-KD, while tuning the hyperparameter λ. Specifically, we wanted to observe the changes in two metrics— 1) a clinical relevance metric as defined by a utility function (based on the clinical application of the model), and 2) a predictive performance metric using AUROC.

In this experiment, we use a real-world dataset collected to learn differential features between individuals who are at risk to develop cardiovascular disease and those who are unlikely to develop the disease in the near future. We assume that the goal of modeling this dataset using BRL here is to develop a medical screening method. Screening methods help identify individuals that are at risk of developing cardiovascular disease in the near future.

Many definitions of clinical relevance is pertinent to screening methods including test specificity, cost-efficiency, and non-invasiveness. Specific tests minimize the chance of incorrectly giving a negative test result to an individual who would eventually develop the disease. Cost-efficient tests are cheap and effective that enable us to compute the risk of a larger population when compared to a more expensive test. Non-invasive tests do not require the use of medical equipment used by medical practitioners to physically enter an individual's body. In this experiment, we will only focus on cost-effectiveness as a measure of clinical relevance. The same ideas developed in this study can be extended to other definitions of clinical relevance.

We assume that the most cost-effective BRL model is the clinically most relevant model. Cost-effective models have both good predictive performance and are cheaper than the clin-

ical standard currently used in practice. In general, cheaper models are clinically more relevant given that there is no significant loss of predictive performance.

In subsection 4.6.1.1, we provide some background on the data collection process and the description about the study cohort. We also describe the pre-processing performed on the dataset to prepare it for analysis using BRL-KD. In subsection 4.6.1.3, we describe the two metrics we monitor in BRL-KD i.e., the clinical relevance using cost of the model and predictive performance using AUROC of the model. We summarize the results from the experiment in section 4.6.2.

4.6.1.1 Data collection and pre-processing

Heart SCORE (**S**trategies **C**oncentrating **O**n **R**isk **E**valuation) is an ongoing longitudinal prospective study, initiated in 2003, that follows 2000 middle-aged, primarily black and white individuals from Allegheny County, Pennsylvania, USA [Bambs et al., 2011]. We consider two types of variables measured from individuals in the study at the time of enrollment. They are— clinical and metabolic variables.

The clinical variables include— age, sex, race, patient medical history, physical examination, medications, Vertical Auto Profile test for lipids, finger stick tests, and questionnaires about the individual's physical activities, lifestyle markers, social network, diet, sleep quality, and various psychological questionnaires. The data has a total of 654 such clinical variables. The metabolic variables had 1228 biochemicals including 893 named and 335 unnamed biochemicals. These biochemicals include xenobiotics, cofactors, vitamins, and metabolites from amino acid, lipid, carbohydrate, and energy metabolism. Out of the 2000 study participants, only 1901 had metabolites measured for them. So the dataset had 1901 instances.

The data had to be pre-processed to prepare it for analysis using BRL. We first had to clean the data. The original dataset had 654 clinical variables. This step included defining discrete value bins for certain variables (e.g race), correcting typographical errors, and removed redundant or unreliable measurement variables in the dataset in accordance to the dataset manual. After discussions with an expert we also dropped the variable indicating the patient history of having undergone percutaneous coronary intervention as it may act as a confounding variable. We also removed variables indicating dates as they are unlikely to

help as predictive variables. The cleaned dataset had 608 variables. For this data analysis, we assume that the missing values are Missing Completely at Random (MCAR). Under the MCAR assumption, when the variable with missing values have $\leq 5\%$ of the values missing, we can impute a single complete dataset, with minimal bias, using median/mode value imputation. In this analysis, we dropped the variables with $> 5\%$ of its values missing. The rest of the variables had missing values imputed with median/model imputation.

We set the outcome variable of interest as Major Adverse Cardiac Event or MACE, here defined to include any of— Cardiac death, myocardial infarction (MI), acute ischemic stroke (AIS), or revascularization.

The final dataset had 1617 candidate predictive variables including 389 clinical variables and 1228 metabolic variables. There are 1901 instances with 101 positive cases of individuals who eventually developed a MACE outcome by 2018. The remaining 1800 individuals were labeled negative.

We split the dataset randomly into 70% training data and 30% test data. We will use the training data to choose the correct value for hyperparameter λ by observing the average cost and average AUROC achieved by the specific value of λ. The average of the metrics is computed over 10-fold cross-validation. We then choose the λ with the ideal trade-off between cost and AUROC, and use that to model the training dataset. We observe these models and evaluate their performance on the held-out test set.

Since BRL can only handle discrete-valued data, we discretize the continuous valued variables from the Heart SCORE data under cross-validation. When the training data is split into 10-folds, in each iteration of the cross-validation, we discretize on the training fold using the supervised minimum description length (MDL) principle method [Fayyad et al., 1993] and use the learned bins to discretize the test dataset. Once we detemine the ideal value for λ from the cross-validation experiment, we discretize the 70% training data using the MDL method and use the learned bins to discretize the 30% test data.

4.6.1.2 Methods compared : We will study the change in behavior of BRL-KD with the change in hyperparameter λ. We first need to encode clinical relevance in terms of cost-efficiency, into the BRL-KD heuristic score. We reproduce the heuristic score for BRL-KD,

from Equation 3.20, below.

$$P(B_S, \Psi, D; \kappa, \alpha) = p(B_S) \cdot p(\Psi|B_S) \cdot \prod_{j=1}^{q_Y} \frac{\Gamma(\frac{\alpha}{q_Y})}{\Gamma(N_j + \frac{\alpha}{q_Y})} \prod_{k=1}^{r_Y} \frac{\Gamma(N_{jk} + \frac{\alpha}{r_Y q_Y})}{\Gamma(\frac{\alpha}{r_Y q_Y})}$$

We encode the term $p(\Psi|B_S)$ with a distribution that represents our belief about which of the models in the hypothesis space is more cost-efficient. We do this using Equation 4.17, similar to how we encoded informative priors for model validity using BRL_p (see 3.3.2).

$$p(\Psi|B_S; \lambda, w) \propto exp\Bigg[\lambda \cdot \Big(w_{PE} \cdot \mathbb{1}_{\{|E(B_S) \cap E_{PE}| \neq 0\}} + w_{VAP} \cdot \mathbb{1}_{\{|E(B_S) \cap E_{VAP}| \neq 0\}} + w_{FS} \cdot \mathbb{1}_{\{|E(B_S) \cap E_{FS}| \neq 0\}} + w_{MB} \cdot \mathbb{1}_{\{|E(B_S) \cap E_{MB}| \neq 0\}}\Big)\Bigg] \quad (4.17)$$

In this equation, λ represents the relative importance of considering cost as opposed to the likelihood term (that tries to optimize predictive performance). The set of weights, $w = \{w_{PE}, w_{VAP}, w_{FS}, w_{MB}\}$, represent the cost of physical examination (w_{PE}), cost of Vertical Auto Profile test for lipids (w_{VAP}), cost of finger stick tests (w_{FS}), and the cost of running a full metabolome profile of 1228 biochemicals (w_{MB}).

We want to emphasize that none of the values encoded here represented reality and were not done consulting a medical practitioner. Instead, they were estimated just to demonstrate BRL-KD as a proof of concept. In reality, the cost can be influenced by many factors including time, insurance, medicare, and the location of point-of-care. For now, we assume that the cost associated with each biomarker presented in the next paragraph is correct and we observe the function of BRL-KD in light of having specified such values in practice.

The Heart SCORE dataset contains clinical and metabolic variables. For all the demographic and questionnaire variables, we set the cost of the biomarkers to \$0. We assume that it is free to obtain markers that can be supplied by asking the individual or can be obtained from their medical history. We set the cost of a physical exam (w_{PE}) to \$146 (estimated from the cost of a 15 minute physical per UPMC Presbyterian/Shadyside hospital's Charge Description Master file). by setting the value of $w_{PE} = 146$, we state that if the BRL model requires the use of any one or more variables from the physical examination (e.g., height,

weight, or body mass index), the model would incur a cost of \$146. We then set the cost of Vertical Auto Profile (VAP) test for lipids w_{VAP} to \$2813 (again estimated from UPMC Presbyterian/Shadyside hospital's Charge Description Master file). The cost of finger stick tests is assumed to be $w_{FS} = \$1$ (estimated from the equipments cost from Cholestech LDX assuming the analyzer is available). The cost of the whole metabolic panel is assumed to be, $w_{MB} = \$1000$ (an arbitrary estimate).

The costs range in thousands leading to this probability ranging widely. To prevent this, we scale the cost using min-max scaling to have the cost values between the range 0 and 1. We also take the negative value of the scaled cost in order to convert this utility function into a maximization problem.

4.6.1.3 Evaluation metrics

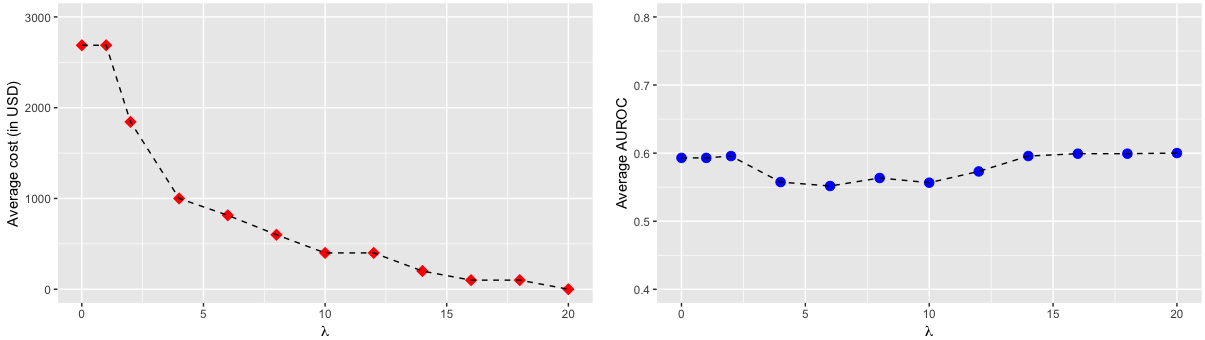
We observe the change in the BRL-KD model behavior under the changing values of the λ hyperparameter with two metrics— 1) clinical relevance using the total cost of the model, and 2) predictive performance using AUROC.

The cost metric is simply the sum of the costs of each marker selected by the BRL classifier. If the BRL model would select any variable from one of the sets with specified costs, the model would incur a cost as specified by the associated weight. For example, if the BRL model selects 2 metabolic variables, it would still only incur the cost of w_{MB} once, since we assume that the whole metabolic panel is run. Again, we emphasize that our description of utility is meant to merely depict a real-world application and that our choice may not reflect reality. Our goal is to evaluate the use of BRL-KD in a possible real-world problem.

4.6.2 Experiment 4: Results

On the 70% training data, we perform 10-fold cross validation. In each fold, the training data is split into 90% train and 10% development data. The 90% train is used to run BRL-KD using the heuristic score in Equation 4.17. We do this for the following different values of the hyperparameter $\lambda = \{0, 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$. We plot the average cost and average AUROC across 10 fold cross validation in Figure 22a and Figure 22b, respectively.

We observe that the average cost steadily declines with the increase in the value of λ.



(a) Change in average cost.

(b) Change in average AUROC.

Figure 22: Average cost and AUROC, over 10 folds, of the models learned under different values of hyperparameter λ.

The average AUROC is more or less steady with the change in λ. The value of $\lambda = 0$ sets the whole expression of $p(\Psi|B_S)$ in Equation 3.20 to 1. As a result this value corresponds to the baseline of not using BRL-KD. The average cost of the BRL model without BRL-KD is \$2687.8 and the associated average AUROC is 0.5930.

As we increase the value of λ, the BRL-KD model starts to pick more cost-efficient markers. This means markers that are of similar or slightly poorer quality but at a cheaper cost. As we increase the λ value, we get cheaper and cheaper BRL-KD models. Value $\lambda = 4$ corresponds to an average cost of \$1000 and average AUROC of 0.5575. This λ value presents with a cheaper option at the loss of some predictive power. Value $\lambda = 10$ corresponds to an average cost of \$400.2 and average AUROC of 0.5566. Value $\lambda = 20$ corresponds to an average cost of just \$0.2 and average AUROC of 0.6001. This λ value results suggest that there are similarly good AUROC performing models by simply using variables that are available for free (according to our definition).

We now look at the models learned on the overall 70% training dataset, for $\lambda = \{0, 4, 10\}$, to observe the BRL-KD models. Note that we also looked at models with greater values of λ but those models remained consistent after $\lambda = 10$. The value $\lambda > 10$ corresponded

with models with a total cost of \$0 so there was nothing to optimize further using the λ hyperparameter. We estimate the AUROC by evaluating the model on the held-out 30% test dataset. Together, the various BRL-KD models, offering trade-offs between cost and AUROC, present a Pareto set of solutions.

The resulting BRL-KD model cost and AUROC is shown in Figure 23. We see that with increasing λ, we get cheaper models but we get them at a loss of AUROC performance on the test set.

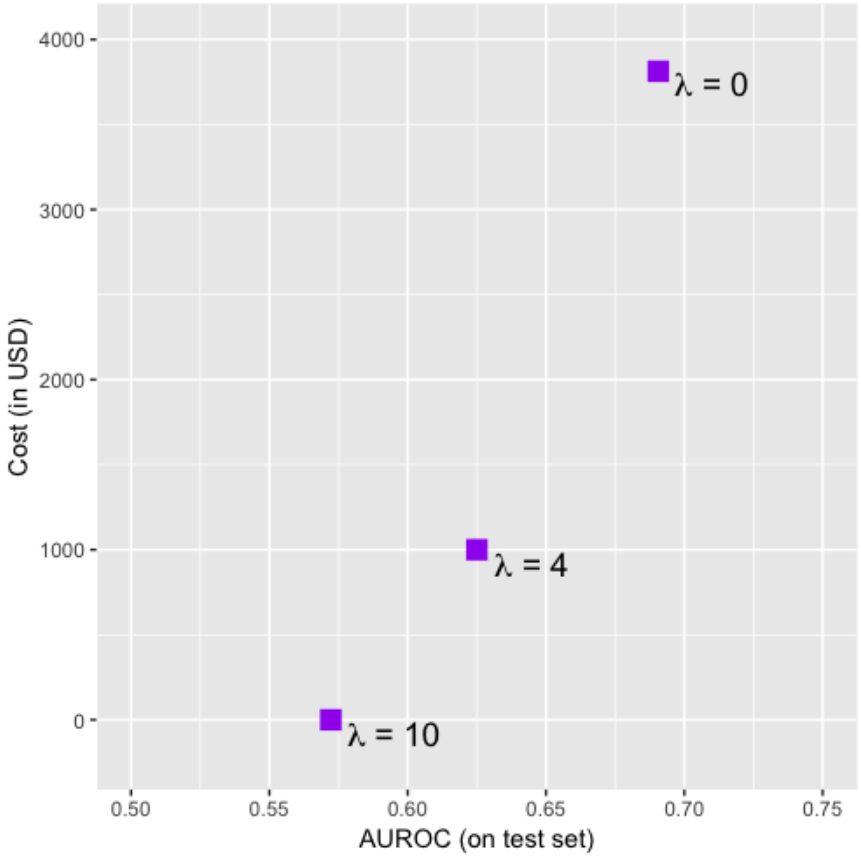


Figure 23: BRL-KD model with $\lambda = \{0, 4, 10\}$.

Figure 24 shows the BRL-KD rule model we get when $\lambda = 0$. This sets the whole expression of $p(\Psi|B_S)$ in Equation 3.20 to 1. This means no attempt is made to search for cost-efficient markers in the dataset and is equivalent of just running plain BRL.

This model picks up a metabolic variable *35137* (cost = \$1000), a variable from the VAP test *LAB_HDL2A* (cost = \$2813), and a demographic variable *SCR_AGE* (cost = \$0). So,

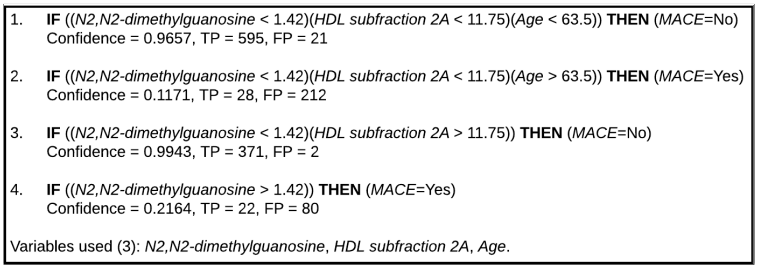


Figure 24: BRL-KD model with $\lambda = 0$.

the total cost of this baseline model is \$3813 and the AUROC performance of this model on the held-out test set is 0.6906.

Figure 25 shows the BRL-KD rule model we get when $\lambda = 4$.

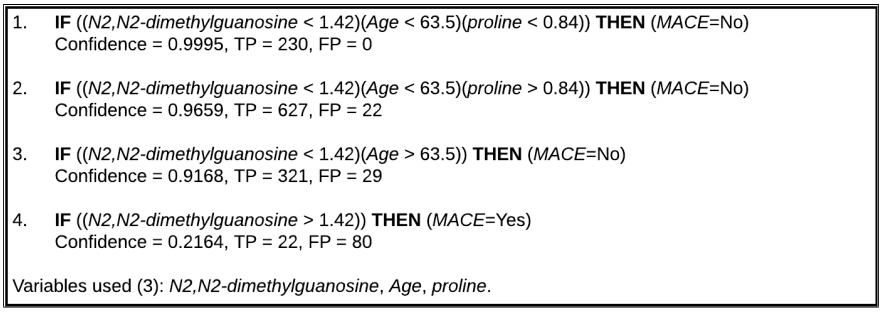


Figure 25: BRL-KD model with $\lambda = 4$.

This model picks up two metabolic variables *35137* and *1898* (cost = \$1000) and a demographic variable *SCR_AGE* (cost = \$0). So, the total cost of this cost efficient model is brought down to \$1000 and the AUROC on the test set is 0.6250. Trying to come up with more cost-efficient markers, BRL-KD loses the most expensive marker from the VAP test and substitutes it with a metabolic variable. However, we see that this also leads to a loss of AUROC on the test set.

Figure 26 shows the BRL-KD rule model we get when $\lambda = 10$.

This model uses a single demographic variable that is available for free, *SCR_AGE*. This leads to a further saving of cost down to \$0. However, we do that by compensating on the AUROC performance of 0.5722.

1. **IF** ((*Age* < 63.5)) **THEN** (*MACE*=No)
   Confidence = 0.9693, TP = 892, FP = 28

2. **IF** ((*Age* > 63.5)) **THEN** (*MACE*=Yes)
   Confidence = 0.11, TP = 45, FP = 366

Variables used (1): *Age*.

Figure 26: BRL-KD model with $\lambda = 10$.

4.6.3 Experiment 4: Conclusion

We saw that increasing the λ hyperparameter helped us specify a range of possible trade-off values between clinical relevance and predictive performance. We used cost as the utility function measuring clinical relevance. Using BRL-KD, we tuned the hyperparameter λ to study a range of different trade-offs between cost and predictive performance.

We saw that by decreasing the cost of the model, we lost predictive performance in terms of AUROC on the test set. Note that not all clinical relevance utility functions may conflict with predictive performance. For example, while venipuncture is more costly than finger stick tests to measure, say blood glucose, they generally have similar measurement accuracy. So, both lead to the same predictive performance. If BRL had chosen venipuncture variable by chance, BRL-KD here would have helped us substitute finger stick test to obtain a cheaper test with similar predictive performance.

The ultimate decision on λ requires us to know a few things about the current clinical standard being used in practice. This will help us determine from the data if there exists a λ value that gives us a model that is clinically more relevant than the one being used in medical practice today. However, from the experiments it appears to be clear that BRL-KD is a useful tool to help us find clinically more relevant models and encourages a real-world application.

5.0 BRAID SYSTEM FOR PREDICTING CARDIOVASCULAR DISEASE RISK

In this chapter, I apply the developed BRL algorithms, within Bayesian Rules for Actionable Informed Decisions (BRAID), for the task of assisting clinicians in computing the risk of developing cardiovascular disease in an individual. In addition to calculating the risk, BRAID will also provide a graphical explanation for the computed risk value. The intelligent core of the system is the BRL suite of algorithms that have been described in Chapter 3.

In this chapter, section 5.1 introduces the reader to the challenge of the clinical management of cardiovascular diseases in the general population. Some of the currently used, clinical state-of-the-art methods are also discussed. We then look at Heart SCORE, a prospective study developed to help improve our understanding of cardiovascular diseases. Section 5.2 will begin with an explanation of how we collected and pre-processed retrospective data from the Heart SCORE study for analysis using BRL. The section also describes the experimental design, evaluation metric, and the various predictive models being evaluated in this study. This section also describes a method we implemented to perform functional analysis (Metabolite set enrichment analysis) of the biomarkers selected from the best performing model. Section 5.3 shows the results from the experiments. Section 5.4 explains how the BRAID concept framework architecture could be deployed in practice. We conclude this case study in section 5.5.

5.1 INTRODUCTION

Globally, in 2016, $\approx$ 17.6 million (95% CI, 17.3-18.1 million) deaths were attributed to cardiovascular diseases (or CVD, here defined as a set of diseases comprising of coronary heart disease, heart failure, stroke, and hypertension) [Benjamin et al., 2019]. This is an increase of 14.5% (95% CI, 12.1%-17.1%) from 2006. Heart diseases remain the leading cause of death in the United States with $2,744,248$ new deaths, in 2016. According to NHANES (National Health and Nutrition Examination Survey) 2013-2016 data, the prevalence of CVD in adults (age $\geq$ 20 years), in the United States, is 48.0% overall (121.5 million in 2016). Excluding hypertension, the CVD prevalence is still 9.0% overall (24.3 million in 2016). The estimated average annual cost, direct and indirect, of CVD and Stroke was $351.2 billion, in 2014 to 2015, in the United States. Better clinical management of CVD is critical to help bring down its immense burden, in terms of the disease mortality, morbidity, and its healthcare-related costs.

CVDs can broadly encompass the diseases of the heart, vascular diseases of the brain, and diseases of blood vessels [Mendis et al., 2011]. CVDs involving the heart include— heart failure, cardiomyopathy, congenital heart disease, heart arrhythmia, and congenital heart disease. Vascular diseases under CVD involving the brain include— cerebrovascular diseases like stroke. Vascular diseases involving the blood vessels include— coronary heart disease (CHD) and peripheral artery disease. Atherosclerotic cardiovascular disease (ASCVD) are diseases that are specifically of atherosclerotic origin (i.e., as a result of plaque clogging the arteries) and includes CHD, stroke, and peripheral artery disease.

The clinical management of cardiovascular diseases today is primarily guided by cardiovascular risk scores. Physicians use these risk scores for screening. In medicine, screening is a process of identifying individuals in a population with a high risk of developing a disease. Medical screening methods are not meant to be diagnostic of the disease and hence often suffer from high false positives. The three popularly used risk scores for cardiac health are— 1) Framingham Risk Score (FRS) [Dagostino et al., 2008], 2) Reynolds Risk Score (RRS) [Ridker et al., 2008, Ridker et al., 2007] , and 3) Pooled Cohort Risk Equations (PCRE) [Goff et al., 2014]. Each of these risk scores take in certain characteristics of an

individual that is known to be associated with the disease, known as risk factors, and based on these characteristics, assign them a score as a percentage. This score is the absolute risk for developing certain cardiac events over a specified period of time (typically the next 10 years). FRS and RRS predicts the risk of developing CHD and CVD, while PCRE assesses the risk ASCVD. [Goff et al., 2014] specify ASCVD events as— coronary death, CHD death, nonfatal myocardial infarction, and fatal or non fatal stroke. The original reason for the development of RRS was to develop separate risk scores for men [Ridker et al., 2008] and women [Ridker et al., 2007]. However, FRS was later developed to include gender as one of the risk factors. PCRE was developed to include race as a risk factor in assessing cardiac event risk. All these risk scores use commonly known CVD risk factors including total cholesterol, HDL cholesterol, systolic blood pressure, and smoking.

The Heart SCORE (**S**trategies **C**oncentrating **O**n **R**isk **E**valuation) study is an ongoing longitudinal prospective study, initiated in 2003, which follows 2,000 middle-aged, primarily black and white individuals from Allegheny County, Pennsylvania, USA [Bambs et al., 2011]. The main goal of the study was to improve CVD risk stratification, identify racial disparities, and evaluate mechanisms for population differences in CVD. To this end, the study measures potential cardiovascular risk factors including demographic (e.g. age, gender, race, etc.), clinical (e.g. LDL cholesterol, HDL cholesterol, diabetes, smoking, etc.), metabolic, genetic risk factors among other factors in the study participants.

In this chapter, we will use the promising methods developed in this dissertation study to learn classifiers that can find biomarkers that discriminate individuals who are likely to develop CVD in the near future, from those who are not likely to develop the disease. By doing so, we align with one of the goals of the Heart SCORE study— to evaluate mechanisms for population differences in CVD. So, from the data analysis performed in this chapter, we aim to discover biomarkers that provide insight into the CVD mechanism as opposed to developing a medical screening method like the cardiovascular risk scores mentioned earlier.

5.2 MATERIALS AND METHODS

We will formulate the problem of— finding differential biomarkers between two classes of people, 1) those who are likely to develop CVD in the near future and 2) those who are not likely to develop CVD in the near future— as a supervised classification problem in machine learning. Here, we will use machine learning algorithms to learn classifiers that discriminate the two classes. Machine learning classifiers use statistical methods to learn a mathematical function of predictive variables to help discriminate the two classes. They learn this mathematical function from the data.

In subsection 5.2.1, we describe the dataset collected for data analysis here. We will also describe the pre-processing done to prepare the dataset for data analysis with machine learning classifiers. Subsection 5.2.2 outlines the experimental design used to evaluate the different classifiers. Subsection 5.2.3, lists and describes the machine learning classifiers that will be compared and evaluated in the experiment. In addition to the machine learning classifiers, we will use the cardiac risk scores as a classifier to use as a baseline to estimate the current clinical standard in use. Subsection 5.2.4 describes our implementation of enrichment analysis, a type of functional analysis of metabolic markers that were selected by our models.

5.2.1 Heart SCORE dataset and pre-processing

There are 2000 individuals enrolled into the Heart SCORE study. For this data analysis task, we will only consider two types of variables measured from individuals in the study at the time of enrollment (i.e., baseline measurements). They are— clinical and metabolic variables.

The clinical variables include— age, sex, race, patient medical history, physical examination, medications, Vertical Auto Profile test for lipids, finger stick tests, and questionnaires about the individual's physical activities, lifestyle markers, social network, diet, sleep quality, and various psychological questionnaires. The data has a total of 2000 individuals or instances, and 654 clinical variables. The metabolic variables had 1228 biochemicals including 893 named and 335 unnamed biochemicals. These biochemicals include xenobi-

otics, co-factors, vitamins, and metabolites from amino acid, lipid, carbohydrate, and energy metabolism. Out of the 2000 study participants, only 1901 had their metabolic profile measured. The metabolic dataset had 2000 instances for 1901 unique individuals from the study. A total of 99 instances in the metabolic dataset were technical replicates. We averaged the technical replicates. The final metabolic dataset had 1901 instances and 1228 variables (metabolites measured).

The data had to be pre-processed to prepare it for analysis using BRL. We first had to clean the data. The original dataset had 654 clinical variables. This step included defining discrete value bins for certain variables (e.g race), correcting typographical errors, and removal of redundant and unreliable measurement variables in the dataset (in accordance to the dataset manual). We also removed variables indicating dates as they are unlikely to help as predictive variables. The cleaned dataset had 608 clinical variables. For this data analysis, we assumes that the missing values were Missing Completely at Random (MCAR). Under the MCAR assumption, when the variable with missing values have $\leq 5\%$ of the values missing, we can impute a single complete dataset, with minimal bias, using median/mode value imputation. In this analysis, we dropped the variables with $> 5\%$ of its values missing. The rest of the variables had missing values imputed with median/model imputation.

We defined the outcome variable of interest as Major Adverse Cardiac Event or MACE. It includes any individual who suffered from either of the following events by 2018— cardiac death, myocardial infarction (MI), acute ischemic stroke (AIS), or revascularization.

The final pre-processed dataset had 1618 candidate predictive variables including 390 clinical variables and 1228 metabolic variables. There were 1901 instances with 101 positive cases of MACE. The remaining 1800 individuals were labeled negative for MACE.

5.2.2 Experiment design

To select the best predictive classifier, for MACE outcome, we evaluated each classifier over 5 runs of 10-fold cross-validation. We do this because as a result of the output variable being highly skewed, models learned for each fold had high variability. We significantly reduced the variability by running the evaluation over 5 runs of 10-folds. Finally, the model that

had the best predictive performance, on average, across 5 runs of 10-folds were judged as the best predictive model.

We evaluated the classifiers using two predictive metrics— AUROC (area under ROC curve) and AUPRG (area under precision-recall gain curve) and one calibration metric— the Brier score.

AUROC is the probability that any positive instance from the data is scored higher than a negative instance. This reflects the ability of the classifier to discriminate the individuals who are likely to develop a MACE outcome (i.e., $MACE = Yes$) from those who are unlikely to develop a MACE outcome (i.e., $MACE = No$). AUROC can be seen as a generalization of accuracy, where we do not *a priori* have to decide the cut-off for the classifier score, above which the classifier would predict the instance as class positive.

AUPRG is a reliable metric like AUROC and is particularly useful when true negatives do not contribute to model predictive performance. This metric is especially helpful in heavily skewed datasets, where the number of negative examples far outnumber the positive examples. This is true for the Heart SCORE dataset and so AUPRG presents a reliable alternative to AUROC. AUPRG can be considered as a generalization of the F-score where we do not have to decide *a priori*, which of precision or recall is more important for our classifier.

Brier score is the mean squared error between the classifier assigned score (or probability) and the actual outcome (value 1 set for positive class and 0 for negative class).

For a more detailed explanation of these metrics, please refer section 4.2.2.

5.2.3 Predictive classifiers compared

To predict the risk of cardiovascular disease, we evaluated the predictive performance of 2 cardiovascular risk scores, 9 state-of-the-art methods in machine learning, and BRL and EBRL classifiers described in chapter 3.

5.2.3.1 Cardiovascular risk scores as baseline classifiers The cardiovascular risk scores are not meant to be used as classifiers. However, we modify them as classifiers to set

a clinical baseline model to compare the machine learning methods. Specifically we will use Framingham Risk Score (FRS) [Dagostino et al., 2008] and Pooled Cohort Risk Equations (PCRE) [Goff et al., 2014]. We do not use Reynolds Risk Score (RRS) here as one of the variables (hemoglobin A1c) was not measured from Heart SCORE participants at the time of enrollment. So, we cannot reliably compute RRS.

The clinical variables needed to compute FRS are— age, sex, LDL cholesterol, HDL cholesterol, systolic blood pressure, blood pressure medication, smoking, and diabetes. PCRE additionally requires race to compute the risk score. Both FRS and PCRE are examples of Cox proportional hazard models [Cox, 1972] in statistics.

Proportional hazards models are a class of survival models in statistics, which analyze the expected time until one or more events occur (e.g. disease or death) and relate this time to covariates that it may be associated with. Survival models have two parts— 1) the baseline hazard function ($\lambda_0(t)$), and 2) effect parameters. The baseline hazard function maps how the risk changes based on the baseline measurements of the covariates. Proportional hazards models assume that the covariates are multiplicatively related to the hazard. The general form of the Cox model is given by the following equation—

$$\lambda(t|X_i) = 1 - \lambda_0(t)^{exp\left(\sum_{i=1}^{|F|}\beta_i X_i - \sum_{i=1}^{|F|}\beta_i \bar{X}_i\right)}$$

Where, X_i are the realized values of the co-variates (e.g. age of the individual in years, LDL cholesterol in mg/dL, etc.). β_i is the estimated regression co-efficients. $\lambda_0(t)$ is the baseline survival at follow-up time t, here $t = 10$ years. $|F|$ is the total number of covariates or risk factors assessed by the model. Finally, $\lambda(t|X_i)$ is the CVD risk at time t.

The covariates were estimated from study participants similar to the Heart SCORE study. For example, FRS covariates were estimated from the Framingham Heart Study that started in 1948 with 5,209 adult subjects in Framingham, Massachusetts. The study now has had three generations of participants.

To treat FRS and PCRE as a classifier, we take the output probability from these Cox models. The probabilities are then converted into percentages. We choose a score $\leq 10\%$ is considered low-risk, $10\% < score \leq 20.0\%$ is intermediate-risk, and score $> 20.0\%$ is high-risk. These cut-offs are arbitrary. To evaluate risk factors as classifiers, we classify

any individual with risk score ≥ 10.0 as a positive case, else the individual is predicted as a negative case. By doing so, we treat these clinical CVD risk scores as classifiers.

Note that it does not matter what cut-offs we choose because none of our evaluation metrics (AUROC, AUPRG, and Brier score) depend upon this cut-off. As a result, it does not matter what cut-offs we choose here.

5.2.3.2 Machine learning classifiers We also compared these models with 9 state-of-the-art classifiers that includes three interpretable classifiers namely— C4.5 [Quinlan, 2014], RIPPER [Cohen, 1995], and PART [Frank and Witten, 1998]; and 6 complex and non-interpretable classifiers namely— multivariate logistic regression [le Cessie and van Houwelingen, 1992], support vector machines [Platt, 1999], naïve Bayes [John and Langley, 1995], Bagged-C4.5 [Breiman, 1996], Boosted-C4.5 [Freund et al., 1996], and Random Forests [Breiman, 2001].

C4.5, RIPPER, and PART are interpretable classifiers i.e., the statistical model is human readable. These models offer intelligible explanations for their predictions. C4.5 is a popular decision tree learning algorithm, where each path of the decision tree can be interpreted as rules. RIPPER (Repeated Incremental Pruning to Produce Error Reduction) is a propositional rule learning algorithm that uses a divide-and-conquer strategy during model training. PART is a rule learning method that combines the approaches of both C4.5 and RIPPER by building partial decision trees, inferring rules from the trees, and using a divide-and-conquer strategy to build the rule model.

Multivariate logistic, support vector machines, naïve Bayes, Bagged-C4.5, Boosted-C4.5, and Random Forest are non-interpretable classifiers. They do not offer human-readable explanations for their predictions. Bagged-C4.5, Boosted-C4.5, and Random Forest are ensemble learning methods that learn a number of decision trees during training, and combine predictions from them during inference. For the experiments here, we learn 100 trees per ensemble. The naïve Bayes classifier is a simple probabilistic classifier that learns a network with strong independence assumption between the variables and uses the Bayes theorem for inference from the learned network. Support Vector Machines is an algorithm that learns a hyperplane function to differentiate the classes in the problem space.

We ran these classifiers from the Weka [Frank et al., 2016] workbench (version 3.8.1) using the default parameters for each classifier.

We also evaluate the BRL and EBRL models as we described in section 3.1 and 3.2, respectively. BRL methods are simple rule-based classifiers, so they are interpretable. EBRL are ensemble methods that are less interpretable but offer some human understandable explanations via BREVity (see section 3.2.4).

5.2.4 Metabolite set enrichment analysis (MSEA)

We wanted to perform functional analysis of the set of metabolites selected by the best performing classifier. One way to do this is using enrichment analysis [Subramanian et al., 2005]. Enrichment analysis is a statistical method to help identify if a class of metabolites is over-represented in a selected set of metabolites. The metabolites can be classed by the pathways they are involved in. This is called pathway enrichment analysis. The metabolites can also be classed by known associated diseases with the metabolites.

While there are reliable implementations of *gene set enrichment analysis*, where enrichment analysis is performed for a given set of genes. We found no available tool that could perform the same for metabolites. So, we implemented this using the human metabolite ontology from Human Metabolite Database (HMDB) [Wishart et al., 2017].

The analysis of datasets from high-throughput studies, such as the one we perform on the Heart SCORE metabolic dataset, involves the selection of a set of metabolites that are found to be differentially expressed for a disease outcome. These selected set of metabolites may include signatures of an underlying cellular process. The goal of the enrichment analysis is to retrieve a functional profile for these metabolites and help identify the cellular processes associated with the selected meatabolites. These cellular processes in turn can provide hints to the overall biological system involved with the disease. This can be done by comparing the selected set of metabolites to terms in biological ontologies that map known association between metabolites and these terms. For example— say, we want to compare to an ontology that shows known associations between metabolites and diseases. A disease term of interest for our analysis is 'myocardial infarction', which is a type of MACE. If we happen to

select metabolites from our data analysis that is also known to be associated to 'myocardial infarction', it helps us map our finding to the biological system that is well-understood. At the same time, it should be noted that, an absence of expected associations does not invalidate our findings and instead can potentially indicate novel biomarkers that may need to be validated to complete the profile of the disease.

In this analysis, we explored two types of ontologies— one containing disease terms and another containing pathways and super-pathways. A (metabolic) pathway is an interconnected sequence of chemical reactions. A super-pathway is a combination of biochemical pathways that collectively describe the metabolism of related compounds. An example of super-pathway is 'Glucose metabolism'. An example of a pathway is 'Glycolysis', which is a pathways under 'Glucose metabolism' that specifically concerns with the breakdown of glucose by enzymes, leading to the release of pyruvic acid and energy.

We developed a statistical test to compute the association between a set of metabolites to the terms. We compute this probability of association using a hypergeometric distribution, which describes the probability of k successes (metabolites associated to a term), in n draws (set of metabolites) without replacement, from a finite population of N (total metabolites) with K possible successes (metabolites known to be associated with the term). The result of the random draw may be a success (associated with the term) or a failure (not associated with the term). Say, X is a random variable that performs these random draws to select metabolites from an analysis i.e. this is the data mining algorithm that selects metabolites associated with CVD outcome. The p-value, i.e., the probability of X finding greater than or equal to k metabolites associated with the term (p-value) is given by Equation 5.1.

$$Pr(X \geq k) = \sum_{i=k}^{min(n,K)} \frac{\binom{K}{i} \cdot \binom{N-K}{n-i}}{\binom{N}{n}} \tag{5.1}$$

Where, N is the population size (total number of metabolites in the database), K is the number of success (total number of metabolites in the database that maps to a keyword of interest, like 'myocardial infarction'), n is the number of draws (total number of metabolites selected by the learned classifier), and k is the number of trial successes (of the metabolites selected by the learned classifier, how many map to a keyword of interest, like 'myocardial

infarction').

A small p-value indicates that the data mining methods selecting a metabolite associated with the term did not happen by chance. On the other hand, a large p-value indicates that the data mining method finding a metabolite associated with the term could have happen by chance. In statistical testing, we select a threshold for the p-value, below which, finding metabolites associated with a term is considered significant. This threshold is represented by α. For our analysis, we set $\alpha = 0.05$.

An important limitation of existing methods in literature to perform our metabolite set enrichment analysis is not being able to define the metabolites present in the population N. In the existing methods, N includes a set of all known metabolites from the database. However, Heart SCORE project metabolite dataset only includes 1228 metabolites. It should be noted that these 1228 were not chosen in any informed way (for e.g. metabolites that describe cellular pathways for functions related to the cardiovascular system) and instead, were chosen based on the metabolites available from the Metabolon's untargeted Precision Metabolomics platform and were therefore random. So, we re-constructed the ontology from the Human Metabolite Database [Wishart et al., 2017]. This enabled us to constrain N from a list of all known metabolites to only the 1228 metabolites studied in Heart SCORE. We were now able to compute the p-values both with and without constraining the population set N.

5.3 RESULTS

We now look at the results from our experiments. Subsection 5.3.1 shows the AUROC, AUPRG, and Brier scores achieved by each classifier that were evaluated. Subsection 5.3.2 lists the ranked list of the most important markers for classifying according to the best performing EBRL classifier. Subsection 5.3.3 displays the best performing BRL and EBRL classifier. Finally, 5.3.4 shows the results from the metabolite set enrichment analysis.

5.3.1 Classifier predictive performance comparison

In these experiments, the best performing BRL classifier was using the decision tree representation i.e., BRL.DT. The best performing EBRL classifier was Bagging with BRL.G i.e., Bagged-BRL.G-LC.

The AUROCs achieved by each classifier is compared in Figure 27.

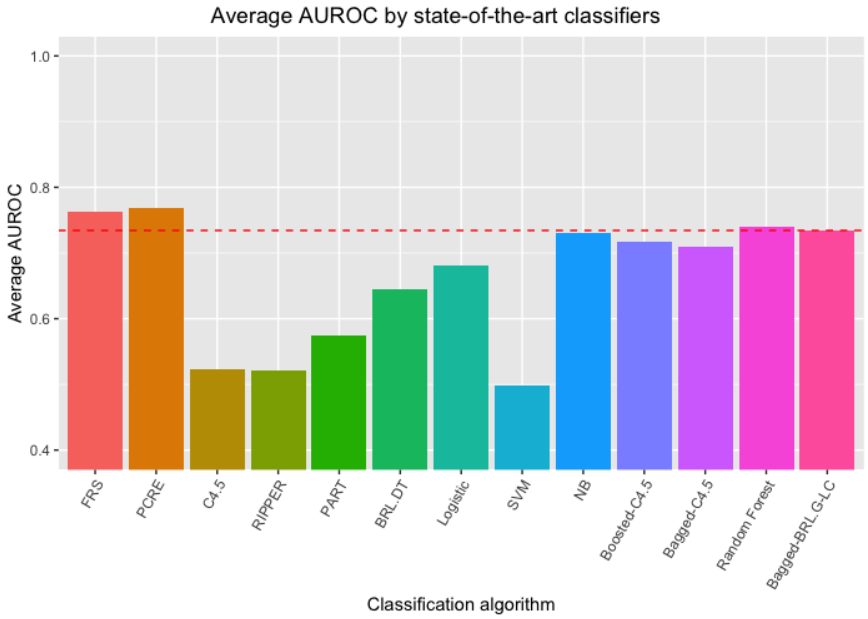


Figure 27: Average AUROC (higher is better) achieved by classifiers over 5 runs of 10-fold cross-validation.

The AUROCs of clinical standard baselines FRS (0.7637) and PCRE (0.7681) were much better than the machine learning models learned from the Heart SCORE dataset. The AUROC of BRL.DT (0.6442) was much better than other interpretable classifiers— C4.5 (0.5228), RIPPER (0.5206), and PART (0.5736). The ensemble classifiers Bagged-BRL.G-LC (0.7343), Random Forest (0.7403) and naïve Bayes (0.7303) perform similarly. Bagged-C4.5 (0.7089) and Boosted-C4.5 (0.7167) were close behind. Logistic (0.6819) and SVM (0.4990) models perform poorly.

The AUPRGs achieved by each classifier is shown in Figure 28. Bagged-BRL.G-LC (0.7903) achieves a much better AUPRG than other methods. Only BRL.DT comes close (0.7495). Clinical standard baselines FRS (0.6450) and PCRE (0.6581) perform worse. Other

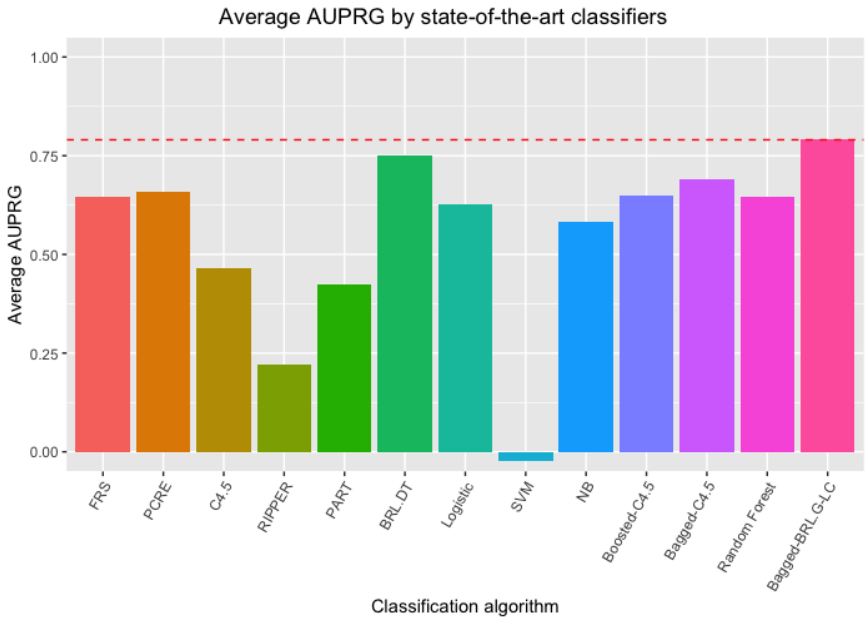


Figure 28: Average AUPRG (higher is better) achieved by classifiers over 5 runs of 10-fold cross-validation.

ensembles also don't do so well— Random Forest (0.6471), naïve Bayes (0.5828), Bagged-C4.5 (0.6916), Boosted-C4.5 (0.6487), Logistic (0.6256), and SVM (−0.0226). Note that AUPRGs can take negative values [Flach and Kull, 2015].

The Brier scores are shown in Figure 29. The BRL-based classifiers performed the best on calibration— BRL.DT(0.0481) and Bagged-BRL.G-LC (0.0472). The naïve Bayes (0.1488) does poorly as is notoriously known. Others perform more or less similarly— FRS (0.0622), PCRE (0.0532), C4.5 (0.0555), RIPPER (0.0568), PART (0.0790), Logistic (0.0584), SVM (0.0584), Boosted-C4.5 (0.0590), Bagged-C4.5 (0.0502), and Random Forest (0.0486).

5.3.2 Variable importance

Bagged-BRL.G-LC classifier performs competitively on AUROC compared to other machine learning methods. It does exceedingly well on AUPRG. It also does well on calibration using Brier score. Table 40 shows the ranked list of the most important variables used for prediction by Bagged-BRL.G-LC. The detailed explanation of how we compute this is shown

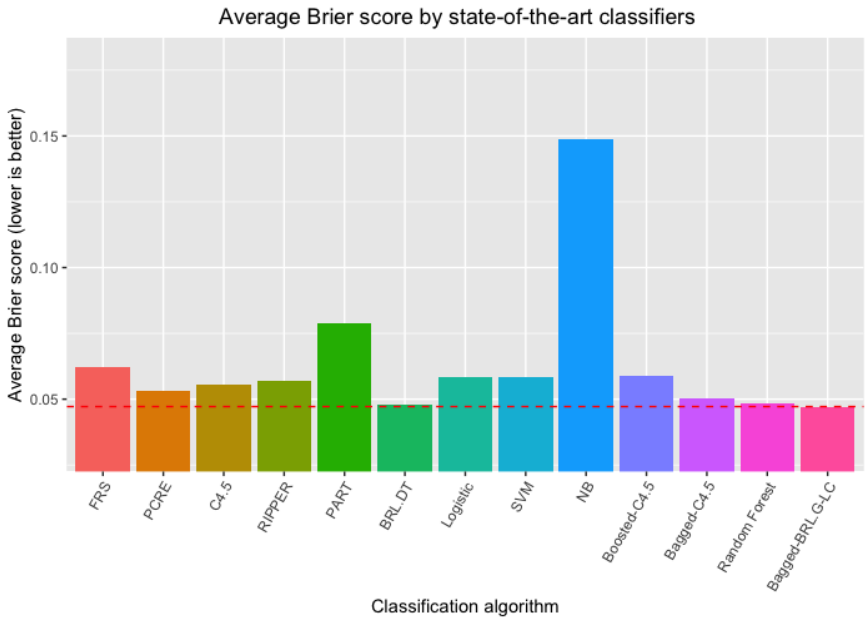


Figure 29: Average Brier score (lower is better) achieved by classifiers over 5 runs of 10-fold cross-validation.

in section 3.2.3. An intuitive way to think of the variable importance score is the fraction of 100 BRL models (that were combined into the ensemble) that contains the particular variable.

The variable for history of percutaneous coronary intervention is overwhelmingly the most important predictive variable. This makes intuitive sense since those former patients must already have conditions for poor heart health. A HDL cholesterol measurement from the Vertical Auto Profile (VAP) lipid test also appears to be important. Metabolites—15506 (choline), 35137 (N2,N2-dimethylguanosine), (unknown biochemical), and 48351 (N1-methylinosine) all achieve variable importance of > 0.1.

5.3.3 Visualizing Bagged-BRL-L with BREVity

Figure 30 shows the rule model from our experiments. We use the same 0.1 cut-off for BRL classifiers as we did for FRS and PCRE. For example, rule 2 in BRL.DT states that if the individual has not undergone percutaneous coronary intervention but their HDL2A is less

Rank	Variable	Score	Rank	Variable	Score	Rank	Variable	Score
1	*History of PCI*	0.75	21	*Sex*	0.04	41	*Oral hypoglycemic(48 hrs)*	0.01
2	*HDL subfraction 2A*	0.32	22	*sphingomyelin(d17:1/14:0,d16:1/15:0)**	0.04	42	*X - 24686*	0.01
3	*choline*	0.2	23	*N-acetylisoleucine*	0.04	43	*X - 24422*	0.01
4	*N2,N2-dimethylguanosine*	0.18	24	*adenine*	0.03	44	*C-glycosyltryptophan*	0.01
5	*X - 15461*	0.14	25	*sphingomyelin (d18:2/14:0, d18:1/14:1)**	0.03	45	*X - 12026*	0.01
6	*N1-methylinosine*	0.11	26	*2-aminoheptanoate*	0.03	46	*X - 12117*	0.01
7	*Age*	0.08	27	*metformin*	0.03	47	*N-acetylmethionine*	0.01
8	*Finger-stick HDL*	0.07	28	*orotidine*	0.03	48	*N-acetylleucine*	0.01
9	*N1-methylinosine*	0.07	29	*1-ribosyl-imidazoleacetate**	0.02	49	*β-hydroxyisovalerate*	0.01
10	*5α-androstan-3β,17β-diol disulfate*	0.07	30	*octadecanedioylcarnitine(C18-DC)**	0.02			
11	*hypotaurine*	0.06	31	*4-hydroxyphenylacetylglutamine*	0.02			
12	*N-acetylglutamine*	0.06	32	*X - 24334*	0.02			
13	*salicyluric glucuronide**	0.06	33	*X - 15497*	0.02			
14	*Social network Q5*	0.05	34	*isovalerylcarnitine (C5)*	0.02			
15	*X - 24686*	0.05	35	*History of abnormal cath*	0.01			
16	*X - 11564*	0.05	36	*Age category*	0.01			
17	*pyroglutamine**	0.05	37	*Time to peak LDL*	0.01			
18	*formiminoglutamate*	0.05	38	*VAP HDL*	0.01			
19	*N-acetylphenylalanine*	0.05	39	*HDL subfraction 3A*	0.01			
20	*N-acetylneuraminate*	0.05	40	*HDL subfraction 3*	0.01			

Table 40: Ranked list of the most important variable for prediction using Bagged-BRL.G-LC classifier.

1. **IF** ((*History of PCI* = No)(*HDL subfraction 2A* < 10.25)(*choline* < 1.32)) **THEN** (*MACE* = No)
   Confidence = 0.9537, TP = 949, FP = 46

2. **IF** ((*History of PCI* = No)(*HDL subfraction 2A* < 10.25)(*choline* > 1.32)) **THEN** (*MACE* = Yes)
   Confidence = 0.1858, TP = 25, FP = 110

3. **IF** ((*History of PCI* = No)(*HDL subfraction 2A* > 10.25) **THEN** (*MACE* = No)
   Confidence = 0.9873, TP = 712, FP = 9

4. **IF** (*History of PCI* = Yes) **THEN** (*MACE* = Yes)
   Confidence = 0.4204, TP = 21, FP = 29

Variables used (3): *History of PCI*, *HDL subfraction 2A* , *choline*.

Figure 30: Set of rules learned by BRL.DT algorithm.

than 10.25 from the VAP lipid test and their choline levels are above 1.32, then there is still an increased risk of developing CVD disease. TP = 25 affirms that 25 individuals in the Heart SCORE dataset had agreed to both the left and right hand side of the rule. FP = 110, states that 110 individuals with the condition described by the left hand side, did not develop MACE outcome. Confidence is the score similar to FRS about the probability of the individual described with this rule will develop the condition described in the right-hand side of the rule.

Figure 31 displays a portion of the visualization generated by the BREVity tool. In

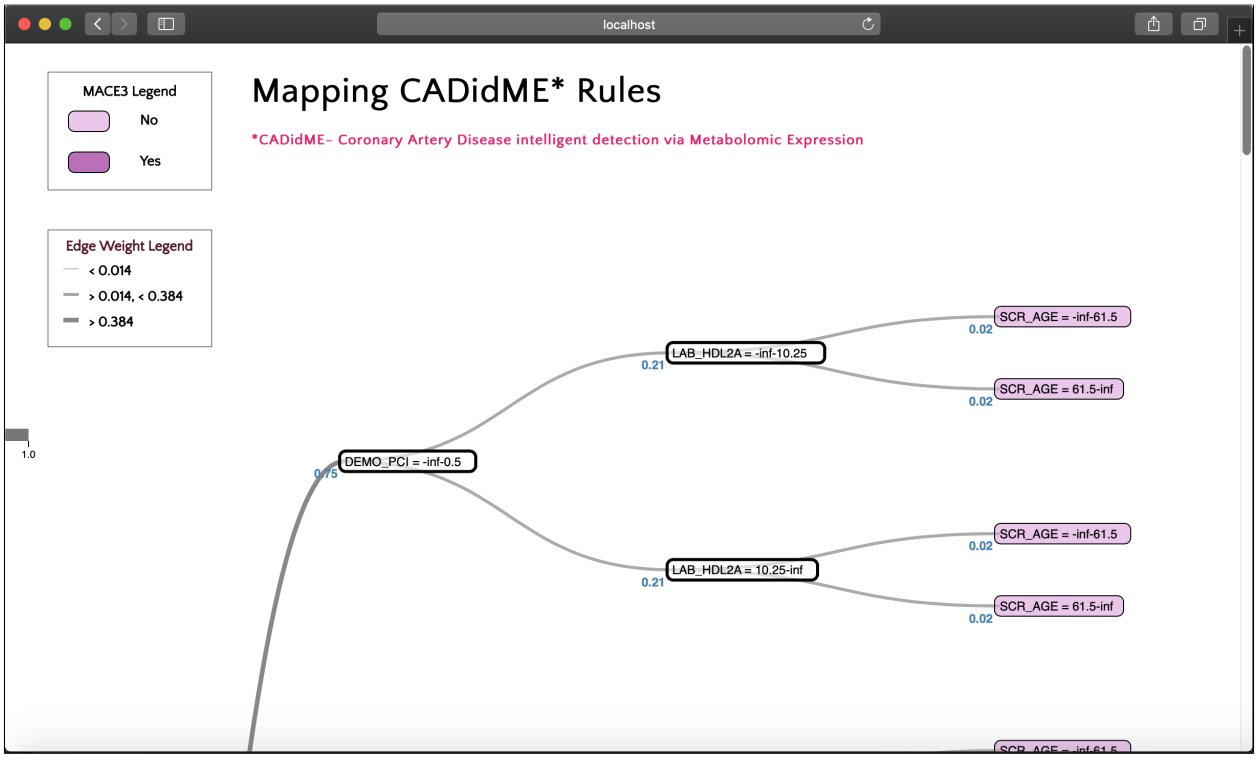


Figure 31: Classifier Bagged-BRL.G-LC visualized using BREVity.

this example, we expanded the node for patients who had undergone percutaneous coronary intervention. This node has an edge value of 0.75, which means that this pattern of having had history percutaneous coronary intervention appears to have a 75% say by the prediction made by Bagged-BRL.G-LC. If we go down the tree, we see the VAP test for HDL2A being less than 10.25. This edge has a weight of 0.21. This means that the combined pattern of having had undergone percutaneous coronary intervention and have a VAP test for HDL2A less than 10.25, together have a 21% say in the prediction made by Bagged-BRL.G-LC.

While the model is clearly more complex than the BRL model. However, the model is still interpretable since it offers still rule-like explanations for its prediction.

5.3.4 Metabolite set enrichment analysis results

We performed metabolite set enrichment analysis as we described in section 5.2.4. These were performed on the metabolites picked by Bagged-BRL.G-LC. We compared it against

the ontology of HMDB.

The top 10 most enriched biochemical pathways were as follows—

1. Histidine Metabolism (p-value = 0.0887)
2. Taurine and hypotaurine metabolism (p-value = 0.1128)
3. Betaine Metabolism (p-value = 0.2132)
4. Phospholipid Biosynthesis (p-value = 0.2441)
5. Amino/Sugar Metabolism (p-value = 0.2441)
6. Sarcosine Oncometabolite Pathway (p-value = 0.3024)
7. Azathioprine Action Pathway (p-value = 0.4521)
8. Mercaptopurine Action Pathway (p-value = 0.4521)
9. Purine Metabolism (p-value = 0.4295)
10. Thioguanine Action Pathway (p-value = 0.4521)

None of the associated biochemical pathway were found to be significant.

The top 10 most enriched known diseases withe metabolites chosen by Bagged-BRL.G-LC were as follows—

1. Aspartylglucosaminuria (p-value = 0.0391)
2. Cholangiocarcinoma (p-value = 0.0391)
3. Mastocytosis (p-value = 0.0391)
4. Cervical cancer (p-value = 0.0391)
5. Ovarian cancer (p-value = 0.0391)
6. Colorectal cancer (p-value = 0.0391)
7. Stomach cancer (p-value = 0.0767)
8. 3-methyl-crotonyl-glycinuria (p-value = 0.0767)
9. Spina bifida (p-value = 0.0767)
10. 3-methylglutaconic aciduria (p-value = 0.0767)

Again none of the associated diseases stand out. One possible explanation for the underwhelming results from enrichment analysis is the lack of studies in metabolomics, in general.

5.4 BRAID CONCEPT FOR CVD RISK

This section describes the architecture of the concept framework of Bayesian Rules for Actionable Informed Decisions (BRAID) to deploy the BRL models, learned in this chapter, as an intelligent clinical decision support system for clinical practice.

The model building for this project was conducted on a cloud, i.e., a remote server. The data repository and case bases from the example here was from the baseline clinical and metabolic factor measurements from the Heart SCORE study patients. From the Bayesian Rule Learning System, only two algorithms were utilized— Bayesian Rule Learning and Ensemble Bayesian Rule Learning. There are future opportunities to perhaps incorporate prior domain knowledge in CVD into BRL_p. Clinical utility can be defined and we can run BRL-KD to find biomarkers that could translate better in clinical practice. These two can be done in consultation with a domain expert. As a result, I do not explore it further in this dissertation.

The learned BRL and EBRL rules from the best performing classifiers are then stored in a Knowledge base. Here, we convert it into a JSON-formatted object file. This is sent into BREVity tool for the expert to visualize, as we saw in the previous section. The next step is validation of the rules, which also involves the domain expert. The physician may test the BRL rules either on prospective studies or retrospective studies from historical studies that measured these variables. If the rules are validated, they can be authored into Ensemble of Rules Integrated Expert (ERIE). This section, shown in Figure 32 as containing two functions (represented as boxes), allows the domain expert to accept, edit, and validate the rules.

Finally, a physician queries a patient record to BRAID. The BRAID generates probabilities of either a positive MACE outcome or a negative one. These probabilities for both BRL and EBRL is generated from posterior probabilities for the rules that fired. These probabilities are the computed risk. If BRL is used, BRAID supports the computed risk with the rule that fired, as the explanation for its prediction. If EBRL is used, rules fired from each of the models in the ensemble are combined into a BREVity tree to offer human-readable explanation for the computed risk.

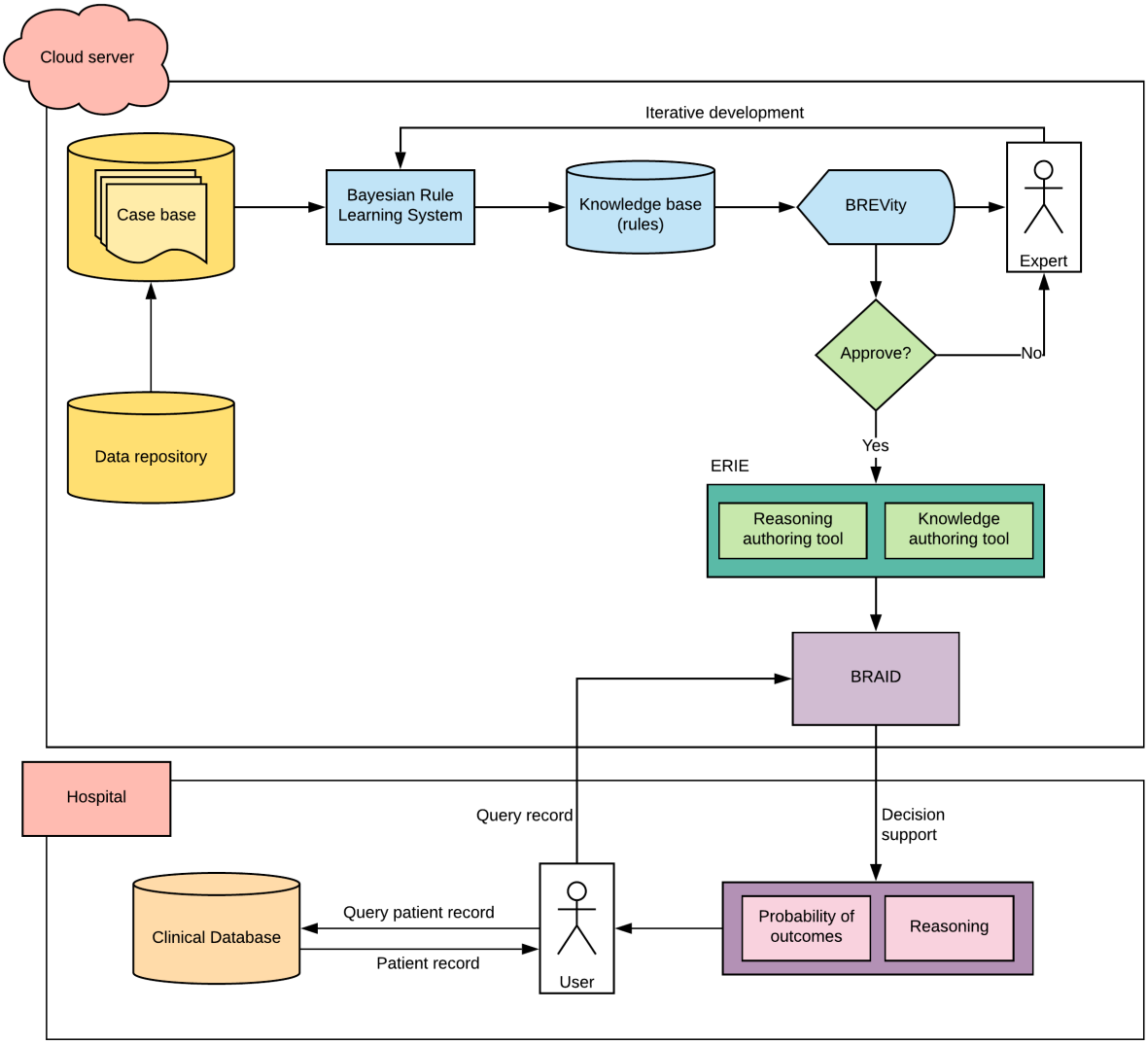


Figure 32: Bayesian Rules for Actionable Informed Decisions (BRAID): An overview of the conceptual cloud-based intelligent clinical decision support system for computing cardiovascular disease risk.

This concept framework was developed inspired by the flexible designs of the Leeds Abdominal Pain System and MYCIN clinical decision support systems. BRAID offers a visionary, intelligent, data-driven approach to generating these clinical decision support systems.

5.5 CONCLUSION

For modeling the Heart SCORE dataset, the BRL model achieved better predictive performance, in terms of AUROC and AUPRG, and better calibration performance, in terms of Brier score, when compared to other state-of-the-art interpretable classifiers. EBRL performed similar or better than state-of-the-art methods in machine learning in terms of AUROC, but was better than them on AUPRG and Brier score. This experiment successfully demonstrates the efficiency of the models developed in this dissertation in biomarker discovery tasks on high-dimensional datasets that contain samples from multifactorial datasets.

With respect to the clinical baselines of FRS and PCRE, both these models outperform EBRL on AUROC but EBRL outperforms FRS and PCRE on AUPRG and Brier score. This result indicates EBRL's better performance on precision and recall when compared to FRS and PCRE. This result is not surprising because FRS and PCRE are screening methods that emphasize on recall more than precision.

Also, note that the coefficients learned to compute FRS were learned over three generations of study participants in Framingham, Massachusetts. The study dates back to 1948 with 5,209 adult participants [Mahmood et al., 2014]. Compared to that our machine learning algorithms only have had access to 2000 participants enrolled in 2003. So, data-driven machine learning classifiers compared in this experiment are at a disadvantage over the number of instances available to learn a statistical model.

6.0 CONCLUSION

In this dissertation, I formulated the problem of discovering both statistically significant and clinically relevant biomarkers, from complex omic datasets, as a knowledge discovery problem. I use the definition for knowledge discovery in databases (KDD) from [Fayyad et al., 1996b] i.e., "a non-trivial process of the extraction of **valid**, **novel**, potentially **useful**, and ultimately **understandable** patterns in data". To help solve the KDD problem, I developed a set of BRL algorithms, namely— BRL (see Section 3.1), EBRL (see Section 3.2), BRL_p (see Section 3.3), and BRL-KD (see Section 3.4). These algorithms are included in the Bayesian Rules for Actionable Informed Decisions (BRAID) system. Each BRL algorithm in the BRAID system caters to a specific aspect of the KDD definition and, collectively, these algorithms help solve the KDD problem. The motivation behind the design of each algorithm and the observations made during its evaluation are summarized in the following paragraphs.

BRL is a rule learning algorithm that learns a set of rule patterns. Rule patterns are considered **understandable** because they are in form of explicit propositional logic statements that are intelligible to humans. The version of BRL, at the inception this dissertation study, generated a large number of rules. Rule models with a large number of rules can be challenging to read and comprehend. To improve the understandability of BRL rule models, I studied three representations to better capture regularities in the data. These regularities are called context-specific independences. The three representations were studied— global complete tree (**BRL.G**), decision tree (**BRL.DT**), and decision graph (**BRL.DG**). BRL.G was found to have significantly better predictive performance and required significantly fewer variables than other popular rule learning algorithms like C4.5, PART, and RIPPER (see Section 4.3.1). However, BRL.G required significantly more rules than other rule learning

methods. BRL.DT was found to be an effective alternative to BRL.G by achieving similar predictive performance as BRL.G and required much fewer rules, being able to capture some of these context-specific independencies (see Section 4.3.2). BRL.DG required the fewest number of rules by being able to capture all possible context-specific independences. However, to achieve comparable predictive and calibration performance as BRL.G and BRL.DT, BRL.DG required a more exhaustive search method (see Section 4.3.3). While BRL performed better than other traditional, rule-based, supervised learning methods, they were outperformed by supervised methods that picked a large number of variables for the prediction task (see Section 4.3.4).

Two methods developed in this dissertation helped obtain more **valid** rule models, namely— **EBRL** (Ensemble Bayesian Rule Learning) and $\mathbf{BRL_p}$ (BRL with informative priors).

EBRL is a set of ensemble BRL methods that overcomes the data fragmentation issue of BRL to enable it to model multifactorial diseases, wherein many hundreds of biomarkers may collectively contribute to the disease physiology, each with a small effect. Two ensemble generation strategies were studied— Bagging (Bagged-BRL) and Boosting (Boosted-BRL). Three model combination strategies were studied— Linear Combination (BRL-LC), Bayesian Model Averaging (BRL-BMA), and Bayesian Model Combination (BRL-BMC). In general, model generation using Bagging (see Sections 4.4.1, 4.4.2) was found to perform better than Boosting (see Sections 4.4.3, 4.4.4) on gene-expression datasets. Bagged-BRL-LC (or classic Bagging of BRL classifiers) was found to achieve significantly higher predictive performance than BRL base classifier, C4.5, Bagged-C4.5, and Boosted-C4.5. Bagged-BRL-BMA also performed better than BRL alone (see Sections 4.4.1, 4.4.2). The results indicate the ensemble methods help improve model predictive performance and that BRL is a better choice for a base classifier than C4.5 decision trees for modeling the studied gene-expression datasets. However, both Bagged-BRL-LC and Bagged-BRL-BMC outperformed Bagged-BRL-BMA (see Section 4.4.2). Between Bagged-BRL-LC and Bagged-BRL-BMC, Bagged-BRL-BMC was the more reliable model by being able to account for the uncertainty in the correctness of model combination (see Section 4.4.5). While EBRL models achieve high predictive performance, they are much harder to interpret (i.e., less understandable). To help improve

the **understandablity** of the EBRL models, two methods were designed— 1) to compute relative variable importance in the EBRL model (see Section 3.2.3), and 2) a novel ensemble BRL model visualization tool called Bayesian Rule Ensemble Visualizing tool (**BREVity**) to infer variable relationships in the EBRL model (see Sections 3.2.4, 4.4.7). Finally, *EBRL methods were shown to outperform state-of-the-art, traditional, supervised learning methods, on average, in terms of its predictive performance* (see Section 4.4.6). In other words, patterns learned by EBRL more statistically significant than the traditional supervised learning methods.

BRL$_\mathbf{p}$ allows user to incorporate prior domain knowledge into the BRL model learning process. I demonstrated its use in the analysis of a real-world lung cancer prognostic dataset, where incorporation of an informative prior (EGFR gene associated with lung cancer outcome) led to an improvement in model predictive performance (see Section 4.5.2.2). This showed that using BRL$_\text{p}$, we can learn models with more validity when reliable prior knowledge about the problem is available.

Finally, **BRL-KD** (BRL for knowledge discovery) is a novel method designed to help learn **novel** and **useful** patterns that are clinically more relevant. BRL-KD can incorporate a clinical utility function, like cost-effectiveness to help learn models that not only optimize predictive performance but also the utility function. I demonstrated its application in the analysis of a real-world cardiovascular disease diagnostic dataset. BRL-KD allowed the user to obtain a set of BRL models, each with a different trade-off between cost and predictive performance (see Section 4.6.2). The user now has the option to choose an acceptable trade-off while being able to observe models that are clinically more useful. To my knowledge, *there exists no supervised learning method other than BRL-KD that enables the user to find clinically more relevant classifiers.*

In summary, I re-visit the thesis statement (see Section 1.2.1) and conclude that— *the BRAID system of algorithms, including BRL, EBRL, BRL$_p$, and BRL-KD together help learn classifiers for biomarker discovery that are, on average, statistically more significant compared to traditional supervised learning methods, while also being able to find clinically more relevant classifiers.*

6.1 FUTURE WORK

BRL.DG is capable of representing all context-specific independencies. Expanding the search algorithm's scope (by extending the greedy best-first search to beam search), allowed BRL.DG to obtain comparable predictive performance to BRL.G and BRL.DT. BRL.DG also had the best calibration performance. These results motivate more studies on BRL.DG by further expanding the search space using non-incremental search methods such as particle swarm optimization (e.g. Artificial bee Colony optimization) [Ji et al., 2017].

BRL depends upon discretization methods for pre-processing continuous variables before being able to learn from them. This limits BRL to the capabilities of the discretization method. Most biomarker measurements are continuous-valued. Future developments in BRL should enable it to learn from both discrete- and continuous-valued data. There are many scoring methods available to learn Bayesian networks from such mixed data types [Bøttcher, 2004]. It would be worth exploring if using scoring functions for mixed data types outperform the scoring function that only takes in discrete valued input after having gone through discretization.

In this dissertation, I restricted the Bayesian network to only the outcome variable and its parents. In a Bayesian network, the value of the outcome variable does not only depend upon the parents, but the whole Markov Blanket of the outcome variable. A *Markov Blanket* (MB) of any target node in a Bayesian network (BN), is defined by Pearl [Pearl, 2014], as the set of nodes including the target node's parents, children, and spouses (i.e., other parents of their common children). An interesting property of the MB is that, collectively, given the values of the MB of the target node in a BN, all other nodes are independent of the target node. In other words, the conditional probability of the target node, given values of all other nodes in the BN is equal to the conditional probability of the target node given the MB alone. Furthermore, the *minimal MB* or the *Markov boundary* is defined as the smallest set of variables in the BN that has the MB property. For the purpose of this section, when we refer to MB, we mean the minimal MB. MBs have also been proven useful in practice as a feature selector [Koller and Sahami, 1996, Tsamardinos et al., 2003], for learning BNs [Margaritis and Thrun, 2000], and for discovering causal relationships [Mani

and Cooper, 2004]. One possible way to improve the MB search would be to try improving the calibration of the probability of each edge in the learned BN [Jabbari et al., 2017].

For the development of EBRL, we mainly explored bagging and boosting procedures. However, more recently, ensemble methods using random forest and gradient boosting [Chen et al., 2015], have seen immense success. Experiments can be conducted to test if extending these methods to EBRL can lead to further improvements in performance.

Many biomedical datasets contain variables with missing values. We encountered one such dataset from the case study in the previous chapter. Currently, there is no method available in BRL to handle missing data. In the case study, we performed a simple approach to handle missing data by median/mode imputation. We did this by making an assumption that each variable was missing completely at random (MCAR). We performed the median/-mode imputation for the missing values resulting in a single imputed dataset. To keep these assumptions, we had to drop any variable with more than 5% missing values. However, some of the dropped variables are identified as interesting and potentially crucial by our experts. In cases of variables with >5% missing values it is recommended to either perform multiple imputations using methods like EM algorithm, Multiple Imputation (MI), or Full-Information Maximum Likelihood (FIML) [Graham, 2009]. Each of these methods result in multiple imputed datasets. There is a need for algorithms that can combine evidence from such set of datasets and return a single learned classifier.

The work in this dissertation provides a framework using a machine learning algorithm to perform knowledge discovery in databases. This work will hopefully lead to many applications, specifically in the improvement of biomarker discovery projects that lead to findings that are better primed for clinical use. All the methods developed herein are made open-source under the MIT license and is available at (https://github.com/jeya-pitt/Bayesian-Rule-Learning).

7.0 APPENDIX A

7.1 NOTATIONS

X, Y, B_S, Θ	Random variables in the problem domain. X is specifically used to indicate an independent predictor variable. Y is used to indicate the dependent outcome variable. Variable B_S represents a Bayesian network structure. Θ represents a set of parameters (conditional probability tables for a Bayesian network).
x, y, θ	the state or the value assignment for the variables.
$\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{\Theta}$	A set of random variables.
$\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{\theta}$	A set of variable-value assignments or configurations.
$p(X = x)$ or $p(x)$	Probability that the variable X takes the value x
$p(A\|B)$	Conditional probability of states of A given the state of B.
X_i	i-th variable in a domain.
$X_{i=1:n}$	A set of n variables in the domain.
D	Dataset, a collection of examples and variable-values from the domain.
m	Total number os examples in the dataset.
X^j	j-th example/instance in a domain.
$\Gamma(x)$	Gamma function, equal to $(x-1)!$.
α	Hyperparameter in the BDeu score called prior equivalent sample size.
r_Y	Number of states taken by variable Y.
$\Pi(Y)$	Parents of Y in the Bayesian network structure.
q_Y	Configuration of variable-values of the parents of Y in the Bayesian network structure.

N_{jk}	Number of examples in D with j-th parent configuration while the outcome variable takes the k-th value.
$\hat{y}$	Predicted value of y according to a predictor.
$\mathbb{E}[X]$	Expectation of X. Equal to $\int_x x \cdot p(x) \cdot dx$.
$\arg\max_{x \in X} f(x)$	Returns the value of x that maximizes function $f(x)$.
$\mathbb{1}_{condition}$	Indicator function that returns 1 if the *condition* is true, or 0 otherwise.
Ψ	Clinical utility.
λ	Hyperparameter to control the influence of prior (in BRL_p) or clinical utility function (in BRL-KD).

8.0 APPENDIX B: EXTRACTING AND PRE-PROCESSING GENE EXPRESSION DATASETS

This appendix shows the R code used to extract and process the gene-expression data from GEO data repository, using GEO ID GSE10072 as an example. Similarly all 25 gene expression datasets were prepared for experiments conducted in this dissertation.

1. **Data extraction:** R code to extract gene expression data using *GEOquery*.

GEO data extraction

```
library(GEOquery)
library(Biobase)


geo_id = 'GSE10072'
data <- getGEO(geo_id, GSEMatrix=TRUE)


# Extract the gene expression data
data.exp <- as.data.frame(exprs(data[[1]]))
```

2. **Annoation:** R code to annotate the extracted gene expression data using *GEOmetadb*. This code links probe IDs from the microarray to the gene symbols they map to.

GEO data annotation

```
library(GEOmetadb)
sqlfile = getSQLiteFile()
```

```
library(DBI)
con = dbConnect(RSQLite::SQLite(), sqlfile)

# Function to get the name of the annotation database
# for the microarray technology
get_annotation_db <- function(gpl_id) {
    query = paste("select gpl,title,bioc_package
            from gpl
            where gpl='", gpl_id, "'", sep="")
    res = dbGetQuery(con,query)
    return(res$bioc_package)
}

# Get the name of the annotation database to use
annotation_db = get_annotation_db("GPL96")  # returns "hgu133a.db"

# Probe names in the microarray
probe_ids = rownames(data.exp)
# Match probe names with gene symbols they map.
gene_symbols = unlist(
                mget(probe_ids, hgu133aSYMBOL, ifnotfound=NA))

# Merge columns for probe IDs and gene symbols
# into the gene expression dataset
annotation = as.data.frame(cbind(probe_ids, gene_symbols))
data.exp$probe_ids <- rownames(data.exp)
data.annotated = merge(data.exp, annotated, by.x="probe_ids",
                        by.y="probe_ids")
```

3. **IQR filtering:** Continuing with the annotation, several probes can map to the same gene. For the problem of identifying differentially expressed genes, only one representation in each gene is desired. In this code snippet IQR filtering is done to identify which of the multiple probes mapping to the same gene is used to represent that gene.

IQR filtering

```
# Sorting by gene symbols
data.annotated.sorted = data.annotated[
                                order(data.annotated$gene_symbols),]

# IQR filtering
logdata = data.annotated.sorted[,!(colnames(data.annotated.sorted)
                                %in% c("probe_ids", "gene_symbols"))]
unlogdata = 2^logdata

# Calculating IQR for all probes using unlogged data
iqr <- apply(unlogdata,1,IQR)
data.iqr = cbind(data.annotated.sorted[,
            (colnames(data.annotated.sorted)
                %in% c("probe_ids", "gene_symbols"))],
            iqr,
            unlogdata)

# Keep probe with highest iqr in case of multiple probes
names(iqr) = data.annotated.sorted$probe_ids
iqrs = split.default(iqr, data.annotated.sorted$gene_symbols)
maxes = sapply(iqrs, function(x) names(which.max(x)))
data.singleprobe = data.iqr[data.iqr$probe_ids %in% maxes,
                    !(colnames(data.iqr) == "probe_ids")]
```

9.0 APPENDIX C: ADDITIONAL RESULTS FROM EXPERIMENT 1

In this appendix, we show some additional results from BRL runs from experiment 1 in section 4.3. These results were not necessary to test the hypotheses evaluated by the experiment and so were moved to this supplement merely as a reference for the curious reader.

9.1 EXPERIMENT 1A: BRL.G COMPARED TO STATE-OF-THE-ART INTERPRETABLE CLASSIFIERS

Classification performances achieved by the compared classifiers across the 25 datasets are shown in Tables— 41 (Accuracy), 42 (Precision), 43 (Recall), and 44 (F1 score). In these results, BRL.G used the arbitrary 0.5 threshold to predict the positive class. These metrics were largely similar for all the methods compared.

9.2 EXPERIMENT 1B: COMPARING BRL.G, BRL.DT, AND BRL.DG

Classification performances achieved by the compared classifiers across the 25 datasets are shown in Tables— 45 (Accuracy), 46 (Precision), 47 (Recall), and 48 (F1 score). In these results, all of BRL.G, BRL.DT, and BRL.DG use the arbitrary 0.5 threshold to predict the positive class. These metrics were largely similar for all the methods compared.

Data	C4.5	RIPPER	PART	BRL.G
GSE66360	70.71	73.74	70.71	77.78
GSE62646	92.86	92.86	92.86	92.86
GSE41861	74.64	74.64	74.64	71.01
GSE20881	74.42	81.40	72.67	76.74
GSE3365	81.10	83.46	81.10	90.55
GSE16879	97.26	94.52	97.26	97.26
GSE15245	67.69	66.15	69.23	73.85
GSE6613	58.10	59.05	55.24	47.62
GSE20295	62.37	53.76	64.52	55.91
GSE30999	94.71	96.47	94.12	93.53
GSE55447	69.23	65.38	65.38	71.15
GSE19429	85.50	91.00	86.50	91.50
GSE9006	79.22	77.92	81.82	75.32
GSE48350	100.00	100.00	100.00	100.00
GSE5281	80.75	83.23	81.37	80.75
GSE35978	57.05	64.26	63.93	60.33
GSE53987	64.39	62.44	59.02	66.34
GSE12288	52.25	59.01	51.80	59.01
GSE15852	76.74	79.07	79.07	81.40
GSE42568	97.52	95.87	97.52	91.74
GSE29431	96.97	95.45	96.97	96.97
GSE18520	98.41	98.41	98.41	98.41
GSE19804	88.33	90.00	88.33	88.33
GSE10072	94.39	92.52	94.39	94.39
GSE68571	98.96	98.96	98.96	98.96
Average $\pm$ SEM	80.54 ± 3.01	81.18 ± 2.95	80.63 ± 3.03	81.27 ± 3.04

Table 41: Experiment 1a: Accuracy for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of accuracies are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	C4.5	RIPPER	PART	BRL.G
GSE66360	0.6800	0.7660	0.6800	0.7712
GSE62646	0.9500	0.9500	0.9500	0.9500
GSE41861	0.8115	0.8320	0.8115	0.7834
GSE20881	0.7750	0.8472	0.7626	0.8072
GSE3365	0.8728	0.8853	0.8678	0.9019
GSE16879	1.0000	0.9607	1.0000	1.0000
GSE15245	0.8317	0.7707	0.8450	0.8283
GSE6613	0.5627	0.5848	0.5483	0.4633
GSE20295	0.6267	0.4617	0.6133	0.4362
GSE30999	0.9239	0.9650	0.9139	0.9046
GSE55447	0.8417	0.8067	0.8217	0.8167
GSE19429	0.9334	0.9473	0.9340	0.9514
GSE9006	0.8719	0.8648	0.8719	0.7964
GSE48350	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8766	0.8696	0.8955	0.8262
GSE35978	0.6852	0.7469	0.7398	0.7130
GSE53987	0.7502	0.7356	0.7247	0.7268
GSE12288	0.5189	0.5815	0.4975	0.6031
GSE15852	0.7931	0.8267	0.8300	0.8131
GSE42568	0.9826	0.9644	0.9826	0.9568
GSE29431	0.9857	0.9857	0.9857	0.9857
GSE18520	1.0000	1.0000	1.0000	1.0000
GSE19804	0.8905	0.9264	0.8905	0.8702
GSE10072	0.9514	0.9490	0.9514	0.9514
GSE68571	1.0000	1.0000	1.0000	1.0000
Average ± SEM	0.8446 ± 0.0283	0.8491 ± 0.0287	0.8447 ± 0.0289	0.8343 ± 0.0312

Table 42: Experiment 1a: Precision values for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of precision are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	C4.5	RIPPER	PART	BRL.G
GSE66360	0.7800	0.7550	0.7800	0.7950
GSE62646	0.9667	0.9667	0.9667	0.9667
GSE41861	0.8033	0.7811	0.8033	0.7911
GSE20881	0.7967	0.8378	0.7667	0.8278
GSE3365	0.8306	0.8722	0.8431	0.9625
GSE16879	0.9690	0.9833	0.9690	0.9690
GSE15245	0.7467	0.8000	0.7467	0.8400
GSE6613	0.6600	0.6400	0.5000	0.4600
GSE20295	0.6250	0.4500	0.6750	0.4750
GSE30999	0.9750	0.9639	0.9750	0.9750
GSE55447	0.7250	0.7650	0.7050	0.8350
GSE19429	0.9064	0.9561	0.9175	0.9558
GSE9006	0.8500	0.8300	0.8900	0.9067
GSE48350	1.0000	1.0000	1.0000	1.0000
GSE5281	0.7778	0.8167	0.7667	0.8514
GSE35978	0.6690	0.7067	0.7224	0.6719
GSE53987	0.7667	0.7600	0.6933	0.8400
GSE12288	0.6455	0.6545	0.5636	0.5182
GSE15852	0.7350	0.7700	0.7400	0.8300
GSE42568	0.9909	0.9909	0.9909	0.9518
GSE29431	0.9833	0.9633	0.9833	0.9833
GSE18520	0.9800	0.9800	0.9800	0.9800
GSE19804	0.9000	0.8833	0.9000	0.9333
GSE10072	0.9500	0.9167	0.9500	0.9500
GSE68571	0.9875	0.9875	0.9875	0.9875
Average $\pm$ SEM	0.8408 ± 0.0250	0.8412 ± 0.0275	0.8326 ± 0.0286	0.8503 ± 0.0320

Table 43: Experiment 1a: Recall values for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of recall are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	C4.5	RIPPER	PART	BRL.G
GSE66360	0.7238	0.7400	0.7238	0.7800
GSE62646	0.9474	0.9474	0.9474	0.9474
GSE41861	0.8066	0.8023	0.8066	0.7826
GSE20881	0.7822	0.8384	0.7638	0.8039
GSE3365	0.8554	0.8757	0.8571	0.9318
GSE16879	0.9833	0.9677	0.9833	0.9833
GSE15245	0.7835	0.7885	0.7917	0.8350
GSE6613	0.6000	0.5981	0.5155	0.4554
GSE20295	0.6024	0.4675	0.6353	0.4810
GSE30999	0.9486	0.9647	0.9432	0.9379
GSE55447	0.7949	0.7805	0.7692	0.8235
GSE19429	0.9197	0.9511	0.9256	0.9537
GSE9006	0.8491	0.8381	0.8704	0.8348
GSE48350	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8144	0.8402	0.8171	0.8268
GSE35978	0.6765	0.7268	0.7291	0.6952
GSE53987	0.7591	0.7475	0.7123	0.7850
GSE12288	0.5726	0.6128	0.5368	0.5561
GSE15852	0.7619	0.7857	0.7805	0.8182
GSE42568	0.9856	0.9763	0.9856	0.9519
GSE29431	0.9815	0.9720	0.9815	0.9815
GSE18520	0.9905	0.9905	0.9905	0.9905
GSE19804	0.8852	0.8983	0.8852	0.8889
GSE10072	0.9483	0.9298	0.9483	0.9483
GSE68571	0.9942	0.9942	0.9942	0.9942
Average $\pm$ SEM	0.8387 ± 0.0267	0.8414 ± 0.0279	0.8358 ± 0.0282	0.8395 ± 0.0309

Table 44: Experiment 1a: F-measure values for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of F-measure are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	77.78	81.82	81.82
GSE62646	92.86	92.86	92.86
GSE41861	71.01	73.19	73.19
GSE20881	76.74	79.07	77.33
GSE3365	90.55	81.89	84.25
GSE16879	97.26	97.26	97.26
GSE15245	73.85	80.00	80.00
GSE6613	47.62	47.62	49.52
GSE20295	55.91	65.59	65.59
GSE30999	93.53	92.94	92.94
GSE55447	71.15	69.23	69.23
GSE19429	91.50	92.50	92.50
GSE9006	75.32	75.32	76.62
GSE48350	100.00	100.00	100.00
GSE5281	80.75	82.61	82.61
GSE35978	60.33	59.34	59.67
GSE53987	66.34	61.95	60.49
GSE12288	59.01	59.01	56.76
GSE15852	81.40	80.23	81.40
GSE42568	91.74	93.39	93.39
GSE29431	96.97	96.97	96.97
GSE18520	98.41	98.41	98.41
GSE19804	88.33	90.00	90.00
GSE10072	94.39	91.59	91.59
GSE68571	98.96	98.96	98.96
Average $\pm$ SEM	81.27 ± 3.04	81.67 ± 2.93	81.73 ± 2.94

Table 45: Experiment 1b: Accuracy for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Classifier with higher values of accuracies are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	0.7712	0.8283	0.8283
GSE62646	0.9500	0.9500	0.9500
GSE41861	0.7834	0.8255	0.8273
GSE20881	0.8072	0.8261	0.8358
GSE3365	0.9019	0.8689	0.8788
GSE16879	1.0000	1.0000	1.0000
GSE15245	0.8283	0.8817	0.8817
GSE6613	0.4633	0.4887	0.5157
GSE20295	0.4362	0.5679	0.5679
GSE30999	0.9046	0.8935	0.8935
GSE55447	0.8167	0.7950	0.7950
GSE19429	0.9514	0.9625	0.9625
GSE9006	0.7964	0.8068	0.8235
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.8262	0.8518	0.8518
GSE35978	0.7130	0.7110	0.6909
GSE53987	0.7268	0.7416	0.7402
GSE12288	0.6031	0.5871	0.5700
GSE15852	0.8131	0.8648	0.8731
GSE42568	0.9568	0.9568	0.9568
GSE29431	0.9857	0.9857	0.9857
GSE18520	1.0000	1.0000	1.0000
GSE19804	0.8702	0.8905	0.8905
GSE10072	0.9514	0.9062	0.9062
GSE68571	1.0000	1.0000	1.0000
Average ± SEM	0.8343 ± 0.0312	0.8476 ± 0.0279	0.8490 ± 0.0278

Table 46: Experiment 1b: Precision values for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Classifier with higher values of precision are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	0.7950	0.8200	0.8200
GSE62646	0.9667	0.9667	0.9667
GSE41861	0.7911	0.7811	0.7700
GSE20881	0.8278	0.8278	0.7878
GSE3365	0.9625	0.8583	0.8931
GSE16879	0.9690	0.9690	0.9690
GSE15245	0.8400	0.8600	0.8600
GSE6613	0.4600	0.4000	0.4800
GSE20295	0.4750	0.6250	0.6250
GSE30999	0.9750	0.9750	0.9750
GSE55447	0.8350	0.8400	0.8400
GSE19429	0.9558	0.9558	0.9558
GSE9006	0.9067	0.8900	0.8900
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.8514	0.8375	0.8375
GSE35978	0.6719	0.6681	0.7155
GSE53987	0.8400	0.7400	0.7133
GSE12288	0.5182	0.6091	0.5455
GSE15852	0.8300	0.7350	0.7600
GSE42568	0.9518	0.9718	0.9718
GSE29431	0.9833	0.9833	0.9833
GSE18520	0.9800	0.9800	0.9800
GSE19804	0.9333	0.9333	0.9333
GSE10072	0.9500	0.9500	0.9500
GSE68571	0.9875	0.9875	0.9875
Average $\pm$ SEM	0.8503 ± 0.0320	0.8466 ± 0.0300	0.8484 ± 0.0289

Table 47: Experiment 1b: Recall values for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Classifier with higher values of recall are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.DT	BRL.DG
GSE66360	0.7800	0.8163	0.8163
GSE62646	0.9474	0.9474	0.9474
GSE41861	0.7826	0.7933	0.7910
GSE20881	0.8039	0.8200	0.8000
GSE3365	0.9318	0.8639	0.8837
GSE16879	0.9833	0.9833	0.9833
GSE15245	0.8350	0.8713	0.8713
GSE6613	0.4554	0.4211	0.4752
GSE20295	0.4810	0.6098	0.6098
GSE30999	0.9379	0.9326	0.9326
GSE55447	0.8235	0.8140	0.8140
GSE19429	0.9537	0.9589	0.9589
GSE9006	0.8348	0.8319	0.8393
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.8268	0.8391	0.8391
GSE35978	0.6952	0.6884	0.7050
GSE53987	0.7850	0.7400	0.7254
GSE12288	0.5561	0.5956	0.5556
GSE15852	0.8182	0.7901	0.8049
GSE42568	0.9519	0.9619	0.9619
GSE29431	0.9815	0.9815	0.9815
GSE18520	0.9905	0.9905	0.9905
GSE19804	0.8889	0.9032	0.9032
GSE10072	0.9483	0.9244	0.9244
GSE68571	0.9942	0.9942	0.9942
Average $\pm$ SEM	0.8395 ± 0.0309	0.8429 ± 0.0289	0.8443 ± 0.0283

Table 48: Experiment 1b: F-measure values for each dataset, averaged over 10-fold cross-validation, comparing BRL.G, BRL.DT, and BRL.DG using greedy best-first search. Classifier with higher values of F-measure are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

9.3 EXPERIMENT 1C: COMPARING BRL CLASSIFIERS USING BEAM SEARCH

Classification performances achieved by the compared classifiers across the 25 datasets are shown in Tables— 49 (Accuracy), 50 (Precision), 51 (Recall), and 52 (F1 score). In these results, all of BRL.G, BRL.DT, and BRL.DG use the arbitrary 0.5 threshold to predict the positive class. These metrics were largely similar for all the methods compared.

9.4 EXPERIMENT 1: BRL COMPARED TO OTHER STATE-OF-THE-ART CLASSIFIERS

This section contains the complete tables for AUROC 53, AUPRG 54, and Brier scores 55 achieved by all state-of-the-art classifiers compared in experiment 1.

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	77.78	78.79	81.82	85.86	81.82	87.88
GSE62646	92.86	92.86	92.86	92.86	92.86	92.86
GSE41861	71.01	73.19	73.19	76.81	73.19	76.81
GSE20881	76.74	79.65	79.07	81.98	77.33	83.14
GSE3365	90.55	86.61	81.89	87.40	84.25	87.40
GSE16879	97.26	97.26	97.26	97.26	97.26	97.26
GSE15245	73.85	70.77	80.00	73.85	80.00	73.85
GSE6613	47.62	53.33	47.62	43.81	49.52	49.52
GSE20295	55.91	55.91	65.59	60.22	65.59	56.99
GSE30999	93.53	93.53	92.94	92.94	92.94	92.94
GSE55447	71.15	80.77	69.23	82.69	69.23	86.54
GSE19429	91.50	94.50	92.50	94.00	92.50	94.50
GSE9006	75.32	76.62	75.32	77.92	76.62	79.22
GSE48350	100.00	100.00	100.00	100.00	100.00	100.00
GSE5281	80.75	83.23	82.61	89.44	82.61	85.09
GSE35978	60.33	62.30	59.34	58.36	59.67	62.30
GSE53987	66.34	59.51	61.95	60.98	60.49	59.02
GSE12288	59.01	48.65	59.01	56.31	56.76	58.56
GSE15852	81.40	77.91	80.23	83.72	81.40	86.05
GSE42568	91.74	93.39	93.39	94.21	93.39	95.04
GSE29431	96.97	96.97	96.97	96.97	96.97	96.97
GSE18520	98.41	98.41	98.41	98.41	98.41	98.41
GSE19804	88.33	92.50	90.00	91.67	90.00	88.33
GSE10072	94.39	94.39	91.59	91.59	91.59	89.72
GSE68571	98.96	98.96	98.96	98.96	98.96	98.96
Average $\pm$ SEM	81.27 ± 3.04	81.60 ± 3.14	81.67 ± 2.93	82.73 ± 3.12	81.73 ± 2.94	83.09 ± 2.98

Table 49: Experiment 1c: Accuracy for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. Classifier with higher values of accuracies are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	0.7712	0.7867	0.8283	0.8581	0.8283	0.8781
GSE62646	0.9500	0.9500	0.9500	0.9500	0.9500	0.9500
GSE41861	0.7834	0.7792	0.8255	0.8244	0.8273	0.8333
GSE20881	0.8072	0.8326	0.8261	0.8388	0.8358	0.8379
GSE3365	0.9019	0.9221	0.8689	0.9014	0.8788	0.9092
GSE16879	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GSE15245	0.8283	0.8190	0.8817	0.8188	0.8817	0.8188
GSE6613	0.4633	0.5117	0.4887	0.4129	0.5157	0.4723
GSE20295	0.4362	0.5362	0.5679	0.5417	0.5679	0.5100
GSE30999	0.9046	0.9046	0.8935	0.8935	0.8935	0.8957
GSE55447	0.8167	0.8850	0.7950	0.8900	0.7950	0.9150
GSE19429	0.9514	0.9592	0.9625	0.9439	0.9625	0.9487
GSE9006	0.7964	0.8217	0.8068	0.8101	0.8235	0.8470
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8262	0.8663	0.8518	0.9300	0.8518	0.9157
GSE35978	0.7130	0.7131	0.7110	0.6890	0.6909	0.7128
GSE53987	0.7268	0.7036	0.7416	0.7169	0.7402	0.7241
GSE12288	0.6031	0.4670	0.5871	0.5594	0.5700	0.5865
GSE15852	0.8131	0.7964	0.8648	0.8693	0.8731	0.9181
GSE42568	0.9568	0.9485	0.9568	0.9568	0.9568	0.9644
GSE29431	0.9857	0.9857	0.9857	0.9857	0.9857	0.9857
GSE18520	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GSE19804	0.8702	0.9321	0.8905	0.9298	0.8905	0.8923
GSE10072	0.9514	0.9514	0.9062	0.9121	0.9062	0.8812
GSE68571	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Average $\pm$ SEM	0.8343 ± 0.0312	0.8429 ± 0.0311	0.8476 ± 0.0279	0.8493 ± 0.0311	0.8490 ± 0.0278	0.8559 ± 0.0296

Table 50: Experiment 1c: Precision values for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. Classifier with higher values of precision are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	0.7950	0.8000	0.8200	0.8800	0.8200	0.9000
GSE62646	0.9667	0.9667	0.9667	0.9667	0.9667	0.9667
GSE41861	0.7911	0.8378	0.7811	0.8378	0.7700	0.8267
GSE20881	0.8278	0.8178	0.8278	0.8567	0.7878	0.8878
GSE3365	0.9625	0.8806	0.8583	0.9167	0.8931	0.9056
GSE16879	0.9690	0.9690	0.9690	0.9690	0.9690	0.9690
GSE15245	0.8400	0.8000	0.8600	0.8600	0.8600	0.8600
GSE6613	0.4600	0.5400	0.4000	0.3400	0.4800	0.5200
GSE20295	0.4750	0.5750	0.6250	0.4750	0.6250	0.4250
GSE30999	0.9750	0.9750	0.9750	0.9750	0.9750	0.9750
GSE55447	0.8350	0.8750	0.8400	0.9000	0.8400	0.9250
GSE19429	0.9558	0.9833	0.9558	0.9944	0.9558	0.9944
GSE9006	0.9067	0.8867	0.8900	0.9300	0.8900	0.9067
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8514	0.8403	0.8375	0.8847	0.8375	0.8139
GSE35978	0.6719	0.7562	0.6681	0.6869	0.7155	0.7207
GSE53987	0.8400	0.7533	0.7400	0.7733	0.7133	0.7133
GSE12288	0.5182	0.3727	0.6091	0.6364	0.5455	0.5455
GSE15852	0.8300	0.8150	0.7350	0.8650	0.7600	0.8150
GSE42568	0.9518	0.9818	0.9718	0.9818	0.9718	0.9809
GSE29431	0.9833	0.9833	0.9833	0.9833	0.9833	0.9833
GSE18520	0.9800	0.9800	0.9800	0.9800	0.9800	0.9800
GSE19804	0.9333	0.9333	0.9333	0.9167	0.9333	0.9000
GSE10072	0.9500	0.9500	0.9500	0.9500	0.9500	0.9500
GSE68571	0.9875	0.9875	0.9875	0.9875	0.9875	0.9875
Average $\pm$ SEM	0.8503 ± 0.0320	0.8504 ± 0.0316	0.8466 ± 0.0300	0.8619 ± 0.0333	0.8484 ± 0.0289	0.8581 ± 0.0318

Table 51: Experiment 1c: Recall values for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. Classifier with higher values of recall are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	BRL.G	BRL.G-Beam	BRL.DT	BRL.DT-Beam	BRL.DG	BRL.DG-Beam
GSE66360	0.7800	0.7879	0.8163	0.8600	0.8163	0.8800
GSE62646	0.9474	0.9474	0.9474	0.9474	0.9474	0.9474
GSE41861	0.7826	0.8042	0.7933	0.8261	0.7910	0.8242
GSE20881	0.8039	0.8223	0.8200	0.8458	0.8000	0.8585
GSE3365	0.9318	0.8982	0.8639	0.9070	0.8837	0.9059
GSE16879	0.9833	0.9833	0.9833	0.9833	0.9833	0.9833
GSE15245	0.8350	0.8119	0.8713	0.8381	0.8713	0.8381
GSE6613	0.4554	0.5243	0.4211	0.3656	0.4752	0.4952
GSE20295	0.4810	0.5349	0.6098	0.5135	0.6098	0.4789
GSE30999	0.9379	0.9379	0.9326	0.9326	0.9326	0.9326
GSE55447	0.8235	0.8810	0.8140	0.8941	0.8140	0.9176
GSE19429	0.9537	0.9704	0.9589	0.9681	0.9589	0.9707
GSE9006	0.8348	0.8393	0.8319	0.8522	0.8393	0.8571
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8268	0.8439	0.8391	0.9006	0.8391	0.8554
GSE35978	0.6952	0.7294	0.6884	0.6895	0.7050	0.7202
GSE53987	0.7850	0.7314	0.7400	0.7436	0.7254	0.7181
GSE12288	0.5561	0.4184	0.5956	0.5907	0.5556	0.5660
GSE15852	0.8182	0.7865	0.7901	0.8409	0.8049	0.8537
GSE42568	0.9519	0.9623	0.9619	0.9668	0.9619	0.9714
GSE29431	0.9815	0.9815	0.9815	0.9815	0.9815	0.9815
GSE18520	0.9905	0.9905	0.9905	0.9905	0.9905	0.9905
GSE19804	0.8889	0.9256	0.9032	0.9167	0.9032	0.8852
GSE10072	0.9483	0.9483	0.9244	0.9244	0.9244	0.9091
GSE68571	0.9942	0.9942	0.9942	0.9942	0.9942	0.9942
Average ± SEM	0.8395 ± 0.0309	0.8422 ± 0.0314	0.8429 ± 0.0289	0.8509 ± 0.0320	0.8443 ± 0.0283	0.8534 ± 0.0299

Table 52: Experiment 1c: F-measure values for each dataset, averaged over 10-fold cross-validation, comparing greedy best-first and greedy beam search. Classifier with higher values of F-measure are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Logistic	SVM	naïve Bayes	Bagged C4.5	Boosted C4.5	Random Forest	C4.5	RIPPER	PART	BRLDT	BRLDT-Beam
GSE66360	0.8840	0.8900	0.8565	0.9500	0.8320	0.9400	0.6940	0.6995	0.7040	0.8490	0.9135
GSE62646	0.9667	1.0000	1.0000	1.0000	0.9083	0.9750	0.9083	0.9083	0.9083	0.9083	0.9083
GSE41861	0.8682	0.7925	0.7983	0.8539	0.8961	0.8800	0.7164	0.7531	0.7267	0.7717	0.7464
GSE20881	0.9515	0.9430	0.8374	0.9118	0.8342	0.8393	0.7518	0.8246	0.7182	0.8649	0.8629
GSE3365	0.9889	0.9421	0.9197	0.9661	0.9875	0.9677	0.7994	0.7931	0.7940	0.9174	0.9021
GSE16879	0.9667	0.9000	0.9417	1.0000	0.9845	0.9667	0.9845	0.8917	0.9845	0.9845	0.9845
GSE15245	0.4350	0.5450	0.5200	0.6850	0.5800	0.6800	0.6083	0.4550	0.6233	0.7150	0.6400
GSE6613	0.6970	0.6167	0.6940	0.5817	0.6130	0.6633	0.5847	0.5887	0.5590	0.4697	0.4610
GSE20295	0.6675	0.7292	0.6000	0.7729	0.7046	0.6417	0.6413	0.5775	0.6650	0.6221	0.6254
GSE30999	0.9604	0.9819	0.9694	0.9826	0.9458	0.9743	0.9458	0.9646	0.9396	0.9764	0.9764
GSE55447	0.7425	0.5875	0.7000	0.5000	0.6375	0.6175	0.6125	0.4825	0.5525	0.5325	0.8375
GSE19429	0.7281	0.6722	0.8478	0.7019	0.8828	0.8404	0.6254	0.7003	0.5838	0.9001	0.8174
GSE9006	0.9200	0.7583	0.8517	0.8967	0.8150	0.8742	0.7383	0.7483	0.7783	0.8658	0.8750
GSE48350	0.9792	0.9542	0.9167	1.0000	1.0000	0.9750	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.9927	0.9525	0.8957	0.9560	0.8808	0.9606	0.8263	0.8540	0.8170	0.8801	0.9232
GSE35978	0.5442	0.6574	0.7009	0.6642	0.6435	0.6763	0.5129	0.5960	0.5917	0.5436	0.5851
GSE53987	0.5223	0.5567	0.5627	0.5291	0.5493	0.5554	0.5502	0.5029	0.5041	0.5564	0.4847
GSE12288	0.5823	0.5553	0.6043	0.5361	0.5339	0.6175	0.5357	0.5996	0.5207	0.5723	0.5394
GSE15852	0.8488	0.8600	0.9337	0.8988	0.8475	0.9362	0.7631	0.7888	0.7831	0.8194	0.8644
GSE42568	0.9534	0.8750	0.9000	0.9682	0.8705	1.0000	0.8955	0.8455	0.8955	0.8109	0.8159
GSE29431	0.9917	0.9500	0.8917	0.9917	0.9417	0.9900	0.9417	0.9317	0.9417	0.9417	0.9417
GSE18520	0.9900	0.9900	0.9400	0.9900	0.9900	1.0000	0.9900	0.9900	0.9900	0.9900	0.9900
GSE19804	0.9694	0.9417	0.9403	0.9861	0.9431	0.9722	0.8806	0.8931	0.8889	0.9083	0.9250
GSE10072	1.0000	0.9900	0.9700	0.9933	0.9425	0.9967	0.9425	0.9258	0.9425	0.9425	0.9425
GSE68571	1.0000	1.0000	1.0000	0.9937	0.9937	1.0000	0.9937	0.9937	0.9937	0.9938	0.9937
Average ± SEM	0.8460 ± 0.0353	0.8256 ± 0.0325	0.8317 ± 0.0288	0.8524 ± 0.0348	0.8303 ± 0.0306	0.8616 ± 0.0304	0.7777 ± 0.0326	0.7723 ± 0.0341	0.7762 ± 0.0334	0.8135 ± 0.0336	0.8222 ± 0.0336

Table 53: Experiment 1: AUROC by state-of-the-art classifiers, averaged across 10-fold cross-validation for each dataset. Last row contains the average across the datasets and the standard error of mean.

Data	Logistic	SVM	naïve Bayes	Bagged C4.5	Boosted C4.5	Random Forest	C4.5	RIPPER	PART	BRLDT	BRLDT-Beam
GSE66360	0.8003	0.7930	0.7313	0.9104	0.6697	0.8888	0.4212	0.4575	0.4313	0.7234	0.8392
GSE62646	0.8667	1.0000	1.0000	1.0000	0.7833	0.9500	0.7833	0.7833	0.7833	0.7833	0.7833
GSE41861	0.6629	0.5693	0.5877	0.6425	0.7755	0.7009	0.4223	0.4780	0.4263	0.4126	0.4384
GSE20881	0.8987	0.8925	0.6706	0.8094	0.6763	0.6635	0.5146	0.6616	0.4456	0.7311	0.7340
GSE3365	0.9701	0.8749	0.8318	0.9063	0.9611	0.9198	0.5734	0.5187	0.5600	0.7627	0.7401
GSE16879	0.9417	0.8000	0.8917	1.0000	0.8845	0.8917	0.8845	0.7417	0.8845	0.8845	0.8845
GSE15245	0.1148	0.0986	0.1362	0.3100	0.1195	0.3756	0.1481	-0.0308	0.1882	0.3070	0.2362
GSE6613	0.4258	0.2743	0.4289	0.2267	0.2677	0.3792	0.1942	0.2067	0.1447	-0.0205	-0.0488
GSE20295	0.3641	0.5218	0.1994	0.5638	0.4296	0.3386	0.3233	0.1661	0.3819	0.3450	0.2800
GSE30999	0.9321	0.9726	0.9539	0.9768	0.8988	0.9653	0.8988	0.9404	0.8863	0.9599	0.9599
GSE55447	0.3275	0.1875	0.3625	0.1025	0.2000	0.2775	0.2500	-0.0075	0.1400	-0.0075	0.5000
GSE19429	0.3521	0.3235	0.5313	0.1807	0.4480	0.4689	0.1514	0.2630	0.1069	0.6025	0.4856
GSE9006	0.8039	0.5167	0.6672	0.6576	0.5752	0.6846	0.4379	0.4405	0.5267	0.6413	0.6607
GSE48350	0.9438	0.9099	0.8309	1.0000	1.0000	0.9481	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.9895	0.9144	0.8058	0.9009	0.7618	0.9329	0.6788	0.7374	0.6619	0.7763	0.8603
GSE35978	0.1290	0.2787	0.3321	0.2591	0.2303	0.2889	0.0644	0.1473	0.2064	0.0948	0.1380
GSE53987	0.1226	0.1085	0.1153	0.0633	0.0597	0.0923	0.0854	0.0417	0.0286	0.0819	0.0124
GSE12288	0.1724	0.1231	0.2211	0.0440	0.0883	0.2523	0.0855	0.2562	0.0591	0.1860	0.1700
GSE15852	0.6630	0.7629	0.8633	0.8100	0.7237	0.8632	0.5590	0.6043	0.6103	0.6777	0.7399
GSE42568	0.8405	0.7500	0.8000	0.8955	0.7455	1.0000	0.7955	0.6955	0.7955	0.5631	0.5909
GSE29431	0.9417	0.9000	0.7917	0.9417	0.8417	0.9400	0.8417	0.7817	0.8417	0.8417	0.8417
GSE18520	0.9400	0.9400	0.8400	0.9400	0.9400	1.0000	0.9400	0.9400	0.9400	0.9400	0.9400
GSE19804	0.9463	0.9033	0.8923	0.9793	0.9083	0.9532	0.7937	0.8244	0.8020	0.8420	0.8770
GSE10072	1.0000	0.9800	0.9400	0.9850	0.8920	0.9947	0.8920	0.8645	0.8920	0.8920	0.8920
GSE68571	1.0000	1.0000	1.0000	0.9438	0.9438	1.0000	0.9438	0.9438	0.9438	0.9438	0.9438
Average ± SEM	0.6860 ± 0.0648	0.6558 ± 0.0645	0.6570 ± 0.0570	0.6820 ± 0.0697	0.6330 ± 0.0618	0.7108 ± 0.0603	0.5473 ± 0.0638	0.5382 ± 0.0656	0.5475 ± 0.0644	0.5986 ± 0.0655	0.6200 ± 0.0643

Table 54: Experiment 1: AUPRG by state-of-the-art classifiers, averaged across 10-fold cross-validation for each dataset. Last row contains the average across the datasets and the standard error of mean.

Data	Logistic	SVM	naïve Bayes	Bagged C4.5	Boosted C4.5	Random Forest	C4.5	RIPPER	PART	BRL.DT	BRL.DT-Beam
GSE66360	0.1843	0.1100	0.1875	0.1012	0.2044	0.1138	0.2889	0.2475	0.2886	0.1675	0.1409
GSE62646	0.0500	0.0000	0.0000	0.0466	0.0700	0.0440	0.0700	0.0701	0.0700	0.0684	0.0684
GSE41861	0.1910	0.1731	0.2225	0.1499	0.1414	0.1339	0.2482	0.2254	0.2470	0.2492	0.2294
GSE20881	0.1151	0.0523	0.2440	0.1313	0.2096	0.1578	0.2442	0.1724	0.2687	0.2059	0.1786
GSE3365	0.0255	0.0551	0.1019	0.0831	0.0371	0.0807	0.1884	0.1590	0.1887	0.1642	0.1200
GSE16879	0.0338	0.0393	0.0286	0.0265	0.0268	0.0486	0.0268	0.0539	0.0268	0.0257	0.0257
GSE15245	0.4725	0.2286	0.2510	0.1461	0.2312	0.1649	0.3128	0.3079	0.2984	0.1910	0.2308
GSE6613	0.3679	0.3773	0.3897	0.2656	0.4011	0.2486	0.4097	0.3620	0.4243	0.4566	0.5334
GSE20295	0.3495	0.2011	0.4422	0.1955	0.3384	0.2451	0.3467	0.3915	0.3228	0.3234	0.3707
GSE30999	0.0510	0.0176	0.0412	0.0298	0.0531	0.0400	0.0530	0.0354	0.0587	0.0552	0.0552
GSE55447	0.3092	0.1767	0.1556	0.1875	0.3071	0.1627	0.3194	0.3348	0.3402	0.2943	0.1737
GSE19429	0.0795	0.0600	0.0751	0.0799	0.0721	0.0590	0.1436	0.0850	0.1302	0.0708	0.0499
GSE9006	0.1804	0.1571	0.1446	0.1348	0.1826	0.1312	0.2049	0.2162	0.1753	0.2111	0.1981
GSE48350	0.0452	0.0429	0.1024	0.0006	0.0000	0.0612	0.0000	0.0000	0.0000	0.0001	0.0001
GSE5281	0.0683	0.0434	0.1176	0.0898	0.1517	0.0848	0.1804	0.1510	0.1847	0.1717	0.1074
GSE35978	0.3961	0.2990	0.3431	0.2427	0.3300	0.2109	0.3918	0.3057	0.3383	0.3237	0.3570
GSE53987	0.3358	0.3079	0.3482	0.2445	0.3003	0.2459	0.2996	0.3101	0.3549	0.2711	0.3082
GSE12288	0.4459	0.4455	0.3948	0.3047	0.4413	0.2689	0.4541	0.3469	0.4587	0.3319	0.3687
GSE15852	0.2434	0.1389	0.1180	0.1086	0.1490	0.1046	0.2271	0.1998	0.2045	0.1929	0.1479
GSE42568	0.0521	0.0327	0.0250	0.0239	0.0379	0.0244	0.0251	0.0417	0.0251	0.0650	0.0568
GSE29431	0.0599	0.0143	0.0429	0.0248	0.0286	0.0275	0.0286	0.0429	0.0286	0.0279	0.0279
GSE18520	0.0333	0.0167	0.0333	0.0176	0.0167	0.0165	0.0167	0.0167	0.0167	0.0159	0.0159
GSE19804	0.0611	0.0583	0.0749	0.0464	0.0743	0.0593	0.1125	0.0980	0.1122	0.0992	0.0828
GSE10072	0.0100	0.0091	0.0273	0.0258	0.0555	0.0264	0.0556	0.0743	0.0556	0.0726	0.0719
GSE68571	0.0106	0.0000	0.0000	0.0140	0.0111	0.0153	0.0111	0.0111	0.0111	0.0106	0.0106
Average $\pm$ SEM	0.1669 ± 0.0304	0.1223 ± 0.0252	0.1565 ± 0.0274	0.1089 ± 0.0178	0.1548 ± 0.0267	0.1110 ± 0.0164	0.1864 ± 0.0283	0.1704 ± 0.0254	0.1852 ± 0.0284	0.1626 ± 0.0248	0.1572 ± 0.0278

Table 55: Experiment 1: Brier scores by state-of-the-art classifiers, averaged across 10-fold cross-validation for each dataset. Last row contains the average across the datasets and the standard error of mean.

10.0 APPENDIX D: ADDITIONAL RESULTS FROM EXPERIMENT 2

In this appendix, we show some additional results from BRL runs from experiment 2 in section 4.4. These results were not necessary to test the hypotheses evaluated by the experiment and so were moved to this supplement merely as a reference for the curious reader.

10.1 EXPERIMENT 2A: COMPARING BAGGED-BRL-LC TO BRL, C4.5, BAGGED-C4.5, AND BOOSTED-C4.5

Classification performances achieved by the compared classifiers across the 25 datasets are shown in Tables— 56 (Accuracy), 57 (Precision), 58 (Recall), and 59 (F1 score). In these results, BRL methods used the arbitrary 0.5 threshold to predict the positive class. These metrics were largely similar for all the methods compared.

10.2 EXPERIMENT 2B: COMPARING BAGGED-BRL-LC , BAGGED-BRL-BMA, AND BAGGED-BRL-BMC

Classification performances achieved by the compared classifiers across the 25 datasets are shown in Tables— 60 (Accuracy), 61 (Precision), 62 (Recall), and 63 (F1 score). In these results, BRL methods used the arbitrary 0.5 threshold to predict the positive class. These metrics were largely similar for all the methods compared.

Data	Bagged-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	86.87	81.82	70.71	87.88	79.80
GSE62646	97.62	92.86	92.86	95.24	92.86
GSE41861	86.23	73.19	74.64	76.09	82.61
GSE20881	83.72	79.07	74.42	83.72	77.91
GSE3365	92.91	81.89	81.10	90.55	96.06
GSE16879	94.52	97.26	97.26	97.26	97.26
GSE15245	81.54	80.00	67.69	81.54	73.85
GSE6613	57.14	47.62	58.10	55.24	57.14
GSE20295	64.52	65.59	62.37	68.82	63.44
GSE30999	96.47	92.94	94.71	97.06	94.71
GSE55447	73.08	69.23	69.23	78.85	69.23
GSE19429	92.00	92.50	85.50	88.00	92.50
GSE9006	88.31	75.32	79.22	81.82	80.52
GSE48350	100.00	100.00	100.00	100.00	100.00
GSE5281	87.58	82.61	80.75	87.58	84.47
GSE35978	74.75	59.34	57.05	65.25	64.59
GSE53987	63.90	61.95	64.39	66.34	64.88
GSE12288	52.70	59.01	52.25	50.00	54.05
GSE15852	87.21	80.23	76.74	83.72	84.88
GSE42568	96.69	93.39	97.52	97.52	95.87
GSE29431	98.48	96.97	96.97	96.97	96.97
GSE18520	100.00	98.41	98.41	98.41	98.41
GSE19804	92.50	90.00	88.33	93.33	92.50
GSE10072	97.20	91.59	94.39	95.33	94.39
GSE68571	98.96	98.96	98.96	98.96	98.96
Average $\pm$ SEM	85.80 ± 2.77	81.67 ± 2.93	80.54 ± 3.01	84.62 ± 2.82	83.51 ± 2.87

Table 56: Experiment 2a: Accuracy for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of accuracies are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Bagged-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	0.9050	0.8283	0.6800	0.9133	0.7905
GSE62646	0.9750	0.9500	0.9500	0.9750	0.9500
GSE41861	0.8689	0.8255	0.8115	0.8362	0.8556
GSE20881	0.8771	0.8261	0.7750	0.8497	0.8125
GSE3365	0.9435	0.8689	0.8728	0.9219	0.9733
GSE16879	0.9714	1.0000	1.0000	1.0000	1.0000
GSE15245	0.8564	0.8817	0.8317	0.8229	0.7936
GSE6613	0.5410	0.4887	0.5627	0.5321	0.5471
GSE20295	0.5762	0.5679	0.6267	0.6850	0.5917
GSE30999	0.9657	0.8935	0.9239	0.9857	0.9239
GSE55447	0.7883	0.7950	0.8417	0.8550	0.8100
GSE19429	0.9381	0.9625	0.9334	0.9208	0.9522
GSE9006	0.8848	0.8068	0.8719	0.8529	0.8695
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8686	0.8518	0.8766	0.8766	0.8746
GSE35978	0.7838	0.7110	0.6852	0.7481	0.7445
GSE53987	0.7204	0.7416	0.7502	0.7467	0.7364
GSE12288	0.5385	0.5871	0.5189	0.4900	0.5480
GSE15852	0.8967	0.8648	0.7931	0.8700	0.8531
GSE42568	0.9652	0.9568	0.9826	0.9742	0.9652
GSE29431	0.9857	0.9857	0.9857	0.9857	0.9857
GSE18520	1.0000	1.0000	1.0000	1.0000	1.0000
GSE19804	0.9214	0.8905	0.8905	0.9179	0.9321
GSE10072	0.9571	0.9062	0.9514	0.9657	0.9514
GSE68571	1.0000	1.0000	1.0000	1.0000	1.0000
Average $\pm$ SEM	0.8691 ± 0.0279	0.8476 ± 0.0279	0.8446 ± 0.0283	0.8690 ± 0.0278	0.8584 ± 0.0278

Table 57: Experiment 2a: Precision for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of precision are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Bagged-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	0.8350	0.8200	0.7800	0.8600	0.8200
GSE62646	1.0000	0.9667	0.9667	0.9667	0.9667
GSE41861	0.9333	0.7811	0.8033	0.8133	0.8900
GSE20881	0.8489	0.8278	0.7967	0.8767	0.8178
GSE3365	0.9528	0.8583	0.8306	0.9403	0.9625
GSE16879	0.9690	0.9690	0.9690	0.9690	0.9690
GSE15245	0.9200	0.8600	0.7467	0.9800	0.9000
GSE6613	0.5000	0.4000	0.6600	0.5600	0.5600
GSE20295	0.6250	0.6250	0.6250	0.6500	0.6500
GSE30999	0.9625	0.9750	0.9750	0.9528	0.9750
GSE55447	0.9050	0.8400	0.7250	0.9050	0.7750
GSE19429	0.9781	0.9558	0.9064	0.9509	0.9670
GSE9006	0.9633	0.8900	0.8500	0.9067	0.8733
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.9306	0.8375	0.7778	0.9083	0.8486
GSE35978	0.8688	0.6681	0.6690	0.7324	0.7214
GSE53987	0.8267	0.7400	0.7667	0.8200	0.8133
GSE12288	0.5091	0.6091	0.6455	0.5636	0.5818
GSE15852	0.8650	0.7350	0.7350	0.8150	0.8550
GSE42568	1.0000	0.9718	0.9909	1.0000	0.9909
GSE29431	1.0000	0.9833	0.9833	0.9833	0.9833
GSE18520	1.0000	0.9800	0.9800	0.9800	0.9800
GSE19804	0.9500	0.9333	0.9000	0.9667	0.9333
GSE10072	1.0000	0.9500	0.9500	0.9500	0.9500
GSE68571	0.9875	0.9875	0.9875	0.9875	0.9875
Average $\pm$ SEM	0.8932 ± 0.0287	0.8466 ± 0.0300	0.8408 ± 0.0250	0.8815 ± 0.0260	0.8709 ± 0.0258

Table 58: Experiment 2a: Recall for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of recall are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

0.8632	0.8163	0.7238	0.8750	0.8000	
0.9825	0.9474	0.9474	0.9643	0.9474	
0.8995	0.7933	0.8066	0.8177	0.8710	
0.8571	0.8200	0.7822	0.8614	0.8100	
0.9474	0.8639	0.8554	0.9302	0.9704	
0.9672	0.9833	0.9833	0.9833	0.9833	
0.8868	0.8713	0.7835	0.8929	0.8440	
0.5263	0.4211	0.6000	0.5437	0.5545	
0.6024	0.6098	0.6024	0.6582	0.6118	
0.9647	0.9326	0.9486	0.9701	0.9486	
0.8444	0.8140	0.7949	0.8736	0.8049	
0.9572	0.9589	0.9197	0.9355	0.9593	
0.9189	0.8319	0.8491	0.8727	0.8598	
1.0000	1.0000	1.0000	1.0000	1.0000	
0.8901	0.8391	0.8144	0.8876	0.8555	
0.8222	0.6884	0.6765	0.7389	0.7327	
0.7702	0.7400	0.7591	0.7810	0.7722	
0.5161	0.5956	0.5726	0.5277	0.5565	
0.8706	0.7901	0.7619	0.8333	0.8506	
0.9811	0.9619	0.9856	0.9858	0.9763	
0.9908	0.9815	0.9815	0.9815	0.9815	
1.0000	0.9905	0.9905	0.9905	0.9905	
0.9268	0.9032	0.8852	0.9355	0.9256	
0.9748	0.9244	0.9483	0.9565	0.9483	
0.9942	0.9942	0.9942	0.9942	0.9942	
Average ± SEM	0.8782 ± 0.0278	0.8429 ± 0.0289	0.8387 ± 0.0267	0.8716 ± 0.0265	0.8619 ± 0.0267

Table 59: Experiment 2a: F-measure for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of F-measure are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Bagged-BRL.DT-LC	Bagged-BRL.DT-BMA	Bagged-BRL.DT-BMC
GSE66360	86.87	80.81	87.88
GSE62646	97.62	97.62	97.62
GSE41861	86.23	78.99	86.96
GSE20881	83.72	76.16	84.88
GSE3365	92.91	89.76	93.70
GSE16879	94.52	94.52	94.52
GSE15245	81.54	80.00	81.54
GSE6613	57.14	60.00	57.14
GSE20295	64.52	61.29	66.67
GSE30999	96.47	96.47	96.47
GSE55447	73.08	71.15	75.00
GSE19429	92.00	90.00	92.00
GSE9006	88.31	80.52	85.71
GSE48350	100.00	100.00	100.00
GSE5281	87.58	83.23	87.58
GSE35978	74.75	64.26	73.11
GSE53987	63.90	59.51	63.41
GSE12288	52.70	52.70	50.90
GSE15852	87.21	86.05	86.05
GSE42568	96.69	96.69	96.69
GSE29431	98.48	98.48	96.97
GSE18520	100.00	100.00	98.41
GSE19804	92.50	90.83	93.33
GSE10072	97.20	97.20	95.33
GSE68571	98.96	98.96	98.96
Average $\pm$ SEM	85.80 ± 2.77	83.41 ± 2.94	85.63 ± 2.76

Table 60: Experiment 2b: Accuracy for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of accuracies are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Bagged-BRL.DT-LC	Bagged-BRL.DT-BMA	Bagged-BRL.DT-BMC
GSE66360	0.9050	0.8348	0.8833
GSE62646	0.9750	0.9750	0.9750
GSE41861	0.8689	0.8498	0.8823
GSE20881	0.8771	0.7926	0.8924
GSE3365	0.9435	0.9058	0.9467
GSE16879	0.9714	0.9714	0.9714
GSE15245	0.8564	0.8495	0.8564
GSE6613	0.5410	0.6112	0.5281
GSE20295	0.5762	0.5283	0.5733
GSE30999	0.9657	0.9675	0.9657
GSE55447	0.7883	0.7800	0.8050
GSE19429	0.9381	0.9314	0.9381
GSE9006	0.8848	0.8581	0.8514
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.8686	0.8581	0.8745
GSE35978	0.7838	0.7190	0.7791
GSE53987	0.7204	0.7193	0.7311
GSE12288	0.5385	0.5386	0.5061
GSE15852	0.8967	0.8721	0.8800
GSE42568	0.9652	0.9652	0.9652
GSE29431	0.9857	0.9857	0.9714
GSE18520	1.0000	1.0000	1.0000
GSE19804	0.9214	0.9214	0.9214
GSE10072	0.9571	0.9571	0.9657
GSE68571	1.0000	1.0000	1.0000
Average $\pm$ SEM	0.8691 ± 0.0279	0.8557 ± 0.0280	0.8665 ± 0.0287

Table 61: Experiment 2b: Precision for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of precision are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Bagged-BRL.DT-LC	Bagged-BRL.DT-BMA	Bagged-BRL.DT-BMC
GSE66360	0.8350	0.7950	0.8750
GSE62646	1.0000	1.0000	1.0000
GSE41861	0.9333	0.8356	0.9344
GSE20881	0.8489	0.8089	0.8489
GSE3365	0.9528	0.9542	0.9653
GSE16879	0.9690	0.9690	0.9690
GSE15245	0.9200	0.9200	0.9200
GSE6613	0.5000	0.5400	0.4800
GSE20295	0.6250	0.5500	0.6500
GSE30999	0.9625	0.9625	0.9625
GSE55447	0.9050	0.8850	0.9050
GSE19429	0.9781	0.9614	0.9781
GSE9006	0.9633	0.8667	0.9633
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.9306	0.8500	0.9194
GSE35978	0.8688	0.7745	0.8388
GSE53987	0.8267	0.7333	0.7933
GSE12288	0.5091	0.5364	0.4909
GSE15852	0.8650	0.8850	0.8650
GSE42568	1.0000	1.0000	1.0000
GSE29431	1.0000	1.0000	1.0000
GSE18520	1.0000	1.0000	0.9800
GSE19804	0.9500	0.9167	0.9667
GSE10072	1.0000	1.0000	0.9500
GSE68571	0.9875	0.9875	0.9875
Average $\pm$ SEM	0.8932 ± 0.0287	0.8693 ± 0.0293	0.8897 ± 0.0291

Table 62: Experiment 2b: Recall for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of recall are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Bagged-BRL.DT-LC	Bagged-BRL.DT-BMA	Bagged-BRL.DT-BMC
GSE66360	0.8632	0.8041	0.8776
GSE62646	0.9825	0.9825	0.9825
GSE41861	0.8995	0.8398	0.9043
GSE20881	0.8571	0.7960	0.8660
GSE3365	0.9474	0.9257	0.9535
GSE16879	0.9672	0.9672	0.9672
GSE15245	0.8868	0.8785	0.8868
GSE6613	0.5263	0.5625	0.5161
GSE20295	0.6024	0.5641	0.6265
GSE30999	0.9647	0.9647	0.9647
GSE55447	0.8444	0.8315	0.8539
GSE19429	0.9572	0.9462	0.9572
GSE9006	0.9189	0.8598	0.9027
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.8901	0.8457	0.8889
GSE35978	0.8222	0.7447	0.8075
GSE53987	0.7702	0.7261	0.7604
GSE12288	0.5161	0.5291	0.4977
GSE15852	0.8706	0.8636	0.8605
GSE42568	0.9811	0.9811	0.9811
GSE29431	0.9908	0.9908	0.9818
GSE18520	1.0000	1.0000	0.9905
GSE19804	0.9268	0.9091	0.9355
GSE10072	0.9748	0.9748	0.9565
GSE68571	0.9942	0.9942	0.9942
Average $\pm$ SEM	0.8782 ± 0.0278	0.8593 ± 0.0283	0.8765 ± 0.0280

Table 63: Experiment 2b: F-measure for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of accuracies are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

10.3 EXPERIMENT 2C: COMPARING BOOSTED-BRL-LC TO BRL, C4.5, BAGGED-C4.5, AND BOOSTED-C4.5

Classification performances achieved by the compared classifiers across the 25 datasets are shown in Tables— 64 (Accuracy), 65 (Precision), 66 (Recall), and 67 (F1 score). In these results, BRL methods used the arbitrary 0.5 threshold to predict the positive class. These metrics were largely similar for all the methods compared.

10.4 EXPERIMENT 2D: COMPARING BOOSTED-BRL-LC , BOOSTED-BRL-BMA, AND BOOSTED-BRL-BMC

Classification performances achieved by the compared classifiers across the 25 datasets are shown in Tables— 68 (Accuracy), 69 (Precision), 70 (Recall), and 71 (F1 score). In these results, BRL methods used the arbitrary 0.5 threshold to predict the positive class. These metrics were largely similar for all the methods compared.

10.5 EXPERIMENT 2: BAGGED-BRL.DT-BMC COMPARED TO OTHER STATE-OF-THE-ART CLASSIFIERS

This section contains the complete tables for AUROC 72, AUPRG 73, and Brier scores 74 achieved by all state-of-the-art classifiers compared in experiment 1.

Data	Boosted-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	83.84	81.82	70.71	87.88	79.80
GSE62646	92.86	92.86	92.86	95.24	92.86
GSE41861	73.19	73.19	74.64	76.09	82.61
GSE20881	81.98	79.07	74.42	83.72	77.91
GSE3365	88.19	81.89	81.10	90.55	96.06
GSE16879	97.26	97.26	97.26	97.26	97.26
GSE15245	73.85	80.00	67.69	81.54	73.85
GSE6613	46.67	47.62	58.10	55.24	57.14
GSE20295	50.54	65.59	62.37	68.82	63.44
GSE30999	97.06	92.94	94.71	97.06	94.71
GSE55447	75.00	69.23	69.23	78.85	69.23
GSE19429	93.00	92.50	85.50	88.00	92.50
GSE9006	84.42	75.32	79.22	81.82	80.52
GSE48350	100.00	100.00	100.00	100.00	100.00
GSE5281	83.85	82.61	80.75	87.58	84.47
GSE35978	59.34	59.34	57.05	65.25	64.59
GSE53987	71.71	61.95	64.39	66.34	64.88
GSE12288	58.56	59.01	52.25	50.00	54.05
GSE15852	83.72	80.23	76.74	83.72	84.88
GSE42568	95.04	93.39	97.52	97.52	95.87
GSE29431	96.97	96.97	96.97	96.97	96.97
GSE18520	98.41	98.41	98.41	98.41	98.41
GSE19804	88.33	90.00	88.33	93.33	92.50
GSE10072	94.39	91.59	94.39	95.33	94.39
GSE68571	98.96	98.96	98.96	98.96	98.96
Average ± SEM	82.68 ± 3.11	81.67 ± 2.93	80.54 ± 3.01	84.62 ± 2.82	83.51 ± 2.87

Table 64: Experiment 2c: Accuracy for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of accuracies are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Boosted-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	0.8681	0.8283	0.6800	0.9133	0.7905
GSE62646	0.9500	0.9500	0.9500	0.9750	0.9500
GSE41861	0.7885	0.8255	0.8115	0.8362	0.8556
GSE20881	0.8911	0.8261	0.7750	0.8497	0.8125
GSE3365	0.9085	0.8689	0.8728	0.9219	0.9733
GSE16879	1.0000	1.0000	1.0000	1.0000	1.0000
GSE15245	0.8195	0.8817	0.8317	0.8229	0.7936
GSE6613	0.1565	0.4887	0.5627	0.5321	0.5471
GSE20295	0.3429	0.5679	0.6267	0.6850	0.5917
GSE30999	0.9650	0.8935	0.9239	0.9857	0.9239
GSE55447	0.8450	0.7950	0.8417	0.8550	0.8100
GSE19429	0.9384	0.9625	0.9334	0.9208	0.9522
GSE9006	0.9014	0.8068	0.8719	0.8529	0.8695
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8472	0.8518	0.8766	0.8766	0.8746
GSE35978	0.6805	0.7110	0.6852	0.7481	0.7445
GSE53987	0.7385	0.7416	0.7502	0.7467	0.7364
GSE12288	0.5266	0.5871	0.5189	0.4900	0.5480
GSE15852	0.8579	0.8648	0.7931	0.8700	0.8531
GSE42568	0.9644	0.9568	0.9826	0.9742	0.9652
GSE29431	0.9857	0.9857	0.9857	0.9857	0.9857
GSE18520	1.0000	1.0000	1.0000	1.0000	1.0000
GSE19804	0.8940	0.8905	0.8905	0.9179	0.9321
GSE10072	0.9514	0.9062	0.9514	0.9657	0.9514
GSE68571	1.0000	1.0000	1.0000	1.0000	1.0000
Average ± SEM	0.8328 ± 0.0419	0.8476 ± 0.0279	0.8446 ± 0.0283	0.8690 ± 0.0278	0.8584 ± 0.0278

Table 65: Experiment 2c: Precision for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of precision are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Boosted-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	0.8200	0.8200	0.7800	0.8600	0.8200
GSE62646	0.9667	0.9667	0.9667	0.9667	0.9667
GSE41861	0.8356	0.7811	0.8033	0.8133	0.8900
GSE20881	0.8089	0.8278	0.7967	0.8767	0.8178
GSE3365	0.9306	0.8583	0.8306	0.9403	0.9625
GSE16879	0.9690	0.9690	0.9690	0.9690	0.9690
GSE15245	0.8600	0.8600	0.7467	0.9800	0.9000
GSE6613	0.2200	0.4000	0.6600	0.5600	0.5600
GSE20295	0.4250	0.6250	0.6250	0.6500	0.6500
GSE30999	0.9750	0.9750	0.9750	0.9528	0.9750
GSE55447	0.8550	0.8400	0.7250	0.9050	0.7750
GSE19429	0.9889	0.9558	0.9064	0.9509	0.9670
GSE9006	0.8867	0.8900	0.8500	0.9067	0.8733
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8653	0.8375	0.7778	0.9083	0.8486
GSE35978	0.7160	0.6681	0.6690	0.7324	0.7214
GSE53987	0.9533	0.7400	0.7667	0.8200	0.8133
GSE12288	0.3636	0.6091	0.6455	0.5636	0.5818
GSE15852	0.8550	0.7350	0.7350	0.8150	0.8550
GSE42568	0.9818	0.9718	0.9909	1.0000	0.9909
GSE29431	0.9833	0.9833	0.9833	0.9833	0.9833
GSE18520	0.9800	0.9800	0.9800	0.9800	0.9800
GSE19804	0.9000	0.9333	0.9000	0.9667	0.9333
GSE10072	0.9500	0.9500	0.9500	0.9500	0.9500
GSE68571	0.9875	0.9875	0.9875	0.9875	0.9875
Average $\pm$ SEM	0.8431 ± 0.0413	0.8466 ± 0.0300	0.8408 ± 0.0250	0.8815 ± 0.0260	0.8709 ± 0.0258

Table 66: Experiment 2c: Recall for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of recall are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Boosted-BRL.DT-LC	BRL.DT	C4.5	Bagged-C4.5	Boosted-C4.5
GSE66360	0.8333	0.8163	0.7238	0.8750	0.8000
GSE62646	0.9474	0.9474	0.9474	0.9643	0.9474
GSE41861	0.8042	0.7933	0.8066	0.8177	0.8710
GSE20881	0.8377	0.8200	0.7822	0.8614	0.8100
GSE3365	0.9133	0.8639	0.8554	0.9302	0.9704
GSE16879	0.9833	0.9833	0.9833	0.9833	0.9833
GSE15245	0.8381	0.8713	0.7835	0.8929	0.8440
GSE6613	0.2821	0.4211	0.6000	0.5437	0.5545
GSE20295	0.4250	0.6098	0.6024	0.6582	0.6118
GSE30999	0.9708	0.9326	0.9486	0.9701	0.9486
GSE55447	0.8471	0.8140	0.7949	0.8736	0.8049
GSE19429	0.9628	0.9589	0.9197	0.9355	0.9593
GSE9006	0.8868	0.8319	0.8491	0.8727	0.8598
GSE48350	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.8523	0.8391	0.8144	0.8876	0.8555
GSE35978	0.7033	0.6884	0.6765	0.7389	0.7327
GSE53987	0.8314	0.7400	0.7591	0.7810	0.7722
GSE12288	0.4651	0.5956	0.5726	0.5277	0.5565
GSE15852	0.8409	0.7901	0.7619	0.8333	0.8506
GSE42568	0.9714	0.9619	0.9856	0.9858	0.9763
GSE29431	0.9815	0.9815	0.9815	0.9815	0.9815
GSE18520	0.9905	0.9905	0.9905	0.9905	0.9905
GSE19804	0.8852	0.9032	0.8852	0.9355	0.9256
GSE10072	0.9483	0.9244	0.9483	0.9565	0.9483
GSE68571	0.9942	0.9942	0.9942	0.9942	0.9942
Average ± SEM	0.8398 ± 0.0374	0.8429 ± 0.0289	0.8387 ± 0.0267	0.8716 ± 0.0265	0.8619 ± 0.0267

Table 67: Experiment 2c: F-measure for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of F-measure are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Boosted-BRL.DT-LC	Boosted-BRL.DT-BMA	Boosted-BRL.DT-BMC
GSE66360	83.84	52.53	83.84
GSE62646	92.86	66.67	92.86
GSE41861	73.19	65.94	75.36
GSE20881	81.98	57.56	83.14
GSE3365	88.19	66.93	88.98
GSE16879	97.26	83.56	97.26
GSE15245	73.85	78.46	73.85
GSE6613	46.67	52.38	43.81
GSE20295	50.54	43.01	59.14
GSE30999	97.06	61.76	95.88
GSE55447	75.00	80.77	75.00
GSE19429	93.00	91.50	93.00
GSE9006	84.42	68.83	83.12
GSE48350	100.00	63.24	100.00
GSE5281	83.85	54.04	84.47
GSE35978	59.34	67.21	61.97
GSE53987	71.71	73.17	60.49
GSE12288	58.56	50.45	56.76
GSE15852	83.72	54.65	83.72
GSE42568	95.04	85.95	95.04
GSE29431	96.97	81.82	96.97
GSE18520	98.41	84.13	98.41
GSE19804	88.33	86.67	90.83
GSE10072	94.39	54.21	94.39
GSE68571	98.96	89.58	98.96
Average $\pm$ SEM	82.68 ± 3.11	68.60 ± 2.84	82.69 ± 3.13

Table 68: Experiment 2d: Accuracy for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of accuracies are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Boosted-BRL.DT-LC	Boosted-BRL.DT-BMA	Boosted-BRL.DT-BMC
GSE66360	0.8681	0.1250	0.8714
GSE62646	0.9500	0.6700	0.9500
GSE41861	0.7885	0.6599	0.8095
GSE20881	0.8911	0.5758	0.8755
GSE3365	0.9085	0.6692	0.9060
GSE16879	1.0000	0.8375	1.0000
GSE15245	0.8195	0.7881	0.8329
GSE6613	0.1565	0.0000	0.2971
GSE20295	0.3429	0.4311	0.5160
GSE30999	0.9650	0.4882	0.9539
GSE55447	0.8450	0.8067	0.8450
GSE19429	0.9384	0.9150	0.9384
GSE9006	0.9014	0.6893	0.8848
GSE48350	1.0000	0.6333	1.0000
GSE5281	0.8472	0.5404	0.8444
GSE35978	0.6805	0.6720	0.7077
GSE53987	0.7385	0.7321	0.7090
GSE12288	0.5266	0.0000	0.6090
GSE15852	0.8579	0.3989	0.8579
GSE42568	0.9644	0.8596	0.9644
GSE29431	0.9857	0.8190	0.9857
GSE18520	1.0000	0.8405	1.0000
GSE19804	0.8940	0.8774	0.9190
GSE10072	0.9514	0.5418	0.9514
GSE68571	1.0000	0.8956	1.0000
Average $\pm$ SEM	0.8691 ± 0.0279	0.8557 ± 0.0280	0.8665 ± 0.0287

Table 69: Experiment 2d: Precision for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of precision are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Boosted-BRL.DT-LC	Boosted-BRL.DT-BMA	Boosted-BRL.DT-BMC
GSE66360	0.8200	0.1750	0.8200
GSE62646	0.9667	1.0000	0.9667
GSE41861	0.8356	1.0000	0.8356
GSE20881	0.8089	1.0000	0.8389
GSE3365	0.9306	1.0000	0.9431
GSE16879	0.9690	1.0000	0.9690
GSE15245	0.8600	1.0000	0.8400
GSE6613	0.2200	0.0000	0.5400
GSE20295	0.4250	1.0000	0.6750
GSE30999	0.9750	0.6778	0.9625
GSE55447	0.8550	1.0000	0.8550
GSE19429	0.9889	1.0000	0.9889
GSE9006	0.8867	1.0000	0.8867
GSE48350	1.0000	1.0000	1.0000
GSE5281	0.8653	1.0000	0.8875
GSE35978	0.7160	1.0000	0.7576
GSE53987	0.9533	1.0000	0.7333
GSE12288	0.3636	0.0000	0.5091
GSE15852	0.8550	0.7000	0.8550
GSE42568	0.9818	1.0000	0.9818
GSE29431	0.9833	1.0000	0.9833
GSE18520	0.9800	1.0000	0.9800
GSE19804	0.9000	0.8833	0.9167
GSE10072	0.9500	1.0000	0.9500
GSE68571	0.9875	1.0000	0.9875
Average $\pm$ SEM	0.8431 ± 0.0413	0.8574 ± 0.0630	0.8665 ± 0.0270

Table 70: Experiment 2d: Recall for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of recall are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Boosted-BRL.DT-LC	Boosted-BRL.DT-BMA	Boosted-BRL.DT-BMC
GSE66360	0.8333	0.2540	0.8333
GSE62646	0.9474	0.8000	0.9474
GSE41861	0.8042	0.7948	0.8172
GSE20881	0.8377	0.7306	0.8513
GSE3365	0.9133	0.8019	0.9195
GSE16879	0.9833	0.9104	0.9833
GSE15245	0.8381	0.8793	0.8350
GSE6613	0.2821	0.0000	0.4779
GSE20295	0.4250	0.6015	0.5870
GSE30999	0.9708	0.6328	0.9591
GSE55447	0.8471	0.8936	0.8471
GSE19429	0.9628	0.9556	0.9628
GSE9006	0.8868	0.8154	0.8785
GSE48350	1.0000	0.7748	1.0000
GSE5281	0.8523	0.7016	0.8603
GSE35978	0.7033	0.8039	0.7277
GSE53987	0.8314	0.8451	0.7309
GSE12288	0.4651	0.0000	0.5385
GSE15852	0.8409	0.5979	0.8409
GSE42568	0.9714	0.9244	0.9714
GSE29431	0.9815	0.9000	0.9815
GSE18520	0.9905	0.9138	0.9905
GSE19804	0.8852	0.8689	0.9091
GSE10072	0.9483	0.7030	0.9483
GSE68571	0.9942	0.9451	0.9942
Average $\pm$ SEM	0.8398 ± 0.0374	0.7219 ± 0.0529	0.8557 ± 0.0289

Table 71: Experiment 2d: F-measure for each dataset, averaged over 10-fold cross-validation, using state-of-the-art rule learning classifiers compared to BRL. Classifier with higher values of F-measure are better performing for a given dataset. The last row calculates the average for each classifiers across 25 datasets and also reports the standard error of mean.

Data	Logistic	SVM	naïve Bayes	Bagged C4.5	Boosted C4.5	Random Forest	C4.5	RIPPER	PART	BRL.DT	BRL.DT-Beam	Bagged-BRL.DT-BMC
GSE66360	0.8840	0.8900	0.8565	0.9500	0.8320	0.9400	0.6940	0.6995	0.7040	0.8490	0.9135	0.9630
GSE62646	0.9667	1.0000	1.0000	1.0000	0.9083	0.9750	0.9083	0.9083	0.9083	0.9083	0.9083	1.0000
GSE41861	0.8682	0.7925	0.7983	0.8539	0.8961	0.8800	0.7164	0.7531	0.7267	0.7717	0.7464	0.9358
GSE20881	0.9515	0.9430	0.8374	0.9118	0.8342	0.8393	0.7518	0.8246	0.7182	0.8649	0.8629	0.9158
GSE3365	0.9889	0.9421	0.9197	0.9661	0.9875	0.9677	0.7994	0.7931	0.7940	0.9174	0.9021	0.9764
GSE16879	0.9667	0.9000	0.9417	1.0000	0.9845	0.9667	0.9845	0.8917	0.9845	0.9845	0.9845	0.9833
GSE15245	0.4350	0.5450	0.5200	0.6850	0.5800	0.6800	0.6083	0.4550	0.6233	0.7150	0.6400	0.6800
GSE6613	0.6970	0.6167	0.6940	0.5817	0.6130	0.6633	0.5847	0.5887	0.5590	0.4697	0.4610	0.6247
GSE20295	0.6675	0.7292	0.6000	0.7729	0.7046	0.6417	0.6413	0.5775	0.6650	0.6221	0.6254	0.7533
GSE30999	0.9604	0.9819	0.9694	0.9826	0.9458	0.9743	0.9458	0.9646	0.9396	0.9764	0.9764	0.9799
GSE55447	0.7425	0.5875	0.7000	0.5000	0.6375	0.6175	0.6125	0.4825	0.5525	0.5325	0.8375	0.6900
GSE19429	0.7281	0.6722	0.8478	0.7019	0.8828	0.8404	0.6254	0.7003	0.5838	0.9001	0.8174	0.9670
GSE9006	0.9200	0.7583	0.8517	0.8967	0.8150	0.8742	0.7383	0.7483	0.7783	0.8658	0.8750	0.9400
GSE48350	0.9792	0.9542	0.9167	1.0000	1.0000	0.9750	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.9927	0.9525	0.8957	0.9560	0.8808	0.9606	0.8263	0.8540	0.8170	0.8801	0.9232	0.9597
GSE35978	0.5442	0.6574	0.7009	0.6642	0.6435	0.6763	0.5129	0.5960	0.5917	0.5436	0.5851	0.7338
GSE53987	0.5223	0.5567	0.5627	0.5291	0.5493	0.5554	0.5502	0.5029	0.5041	0.5564	0.4847	0.4859
GSE12288	0.5823	0.5553	0.6043	0.5361	0.5339	0.6175	0.5357	0.5996	0.5207	0.5723	0.5394	0.5630
GSE15852	0.8488	0.8600	0.9337	0.8988	0.8475	0.9362	0.7631	0.7888	0.7831	0.8194	0.8644	0.9325
GSE42568	0.9534	0.8750	0.9000	0.9682	0.8705	1.0000	0.8955	0.8455	0.8955	0.8109	0.8159	0.9432
GSE29431	0.9917	0.9500	0.8917	0.9917	0.9417	0.9900	0.9417	0.9317	0.9417	0.9417	0.9417	1.0000
GSE18520	0.9900	0.9900	0.9400	0.9900	0.9900	1.0000	0.9900	0.9900	0.9900	0.9900	0.9900	1.0000
GSE19804	0.9694	0.9417	0.9403	0.9861	0.9431	0.9722	0.8806	0.8931	0.8889	0.9083	0.9250	0.9583
GSE100072	1.0000	0.9900	0.9700	0.9933	0.9425	0.9967	0.9425	0.9258	0.9425	0.9425	0.9425	0.9900
GSE68571	1.0000	1.0000	1.0000	0.9937	0.9937	1.0000	0.9937	0.9937	0.9937	0.9938	0.9937	1.0000
Average ± SEM	0.8460 ± 0.0353	0.8256 ± 0.0325	0.8317 ± 0.0288	0.8524 ± 0.0348	0.8303 ± 0.0306	0.8616 ± 0.0304	0.7777 ± 0.0326	0.7723 ± 0.0341	0.7762 ± 0.0334	0.8135 ± 0.0336	0.8222 ± 0.0336	0.8790 ± 0.0314

Table 72: Experiment 2: AUROC by state-of-the-art classifiers, averaged across 10-fold cross-validation for each dataset. Last row contains the average across the datasets and the standard error of mean.

Data	Logistic	SVM	naïve Bayes	Bagged C4.5	Boosted C4.5	Random Forest	C4.5	RIPPER	PART	BRL.DT	BRL.DT-Beam	Bagged-BRL.DT-BMC
GSE66360	0.8003	0.7930	0.7313	0.9104	0.6697	0.8888	0.4212	0.4575	0.4313	0.7234	0.8392	0.9357
GSE62646	0.8667	1.0000	1.0000	1.0000	0.7833	0.9500	0.7833	0.7833	0.7833	0.7833	0.7833	1.0000
GSE41861	0.6629	0.5693	0.5877	0.6425	0.7755	0.7009	0.4223	0.4780	0.4263	0.4126	0.4384	0.8471
GSE20881	0.8987	0.8925	0.6706	0.8094	0.6763	0.6635	0.5146	0.6616	0.4456	0.7311	0.7340	0.8131
GSE3365	0.9701	0.8749	0.8318	0.9063	0.9611	0.9198	0.5734	0.5187	0.5600	0.7627	0.7401	0.9354
GSE16879	0.9417	0.8000	0.8917	1.0000	0.8845	0.8917	0.8845	0.7417	0.8845	0.8845	0.8845	0.8917
GSE15245	0.1148	0.0986	0.1362	0.3100	0.1195	0.3756	0.1481	-0.0308	0.1882	0.3070	0.2362	0.4800
GSE6613	0.4258	0.2743	0.4289	0.2267	0.2677	0.3792	0.1942	0.2067	0.1447	-0.0205	-0.0488	0.2767
GSE20295	0.3641	0.5218	0.1994	0.5638	0.4296	0.3386	0.3233	0.1661	0.3819	0.3450	0.2800	0.5506
GSE30999	0.9321	0.9726	0.9539	0.9768	0.8988	0.9653	0.8988	0.9404	0.8863	0.9599	0.9599	0.9731
GSE55447	0.3275	0.1875	0.3625	0.1025	0.2000	0.2775	0.2500	-0.0075	0.1400	-0.0075	0.5000	0.3275
GSE19429	0.3521	0.3235	0.5313	0.1807	0.4480	0.4689	0.1514	0.2630	0.1069	0.6025	0.4856	0.6828
GSE9006	0.8039	0.5167	0.6672	0.6576	0.5752	0.6846	0.4379	0.4405	0.5267	0.6413	0.6607	0.8207
GSE48350	0.9438	0.9099	0.8309	1.0000	1.0000	0.9481	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GSE5281	0.9895	0.9144	0.8058	0.9009	0.7618	0.9329	0.6788	0.7374	0.6619	0.7763	0.8603	0.9255
GSE35978	0.1290	0.2787	0.3321	0.2591	0.2303	0.2889	0.0644	0.1473	0.2064	0.0948	0.1380	0.3733
GSE53987	0.1226	0.1085	0.1153	0.0633	0.0597	0.0923	0.0854	0.0417	0.0286	0.0819	0.0124	0.0022
GSE12288	0.1724	0.1231	0.2211	0.0440	0.0883	0.2523	0.0855	0.2562	0.0591	0.1860	0.1700	0.0724
GSE15852	0.6630	0.7629	0.8633	0.8100	0.7237	0.8632	0.5590	0.6043	0.6103	0.6777	0.7399	0.8494
GSE42568	0.8405	0.7500	0.8000	0.8955	0.7455	1.0000	0.7955	0.6955	0.7955	0.5631	0.5909	0.8455
GSE29431	0.9417	0.9000	0.7917	0.9417	0.8417	0.9400	0.8417	0.7817	0.8417	0.8417	0.8417	1.0000
GSE18520	0.9400	0.9400	0.8400	0.9400	0.9400	1.0000	0.9400	0.9400	0.9400	0.9400	0.9400	1.0000
GSE19804	0.9463	0.9033	0.8923	0.9793	0.9083	0.9532	0.7937	0.8244	0.8020	0.8420	0.8770	0.9254
GSE100072	1.0000	0.9800	0.9400	0.9850	0.8920	0.9947	0.8920	0.8645	0.8920	0.8920	0.8920	0.9797
GSE68571	1.0000	1.0000	1.0000	0.9438	0.9438	1.0000	0.9438	0.9438	0.9438	0.9438	0.9438	1.0000
Average ± SEM	0.6860 ± 0.0648	0.6558 ± 0.0645	0.6570 ± 0.0570	0.6820 ± 0.0697	0.6330 ± 0.0618	0.7108 ± 0.0603	0.5473 ± 0.0638	0.5382 ± 0.0656	0.5475 ± 0.0644	0.5986 ± 0.0655	0.6200 ± 0.0643	0.7403 ± 0.0616

Table 73: Experiment 2: AUPRG by state-of-the-art classifiers, averaged across 10-fold cross-validation for each dataset. Last row contains the average across the datasets and the standard error of mean.

Data	Logistic	SVM	naïve Bayes	Bagged C4.5	Boosted C4.5	Random Forest	C4.5	RIPPER	PART	BRL DT	BRL DT-Beam	Bagged-BRL DT-BMC
GSE66360	0.1843	0.1100	0.1875	0.1012	0.2044	0.1138	0.2889	0.2475	0.2886	0.1675	0.1409	0.0985
GSE62646	0.0500	0.0000	0.0000	0.0466	0.0700	0.0440	0.0700	0.0701	0.0700	0.0684	0.0684	0.0377
GSE41861	0.1910	0.1731	0.2225	0.1499	0.1414	0.1339	0.2482	0.2254	0.2470	0.2492	0.2294	0.1071
GSE20881	0.1151	0.0523	0.2440	0.1313	0.2096	0.1578	0.2442	0.1724	0.2687	0.2059	0.1786	0.1209
GSE3365	0.0255	0.0551	0.1019	0.0831	0.0371	0.0807	0.1884	0.1590	0.1887	0.1642	0.1200	0.0649
GSE16879	0.0338	0.0393	0.0286	0.0265	0.0268	0.0486	0.0268	0.0539	0.0268	0.0257	0.0257	0.0310
GSE15245	0.4725	0.2286	0.2510	0.1461	0.2312	0.1649	0.3128	0.3079	0.2984	0.1910	0.2308	0.1615
GSE6613	0.3679	0.3773	0.3897	0.2656	0.4011	0.2486	0.4097	0.3620	0.4243	0.4566	0.5334	0.2495
GSE20295	0.3495	0.2011	0.4422	0.1955	0.3384	0.2451	0.3467	0.3915	0.3228	0.3234	0.3707	0.1929
GSE30999	0.0510	0.0176	0.0412	0.0298	0.0531	0.0400	0.0530	0.0354	0.0587	0.0552	0.0552	0.0347
GSE55447	0.3092	0.1767	0.1556	0.1875	0.3071	0.1627	0.3194	0.3348	0.3402	0.2943	0.1737	0.1640
GSE19429	0.0795	0.0600	0.0751	0.0799	0.0721	0.0590	0.1436	0.0850	0.1302	0.0708	0.0499	0.0498
GSE9006	0.1804	0.1571	0.1446	0.1348	0.1826	0.1312	0.2049	0.2162	0.1753	0.2111	0.1981	0.1167
GSE48350	0.0452	0.0429	0.1024	0.0006	0.0000	0.0612	0.0000	0.0000	0.0000	0.0001	0.0001	0.0011
GSE5281	0.0683	0.0434	0.1176	0.0898	0.1517	0.0848	0.1804	0.1510	0.1847	0.1717	0.1074	0.0882
GSE35978	0.3961	0.2990	0.3431	0.2427	0.3300	0.2109	0.3918	0.3057	0.3383	0.3237	0.3570	0.1929
GSE53987	0.3358	0.3079	0.3482	0.2445	0.3003	0.2459	0.2996	0.3101	0.3549	0.2711	0.3082	0.2509
GSE12288	0.4459	0.4455	0.3948	0.3047	0.4413	0.2689	0.4541	0.3469	0.4587	0.3319	0.3687	0.2720
GSE15852	0.2434	0.1389	0.1180	0.1086	0.1490	0.1046	0.2271	0.1998	0.2045	0.1929	0.1479	0.1060
GSE42568	0.0521	0.0327	0.0250	0.0239	0.0379	0.0244	0.0251	0.0417	0.0251	0.0650	0.0568	0.0319
GSE29431	0.0599	0.0143	0.0429	0.0248	0.0286	0.0275	0.0286	0.0429	0.0286	0.0279	0.0279	0.0161
GSE18520	0.0333	0.0167	0.0333	0.0176	0.0167	0.0165	0.0167	0.0167	0.0167	0.0159	0.0159	0.0123
GSE19804	0.0611	0.0583	0.0749	0.0464	0.0743	0.0593	0.1125	0.0980	0.1122	0.0992	0.0828	0.0693
GSE100072	0.0100	0.0091	0.0273	0.0258	0.0555	0.0264	0.0556	0.0743	0.0556	0.0726	0.0719	0.0300
GSE68571	0.0106	0.0000	0.0000	0.0140	0.0111	0.0153	0.0111	0.0111	0.0111	0.0106	0.0106	0.0142
Average ± SEM	0.1669 ± 0.0304	0.1223 ± 0.0252	0.1565 ± 0.0274	0.1089 ± 0.0178	0.1548 ± 0.0267	0.1110 ± 0.0164	0.1864 ± 0.0283	0.1704 ± 0.0254	0.1852 ± 0.0284	0.1626 ± 0.0248	0.1572 ± 0.0278	0.1006 ± 0.0163

Table 74: Experiment 2: Brier scores by state-of-the-art classifiers, averaged across 10-fold cross-validation for each dataset. Last row contains the average across the datasets and the standard error of mean.

BIBLIOGRAPHY

[Antonarakis and Beckmann, 2006] Antonarakis, S. E. and Beckmann, J. S. (2006). Mendelian disorders deserve more attention. *Nature Reviews Genetics*, 7(4):277.

[Ashburner et al., 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25.

[Aslam et al., 2007] Aslam, J. A., Popa, R. A., and Rivest, R. L. (2007). On estimating the size and confidence of a statistical audit. *EVT*, 7:8.

[Balasubramanian and Gopalakrishnan, 2018] Balasubramanian, J. B. and Gopalakrishnan, V. (2018). Tunable structure priors for bayesian rule learning for knowledge integrated biomarker discovery. *World journal of clinical oncology*, 9(5):98.

[Balasubramanian et al., 2019] Balasubramanian, J. B., Kip, K. E., Reis, S. E., and Gopalakrishnan, V. (2019). Knowledge discovery with bayesian rule learning for actionable biomedicine. *bioRxiv*, page 785279.

[Balasubramanian et al., 2014] Balasubramanian, J. B., Visweswaran, S., Cooper, G. F., and Gopalakrishnan, V. (2014). Selective model averaging with bayesian rule learning for predictive biomedicine. *AMIA Summits on Translational Science Proceedings*, 2014:17.

[Bambs et al., 2011] Bambs, C., Kip, K. E., Dinga, A., Mulukutla, S. R., Aiyer, A. N., and Reis, S. E. (2011). Low prevalence of ?ideal cardiovascular health? in a community-based population. *Circulation*, 123(8):850–857.

[Barrett et al., 2012] Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., et al. (2012). Ncbi geo: archive for functional genomics data setsupdate. *Nucleic acids research*, 41(D1):D991–D995.

[Benjamin et al., 2019] Benjamin, E., Muntner, P., Alonso, A., Bittencourt, M., Callaway, C., Carson, A., Chamberlain, A., Chang, A., Cheng, S., Das, S., et al. (2019). Heart disease and stroke statistics2019 update. *Circulation*, 139(10).

[Benson et al., 2018] Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Ostell, J., Pruitt, K. D., and Sayers, E. W. (2018). Genbank. *Nucleic acids research*, 46(D1):D41–D47.

[Bergstra et al., 2015] Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., and Cox, D. D. (2015). Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1):014008.

[Bernardo and Smith, 2009] Bernardo, J. M. and Smith, A. F. (2009). *Bayesian theory*, volume 405. John Wiley & Sons.

[Bethune et al., 2010] Bethune, G., Bethune, D., Ridgway, N., and Xu, Z. (2010). Epidermal growth factor receptor (egfr) in lung cancer: an overview and update. *Journal of thoracic disease*, 2(1):48.

[Bigbee et al., 2012] Bigbee, W. L., Gopalakrishnan, V., Weissfeld, J. L., Wilson, D. O., Dacic, S., Lokshin, A. E., and Siegfried, J. M. (2012). A multiplexed serum biomarker immunoassay panel discriminates clinical lung cancer patients from high-risk individuals found to be cancer-free by ct screening. *Journal of Thoracic Oncology*, 7(4):698–708.

[Bøttcher, 2004] Bøttcher, S. G. (2004). *Learning Bayesian networks with mixed variables*. Citeseer.

[Boutilier et al., 1996] Boutilier, C., Friedman, N., Goldszmidt, M., and Koller, D. (1996). Context-specific independence in bayesian networks. In *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*, pages 115–123. Morgan Kaufmann Publishers Inc.

[Box, 1976] Box, G. E. (1976). Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799.

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[Brier, 1950] Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

[Buntine, 1991] Buntine, W. (1991). Theory refinement on bayesian networks. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers Inc.

[Burke, 2016] Burke, H. B. (2016). Predicting clinical outcomes using molecular biomarkers. *Biomarkers in cancer*, 8:BIC–S33380.

[Burley et al., 2018] Burley, S. K., Berman, H. M., Christie, C., Duarte, J. M., Feng, Z., Westbrook, J., Young, J., and Zardecki, C. (2018). Rcsb protein data bank: Sustaining a

living digital data resource that enables breakthroughs in scientific research and biomedical education. *Protein Science*, 27(1):316–330.

[Butte and Chen, 2006] Butte, A. J. and Chen, R. (2006). Finding disease-related genomic experiments within an international repository: first steps in translational bioinformatics. In *AMIA annual symposium proceedings*, volume 2006, page 106. American Medical Informatics Association.

[Castelo and Siebes, 2000] Castelo, R. and Siebes, A. (2000). Priors on network structures. biasing the search for bayesian networks. *International Journal of Approximate Reasoning*, 24(1):39–57.

[Catalona et al., 1991] Catalona, W. J., Smith, D. S., Ratliff, T. L., Dodds, K. M., Coplen, D. E., Yuan, J. J., Petros, J. A., and Andriole, G. L. (1991). Measurement of prostate-specific antigen in serum as a screening test for prostate cancer. *New England Journal of Medicine*, 324(17):1156–1161.

[Chen et al., 2015] Chen, T., He, T., Benesty, M., Khotilovich, V., and Tang, Y. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, pages 1–4.

[Chickering et al., 1997] Chickering, D. M., Heckerman, D., and Meek, C. (1997). A bayesian approach to learning bayesian networks with local structure. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 80–89. Morgan Kaufmann Publishers Inc.

[Clearwater and Provost, 1990] Clearwater, S. H. and Provost, F. J. (1990). Rl4: A tool for knowledge-based induction. In *Tools for Artificial Intelligence, 1990., Proceedings of the 2nd International IEEE Conference on*, pages 24–30. IEEE.

[Cohen, 1995] Cohen, W. W. (1995). Fast effective rule induction. In *Machine Learning Proceedings 1995*, pages 115–123. Elsevier.

[Collins et al., 2003] Collins, F. S., Morgan, M., and Patrinos, A. (2003). The human genome project: lessons from large-scale biology. *Science*, 300(5617):286–290.

[Consortium et al., 2018] Consortium, U. et al. (2018). Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 46(5):2699.

[Cooper and Herskovits, 1992] Cooper, G. F. and Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347.

[Cortés et al., 2016] Cortés, X., Serratosa, F., and Riesen, K. (2016). On the relevance of local neighbourhoods for greedy graph edit distance. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 121–131. Springer.

[Cox, 1972] Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202.

[da Cunha Santos et al., 2011] da Cunha Santos, G., Shepherd, F. A., and Tsao, M. S. (2011). Egfr mutations and lung cancer. *Annual Review of Pathology: Mechanisms of Disease*, 6:49–69.

[De Dombal et al., 1972] De Dombal, F., Leaper, D., Staniland, J. R., McCann, A., and Horrocks, J. C. (1972). Computer-aided diagnosis of acute abdominal pain. *Br Med J*, 2(5804):9–13.

[DeGroot and Fienberg, 1983] DeGroot, M. H. and Fienberg, S. E. (1983). The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22.

[Demšar, 2006] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

[Domingos, 2000] Domingos, P. (2000). Bayesian averaging of classifiers and the overfitting problem. In *ICML*, volume 2000, pages 223–230.

[Domingos, 1997] Domingos, P. M. (1997). Why does bagging work? a bayesian account and its implications. In *KDD*, pages 155–158. Citeseer.

[Dagostino et al., 2008] Dagostino, R. B., Vasan, R. S., Pencina, M. J., Wolf, P. A., Cobain, M., Massaro, J. M., and Kannel, W. B. (2008). General cardiovascular risk profile for use in primary care. *Circulation*, 117(6):743–753.

[Edgar et al., 2002] Edgar, R., Domrachev, M., and Lash, A. E. (2002). Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210.

[Efron, 1992] Efron, B. (1992). Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer.

[Ein-Dor et al., 2006] Ein-Dor, L., Zuk, O., and Domany, E. (2006). Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proceedings of the National Academy of Sciences*, 103(15):5923–5928.

[Esfandiari et al., 2014] Esfandiari, N., Babavalian, M. R., Moghadam, A.-M. E., and Tabar, V. K. (2014). Knowledge discovery in medicine: Current issue and future trend. *Expert Systems with Applications*, 41(9):4434–4463.

[Fabris and Freitas, 2000] Fabris, C. C. and Freitas, A. A. (2000). Discovering surprising patterns by detecting occurrences of simpsons paradox. In *Research and Development in Intelligent Systems XVI*, pages 148–160. Springer.

[Fawcett, 2006] Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

[Fayyad et al., 1996a] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996a). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37.

[Fayyad et al., 1996b] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (1996b). *Advances in knowledge discovery and data mining*, volume 21. AAAI press Menlo Park.

[Fayyad et al., 1993] Fayyad et al. (1993). *Multi-interval discretization of continuous-valued attributes for classification learning*, volume 2. International Joint Conferences on Artificial Intelligence.

[Flach and Kull, 2015] Flach, P. and Kull, M. (2015). Precision-recall-gain curves: Pr analysis done right. In *Advances in neural information processing systems*, pages 838–846.

[Forman and Scholz, 2010] Forman, G. and Scholz, M. (2010). Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *ACM SIGKDD Explorations Newsletter*, 12(1):49–57.

[Frank et al., 2016] Frank, E., Hall, M. A., and Witten, I. H. (2016). *The WEKA workbench*. Morgan Kaufmann.

[Frank and Witten, 1998] Frank, E. and Witten, I. H. (1998). Generating accurate rule sets without global optimization. *International Conference on Machine Learning*.

[Freund et al., 1999] Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

[Freund et al., 1996] Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156. Bari, Italy.

[Friedman, 1937] Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701.

[Friedman and Goldszmidt, 1998] Friedman, N. and Goldszmidt, M. (1998). Learning bayesian networks with local structure. In *Learning in graphical models*, pages 421–459. Springer.

[Fuchsberger et al., 2016] Fuchsberger, C., Flannick, J., Teslovich, T. M., Mahajan, A., Agarwala, V., Gaulton, K. J., Ma, C., Fontanillas, P., Moutsianas, L., McCarthy, D. J., et al. (2016). The genetic architecture of type 2 diabetes. *Nature*, 536(7614):41.

[Fürnkranz et al., 2012] Fürnkranz, J., Gamberger, D., and Lavrač, N. (2012). *Foundations of rule learning*. Springer Science & Business Media.

[Ganchev et al., 2011] Ganchev, P., Malehorn, D., Bigbee, W. L., and Gopalakrishnan, V. (2011). Transfer learning of classification rules for biomarker discovery and verification from molecular profiling studies. *Journal of biomedical informatics*, 44:S17–S23.

[García et al., 2010] García, S., Fernández, A., Luengo, J., and Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064.

[Geng and Hamilton, 2006] Geng, L. and Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, 38(3):9.

[Ginsburg and Willard, 2009] Ginsburg, G. S. and Willard, H. F. (2009). Genomic and personalized medicine: foundations and applications. *Translational research*, 154(6):277–287.

[Goff et al., 2014] Goff, D. C., Lloyd-Jones, D. M., Bennett, G., Coady, S., DAgostino, R. B., Gibbons, R., Greenland, P., Lackland, D. T., Levy, D., ODonnell, C. J., et al. (2014). 2013 acc/aha guideline on the assessment of cardiovascular risk. *Circulation*, 129(25 suppl 2):S49–S73.

[Goldstein, 1992] Goldstein, D. J. (1992). Beneficial health effects of modest weight loss. *International journal of obesity and related metabolic disorders: journal of the International Association for the Study of Obesity*, 16(6):397–415.

[Goossens et al., 2015] Goossens, N., Nakagawa, S., Sun, X., and Hoshida, Y. (2015). Cancer biomarker discovery and validation. *Translational cancer research*, 4(3):256.

[Gopalakrishnan et al., 2006] Gopalakrishnan, V., Ganchev, P., Ranganathan, S., and Bowser, R. (2006). Rule learning for disease-specific biomarker discovery from clinical proteomic mass spectra. In *International Workshop on Data Mining for Biomedical Applications*, pages 93–105. Springer.

[Gopalakrishnan et al., 2010] Gopalakrishnan, V., Lustgarten, J. L., Visweswaran, S., and Cooper, G. F. (2010). Bayesian rule learning for biomedical data mining. *Bioinformatics*, 26(5):668–675.

[Gopalakrishnan et al., 2004] Gopalakrishnan, V., Williams, E., Ranganathan, S., Bowser, R., Cudkowic, M. E., Novelli, M., Lattazi, W., Gambotto, A., and Day, B. W. (2004). Proteomic data mining challenges in identification of disease-specific biomarkers from variable resolution mass spectra. In *Proceedings of SIAM Bioinformatics Workshop*, volume 10. FL: Lake Buena Vista.

[Graham, 2009] Graham, J. W. (2009). Missing data analysis: Making it work in the real world. *Annual review of psychology*, 60:549–576.

[Group et al., 2001] Group, B. D. W., Atkinson Jr, A. J., Colburn, W. A., DeGruttola, V. G., DeMets, D. L., Downing, G. J., Hoth, D. F., Oates, J. A., Peck, C. C., Schooley, R. T., et al. (2001). Biomarkers and surrogate endpoints: preferred definitions and conceptual framework. *Clinical pharmacology & therapeutics*, 69(3):89–95.

[Günther et al., 2012] Günther, O. P., Chen, V., Freue, G. C., Balshaw, R. F., Tebbutt, S. J., Hollander, Z., Takhar, M., McMaster, W. R., McManus, B. M., Keown, P. A., et al. (2012). A computational pipeline for the development of multi-marker bio-signature panels and ensemble classifiers. *BMC bioinformatics*, 13(1):326.

[Hanley and McNeil, 1982] Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

[Harary and Palmer, 2014] Harary, F. and Palmer, E. M. (2014). *Graphical enumeration*. Elsevier.

[Hastie et al., 2005] Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85.

[Heckerman, 1990] Heckerman, D. (1990). Probabilistic similarity networks. *Networks*, 20(5):607–636.

[Heckerman, 2008] Heckerman, D. (2008). A tutorial on learning with bayesian networks. In *Innovations in Bayesian networks*, pages 33–82. Springer.

[Hewett et al., 2002] Hewett, M., Oliver, D. E., Rubin, D. L., Easton, K. L., Stuart, J. M., Altman, R. B., and Klein, T. E. (2002). Pharmgkb: the pharmacogenetics knowledge base. *Nucleic acids research*, 30(1):163–165.

[Hoeting et al., 1999] Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401.

[Iman and Davenport, 1980] Iman, R. L. and Davenport, J. M. (1980). Approximations of the critical region of the fbietkan statistic. *Communications in Statistics-Theory and Methods*, 9(6):571–595.

[Jabbari et al., 2017] Jabbari, F., Naeini, M. P., and Cooper, G. F. (2017). Obtaining accurate probabilistic causal inference by post-processing calibration. *arXiv preprint arXiv:1712.08626*.

[Jabbari et al., 2018] Jabbari, F., Visweswaran, S., and Cooper, G. F. (2018). Instance-specific bayesian network structure learning. *Proceedings of machine learning research*, 72:169.

[Japkowicz and Stephen, 2002] Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449.

[Jeffreys, 1998] Jeffreys, H. (1998). *The theory of probability*. OUP Oxford.

[Ji et al., 2017] Ji, J., Yang, C., Liu, J., Liu, J., and Yin, B. (2017). A comparative study on swarm intelligence for structure learning of bayesian networks. *Soft Computing*, 21(22):6713–6738.

[John and Langley, 1995] John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc.

[Kanehisa et al., 2011] Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., and Tanabe, M. (2011). Kegg for integration and interpretation of large-scale molecular data sets. *Nucleic acids research*, 40(D1):D109–D114.

[Kégl, 2013] Kégl, B. (2013). The return of adaboost. mh: multi-class hamming trees. *arXiv preprint arXiv:1312.6086*.

[Koller and Friedman, 2009] Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

[Koller and Sahami, 1996] Koller, D. and Sahami, M. (1996). Toward optimal feature selection. Technical report, Stanford InfoLab.

[Koscielny, 2010] Koscielny, S. (2010). Why most gene expression signatures of tumors have not been useful in the clinic. *Science translational medicine*, 2(14):14ps2–14ps2.

[Kuperman et al., 1991] Kuperman, G. J., Maack, B., Bauer, K., and Gardner, R. (1991). Innovations and research review: the impact of the help computer system on the lds hospital paper medical record. *Topics in health record management*, 12(2):76–85.

[Lavrac and Dzeroski, 2001] Lavrac, N. and Dzeroski, S. (2001). Relational data mining.

[le Cessie and van Houwelingen, 1992] le Cessie, S. and van Houwelingen, J. (1992). Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201.

[Lenfant, 2003] Lenfant, C. (2003). Clinical research to clinical practicelost in translation? *New England Journal of Medicine*, 349(9):868–874.

[Lesk, 2019] Lesk, A. (2019). *Introduction to bioinformatics*. Oxford University Press.

[Lesk, 2017] Lesk, A. M. (2017). *Introduction to genomics*. Oxford University Press.

[Levy-Lahad and Friedman, 2007] Levy-Lahad, E. and Friedman, E. (2007). Cancer risks among brca1 and brca2 mutation carriers. *British journal of cancer*, 96(1):11.

[Liu et al., 1999] Liu, B., Hsu, W., Mun, L.-F., and Lee, H.-Y. (1999). Finding interesting patterns using user expectations. *IEEE Transactions on Knowledge and Data Engineering*, 11(6):817–832.

[Lu et al., 2010] Lu, T.-P., Tsai, M.-H., Lee, J.-M., Hsu, C.-P., Chen, P.-C., Lin, C.-W., Shih, J.-Y., Yang, P.-C., Hsiao, C. K., Lai, L.-C., et al. (2010). Identification of a novel biomarker, sema5a, for non–small cell lung carcinoma in nonsmoking women. *Cancer Epidemiology and Prevention Biomarkers*, pages 1055–9965.

[Lustgarten et al., 2017] Lustgarten, J. L., Balasubramanian, J. B., Visweswaran, S., and Gopalakrishnan, V. (2017). Learning parsimonious classification rules from gene expression data using bayesian networks with local structure. *Data*, 2(1):5.

[Lustgarten et al., 2008] Lustgarten, J. L., Gopalakrishnan, V., Grover, H., and Visweswaran, S. (2008). Improving classification performance with discretization on biomedical datasets. In *AMIA annual symposium proceedings*, volume 2008, page 445. American Medical Informatics Association.

[Mahmood et al., 2014] Mahmood, S. S., Levy, D., Vasan, R. S., and Wang, T. J. (2014). The framingham heart study and the epidemiology of cardiovascular disease: a historical perspective. *The lancet*, 383(9921):999–1008.

[Mani and Cooper, 2004] Mani, S. and Cooper, G. F. (2004). Causal discovery using a bayesian local causal discovery algorithm. In *Medinfo*, pages 731–735.

[Mardis, 2011] Mardis, E. R. (2011). A decades perspective on dna sequencing technology. *Nature*, 470(7333):198.

[Margaritis and Thrun, 2000] Margaritis, D. and Thrun, S. (2000). Bayesian network induction via local neighborhoods. In *Advances in neural information processing systems*, pages 505–511.

[Martinez-Cantin, 2014] Martinez-Cantin, R. (2014). Bayesopt: A bayesian optimization library for nonlinear optimization, experimental design and bandits. *The Journal of Machine Learning Research*, 15(1):3735–3739.

[McGarry, 2005] McGarry, K. (2005). A survey of interestingness measures for knowledge discovery. *The knowledge engineering review*, 20(1):39–61.

[Mendis et al., 2011] Mendis, S., Puska, P., Norrving, B., Organization, W. H., et al. (2011). *Global atlas on cardiovascular disease prevention and control*. Geneva: World Health Organization.

[Micheel et al., 2012] Micheel, C. M., Nass, S. J., Omenn, G. S., et al. (2012). *Evolution of translational omics: lessons learned and the path forward*. National Academies Press.

[Minka, 2000] Minka, T. P. (2000). Bayesian model averaging is not model combination. *Available electronically at http://www. stat. cmu. edu/minka/papers/bma. html*, pages 1–2.

[Mitchell et al., 1997] Mitchell, T. M. et al. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):174–176.

[Monteith et al., 2011] Monteith, K., Carroll, J. L., Seppi, K., and Martinez, T. (2011). Turning bayesian model averaging into bayesian model combination. In *The 2011 International Joint Conference on Neural Networks*, pages 2657–2663. IEEE.

[Mukherjee and Speed, 2008] Mukherjee, S. and Speed, T. P. (2008). Network inference using informative priors. *Proceedings of the National Academy of Sciences*, 105(38):14313–14318.

[Naeini et al., 2015] Naeini, M. P., Cooper, G. F., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, pages 2901–2907.

[Ogoe et al., 2015] Ogoe, H. A., Visweswaran, S., Lu, X., and Gopalakrishnan, V. (2015). Knowledge transfer via classification rules using functional mapping for integrative modeling of gene expression data. *BMC bioinformatics*, 16(1):226.

[Olson et al., 2017] Olson, R. S., La Cava, W., Mustahsan, Z., Varik, A., and Moore, J. H. (2017). Data-driven advice for applying machine learning to bioinformatics problems. *arXiv preprint arXiv:1708.05070*.

[Organization et al., 2017] Organization, W. H. et al. (2017). Global health observatory (gho) data: Top 10 causes of death.

[Osheroff et al., 2007] Osheroff, J. A., Teich, J. M., Middleton, B., Steen, E. B., Wright, A., and Detmer, D. E. (2007). A roadmap for national action on clinical decision support. *Journal of the American medical informatics association*, 14(2):141–145.

[Paik et al., 2004] Paik, S., Shak, S., Tang, G., Kim, C., Baker, J., Cronin, M., Baehner, F. L., Walker, M. G., Watson, D., Park, T., et al. (2004). A multigene assay to predict recurrence of tamoxifen-treated, node-negative breast cancer. *New England Journal of Medicine*, 351(27):2817–2826.

[Parkinson et al., 2014] Parkinson, D. R., McCormack, R. T., Keating, S. M., Gutman, S. I., Hamilton, S. R., Mansfield, E. A., Piper, M. A., DeVerka, P., Frueh, F. W., Jessup, J. M., et al. (2014). Evidence of clinical utility: an unmet need in molecular diagnostics for patients with cancer.

[Pearl, 2009] Pearl, J. (2009). *Causality*. Cambridge university press.

[Pearl, 2014] Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

[Piatetsky-Shapiro and Matheus, 1994] Piatetsky-Shapiro, G. and Matheus, C. J. (1994). The interestingness of deviations. In *Proceedings of the AAAI-94 workshop on Knowledge Discovery in Databases*, volume 1, pages 25–36.

[Platt, 1999] Platt, J. C. (1999). 12 fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods*, pages 185–208.

[Polikar, 2006] Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45.

[Poole, 1993] Poole, D. (1993). Probabilistic horn abduction and bayesian networks. *Artificial intelligence*, 64(1):81–129.

[Poulter, 1999] Poulter, N. (1999). Coronary heart disease is a multifactorial disease. *American Journal of Hypertension*, 12(S6):92S–95S.

[Quinlan, 2014] Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

[Ranganathan et al., 2005] Ranganathan, S., Williams, E., Ganchev, P., Gopalakrishnan, V., Lacomis, D., Urbinelli, L., Newhall, K., Cudkowicz, M. E., Brown, R. H., and Bowser, R. (2005). Proteomic profiling of cerebrospinal fluid identifies biomarkers for amyotrophic lateral sclerosis. *Journal of neurochemistry*, 95(5):1461–1471.

[Ridker et al., 2007] Ridker, P. M., Buring, J. E., Rifai, N., and Cook, N. R. (2007). Development and validation of improved algorithms for the assessment of global cardiovascular risk in women: the reynolds risk score. *Jama*, 297(6):611–619.

[Ridker et al., 2008] Ridker, P. M., Paynter, N. P., Rifai, N., Gaziano, J. M., and Cook, N. R. (2008). C-reactive protein and parental history improve global cardiovascular risk prediction: the reynolds risk score for men. *Circulation*, 118(22):2243–2251.

[Riesen, 2015] Riesen, K. (2015). Structural pattern recognition with graph edit distance. *Advances in Computer Vision and Pattern Recognition. Springer, Cham*.

[Riordan et al., 1989] Riordan, J. R., Rommens, J. M., Kerem, B.-s., Alon, N., Rozmahel, R., Grzelczak, Z., Zielenski, J., Lok, S., Plavsic, N., Chou, J.-L., et al. (1989). Identification of the cystic fibrosis gene: cloning and characterization of complementary dna. *Science*, 245(4922):1066–1073.

[Rissanen, 1978] Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465–471.

[Ryberg et al., 2010] Ryberg, H., An, J., Darko, S., Lustgarten, J. L., Jaffa, M., Gopalakrishnan, V., Lacomis, D., Cudkowicz, M., and Bowser, R. (2010). Discovery and verification of amyotrophic lateral sclerosis biomarkers by proteomics. *Muscle & nerve*, 42(1):104–111.

[Sahar, 2002] Sahar, S. (2002). On incorporating subjective interestingness into the mining process. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 681–684. IEEE.

[Selleck et al., 2017] Selleck, M. J., Senthil, M., and Wall, N. R. (2017). Making meaningful clinical use of biomarkers. *Biomarker insights*, 12:1177271917715236.

[Seni and Elder, 2010] Seni, G. and Elder, J. F. (2010). Ensemble methods in data mining: improving accuracy through combining predictions. *Synthesis lectures on data mining and knowledge discovery*, 2(1):1–126.

[Shah and Newgard, 2015] Shah, S. H. and Newgard, C. B. (2015). Integrated metabolomics and genomics. *Circulation: Cardiovascular Genetics*, 8(2):410–419.

[Shaw et al., 2013] Shaw, A. T., Kim, D.-W., Nakagawa, K., Seto, T., Crinó, L., Ahn, M.-J., De Pas, T., Besse, B., Solomon, B. J., Blackhall, F., et al. (2013). Crizotinib versus chemotherapy in advanced alk-positive lung cancer. *New England Journal of Medicine*, 368(25):2385–2394.

[Shigematsu et al., 2005] Shigematsu, H., Lin, L., Takahashi, T., Nomura, M., Suzuki, M., Wistuba, I. I., Fong, K. M., Lee, H., Toyooka, S., Shimizu, N., et al. (2005). Clinical and biological features associated with epidermal growth factor receptor gene mutations in lung cancers. *Journal of the National Cancer Institute*, 97(5):339–346.

[Shortliffe, 1977] Shortliffe, E. H. (1977). Mycin: A knowledge-based computer program applied to infectious diseases. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 66. American Medical Informatics Association.

[Shortliffe and Cimino, 2013] Shortliffe, E. H. and Cimino, J. J. (2013). *Biomedical informatics: computer applications in health care and biomedicine*. Springer Science & Business Media.

[Silberschatz and Tuzhilin, 1996] Silberschatz, A. and Tuzhilin, A. (1996). What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and data engineering*, 8(6):970–974.

[Soufi et al., 2018] Soufi, M. D., Samad-Soltani, T., Vahdati, S. S., and Rezaei-Hachesu, P. (2018). Decision support system for triage management: A hybrid approach using rule-based reasoning and fuzzy logic. *International journal of medical informatics*, 114:35–44.

[Steinberg, 1990] Steinberg, W. (1990). The clinical utility of the ca 19-9 tumor-associated antigen. *American Journal of Gastroenterology*, 85(4).

[Subramanian et al., 2005] Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550.

[Sung et al., 2003] Sung, N. S., Crowley Jr, W. F., Genel, M., Salber, P., Sandy, L., Sherwood, L. M., Johnson, S. B., Catanese, V., Tilson, H., Getz, K., et al. (2003). Central challenges facing the national clinical research enterprise. *Jama*, 289(10):1278–1287.

[Tsamardinos et al., 2003] Tsamardinos, I., Aliferis, C. F., and Statnikov, A. (2003). Time and sample efficient discovery of markov blankets and direct causal relations. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 673–678. ACM.

[Vonsattel and DiFiglia, 1998] Vonsattel, J. P. G. and DiFiglia, M. (1998). Huntington disease. *Journal of neuropathology and experimental neurology*, 57(5):369.

[Weinstein et al., 2013] Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R. M., Ozenberger, B. A., Ellrott, K., Shmulevich, I., Sander, C., Stuart, J. M., Network, C. G. A. R., et al. (2013). The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113.

[Wilcoxon, 1992] Wilcoxon, F. (1992). Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer.

[Wishart et al., 2017] Wishart, D. S., Feunang, Y. D., Marcu, A., Guo, A. C., Liang, K., Vázquez-Fresno, R., Sajed, T., Johnson, D., Li, C., Karu, N., et al. (2017). Hmdb 4.0: the human metabolome database for 2018. *Nucleic acids research*, 46(D1):D608–D617.

[Yang et al., 2010] Yang, P., Hwa Yang, Y., B Zhou, B., and Y Zomaya, A. (2010). A review of ensemble methods in bioinformatics. *Current Bioinformatics*, 5(4):296–308.

[Yeung et al., 2005] Yeung, K. Y., Bumgarner, R. E., and Raftery, A. E. (2005). Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. *Bioinformatics*, 21(10):2394–2402.

[Zaidi et al., 2014] Zaidi, A. H., Gopalakrishnan, V., Kasi, P. M., Zeng, X., Malhotra, U., Balasubramanian, J., Visweswaran, S., Sun, M., Flint, M. S., Davison, J. M., et al. (2014). Evaluation of a 4-protein serum biomarker panelbiglycan, annexin-a 6, myeloperoxidase, and protein s 100-a 9 (b-amp)for the detection of esophageal adenocarcinoma. *Cancer*, 120(24):3902–3913.

[Zeng et al., 2011] Zeng, X., Hood, B. L., Zhao, T., Conrads, T. P., Sun, M., Gopalakrishnan, V., Grover, H., Day, R. S., Weissfeld, J. L., Wilson, D. O., et al. (2011). Lung cancer serum biomarker discovery using label-free liquid chromatography-tandem mass spectrometry. *Journal of Thoracic Oncology*, 6(4):725–734.